



Processo de Seleção de Participantes numa Plataforma de Testes de Usabilidade

SOFIA ALEXANDRA MOREIRA CANELAS

Junho de 2023

Participant Selection Process in a Usability Testing Platform

Sofia Alexandra Moreira Canelas

Dissertation for obtaining the Master's Degree in
Informatics Engineering, Area of Specialization in
Graphic Systems and Multimedia

Advisor: Filipe Pacheco

Porto, June 2023

Declaration of Integrity

I declare that I have conducted this academic work with integrity.

I have not plagiarized or applied any form of misuse of information or falsification of results throughout the process that led to its elaboration.

Therefore, the work presented in this document is original and my authorship, not having previously been used for no other purpose.

I also declare that I am fully aware of the P.PORTO Code of Ethical Conduct.

ISEP, Porto, June 30 2023

Sofia Alexandra MOREIRA Canelas

Abstract

With the exponential growth of web and mobile applications, the practice of usability tests began to be more prevalent in different organizations and revealed a positive impact on the empathy of their products with the intended users. Skipping this step can cause major usability issues on the final product, as it was not tested with its users before it was launched.

Despite companies realizing the importance of user testing and starting to make use of it, this task requires participants who are willing to participate in the tests, and this aspect is often a major obstacle for UX (User eXperience) researchers, not only in terms of their recruitment but also the suitability of the participants' attributes to the product to be created.

The aim of this project is to develop a web application that unifies the steps present in carrying out usability tests, from their creation to their execution, with a special focus on the selection of participants for them, which is the main problem to be solved. The hypothesis of research here is to confirm that using other platforms' APIs (Application Programming Interface) to recruit testers for usability tests is an efficient way of finding testers from a wide variety of market niches, which we will confirm or deny by the end of the project.

In the project's initial phase, it is intended to study the state of the art in order to bring to light deeper understandings of the UX field and usability testing, as well as current applications on the market for usability testing management and potential technologies for this project's development. Then, possible solutions for approaching the problem will be described, and a value analysis will be carried out focusing on strategic and business concepts for the project, namely its value proposition. In this stage, it is intended to choose a solution for the problem, taking into account factors such as time, adequation, and complexity of each one.

As a result of the previous steps, an initial concept for the UI (User Interface) of the chosen solution will be sketched, and its usability will be tested in order to find any issues and fix them before going on to the final design. After this phase, the solution will be implemented, and the technique for doing so, including the technology used, the code architecture and documentation, will be described. Accordingly, the project will be experimented and evaluated again after its implementation phase, in order to assess the effectiveness of its requirements' execution and potential problems the testers may have run into, which we will have to analyze and consider fixing at a later stage.

At last, thorough conclusions about the project will be held out, including those regarding the challenges and limitations faced, achieved objectives, and, lastly, the work to be developed in the future.

Keywords: User Experience, Usability Tests, UX Research, Testers, Selection, User Testing.

Resumo

Com o crescimento das aplicações *web* e *mobile*, a prática de testes de usabilidade começou a estar mais presente nas diversas organizações e revelou um impacto positivo na empatia dos seus produtos com os utilizadores a que se destinam. Saltar esta etapa pode causar grandes problemas de usabilidade no produto final, já que não foi testado com o público-alvo antes de ser lançado.

Apesar das empresas perceberem a importância de testes de usabilidade e começarem a fazer uso deste método, estes testes requerem participantes que estejam dispostos a participarem nos mesmos (conhecidos como avaliadores ou, em inglês, *testers*), sendo este aspeto, muitas vezes, um grande obstáculo para os investigadores na área de UX (*User eXperience*), não só no que toca à sua procura, como também à adequação dos atributos destes ao produto a criar.

O objetivo deste projeto é desenvolver uma aplicação *web* que unifique as etapas presentes na realização de testes de usabilidade, desde a sua criação à sua execução, com um foco especial na seleção de participantes para os mesmos, sendo este o problema a resolver. A aplicação terá as suas funcionalidades principais para criação e partilha dos testes, assim como seleção de participantes para os mesmos, em funcionamento, podendo, futuramente, ser desenvolvida na sua totalidade, isto é, com a presença de mais funcionalidades extra, úteis para a realização destes testes. A hipótese de investigação aqui é confirmar que usar APIs (*Application Programming Interface*) de outras plataformas para recrutar avaliadores para testes de usabilidade é uma forma eficiente de encontrar *testers* dos mais diversos nichos de mercado, o que iremos confirmar ou negar no final do projeto.

Na fase inicial do projeto, pretende-se estudar o estado da arte de forma a obter conhecimentos mais aprofundados sobre a área de UX e testes de usabilidade (respondendo-se, para isso, a questões como: o que é *UX Design*, o que é *usability testing* e qual a sua importância; que tipos de testes de usabilidade existem; quando se conduzem; quantos participantes são necessários; entre outras questões), bem como perceber as aplicações atualmente existentes no mercado para gestão destes testes (percebendo quem é a nossa concorrência e que funcionalidades é que oferecem nas suas aplicações, identificando-se também a nossa oportunidade), e, por fim, potenciais tecnologias para o desenvolvimento deste projeto (incluindo possíveis APIs de aplicações externas que poderão ser usadas para procura de avaliadores).

Em seguida, serão planeadas possíveis soluções para a abordagem do problema, incluindo-se uma solução completa (que não será implementada neste projeto devido à sua complexidade e tempo que exige), e três outras soluções simplificadas possíveis de serem implementadas, permitindo, caso haja algum imprevisto na fase de desenvolvimento que impossibilite a implementação da solução escolhida a tempo, a implementação de uma outra solução aqui especificada, visto que estas variam no número de requisitos que exigem.

Será ainda realizada uma análise de valor com foco nos conceitos estratégicos e de negócio do projeto, nomeadamente a oportunidade, uma análise SWOT (*Strengths, Weaknesses, Opportunities, Threats*), proposta de valor, modelo de negócio, e os requisitos a ter em conta para satisfazer os desejos dos clientes, através de uma técnica denominada *Quality Function Deployment* (QFD). Nesta etapa, pretende-se também escolher uma solução para o problema, levando em consideração fatores como tempo, adequação e complexidade de cada uma. Para isso, será utilizado o método de *Analytic Hierarchy Process* (AHP).

Como resultado das etapas anteriores, será desenhada a solução para o problema. Em primeiro lugar, serão definidos os tipos de utilizador que poderão registar-se na aplicação, esboçando também um mapa de navegação distinto para cada um destes. Em segundo lugar, será esboçado um conceito inicial para o UI (*User Interface*) da solução escolhida, onde será depois conduzido um teste de usabilidade – *System Usability Scale* (SUS) – para que se possam identificar e corrigir problemas antes da sua versão final. Na versão final do design será também construída uma identidade visual, escolhendo-se um nome para a aplicação e desenhando-se o seu logótipo, definindo-se também a paleta de cores e tipografia a utilizar, mantendo coerência ao longo das páginas da aplicação.

Após esta fase, a solução será implementada: será escolhida e fundamentada a tecnologia a utilizar; será descrito o processo de implementação, nomeadamente a definição de requisitos e a utilização de um sistema de controlo de versões; será explicada a arquitetura do código e a sua respetiva documentação, com foco em casos especiais do projeto; e, por último, será disponibilizada a aplicação final.

Assim, o projeto será novamente experimentado e avaliado após a sua fase de implementação, de forma a classificar-se o sucesso da implementação dos seus requisitos e potenciais problemas que os *testers* possam ter encontrado, os quais terão de ser analisados e ponderados para corrigir numa fase posterior. Para isto, será utilizado um QEF (*Quantitative Evaluation Framework*) e um formulário de *feedback*, por forma a obtermos uma avaliação quantitativa e qualitativa da aplicação desenvolvida.

Por último, serão tiradas conclusões aprofundadas sobre o projeto, incluindo as relativas aos desafios e limitações enfrentados, objetivos alcançados (focando-nos também em justificar a hipótese apresentada) e, por último, o trabalho a desenvolver-se no futuro.

Palavras-chave: *User Experience*, Testes de Usabilidade, *UX Research*, *Testers*, Seleção, *User Testing*.

Acknowledgments

To my advisor, Professor Filipe Pacheco, for all the help and precious suggestions provided throughout the writing of this dissertation.

To my parents, to whom I owe the attendance in this master's degree, for always helping me to overcome all the challenges I encounter. To my grandparents, for their unceasing assistance and for always keeping me well fed. To the rest of the family - godparents, uncles and cousins -, for understanding my absences during this period and for the support at all times.

To the best person I met during this journey and without whom I would not have imagined myself enjoying it so much, Tomás, thank you for your tireless help, for all your patience, attention, kindness and, above all, for being not only my boyfriend but also my best friend.

To my closest friends, Francisca, Inês and Tiago, for keeping me sane, for all the fun times we always have, and for consistently encouraging me to make the wisest decisions.

To my classmates, especially to the *Famiglia Unicorni* group - André Morais, André Gonçalves, Cárina Alas, Daniel Dias, Francisco Dias, Luís Sousa, Miguel Cabeleira, Narciso Correia and Vítor Neto -, for accompanying me at this stage, for trusting me to make the final design decisions in all the projects we carried out, for all the fun hangouts and meetings we had, and for the very special group that was formed thanks to this master's degree, which I hope to be able to keep with me for the rest of my life.

Finally, to all the professors with whom I came across throughout my academic years, for contributing to my education and to my formation as a designer and, now, informatics engineer.

Table of Contents

1	Introduction	25
1.1	Initial Context.....	25
1.2	Problem	26
1.3	Objectives	26
1.4	Motivation and Hypothesis.....	27
1.5	Work Plan	28
1.6	Document Structure	28
2	State of Art	31
2.1	Context.....	31
2.1.1	User Experience Design	31
2.1.2	Usability Testing.....	32
2.1.3	Participant Selection in Usability Testing.....	40
2.2	Usability Testing Management Applications	45
2.3	Technology	47
2.3.1	Criteria of Choice	48
2.3.2	Technology for the Solution's Development.....	49
2.3.3	Technology for the Participant Selection Process.....	52
3	Solution Planning.....	55
3.1	Complete Solution.....	55
3.2	Possible Approaches	56
3.2.1	Solution 01: Complete Solution (but not all features work)	56
3.2.2	Solution 02: Simplified Solution - UX Researcher and Participant.....	57
3.2.3	Solution 03: Simplified Solution - UX Researcher Only	57
3.3	Advantages and Disadvantages	57
4	Value Analysis.....	59
4.1	Value Analysis Definition	59
4.2	Innovation Process	60
4.2.1	Opportunity Identification.....	61
4.2.2	Opportunity Analysis.....	62
4.2.3	Idea Generation and Enrichment	64
4.2.4	Idea Selection	64
4.2.5	Concept Definition	69
4.3	Solution's Value	70
4.3.1	Value for the Customer and Perceived Value	70
4.3.2	Value Proposition.....	71
4.3.3	Business Model	72

4.4	Quality Function Deployment	73
5	Solution Design.....	75
5.1	Solution Chosen.....	75
5.2	Navigation Map.....	75
5.3	Design Requirements.....	76
5.4	User Interface: First Draft	76
5.5	User Interface: Final Version	77
5.5.1	Visual Identity.....	77
5.5.2	Final User Interface	78
6	Solution Implementation	79
6.1	Technology Used.....	79
6.1.1	Next.js	80
6.1.2	PostgreSQL.....	80
6.2	Implementation Process.....	81
6.2.1	Requirements Review and Prioritization	82
6.2.2	Version Control System.....	82
6.3	Code Architecture	83
6.3.1	Folder Structure	83
6.3.2	Components	84
6.3.3	Pages	85
6.3.4	Database	87
6.4	Code Documentation.....	90
6.4.1	Queries Explanation.....	91
6.4.2	Create Test	93
6.4.3	Find Testers	94
6.4.4	Testers Search API	95
6.5	Final Project.....	98
7	Experimentation and Evaluation	99
7.1	Evaluation Methodology	99
7.1.1	Before Implementation Stage - System Usability Scale.....	99
7.1.2	After Implementation Stage - Quantitative Evaluation Framework	100
7.1.3	After Implementation Stage - Feedback Survey	102
7.2	Results Analysis	102
7.2.1	System Usability Scale.....	102
7.2.2	Quantitative Evaluation Framework	104
7.2.3	Feedback Survey.....	106
8	Conclusions	107
8.1	Challenges and Limitations	107
8.2	Achieved Objectives	108

8.3 Future Work	110
References	111
Attachments	121
Attachment A	121
Attachment B	125
Attachment C	125
Attachment D	126
Attachment E	130
Attachment F	130
Attachment G	131
Attachment H	131
Attachment I	132
Attachment J	134
Attachment K	135
Attachment L	142

List of Figures

Figure 1. Gantt Chart of the Project's Timeline and Tasks	28
Figure 2. Usability Testing: Flow of Information (Moran, 2019)	33
Figure 3. Example of a First-Click Test Result (UsabilityHub, 2022)	35
Figure 4. Example of an Eye Tracking Test Result (Babich, 2019)	35
Figure 5. Example of an A/B Test (Crady, 2018)	36
Figure 6. Example of Card Sorting (Santana, 2017)	38
Figure 7. Example of a Persona (Marie, 2021)	40
Figure 8. Saturation Point (Rosala, 2021)	41
Figure 9. Example of Screening Questions (Pollfish, n.d.)	43
Figure 10. Sudzina measure of Tech Savviness (Lewis, 2022)	43
Figure 11. Menon measure of Tech Savviness (Lewis, 2022)	43
Figure 12. Pew Digital Savviness classifier (Lewis, 2022)	44
Figure 13. Innovation Process (Koen, et al., 2002)	60
Figure 14. NCD Model (Koen, et al., 2002)	60
Figure 15. Hierarchical Decision Tree	65
Figure 16. Saaty's Fundamental Scale (Nilsson, et al., 2016)	65
Figure 17. Value Proposition Canvas	71
Figure 18. Business Model Canvas	72
Figure 19. House of Quality	74
Figure 20. Navigation Map – User Type: UX Researcher (Left) and Tester (Right)	76
Figure 21. Application's Logotype	77
Figure 22. Application's Color Palette	78
Figure 23. Example of some of the Project's Issues on GitHub	82
Figure 24. Project's Main Folders	83
Figure 25. Project "src" Folder Contents	83
Figure 26. Project "components" Folder	84
Figure 27. Project "pages" Folder	85
Figure 28. Project "tests" Folder	86
Figure 29. Entity-Relationship Diagram	87
Figure 30. Figma's Final UI for 'Find Testers' Page	95
Figure 31. Final Design for 'Find Testers' Page	95
Figure 32. Select the Platform Input	96
Figure 33. SUS' Average Score	103
Figure 34. SUS Score Interpretation (Sauro, 2018)	104

List of Tables

Table 1. Usability Test Examples	34
Table 2. Usability Testing Applications	47
Table 3. Advantages and Disadvantages of Each Approach	58
Table 4. Second-level Criteria Comparison Matrix	66
Table 5. Second-level Criteria Normalized Matrix	66
Table 6. Priority Vector	66
Table 7. IR values for Square Matrices of Order n (Saaty, 1980)	67
Table 8. Parity Comparison Matrix for Time	67
Table 9. Parity Comparison Matrix for Adequation	68
Table 10. Parity Comparison Matrix for Complexity	68
Table 11. Priority Vector for Time	68
Table 12. Priority Vector for Adequation	68
Table 13. Priority Vector for Complexity	68
Table 14. Benefits and Sacrifices for the Customer	71
Table 15. Queries Explanation Table	92
Table 16. Testers Information	97
Table 17. QEF'S Not Fully Implemented Requirements Explanation	105
Table 18. QEF Results Aggregated	105

List of Equations

Equation 1. Multiplication of Normalized Matrix A by Priority Vector x	66
Equation 2. Calculation of Eigenvalue λ_{\max}	66
Equation 3. Consistency Index Calculation	67
Equation 4. Consistency Ratio Formula	67
Equation 5. Consistency Ration Calculation.....	67
Equation 6. Weight of “Time” Factor	69
Equation 7. Weight of “Adequation” Factor	69
Equation 8. Weight of “Complexity” Factor.....	69
Equation 9. Composite Priority Calculation	69
Equation 10. W_{ij} (Factor Weight j in Dim i) $\in [0,1]$ Calculation (For each Factor)	101
Equation 11. Q_j Calculation (For each Factor)	101
Equation 12. Q_i Calculation (For each Q_i Dimension).....	101
Equation 13. D Calculation (For each Q_i Dimension)	101
Equation 14. q Calculation (For each Q_i Dimension - Final Quality Measure).....	102

List of Code Snippets

Code Snippet 1. Example of a Request in Graph API	53
Code Snippet 2. Example of a Response in Graph API	54
Code Snippet 3. User Type Recognition	87
Code Snippet 4. Database Connection	88
Code Snippet 5. Login Request Handler	89
Code Snippet 6. Sign In Process	90
Code Snippet 7. Test Creator Helper Interfaces	93
Code Snippet 8. Example of a fictional Facebook User	97
Code Snippet 9. Example of a fictional LinkedIn User	98
Code Snippet 10. Example Request using the Mock API	98
Code Snippet 11. Example Response using the Mock API	98

Acronyms

AHP	Analytic Hierarchy Process
API	Application Programming Interface
BaaS	Backend as a Service
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DOM	Document Object Model
FFE	Fuzzy Front End
GDPR	General Data Protection Regulation
HOQ	House of Quality
HR	Human Resources
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IA	Information Architecture
JS	JavaScript
JSON	JavaScript Object Notation
JSX	JavaScript XML
MVVM	Model View-View Model
NCD	New Concept Development
NPD	New Product Development
PDS	Pew Digital Savviness
QEF	Quantitative Evaluation Framework
SPA	Single-Page Application
SQL	Structured Query Language
SUS	System Usability Scale

SWOT	Strengths, Weaknesses, Opportunities and Threats
TSG	Technology Stage Gate
TS	TypeScript
TSX	TypeScript XML
UEQ	User Experience Questionnaire
UI	User Interface
UX	User eXperience
VA	Value Analysis
XML	eXtensible Markup Language

1 Introduction

This dissertation was written within the scope of the curricular unit Master Thesis (TMDEI) of the second year of the Master's Degree in Informatics Engineering in the Graphic Systems and Multimedia branch held at the Instituto Superior de Engenharia do Porto (ISEP).

This chapter presents a brief contextualization of user experience (UX) design and usability testing, as well as the goal of the project - to develop a web application for managing usability tests and selecting participants with ease. Then, it is introduced a description of the problem, the objectives that must be met for the completion of the project, and, finally, the work plan followed and the structure of this document.

1.1 Initial Context

With the exponential growth of web and mobile applications, the practice of usability tests began to be more prevalent in different organizations and revealed a positive impact on the empathy of their products with the intended users.

A report released by Microsoft in 2015 (McSpadden, 2015) detailed how people's average attention spans are approximately eight seconds, so investing in user experience design can be crucial to win over customers, boost conversions and guarantee user retention. However, even the best user experience designers cannot foresee every potential user action or response, despite being trained to estimate what will be most successful based on their knowledge and expertise. This is where usability testing comes in: it can verify these hypotheses and lessen the risks brought on by poor user experiences (Wonder, 2020).

Usability testing is a user experience research method that combines qualitative and quantitative research techniques. Qualitative research techniques, such as observation and interviews, focus on understanding the subjective experience of users, including their opinions, attitudes, and behaviors. On the other hand, quantitative research techniques, such as surveys and questionnaires, focus on collecting objective data, such as task completion times and error rates. Therefore, the combination of qualitative and quantitative research techniques in usability testing can be both subjective and objective.

This practice aims to center the design on the human being in order to understand not only the users of the product but also their "objectives, tasks, resources and environments" (UXQB e. V., 2020). It is essential to learn how the users need the product to meet their own needs and how they can effectively and efficiently use it in their daily lives. Skipping this step can cause major usability issues on the final product, as it was not tested with its users before it was launched. Moreover, it "can result in money and time being wasted" (Francis, n.d.).

1.2 Problem

Despite companies realizing the importance of user testing and starting to make use of it, this task requires participants who are willing to participate in the tests, and this aspect is often a major obstacle for UX researchers, not only in terms of their recruitment but also the suitability of the participants' attributes to the product to be created. While the creation and management of usability tests, its scheduling logistics and their analysis can be time-consuming tasks and, therefore, a problem as well, the main issue to solve here is the participant selection procedure.

When it comes to the suitability of participants' attributes, we often use the term "real users": "people that match your ideal or current user profile, that is, someone that can become your client." (TeaCup Lab, 2019). It is extremely unlikely that the characteristics of the users of your product will match the overall population. Let's say we are building an application for drivers: there must be a need to conduct research with car owners or individuals who possess a drivers' license, as people who don't have the ability to drive won't provide any useful feedback for this product.

Recruiting participants for usability testing is time-consuming due to the need to find the right participants, manage their schedules, and screen them for relevance to the target audience, which can be challenging for niche or specialized products. In an ideal scenario, a usability test should involve participants who represent the target audience of the product, from whom UX researchers need to collect data, "whether it's a specific age group, occupation, or customer." (UserTesting, 2022). Therefore, regardless of being a small-scale test or one that requires a large sample of results, "finding available users [to participate in them] can be a burden." (Sauro, 2013).

1.3 Objectives

The aim of this project is to develop a web application that unifies the steps present in carrying out usability tests, from their creation to their execution, with a special focus on the selection of participants for them, which is the main problem to be solved.

The application will have its main functionalities for the creation and sharing of tests, as well as the selection of participants for them, and may, in the future, be developed in its entirety, that is, with the presence of more useful and extra functionalities for carrying out these tests.

For this, the following steps must be followed:

- Contextualization in the area of usability testing and tools currently on the market;
- Research on the process of selecting participants on these applications and which approaches are most suitable for doing so, as well as which technologies should be used;
- Planning, design and implementation of the application;

- Experimentation and evaluation of the achieved solution.

1.4 Motivation and Hypothesis

The motivation to follow this idea came from attending an online usability testing course, where people reported their difficulties when recruiting testers for them. As a consequence, my curiosity about the subject of UX led me to some useful research questions for the project:

- "What are the methods currently used for recruiting testers in usability testing management applications?";
- "What attributes should be taken into account when selecting testers?";
- "How to validate the existence of these attributes?";
- "How many testers should one recruit?";
- "What are the ways to captivate testers to participate in usability tests?".

Therefore, after carrying out a brief research on current usability testing management applications, it was discovered a way to distinguish this project from the ones already developed - our opportunity. Usability testing management tools with their own tester databases are currently available on the market, as we will review later on the state of art chapter, but it is possible that **the testers in these databases do not sufficiently represent different market segments/niches**. This is where we find our opportunity: **we are looking for testers by using other platforms' databases of users**. We believe we could discover more volunteers and gain a better representation of the target market for the product being evaluated by using datasets from several platforms and clearly stating the criteria we are seeking.

Given this, the **hypothesis of research** in this project is to confirm that:

Using other platforms' APIs (Application Programming Interface) to recruit testers for usability tests is an efficient way of finding testers from a wide variety of market niches.

While also increasing the diversity of the testing pool, we are ensuring that the feedback received is more representative of the target audience. This should lead to more accurate and useful usability testing results, as well as less time spent looking for testers. As this idea can easily be included in a usability testing management application, a deeper study on the UX area and, more specifically, usability testing, will be useful to understand the requirements that these types of applications demand. For this, several online articles that contribute to the investigation of these themes will be analyzed.

Moreover, it is important to identify possible APIs that can be used to recruit testers given a certain request. Due to the fact that the majority of people currently have an account on any social network, we came up with the idea of using APIs from well-known social networks. Accordingly, it should be easier to find people for usability tests (which are typically from a specific niche market), given the information that users make available there (i.e. age, location,

career, hobbies, etc.). With this in mind, the search for APIs should focus on popular social media APIs where we can easily have access to user’s information.

1.5 Work Plan

This dissertation has two major milestones: deliveries P1 and P2. The delivery of P1 focuses on the study and planning of the project, carrying out a state-of-the-art survey and a value analysis, as well as planning the initial design of the solution and the experimentation and evaluation methodology to be followed at a later stage. On the other hand, the delivery of P2 focuses on the implementation of the final design of the solution and its development, that is, the programming of the solution and its respective tests, following the previously established experimentation and evaluation methodology. At the end, the conclusions of the project are drawn and aspects to be taken into account as future work are reflected.

To ensure all the tasks from the beginning until the end of the project were completed within the specified timeframe for both the P1 and P2 deliveries, a Gantt Chart was created and is demonstrated in the image below:

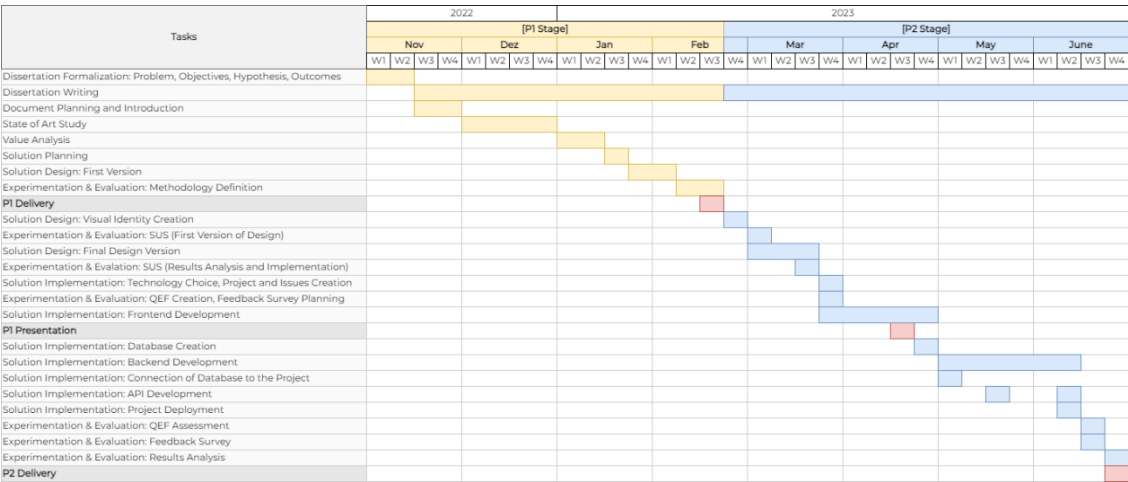


Figure 1. Gantt Chart of the Project’s Timeline and Tasks

This chart consists in a visual representation of the work plan followed throughout the timeline stipulated for this project, providing a better understanding of how time was managed to ensure that all of the priorities were attended, while also helping the owner of the project track her progress and adjust the plan if necessary.

1.6 Document Structure

This document contains eight chapters:

1. The first chapter contains the introduction, consisting of an initial context about the theme of this dissertation, followed by a description of the problem, the objectives, the hypothesis of research, the work plan, and a presentation of the document's structure;
2. The second chapter is dedicated to the review of the state of the art, which was divided into three parts:
 - a. Contextualization of UX Design, Usability Testing and Participant Selection;
 - b. Analysis of Usability Testing applications currently on the market;
 - c. Study of the appropriate technology for the development of this project.
3. The third chapter is dedicated to planning the solution, where several possible approaches to solving the problem will be discussed, and their advantages and disadvantages will be reviewed, in order to choose the one that best suits the estimated time and effort of implementation for this project;
4. In the fourth chapter, a value analysis is carried out, describing the value that this solution can bring to an organization and to its customers;
5. The fifth chapter presents the design of the solution, acknowledging the chosen solution with the respective user types definition, a navigation map for each user type, the design requirements for each of them, a first draft of the design of the user interface in the form of an interactive prototype, and, finally, the final design, including the application's visual identity creation;
6. The sixth chapter will describe the implementation of the solution, namely the technology used, implementation process, code architecture and documentation, and how to access the final project;
7. In the seventh chapter, the proposed solution is tested and evaluated:
 - a. Firstly, the test methodology is described;
 - b. Then, the application will have its three tests performed and the results obtained are exposed and analysed.
8. The eighth chapter is dedicated to exposing the project's conclusions, focusing on an overview of the project, challenges encountered, achieved objectives, and suggestions for future work.

All the references used and attachments provided throughout the writing of this dissertation will be presented by the end of the document.

2 State of Art

A state-of-the-art analysis will be performed in order to interpret and synthesize the existing knowledge related to the problem:

- Firstly, a background will be provided for the user experience area, usability tests, and participant selection procedure;
- Next, as an attempt to comprehend the competition and identify how this application can stand out from the others on the market, applications identical to the one being developed will be examined;
- At last, a study will be carried out on the most appropriate technology to be used in the implementation phase, focusing on the external API that will assist in the selection of participants.

2.1 Context

Understanding the problem that needs to be solved requires knowledge of the context. The first step is to acknowledge the UX design area, namely what a UX designer and a UX researcher do, and then their duties, such as usability testing - the focus of this dissertation.

When mentioning the subject of usability testing, it is important to know what elements are part of a usability test, what types of tests exist, when to conduct this task, and, most importantly, why it is crucial for a better user experience on the final product.

Finally, usability tests require participants to perform them. It is essential to know how many participants one should look for depending on the test we want to do, which attributes should be taken into account when selecting these people, how the existence of those attributes can be validated, and how one can recruit them. After reading this part, the reader should understand why it is important to recruit participants who have traits that are appropriate for the final product being developed and why doing otherwise might be problematic.

2.1.1 User Experience Design

2.1.1.1 What is User Experience Design?

User experience (UX) design focuses on the creation of products that offer customers a meaningful and relevant experience, focusing on their needs. In addition to being usable, a product should also elicit positive emotions in the user, such as enjoyment, fulfilment, or pleasure - all of which describe a good user experience.

This area is not related to the software's interface only, as UX designers can integrate other aspects of the product like its functionality, branding, marketing campaigns, packaging, etc. This happens because the users' experience does not start from the moment that they interact with the software for the first time but, instead, from the moment they acquire the product.

User experience is defined by the International Organization for Standardization (ISO) as follows: "A person's perceptions and responses that result from the use or anticipated use of a product, system or service." (ISO 9241-210, 2019). In other words, UX designers conceptualize, plan and alter elements of a "product, system or service" that ultimately influence "A person's perceptions and responses" to it.

2.1.1.2 UX Designer vs UX Researcher

When it comes to the user experience area, we are often presented with two career paths: UX Designer and UX Researcher.

Software or Product Usability, Interaction Design, Information Architecture and Wireframing are the main concerns of a UX Designer. This field requires technical abilities such as the domain of prototyping software like Adobe XD, Figma or Sketch. It is also important to note that a UX Designer is advised to speak with a UX Researcher before beginning any of his/her tasks because they all call for extensive user research.

On the other hand, a UX Researcher does exactly as its title suggests: research. At the initial stage of the product's idea, he/she investigates the customer's psychology to learn about the drivers behind their behaviors when it comes to using applications, as well as their motivations. To do this, they use a variety of UX research techniques like market analysis, consumer surveys, product evaluations, studies and, the focus of this dissertation, usability testing. "The most important skill for a UX researcher is the ability to conduct efficient research in qualitative as well as quantitative methods. Individuals with strong data analysis and problem-solving skills will be prioritized for this job role." (Mathew, 2022).

Most of the time, a UX Designer is also a UX Researcher and their tasks overlap, however, these are two separate functions and their separation can provide a better user experience of the product itself, as it acknowledges the time and level of detail that both of these roles require in order to achieve a better performance of their tasks.

2.1.2 Usability Testing

2.1.2.1 What is Usability Testing?

Usability testing is a UX research method that consists of evaluating a product's experience with its target audience. The objective is to watch people perform a series of actions and discover any issues they might run into. It's an approach that helps to reveal why users accomplish tasks in the way they do, including understanding their motivations and needs since each user performs tasks differently (Evans, et al., n.d.).

2.1.2.2 Elements of Usability Testing

Usability Tests assume three main elements: the facilitator, the tasks and the participant. The facilitator (can be the UX Researcher or not) guides the participants throughout the process and administrates him/her the tasks, analyzing his/her behavior. The tasks depend ultimately on what we are testing, so we will clarify them in the next section. Finally, the participant (usually more than one person) is a real user of the product and will perform the tasks that are given to him/her. This person is often asked to think aloud during the tests to provide richer feedback for the facilitator (Moran, 2019).



Figure 2. Usability Testing: Flow of Information (Moran, 2019)

Usability tests can be performed in-person or remotely, and they can be moderated or non-moderated: when moderated, the facilitator can interact with the participant and provide him with help or ask follow-up questions; when unmoderated, the participant must figure things out by himself/herself and the results are only analyzed after the session is completed.

