Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

João Diogo Burrica Barroso

HEALTHCARE RULE-BASED EXPERT SYSTEM FRAMEWORK TO HELP, EDUCATE AND PREPARE USERS

Project submitted as partial requirement for obtaining the **Master's Degree in Software Engineering**.

Jury

Presidente: Dr. Cláudio Sapateiro, ESTS

Orientador: Dr. José Sena Pereira, ESTS

Arguente: Dr. Cédric Grueau, ESTS

Setembro, 2021

I dedicate this work to my Parents

who have always been there to help and

support me

Acknowledgements

This project was made to conclude the master's degree of Software Engineering which took part throughout 3 years of academic work. Therefore, I'd like to thank several people who made it possible for me conclude this great part of my life:

To Professor José Sena Pereira for his availability and guidance throughout the many months of development, providing the necessary information and clarifications.

I would like to thank to my father and Head Nurse Fernando Barroso for proposing the topic that served as the basis for this thesis.

To my family for always supporting me, incentivizing, and pushing me to always do better.

Resumo

Existe neste momento um grande problema com a abundância de informação relativa a saúde, a desinformação. Este problema tem causado que uma grande percentagem de pessoas faça visitas ao hospital desnecessariamente, ou pior. Como não foi possível encontrar uma aplicação móvel que disponibiliza um conjunto de informação cientificamente correta, disponível para todos, organizada e acessível, foi procurado neste documento resolver este problema. Para isto, um estudo foi conduzido para entender o estado da arte sobre aplicações móveis e sobre o ecossistema de aplicações para a saúde, desta forma endereçando diversos tópicos como acessibilidade, UX/UI e usabilidade. Para combater a desinformação, é importante ser capaz de disponibilizar aos utilizadores informação de confiança e para isso concluiu-se que a melhor e mais confiável informação teria que ser recolhida de especialistas os profissionais de saúde. Um protótipo foi construído que incorpora não só um sistema pericial que, baseado em regras criadas através da informação recolhida dos especialistas, podes disponibilizar um sistema de pesquisa de sintomas, mas também diversas funcionalidades de usabilidade e opções de acessibilidade. Este protótipo abre a possibilidade de desenvolver conjunto de novas funcionalidades e/ou melhorar aquelas já implementadas, como um algoritmo baseado em regras mais dinâmico ou testes de utilizadores mais extensos que permitem uma aplicação móvel mais completa em termos de acessibilidade e usabilidade.

Palavras-chave: sistema pericial, rule-base, aplicações móveis, acessibilidade, UX/UI, literacia em saúde

Abstract

There is currently a major problem with the abundance of information regarding health, misinformation. This problem causes a large percentage of people that take unnecessary trips to the hospital or worse. As there is not a suitable mobile app that conveys a set of scientifically correct, easily available, and organized information that is accessible to everyone, in this document we sought to solve this issue. For this, a study was done to understand the state of the art of mobile applications and about the current ecosystem of healthcare apps, addressing several issues such as accessibility, UX/UI and usability. To tackle misinformation, it was important being able to provide reliable information to users, and therefore it was concluded that the best and most reliable information had to be gathered from experts, health professionals. A prototype was built that not only encompasses an expert-system that can, based on rules made with the information gathered from experts, provide a symptom search system, but also incorporates several usability features and accessibility options. This prototype opens the opportunity to develop a range of new features and/or enhance those already implemented, such as an improved and more dynamic rule-based algorithm or extensive user testing that allows for a full-fledged accessible and easy to use mobile application.

Keywords: expert system, rule-base, mobile application, accessibility, UX/UI, health literacy

Index

Image Index

Table Index

1. Introduction

In this chapter it will be introduced the main problems regarding the online misinformation in the health landscape as well as present the main goals for this project.

Not so long ago, the biggest companies only had information displayed through web browsers only available in computers, but we are already in the next era of technology and dissipation of information, the Mobile era. Nowadays we live more connected to each other than ever before. Through our smartphones we can quickly access an exorbitant amount of information that was not easily available a decade ago. Interacting with our mobile devices is becoming a must. Information is circulating faster than ever.

In the context of health, more specifically the degree of "Health Literacy", people are constantly searching and browsing the web, through their computers or smartphones, to find answers for their concerns.

This is where we find a meaningful flaw. We cannot find a suitable mobile app that conveys a set of scientifically correct, easily available and organized information in the case of a critical event – a hospitalisation in a health institution, symptom or disease search. Does everybody know what should they take with them to a hospital? What data will they be solicited? What aspects must be taken into consideration to guarantee their safety during their hospital visit or hospitalisation? The answers to these questions should be easily available in an easy to access mobile application that users can rely on to help and guide them reacting to symptoms, understand their meaning and preparing for possible hospital visits.