2.1.2.3 Types of Usability Tests

In a usability testing study, the following two types of data may be gathered:

1. Qualitative data: consists of observations that categorize how intuitively usable a design component is;
2. Quantitative data: in the form of one or more metrics that indicate the difficulty of a task, such as task completion rates or task timeframes.

When collecting qualitative data, a UX researcher will pay attention to the participants' interactions with UI elements, understanding "which aspects of the design are problematic and which work well." (Bodiu, 2017). This way, the design decisions are supported by the results obtained, and it is also possible to find usability issues in advance. The questions in these tests usually begin with "Why?".

On the other hand, when looking for quantitative data, the researcher analyzes the "users' performance on a given task (e.g., task-completion times, success rates, number of errors)"

(Bodiu, 2017), as well as their “perception of usability (e.g., satisfaction ratings)” (Bodiu, 2017). This time, the questions frequently begin with “How?”.

Usability tests can be applied in many areas, but as we are focusing on the UX area, the most common types of tests used are those based on: the content, information architecture, or surveys (Rog, 2019). Some examples of tests used in these categories would be:

Category	Test Examples
Content	<ul style="list-style-type: none">· 5-Second Testing· First-Click Testing· Eye Tracking· A/B Testing· Focus Groups· Cognitive Walkthrough
Surveys	<ul style="list-style-type: none">· Feedback Survey· System Usability Scale (SUS)· User Experience Questionnaire (UEQ)· Heuristic Evaluation
Information Architecture	<ul style="list-style-type: none">· Card Sorting· Tree Testing

Table 1. Usability Test Examples

There are several other types of tests apart from those already stated, so this section merely aims to highlight the differences between the most popular ones, along with their intended purpose, which is covered below.

5-Second Testing

A method of user research that allows the assessment of the first impressions that users get from a web page design, within the first five seconds of looking at it. These tests are frequently used to determine if the design is smart, which means that it is conveying the correct message to the user. Additionally, it is important to note that “there is a growing trend for website visitors to open many sites at once, reducing the attention each receives and increasing the importance of effective design and messaging.” (UsabilityHub, 2022). An example of questions that could be asked in this test would be (Maze, 2022):

- “What was your impression of the design?”;
- “What is the main thing you can recall?”;
- “What did you like best/least about the design?”;
- “What would you change about the design?”.

First-Click Testing

This test aims to measure the usability of a product by determining how simple it is to carry out a certain task. “Web analytics packages can tell you where users clicked, but not what they were trying to achieve. Click testing allows you to ask users to carry out a specific task, letting

you isolate and investigate user behaviour around each different scenario separately.” (UsabilityHub, 2022). A first-click test result can be seen in the form of a heat map:

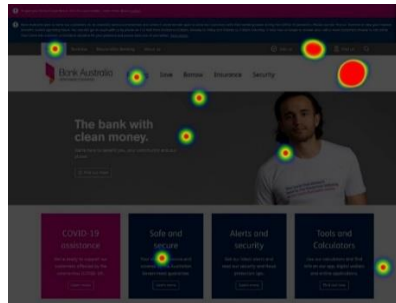


Figure 3. Example of a First-Click Test Result (UsabilityHub, 2022)

Eye Tracking

With the help of eye tracking technology, it is possible to determine where a person is looking, what they are looking at, and how long they have their gaze fixed on a certain area. Eye tracking devices are tremendously helpful for studying a user's visual activity because they do not rely on participant reports or participant recollection, and all data is automatically recorded. For instance, it is important to use eye tracking to determine whether the design of your form fields is suited for quick scanning, which allows users to scan the form and identify all needed fields at a glimpse (Babich, 2019). An example of an eye-tracking report can be seen in the following image:

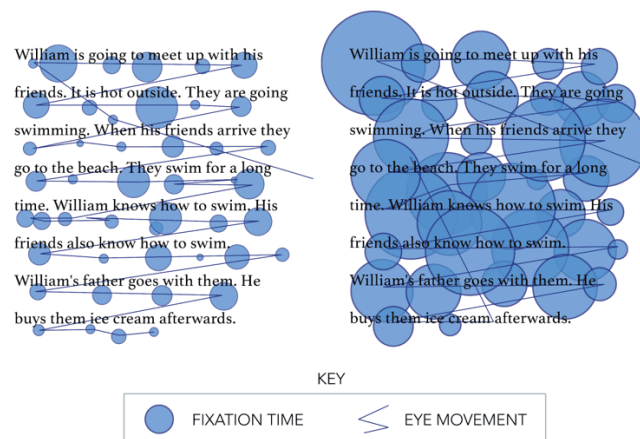


Figure 4. Example of an Eye Tracking Test Result (Babich, 2019)

A/B Testing

A/B testing, also known as split testing, is a straightforward study in which users are randomly presented with two or more variations of a design to see which one performs better. With this technique, any user interface (UI) element can be analyzed, from text or wording to different element's sizes, colors, shapes, and positioning. In addition, it can compare using photos against illustration, as well as multiple form lengths, labels, and placements (Studio by UXPin, n.d.).

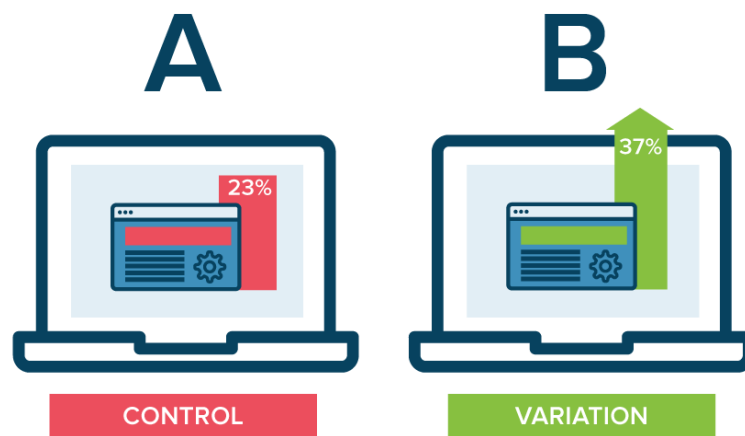


Figure 5. Example of an A/B Test (Crady, 2018)

Focus Groups

This is a form of collaborative testing that brings together a small group of participants (usually from 6 to 9 participants) to “discuss issues and concerns about the features of a user interface” (Nielsen, 1997), whether it has been implemented or not. This test frequently elicits spontaneous responses and ideas from participants while also allowing the researcher to notice some group dynamics and organizational problems. “In interactive systems development, the proper role of focus groups is not to assess interaction styles or design usability, but to discover what users want from the system.” (Nielsen, 1997).

Cognitive Walkthrough

It is a task approach used to evaluate the usability of a product, intended to determine whether a new user can do tasks within a particular system with ease. Comparing a cognitive walkthrough to many other types of usability testing, the major advantage is that it is incredibly quick and affordable to conduct. It can also be used in the design phase prior to development, providing quick insight before money is wasted creating a useless product (Interactive Design Foundation, n.d.). According to the “Cognitive walkthrough for the Web” paper (Blackmon, et al., 2002), four questions must be asked during a cognitive walkthrough:

1. “Will the user try and achieve the right outcome?”;
2. “Will the user notice that the correct action is available to them?”;
3. “Will the user associate the correct action with the outcome they expect to achieve?”;
4. “If the correct action is performed, will the user see that progress is being made towards their intended outcome?”.

An example of a task to follow could be, for example, a login on a website. The process would be described as followed (Interactive Design Foundation, n.d.):

1. “Open browser”;
2. “Navigate to site”;
3. “Enter the user name in the user name field”;

4. "Enter the password in the password field";
5. "Click the login button".

Feedback Survey

A feedback survey, in UX, gathers user feedback to help understand user behavior. They provide the researcher with an understanding of how customers are using the product in their own words, enabling him/her to determine what improvements to make to enhance the user experience.

Such surveys are typically carried out immediately following the release of software or the debut of a website. Therefore, you can start these surveys as soon as a customer starts using your product. You may get both quantitative and qualitative user data and learn what your target market thinks of your product by conducting a user experience survey. Their preferences, expectations, and pain areas may become clearer as a result (Cornell, 2022).

System Usability Scale (SUS)

For testing usability, the System Usability Scale (SUS), originally designed by John Brooke in 1986, consists of a 10-item questionnaire with five possible responses - from Strongly agree to Strongly disagree:

1. I think that I would like to use this system frequently;
2. I found the system unnecessarily complex;
3. I thought the system was easy to use;
4. I think that I would need the support of a technical person to be able to use this system;
5. I found the various functions in this system were well integrated;
6. I thought there was too much inconsistency in this system;
7. I would imagine that most people would learn to use this system very quickly;
8. I found the system very cumbersome to use;
9. I felt very confident using the system;
10. I needed to learn a lot of things before I could get going with this system.

SUS has been cited in more than 1300 articles and publications, making it an industry standard. It is a fairly simple scale to use and it is reliable because it can distinguish between systems that are usable and those that are not (usability.gov, n.d.).

User Experience Questionnaire (UEQ)

UEQ is a quick and accurate survey to evaluate interactive goods' user experiences. The questionnaire's rating scales cover every aspect of the user experience. Both traditional user experience elements (originality, stimulation) and usability elements (efficiency, perspicuity, dependability) are measured.

In more detail (Hinderks, et al., 2018):

- Attractiveness: "Overall impression of the product. Do users like or dislike it?";
- Perspicuity: "Is it easy to get familiar with the product and to learn how to use it?";

- Efficiency: “Can users solve their tasks without unnecessary effort? Does it react fast?”;
- Dependability: “Does the user feel in control of the interaction? Is it secure and predictable?”;
- Stimulation: “Is it exciting and motivating to use the product? Is it fun to use?”;
- Novelty: “Is the design of the product creative? Does it catch the interest of users?”.

Heuristic Evaluation

A heuristic evaluation can be used to determine how user-friendly a website or application is. In this test, as opposed to user testing (where the product is analyzed by users), the product is examined by usability specialists. Jakob Nielsen from Nielsen Norman Group created 10 general principles for interaction design, known as heuristics (Nielsen, 2020). These are:

1. Visibility of system status;
2. Match between system and the real world;
3. User control and freedom;
4. Consistency and standards;
5. Error prevention;
6. Recognition rather than recall;
7. Flexibility and efficiency of use;
8. Aesthetic and minimalist design;
9. Help users recognize, diagnose, and recover from errors;
10. Help and documentation.

Card Sorting

This is a UX research technique where people surveyed arrange individual labels on notecards/post-it into groups based on standards that make sense to them. This approach helps to develop an information architecture (IA) that corresponds to users' expectations by revealing the organizational structure of the target audience's domain knowledge. Card sorting is essential to make the UI more predictable, or, in other words, to make the information more accessible and help design better products for the users (Babich, 2019).

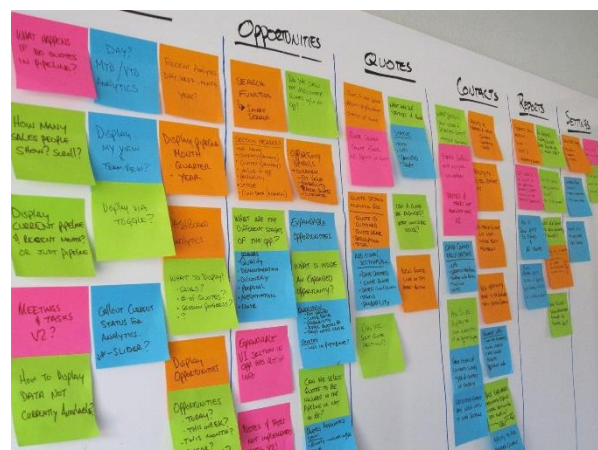


Figure 6. Example of Card Sorting (Santana, 2017)

Tree Testing

Finally, tree testing enables the assessment of the organization and discoverability of topics on a website or mobile application. Participants in a tree test are shown a text-only hierarchy of the website and asked to perform a series of tasks. Ultimately, the question of "Can consumers locate what they are seeking for?" must be addressed. Tree testing examines a tree, which is a hierarchical category structure, and is sometimes referred to as "reverse card sorting." (Maze, 2022). Examples of Tree Testing would be (Maze, 2022):

- "A TV streaming service may task participants with locating where they can upgrade their account plan to add more devices.";
- "An online grocery store could ask users to find how to change their delivery time slot or add a new payment method.";
- "A bus service might set participants the goal of finding where they can see the next bus stopping at their local station.";
- "A tech company may request users navigate to the settings page of their laptop and find where to change their screensaver.".

2.1.2.4 When to conduct Usability Testing?

As soon as you start working on the strategy for a brand-new website or app, you should, if possible, do usability testing on the current version of the product. It is crucial to use the results of your initial usability test throughout the design process of your project and to continue testing consumers afterwards (Bathia, 2016).

However, knowing when to conduct usability testing can be a tricky task. According to the Director of Content of DePalma Studios (Watson, 2018), despite the fact that most designers are aware of the need for usability testing, he found that there's less clarity on when to conduct a test. Given this, he advises to conduct usability testing at four stages:

1. "Before Any Design Decisions Are Made";
2. "When It's Time to Evaluate and Iterate";
3. "After Launch";
4. "In High-Risk, Low-Certainty Situations".

Watson believes that the correct answer to "When should you run a usability test?" should be "early and often".

2.1.2.5 Why is Usability Testing important?

This question is vital to be answered because we cannot fix something if we do not realize it is broken in the first place. Based on these findings, companies and organizations that promote a testing culture can find out what is faulty or not functioning at its best and provide fixes. In addition to that (Bathia, 2016) (Rog, 2019):

- When designing in a vacuum, it is hard to comprehend the desires, requirements, pains, and pleasures of your users - Usability testing can help building empathy towards them, which can help you become a better designer for this particular audience or project;
- Designers often make assumptions about what their consumers will find "useful, functional, learnable, and pleasant." and they do not always correspond to their realities - Usability testing can help on getting unbiased feedback coming from a more objective source, that is the actual target audience for the product;
- The cost of design and development is high - Testing for usability ensures that the effort invested in design and development is not lost and helps reduce business risk by not allowing bad solutions to get to the market.

In conclusion, the earlier we test, the easier it is to iterate and the less it costs to redo some of the concepts or screens of the application.

2.1.3 Participant Selection in Usability Testing

In usability tests, it is a common practice to establish direction and outcomes for a particular testing study. In other words, it is important to outline the purpose of the test and, with that, define the audience: indicate the number of participants and the criteria for their selection (Rog, 2019).

2.1.3.1 Definition of a Persona

The first task before participant selection is the definition of a persona: "A persona is a fictional, yet realistic, description of a typical or target user of the product." (Harley, 2015). This task should be carried out as early as possible, as it should be a part of the research stage even before the design process starts.

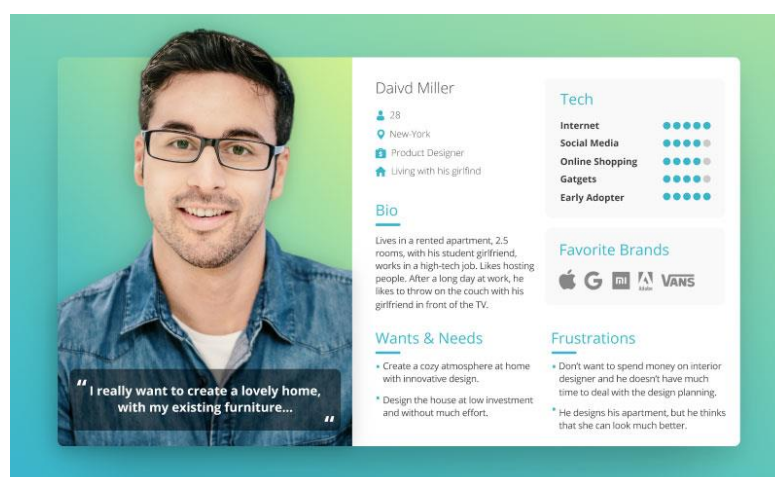


Figure 7. Example of a Persona (Marie, 2021)

It is common to include the following pieces of information on the persona-creation (Harley, 2015):

- “Name, age, gender, and a photo”;
- “Tag line describing what they do in ‘real life’”;
- “Experience level in the area of your product or service”;
- “Context for how they would interact with your product: Through choice or required by their job? How often would they use it? Do they typically use a desktop computer to access it, or their phone or other device?”;
- “Goals and concerns when they perform relevant tasks: speed, accuracy, thoroughness, or any other needs that may factor into their usage”;
- “Quotes to sum up the persona’s attitude”.

Personas make it simpler to find subjects for usability studies. We can think of personas as templates for potential recruits: by making characteristics of key user segments more salient, traits that are shared by numerous personas or the unique qualities of a persona may be helpful screening criteria for a study’s participants.

2.1.3.2 How many participants are necessary?

A rule of thumb popular in UX is using five participants per persona. You can identify almost as many usability issues by testing with five individuals as you would by having many more test subjects. There are certain exceptions, though (Nielsen, 2012):

- Quantitative research (focusing on statistics rather than insights): To obtain statistically significant results in quantitative research, test at least 20 people;
- Card sorting: Test each user group with a minimum of 15 users;
- Eye tracking: To reach reliable heat maps, test 39 users.

In addition to these exceptions, for many exploratory research studies, 5 participants can either be a very small sample or result in more interviews than necessary, especially when the number of personas is high (five or more). Therefore, a good methodology to use when recruiting participants is the “saturation method”: “Saturation in a qualitative study is a point where themes emerging from the research are fleshed out enough such that conducting more interviews won’t provide new insights that would alter those themes.” (Rosala, 2021).

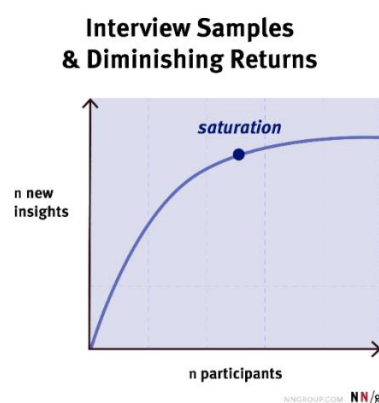


Figure 8. Saturation Point (Rosala, 2021)

In other words, there are diminishing returns after a certain number of interviews, meaning that little new information is discovered regarding the subject of the study. At this moment, saturation has been attained. Taking this into consideration, it is a positive practice to start with a few number of participants (i.e. a sample of 5-6 people) and then analyze as we go. “If you’re still learning new things and generating new codes, you can recruit a few more participants until you reach the point where your themes are complete and little new insight is being discovered with each new interview.” (Rosala, 2021).

2.1.3.3 Recruitment Criteria

As previously mentioned, when selecting participants, it is a good procedure to recruit people who match the personas created. It is even more important to match them when we have an industry-specific product, for example, a Human Resources (HR) software that needs to be tested with people that work in HR. Unless we are Google or Amazon and have a product for a very broad audience, it is crucial that the participants match the criteria defined earlier (Rog, 2019).

When the personas are not a part of the design process, the term “real users” comes in: as explained in the Problem section, real users are “people that match your ideal or current user profile, that is, someone that can become your client.” (TeaCup Lab, 2019). Finding actual customers (or personas) for your product or service is not a requirement for doing research with real users. In fact, sometimes including your customers can damage your research. For instance, if you are creating an HR software and your customer is already familiar with that kind of software, he/she likely will not have any problems using it. Given this, it would be a better idea to test with new users, those who do not have any familiarity with the product, because the majority of the interface's usability problems will be difficult for new users to handle, which will make them easier for you to find and repair.

A list of helpful attributes to look for when selecting the audience is (Rog, 2019):

- Exposure to the product - Have they used any similar software before? What are their expectations?
- Age (range) - Can be important to define if the product should be adapted to a specific age range such as children or the elders;
- Tech savviness - This is essential to understand the user’s knowledge about modern technology. For instance, if we are creating a product for iOS (Apple’s Operating System for mobile devices - iPhone and iPad), it should matter if the user has previously been in contact with this system, otherwise it might be harder for him/her to use the product and the feedback might be negative, as he/she is not familiarized with the device;
- Interests, Behaviors and Experience (related to the project) - As an example, if we are creating an application to be used in museums, it should be a requirement that the user visits museums at least once a year; If we are creating an online store, then the user should frequently buy online products.

“The more specific the requirements you have, the more effort you must put into recruiting, so you should carefully consider the necessity of each criterion.” (Mortensen, 2020).

2.1.3.4 How to validate Recruitment Criteria?

Before performing any kind of test, it is necessary to validate the existence of these attributes.

On the one hand, when evaluating attributes such as exposure, age or personal interests, we can make use of screener questions/surveys: a questionnaire to test potential participants. These questions should be very simple to answer, usually in multiple-choice format, and their purpose is to help determine if the users meet the criteria defined.

STEP 1: TARGETING

Targeting men with children.

Gender

☒ Male ☐ Female

Number of Children

☐ None ☒ Three

☒ One ☒ Four

☒ Two ☒ Five or More

STEP 2: SCREENING

Of the men with children, screening for those who have a child enrolled in school.

SQ1: How many children do you currently have enrolled in school?

☐ Three X Not allowed

☐ Two X Not allowed

☐ One ✓ Allowed

☐ None of the Above X Not allowed

Figure 9. Example of Screening Questions (Pollfish, n.d.)

On the other hand, when evaluating tech savviness, it is important to assess (Lewis, 2022):

- “what someone knows”;
- “what someone does (or reports doing)”;
- “what someone feels (attitudes, especially self-assessments)”.

There are three commonly used methods to measure tech savviness: the Sudzina Measure, the Menon Measure and the Pew Digital Savviness (PDS) classifier.

★ Please indicate to what degree you agree with the following statements.

	Strongly Disagree 1	2	3	4	Strongly Agree 5
People consider me to be tech savvy.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I consider myself to be tech savvy.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 10. Sudzina measure of Tech Savviness (Lewis, 2022)

★ Please indicate to what degree you agree with the following statements.

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
I prefer to read information on the internet rather than in a brochure.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am interested in new technology.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is easy to locate the information I want on the internet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 11. Menon measure of Tech Savviness (Lewis, 2022)

★ Overall, how often do you use the internet?

Never 0	Less than once a week 1	Once a week 2	Several times a week 3	At least once a day 4	Multiple times a day 5	Most of the day 6
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

★ Overall, how confident do you feel using computers, smartphones, or other electronic devices to do the things you need to do online?

Not at all confident 1	Only a little confident 2	Somewhat confident 3	Very confident 4
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 12. Pew Digital Savviness classifier (Lewis, 2022)

“The Sudzina questionnaire has two self-assessment items; the Menon measure has three attitudinal agreement items; and the Pew Digital Savviness classifier has two items, one about reported behavior (frequency of internet use) and one for online confidence.” (Lewis, 2022).

All of these methods have found success in the research contexts for which they were designed, but because none of them have been subjected to criterion-related validation against successfully completing tasks or levels of perceived usability, they are not particularly helpful for categorizing the tech savvy of participants in UX research. According to (Lewis, 2022), measures based on lists of activities that people feel at ease with could be just as effective as or even superior to metrics based on tests or brief attitude/self-assessment questionnaires.

2.1.3.5 How to recruit participants?

Although hiring a recruitment agency is frequently an effective way to find participants, it is also expensive. In the next section, we will cover some platforms that may offer their own database of participants; however, it is not certain that you will find the participants you are looking for just by using these applications. Furthermore, the more specialized is the audience you need, the more you have to pay when using these services.

You must conduct your own participant research if you lack the means to hire a recruitment firm. The effectiveness of using your network or posting on social media depends on how close the participants are to your social circle, but if they do not know you personally, there is less chance of bias. People commonly employ their co-workers as test subjects when they are pressed for time. Nevertheless, given the possibility of bias, it would be preferable if you were cautious when using your co-workers.

You can also make use of online forums or interest groups where you believe members of your target group may be active if you need to recruit members of hard-to-reach user groups (Mortensen, 2020).

2.1.3.6 Participant Compensation

As expected, it is easier to persuade people to engage in an activity if they stand to gain something in return. This return may come in the form of a cash payment or a discount coupon, similar to an Amazon voucher.

So, what is the estimated compensation for taking part in a usability test? Amounts range depending on the test type: “Most tests pay \$10 for successful completion. Live Tests are longer tests that will have a moderator present, and you can earn \$30 or \$60 depending on the length of time.” (Ly, 2022). And what are the advantages? “Not only do you get paid to test, but you learn something new every time and get a sneak peek of what the brands you love most are working on. Even better, you will play a vital role in making the world more user-friendly!” (Userlytics, 2022).

2.2 Usability Testing Management Applications

To understand the opportunity of the project to be developed, that is, how it can be distinguished from the applications currently on the market, we will have to analyze them and understand their key characteristics.

Usability testing management applications aim to put together proper materials to optimize the time spent on this task. Moreover, it can be more economical to opt for this kind of apps, as they can be highly effective and beneficial when the right resources are provided. Some of them even offer their own database of testers, however, if you are looking for a very specific niche, it can be hard to find all types of users there. How to choose the best application to use would depend on your project’s needs and your budget, as some of them can be quite expensive.

This led to the research of a handful of applications that are commonly used in the context of usability testing. The data for ten of these applications is summarized in the table presented below. The information includes the name of the application, its pricing, its key features, and how their participant selection method works (in case it exists). The participant selection method was divided into two categories:

1. Through their own database - this means that the application has its own pool of testers who have previously registered in it;
2. None - it is up to the UX Researcher to find its participants by his/her own means.

The applications were organized in this table according to their price (from lowest to highest):

#	Name	Price	Key Features	Participant Selection Method
1	Google Optimize (https://marketingplatform.google.com/about/optimize/)	Free plan or paid one (value is not specified).	<ul style="list-style-type: none"> · Provides a Visual Editor for the creation of tests; · Includes A/B, multivariate, and redirect tests; · Compares website's traffic and conversion rates; · Integration with Google Analytics, Google Ads and Firebase. 	None.
2	Maze (https://maze.co/)	Free plan allows one usability testing project. Plans start from 23€/month.	<ul style="list-style-type: none"> · Provides templates for tests; · Test sharing via a link; · Integration with Adobe XD, Figma, InVision, Marvel and Sketch; · Real-time results. 	None.
3	Lookback (https://www.lookback.com/)	Free 60-day trial. Plans start from 23€/month (limited sessions/year).	<ul style="list-style-type: none"> · Video conferencing tool adapted to usability testing; · Provides a place where teams can collaborate on insights, share time stamped notes on videos and chat. 	None.
4	Hotjar (https://www.hotjar.com/)	Free plan or paid one starting from 32€/month.	<ul style="list-style-type: none"> · Monitor real-time user behavior (live analytics); · Provides survey templates; · Integration with Google Analytics. 	Through their own database.
5	UserTesting (https://www.usertesting.com/)	From 46€ per video session.	<ul style="list-style-type: none"> · Video conferencing tool adapted to usability testing; · Gives access to videos of testers interacting with the product; · Provides templates for tests; · Use of Data and Machine Learning for sentiment or behavioral analysis; · Integration with Slack, Trello, JIRA, Adobe XD, Google Calendar, Outlook, and more. 	Through their own database.
6	UsabilityHub (https://usabilityhub.com/)	Free for tests up to 2 minutes long. Basic plan 83€/month and Pro plan 187€/month.	<ul style="list-style-type: none"> · Provides templates for tests; · Analyzes the results - heat maps indicating where users clicked. 	None or Through their own database depending on the plan chosen.

7	Loop 11 (https://www.loop11.com/)	14-day free trial. Plans start from 168€/month.	<ul style="list-style-type: none"> · Available on mobile; · Creation of multiple tests; · Real-time reporting; · Integration with Axure, JustInMind and InVision. 	None.
8	UserBrain (https://www.userbrain.com/en/)	Plans start from 299€/month.	<ul style="list-style-type: none"> · You can sign up as a customer or a tester (and get paid to be a tester); · Provides templates for tests; · Test sharing via a link; · Gives access to videos of testers interacting with the product. 	Through their own database.
9	TryMata (formerly TryMyUI) (https://trymata.com/)	15-day free trial. Plans start from 315€/month.	<ul style="list-style-type: none"> · You can sign up as a customer or a tester (and get paid to be a tester); · Monitor real-time user behavior (live analytics); · Provides templates for tests. 	Through their own database.
10	Userlytics (https://www.userlytics.com/)	Plans start from 469€/month.	<ul style="list-style-type: none"> · Provides templates for tests; · Analyzes the results; · Over 20 languages supported; · Gives access to videos of testers interacting with the product. 	Through their own database.

Table 2. Usability Testing Applications

By analyzing this table, we can conclude that:

1. The prices vary depending on the quantity of features presented;
2. Each of them has a different purpose: some of these applications are more focused on data analysis, others on videoconferencing for behavior analysis and others on templates for testing and analysis of results. Very few of them combine all of these items in one;
3. Most offer their own database of testers, some even paying anyone who signs up as a tester;
4. Some of these applications offer very useful integrations with other platforms, which can be a big advantage if the UX Researcher is used to using them as well.

2.3 Technology

For the development of the web application that will solve the problem, it is necessary to research the most appropriate technologies and understand which frameworks are popularly used nowadays for this type of project. Studying the technologies before starting a development project is important for several reasons, including:

1. **Choosing the most adequate technology** (TP&P Technology, 2021): It is crucial for the developers to select the technology that best satisfies the project's needs because each

of them will have their own strengths and weaknesses that will ultimately interfere with the project's requirements. Moreover, using the right technology can increase productivity and efficiency, decreasing the amount of time and effort required for development;

2. **Analyzing their cost** (Kessler, 2019): It is essential to study the cost-effectiveness of the technology and understand their future cost of maintenance. In other words, choosing a technology that is well-established and has a long-term support plan ensures that the project can be maintained and updated easily, even if the development team changes;
3. **Popularity/Keeping up with trends** (Croeser, 2022): As technologies are constantly evolving and new ones are emerging regularly, using up-to-date tools can be useful both in terms of maintenance or changes in the development team.

2.3.1 Criteria of Choice

The purpose of this study will be to assist in choosing the technology that best suits the project at the implementation stage. To choose it, we should be taking into account factors such as:

1. **Project Requirements:** The technology chosen should be able to meet the project's specific needs, which will be presented during the design stage;
2. **Development Expertise:** Personal experience and preference or knowledge of the chosen technology should be taken into account, as the more familiar we are to the technology, the smoother and quicker is the development process, avoiding possible delays or complications;
3. **Availability of Resources:** Each technology has its own unique collection of libraries, frameworks, and documentation. As a result, it is fundamental to choose a technology that has a strong community and resources available to assist with any potential technical difficulties. In order to ensure that the project can be easily updated and maintained in the future, it is also crucial to take into account the long-term support and maintenance of the technology;
4. **Cost:** Finally, the cost of the technology should be taken into account. Although most of the technologies to be presented are free or open-source, some may have costs, which can affect their choice.

In addition to considering these factors, when choosing the API, it should also be noted that the companies that make them available could impose their implications on their use. Thus, in case there are limitations in choosing one of the presented APIs, there will be the possibility of developing a mock API as a proof of concept of the idea.

We are dividing the technology research into two parts:

1. First, we will cover which technologies can be used for the application in general, that is, its frontend and backend development – for each of these, there will be at least three examples of possible technologies to use;

2. Secondly, we will study which APIs can be used for the development of the participant selection process.

2.3.2 Technology for the Solution's Development

Starting with the frontend development of the web application - the client-side of the development: the creation of the graphical interface that the user can interact with - it is common to make use of the three fundamental programming languages for the web: HTML, CSS, and JavaScript.

Accordingly, it is essential to pay attention to the most recent frontend frameworks that promote a successful development. In light of this, below are presented three of the most popular frontend innovations.

2.3.2.1 React.js

A well-known open-source JavaScript frontend library for generating dynamic, interactive apps and enhancing UI/UX design is called React.js (React.js, 2023), frequently referred to as React. The technology was initially created and maintained by Facebook and was later incorporated into some of its products, including Instagram and WhatsApp. At the moment, it is used by more than 220,000 websites. Apple, PayPal, BBC, Dropbox, Reddit, and Netflix are just a handful of the big names that use this platform for their websites (InterviewBit, 2022). Some of React's key features are (InterviewBit, 2022):

- Component-based architecture;
- Unidirectional data flow and one-way data binding;
- Virtual DOM;
- JSX (JavaScript XML);
- Declarative UI.

2.3.2.2 Angular.js

An essential component of the MEAN stack (MongoDB, Express.js, Angular, and Node.js) is the open-source JavaScript front-end framework Angular.js (Angular.js, 2023). Creating single-page web applications (SPAs) is its main use, and it is considered a very flexible framework. Angular.js is used to create a number of well-known websites and services, including Gmail, Forbes, Upwork, Deutsche Bank, etc.

Developers may quickly create UI views with the tool's built-in templates. Data binding in Angular is easy and quick, without the need for developer involvement. It lessens the stress placed on the CPU by allowing caching. The quick loading times and little navigation make for a good user experience (InterviewBit, 2022). A list of Angular's key features are (InterviewBit, 2022):

- Data binding;

- No browser-specific constraint;
- Built-in dependency injection;
- Built-in services;
- Templates.

2.3.2.3 Vue.js

Evan You developed the well-known JavaScript framework Vue.js (Vue.js, 2023), which is used to create interactive UIs and SPAs. It is one of the best JavaScript frameworks for building flexible and lightweight interactive UI elements. Vue.js is incredibly simple to implement because of the MVVM (Model View-View Model) architecture pattern and its lightweight design. Some well-known websites and programs created with Vue.js include 9GAG, GitLab, Nintendo, Behance, and Laravel.

To utilize Vue.js, one has to be familiar with HTML and CSS. Additionally, the main library of Vue.js is entirely devoted to the view layer. Large front-end development projects can easily include these frameworks in them without experiencing any complications or problems (InterviewBit, 2022). Some of Vue's key features include (InterviewBit, 2022):

- Computed properties;
- Built-in directives;
- Watchers;
- Components;
- Transitions.

Numerous aspects, including individual preferences, project deadlines, market conditions, server-side rendering, mobile development, budgets, volumes, flexibility, productivity, futuristic support, and execution, should be taken into account when choosing frontend technologies. Making a decision between development frameworks can be difficult, especially with frequent updates or the introduction of new tools (InterviewBit, 2022).

As for the backend development - the server-side that focuses on the app architecture and is the bridge between the frontend and the database while also being invisible to the users - we will cover below three of the most commonly used backend technologies.

2.3.2.4 JavaScript/Node.js

JavaScript continues to be the most widely used and popular programming language in the world, according to Statista (Vailshery, 2022), with 65.36% of all usage. When we discuss using JavaScript for the backend, we mean using the Node.js backend (Node.js, 2023), which is a JS runtime environment. It has great benefits both in terms of scalability and high performance (Osadchuk, 2022):

- “Frontend and backend developers can benefit from the **high level of scalability** that Node.js offers. This language provides both vertical and horizontal scalability options and allows the addition of extra nodes as well as additional resources.”;

- “Node.js uses **Google’s V8 engine** to swiftly compile the JavaScript code into the machine code. The V8 engine is known for allowing an easier and faster code interpretation, which saves time and money for the software development process.”.

2.3.2.5 Python

Python (Python, 2023) is the most often used backend technology, even though JavaScript is the most popular programming language overall. It receives attention due to its security, conciseness, and support of libraries/modules (Osadchuk, 2022).

2.3.2.6 Ruby

Another language for the backend of web development is Ruby (Ruby, 2023). Ruby is mostly used for prototyping and supports many programming paradigms like functional, object-oriented, and procedural. It is also very simple to learn. Its main advantages are focused on metaprogramming, the framework Ruby on Rails and its libraries, the so-called RubyGems (Osadchuk, 2022).

Inside the backend development we will also need a database to store the data of the web application. According to a developer survey made on Stack Overflow in 2021 (Stack Overflow, 2021), examples of three top-rated databases would be:

2.3.2.7 MySQL

The MySQL (MySQL, 2023) database was first made available in 1995 by Oracle Corporation. It has long held the top spot on several ranking charts. This is due to the fact that it was one of the earliest open-source databases and that it has a ton of sturdy and quite helpful features. It is used by Facebook, Uber and YouTube.

The business that owns MySQL also provides premium versions of the software with extra features and support. Given its maturity and open-source nature, MySQL is a well-liked SQL database for web applications. MySQL is not the greatest option if you require sophisticated data protection features like throttling and masking, but it is an excellent place to start. When dealing with semi-structured data like JSON, it is also not the ideal (Kathuria, 2022).

2.3.2.8 MongoDB

In contrast to traditional databases, which store data in rows and columns, MongoDB (MongoDB, 2023) is an object-oriented document-based database that saves information inside collections of documents. It is founded on the model of a NoSQL document storage. Among its allegedly more than 28,000 clients are Uber, Lyft, Barclays, Forbes, and EA Sports.

Consider archiving blog posts and material or an online store's product inventory - it might not always be simple to represent such data in rows and columns, that's why MongoDB is helpful in this situation, as you may need to work with semi-structured data such as JSON or XML that are more like documents (Kathuria, 2022).

2.3.2.9 Firebase

Firebase (Firebase, 2023), a Google platform for building mobile and web applications, is built on the backend as a service (BaaS) model and includes a number of useful services and useful APIs. Additionally, it includes a simple integration procedure with Android, iOS, and Unity installations that can be used to create programs for all popular mobile and online platforms. Along giving consumers a better app-usage experience, some of its advantages are (Vyas, 2022):

- Authentication Libraries;
- Real-time Database;
- Google Analytics;
- Cloud Storage.

Naturally, there has been an upsurge in offerings and products linked to SQL and relational databases. Switching between the two is not too difficult once you understand the fundamentals of making SQL queries (Kathuria, 2022).

2.3.3 Technology for the Participant Selection Process

Although the section 2.2 demonstrated several usability testing applications that have their own database of testers, these may be an expensive and non-feasible option for the project in question, given that we might be looking for a specific niche of market participants who are less likely to be found on these platforms. This can be justified with a practical example from Dropsourc:

“The fewer filters you need to apply when recruiting for user testing participants the easier they will be to find. Unfortunately, since our target market is more technical we had to recruit a very specialized type of participant. In other words, we wouldn’t be able to walk outside and pick 5 participants off the street. Because of this challenge we decided to look at 3rd party tools like Useresting.com or to help recruit participants but the more specialized the audience the more we had to pay. **How we overcame it: (...) we tried many things but ultimately found that using the targeting features of Google and Facebook we were able to run effective campaigns to recruit participants.**” (Berg, 2017)

Given this, the objective of the web application to be developed would be to return users given a certain search on any popularly used platform nowadays, where we should practically find people from all niches – this is the hypothesis that we will try to confirm, as stated on 1.4.

According to (Statista, 2023), the most popular social networks worldwide as of January 2023, ranked by number of monthly active users, are Facebook, YouTube, WhatsApp, Instagram, WeChat, among others. Since the search for users that we are trying to perform may include factors such as location, age, gender, personal interests/hobbies, or career, these parameters could be possibly found in Facebook or Instagram, being harder to find users using video platforms such as YouTube or messenger/chat/video call applications such as WhatsApp or WeChat. In addition, we also added LinkedIn to our research because its API seemed to be appropriate for this project as well. Therefore, we are studying next two possible APIs for

retrieving users given a certain search: the Graph API from Meta (Facebook) and the People API from LinkedIn.

2.3.3.1 Meta for Developers - Graph API

The main method for transferring data to and from the Facebook network is through the Graph API. Apps can use this HTTP-based API to upload photographs, handle ads, post new stories, and do a broad range of other functions programmatically.

“The Graph API is named after the idea of a “social graph” — a representation of the information on Facebook. It's composed of nodes, edges, and fields.” (Meta, 2023). Normally, you utilize nodes to obtain information about a particular item, edges to obtain collections of objects on a single object, and fields to obtain information regarding a single object or every object in a collection.

Inside this API, you are able to access another API called “Pages API”. “The Pages API is a set of Facebook Graph API endpoints that apps can use to create and manage a Page's settings and content.” (Meta, 2023). An example of a request in this API would be:

```
curl -i -X GET \
  "https://graph.facebook.com/pages/search?q=Facebook
  &fields=id,name,location,link
  &access_token={access-token}"
```

Code Snippet 1. Example of a Request in Graph API

This will return a list of Pages that meet the query's criteria. “Set the q parameter value to a keyword or search term (e.g. q=Facebook). Use the fields parameter to list any fields you want included with each Page returned in the response.” (Meta, 2023). The response would be something like this:

```
{
  "data": [
    {
      "id": "309968765748101",
      "name": "Facebook HQ",
      "location": {
        "city": "Menlo Park",
        "country": "United States",
        "latitude": 37.483183,
        "longitude": -122.149999,
        "state": "CA",
        "street": "1 Hacker Way",
        "zip": "94025"
      },
      "link": "https://www.facebook.com/Facebook-HQ-166793820034304/"
    },
    {
      "id": "194776097220801",
      "name": "Facebook Seattle",
      "location": {
        "city": "Seattle",
        "country": "United States",
        "latitude": 47.628293260721,
        "longitude": -122.34263420105,
```

```

        "state": "WA",
        "street": "1101 Dexter Ave N",
        "zip": "98109"
    },
    "link": "https://www.facebook.com/fbseattle/"
},
...
]
}

```

Code Snippet 2. Example of a Response in Graph API

It essentially allows you to search for Facebook pages given some parameters, but the fields provided for the search (id, name, location, and link) are insufficient if we are looking for a very particular niche of participants. However, it can still help us search for participants, despite having very little parameters for the search.

2.3.3.2 LinkedIn - People API

LinkedIn could be a very useful platform to utilize in this project given that it can return people's location, careers, interests, and more.

People API from LinkedIn allows the developer to (Microsoft, 2023):

- Retrieve Member Profile;
- Edit Member Profile;
- Retrieve Member Primary Contact;
- Retrieve Member Email Address;
- Retrieve Member's Connections;
- Search Member Profile by Handle.

However, the use of this API is restricted to those developers approved by LinkedIn and subject to applicable data restrictions in their agreements. In other words, to gain access to the official LinkedIn API, you will need to be a LinkedIn Partner.

“The LinkedIn Partner Program is open to four segments of businesses, namely:

- Talent Solutions, for HR tech companies;
- Marketing Solutions, for marketing automation companies;
- Sales Navigator Application Development, for startups looking to enhance LinkedIn Sales Navigator;

LinkedIn Learning Integration, for content companies.” (Goh, 2021).

This could be a complicated step given the time stipulated for this project, nonetheless, we will have to evaluate if we can use this API, as it seems to be very appropriate for what we are looking for.

3 Solution Planning

This chapter is dedicated to planning the final solution for the problem. Firstly, a complete solution to it will be presented, that is the ideal solution to be followed. However, as this solution would not be feasible until the expected delivery date of the project, we will cover next several potential solutions that intend to simplify the development stage and better concentrate on the issue to be solved: the participant selection for usability tests.

In order to allow the selection of the best solution to implement, the detailed advantages and disadvantages of each of these will be listed, so that the most appropriate one can be chosen later on this dissertation.

3.1 Complete Solution

After studying the state of the art and focusing on the conclusions obtained in subchapter 2.2. Usability Testing Management Applications, it is possible to notice that the most common features in usability testing applications currently on the market consist of the following:

- Creation of tests through templates (visual editor);
- Real-time reporting/Results analysis (through reports, heat maps, etc.);
- Video Conferencing and Screen Recording tools;
- Multi Language support;
- Sign up as a UX researcher or a tester;
- Team collaboration on sharing insights (through a messaging system, timestamps on videos, etc.);
- Integration with platforms (such as Google Analytics, Figma, Adobe XD, Trello, Google Calendar, Outlook, and more).

As previously mentioned, the prices of these applications increase given the quantity of features, which is something we must take into account when developing our chosen solution. However, the features on this list must be included in a complete solution of the problem, as they are useful to the UX researcher, which is our target audience. Other features that may be important to include in a complete solution are:

- Payment System (to facilitate the return offered to participants for participating in the tests);
- Tips/Suggestions for Junior UX researchers that don't know where to start (i.e. could guide them through which test would be the most appropriate and could suggest them an adequate number for participants);
- Notification System (for both the UX researcher - to let them know when the tests are completed, for example, or when a payment has been made - and for the participant -

to let them know of more tests that they could be useful in or to notify them about the payment);

- Extra customization/Settings (i.e. to switch between light or dark mode, accessibility factors: colorblind mode, etc.);
- Contact us/Feedback (Constant improvement of the application's features by answering user's reviews and feedback).

Although most usability testing applications offer their own database of testers, what distinguishes our application from the ones already on the market is the fact that we are not using our own platform as a database for testers. In other words, yes, we can provide our own testers if they wish to create an account in our application and participate in more tests; nonetheless, our main goal is to contact participants with the help of other platforms, such as LinkedIn or Facebook. Ideally, the researcher should be able to choose the platform on which he/she wishes to search for participants after a test has been designed.

Taking into account the time stipulated for the writing of this dissertation as well as the knowledge and effort that would be necessary to implement all the features described above, this solution, despite being a complete solution to the problem, cannot be chosen, as it would not be feasible until the expected delivery date of the project.

3.2 Possible Approaches

Given the previous description of a complete solution to the problem, below are presented solutions that do not implement all the previously mentioned requirements but still address the issue of selecting participants for usability testing. As these are not complete solutions, they are meant to be included in a usability testing platform that contains all of the features mentioned above, which means they are merely a proof of concept. In order to understand the various approaches that could be taken to solve the problem, the detailed advantages and disadvantages of each of these will be listed thereafter, so that the most appropriate one can be chosen.

3.2.1 Solution 01: Complete Solution (but not all features work)

This solution intends to simulate what the complete solution would look like, implementing its design completely, but only the following features work:

- Login/Register as a UX researcher or as a tester.

For the UX Researcher:

- An example of a usability test, in order to understand how these could be created and edited;
- Selection of criteria for choosing testers;
- Possibility to share the test with other testers;

- Selection of participants through a platform of choice;
- Access to tests already created and statistics for each question answered.

For the Tester:

- Answer a test;
- Access to tests already answered.

3.2.2 Solution 02: Simplified Solution - UX Researcher and Participant

This solution contains the same features as Solution 01 but does not implement the complete design, just the design of the mentioned features. This means that everything that is clickable will be functional, not containing any feature that does not work.

3.2.3 Solution 03: Simplified Solution - UX Researcher Only

This solution contains:

- Login/Register as a UX researcher;
- An example of a usability test, in order to understand how these could be created and edited;
- Selection of criteria for choosing testers;
- Possibility to share the test with other testers;
- Selection of testers through a platform of choice;
- Access to tests already created.

It aims to simplify the implementation phase by not requiring a registry for testers.

3.3 Advantages and Disadvantages

For a better understanding of the advantages and disadvantages associated with each approach, these have been organized in a table. The analysis of these three potential solutions to the problem will be helpful later on, specifically during the implementation stage when, for whatever reason, unforeseen events may occur that make it difficult to build an ideal solution, leading to the development of a more practical one, demonstrated in this chapter.

	Advantages	Disadvantages
Solution 01	<ul style="list-style-type: none"> · The user perceives what a complete solution would look like, while it can still be included in any platform – this solution provides the most value to users. · Because all of the menu pages are shown, even the ones whose features are not completely functional, the user can better comprehend where to access certain information in the application and how to navigate in it. 	<ul style="list-style-type: none"> · There might be limitations on the tester selection platforms, in which case a mock API needs to be developed.
Solution 02	<ul style="list-style-type: none"> · Effectively solves the problem in a simpler way and can be included in any platform. 	<ul style="list-style-type: none"> · There might be limitations on the tester selection platforms, in which case a mock API needs to be developed; · The user does not get a perception of what a complete solution would look like, because very few features of the design are being developed.
Solution 03	<ul style="list-style-type: none"> · Effectively solves the problem in a simpler way and can be included in any platform. 	<ul style="list-style-type: none"> · There might be limitations on the tester selection platforms, in which case a mock API needs to be developed; · Does not provide statistics for the UX Researcher after a test has been done; · As it does not require a registry for testers, they cannot interact with the application; · May not meet the user's needs as well as the other solutions, as it sacrifices some key features.

Table 3. Advantages and Disadvantages of Each Approach

As we can see, every approach suffers from the same drawback: creating a mock API in case the tester selection platform has any limitations. In summary, Solution 01 offers the most value to both user types by simulating a complete solution and providing a comprehensive navigation experience. Solution 02 solves the problem in a simpler way, but it lacks a complete design, making it difficult for users to perceive the full scope of the application. Solution 03 is the simplest of the three, but it sacrifices some key features and may not meet the user's needs as well as the other solutions.

4 Value Analysis

This chapter is dedicated to a value analysis of the product to be developed, in which we will first understand what is value analysis and what are its benefits, then it will be described the innovation process and its five stages (opportunity identification, opportunity analysis, idea generation and enrichment, idea selection, and concept definition), followed by the solution's value (where it will be presented a value proposition and the business model), and, finally, a quality function deployment.

4.1 Value Analysis Definition

Value analysis is a systematic assessment method that is applied to an existing product design to compare how well it serves the functions that customers demand at the lowest possible cost while maintaining the required performance and dependability. This activity requires (Rich, 2000):

- Planning, control and coordination, as it is a structured, formal procedure;
- Knowing the use for which the product is intended, in order to fulfill the customer's needs - implies the creation of standards to evaluate the degree to which the product fits the value that the customer or consumer derives from it;
- Matching of certain functional specifications and performance requirements in order to provide value to the client.

A complex array of factors makes the organized method of value analysis a reasonable cost-reduction strategy. These factors may be broken down into two categories: those that originate from the business itself and those that are influenced by the market for the product or service (Rich, 2000).

Business Reasons:

- "Design related issues";
- "Products with known problems";
- "Customer Demands";
- "Safety and Compliance Requirements";
- "The Improvement of Product Margins";
- "Corrective Action".

Market Reasons:

- "Pricing Practice";
- "The Advent of E-Commerce";
- "Reducing Complexity";
- "Compliance with Quality Regulations";
- "New Technology and Materials";
- "Environmentalism".

Therefore, there are several advantages of a structured and efficient value analysis process, some of which are significant sources of competitive advantage for any company. In order to begin the value analysis for the project that will be built in the scope of this dissertation we

must first comprehend the innovation process and its five stages of the NCD model (New Concept Development).

4.2 Innovation Process

As shown in the figure below, the innovation process can be separated into three sections: the fuzzy front end (FFE), the new product development process (NPD process), and commercialization.

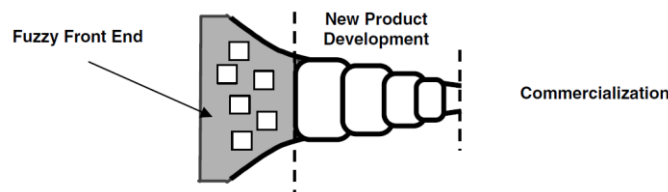


Figure 13. Innovation Process (Koen, et al., 2002)

Broadly speaking, the FFE is thought to be one of the best options for enhancing the overall innovation process. It includes the initial and foremost creative stages of the innovation process and offers business plans and innovation concepts for the creation and deployment of new products. The effectiveness of innovation depends on the quality of the work done at the fuzzy front end, where the opportunities for innovation are found, examined, and developed (Eschberger-Friedl, 2018).

Nevertheless, the FFE practice is flawed due to the absence of standard terminology and definitions for its crucial components, in addition to a lack of research on its best practices. Given this, with the goal to increase the comprehension of FFE and provide insight as well as a common terminology for it, Koen P. and his co-authors created “The PDMA ToolBook for New Product Development”, introducing the NCD model (New Concept Development):

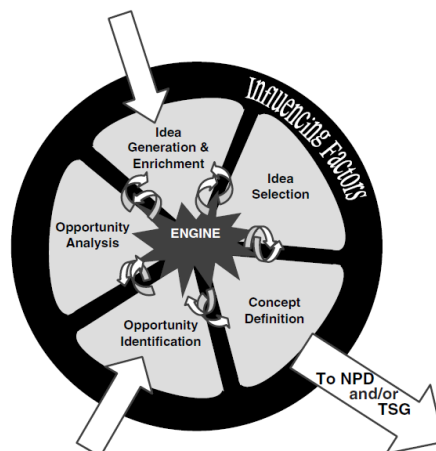


Figure 14. NCD Model (Koen, et al., 2002)

The NCD model consists of three key parts (Koen, et al., 2002):

- **The engine**, constituted by the organization's leadership, culture, and business strategy, which drive the five major variables that the company can influence;
- **The inner spoke area**, where are identified the five controllable activity elements of the FFE: opportunity identification, opportunity analysis, idea generation and enrichment, idea selection, and concept definition;
- **The influencing factors**, namely organizational capabilities, the outside world (distribution channels, law, government policy, customers, competitors, and the political and economic environment) and any relevant enabling sciences, both internal and external. These elements have an impact on innovation at every stage, from conception through commercialization and the firm has little to no control over these affecting elements.

It is also important to note that the arrows that point towards the inside of the model indicate the starting points, and the arrow that “leaves” the model represents how the concepts leave the NCD model and enter a new stage [New Product Development (NPD) or Technology State Gate (TSG)].

By applying these various NCD model components, we can locate and analyze factors that may have been previously unknown to us. To achieve this, the second point of the model - the five elements of controllable activity - will be applied to the current project, as it is the one that primarily addresses the creation of the fundamental idea or concept.

4.2.1 Opportunity Identification

According to (Koen, et al., 2002), an opportunity is “A business or technology gap, that a company or individual realizes, that exists between the current situation and an envisioned future in order to capture competitive advantage, respond to a threat, solve a problem, or ameliorate a difficulty.”. The opportunity can be a short-term countermeasure to a competitive threat, pursuing the chance to gain a competitive edge by improving the current product, making it more efficient or less expensive. It can also be a brand-new product, a new service, or a new marketing or sales strategy. The market or technological environment that the company may seek to enter is defined by the overall opportunity identification.

Finding suitable participants for user testing can be a challenge for UX researchers. Their goal is to recruit participants who match the target audience for the product being tested in order to collect relevant and useful feedback. This is why it is important to focus on recruiting "real users" with appropriate attributes (age, occupation, etc.), as covered on the introduction chapter. However, this can be difficult and time-consuming, especially for large-scale tests. In addition to recruiting participants, the creation and management of usability tests, scheduling logistics, and analysis can also be time-consuming tasks. Overall, participant selection and the management of user testing are significant obstacles that UX researchers face. To overcome these challenges, companies need to allocate sufficient resources and plan carefully to ensure a successful user testing process.

The core objective to be addressed in this project is the development of a web application that integrates all of the procedures involved in conducting usability tests, from their conception to their execution, with a special emphasis on recruiting testers for them. With the use of this application, UX researchers will be able to quickly find participants that match the target audience, by identifying their demographics, characteristics or behaviors, while also being able to plan, coordinate and organize their tests in a more effective way.

Through the research made on subchapter 2.2. where we covered ten usability testing management applications currently available on the market, we realized that some of them offer their own database of testers; however, if we are looking for a very specific niche, it can be hard to find all types of users there. Essentially, this is where we find our opportunity: what distinguishes this application from the rest is that we are looking for testers by using other platforms' databases of users. This way, if we are correctly indicating the criteria we are looking for, it can be easier to find participants from all niches. This should also lead to more accurate and useful usability testing results, as well as less time spent looking for testers, making the process more efficient.

4.2.2 Opportunity Analysis

To translate opportunity identification into specific business needs, it is necessary to conduct an opportunity analysis, in which more resources are used to provide more detail on the appropriateness and attractiveness of the selected opportunity.

In our case, we are using strategic framing by conducting a SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis on the product we intend to develop. In doing so, we can more accurately assess the benefits and drawbacks of the proposed solution. The SWOT analysis will be performed on the complete solution presented in the previous chapter (3. Solution Planning), which is the ideal solution to be developed despite not being the one followed in this project.

4.2.2.1 SWOT Analysis

Strengths:

- **Time-saving:** Combines useful features in one platform, provides templates for user tests and reduces time spent on logistics;
- **Thorough understanding of target audience** (UX Researchers) and adequation to their needs;
- **User-friendly/Easy to use:** Continuous improvement of features by performing user tests in our own application and answering our user's feedback;
- **Free plan:** Any user can test most of our application for free, even if they don't get access to all features.
- **Easy to maintain:** The development is done on recent technologies, which makes it easier to maintain.

Weaknesses:

- **Cost and time of development:** In the current project, not all features will be developed because it would require more time and effort. When developing the complete solution, it can become expensive paying a full team of developers to develop all features;
- **Not responsive:** In the first stage of the complete solution, the application is not responsive yet, which means it can only be used on a desktop. Designing a responsive solution requires more time and can easily increase the cost of the project;
- **Price for the customer:** Although a free plan is offered, it can become pricey accessing all the features in the application, as the user would have to pay for a subscription (which is our revenue stream);
- **Lack of testers:** As soon as the application is launched to the market there will be no testers registered in the application, which ultimately requires more effort for the UX Researcher to search for them, as we can't provide our own previously registered testers.

Opportunities:

- **Easy to retain more users:** By using other platforms to search for testers, we are able to reach more niches of participants;
- **Growing trend of usability testing and UX area:** As we covered on the introduction section, usability testing is becoming a more prevalent practice due to the exponential growth of web and mobile applications;
- **Rise of remote testing:** With the COVID-19 pandemic, remote activities became more common and, therefore, remote testing also increased its popularity, providing users more comfort by saving them from the logistics of having to be in a certain place at a certain time to perform usability tests.

Threats:

- **Substitute products:** It is not that complicated to reproduce this idea and develop a similar product that will become a direct competitor of this application;
- **GDPR issues:** Because we are collecting user's information, although we provide our own terms and conditions which comply to the GDPR norms, we cannot control how our users will use it;
- **Fraudulent activities:** Lastly, because the complete solution has a payment system included, we cannot control any fraudulent activities that might occur, despite working hard to develop a safe application.

Taking into account that the weaknesses presented are, for the most part, circumventable, it is remarkable that the strengths are sufficient arguments to develop this project. In addition, the opportunities confirm that the UX area is currently experiencing exponential growth, and a platform that solves the presented problem is needed. Hence, when the application is

launched, the threats mentioned can be a focus for improvement by conducting additional research and setting up a successful marketing strategy.

4.2.3 Idea Generation and Enrichment

Idea generation and enrichment is an evolutionary element that concerns the emergence, growth, and evolution of a certain idea. It can be a formal process, as brainstorming sessions and idea banks may be used in order to encourage a company to come up with fresh or revised ideas for the identified opportunity, but it can also be an informal one: an experiment gone wrong, a new material being offered by a supplier, or an unexpected request from a consumer are just a few examples of how a new idea might appear outside of any formal process (Koen, et al., 2002).

After deciding that the opportunity was worthwhile to pursue, it is now required to come up with potential ideas and/or solutions that could provide a fix for the problem. The complete solution was the first solution covered in the previous chapter; however, we have already mentioned that this solution is not feasible to develop given the project's expected deadline. Thereby, three other solutions were presented as possible alternatives to solve the problem while simplifying the implementation of the features presented in the complete solution. To recap, the solutions were:

- Solution 01 - Complete Solution (but not all features work);
- Solution 02 - Simplified Solution - UX Researcher and Participant;
- Solution 03 - Simplified Solution - UX Researcher Only.

4.2.4 Idea Selection

Most of the time, coming up with original ideas is not the issue. There are always fresh ideas, even when companies are diminishing. The challenge for the majority of corporations is deciding which ideas to pursue in order to maximize business value. The future health and profitability of the company depend on making a wise choice. A business typically has much more ideas than resources to work on, so it must figure out how to identify the concepts that are most appealing (Koen, et al., 2002).

To assist in deciding which of the ideas presented in the previous topic should be followed, we will resort to a multicriteria decision method. This method resorts to numerical techniques that help in choosing an option from a discrete set of alternatives based on crossing the alternatives with the existing criteria. In this case, we will use the Analytic Hierarchy Process (AHP), whose main objective is to divide the decision problem into hierarchical levels, thus facilitating its understanding and evaluation.

4.2.4.1 Analytic Hierarchy Process

First, it is necessary to build the hierarchical decision tree that presents the objective of the decision, the criteria, and the alternatives. In terms of decision criteria, the following were defined:

- **Time:** Estimated duration for implementation of its features, given the project's deadline;
- **Adequation:** Ability to solve the proposed problem and respond to the needs of the target audience (UX Researchers);
- **Complexity:** Taking into account the number of features to be developed and their respective implementation difficulty.

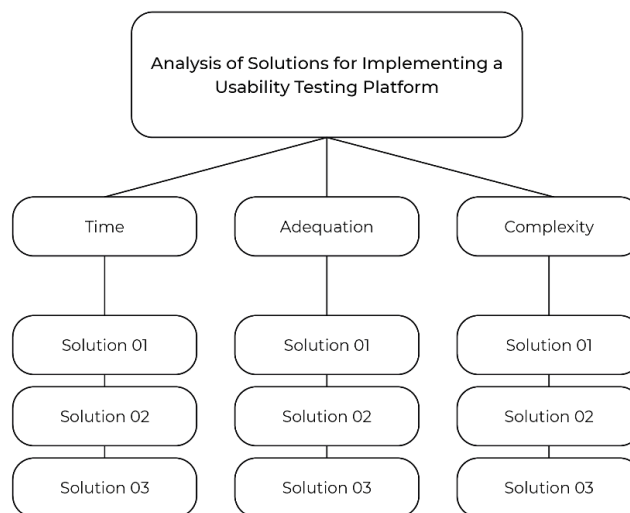


Figure 15. Hierarchical Decision Tree

Second, we need to establish priorities among the elements for each level of the hierarchy through a comparison matrix. For this, levels of importance were assigned in the comparisons following Saaty's Fundamental Scale (Saaty, 1980):

Intensity of Importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgment slightly favor one activity over another
5	Essential or strong importance	Experience and judgment strongly favor one activity over another
7	Demonstrated importance	An activity is strongly favored and its dominance is demonstrated in practice
9	Absolute importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2, 4, 6, 8	Intermediate values between the two adjacent judgments	When compromise is needed

Figure 16. Saaty's Fundamental Scale (Nilsson, et al., 2016)

The following table demonstrates the second-level criteria comparison matrix, containing a line that shows the sum of the values of each column:

	Time	Adequation	Complexity
Time	1	1/5	3
Adequation	5	1	4
Complexity	1/3	1/4	1
Sum	19/3	29/20	8

Table 4. Second-level Criteria Comparison Matrix

Then, the matrix values were normalized, dividing each value by the sum of its respective column.

	Time	Adequation	Complexity
Time	3/19	4/29	3/8
Adequation	15/19	20/29	1/2
Complexity	1/19	5/29	1/8

Table 5. Second-level Criteria Normalized Matrix

To obtain the priority vector, the average of the values of each line was calculated. Note that the values in the "Relative Priority" column have been rounded to two decimal places.

	Time	Adequation	Complexity	Relative Priority
Time	3/19	4/29	3/8	0.22
Adequation	15/19	20/29	1/2	0.65
Complexity	1/19	5/29	1/8	0.12

Table 6. Priority Vector

Once the priority vector is obtained, it is necessary to calculate the Consistency Ratio (RC) to measure how consistent the judgments were with respect to large samples of completely random judgments. First, we multiply the normalized matrix (A) by the priority vector (x):

$$Ax = \begin{bmatrix} 3/19 & 4/29 & 3/8 \\ 15/19 & 20/29 & 1/2 \\ 1/19 & 5/29 & 1/8 \end{bmatrix} \times \begin{bmatrix} 0.22 \\ 0.65 \\ 0.12 \end{bmatrix} \approx \begin{bmatrix} 0.17 \\ 0.69 \\ 0.14 \end{bmatrix}$$

Equation 1. Multiplication of Normalized Matrix A by Priority Vector x

Next, Ax is averaged with the priority vector (x), obtaining the eigenvalue (λ_{\max}).

$$\lambda_{\max} = \frac{\left(\frac{0.17}{0.22} + \frac{0.69}{0.66} + \frac{0.12}{0.14}\right)}{3} \approx 0.99$$

Equation 2. Calculation of Eigenvalue λ_{\max}

Then, we calculate the consistency index (IC) – the n value being the total number of second-level criteria:

$$IC = \frac{(\lambda_{\max} - n)}{(n - 1)} = \frac{(0.99 - 3)}{(3 - 1)} = -1.005$$

Equation 3. Consistency Index Calculation

The next step is to calculate the RC with the formula:

$$RC = \frac{IC}{IR}$$

Equation 4. Consistency Ratio Formula

IR is a random index referring to a large number of pairwise comparisons performed. This is a random index calculated for square matrices of order n (Saaty, 1980). The following table defines the IR values according to the number of criteria:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Table 7. IR values for Square Matrices of Order n (Saaty, 1980)

Each of the numbers in this table is the mean ICs derived from a sample of randomly selected reciprocal matrices using the AHP scale. An RC of 10% or less implies that the adjustment is small compared to the current values of the inputs. An RC as high as, say, 90% would mean that the judgments are virtually randomly paired and are completely unreliable.

In this case, we should look at the value of $n = 3$, which is 0.58, and calculate our RC:

$$RC = \frac{-1.005}{0.58} \approx -1.73$$

Equation 5. Consistency Ration Calculation

Because our RC is < 0.1 , we can conclude that our relative priorities' values are consistent.

The next phase is the construction of the parity comparison matrix for each criterion, considering each of the selected alternatives:

	Solution 01	Solution 02	Solution 03
Solution 01	1	7	9
Solution 02	1/7	1	5
Solution 03	1/9	1/5	1
Sum	79/63	41/5	15

Table 8. Parity Comparison Matrix for Time

	Solution 01	Solution 02	Solution 03
Solution 01	1	4	5
Solution 02	1/4	1	3
Solution 03	1/5	1/3	1
Sum	29/20	16/3	9

Table 9. Parity Comparison Matrix for Adequation

	Solution 01	Solution 02	Solution 03
Solution 01	1	5	7
Solution 02	1/5	1	4
Solution 03	1/7	1/4	1
Sum	47/35	25/4	12

Table 10. Parity Comparison Matrix for Complexity

Then we normalize each of these matrices and obtain the priority vector of each one. It is important to note that the “Relative Priority” values were, again, rounded to two decimal places.

	Solution 01	Solution 02	Solution 03	Relative Priority
Solution 01	63/79	35/41	3/5	0.75
Solution 02	9/79	5/41	1/5	0.15
Solution 03	7/79	1/41	1/15	0.06

Table 11. Priority Vector for Time

	Solution 01	Solution 02	Solution 03	Relative Priority
Solution 01	20/29	3/4	5/9	0.67
Solution 02	5/29	3/16	1/3	0.23
Solution 03	4/29	1/16	1/9	0.1

Table 12. Priority Vector for Adequation

	Solution 01	Solution 02	Solution 03	Relative Priority
Solution 01	35/47	4/5	7/12	0.71
Solution 02	7/47	4/25	1/3	0.21
Solution 03	5/47	1/25	1/12	0.08

Table 13. Priority Vector for Complexity

Almost finishing, we now need to multiply each relative priority value obtained with the one we got at the beginning of the method, giving us the weight of each criteria:

$$Time = \begin{bmatrix} 0.75 \\ 0.15 \\ 0.06 \end{bmatrix} \times \begin{bmatrix} 0.22 \\ 0.66 \\ 0.12 \end{bmatrix} \approx 0.27$$

Equation 6. Weight of “Time” Factor

$$Adequation = \begin{bmatrix} 0.67 \\ 0.23 \\ 0.10 \end{bmatrix} \times \begin{bmatrix} 0.22 \\ 0.66 \\ 0.12 \end{bmatrix} \approx 0.31$$

Equation 7. Weight of “Adequation” Factor

$$Complexity = \begin{bmatrix} 0.71 \\ 0.21 \\ 0.08 \end{bmatrix} \times \begin{bmatrix} 0.22 \\ 0.66 \\ 0.12 \end{bmatrix} \approx 0.3$$

Equation 8. Weight of “Complexity” Factor

In the last step, we obtain the composite priorities of the alternatives, multiplying the priority matrix (a matrix containing all of the relative priorities for each criteria) with each criteria’s weight:

$$\begin{bmatrix} 0.75 & 0.67 & 0.71 \\ 0.15 & 0.23 & 0.21 \\ 0.06 & 0.1 & 0.08 \end{bmatrix} \times \begin{bmatrix} 0.27 \\ 0.31 \\ 0.3 \end{bmatrix} \approx \begin{bmatrix} 0.62 \\ 0.17 \\ 0.07 \end{bmatrix}$$

Equation 9. Composite Priority Calculation

Finally, based on the established criteria and their significance, **Solution 01** would be the most favorable to follow because it received a score of 0.62. The next option to follow would be Solution 02, even though it does not significantly differ from Solution 03. However, it is important to note that the criteria, in this case, is working “backwards”, meaning that, if Solution 01 received a higher score, it’s because it takes more time, is more complex but is more adequate to the problem. This is due to the attribution of levels of importance made in the beginning of the method: a higher number of importance, in this case, means that it would be important to choose this solution over others, as defined on the Saaty’s Fundamental Scale, but it does not mean that it is simpler.

4.2.5 Concept Definition

The new concept development model's final component is concept definition, which provides the exit to the NPD and/or TSG. The innovator must present a strong case for investing in the commercial or technological offer in order to get past the gate. This is referred to as a "win statement" by some organizations. The final result for this part as the concept proceeds through NPD and/or TSG is often developing a business plan and/or a formal project proposal. This offers a chance to reinforce and improve the concept.

By creating a web application that unifies the steps involved in conducting usability tests, from their creation to execution, with a special focus on the selection of participants for them, which is the main problem to be solved, it is possible to profit from the opportunity identified in the

first step of this list. In our case, the objectives of the project have previously been defined on the introduction section of this dissertation.

4.3 Solution's Value

Value creation is essential to any business: any commercial activity revolves around exchanging some tangible and/or intangible good or service and having its worth recognized and rewarded by customers or clients, whether inside the firm or collaborative network or outside. "Value has been defined in different theoretical contexts as need, desire, interest, standard /criteria, beliefs, attitudes, and preferences." (Nicola, et al., 2012).

Depending on whether it refers to the producer or the client, value typically has various meanings. From the producer side, loyal clients can be valuable because they have several advantages for the business. On the other hand, the client's value typically results from factors like the standard of a good or service they purchased, the amount of work they used to do so, the social contacts they had with the provider, among others.

4.3.1 Value for the Customer and Perceived Value

Value for the customer refers to the perception of benefits, costs, and satisfaction that a customer experiences from purchasing and using a product or service. It is the difference between the customer's overall experience and their expectations. This concept is a key idea in value analysis because it enables companies to comprehend the needs and wants of their consumers and customize their offers to fit those needs.

As different customers perceive different values for the same products/services, the concept "perceived value" appears: it refers to the value that a customer attributes to a product or service based on their personal evaluation of the benefits they receive compared to the costs they incur. In value analysis, this is an important concept because it can help us understand how customers perceive the value of their offerings and identify areas for improvement to enhance customer satisfaction and loyalty. To better assess the value for the customer, a benefits and sacrifices table based on the application's features is demonstrated below:

Features	Benefits	Sacrifices
Easily Find Participants for Tests	Quick Participant Finding.	Initial lack of participants registered in the application.
Provide Templates for Tests and Results' Analysis	Time Saving.	-
Combine multiple features on one platform	Time Saving.	Must pay a subscription plan to access more features.
Share insights with the team involved in the project	Team Collaboration.	-
Pay participants effectively	Less time spent on payments' logistics.	Possible security risks (fraudulent activities).

Share tests via a link	Time Saving.	-
Possible Integrations with other platforms	Practical to use.	-

Table 14. Benefits and Sacrifices for the Customer

4.3.2 Value Proposition

To demonstrate the correspondence between the needs of the clients and the benefits that result from using the application to be developed, we are using a value proposition canvas. Dr. Alexander Osterwalder created the Value Proposition Canvas as a framework to make sure that the product and market are compatible. The two main components of this instrument are the client profile and the value proposition of the organization (B2B International, 2023):

Client Profile:

- **Gains** – “the benefits which the customer expects and needs, what would delight customers and the things which may increase likelihood of adopting a value proposition”.
- **Pains** – “the negative experiences, emotions and risks that the customer experiences in the process of getting the job done”.
- **Job-to-be-done** – “the functional, social and emotional tasks customers are trying to perform, problems they are trying to solve and needs they wish to satisfy”.

Value Proposition of the Organization:

- **Gain Creators** – “how the product or service creates customer gains and how it offers added value to the customer”.
- **Pain Relievers** – “a description of exactly how the product or service alleviates customer pains”.
- **Products and Services** – “the products and services which create gain and relieve pain, and which underpin the creation of value for the customer”.

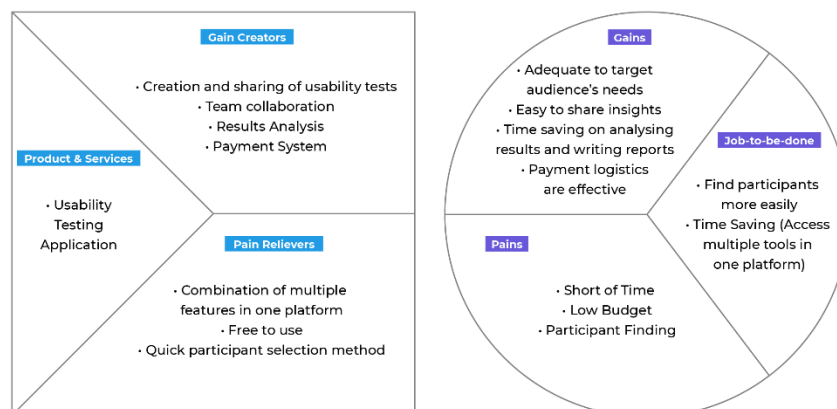


Figure 17. Value Proposition Canvas

In this case, the value proposition was made for the complete solution of the problem to be solved, which is the most appropriate to follow in an ideal situation (in which the time and effort of implementation will be greater than those foreseen for this dissertation). The product to be created is a usability test management application, which will combine multiple functionalities on a single platform, have a free plan, and aim to find test participants quickly, thus solving customer needs. In addition, it allows the creation and sharing of tests, collaboration between teams, analysis of results, and has a payment system.

4.3.3 Business Model

Another way of describing how an organization can create, deliver or capture value is through the development of a business model. In other words, it captures value for the organization and delivers this value to relevant stakeholders. This time, we are using a visual representation of a business model, commonly known as the business model canvas. This diagram highlights all of the important strategic components of a company, by providing a comprehensive and general overview of the business' operations, clients, sources of income, and more. Just like the value proposition canvas, the actual business model canvas was first proposed by Dr. Alexander Osterwalder as well. The blocks present on the business model canvas are (ProductPlan, 2023):

- **Customer segments** (who it's for);
- **Customer relationships** (how the team will support and work with its customer base);
- **Channels** (how the organization will market and sell it);
- The product's **value propositions** (what it does and promises);
- **Revenue streams** (how the product will make money);
- **Key activities** (the steps the team must complete to make it successful);
- **Key resources** (what personnel, tools, and budget the team will have access to);
- **Key partners** (how third parties will fit into the plan);
- **Cost structure** (what it costs to build the product as well as how to sell and support it).

Business Model Canvas				
Key Partners <ul style="list-style-type: none"> • Integration Platforms (i.e. Figma, Google, Adobe); • Participant Selection Platforms (i.e. Meta, LinkedIn). 	Key Activities <ul style="list-style-type: none"> • Database maintenance; • Application Development/Maintenance; • Customer Relationships; • Marketing. 	Value Propositions <ul style="list-style-type: none"> • Quick Participant Selection Method; • Combination of multiple features in one platform; • Reduces time spent on logistics; • Constant app improvement/adequation to our audience's needs; • Free plan. 	Customer Relationships <ul style="list-style-type: none"> • User Testing on our application; • Answer to user's reviews/phone calls or e-mails. 	Customer Segments <ul style="list-style-type: none"> • Companies with UI/UX design departments; • UX Researchers.
Key Resources <ul style="list-style-type: none"> • Web Hosting Services; • Database. 		Channels <ul style="list-style-type: none"> • Web Application. 		
Cost Structure <ul style="list-style-type: none"> • Hosting Service; • Database Management; • Application Maintenance; • Salaries; • Marketing. 			Revenue Streams <ul style="list-style-type: none"> • Paid plans to access more features; • Possible agreements with integrated platforms. 	

Figure 18. Business Model Canvas

The diagram in itself is self-explanatory but it is important to point out that our customer segments are, essentially, companies with UI/UX departments and UX researchers (our target audience). These are the two segments that contribute to our revenue streams: “paid plans to access more features” and “possible agreements with integrated platforms” (those being mentioned in the key partners section). Our cost structure would revolve around the application’s development and maintenance, as well as the salaries to pay to its development team (designers, developers, marketers, testers, etc.), along with the marketing costs to promote it and sell it.

4.4 Quality Function Deployment

To finalize the value analysis chapter, given the complete solution of the problem, we used a method intended to ensure that the client’s “wants” are kept in mind throughout the development of a new product. This method is called Quality Function Deployment and its initial phase is the House of Quality (HOQ), which collects desired traits from consumers and converts them into engineering characteristics. In other words, it helps on the identification of requirements and assigning of their priority levels.

					<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>						
--	--	--	--	--	---	--	--	--	--	--	--

Legend		
⊕	Strong Relationship	9
○	Moderate Relationship	3
▲	Weak Relationship	1
++	Strong Positive Correlation	
+	Positive Correlation	
—	Negative Correlation	
▼	Strong Negative Correlation	
▼	Objective Is To Minimize	
▲	Objective Is To Maximize	
X	Objective Is To Hit Target	

Figure 19. House of Quality

By analyzing the complete solution's HOQ, it is possible to verify that the top three quality characteristics (or functional requirements) more important to the demanded quality (the customer's requirements) are, in order:

- "Quick participant selection method";
- "Automatic Reports";
- "Combination of multiple tools in one platform".

This means that, when developing a complete solution to the problem, these are the main features to be developed and should be the ones to focus on a first stage of development.

5 Solution Design

The solution design had two different phases described in this chapter:

1. On the first phase, it was chosen the final solution to follow and presented a map of navigation through the application, its requirements, and the first sketch of its user interface for both of its user types – UX Researcher and Tester.
2. On the second phase, a usability test (System Usability Scale) was carried out on the first version of the design so that the final design could be elaborated through the feedback of the tested users. When designing the final version of the user interface, a visual identity of the application was created and the corrections made to the first version of the design were implemented.

5.1 Solution Chosen

In the previous chapter, it was determined that **solution 01** would be the most advantageous after applying the AHP technique to examine the three suggested solutions to the problem in accordance with the established criteria (time, adequation, and complexity) and their respective assigned importance. Solution 01, named "Complete Solution (but not all features work)," consists mainly of designing the complete solution as if it were to be implemented in its entirety, with only the primary features for solving the problem being developed, as a proof of concept.

When designing this solution, we realized that we had two types of users to address: the UX Researchers, whose purpose is to create and manage usability tests; and the Testers, who will participate in those tests. Along with having different purposes when it comes to using this application, both of these users will also have access to different pages (i.e. UX researchers would have access to test results but the testers would not; UX researchers can create usability tests but testers cannot; etc.). Consequently, we had to create two different designs, concerning each of these users' needs.

5.2 Navigation Map

Firstly, a navigation map was designed so that the navigation between the different pages could be better understood. The map is presented for both user types: UX Researcher and Tester. When signing up in the application you will need to choose which type of user you are and only then the application will adjust its pages according to your user type.

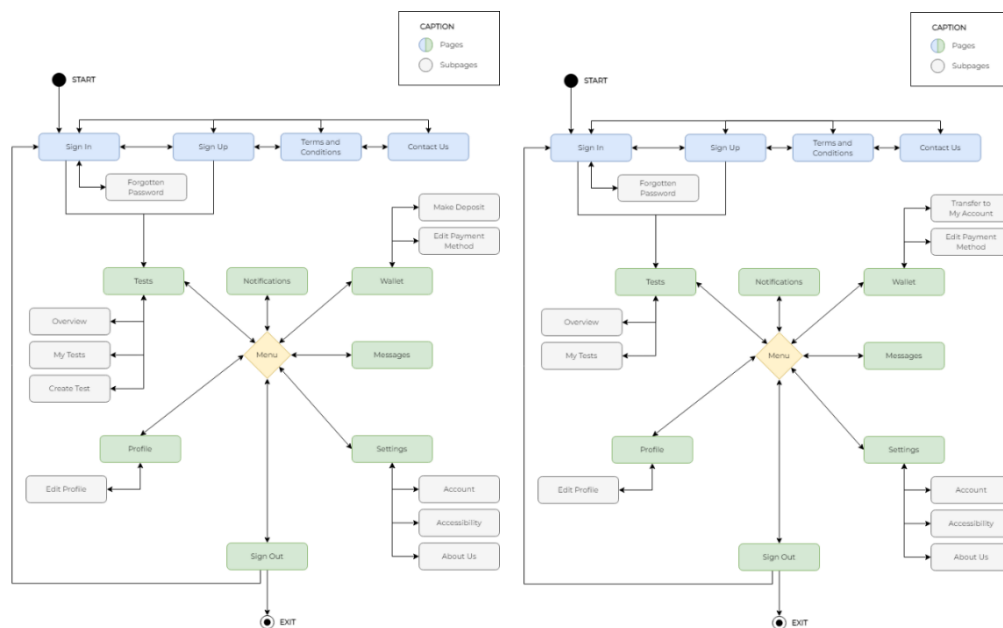


Figure 20. Navigation Map – User Type: UX Researcher (Left) and Tester (Right)

5.3 Design Requirements

In order to summarize what was covered in the solution planning chapter and more effectively incorporate them into the pages identified previously in both maps, the requirements for each user type are listed in Attachment A. It is important to note again that these requirements will not all be implemented in the project development phase, prioritizing only the implementation of those that directly solve the proposed problem – which will be later covered on the implementation stage.

Despite having some pages in common, what essentially distinguishes these two types of users is the “Tests” page and the information collected in the registration process. The “Tests” page shows completely different information for each of these users: UX researchers can create, edit and view created tests, while testers can only see available tests and tests to which they have already responded. Therefore, the “Overview” page also reflects different information for each of these users, adjusting the content that seems to be most suitable for their personal dashboards. Apart from this, the information of these two users is also distinguished on the “Notifications”, “Wallet” and “Profile” pages.

5.4 User Interface: First Draft

The first draft of the application's interface design was carried out on the Figma platform, since it allows for an interactive prototype of the solution, reflecting not only its UI but also fundamental UX aspects such as navigation between pages and the result of interaction with the various elements. It should be noted that, as this is a first draft, aspects such as a visual

identity - an application logo, its color palette, iconography, and typography - are not yet present. Moreover, a test of the application design will be performed before moving on to the final UI in order to more readily discover and fix issues before the implementation phase, depending on the experience of different tested users. This test will be described in the experimentation and evaluation chapter (7.1.1 and 7.2.1).

In Attachment B of this dissertation there are two links that reflect the UI of the pages for the two user types defined: UX Researcher and Tester, where 40 different artboards were designed. Consequently, in Attachment C there are the respective links to the interactive prototypes, being able to click on the various UI elements to interact with them and realizing their actions.

5.5 User Interface: Final Version

The final version of the application's interface design had two main tasks:

1. Creation of the visual identity (application name, logo, choice of color palette and typography);
2. Moving from the first draft of the design to its final version:
 - a. Firstly, by applying the visual identity created;
 - b. Then, by following good UI practices such as creating a grid and making sure all the elements were aligned onto it;
 - c. By personalizing all of the used icons to match the color palette chosen;
 - d. Finally, by taking into account the feedback received on the System Usability Scale (the results of which are presented in section 7.2.1).

As a result, the final design of the application allows its visualization as a high fidelity interactive prototype, as well as a better understanding of what is expected in the implementation phase of the solution.

5.5.1 Visual Identity

A clear and consistent visual identity can improve the user experience by making it easier for users to navigate and understand the application's interface. It can also help to create a cohesive and professional look and feel for the application, which can enhance its perceived value and appeal. For the name of the application, its main objective was taken into account: this application promotes speed in the creation and management of usability tests. Thus, the name "Rocketest" arose, combining the words "rocket" and "test": rocket as an allusion to the main characteristic of a rocket (its speed) and test as the focus of the application (usability testing). Then, its logotype was designed:



Figure 21. Application's Logotype

The logo intends to incorporate the design of a rocket in an arrow, alluding to the web application to which it belongs, while also bringing into shape the letter “t”, used for both the words “rocket” and “test”. The arrow, in turn, is associated with the user's mouse click when browsing the application or the digital products he/she will test.

The colors used belong to the chosen color palette, shown in the following image:



Figure 22. Application's Color Palette

Blue is frequently linked to professionalism, trust, and reliability, all of which are crucial characteristics for a program that is intended to simplify important tasks like usability testing. The shades of blue are visible in the first seven colors of the palette, with the last three intended to be used, respectively, in the background or in buttons' text (white), in the text inside inputs or small card subtitles (grey), and in alerts or as an indication of mandatory fields (red).

Finally, the typography chosen for both the logo and the application's text fields was Montserrat. It is a highly legible font with a modern, geometric style that makes it easy to read on screens of all sizes. Montserrat comes in a wide range of weights and styles, from thin to black, making it a versatile font that can be used for various types of content. It is also a web-safe font, which means that it is widely available and can be used across different browsers and devices without any issues.

5.5.2 Final User Interface

To design the final user interface, the previously created visual identity was implemented in the first draft of the UI design, assigning it its logo and chosen colors. Furthermore, all the present icons were adjusted so that they could incorporate the colors of the defined palette. Free-to-use images were also inserted, so that the fields in which they should appear could be better distinguished in the design.

Through the usability test – System Usability Scale – explained in section 7.2.1, it was noticed that the current design met the expectations and needs of the tested users, only obtaining as feedback the fact that it was not possible to access the testers who have already participated in the tests created. In order to respond to this suggestion for improvement, the design was modified so that it could include not only users who had already responded to the test, but also give more emphasis to the 'find testers' functionality, which is the main focus of this application, placing it on a separate page. This page can be accessed via the “Find Testers” button on the page called “Test Detail”.

The final user interface can be accessed on Attachments E and F of this dissertation, corresponding, respectively, to the UI artboards designed and the interactive prototype for both user types defined (UX Researcher and Tester).

6 Solution Implementation

The implementation of the solution went through the following steps that will be described throughout this chapter:

1. Technology used and its relation to the criteria previously defined to choose it;
2. Description of the implementation process (requirements review and prioritization, time management, version control system);
3. Explanation of the code's architecture (namely the folder structure, its components, pages, and database);
4. Brief overview of the queries used as well as two of the application's main pages on the code documentation (highlighting the 'Create Test' page and the 'Find Testers' one that makes use of an API created especially for the tester searching process);
5. Results obtained on the testing of the solution implemented (using the methodology described in the Experimentation and Evaluation chapter);
6. Final project (how to access it and run it).

6.1 Technology Used

When reviewing the state of art of the most common technologies for this type of project we decided to base our decision for the technology choice (section 2.3.1) on factors such as project's requirements, development expertise, availability of resources, and cost.

When it comes to the development expertise, I was already familiarized with the three main web technologies (HTML, CSS and JavaScript), and had a slight interest in learning React.js, as I already had taken part in a workshop, and it seemed suitable for this project to use this technology. React leverages JSX, a markup syntax that combines HTML and JavaScript, as discussed in the state of the art chapter, making it intuitive for anyone who has prior experience with web development.

It is an undeniable fact that React has gained significant popularity and has a large and active community of developers. This means there are abundant resources, libraries, and community-driven solutions available for common challenges, which ultimately contributed to the availability of resources factor. It is also a cost-free open-source technology. As also reviewed in the state of art chapter, React follows a component-based architecture, which promotes reusability, modularity, and maintainability of code. When designing this application on Figma, we also had several design components that could easily become React components.

Given all these reasons, React.js seemed to be the most appropriate technology to follow in this project, when it came to the frontend development. When performing more research on this technology, I came across a framework called Next.js (Next.js, 2023).

6.1.1 Next.js

According to (Vercel, 2023), some of the main Next.js features include:

- **Routing:** A file-system based router built on top of Server Components that supports layouts, nested routing, loading states, error handling, and more;
- **Rendering:** Client-side and Server-side Rendering with Client and Server Components. Further optimized with Static and Dynamic Rendering on the server with Next.js. Streaming on Edge and Node.js runtimes;
- **Data Fetching:** Simplified data fetching with *async/await* support in React Components and the *fetch()* API that aligns with React and the Web Platform;
- **Styling Support:** for your preferred styling methods, including CSS Modules, Tailwind CSS, and CSS-in-JS;
- **Optimizations:** Improved Image Component with native browser lazy loading. New Font Module with automatic font optimization;
- **Typescript:** Improved support for TypeScript, with better type checking and more efficient compilation, as well as support for optional chaining and nullish coalescing;
- **API Reference:** Updates to the API design throughout Next.js.

This framework seemed very convenient for this project because while other React libraries or frameworks may require additional setup/configuration or third party packages for many features (namely: routing, code splitting, server-side rendering, static site generation, API routes, etc.), Next.js offers these features by default. Additionally, Next.js benefits from a robust environment and a sizable developer community. It has extensive documentation, tutorials, and community-driven packages, making it easier to find solutions and get support when needed, which was a very important criteria for choosing this framework.

However, the fact that Next.js has built-in API routes, enabling the establishment of serverless API endpoints within our application, was what truly stood out when picking this framework. This streamlines the development process and lessens the complexity of the infrastructure by eliminating the need to set up a separate backend server to handle API requests. In other words, we would not require any other external backend solutions if we used this framework for both the frontend and the backend of the application. Because they were no longer required, we discarded the backend technologies covered in the state of the art chapter.

Unfortunately, Next.js does not come with a database, therefore we had to choose one for this project:

6.1.2 PostgreSQL

The choice of database was essentially based on the development expertise factor: although the databases reviewed in the state of the art chapter were quite popular, my experience with databases was not very extensive, having only worked with PostgreSQL and Firebase in the past, in a very simplified way. Taking into account the other previously defined factors for choosing

the most appropriate technology for this project, we chose PostgreSQL as the ideal solution, as we will explain below.

“PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.” (PostgreSQL, 2023). It is a powerful, flexible, feature-rich and cost-free database server known for its adherence to SQL standards. When it comes to this project’s specific requirements, this database not only fulfills them but can also handle future advanced requirements, providing support for complex data types and an extensive list of useful features (PostgreSQL, 2023).

Since I had prior experience with this database, using it allows me to speed up the development process and reduce the learning curve associated with working with a new database technology. Additionally, PostgreSQL integrates well with many development frameworks and tools, including Next.js, ensuring a wide range of compatible resources and libraries.

By choosing this database management system we needed two more tools: one for setting up the database (ElephantSQL) and another one for administering and developing it (pgAdmin).

6.1.2.1 ElephantSQL

ElephantSQL (ElephantSQL, 2023) is a managed PostgreSQL database that offers easy creation and configuration of the PostgreSQL database, as well as managed backups and monitoring, scalability options, and high availability. It is also a cloud-based service.

6.1.2.2 pgAdmin

pgAdmin (pgAdmin, 2023) is a popular open-source administration and development platform for PostgreSQL that provides features like database creation, table management, query execution, backup and restore operations, user management, and more (pgAdmin, 2023). It offers a visual interface for interacting with the database, writing and executing SQL queries, and monitoring database performance.

6.2 Implementation Process

The methodology employed for the implementation of the chosen solution focused on a well-defined and structured approach where we:

1. Reviewed the requirements to implement and prioritized them according to their importance;
2. Defined a version control system, so that all requirements could be properly met and referred to as issues on a commit, and a history of platform development could also be accessed.

6.2.1 Requirements Review and Prioritization

To recap, solution 01 aims to implement all the designed pages in Figma shown in the previous chapter, however, only the features mentioned in section 3.2.1 of this dissertation would work. In other words, the most important pages to be developed were the authentication ones and the “Tests” page.

The first step to begin the implementation process was to make a list of the requirements that needed to be implemented. For this, since the construction of a QEF (Quality Evaluation Framework) was already defined as one of the solution’s evaluation methodologies in chapter 7, and taking into account that the first task of the QEF’s creation is to make a list of the requirements to be evaluated, this task was merged into one. The QEF methodology also requires the assignment of a relevance to each requirement based on its priority for implementation, which is explained in section 7.1.2.

6.2.2 Version Control System

With a version control system, we could easily track and manage code changes, maintaining a clear history of each change. If needed, it becomes simpler to revert to a previous working version and in case of any unexpected event such as loss of files, the project would always be hosted in a remote repository, which could allow for greater confidence and the rescue of previous stable versions.

In our case, we used GitHub as the version control system for this project, which also allowed us to create issues (correspondent to the QEF requirements) that would be referenced in the commits made, ensuring that all of the requirements would be attended. This practice has also advantages in terms of streamlining the debug process, contributing to a more organized and efficient development workflow.

The project’s repository on GitHub can be visited in Attachment G of this dissertation (please make sure to read the README.md file for instructions on how to run the project). To run this project on your local environment you have to enter the credentials on “.env.local” file. “EXAMPLE.env.local” is an example of how this file should look like.

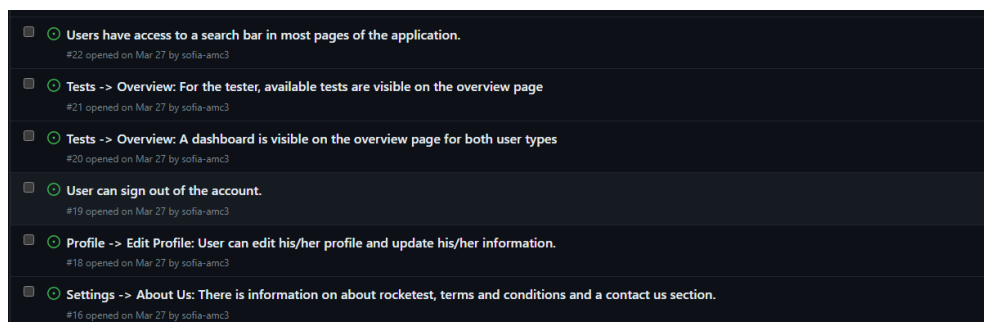


Figure 23. Example of some of the Project’s Issues on GitHub

6.3 Code Architecture

In this section, the architecture of the code will be summarized, focusing on important aspects such as:

1. The organization of the project's folders;
2. The structure of the components;
3. The pages created, reviewing the main pages of the project, the concept of routing, and the distinction of user types;
4. The database developed, showing an entity-relationship diagram, how the database was connected to the project, an example of an endpoint and a brief explanation of all the queries implemented.

6.3.1 Folder Structure

When we first open the folder containing the Rocketest application, we are faced with the four main folders of the project, followed by some configuration files:

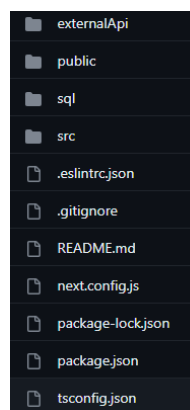


Figure 24. Project's Main Folders

The “externalApi” folder, regarding the API developed, will be reviewed later on 6.3.5; the “public” folder contains static files such as images and icons; the “sql” folder contains examples of database queries used throughout the project (i.e. database creation, insert user, etc.), used only for testing purposes (on pgAdmin), since the project's queries will be present inside the “src” folder; and the “src” (source) folder contains essentially the application code organized in different folders, which we can observe in the image below:

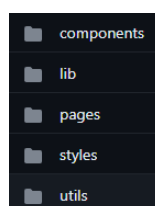


Figure 25. Project “src” Folder Contents

These five folders play a vital role in organizing and structuring the codebase of our project, contributing to the overall functionality of it:

- The “components” folder holds reusable UI components that are used across different pages of the application. On section 6.3.2 we will better understand their structure;
- The “lib” folder, which typically serves as a centralized location for shared logic and functionality, contains the database connection file (which we will review on section 6.3.4) and a connection to the application’s e-mail;
- The “pages” folder is the fundamental directory of the project, as it contains the pages of the application. On section 6.3.3 we will get more insight on how this folder is organized;
- The “styles” folder is used to store the CSS files as well as the fonts used in the project. It contains not only a general CSS file, but also a specific one, used across different pages and components;
- Finally, the “utils” folder contains utility functions or modules that are used across the application for various purposes, in our case:
 - lists of hobbies and careers that a user can choose from when creating his/her account or when choosing criteria for usability test’s participants;
 - a test creation helper file, containing all the necessary functions to create a test (the test creation process will be reviewed on 6.3.6);
 - a “user” file, used to store the user's authentication data.

6.3.2 Components

As previously reviewed on 2.3.2.1, React.js uses a component-based architecture that encapsulates specific functionalities and UI elements, promoting code reusability and modularity. In our case, we were following the Figma prototypes shown on the solution design chapter to develop the application and Figma itself works with components to ease the design process and maintain consistency. This means that we could predict beforehand which design elements could become components on the code.

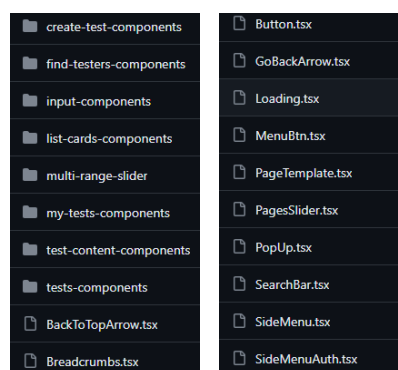


Figure 26. Project “components” Folder

The “components” folder was organized according to the different pages of the application, as seen on Figure 62. As an example, pages like “My Tests” would have their components inside the folder “my-tests-components” to provide a better organization of this folder. On the contrary, components that would be commonly used in many pages of the application – such as “Breadcrumbs”, “Button”, “GoBackArrow”, “Loading”, etc. - would not have a folder assigned.

6.3.3 Pages

When looking at the “pages” folder inside the “src” of our code, we can visualize the many different pages created, along with a folder called “api”:

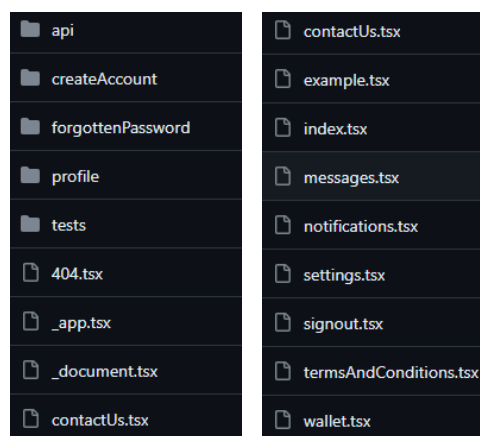


Figure 27. Project “pages” Folder

The HTML page that is rendered on the server and provided to the client can be modified with the help of a special Next.js file called “**_document.tsx**”. It gives users a mechanism to include specific markup, stylesheets, scripts, or meta tags on every page of the application.

The “**_app.tsx**” file contains important instructions for initializing the application, serving as a wrapper component that wraps around all the other pages. This file allows the definition of global styles, setting up shared components or context providers, and handling any global application state or behavior. For example, in our application there is always a side menu to display on the left side of the screen, so here we include this component, because it is supposed to appear inside every page.

The default page that serves as the entry point or start of the application is the “**index.tsx**” file. In our case, this file goes for the “Sign In” page in case there is not an authentication saved on the browser’s local storage, and in case there is, then it goes to the “Tests” page, which is the main page of Rocketest.

The “**api**” folder, in turn, does not contain any application pages but rather files containing the server-side code and logic for handling API requests and responses. In other words, it manages the backend of the application, with files that define various API routes and endpoints, which we will explain later in section 6.4.1.

We will explain next a few more details used on the development of the application's pages.

6.3.3.1 Routing

In a Next.js application, each page corresponds to a particular route. Developers can effortlessly design multi-page applications since Next.js serves the corresponding page whenever a user accesses a specific URL. In this case, the file name represents the route path, being "index.tsx" the starting point or the entry file. Let us look inside the "tests" folder:

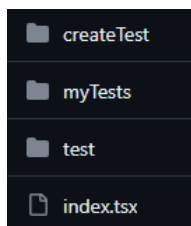


Figure 28. Project "tests" Folder

In this case, "index.tsx" is the file to be rendered whenever we click on the "Tests" page of our application (corresponding to "Overview" page). Then, we can access the other folders whenever we click on one of the top menu pages of the "Tests" page.

Assuming we are running the project locally on port 3000, the URL when clicking on the "Tests" page should look like:

<https://localhost:3000/tests>

In turn, if we click inside any page inside the "Tests" page, such as "My Tests", it should look like:

<https://localhost:3000/tests/myTests>

And in this case, we are rendering "index.tsx" inside the "myTests" folder.

Next.js supports dynamic routes as well by using brackets [] in the file name. For example, inside the "test" folder we have a file called "[_id].tsx" that will match routes like:

[https://localhost:3000/tests/test/\[_id\]](https://localhost:3000/tests/test/[_id])

Where the "_id" can be, for example:

<https://localhost:3000/tests/test/01>

The value within the brackets can vary, but it is useful to get the test we are looking for by giving it the same id as the one in its respective database column's row.

6.3.3.2 User Type Recognition

As it was mentioned in 5.1.1, Rocketest has two possible user types, defined at the moment of registering in the application: UX Researcher or Tester. This means that some pages of the

application, despite having the same name, should render differently for both these users, that is, they should display different designs.

To do that, we must check the user type saved in the database before rendering the page we want, using a simple piece of coding such as:

```
{userType == "1" ? <SignUpTester /> : <SignUpUxResearcher />}
```

Code Snippet 3. User Type Recognition

In this case, we are saving the user type as “0” in case it is a UX Researcher and “1” if it is a Tester. This code shows that, if the user type is equal to 1 (therefore, it is a Tester), the page to be rendered should be the Sign Up for Tester (<SignUpTester />), otherwise, it should show the Sign Up for the UX Researcher (<SignUpUxResearcher />), having these pages different layouts and information to display.

6.3.4 Database

6.3.4.1 Entity-Relationship Diagram

Before creating our database, it was designed an Entity-Relationship Diagram shown in the image below:

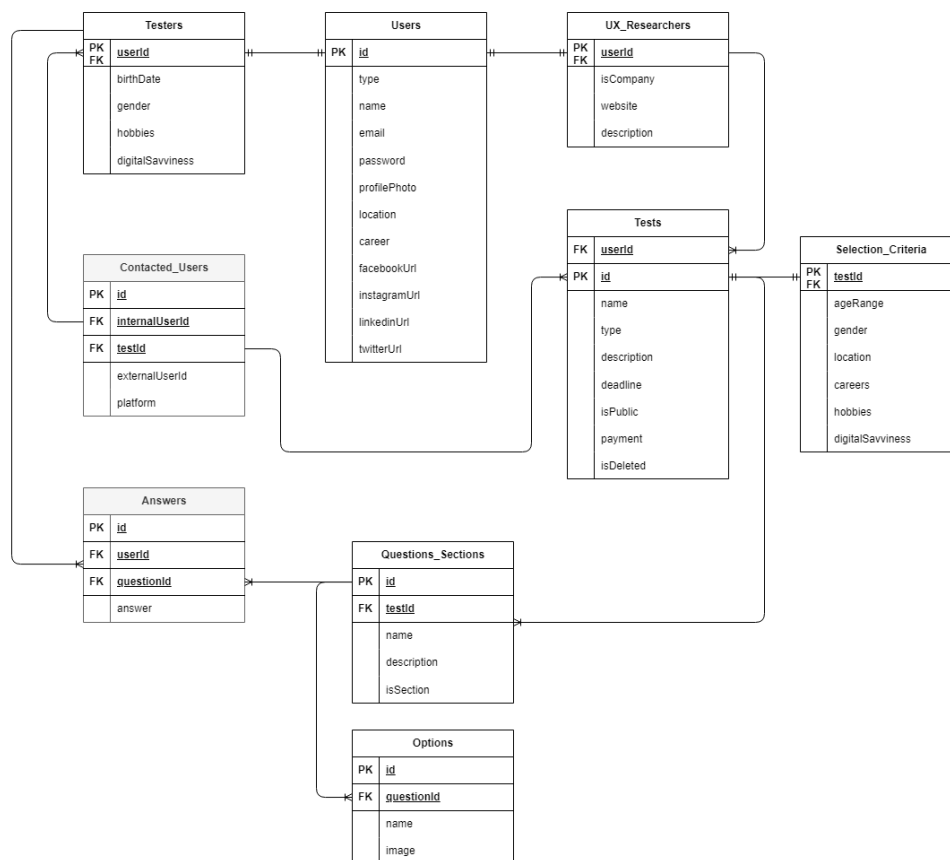


Figure 29. Entity-Relationship Diagram

This helped on understanding and planning the relationships and connections between the different entities in our database before actually implementing it. This diagram can also be useful for the people who are not familiar with the project to visualize the database structure.

In our case, the main table would be the “Users” table, as most of the information on our application is related to or depends on them. The second most important table would be the “Tests” one, as it stores all the information related to the usability tests (from its content to the answers received, as well as criteria for those who wish to answer that test). It is important to note that this database was specifically designed for the application’s features that are currently developed. In the future, if the application were to work in its entirety, with expansions in its functionalities, additional work would be required to enhance the database accordingly.

6.3.4.2 Database Connection

The connection from the Next.js project to our PostgreSQL database was made on the “db.js” file present inside “src/lib”. For this, we installed in our project a package called “Pool” (npm, 2023), and the code for the connection looked like this:

```
import { Pool } from "pg";

export const pool = new Pool({
  database: process.env.PGDATABASE,
  host: process.env.PGHOST,
  port: process.env.PGPORT,
  user: process.env.PGUSER,
  password: process.env.PGPASSWORD,
});
```

Code Snippet 4. Database Connection

On the root directory of the project there is a file called “.env.local” containing the following environment variables: database name, host, port, user and password (these are the variables that this file is accessing). Because we are talking about private credentials, storing them in a separate file ensures that sensitive information is not exposed in public repositories.

6.3.4.3 Endpoint Example

Inside the “api” folder of the project is where we find the logic for handling database requests and responses (endpoints that the frontend can access to retrieve information) – separated into two different folders: “user” and “tests”, concerning, respectively, the user and the tests. As an example, let us check the “login.tsx” file from the “user” folder:

```
import { NextApiRequest, NextApiResponse } from "next";
import { pool } from "@/lib/db";

const loginHandler = async (req: NextApiRequest, res: NextApiResponse) => {
  switch (req.method) {
    case "POST":
      const { email, password } = req.body;

      // Checks if both fields were fulfilled
      if (!email || !password)
```

```

        return res.status(400).send("Please provide both e-mail and
password.");

    try {
        const result = await pool.query(
            `SELECT * FROM "Users" WHERE email = '${email}';`
        );

        if (result.rows.length === 1) {
            const user = result.rows[0];

            if (user.password === password)
                return res.status(200).send(user); // Login successful
            else return res.status(400).send("Incorrect Password."); // The
password does not match the one in the database
        } else {
            return res.status(400).send("Invalid E-mail."); // The e-mail
does not exist
        }
    } catch (error) {
        return res.status(400).send(error);
    }
    break;

    default:
        return res.status(500).send("There was a problem with the
connection.");
        break;
    }
};

export default loginHandler;

```

Code Snippet 5. Login Request Handler

The “loginHandler” function takes in the request (“req”) and response (“res”) objects as parameters. Inside the function, it uses a switch statement to handle different HTTP (Hypertext Transfer Protocol) methods. In this case, it specifically handles the "POST" method, since we want to ensure that the data is sent in the request body rather than as part of the URL or query parameters.

When the method is "POST," the function extracts the email and password from the request body. It then attempts to execute a SQL query using the pool object to check if there is a matching user in the "Users" table with the provided email. If a user is found and the password matches the one in the database, it sends a response with a status code of 200 (meaning that the request was successful) and the user data. If no user is found, it throws an error to indicate a login failure. Any caught errors trigger a response with a status code of 400 (meaning Bad Request). If the HTTP method is not "POST," it sends a response with a status code of 500 (meaning Internal Server Error) and an error message indicating a problem with the connection.

Finally, we can reference this “login.tsx” file in our “index.tsx” whenever the “Sign In” button is clicked on, by calling this function:

```
const handleLogin = async () => {
```

```

        await axios
        .post("/api/user/login", form)
        .then(async (res) => {
            await userSession.setItem(JSON.stringify(res.data),
form.rememberMe);
            router.push("/tests");
        })
        .catch((error) => {
            if (error.response && error.response.data) {
                alert(error.response.data); // specific error
messages defined in the login.tsx file
            } else {
                alert(error.message); // default error message
            }
        });
    });
};

```

Code Snippet 6. Sign In Process

Using the “Axios” library (Axios, 2023), we are firstly sending a “POST” request to the server's “api/user/login” endpoint, passing the “form” data as the request payload (in this case, the “form” consists of the e-mail and password fields, as well as the “remember me” checkbox - used only in the frontend to save the login information for future sessions). If the request was successful:

1. We are then storing the response data in the browser's storage. In this case, the “auth” item will be useful to understand if the user is authenticated or not, which is why we previously used it in a state variable initialized with the value of *null* in our file “_app.tsx”.
2. We redirect the user to the “Tests” page, using the “router” object.

6.4 Code Documentation

In this section, four essential aspects of the source code of our project will be documented, highlighting:

1. An explanation of the queries used, to better understand their context, purpose and the type of user they are intended for;
2. A special case of the project – the 'Create Test' page – which was a target of a more complex implementation process, making use of an auxiliary file created to facilitate the creation of tests;
3. The 'Find Testers' page, which underwent changes compared to its initial design in Figma. This page stands out for being the focus of our project and the one that intends to prove the proposed hypothesis;
4. Finally, still on the 'Find Testers' page, the API developed for searching for testers when creating a test, mentioning the limitations encountered when using APIs from external platforms and briefly explaining the process of developing a Mock API to simulate what would be expected to happen when searching for users in external databases.

6.4.1 Queries Explanation

For a comprehensive understanding of how data is fetched, stored, and manipulated within the application, a table has been organized to provide information about all the queries used in the application.

This table offers an overview of each query's file name in the application's source code, which can be found inside the "api" folder. Additionally, it includes the HTTP methods utilized, such as "GET" or "POST," along with detailed descriptions that outline the purpose and functionality of each query. Furthermore, the table specifies the pages within the application where these queries are invoked, facilitating a clear understanding of their usage context, as well as the user type that has access to them (either UX Researcher, Tester, or both).

File Name	Method Used	Description of Query	Pages where it is called	User Type
login.tsx	POST	Verifies if the user's e-mail and password are correct and then starts the sign in process.	· Sign In	Both
registerTester.tsx	POST	Verifies if all the information provided is valid and if all mandatory fields were fulfilled and registers the user in the database.	· Sign Up	Tester
registerUx Researcher.tsx	POST	Verifies if all the information provided is valid and if all mandatory fields were fulfilled and registers the user in the database.	· Sign Up	UX Researcher
sendEmail.tsx	POST	Verifies if e-mail exists in database just so the user can receive a link in his/her e-mail to reset the password.	· Recover Password	Both
setNewPassword .tsx	POST	If the user exists in Rocketest (we verify if the e-mail exists in the database), lets the user update his/her password.	· Recover Password	Both
overview Tester.tsx	GET	Gets available tests that are public and checks for matched criteria between the tester and the test's defined criteria.	· Tests	Tester
answerTest.tsx	POST	Saves the test's answers provided by the tester.	· Tests	Tester
getTestForm.tsx	GET	Gets the test's data (such as questions, sections, options, etc.) to show to the tester when he/she wants to take a test and	· Tests > Create Test (UX Researcher)	Both

		to show to the UX Researcher when he/she wants to preview the test.	· Tests (Tester)	
insert.tsx	POST	Saves test's information after it is created.	· Tests > Create Test > Test Details	UX Researcher
myTestsTester.tsx	GET	Gets tests already answered by the Tester.	· Tests > My Tests	Tester
myTestsUxResearcher.tsx	GET	Gets tests created by the UX Researcher.	· Tests > My Tests	UX Researcher
testDetailUxResearcher.tsx	GET	Retrieves the test's information, as well as testers who have answered this test, also checking for matched criteria.	· Tests > My Tests > Test Detail	UX Researcher
getAnsweredTestForm.tsx	GET	Gets the test's answers given by the tester.	· Tests > My Tests > Test Detail	Tester
deleteTestUxResearcher.tsx	POST	Updates the test to become unavailable to all users.	· Tests > My Tests > Test Detail	UX Researcher
contactUsersUxResearcher.tsx	POST	When searching for testers on Find Testers, this query marks users as contacted by saving them in a Contacted Users table. If the user is from Rockettest, we are using an internalId correspondent to the user id and, if not, an externalId is given.	· Tests > My Tests > Test Detail > Find Testers	UX Researcher
findTestersUxResearcher.tsx	GET	Find testers that match the previously defined criteria for the test.	· Tests > My Tests > Test Detail > Find Testers	UX Researcher
getTestersApiUxResearcher.tsx	GET	Get contacted users from external platforms (in this case, from the API created).	· Tests > My Tests > Test Detail > Find Testers	UX Researcher
getTestResultsUxResearcher.tsx	GET	Gets test's results, aggregates the number of answers obtained for each option and the total number of testers that have answered the test.	· Tests > My Tests > Test Detail > Test Results	UX Researcher

Table 15. Queries Explanation Table

6.4.2 Create Test

The “Create Test” page, regarding especially the “editTest.tsx” file, was a special page in this project since it made use of the file “testCreatorHelper.tsx” present in the “utils” folder, which ultimately worked as a structure for the tests creation. Because these two mentioned files are correlated, this page had a more complex implementation than the other pages, which we will review in this subchapter by explaining the test creator helper file.

The “testCreatorHelper.tsx” file defines several interfaces that represent different aspects of the main form used for test creation:

```
export interface Option {
  id: number;
  name: string;
  imgSrc: string;
  imgName: string;
}

export interface Question {
  id: number;
  name: string;
  options: Option[];
  isSection: false;
}

export interface Section {
  id: number;
  name: string;
  description: string;
  isSection: true;
}

export interface Form {
  testName: string;
  testType: string;
  testCreator: string;
  testDescription: string;
  question_section: (Question | Section)[];
}
```

Code Snippet 7. Test Creator Helper Interfaces

The “Form” structure is the main form used for creating tests, including the test’s main properties (name, type, creator and description) and an array of questions and sections representing the test’s content (Questions, Sections and Options – the latter being inside the questions).

Then, after creating the base variables and helper methods (“optionTemplate”, “question_sectionTemplate” and “formTemplate”) to assign initial values to each one of the interfaces’ properties, we created the following functions: “optionCreator”, “optionDelete”, “question_sectionCreator” and “question_sectionDelete”, whose names are self-explanatory, as they basically let the user create or delete these structures. It is important to note that in

these functions we also validate certain criteria, such as fields being non-empty and within character limits.

Finally, we created update functions for each element that the user could interact with, to update the test data according to the user's input:

- "updateNameQuestionSection": To update the name of a question or section;
- "updateDescriptionSection": To update the description of a section;
- "updateTextOption": To update the text inside each option;
- "updateImageOption": To update the image inside each option;
- "deleteImageOption": To delete an image inside an option.

The file "editTest.tsx" makes use of helper functions from the "testCreatorHelper.tsx" to perform various operations, such as validating inputs, updating the test data, and generating unique IDs for questions. The helper functions from "testCreatorHelper.tsx", having a more complex logic, make the code in "editTest.tsx" more concise and modular, which promotes code reusability and maintainability, as the same functions can be used in other parts of the application if needed and they can also be modified without affecting the main page's code. This also allows "editTest.tsx"'s code to focus on the core functionalities and user interactions, improving the readability.

6.4.3 Find Testers

The 'Find Testers' page is one of Rocketest's main pages: not only it is the one that represents our opportunity – what distinguishes us from other applications on the market: the fact that we can search for testers on different platforms without leaving our application –, but also because it is the one containing the API designed to search for testers, which we will review in the next section.

This page has undergone changes compared to its initial design, since the first one did not include any way of seeing what were the criteria defined in the test created to search for participants and, after having done the search for them, it was not known which criteria these were matching (on their card components). We thought it was an important decision to change the design based on this aspect, as it makes the application more usable and makes it simpler for the UX Researcher to decide which testers to contact, by better analysing their criteria. The two images down below show the before and after changes to the page design.

The input for sending messages to participants, in turn, does not actually send any messages since Rocketest's own messaging system is not functional and, in the case of these being from an external platform, we do not have access to real e-mails (in the next section we will explain why the users found are fictitious). Instead, this input allows the distinction between contacted and non-contacted testers, something that is also visible on their card component and allows the UX Researcher not to send the same message to the same people, if they have already been contacted.

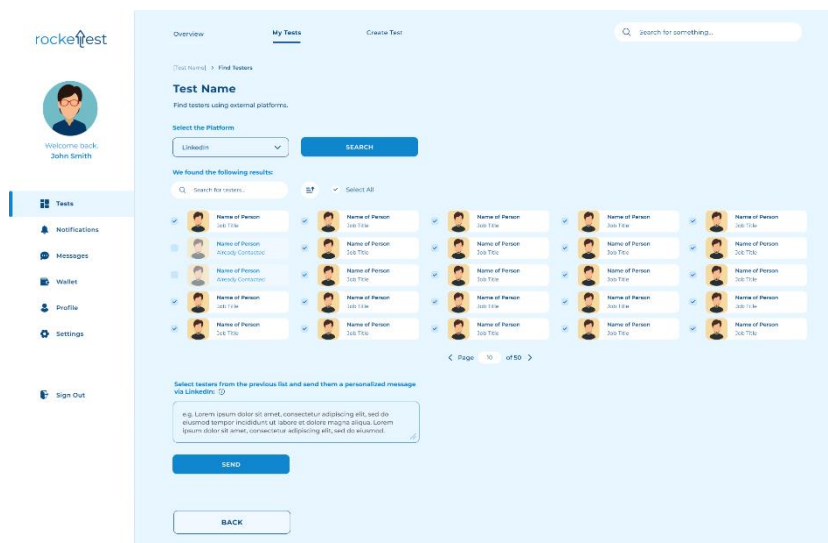


Figure 30. Figma's Final UI for 'Find Testers' Page

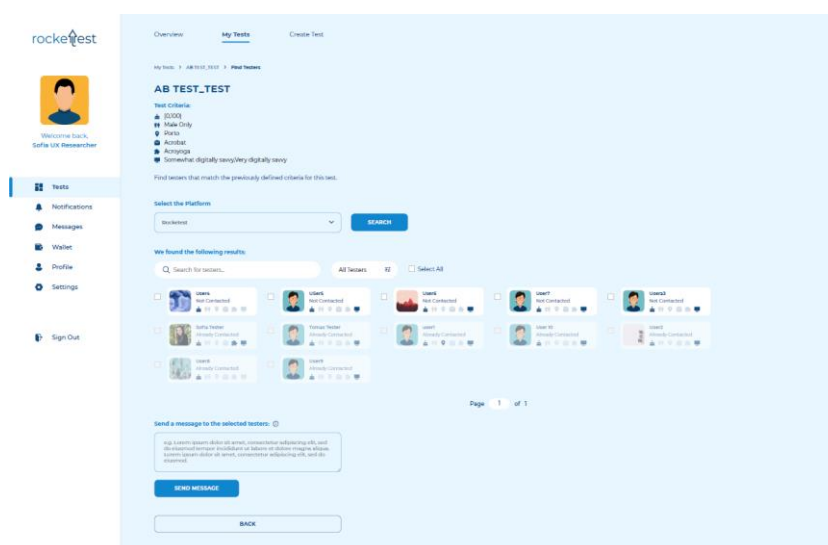


Figure 31. Final Design for 'Find Testers' Page

6.4.4 Testers Search API

6.4.4.1 Limitations

In section 2.3.3 of this dissertation, two possible APIs were suggested so that users from various niche markets could be found on the various popular platforms currently used, such as Facebook (Graph API) or LinkedIn (People API).

At this stage, the functionalities that these APIs offered were studied, realizing that they would be suitable for our project - the most appropriate being the LinkedIn one, since Meta's

(Facebook) contained more restrictions on search parameters, which would end up conditioning the search for more specific niches. However, it was also mentioned that in order to use the LinkedIn API, we would need to be a "LinkedIn Partner" and obtain their approval to use their API. This happens due to GDPR (General Data Protection Regulation), which is understandable, as we are requesting access to their platform's users' information. Since this process turned out to be expensive and time-consuming, it was discarded in favor of the available implementation time and resources.

With the API provided by Meta, it would be necessary to create a Facebook account and register as a "Facebook Developer" to access the Graph API. Despite being a relatively simple process, its API still involves a long learning curve, something that is not possible in a project of such short duration for the development phase as this one. In addition, we had already reviewed on the state of the art chapter that this API would not be the most suitable for this particular project, since it contained some restrictions in the parameters for user searching. Nonetheless, it could be an interesting option to be explored as future work, which may arise the feasibility of integrating it with our application and having access to Facebook users.

Given these limitations, it was considered a better option, given the time and cost factors, to resort to the plan B already mentioned in the solution planning chapter: the development of a Mock API. It is important to note that **the emergence of these limitations does not mean that it is impossible to use APIs from external platforms to search for testers**. On the contrary, it indicates that more time would be needed in the search for appropriate APIs, studying them in detail and, eventually, even requesting their use to the platforms themselves - as is the case with LinkedIn - just bearing in mind that the process may have costs. Furthermore, it would also take time to test whether it is really possible to find users from various niches, and only then it would be possible to confirm the hypothesis of this dissertation.

6.4.4.2 Mock API Development

The word "Mock" in Mock API indicates that it will be an imitation of an API that, in our case, intends to demonstrate how the search for testers would be on a platform of the user's choice:

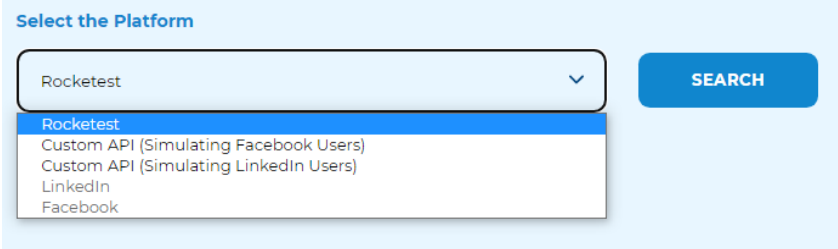
The image shows a web interface element titled "Select the Platform". It features a dropdown menu with a light blue background. The currently selected option is "Rocketest", which is highlighted in a darker blue. Below it, the dropdown menu is open, showing five options: "Rocketest", "Custom API (Simulating Facebook Users)", "Custom API (Simulating LinkedIn Users)", "LinkedIn", and "Facebook". To the right of the dropdown menu is a blue button with the word "SEARCH" in white capital letters.

Figure 32. Select the Platform Input

The user can search for testers inside Rocketest (meaning that the users retrieved will already have a registered account in the application) or using one of the "Custom API" options: one of them simulating Facebook users and the other simulating LinkedIn ones. Notice that the options "LinkedIn" and "Facebook" are disabled. In this case, we will not resort to any external platform's database of users but to two databases of fictitious users – one for LinkedIn and one

for Facebook (created only for demonstrative purposes) –, as a proof of concept of what was supposed to happen if we were to use these external platforms to find users.

Why are we separating Facebook from LinkedIn users? On Rocketest, when signing up as a tester, users can enter the information present on the left column of the table below. Most of that information can be found on Facebook, while some of it can be accessed on LinkedIn, that is, if the users upload it to those platforms when creating/editing their profiles. This is essentially why we are giving the user two different options, we let him/her decide what type of information he/she wants to focus his/her search on and hopefully finding users from more niches, using more than one external platform to search for users.

Tester Information (Rocketest)	Can be found on Facebook?	Can be found on LinkedIn?
Name	Yes	Yes
Profile Picture	Yes	Yes
Age	Yes	No
Gender	Yes	No
Location	Yes	Yes
Career	Yes	Yes
Hobbies	Yes	No
Digital Savviness	No	No

Table 16. Testers Information

When creating the fictional users for these two platforms, we created two different JSON files with 300 examples of users in each of them. Since we wanted to make it as real as possible, not all users have available information for all the fields, some of them appear as “null”, meaning that the user did not upload that information into his/her account.

```
{
  "id": 10,
  "profilePhoto":
  "https://robohash.org/omnisquiadeleniti.png?size=50x50&set=set1",
  "name": "Lukas Iddy",
  "age": 57,
  "gender": null,
  "location": "Philippines",
  "career": null,
  "hobbies": "Storytelling"
}
```

Code Snippet 8. Example of a fictional Facebook User

```
{
  "id": 7,
  "profilePhoto":
  "https://robohash.org/estautut.png?size=50x50&set=set1",
  "name": "Henrieta Gamwell",
  "location": null,
  "career": "Financial Advisor"
}
```

Code Snippet 9. Example of a fictional LinkedIn User

Inside the folder “externalApi”, we have our “index.js” file that sets up a simple server using Express (Express.js, 2023), which will expose two routes for the simulation of both Facebook and LinkedIn’s APIs (in our case, by performing “GET” requests to get users from the two JSON files created). The fake user data received is then filtered by the user’s provided test criteria (left column of Table 17).

To sum up, an example of a request and response using this Mock API developed would be:

```
"GET  
/api/simulatingFacebook/getUsers?ageRange=20,30&gender=Female&location=New%  
20York&career=Engineer&hobbies=Reading,Sports"
```

Code Snippet 10. Example Request using the Mock API

```
[ {  
  "id": 1,  
  "name": "Alice",  
  "age": 25,  
  "gender": "Female",  
  "location": "New York",  
  "career": "Engineer",  
  "hobbies": ["Reading", "Sports"]  
},  
{  
  "id": 2,  
  "name": "Emma",  
  "age": 28,  
  "gender": "Female",  
  "location": "New York",  
  "career": "Software Developer",  
  "hobbies": ["Reading", "Sports", "Travel"]  
},  
{ ... }  
]
```

Code Snippet 11. Example Response using the Mock API

6.5 Final Project

The final project was deployed using the Vercel platform (Vercel, 2023) from the creators of Next.js. The link to access it is present on Attachment H of this dissertation – <https://master-thesis-rocketest.vercel.app>.

Because Vercel would not let us deploy both the Rocketest application and the external API developed, we had to make the latter available by storing it in a Raspberry Pi. By running the API on a separate server, distinct from the Rocketest application, it further simulates the connection that we would have to have by using external platforms’ APIs, providing a more accurate representation of the intended functionality on “Find Testers” page.

7 Experimentation and Evaluation

This chapter describes the three fundamental moments of experimentation and evaluation of the developed project. Firstly, a usability test will be carried out on the first version of the design through a SUS (System Usability Scale); then, the implementation of the requirements stipulated in the implementation phase will be verified through a QEF (Quantitative Evaluation Framework); finally, a feedback questionnaire will be conducted on the already implemented application based on a cognitive walkthrough. The results of each test will be discussed in this chapter, and the changes to be made to the application will be analyzed so that they can meet the assessments obtained.

In this case, participants belonging to the application's target audience - UX Researchers - will not be used, since feedback is only intended on the proof of concept carried out and not on a fully finalized application that should, in fact, be tested with this niche of participants.

7.1 Evaluation Methodology

7.1.1 Before Implementation Stage – System Usability Scale

Having completed the first design sketch of the application, we intend to evaluate the effectiveness of its usability through the interactive prototype designed in Figma. For this, a usability test – SUS, System Usability Scale (Brooke, 1995) – will be performed via Google Forms. As reviewed in the state of the art chapter, SUS was constructed using a Likert scale, using 10 statements that should be evaluated on a scale of 1 to 5 (where 1 is “Strongly Disagree” and 5 is “Strongly Agree”).

Given these ten statements, “there will generally be some where there is a lot of agreement between respondents” (Brooke, 1995). The objective will be to identify and correct usability problems in a phase prior to the development phase, increasingly aligning the design with the client's needs and realizing what they may or may not like. For this, some participants will be selected (a minimum of 10 participants), whose characteristics must go through:

1. Having technological dexterity - must be used to working on a computer regularly;
2. Being over 18 years old - since payment systems are involved in a future version of the application, the user should only sign up in Rocketest if he/she's older than 18;
3. Not having any accessibility deficit (for example, any visual or motor deficiency) - since the application is not foreseen for these cases in its first version.

At the end, an area will be made available where participants can write their feedback in more detail as well as give suggestions that seem useful to them to be included in the project. This

type of qualitative feedback can provide more detailed and specific insights than a quantitative survey alone. The results of this test will then be analyzed, and the necessary corrections will be implemented in the final design of the application, according to the feedback obtained.

This test was chosen due to its easy administration and understanding of the scale by the participants. While it is always ideal to conduct extensive user testing and obtain a large sample size, it is important to make the best use of the resources and time available, which is why we chose it as the only form to evaluate the design of this project, since we are only developing a proof of concept and not a full functional web application. Additionally, the open field input provided at the end of the SUS questionnaire can be a valuable source of feedback and suggestions for improvement at this stage of the project. Nonetheless, if this application were to be developed in its entirety, it is important to keep in mind that the design should be the subject of more usability testing, particularly with testers belonging to the target audience of the product – UX Researchers.

7.1.2 After Implementation Stage – Quantitative Evaluation Framework

At the beginning of the solution's implementation, the requirements to be implemented were defined, since the design of the chosen solution contains more requirements than the ones that need to be developed. Thus, a good way to verify the success of the implementation of the various requirements will be through a Quantitative Evaluation Framework (QEF). This process can be done using Google Sheets, as we will have to deal with calculations.

A QEF (Escudeiro & Bidarra, 2008) is a methodology that aims to evaluate the quality of business processes, methodologies, and web engineering systems using objective metrics. Since methodologies are always evolving, it may be necessary to assess quality as well as determine how it may be improved and how the quality improvement process can be optimized to reduce costs.

The QEF uses a quality benchmark based on dimensions, factors, and requirements:

- **Dimensions:** The components of the software system that are important for judging quality are represented by dimensions. They might cover things like effectiveness, dependability, usability, maintenance, and security, among other things;
- **Factors:** A factor is a single element inside each dimension that contributes to the evaluation of the quality. It stands for more detailed and quantifiable features of a dimension. For instance, elements within the usability dimension may include things like flexibility, simplicity of use, and efficiency;
- **Requirements:** Finally, requirements are criteria or standards that are established for each factor within a dimension. They define the desired level of performance or quality that the system must meet. Requirements can be quantitative, such as specific response times, acceptable error rate, etc., or they can be qualitative, such as usability guidelines, regulatory compliance, etc.

To build the QEF present on Attachment I, we followed the following steps:

1. Define what our requirements are. In Rocketest's case, we defined the requirements based on what needed to be implemented for the chosen solution;
2. Classify the requirements into dimensions and group them into factors, inside the dimensions created. In our case, we are assigning the requirements into three different dimensions: Functionality (with the factors "Security and Support" and "Services and Features"), Usability (with the factors "Effectiveness", "Accessibility and Navigation" and "Help and Recover") and Reliability (with the factors "Maintenance" and "Programming Practices");
3. Assign a relevance ($rwjk$ – requirement weight k in factor j) to each requirement, which varies between 2, 4, 6, 8, or 10. At this stage, we must assign these values based on the requirement's priority: for example, for this project, requirements such as creating a test are fundamental to the operation of the solution, so they should receive a 10, while other requirements such as editing the user's profile or having a loading animation are less relevant to solving the presented problem in this dissertation, therefore, they should have a smaller weight;
4. For each dimension, we create three separate pages that explain each requirement's fulfillment – that is, what we should fill in in the column wfk % (requirement fulfillment k that varies between 0, 25, 50, 75 and 100). For example, in the page "Functionality" are present all the requirements assigned to that dimension and explained how they should be evaluated – 0 being not implemented at all and 100 being fully implemented;
5. Fill in the wfk % column according to each requirement's assessment previously explained;
6. Finally, we apply all the formulas necessary to calculate the system's quality, which are detailed in QEF's original article (Escudeiro & Bidarra, 2008), by this order:

$$W_{ij} = n_{rwjk \text{ of the current factor}} \div n_{rwjk's \text{ of the dimension}}$$

Equation 10. W_{ij} (Factor Weight j in Dim i) $\in [0,1]$ Calculation (For each Factor)

$$Q_j = (1 \div \sum rwjk \text{ of the current factor}) \times (\sum rwjk \text{ of the current factor} \times wfk \text{ of the current factor})$$

Equation 11. Q_j Calculation (For each Factor)

$$Q_i = \sum Q_j \times W_{ij}$$

Equation 12. Q_i Calculation (For each Q_i Dimension)

$$D = \sqrt{\left(1 - \frac{Q_i}{100}\right)^2}$$

Equation 13. D Calculation (For each Q_i Dimension)

$$q = \frac{\log \frac{1 + Qi}{2}}{3 \times \log 2}$$

Equation 14. q Calculation (For each Qi Dimension - Final Quality Measure)

In the end, the quality measure obtained (q) is linear in relation to the percentage of compliance with the requirements. In other words, we will obtain a percentage that will correspond to the success of the implemented requirements.

7.1.3 After Implementation Stage – Feedback Survey

Finally, in order to understand the reaction of potential customers to the developed proof of concept, a feedback survey will be conducted, whose questions will be based on the success obtained in completing tasks in the application.

The participants selected for this test must have the same characteristics defined in the first usability test so that their answers are coherent. This survey can be carried out through Google Forms, as it makes it easier to send people the link to participate in it, just like on the SUS. The questions will vary into three different inputs: yes and no answers, ratings from 1 to 5 (for example, rating on the ease of use of certain functionality), and open-field answer. Throughout the questionnaire, an optional field of feedback will be provided often to improve the feature and understand what went wrong. By the end of it, participants will also be able to leave their general feedback of the application, which will be taken into account for its development at a later stage.

The participants will test both user types to understand how the application works for each one of them. In summary, the purpose of this test will be to evaluate the ease of use of the application, which is a relatively quick way to obtain feedback on its usability. Furthermore, having both the QEF and this Feedback Survey as the ways of testing the application after its implementation, we can obtain richer feedback from a quantitative (QEF) and qualitative (Feedback Survey) perspective, although the latter can also provide us with quantitative feedback.

7.2 Results Analysis

7.2.1 System Usability Scale

With a total of 12 participants with the previously established characteristics (having technological dexterity, being over 18 years old, and not having any accessibility deficit), a SUS survey (which can be reviewed in Attachment D) was carried out on the first version of the application design, testing the two interactive prototypes designed for each user type defined.

The results obtained can be visited on Attachment J of this dissertation. To obtain a percentage of the system's usability as perceived by the participants in this test we followed the following steps:

1. Convert each one of the user ratings (1 being Strongly Disagree and 5 being Strongly Agree) into points:
 - a. For odd numbered questions subtract 1 to the points given (points – 1);
 - b. For even numbered questions subtract 5 before the points (5 – points).
2. Add up the points of each user to obtain their individual scores (Question 1's points + Question 2's points + ... Question 10's points = Total points from user);
3. Multiply the previous obtained score by 2.5 to get the score in a scale from 1-100;
4. Repeat steps 1-3 for all users, and then average all user's scores together to get the final SUS score.

As explained by (Betteridge, n.d.), "Questions that are odd (1, 3, 5, 7, 9) generate a positive response, but questions that are even (2, 4, 6, 8, 10), generate a negative response which is why their scores have to be inverted. All the points added up together could gain a maximum of forty, thus the multiplication by 2.5 to make the scale out of 100."

The image that follows illustrates our results from the 12 participants:

SUS RESULTS										
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
User 1	4	3	4	3	4	1	4	2	3	1
User 2	5	1	5	2	4	2	4	3	4	2
User 3	3	2	4	2	4	2	3	2	3	3
User 4	4	1	5	2	4	1	4	1	5	2
User 5	3	2	4	4	4	2	4	2	4	3
User 6	5	1	5	1	5	1	5	1	5	1
User 7	4	1	5	1	4	1	4	1	3	1
User 8	4	2	4	3	4	2	4	2	4	2
User 9	5	1	3	1	4	1	3	2	4	1
User 10	5	1	5	1	5	1	5	1	5	1
User 11	2	1	5	2	5	1	2	1	2	4
User 12	5	1	5	1	5	1	5	1	5	1

SCALE	
1	Strongly Disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly Agree

CONVERTED RESULTS												
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total	1-100
User 1	3	2	3	2	3	4	3	3	2	4	29	72.5
User 2	4	4	4	3	3	3	3	2	3	3	32	80
User 3	2	3	3	3	3	3	2	3	2	2	26	65
User 4	3	4	4	3	3	4	3	4	4	3	35	87.5
User 5	2	3	3	1	3	3	3	3	3	2	26	65
User 6	4	4	4	4	4	4	4	4	4	4	40	100
User 7	3	4	4	4	3	4	3	4	2	4	35	87.5
User 8	3	3	3	2	3	3	3	3	3	3	29	72.5
User 9	4	4	2	4	3	4	2	3	3	4	33	82.5
User 10	4	4	4	4	4	4	4	4	4	4	40	100
User 11	1	4	4	3	4	4	1	4	1	1	27	67.5
User 12	4	4	4	4	4	4	4	4	4	4	40	100

Average SUS Score: 81.66667

Figure 33. SUS' Average Score

In the first table, the ratings attributed by the users in each question are visible corresponding to the scale shown on the right side. In the second table, the results were converted using the

previously mentioned steps: the ratings were converted into points, the column “Total” shows a sum of the points given by each user, and the column “1-100” shows the multiplication of the sum result by 2.5 to get the score in a scale from 1-100. In the end, the average of the results was obtained (Average SUS Score: ≈ 81.67).

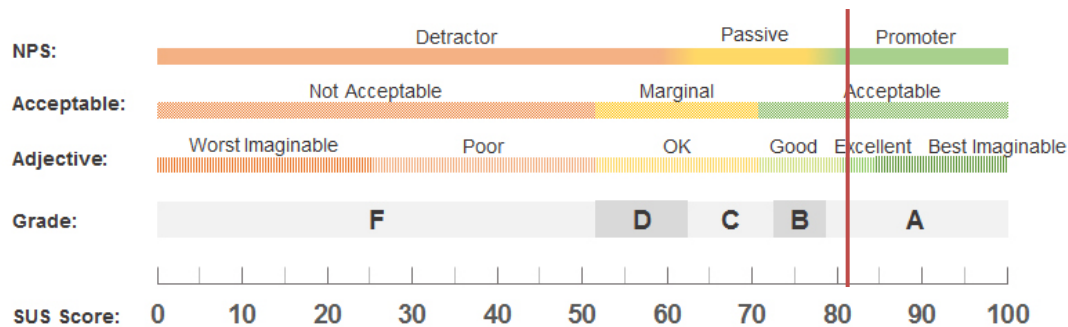


Figure 34. SUS Score Interpretation (Sauro, 2018)

As seen on the previous image (the red vertical bar representing the score obtained), receiving an average SUS Score of approximately **82%** indicates that users found the system to be usable and user-friendly. Based on the findings, it can be said that all 10 of the statements were mostly positive.

By the end of the questionnaire, a space was provided for participants to leave their feedback on design ideas, improvements, bug reports, suggestions, etc. The only feedback received was that, when using the application as a UX Researcher, there was no way to check which testers had already answered the questionnaire. As a result, the final design suffered improvements explained in chapter 5, so that this requirement could be taken into account.

7.2.2 Quantitative Evaluation Framework

Having the QEF (Attachment I) fulfilled according to the success of the requirements implemented, we can observe that most of the requirements were implemented successfully, having “100” in the “wfk %” column. The requirements that did not achieve 100% (meaning that they were not fully implemented) are explained in the following table:

Requirement	Fulfilment [0, 100]	Justification
FSS09	50	The settings page is only shown as a prototype, therefore, the user cannot interact with it and change his/her personal information.
FSF05	75	The API is not from an external platform. Instead, it was created a Mock API to simulate an external application.
FSF09	50	The profile page is only shown as a prototype, therefore, the user cannot interact with it and edit his/her profile or update their information.

UE01	0	A “Back to Top” arrow component is available on the source code but it was not implemented on the final application since it was causing some issues. As it does not appear at all, it had to receive a 0 in its fulfilment, as the user cannot see it neither interact with it.
UE05	50	The Search Bar is indeed visible in some pages of the application but it does not work. Since this requirement was not crucial to the application’s performance, we discarded its implementation to focus our time on improving the other requirements.
RPP03	75	The API is not from an external platform. Instead, it was created a Mock API to simulate an external application, therefore, the testers are fictional.

Table 17. QEF’S Not Fully Implemented Requirements Explanation

The QEF results were then organized in the table shown below. Note that “Factor %” corresponds to the percentage obtained in each factor, the “Dimension %” corresponds to the percentage obtained in each dimension and the “Total %” shows the final result, that is the quality measure obtained.

Dimension	Factor	Factor % (Qj)	Dimension % (Qi)	Total % (q)
Functionality	Security and Support	≈98%	≈96.6%	97%
	Services and Features	≈95.1%		
Usability	Effectiveness	≈88.2%	≈96.1%	
	Accessibility and Navigation	≈100%		
	Help and Recover	≈100%		
Reliability	Maintenance	≈100	≈96.4%	
	Programming Practices	≈93.8%		

Table 18. QEF Results Aggregated

Rocketest obtained a result of **97%** regarding the requirements implemented. From a quantitative point of view, we can conclude that this result signifies a high level of excellence in terms of performance across the defined evaluation metrics. In our case, the dimensions defined had a similar percentage (≈96%), which means that the success of design and development efforts were reflected on the areas of Functionality, Usability and Reliability. We can also conclude, by looking at Table 18, that the requirements that were not fully implemented did not have a high relevance, which means that the developer prioritized the implementation of the requirements that better contributed to the application’s functionality.

7.2.3 Feedback Survey

The Feedback Survey was conducted involving 10 testers who met the predetermined criteria: having technological dexterity, being 18 years or older, and having no accessibility impairments. The survey, which can be found in Attachment K, aimed to gather valuable insights and opinions from the participants regarding the Rocketest's platform for both of its user types.

As previously stated on 7.1.3, the questions on this form were based on the success obtained in completing tasks in the application. By the results obtained, which are visible in Attachment L, we can conclude that **the majority of tasks were accomplished successfully**, receiving most results of 5 on the rating type of questions and "Yes" on questions asking if the user was able to complete a certain task, affirming its completion.

During our first test, we encountered some problems in some of the application's most important tasks, namely:

- The sign up process (the database was denying the use of single quotes, and some fields, like the location one, would not accept special characters);
- Using search filters (we were not paying attention to case matching);
- Creating a test (an error would occur because the query had an unnecessary comma).

This led us to create a new issue on GitHub to solve what our testers would point out that was not working as expected. We also decided to make cosmetic improvements, such as putting a loading animation on the password recovery section while the e-mail is being sent to the user with the recovery link, to better inform the user of what is happening. This is essentially why this test is the most negative one out of all the tests performed.

After this first test, we continued to ask more people to test our application and the results clearly improved. The feedback obtained was very useful and some of the suggestions received can even be taken into account for future work, when developing this application with all of its functionalities.

Having a positive feedback on Rocketest, we can confidently conclude that this project has demonstrated promising outcomes and holds significant potential for further advancement. It is important to note, however, that more participants should be used to test the application in the future, especially focusing on those belonging to the target audience, as well as more usability tests that focus on different areas (such as content or information architecture), just so the application can improve its usability and better satisfy its users.

8 Conclusions

At an early stage of the project, an initial context about the area of UX and usability testing was presented, as well as the problem to be solved – the difficulty of finding participants for the tests – and the project objectives, which involve the implementation of a usability testing management application whose main focus is to solve the presented problem.

Studying the state of the art revealed more in-depth concepts related to UX and, more specifically, to usability tests, in addition to applications for managing them that are currently on the market, and possible technologies for the development of this project. Then, three possible approaches to solving the problem were demonstrated, and, in the value analysis chapter, the one that best corresponded to the parameters of time, adequation, and complexity stipulated for this project was chosen.

Accordingly, the first version of the design was sketched, and a usability test – System Usability Scale – was conducted so that any potential design problems could be identified and corrected before moving on to the final version. After the final version of the design was created, along with its visual identity, we implemented the solution, describing the technology chosen, the implementation process, and the code architecture and documentation. This phase was essentially the most extensive one in our project, which made it the most challenging one, as we will review below. Finally, we tested the solution in two more ways: through a Quantitative Evaluation Framework and a Feedback Survey.

In this chapter, in-depth conclusions will be drawn about the project, namely: the difficulties encountered; achieved objectives, which will also focus on providing a justification for our hypothesis; and, finally, the work to be developed in the future.

8.1 Challenges and Limitations

During the initial phase of the development process, I faced the challenge of familiarizing myself with the essential concepts of Next.js, as my prior knowledge of React.js was limited. To overcome this obstacle, I dedicated significant time to studying tutorials and exploring the documentation. Despite the learning curve, I found Next.js to be an excellent technology choice for the project, as its flexibility allowed me to implement all the desired features seamlessly.

On the backend development front, I encountered limitations due to my lack of confidence and limited experience in this area. As a result, the initial timeline created (the Gantt chart presented in section 1.5), had to be revised. Originally, I had projected the completion of backend development by the end of May. However, the process turned out to be more complex than expected and some queries were very extensive so I had to readjust the timeline, extending the completion deadline to the first two weeks of June. This adjustment ensured that

the backend development phase received the attention and effort it required for a successful outcome, even if it meant having less time to focus on other tasks that were less important and, therefore, not fully developed, as reviewed in section 7.2.2.

Among the various pages of the application, the "Create Test" and "Find Testers" pages were the most important ones, particularly the latter, as it directly connected to the hypothesis of this dissertation. Recognizing their importance, I realized the need to refine the design of the "Find Testers" page to better align with the users' needs and expectations. The complexity of these two pages made their development process particularly challenging, as highlighted in section 6.4. In addition to this, the development of the external API to search for testers, despite it not being a complicated development process, meant that we would not be using "real users" but rather fictitious ones, which interfered with the proof of the hypothesis of this project, as we will verify on the next section.

Overall, despite the limitations and adjustments made along the way, the development journey proved to be a valuable learning experience. Overcoming the challenges of learning a new technology and managing backend complexities allowed me to enhance my skills and gain a deeper understanding of the intricacies involved in building a robust application, despite Rocketest not being a fully developed one.

8.2 Achieved Objectives

On section 1.4, we stated the hypothesis of research in this project: to confirm that using other platforms' APIs to recruit testers for usability tests is an efficient way of finding testers from a wide variety of market niches. Although at first this seems obvious to confirm, we cannot fully prove this hypothesis based on what we developed because we are not using real platforms and real users, due to the limitations of using LinkedIn and Facebook's APIs, as we verified on section 6.4.4. Instead, we are using fake users with invented parameters, which makes it difficult to prove our hypothesis, as we cannot affirm that testers from all niches would be found by using external platforms.

However, we must also indicate that we put our efforts into imitating the data retrieved from the two platforms we had in mind – Facebook and LinkedIn – in the realest way possible, just so the results did not seem fake. As it was also stated before in 6.4.4, we could confirm this hypothesis if we had more time to learn how to use the suggested APIs or even study more appropriate ones. Only then, it would be possible to test whether it is possible to find users from various niches.

Despite our hypothesis remaining unanswered, we still achieved various objectives throughout the development of this project:

1. **Successful Implementation:** Firstly, we can confidently state that the most important requirements for this solution were all implemented and, therefore, the project

achieved its desired outcome, which can also be proved by the results obtained in the experimentation and evaluation phase:

- a. The System Usability Scale conducted on the design phase showed that users found the system usable at the time it was tested, receiving a score of approximately 82%. At this stage, we were also offered useful feedback to improve the final design;
 - b. The Quantitative Evaluation Framework showed that the system had its most important requirements implemented successfully, receiving a final result of 97% on this assessment;
 - c. Finally, the Feedback Survey demonstrated that users could easily complete tasks within our application and gave them a good rating. Moreover, it showed us that this project holds significant potential for further advancement, given the suggestions received for future work.
2. **Core Functionality Development:** One of the primary objectives of this project was to implement the core functionality of finding testers through an external API. Despite not using external platforms' APIs, we still developed our own external API, which means that users can understand how the search for testers would work on an ideal scenario;
3. **Robustness and User-Centric Design:** Bearing in mind that Rocketest is a proof of concept and not a fully developed application, we can still affirm that this platform has been successfully implemented. Moreover, we continuously improved the design based on the user's needs, although user testing should be an ongoing practice if the project was to be developed in its entirety. Users can perceive how the application would look like if it were to be fully functional, also allowing them to distinguish which features are currently working and which are not. This happened because we paid special attention to the application's design as a whole, making sure the users would realise its full potential;
4. **Time-Management and Organization Skills:** Given the fact that this project was not created in association with any companies, it is a huge achievement having it completed on time and with success on the implementation's requirements. This was due to a good organization and project planning, always having the best-intended outcome in mind;
5. **Possibility of including it on any platform:** Finally, we must note that this application was designed as a whole (that is, it works independently), but, if we focus especially on the "Find Testers" page, we realise that its features could easily be included in any existent application of usability testing on the market. For this to happen, we would only have to introduce the existence of criteria for both the tests and the testers: the testers' criteria must match with the test's criteria, which is what happens on Rocketest. This could be a potential idea for investing in this project, as it could be sold to other companies to enhance their own platforms of usability testing.

8.3 Future Work

Some areas of expansion that can further enhance the capabilities and impact of Rocketest are:

1. **Development of the remaining features and extras:** It is crucial to focus on completing the remaining features of the application, making sure to update the database and the design according to the users' needs (also considering the feedback received on the Feedback Survey). Additionally, considering the evolving landscape of UX research, it would be beneficial to explore opportunities for integrating the platform with other widely-used tools and platforms employed by UX researchers, as previously mentioned on 3.1. This integration could enhance the platform's capabilities and provide users with a more comprehensive and streamlined research experience, as well as staying aligned with advancements in the field of UX research, providing long-term value;
2. **Design and implementation for more resolutions (responsiveness):** To make sure this application could be used in any device, its design should be updated, turning Rocketest responsive to more screen resolutions;
3. **Study and implementation of external platforms' APIs:** To further validate the hypothesis stated in this dissertation and expand the platform's reach, it is important to study and incorporate APIs from external platforms into our solution. This integration would facilitate seamless data exchange, enable users to access a broader pool of potential participants, and enhance the platform's overall effectiveness and usability;
4. **Conduction of comprehensive usability testing and gather richer feedback:** While the tests performed during this dissertation had very few participants due to time and resource constraints, future work should focus on conducting more robust usability testing. This includes employing a combination of qualitative and quantitative methods to obtain comprehensive feedback from users, identify areas for improvement, and refine the platform's design and functionality accordingly, using more testers for this effect. This will ensure that the final design and implementation of the solution meets the needs and expectations of the target users.

By addressing these aspects, the platform can evolve into a comprehensive and powerful tool that meets the evolving needs of UX researchers, while facilitating efficient and impactful research practices.

References

Angular.js, 2023. *Angular.js*. [Online]

Available at: <https://angularjs.org/>

[Accessed 5 January 2023].

Axios, 2023. *Axios / Getting Started*. [Online]

Available at: <https://axios-http.com/docs/intro>

[Accessed 8 May 2023].

B2B International, 2023. *What is the Value Proposition Canvas?*. [Online]

Available at: <https://www.b2binternational.com/research/methods/faq/what-is-the-value-proposition-canvas/>

[Accessed 23 January 2023].

Babich, N., 2019. *Card Sorting Best Practices for UX*. [Online]

Available at: <https://xd.adobe.com/ideas/process/information-architecture/card-sorting-best-practices/>

[Accessed 15 December 2022].

Babich, N., 2019. *Eye Tracking and Usability: How Does it Work?*. [Online]

Available at: <https://xd.adobe.com/ideas/process/user-research/eye-tracking-and-usability/>

[Accessed 14 December 2022].

Bathia, A., 2016. *Why, How, and When to Utilize Usability Testing*. [Online]

Available at: <https://www.shopify.com/partners/blog/108023878-why-how-and-when-to-utilize-usability-testing>

[Accessed 15 December 2022].

Berg, P., 2017. *5 User Testing Challenges (and How We Overcame Them)*. [Online]

Available at: <https://www.dropsources.com/blog/5-user-testing-challenges/>

[Accessed 5 January 2023].

Betteridge, K., n.d. *What Every UXC Client Should Know About SUS Scores*. [Online]

Available at: <https://www.bentley.edu/centers/user-experience-center/what-every-client-should-know-about-sus-scores>

[Accessed 15 March 2023].

Blackmon, M., Polson, P., Kitajama, M. & Lewis, C., 2002. *Cognitive walkthrough for the Web*. [Online]

Available at:

https://www.researchgate.net/publication/221514534_Cognitive_walkthrough_for_the_Web

[Accessed 14 December 2022].

Bodiu, R., 2017. *Quantitative vs. Qualitative Usability Testing*. [Online]

Available at: <https://www.nngroup.com/articles/quant-vs-qual/>

[Accessed 14 December 2022].

Brooke, J., 1995. *SUS: A quick and dirty usability scale*. [Online]

Available at:

https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale

[Accessed 28 April 2023].

Cornell, J., 2022. *User Experience Surveys: 10 Questions to Ask*. [Online]

Available at: https://www.proprofssurvey.com/blog/user-experience-survey/#What_Are_User_Experience_Surveys

[Accessed 14 December 2022].

Crady, A., 2018. *A/B Testing your Designs*. [Online]

Available at: <https://uxdesign.cc/a-b-testing-your-designs-b648e413b124>

[Accessed 14 December 2022].

Croeser, E., 2022. *How to Keep Up with Trends & Technologies: A Simple (but Powerful) Guide*. [Online]

Available at: <https://www.itonics-innovation.com/blog/stay-up-to-date-with-trends-and-emerging-technologies-guide>

[Accessed 15 March 2023].

ElephantSQL, 2023. *ElephantSQL - PostgreSQL as a Service*. [Online]

Available at: <https://www.elephantsql.com/>

[Accessed 6 May 2023].

Eschberger-Friedl, T., 2018. *The fuzzy front end of the innovation process*. [Online]

Available at: <https://www.lead-innovation.com/en/insights/english-blog/fuzzy-front-end>

[Accessed 18 January 2023].

Escudeiro, P. M. & Bidarra, J., 2008. *Quantitative Evaluation Framework (QEF)*. [Online]

Available at:

https://www.researchgate.net/publication/257579051_Quantitative_Evaluation_Framework_QEF

[Accessed 9 May 2023].

Evans, L. et al., n.d. *Usability testing*. [Online]

Available at: <https://about.gitlab.com/handbook/product/ux/ux-research/usability-testing/>

[Accessed 6 December 2022].

Express.js, 2023. *Express Framework*. [Online]

Available at: <https://expressjs.com/>

[Accessed 15 June 2023].

Firebase, 2023. *Firebase - Make your app the best it can be*. [Online]

Available at: <https://firebase.google.com/>

[Accessed 5 January 2023].

Francis, N., n.d. *Every business should understand the importance of user testing*. [Online]

Available at: <https://marvelapp.com/blog/importance-user-testing/>

[Accessed 5 December 2022].

Goh, S., 2021. *The Ultimate Guide To LinkedIn API: People Profile API (With Python Example)*. [Online]

Available at: https://nubela.co/blog/ultimate-guide-to-linkedin-api_people-profile-api_with-python-examples/

[Accessed 7 January 2023].

Harley, A., 2015. *Personas Make Users Memorable for Product Team Members*. [Online]

Available at: <https://www.nngroup.com/articles/persona/>

[Accessed 16 December 2022].

Hinderks, A., Schrepp, M. & Thomaschewski, J., 2018. *User Experience Questionnaire*. [Online]

Available at: <https://www.ueq-online.org/>

[Accessed 16 December 2022].

Interactive Design Foundation, n.d. *How to Conduct a Cognitive Walkthrough*. [Online]

Available at: <https://www.interaction-design.org/literature/article/how-to-conduct-a-cognitive-walkthrough>

[Accessed 14 December 2022].

Interactive Design Foundation, n.d. *User Experience (UX) Design*. [Online]

Available at: <https://www.interaction-design.org/literature/topics/ux-design>

[Accessed 5 December 2022].

InterviewBit, 2022. *Top Front End Technologies You Must Know [2022]*. [Online]

Available at: <https://www.interviewbit.com/blog/front-end-technologies/>

[Accessed 5 January 2023].

ISO 9241-210, 2019. *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. [Online]

Available at: <https://www.iso.org/standard/77520.html>

[Accessed 5 December 2022].

Kalpokas, N., n.d. *How to perform automatic focus group coding using ATLAS.ti Desktop*. [Online]

Available at: <https://atlasti.com/research-hub/how-to-perform-automatic-focus-group-coding-using-atlas-ti>

[Accessed 14 December 2022].

- Kathuria, H., 2022. *The Most Popular Databases for 2022*. [Online]
Available at: <https://learnsql.com/blog/most-popular-databases-2022/>
[Accessed 5 January 2023].
- Kessler, J., 2019. *5 Things You Need to Do Before You Start Software Development*. [Online]
Available at: <https://neoteric.eu/blog/5-things-you-need-to-do-before-you-start-software-development/>
[Accessed 15 March 2023].
- Koen, P. A. et al., 2002. *Fuzzy Front End: Effective Methods, Tools, and Techniques*. [Online]
Available at: https://media.wiley.com/product_data/excerpt/13/04712061/0471206113.pdf
[Accessed 18 January 2023].
- Lapierre, J., 2000. Customer-perceived value in industrial contexts. *Journal of Business & Industrial Marketing*, pp. 122 - 145.
- Lewis, J., 2022. *Measuring Tech Savviness*. [Online]
Available at: <https://measuringu.com/measuring-tech-savviness/>
[Accessed 3 January 2023].
- Ly, A., 2022. *UserTesting Review – How Much Money Can You Really Make?*. [Online]
Available at: <https://thesmartwallet.com/usertesting-review-how-much-money-can-you-really-make/?articleid=44617>
[Accessed 17 January 2023].
- Marie, J., 2021. *Using Personas to Create a Great User Experience*. [Online]
Available at: <https://slickplan.com/blog/using-personas-and-great-user-experience>
[Accessed 16 December 2022].
- Mathew, A., 2022. *The difference between UX designers and UX researchers*. [Online]
Available at: <https://www.testingtime.com/en/blog/difference-between-ux-designer-and-ux-researcher/>
[Accessed 6 December 2022].
- Maze, 2022. *What is UX Research: The Ultimate Guide for UX Researchers*. [Online]
Available at: <https://maze.co/guides/ux-research/>
[Accessed 14 December 2022].
- McSpadden, K., 2015. *You Now Have a Shorter Attention Span Than a Goldfish*. [Online]
Available at: <https://time.com/3858309/attention-spans-goldfish/>
[Accessed 5 December 2022].
- Meta, 2023. *Graph API*. [Online]
Available at: <https://developers.facebook.com/docs/graph-api/overview>
[Accessed 7 January 2023].

Meta, 2023. *Pages API*. [Online]

Available at: https://developers.facebook.com/docs/pages/searching/?locale=pt_BR
[Accessed 7 January 2023].

Microsoft, 2023. *People APIs*. [Online]

Available at: <https://learn.microsoft.com/en-us/linkedin/shared/integrations/people/overview>
[Accessed 7 January 2023].

MongoDB, 2023. *MongoDB Atlas*. [Online]

Available at: <https://www.mongodb.com/>
[Accessed 5 January 2023].

Moran, K., 2019. *Usability Testing 101*. [Online]

Available at: <https://www.nngroup.com/articles/usability-testing-101/>
[Accessed 6 December 2022].

Mortensen, D., 2020. *The Basics of Recruiting Participants for User Research*. [Online]

Available at: <https://www.interaction-design.org/literature/article/the-basics-of-recruiting-users-for-usability-testing>
[Accessed 17 December 2022].

MySQL, 2023. *MySQL*. [Online]

Available at: <https://www.mysql.com/>
[Accessed 5 January 2023].

Next.js, 2023. *Next.js - The React Framework for the Web*. [Online]

Available at: <https://nextjs.org/>
[Accessed 6 May 2023].

Nicola, S., Ferreira, E. P. & Ferreira, J. J. P., 2012. *Conceptual Model for Decomposing the Value for the Customer*. [Online]

Available at:
https://recipp.ipp.pt/bitstream/10400.22/1248/3/COM_SusanaNicola_2012_GECAD.pdf
[Accessed 22 January 2023].

Nielsen, J., 1997. *The Use and Misuse of Focus Groups*. [Online]

Available at: <https://www.nngroup.com/articles/focus-groups/>
[Accessed 14 December 2022].

Nielsen, J., 2012. *How Many Test Users in a Usability Study?*. [Online]

Available at: <https://www.nngroup.com/articles/how-many-test-users/>
[Accessed 16 December 2022].

Nielsen, J., 2020. *10 Usability Heuristics for User Interface Design*. [Online]
Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/>
[Accessed 14 December 2022].

Nilsson, H., Nordström, E. & Öhman, K., 2016. *Decision Support for Participatory Forest Planning Using AHP and TOPSIS*. [Online]
Available at:
https://www.researchgate.net/publication/302060963_Decision_Support_for_Participatory_Forest_Planning_Using_AHP_and_TOPSIS
[Accessed 19 January 2023].

Node.js, 2023. *Node.js*. [Online]
Available at: <https://nodejs.org/>
[Accessed 5 January 2023].

npm, 2023. *pg-pool*. [Online]
Available at: <https://www.npmjs.com/package/pg-pool>
[Accessed 8 May 2023].

Optimal Workshop Ltd., 2022. *Learn about tree testing*. [Online]
Available at: <https://blog.optimalworkshop.com/tree-testing/>
[Accessed 15 December 2022].

Osadchuk, S., 2022. *Top Backend Technologies in 2022: choose the right one for your solution*. [Online]
Available at: <https://doit.software/blog/backend-technologies#screen1>
[Accessed 5 January 2023].

pgAdmin, 2023. *pgAdmin*. [Online]
Available at: <https://www.pgadmin.org/>
[Accessed 6 May 2023].

pgAdmin, 2023. *pgAdmin - Features*. [Online]
Available at: <https://www.pgadmin.org/features/>
[Accessed 6 May 2023].

Pollfish, n.d. *How to use screening questions to reach your target audience*. [Online]
Available at: <https://resources.pollfish.com/market-research/using-screening-questions-to-find-the-right-audience/>
[Accessed 17 December 2022].

PostgreSQL, 2023. *PostgreSQL About*. [Online]
Available at: <https://www.postgresql.org/about/>
[Accessed 6 May 2023].

PostgreSQL, 2023. *PostgreSQL: The World's Most Advanced Open Source Relational Database*. [Online]

Available at: <https://www.postgresql.org/>
[Accessed 6 May 2023].

ProductPlan, 2023. *How to Build a Product Roadmap Based on a Business Model Canvas*. [Online]
Available at: <https://www.productplan.com/learn/business-model-canvas/>
[Accessed 23 January 2023].

Python, 2023. *Python*. [Online]
Available at: <https://www.python.org/>
[Accessed 5 January 2023].

Rauschenberger, M. et al., 2013. *Efficient Measurement of the User Experience of Interactive Products. How to use the User Experience Questionnaire (UEQ)..* [Online]
Available at:
https://www.researchgate.net/publication/235850976_Efficient_Measurement_of_the_User_Experience_of_Interactive_Products_How_to_use_the_User_Experience_Questionnaire_UEQ_Example_Spanish_Language_Version
[Accessed 16 December 2022].

React.js, 2023. *React - The library for web and native user interfaces*. [Online]
Available at: <https://react.dev/>
[Accessed 5 January 2023].

Rich, N., 2000. *Value Analysis, Value Engineering*. [Online]
Available at: https://www.urenio.org/tools/en/value_analysis.pdf
[Accessed 18 January 2023].

Rog, G., 2019. *UX Fundamentals: User Testing*. [Online]
Available at: <https://www.udemy.com/course/ux-fundamentals-user-testing/>
[Accessed 6 December 2022].

Rosala, M., 2021. *How Many Participants for a UX Interview?*. [Online]
Available at: <https://www.nngroup.com/articles/interview-sample-size/>
[Accessed 16 December 2022].

Ruby, 2023. *Ruby - A Programmer's Best Friend*. [Online]
Available at: <https://www.ruby-lang.org/>
[Accessed 5 January 2023].

Saaty, T., 1980. *The Analytic Hierarchy Process*. New York: McGraw-Hill.

Santana, F., 2017. *Card Sorting: a arquitetura de informação construída pelos usuários*. [Online]
Available at: <https://coletivoux.com/card-sorting-a-arquitetura-de-informa%C3%A7%C3%A3o->

constru%C3%ADda-pelos-usu%C3%A1rios-54b9f892e38e

[Accessed 15 December 2022].

Sauro, J., 2013. *7 Ways to Find Users for Usability Testing*. [Online]

Available at: <https://measuringu.com/finding-users/>

[Accessed 3 November 2022].

Sauro, J., 2018. *5 Ways to Interpret a SUS Score*. [Online]

Available at: <https://measuringu.com/interpret-sus-score/>

[Accessed 15 March 2023].

Stack Overflow, 2021. *2021 Developer Survey*. [Online]

Available at: <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-database>

[Accessed 5 January 2023].

Statista, 2023. *Most popular social networks worldwide as of January 2023, ranked by number of monthly active users*. [Online]

Available at: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>

[Accessed 27 June 2023].

Studio by UXPin, n.d. *A/B Testing in UX Design: When and Why It's Worth It*. [Online]

Available at: <https://www.uxpin.com/studio/blog/ab-testing-in-ux-design-when-and-why/>

[Accessed 14 December 2022].

TeaCup Lab, 2019. *Choosing the right participants for your user research*. [Online]

Available at: <https://www.teacuplab.com/blog/choosing-right-participants-for-user-research/>

[Accessed 5 December 2022].

TP&P Technology, 2021. *Software Development: Choosing the Right Technology for Your Project*. [Online]

Available at: <https://www.tpptechnology.com/blog/software-development-choosing-the-right-technology-for-your-project/>

[Accessed 15 March 2023].

Ulanovsky, J., Matas, S., Peral, J. & Bailly, J., n.d. *Montserrat*. [Online]

Available at: <https://fonts.google.com/specimen/Montserrat>

[Accessed 22 March 2023].

usability.gov, n.d. *System Usability Scale (SUS)*. [Online]

Available at: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

[Accessed 14 December 2022].

UsabilityHub, 2022. *An introduction to first click testing*. [Online]
Available at: <https://usabilityhub.com/guides/first-click-testing>
[Accessed 14 December 2022].

UsabilityHub, 2022. *Five second tests*. [Online]
Available at: <https://fivesecondtest.com/>
[Accessed 14 December 2022].

Userlytics, 2022. *Change the world one test at a time*. [Online]
Available at: <https://www.userlytics.com/paid-ux-testing/>
[Accessed 17 January 2023].

UserTesting, 2022. *UX testing: how to find the right target audience with screener questions*. [Online]
Available at: <https://www.usertesting.com/blog/UX-testing-screener-questions>
[Accessed 3 November 2022].

UXQB e. V. , 2020. *CPUX-F Curriculum and Glossary*. [Online]
Available at: https://uxqb.org/public/documents/CPUX-F_EN_Curriculum-and-Glossary.pdf
[Accessed 3 November 2022].

Vailshery, L., 2022. *Most used programming languages among developers worldwide as of 2022*. [Online]
Available at: <https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>
[Accessed 6 January 2023].

Vercel, 2023. *Let's build something new*. [Online]
Available at: https://vercel.com/new?utm_medium=default-template&filter=next.js&utm_source=create-next-app&utm_campaign=create-next-app-readme
[Accessed 19 June 2023].

Vercel, 2023. *Next.js Introduction*. [Online]
Available at: <https://nextjs.org/docs>
[Accessed 6 May 2023].

Vue.js, 2023. *Vue.js - The Progressive JavaScript Framework*. [Online]
Available at: <https://vuejs.org/>
[Accessed 5 January 2023].

Vyas, I., 2022. *Advantages of Firebase Mobile App Development*. [Online]
Available at: <https://citrusbug.com/blog/advantages-of-firebase-mobile-app-development>
[Accessed 6 January 2023].

Watson, Z., 2018. *When is the best time to conduct usability testing?*. [Online]

Available at: <https://maze.co/blog/best-time-usability-testing/>

[Accessed 15 December 2022].

Wonder, S., 2020. *Why User-Based Testing Is Important to Successful UX Tests*. [Online]

Available at: <https://www.loop11.com/why-user-based-testing-is-important-to-successful-ux-tests/>

[Accessed 5 December 2022].

Attachments

Attachment A

Design Requirements

https://docs.google.com/document/d/1aTfrgVsS7tbbYN_Y94ZCk0MjC7NMCYm73PwH_30DHJE/edit?usp=sharing

UX Researcher's Requirements

Before Authentication:

- **Sign In:**
 - Forgotten Password (Possibility of recovering it by e-mail).
- **Sign Up:**
 - Choose user type;
 - Name, E-mail and Password fields are mandatory;
 - Optional Fields: Location, Website, Description and Profile Photo in case the UX Researcher is signing up as a company, and also Job Title in case he/she is signing up as a normal user;
- Read the Terms and Conditions, About the App, and Contact Us;

After Authentication:

- **Tests:**
 - Overview:
 - Dashboard containing information like on going tests, latest transactions, number of testers per test, a calendar showing tests' deadlines, payment activity, wallet balance, etc.
 - My Tests:
 - Search for created tests;
 - Filter by tests "In Progress" or "Finished";
 - Filter by test name, number of testers, date or test type;
 - Sort results in ascending or descending order;
 - Show Test Detail (Edit Test, Delete Test, Copy Link, Share Link, View Test Results and Find Testers via a platform of choice, also giving the possibility of filtering the results obtained and sending a personalized message through the platform chosen);
 - Show Test Results, with the possibility of exporting them.
 - Create Test:
 - Choose a template from a variety of usability test templates;
 - Edit the test's fields and see a preview of how the test will look like;
 - Define test details (such as participant's criteria – age range, gender, location, career, personal interests and helpful attributes, digital savviness – and test's information – deadline, incentive given and privacy).
- **Notifications:**
 - Show notifications about the tests, wallet transactions and messages received.
- **Messages:**
 - Possibility of messaging a user or a group of users and make a call, video call, send audio messages, send emojis and send attachments.
- **Wallet:**
 - Change currency;

- Access my cards (possibility of making a deposit in case there's a lack of funds, and edit the payment method – i.e. credit card, PayPal, etc.);
 - Check the current balance and payment activity of the last six months;
 - Check transaction history, showing which users received payments from this account.
- **Profile:**
 - Personal information showcase, as well as social media;
 - Number of tests created, number of followers and following;
 - Possibility of following/unfollowing;
 - Show the three latest on going tests;
 - Show badges (i.e. tests completed, total number of participants in tests, number of followers);
 - Show Related Users (i.e. companies with the same test types);
 - Edit Profile:
 - Edit profile photo;
 - Edit personal information used in the sign up process – name/company name, job title, location, website and description;
 - Add/edit social media (Facebook, LinkedIn, Instagram, Twitter).
- **Settings:**
 - Account:
 - Change personal information (e-mail and/or password);
 - Delete account.
 - Accessibility:
 - Change language;
 - Change color mode.
 - About the Application:
 - About Us;
 - Terms and Conditions;
 - Contact Us.
- **Sign Out.**

Tester's Requirements

Before Authentication:

- **Sign In;**
 - Forgotten Password (Possibility of recovering it by e-mail).
- **Sign Up:**
 - Choose user type;
 - Name, E-mail and Password fields are mandatory;
 - Optional Fields: Birth Date, Gender, Location, Career, Personal Interests and Helpful Attributes, Digital Savviness (using the PEW Digital Savviness Classifier), Profile Photo.
- Read the Terms and Conditions, About the App, and Contact Us.

After Authentication:

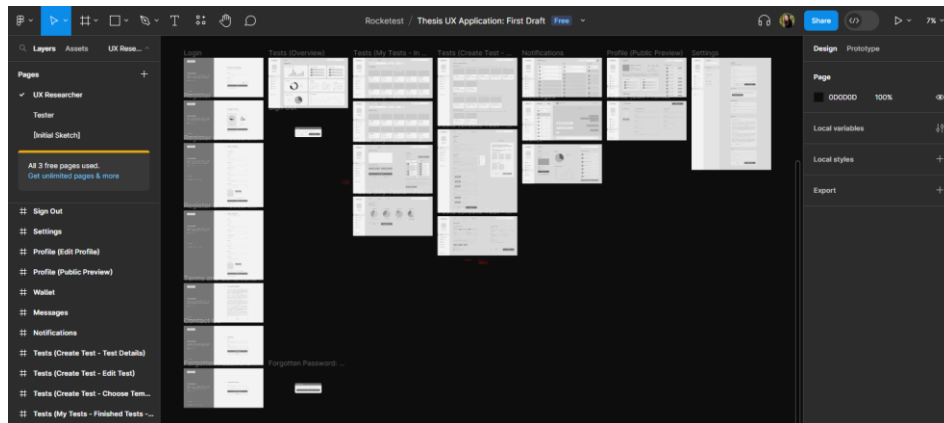
- **Tests:**
 - Overview:
 - Available tests (with the possibility of filtering the results by "all tests" or "matched criteria" – the ones that match with the tester's previously uploaded personal information -, filtering by test name, company name, date or test type, and sorting the results by ascending or descending order) – these are public tests that are available for all users;
 - Information like current wallet balance, latest transactions and answered tests.
 - My Tests:
 - Search for answered tests;
 - Filter by "Paid" or "Payment Pending";
 - Filter by test name, company name, date or test type;
 - Sort results in ascending or descending order;
 - Show Test Detail (what were the answers to that test).
- **Notifications:**
 - Show notifications about the tests, wallet transactions and messages received.
- **Messages:**
 - Possibility of messaging a user or a group of users and make a call, video call, send audio messages, send emojis and send attachments.
- **Wallet:**
 - Change currency;
 - Access my cards (edit the payment method – i.e. credit card, PayPal, etc.);
 - Check the current balance (possibility of transferring the money to personal account) and payment activity of the last six months;
 - Check transaction history, showing which companies paid this user for his/her answered tests.
- **Profile:**
 - Personal information showcase, as well as social media;

- Number of tests answered, number of followers and following;
- Possibility of following/unfollowing and checking/unchecking the "receive notifications for new tests" checkbox;
- Show the three latest answered tests;
- Show badges (i.e. tests answered, total number of companies helped, number of followers);
- Show Related Users (i.e. users with the same attributes);
- Edit Profile:
 - Edit profile photo;
 - Edit personal information used in the sign up process – name, career, birth date, gender, location, personal interests and helpful attributes, and digital savviness;
 - Add/edit social media (Facebook, LinkedIn, Instagram, Twitter).
- **Settings:**
 - Account:
 - Change personal information (e-mail and/or password);
 - Delete account.
 - Accessibility:
 - Change language;
 - Change color mode.
 - About the Application:
 - About Us;
 - Terms and Conditions;
 - Contact Us.
- **Sign Out.**

Attachment B

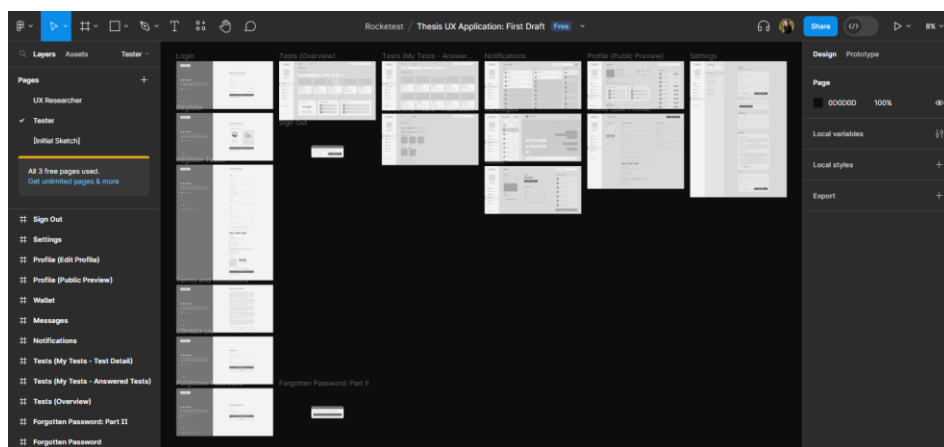
User Interface - First Draft (User Type: UX Researcher)

<https://www.figma.com/file/BI31wNYO9BuQdw2bcXjkXp/Thesis-UX-Application%3A-First-Draft?node-id=415%3A822&t=Lj6TQh4Gx7IsGKEEm-1>



User Interface - First Draft (User Type: Tester)

<https://www.figma.com/file/BI31wNYO9BuQdw2bcXjkXp/Thesis-UX-Application%3A-First-Draft?node-id=555%3A2569&t=Lj6TQh4Gx7IsGKEEm-1>



Attachment C

Interactive Prototype - First Draft (User Type: UX Researcher)

<https://www.figma.com/proto/BI31wNYO9BuQdw2bcXjkXp/Thesis-UX-Application%3A-First-Draft?node-id=415%3A917&scaling=min-zoom&page-id=415%3A822&starting-point-node-id=415%3A917>

Interactive Prototype - First Draft (User Type: Tester)

<https://www.figma.com/proto/BI31wNYO9BuQdw2bcXjkXp/Thesis-UX-Application%3A-First-Draft?node-id=555%3A2575&scaling=min-zoom&page-id=555%3A2569&starting-point-node-id=555%3A2575>

Attachment D

Form – System Usability Scale of the Design's First Version

<https://forms.gle/sgdefiv3zhQkKYJb6>


System Usability Scale · First Version of Application's Design


This form is a part of the Experimentation and Evaluation chapter from Sofia Alexandra Moreira Canelas' Master's Thesis, entitled "Participant Selection Process in a Usability Testing Platform".

This dissertation was written within the scope of the curricular unit Master Thesis (TMDEI) of the second year of the Master's Degree in Informatics Engineering in the Graphic Systems and Multimedia branch held at the Instituto Superior de Engenharia do Porto (ISEP).

Please read the context before moving on to the questionnaire.

sofia.amc3@gmail.com [Mudar de conta](#)

 Não partilhado



Context

An application for the management of Usability Tests was designed as a solution to the problem presented in the dissertation. This application has two main user types:


- the UX Researcher (whose role is to create and manage of Usability Tests);
- and the Tester (whose role is to participate in Usability Testing).

This application is not yet developed, therefore, **you will be asked to test two versions of a design prototype** made in Figma (one version for each user type). Essentially, this means that not all features or clickable items (buttons, icons, etc) are working on the prototype, only the main ones. The design is also on its first version and does not have a visual identity - the colors are black and white and there is no presence of a logotype or application name yet.

By participating in this questionnaire you will allow us to improve the first version of the design to its final version, as well as identify and correct usability problems before the development phase.


Thank you in advance for participating in this survey!

[Seguinte](#) [Limpar formulário](#)

 Nunca envie palavras-passe através dos Google Forms.

System Usability Scale · First Version of Application's Design

sofia.amc3@gmail.com [Mudar de conta](#)

 Não partilhado



* Indica uma pergunta obrigatória

Personal Information

This information will only be used to check if our participants' characteristics match the target audience we are looking for. The questionnaire is completely anonymous.

Are you used to working on a computer regularly? *

☐ Yes

☐ No

Are you 18 years old or over? *

☐ Yes

☐ No

Do you have any accessibility deficit? (i.e. visual or motor impairment) *


☐ Yes

☐ No



System Usability Scale · First Version of Application's Design

sofia.amc3@gmail.com [Mudar de conta](#)

 Não partilhado



* Indica uma pergunta obrigatória

System Usability Scale

PLEASE TEST THE FOLLOWING PROTOTYPES:

User Type: UX Researcher

- <https://www.figma.com/proto/8l3wNYO9BuOdw2bcXjkXq/Thesis-UX-Application%3A-First-Draft?node-id=415%3A917&scaling=min-zoom&page-id=415%3A822&starting-point-node-id=415%3A917>

User Type: Tester

- <https://www.figma.com/proto/8l3wNYO9BuOdw2bcXjkXq/Thesis-UX-Application%3A-First-Draft?node-id=555%3A257&scaling=min-zoom&page-id=555%3A256&starting-point-node-id=555%3A257&>

When testing the prototypes, see if you can easily navigate between pages, and **take notes** of any issues you might have encountered or any suggestion you might have for us, as you will be able to provide us your feedback by the end of the questionnaire.

Testing Tip: You can click anywhere on the prototype that is not a clickable area and it will highlight the parts that are clickable.

After testing the interactive prototypes provided, please answer the following statements.

I think that I would like to use this system frequently. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I found the system unnecessarily complex. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I thought the system was easy to use. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I think that I would need the support of a technical person to be able to use this system. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I found the various functions in this system were well integrated. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I thought there was too much inconsistency in this system. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I would imagine that most people would learn to use this system very quickly. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I found the system very cumbersome to use. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I felt very confident using the system. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

I needed to learn a lot of things before I could get going with this system. *

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neutral
- ☐ Agree
- ☐ Strongly agree

Use this space to provide us with additional feedback on the prototype - i.e. design ideas, improvements, bug report, suggestions, etc.

A sua resposta

[Anterior](#)

[Enviar](#)

[Limpar formulário](#)

Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) - [Termos de Utilização](#) - [Política de privacidade](#)

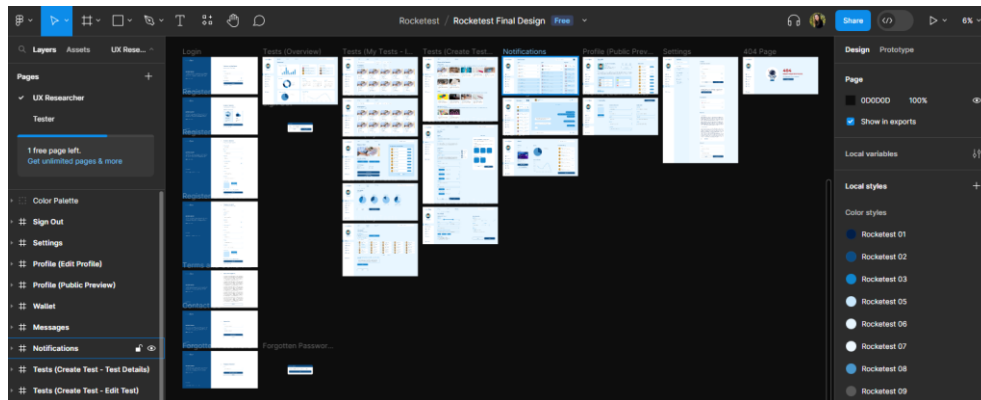
Google Formulários



Attachment E

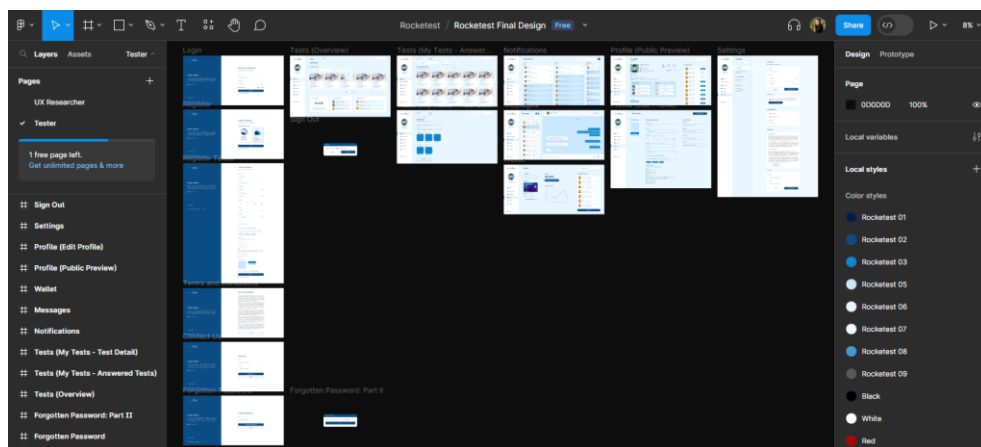
User Interface – Final Version (User Type: UX Researcher)

<https://www.figma.com/file/q7i0sDWO66A6tiP8IEv6sd/Rocketest-Final-Design?node-id=415%3A822&t=FbjuUapGmzvaml8M-1>



User Interface - Final Version (User Type: Tester)

<https://www.figma.com/file/q7i0sDWO66A6tiP8IEv6sd/Rocketest-Final-Design?node-id=555%3A2569&t=FbjuUapGmzvaml8M-1>



Attachment F

Interactive Prototype - Final Version (User Type: UX Researcher)

<https://www.figma.com/proto/q7i0sDWO66A6tiP8IEv6sd/Rocketest-Final-Design?node-id=415-917&scaling=min-zoom&page-id=415%3A822&starting-point-node-id=415%3A917>

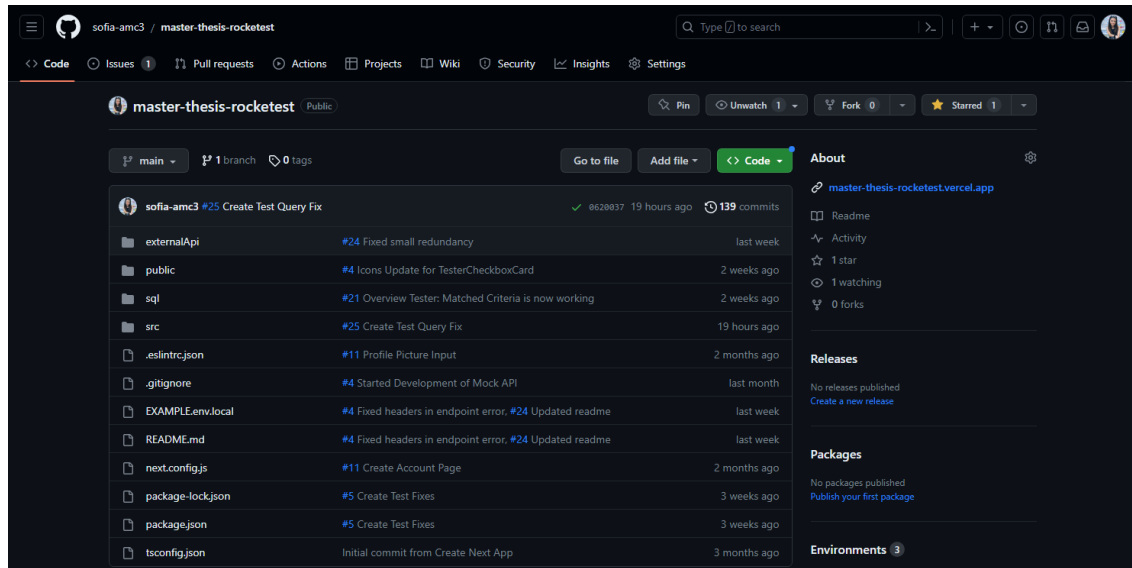
Interactive Prototype - Final Version (User Type: Tester)

<https://www.figma.com/proto/q7i0sDWO66A6tiP8IEv6sd/Rocketest-Final-Design?node-id=555-2575&scaling=min-zoom&page-id=555%3A2569&starting-point-node-id=555%3A2575>

Attachment G

Project's Repository on GitHub

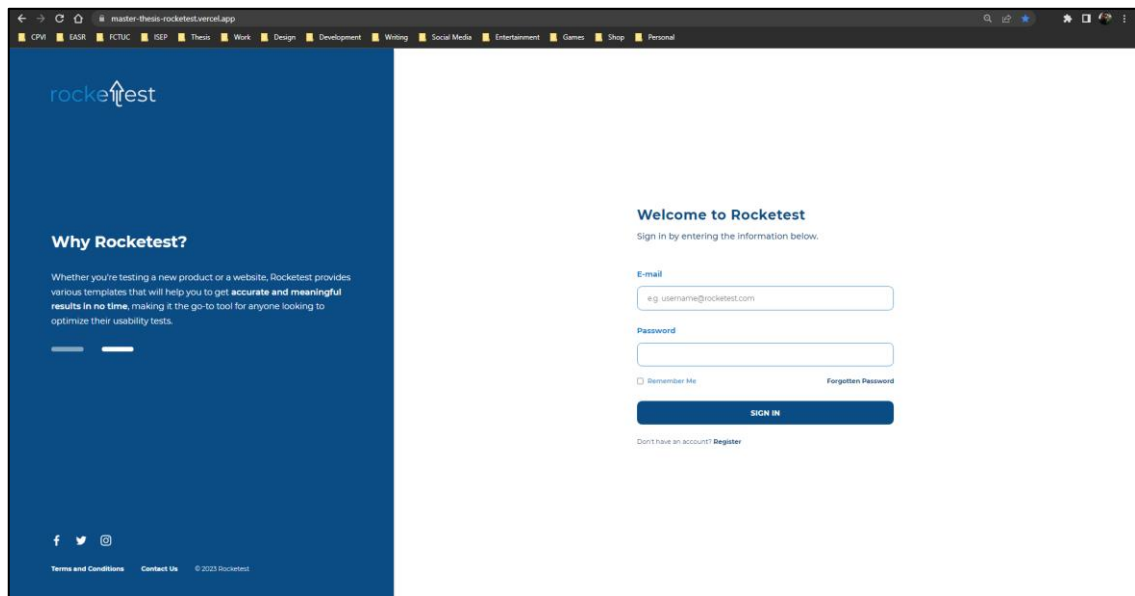
<https://github.com/sofia-amc3/master-thesis-rocketest>



Attachment H

Rocketest Application (Final Project Deployed)

<https://master-thesis-rocketest.vercel.app>



Attachment I

Quantitative Evaluation Framework (QEF)

<https://docs.google.com/spreadsheets/d/1nXlxaeiYCseeS0NiHJir1SKcgNiUQkrF/edit?usp=sharing&ouid=110499838716534677784&rtpof=true&sd=true>

q	D	Q	Dimension	Q2	Wj [Factor Weight] in Dim j [0,1]	Factor	n _{req} [Requirement weight k in Factor j] [2, 4, 6, 8, 10]	Requirement	wk % requirement fulfillment k [0, 100]	GRub Issue Reference		
97%	0.063	96,602	Functionality	97,959	0.524	Security and Support	10	PS101. Before Authentication: Sign In	100	#9		
							8	PS102. Before Authentication: Sign In + Remember Me	100	#9		
							8	PS103. Before Authentication: Recover Password	100	#10		
							10	PS104. Before Authentication: Sign Up as UX Researcher	100	#11		
							10	PS105. Before Authentication: Sign Up as Tester	100	#12		
							10	PS106. Before Authentication: Access information about Rockettest and its Terms and Conditions	100	#13		
							8	PS107. Before Authentication: Contact Rockettest's Team	100	#14		
							10	PS108. Sign Out of the Account	100	#19		
							4	PS109. Settings + Account: User is able to change his/her personal information	50	#15		
							10	PS110. User cannot access the application's pages (after authentication) without an authentication	100	-		
							10	PS111. User cannot access the Sign Up page without choosing an User Type first	100	-		
							10	PS112. Users can navigate throughout the side menu pages	100	-		
							10	PS113. Tests + My Tests: Search for Created Tests (UX R) or Answered Tests (Tester)	100	#2		
							10	PS114. Tests + My Tests: Users are able to use the search functionality, use filters and sort results	100	#1		
							10	PS115. Tests + My Tests: See Test Results (UX R) or Test Detail (Tester)	100	#3		
		95,109		0.476	Services and Features		10	PS116. Tests + My Tests: UX Researchers can find testers through an external API	75	#4		
							10	PS117. Tests + Create Test: UX Researchers can create a usability test from a template	100	#5		
							10	PS118. Tests + Create Test: UX Researchers can choose the testers' criteria for the test	100	#6		
							10	PS119. Tests + Testers can participate in a test	100	#8		
							4	PS120. Profile + Edit Profile: Users can edit their profiles and update their information	50	#18		
							10	PS121. Tests + Overview: For the tester, available tests are visible on this page	100	#21		
							2	UE01. A Back to Top arrow is provided when the vertical scroll value is big	100	#24		
							10	UE02. The application follows the color palette defined in the design stage	100	-		
							10	UE03. When clicking on the Rockettest logo, the user is redirected to the homepage	100	-		
							8	UE04. A loading animation is visible when the page has not fully loaded yet	100	#24		
							10	UE05. Users have access to a Search Bar in some pages of the application	50	#22		
							10	UIAN01. Tests + Overview: A Dashboard prototype is visible on this page for both user types	100	#20		
							10	UIAN02. Notifications: A prototype of this page is visible for both user types	100	#23		
							10	UIAN03. Messages: A prototype of this page is visible for both user types	100	#23		
							96,078	Usability	100	0.467	Accessibility and Navigation	10
10	UIAN05. Profile: A prototype of this page is visible for both user types	100	#23									
10	UIAN06. Profile: A prototype of this page is visible for both user types	100	#23									
10	UIAN07. Settings: A prototype of this page is visible for both user types	100	#16									
10	UIAR01. Not working pages/features are disabled and shown with decreased opacity	100	#23									
8	UIER02. Error messages are clear and provide enough information to recover from the error	100	-									
100		0.200	Help and Recover		6	UIHR03. A customized 404 page is shown when the URL does not exist					100	#24
					10	RM01. The project was developed using a version control system					100	-
					10	RM02. Code files are well structured and organized into folders					100	-
					10	RM03. A readme file is provided					100	-
					10	RP001. The technologies chosen have updated documentation					100	-
					10	RP002. The frontend of the application was developed following the final design prototype					100	-
					10	RP003. An API for searching for testers was used					75	-
					10	RP004. A database was developed and connected to this project					100	-
					100						0.429	Maintenance
8	PS102. Before Authentication: Sign In + Remember Me	100	-									
8	PS103. Before Authentication: Recover Password	100	-									
10	PS104. Before Authentication: Sign Up as UX Researcher	100	-									
10	PS105. Before Authentication: Sign Up as Tester	100	-									
10	PS106. Before Authentication: Access information about Rockettest and its Terms and Conditions	100	-									
8	PS107. Before Authentication: Contact Rockettest's Team	100	-									
10	PS108. Sign Out of the Account	100	-									
4	PS109. Settings + Account: User is able to change his/her personal information	50	-									
96,429	Reliability	93,750	0.571	Programming Practices		10	PS110. User cannot access the application's pages (after authentication) without an authentication	100	-			
						10	PS111. User cannot access the Sign Up page without choosing an User Type first	100	-			
						10	PS112. Users can navigate throughout the side menu pages	100	-			
						10	PS113. Tests + My Tests: Search for Created Tests (UX R) or Answered Tests (Tester)	100	-			
						10	PS114. Tests + My Tests: Users are able to use the search functionality, use filters and sort results	100	-			
						10	PS115. Tests + My Tests: See Test Results (UX R) or Test Detail (Tester)	100	-			
						10	PS116. Tests + Create Test: UX Researchers can create a usability test from a template	100	-			
						10	PS117. Tests + Create Test: UX Researchers can choose the testers' criteria for the test	100	-			
						10	PS118. Tests + Testers can participate in a test	100	-			

Dimension		Functionality		Factor		Wk - Fulfillment (%)				
Requirement		Metric Evaluation		0		25	50	75	100	
PS101. Before Authentication: Sign In		User is able to login on Rockettest after creating an account.		Not implemented		Not implemented				
PS102. Before Authentication: Sign In + Remember Me		When filling in the login form, user has access to a "Remember Me" checkbox that, when checked, saves the login information in local storage until the user logs out of his/her account.		Not implemented		The login information is stored on session storage only				
PS103. Before Authentication: Recover Password		User is able to recover password through his/her e-mail and set a new one.		Not implemented		An e-mail is sent but the user cannot set a new password				
PS104. Before Authentication: Sign Up as UX Researcher		User can Sign Up as UX Researcher by choosing it as his/her user type.		Not implemented		-				
PS105. Before Authentication: Sign Up as Tester		User can Sign Up as Tester by choosing it as his/her user type.		Not implemented		-				
PS106. Before Authentication: Access information about Rockettest and its Terms and Conditions		A slider showing Rockettest's app information is shown on the side menu before authenticating. In that same side menu, there is a "Terms and Conditions" page.		Not implemented		Only one of them ("Terms and Conditions"/information about the app) is available				
PS107. Before Authentication: Contact Rockettest's Team		A Contact UX page is available on the side menu before authenticating. When filling in the form, the default e-mail application of the user's device should open and store the information written on the "Subject" and "Message" fields of the e-mail to send.		Not implemented		The "Contact UX" page exists but the form does not open the e-mail app				
PS108. Sign Out of the Account		User can logout of his/her account.		Not implemented		-				
PS109. Settings + Account: User is able to change his/her personal information on the first section.		When accessing the Settings page, user is able to change his/her personal information on the first section.		Not implemented		User can access the settings page as a prototype but is not able to edit his/her personal information				
PS110. User cannot access the application's pages (after authentication) without an authentication		If the user changes the URL to one that is part of a page that requires authentication, access to that page must be blocked, returning to the login page.		Not implemented		-				
PS111. User cannot access the Sign Up page without choosing an User Type first		If the user changes the Sign Up page, access to that page must be blocked because a user type must be chosen first.		Not implemented		-				
Factor		Services and Features				Wk - Fulfillment (%)				
Requirement		Metric Evaluation		0		25	50	75	100	
PS101. Users can navigate throughout the side menu pages		After authenticating, users can access all of the pages present on the side menu of Rockettest.		Not implemented		Users can only access some of the pages present on the side menu				
PS102. Tests + My Tests: Search for Created Tests (UX R) or Answered Tests (Tester)		As a UX Researcher, when accessing the page "Testability Tests", created tests should be shown. As a Tester, answered tests should be shown.		Not implemented		The page is working only for one of the user types				
PS103. Tests + My Tests: Users are able to use the search functionality, use filters and sort results		Both UX Researchers and Testers can use the search, filters and sort functionalities on their created/answered tests.		Not implemented		The functionalities are present but they do not work as expected				
PS104. Tests + My Tests: See Test Results (UX R) or Test Detail (Tester)		As a UX Researcher, when accessing the page "Testability Tests" and clicking on a test, test results should be shown. As a Tester, test details should be shown.		Not implemented		The page is working only for one of the user types				
PS105. Tests + My Tests: UX Researchers can create a usability test from a template		As a UX Researcher, when accessing the page "Testability Tests" and clicking on a test, test results should be shown. In that same page, a button saying "Test" should lead to a view where he/she can freely create an external UX (LinkedIn, Instagram, Facebook, etc.).		Not implemented		The page is visible but the search for testers does not work				
PS106. Tests + Create Test: UX Researchers can create a usability test from a template		As a UX Researcher, when accessing the "Tests>Create Test" page and creating one test, he/she should be able to choose the testers' criteria for said test.		Not implemented		The search for testers is working, however, it has some bugs				
PS107. Tests + Create Test: UX Researchers can choose the testers' criteria for the test		As a UX Researcher, when accessing the "Tests>Create Test" page and creating one test, he/she should be able to choose the testers' criteria for said test.		Not implemented		The template for the test is visible but the test creation does not work				
PS108. Tests + Create Test: UX Researchers can participate in a test		As a Tester, when clicking on an available test, he/she can take the test.		Not implemented		-				
PS109. Profile + Edit Profile: Users can edit their profile and update their information		When accessing the Edit Profile page (inside Profile), user is able to update his/her information.		Not implemented		User can access the "Profile/Edit Profile" page as a prototype but is not able to edit his/her personal information				
PS110. Tests + Overview: For the tester, available tests are visible on this page		As a Tester, when accessing the "Tests/Overview" page, available tests should be visible in case he/she hasn't participated in them		Not implemented		Tester can access the "Tests/Overview" page as a prototype but is not able to access available tests				

Dimension		Reliability				
Factor		Effectiveness				
Requirement		Metric Evaluation		Wk - Fulfillment (%)		
		0	25	50	75	100
UE01: A Back to Top arrow is provided when the vertical scroll value is big		When the vertical scroll value surpasses a certain value, an icon with a Back to Top arrow should be visible and when clicked on it should redirect the user to the beginning of the current page.	Not Implemented	-	The icon is visible but it does not work	Implemented
UE02: The application follows the color palette defined in the design stage		The colors shown in the application are the same as the ones used on the interactive prototype made on Figma.	Not Implemented	-	-	Implemented
UE03: When clicking on the Rocketeet logo, the user is redirected to the homepage		If the user is authenticated, clicking on Rocketeet's logo present on the side menu should redirect the user to the Tests page, and if unauthenticated it should redirect him/her to the sign in page.	Not Implemented	-	Only implemented on one of the menus (before or after authentication)	Implemented
UE04: A loading animation is visible when the page has not fully loaded yet		When the page is not fully loaded, a loading animation should play indicating that the page is still loading.	Not Implemented	-	-	Implemented
UE05: Users have access to a Search Bar in some pages of the application		In some pages of the application, a Search Bar is visible.	Not Implemented	-	The Search Bar is visible but is not working	Implemented
Factor		Accessibility and Navigation				
Requirement		Metric Evaluation		Wk - Fulfillment (%)		
		0	25	50	75	100
UAM01: Tests > Overview: A Dashboard prototype is visible on this page for both user types		An image prototype (just like the interactive prototype on Figma) is visible for both user types representing the features in this page that were not supposed to be implemented and, therefore, are disabled.	Not Implemented	-	The image is only visible to one of the user types	Implemented
UAM02: Notifications: A prototype of this page is visible for both user types		An image prototype (just like the interactive prototype on Figma) is visible for both user types representing the features in this page that were not supposed to be implemented and, therefore, are disabled.	Not Implemented	-	The image is only visible to one of the user types	Implemented
UAM03: Messages: A prototype of this page is visible for both user types		An image prototype (just like the interactive prototype on Figma) is visible for both user types representing the features in this page that were not supposed to be implemented and, therefore, are disabled.	Not Implemented	-	The image is only visible to one of the user types	Implemented
UAM04: Watch: A prototype of this page is visible for both user types		An image prototype (just like the interactive prototype on Figma) is visible for both user types representing the features in this page that were not supposed to be implemented and, therefore, are disabled.	Not Implemented	-	The image is only visible to one of the user types	Implemented
UAM05: Profile: A prototype of this page is visible for both user types		An image prototype (just like the interactive prototype on Figma) is visible for both user types representing the features in this page that were not supposed to be implemented and, therefore, are disabled.	Not Implemented	-	The image is only visible to one of the user types	Implemented
UAM06: Profile - Edit Profile: A prototype of this page is visible for both user types		An image prototype (just like the interactive prototype on Figma) is visible for both user types representing the features in this page that were not supposed to be implemented and, therefore, are disabled.	Not Implemented	-	The image is only visible to one of the user types	Implemented
UAM07: Settings: A prototype of this page is visible for both user types		An image prototype (just like the interactive prototype on Figma) is visible for both user types representing the features in this page that were not supposed to be implemented and, therefore, are disabled.	Not Implemented	-	The image is only visible to one of the user types	Implemented
Factor		Help and Recover				
Requirement		Metric Evaluation		Wk - Fulfillment (%)		
		0	25	50	75	100
UHR01: Not working pages/features are disabled and shown with decreased opacity		To better understand what is working and what is not, all features that are not working in Rocketeet should be disabled (represented by decreased opacity) and not clickable/interactable.	Not Implemented	-	Some non working features are not disabled	Implemented
UHR02: Error messages are clear and provide enough information to recover from the error		Whenever an error occurs, an alert or popup should be shown stating what error occurred and how the user can recover from it.	Not Implemented	-	Some errors are not well explained/easy to understand	Implemented
UHR03: A customized 404 page is shown when the URL does not exist		Whenever a page does not exist, a customized 404 page should be shown.	Not Implemented	-	The page appears but is not customized	Implemented

Dimension		Reliability				
Factor		Maintenance				
Requirement		Metric Evaluation		Wk - Fulfillment (%)		
		0	25	50	75	100
BM01: The project was developed using a version control system		When entering the development phase, a version control system was used to store the project's commits and progress.	A version control system was not used	-	-	A version control system was used
BM02: Code files are well structured and organized into folders		It is easy to understand the different coding pages and files because they were organized into their respective folders.	The files are not organized	Very few files are organized	Some files are organized	All the files are organized
BM03: A 'readme' file is provided		A 'readme' file providing documentation on how to run the project was written.	The file does not exist	-	The file exists but is incomplete or not well explained	The file exists
Factor		Programming Practices				
Requirement		Metric Evaluation		Wk - Fulfillment (%)		
		0	25	50	75	100
RPP01: The technologies chosen have updated documentation		The technologies used for the development of this project are updated and it is easy to access their documentation.	The documentation is not updated	-	The documentation is somewhat updated	The documentation is updated
RPP02: The frontend of the application was developed following the final design prototype		The interactive prototype made on Figma regarding the final design of Rocketeet was followed throughout the frontend development.	The prototype was not followed	-	The prototype was somewhat followed	The prototype was followed
RPP03: An API for searching testers was used		An API for searching testers on external platforms was used.	An API was not used	-	-	An API was used but it was developed from scratch with fictional users
RPP04: A database was developed and connected to this project		All of the information present on Rocketeet is stored on a database that was developed specifically for this project's requirements.	A database was not used	-	A database was used, however, it does not properly connect to the project	A database was used

Attachment J

System Usability Scale Results

<https://docs.google.com/spreadsheets/d/1ksYChROirsjTBQAWPbcFf2XyVJGM9wBW-1QhXZ2VEcs/edit?usp=sharing>

Date/Time Stamp	Are you used to worki	Are you 18 years old or	Do you have any acce	I think that I would lik	I found the system un	I thought the system \	I think that I would ne
2023/03/06 14:16:53	Yes	Yes	No	Agree	Neutral	Agree	Neutral
2023/03/07 22:18:05	Yes	Yes	Yes	Strongly agree	Strongly disagree	Strongly agree	Disagree
2023/03/08 01:36:34	Yes	Yes	No	Neutral	Disagree	Agree	Disagree
2023/03/13 16:42:50	Yes	Yes	No	Agree	Strongly disagree	Strongly agree	Disagree
2023/03/13 17:58:43	Yes	Yes	No	Neutral	Disagree	Agree	Agree
2023/03/14 15:04:46	Yes	Yes	No	Strongly agree	Strongly disagree	Strongly agree	Strongly disagree
2023/03/14 15:06:13	Yes	Yes	No	Agree	Strongly disagree	Strongly agree	Strongly disagree
2023/03/14 15:07:10	Yes	Yes	No	Agree	Disagree	Agree	Neutral
2023/03/14 15:08:09	Yes	Yes	No	Strongly agree	Strongly disagree	Neutral	Strongly disagree
2023/03/14 15:09:14	Yes	Yes	No	Strongly agree	Strongly disagree	Strongly agree	Strongly disagree
2023/03/14 15:13:15	Yes	Yes	No	Disagree	Strongly disagree	Strongly agree	Disagree
2023/03/14 15:13:56	Yes	Yes	No	Strongly agree	Strongly disagree	Strongly agree	Strongly disagree

I found the various fur	I thought there was to	I would imagine that r	I found the system ver	I felt very confident us	I needed to learn a lot
Agree	Strongly disagree	Agree	Disagree	Neutral	Strongly disagree
Agree	Disagree	Agree	Neutral	Agree	Disagree
Agree	Disagree	Neutral	Disagree	Neutral	Neutral
Agree	Strongly disagree	Agree	Strongly disagree	Strongly agree	Disagree
Agree	Disagree	Agree	Disagree	Agree	Neutral
Strongly agree	Strongly disagree	Strongly agree	Strongly disagree	Strongly agree	Strongly disagree
Agree	Strongly disagree	Agree	Strongly disagree	Neutral	Strongly disagree
Agree	Disagree	Agree	Disagree	Agree	Disagree
Agree	Strongly disagree	Neutral	Disagree	Agree	Strongly disagree
Strongly agree	Strongly disagree	Strongly agree	Strongly disagree	Strongly agree	Strongly disagree
Strongly agree	Strongly disagree	Disagree	Strongly disagree	Disagree	Agree
Strongly agree	Strongly disagree	Strongly agree	Strongly disagree	Strongly agree	Strongly disagree

Attachment K

Form – Feedback Survey

<https://forms.gle/rqfp5XKJ2BSd72M9A>

The image shows a Google Form titled "Rocketest · Feedback Survey". The form is part of a Master's Thesis by Sofia Alexandra Moreira Canelas, titled "Participant Selection Process in a Usability Testing Platform". The form is designed for a Rocketest platform. The form is divided into two main sections: a header section and a questionnaire section. The header section includes the Rocketest logo, the title "Rocketest · Feedback Survey", a brief introduction about the form's purpose, and contact information for Sofia (sofia.amc3@gmail.com). The questionnaire section starts with a question: "Are you 18 years old or over?". Below this is another question: "Do you have any accessibility deficit?". The form also includes a "Personal Information" section with a "Name" field and a question: "Are you used to working on a computer regularly?". The form is created using Google Forms and is hosted on the Rocketest platform.

rocketest

Rocketest · Feedback Survey

This form is a part of the Experimentation and Evaluation chapter from Sofia Alexandra Moreira Canelas' Master's Thesis, entitled "Participant Selection Process in a Usability Testing Platform".

This dissertation was written within the scope of the curricular unit Master Thesis (TMDEI) of the second year of the Master's Degree in Informatics Engineering in the Graphic Systems and Multimedia branch held at the Instituto Superior de Engenharia do Porto (ISEP).

Please read the context before moving on to the questionnaire.

sofia.amc3@gmail.com [Mudar de conta](#)

Não partilhado

rocketest

Rocketest · Feedback Survey

sofia.amc3@gmail.com [Mudar de conta](#)

Não partilhado

* Indica uma pergunta obrigatória

Personal Information

This information will only be used to check if our participants' characteristics match the target audience we are looking for. Note that the name field is optional, you can remain completely anonymous if you prefer.

Name

A sua resposta

Are you used to working on a computer regularly? *

☐ Yes

☐ No

Are you 18 years old or over? *

☐ Yes

☐ No

Do you have any accessibility deficit? *

☐ Yes

☐ No

[Anterior](#) [Seguinte](#) [Limpar formulário](#)

Nunca envie palavras pessoais através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) [Termos de Utilização](#) [Política de privacidade](#)

Google Formulários



Rocketest · Feedback Survey

sofia.amc3@gmail.com [Mudar de conta](#)

 Não partilhado



FEEDBACK SURVEY

Please test the following application by opening this link:

<https://master-thesis-rocketest.vercel.app/>

And after you open it, please come back to this questionnaire and proceed to the next section after reading the paragraphs below.

If possible, please test it on a PC with the resolution of **1920x1080px**, as the application is not responsive to all screen resolutions at this stage. Also ensure that your browser's window is maximized just so it stays responsive.

When testing Rocketest, see if you can easily navigate between pages, and **take notes** of any issues you might have encountered or any suggestion you might have for us, as you will be able to provide us your feedback along the questionnaire.

The parts of the questionnaire that are supposed to be tested will be indicated in the section's name. For example, you are going to start the following questionnaire by testing the "Terms and Conditions", which you should be able to find by yourself when opening the application.

[Anterior](#)

[Seguinte](#)


[Limpar formulário](#)

Nunca envie palavras-passe através dos Google Forms.




Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) - [Termos de Utilização](#) - [Política de privacidade](#)






Rocketest · Feedback Survey

sofia.amc3@gmail.com [Mudar de conta](#)

 Não partilhado



* Indica uma pergunta obrigatória

QUESTIONNAIRE FOR UX RESEARCHER

In this part of the questionnaire we will ask you to test Rocketest as a **UX Researcher**, therefore, you should choose this user type when we ask you to test the Sign Up page.

BEFORE AUTHENTICATION

Terms and Conditions

Please navigate to this page.

How easy was it to find the terms and conditions? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

How well-presented and structured was the information? *

Poorly structured and presented

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very well structured and presented

If you wish, you can provide us with your feedback in this section.

A sua resposta

Contact Us

Please navigate to this page.

How easy was it to find the contact us page? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

After filling in the contact us form and clicking on "Send Message" did the default email application of your device open with the information you wrote on the form? *

☐ Yes

☐ No

Was the email sent? *

☐ Yes

☐ No

If you encountered any issue during this phase, please provide us feedback.

A sua resposta

Sign Up as UX Researcher (Create Account)

Please navigate to this page.

How easy was it to find the register button? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

Were you able to choose the user type "UX Researcher" before registering? *

☐ Yes

☐ No

How easy was the registry process to complete? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

If you encountered any issue during this phase, please provide us feedback.

A sua resposta

AFTER AUTHENTICATION

Tests

Please navigate to this page.

After signing in, were you able to see your information on the left side menu? *

☐ Yes

☐ No

When on the "Tests" page, was the "Overview" page visible? (Note that this is one of the pages that is disabled, so none of the features work or are interactable, except for the top menu) *

☐ Yes

☐ No

Tests | Create Test

Please navigate to this page.

When clicking on "Create Test", were you able to select the "A/B Test" template? *

☐ Yes

☐ No

Try filling up the fields on the "Edit Test" part. When editing the test, were you able to see on the right side a preview of the elements you were editing? *

☐ Yes

☐ No

Were you able to use the controls "Add Question", "Add Section", and "Delete"? *

☐ Yes

☐ No

When having more than one element (questions, sections, etc.) were you able to know the specific element you were editing by the "focus" it had when you clicked on it? *

☐ Yes

☐ No

How would you rate this process of editing a test? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

When moving onto the next step - "Test Details" - were you able to set the testers' criteria and the test's information? (Please set the test privacy to "Public" just so we can use it later on) *

☐ Yes

☐ No

How easy was it to understand how to fill in these fields? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

Were you able to create a test? *

☐ Yes

☐ No

If you encountered any issue during this phase, please provide us feedback.

A sua resposta


Tests | My Tests

Please navigate to this page.

When clicking on "My Tests", were you able to see the test you had just created? *

☐ Yes

☐ No



Rocketest · Feedback Survey

sofia.amc3@gmail.com [Mudar de conta](#)

Não partilhado

* Indica uma pergunta obrigatória

QUESTIONNAIRE FOR TESTER

In this part of the questionnaire we will ask you to test Rocketest as a **Tester**, therefore, you should choose this user type when creating an account. Note that some general things common to both users were already tested before, so we won't ask you to test them again.

BEFORE AUTHENTICATION

Sign Up as Tester (Create Account)

Please navigate to this page.

Were you able to choose the user type "Tester" before registering? *

☐ Yes

☐ No

How easy was the registry process to complete? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

If you encountered any issue during this phase, please provide us feedback.

A sua resposta

AFTER AUTHENTICATION

Tests | Overview

Please navigate to this page.

After signing in, were you able to see your information on the left side menu? *

☐ Yes

☐ No

When on the "Tests/Overview" page, are you able to see available tests, * including the one you had just created using the UX Researcher account? (Note that the test should only appear if you set its privacy to "Public" and if you're viewing "All Tests", not just the ones with "Matched Criteria")

☐ Yes

☐ No

Were you able to use the search bar, filters and sort functionalities? *

☐ Yes

☐ No

Click on one of the tests. Are you able to fulfill all the fields and submit * your answers?

☐ Yes

☐ No

If you encountered any problem during this stage please tell us about it. You can also provide us with your feedback on this part.

A sua resposta

Tests | My Tests

Please navigate to this page.

When accessing "Tests/My Tests" are you able to see the test you've just answered? *

☐ Yes

☐ No

When clicking on that test are you able to revisit your answers? *

☐ Yes

☐ No

If you wish, you can provide us with your feedback in this section.

A sua resposta

Other Pages

Please navigate to this page.

When navigating through "Notifications", "Messages", "Wallet", "Profile" and "Settings" are you able to see a prototype of these pages, even if they're not functional? *

☐ Yes

☐ No

Go to the "Profile" page. Are you able to click on the "Edit Profile" icon and see this page's prototype as well? *

☐ Yes

☐ No

If you wish, you can provide us with your feedback in this section.

A sua resposta

Sign Out
Please navigate to this page.

Were you able to Sign Out of the account? *

☐ Yes

☐ No

If you encountered any issue during this phase, please provide us feedback.

A sua resposta

BEFORE AUTHENTICATION

Forgotten Password
Please navigate to this page.

How easy was it to find and understand the process of recovering/resetting a password? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

Did you encounter any problems when doing this process? *

☐ Yes

☐ No

If you encountered any problems, please specify.

A sua resposta

Did you receive an email that allowed you to recover the password? *

☐ Yes

☐ No

Were you able to reset the password? *

☐ Yes

☐ No

If you weren't able to reset the password, please tell us why.

A sua resposta

Sign In
Please navigate to this page.

How easy was it to understand what fields you needed to fill? *

Very Hard

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Very Easy

Did you authenticate successfully? *

☐ Yes

☐ No

If you weren't able to authenticate, please tell us why. You can also provide us with your feedback on this part.

A sua resposta

[Anterior](#) [Seguinte](#) [Limpar formulário](#)

Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) · [Termos de Utilização](#) · [Política de privacidade](#)

Google Formulários



Rocketest · Feedback Survey

sofia.amc3@gmail.com [Mudar de conta](#)

 Não partilhado



Additional Feedback

In this space, you can provide us with **general feedback** for improving Rocketest.

We would very much appreciate your feedback regarding aspects like your general experience with the application (i.e., how easy was it to learn how to use, is its content understandable and grammatically correct, how quick was it to perform different tasks in it, does it provide good feedback on the operations performed, etc.), and, especially, if you've encountered any bugs or errors throughout the process of testing it. You can also leave here any suggestions you might have.

A sua resposta

[Anterior](#)

[Enviar](#)

[Limpar formulário](#)

Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) - [Termos de Utilização](#) - [Política de privacidade](#)

Google Formulários



Attachment L

Feedback Survey Results

<https://docs.google.com/spreadsheets/d/1zmFDL-17II71MsmBwRAiH55luZQdkBZQDD2xgIH9VXQ/edit?usp=sharing>

Date/Time Stamp	Name	Are you used to work?	Are you 18 years old or	Do you have any accor	How easy was it to find	How well-presented a	If you wish, you can pr	How easy was it to find	After filling in the cont
2023/06/20 16:25:40	André 'O BARBUDO'	Yes	Yes	No	5	5	Love the colors	5	Yes
2023/06/21 20:39:15	Tomás	Yes	Yes	No	5	5		5	Yes
2023/06/23 18:06:05	Cárina	Yes	Yes	No	3	4	Foi um pouco difícil er	3	Yes
2023/06/25 18:06:08	Andre Oliveira Gonçalves	Yes	Yes	No	4	4	It was fine, but I was t	5	Yes
2023/06/26 16:49:12		Yes	Yes	No	5	3		5	Yes
2023/06/26 16:55:44		Yes	Yes	No	5	5		5	Yes
2023/06/26 23:05:57	Narciso Correia	Yes	Yes	No	5	5		5	Yes
2023/06/26 23:31:34	Francisca Teixeira	Yes	Yes	No	5	5		5	Yes
2023/06/27 00:21:34	Vitor Neto	Yes	Yes	No	5	5		5	Yes
2023/06/27 14:16:32		Yes	Yes	No	5	4		5	Yes
Average Results					5,00	5,00		5,00	

Was the email sent?	If you encountered an	How easy was it to find	Were you able to choo	How easy was the regi	If you encountered an	After signing in, were	When on the "Tests" p	When clicking on "Cre	Try filling up the fields
					aparece um erro esta { "length": 96, "name": "error", "severity": "ERROR", "code": "42601", "position": "78652", "file": "scan.js", "line": "1176", "routine": "scanner.js" }				
Yes	Tudo funcionou.	5	Yes	1	quando retirei uma pl	Yes	Yes	Yes	Yes
Yes		5	Yes	4		Yes	Yes	Yes	Yes
Yes	Mesmo feedback de a	5	Yes	5		Yes	Yes	Yes	Yes
Yes	Nothing so far, but I'm	5	Yes	5		Yes	Yes	Yes	Yes
Yes		5	Yes	5		Yes	Yes	Yes	Yes
Yes		5	Yes	5		Yes	Yes	Yes	Yes
Yes		5	Yes	5		Yes	Yes	Yes	Yes
Yes		5	Yes	5		Yes	Yes	Yes	Yes
Yes		5	Yes	5		Yes	Yes	Yes	Yes
Yes		5	Yes	4		Yes	Yes	Yes	Yes
		5,00		5,00					

Were you able to use t	When having more th	How would you rate t	When moving onto th	How easy was it to unt	Were you able to creat	If you encountered an	When clicking on "My	Please repeat the step	Click on one of the tes
						voitou a dar um erro, c { "length": 111, "name": "error", "severity": "ERROR", "code": "42703", "position": "2604", "file": "parse_relation "line": "3548", "routine": "errorMissi }			
Yes	Yes	5	Yes	5	Yes		Yes	Yes	Yes
Yes	Yes	4	Yes	5	Yes		Yes	Yes	Yes
Yes	Yes	4	Yes	5	Yes	A feature de adicionar	Yes	Yes	Yes
Yes	Yes	4	Yes	5	Yes	No issue.	Yes	Yes	Yes
Yes	Yes	5	Yes	3	Yes	I'm assuming if you're	Yes	Yes	Yes
Yes	Yes	5	Yes	5	Yes		Yes	Yes	Yes
Yes	Yes	5	Yes	5	Yes	The inputs need an co	Yes	Yes	Yes
Yes	Yes	5	Yes	5	Yes	The inputs need a stat	Yes	Yes	Yes
Yes	Yes	5	Yes	5	Yes	Awesome process 🥰	Yes	Yes	Yes
Yes	Yes	5	Yes	5	Yes		Yes	Yes	Yes
		5,00		5,00					

Go to the "Find Teste	Were you able to send	After sending this me	Go back to the "Test	Go back to the "Test	D	How easy was it to nav	If you encountered an	When navigating thro	Go to the "Profile" pag	How easy was it to unt
Yes	Yes	Yes	Yes	Yes		5	Tudo parece funcional	Yes		5
Yes	Yes	Yes	Yes	Yes		5		Yes	Yes	5
Yes	Yes	Yes	No	Yes		4	Na pagina de "View R	Yes	Yes	5
Yes	Yes	Yes	Yes	Yes		5		Yes	Yes	5
Yes	Yes	Yes	Yes	Yes		4	No issues. The use of	Yes	Yes	5
Yes	Yes	Yes	Yes	Yes		5		Yes	Yes	5
Yes	Yes	Yes	Yes	Yes		5	The inputs need an co	Yes	Yes	5
Yes	Yes	Yes	Yes	Yes		5	The inputs need a stat	Yes	Yes	5
Yes	Yes	Yes	No	No		5	I could not delete my	Yes	Yes	5
Yes	Yes	Yes	Yes	Yes		5	The idea of finding tes	Yes	Yes	5
						5,00				5,00

If you wish, you can print	Were you able to Sign In?	If you encountered an error	Were you able to choose a password?	How easy was the registration process?	If you encountered an error	After signing in, were you able to see the tests?	When on the "Tests/Reports" page, were you able to use the filters?	Were you able to use the "Print" button?	Click on one of the tests
O botão de editar está funcionando?	Yes	Tudo pareceu funcionar?	Yes	4	Não permite caracteres especiais	Yes	Yes	No	Yes
Yes	Yes	Yes	Yes	4	Yes	Yes	Yes	Yes	Yes
A página poderia ser melhorada?	No	Yes	Yes	5	Yes	Yes	Yes	Yes	Yes
It's fine	Yes	It's fine	Yes	5	Yes	Yes	Yes	Yes	Yes
The "Edit Profile" button works?	Yes	Yes	Yes	4	No issues. The tester is satisfied	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	5	Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	5	Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	5	Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	5	Yes	Yes	Yes	Yes	No
Yes	Yes	Yes	Yes	5	Yes	Yes	Yes	Yes	Yes
				5,00					

If you encountered an error	When accessing "Test" page, when clicking on that	If you wish, you can print	When navigating through the "Profile" page	Go to the "Profile" page	If you wish, you can print	Were you able to Sign In?	If you encountered an error	How easy was it to find the tests?	
O teste que criei chama-se	Yes	Yes	Parece-me bem.	Yes	Yes	tal como na área de Respostas	Yes	Tudo parece estar a funcionar	5
	Yes	Yes		Yes	Yes		Yes		4
	Yes	Yes		Yes	Yes		Yes		5
	Yes	Yes		Yes	Yes		Yes		5
	Yes	Yes		Yes	Yes		Yes		5
	Yes	Yes		Yes	Yes		Yes		4
	Yes	Yes		Yes	Yes		Yes		5
	Yes	Yes		Yes	Yes		Yes		5
I could not submit the test	No	No	I could not answer any question	Yes	Yes		Yes		5
	Yes	Yes		Yes	Yes		Yes		5
									5,00

Did you encounter an error? If you encountered an error, Did you receive an email? Were you able to reset? If you weren't able to, How easy was it to find? Did you authenticate successfully? If you weren't able to, How would you very much appreciate your feedback regarding aspects like your general experience with the application (i.e., how easy was it to use)?									
No	Demorou um bocadinho	Yes	Yes	Tudo funcionou na perfeição	5	Yes	Tudo funcionou na perfeição	Coder muito do Logo, o design do website está muito clean. Por questões de segurança, a área de Wallac, o cartão não pode ter os nomes reais.	
Yes	The email went to the inbox	Yes	Yes		5	Yes		I very much liked the app. I looking forward to try it again.	
No	No	No	No	Tive um erro "Email not found"	5	Yes		Gostei muito do design, do logo e da facilidade de melhorar os pontos anteriores especialmente as páginas serem responsivas.	
No	Yes	Yes	Yes		5	Yes		For some reason there where two of the same tests created but it only happens once. Besides that, no complaints.	
No	Yes	Yes	Yes		5	Yes		It was easy to learn how to use despite having never used these type of applications before. The tests are performed quickly and it does not have any complaints or suggestions, the application worked fine.	
Yes	The process works but	Yes	Yes		5	Yes			
No	Yes	Yes	Yes		5	Yes		I think that the app was quite easy to navigate and intuitive to use. I haven't detected any errors or bugs from it. Overall, it's a really nice app.	
No	Yes	No	No	Error when setting new password	5	Yes			
No	Yes	Yes	Yes		5	Yes		Very nice	
					5,00				