A major problem with the abundance of information in the context of health is the ever-present misinformation and unreliable sources that can provide, for example, scientifically incorrect symptomatology. This misinformation can unlawfully guide users/patients to make bad choices, scare them or possibly aggravate their health.

Health is essential to everyone, therefore people with disabilities and the elderly also need to be taken into consideration. Building accessible applications is needed to comply and allow everyone to properly navigate and benefit from scientifically correct and reliable information.

1.1. Problem Definition

There is currently a large percentage of people who take unnecessary trips to the hospital to seek help for symptoms or problems that can easily be taken care of at the comfort of their own homes or, even worse, cases were the person has not taken proper timely primary care and is in dire need of medical assistance. The lack of information/education regarding symptoms and diseases a person might have, can lead to an influx of avoidable hospital visits, possibly putting the health of those people at even more risk.

Appropriate information and education about healthcare are essential and should be easily available to people. The information that is presented to the users must be gathered from professionals (experts).

As we move to a more mobile-connected world, people are becoming more connected to their phones and apps. This way people and patients can benefit from an easily available application with all the information and solutions that they need.

Since health is essential for everyone, young or old, with or without disabilities, it is important that everyone can easily find information and educate themselves. It is important that the application is accessible and of easy use.

1.2. Objectives

It is important that the problems mentioned previously gain visibility so that everyone can work together and tackle them.

As there are not many mobile applications focused on health literacy, a high-level study about these apps will be made regarding accessibility, UX and UI so that the problems can be identified. This is essential to understand how a better mobile application can be made for everyone.

For combating misinformation, a prototype will be built using a type of Expert System that represents the capturing of knowledge from experts and through rules, is able to answer effectively to a symptom search.

Furthermore, this prototype will be built with accessibility in mind, providing as many people as possible several ways they can interact and navigate through the app.

1.3. Document's Structure

The chapters are structured in the following manner:

- **Chapter 2**: Contextualization and state of the art of the several topics regarding mobile applications, accessibility and UX
- **Chapter 3**: Information architecture, exploring how to display information to users
- **Chapter 4**: Analysis of similar apps to gather information and make improvements to the mobile application
- **Chapter 5**: Expert system and contextualizing how it can help provide a reliable information to user
- **Chapter 6**: Conceptual basis of the desired mobile application
- **Chapter 7**: Built prototype, languages and features implemented
- **Chapter 8**: Conclusion and future work

2. Contextualization

In this chapter it will be presented the state of the art for several main topics to better understand their current impact and how it is possible to take advantage of them towards tackling the issues mentioned previously.

2.1. State of the art for Mobile Applications

By definition, a mobile application, mostly referred as “app”, is a software made to run on smartphones and/or tablets. Mobile Apps frequently provide users with similar services to those accessed on computers characterized for being small, individual software with limited function.

Apps where first popularized by Apple Inc. and its App Store, which offers thousands of applications for the iPhone and iPad.

The purpose of these apps is to fulfill different aspects of the user such as utility, productivity, navigation, entertainment and others. Social media is one of the most popular fields of mobile apps development and adoption.

When considering operating system compatibility, a developer creates an app specifically for the operating system in which he intends his application to run. That means apps made for the Apple ecosystem will not run in an Android (Google) ecosystem.

Regardless of the choice of operating system, whenever you buy a mobile device, either being a smartphone, tablet or even a smartwatch, you are granted pre-installed apps such as a web browser, email client, calendar, mapping program, media and other apps.

Nowadays we have a wide range of app categorizations, but they can always be identified by their type. There are 3 main types of mobile applications (Rosul, 2020):

- Native Apps – are made for a single operating system exclusively, therefore being “native” for that platform or device. That being said, a native app made for iOS can’t be run on Android. These apps have a clear advantage of their higher performance as developers use native device UI, this way ensuring a good user experience. A clear disadvantage associated with these apps is their higher cost compared to other types of apps. As there may be a need to create an equal application for another operating system, separating support and maintenance can easily result in a bigger product price.
- Hybrid Apps – are built for multiple platform web technologies but wrapped inside a native app. These are fast and relatively easy to develop. Having a single code base for all platforms ensures low-cost maintainability and ease to update. They also have available widely used APIs such as gyroscope, accelerometer, geolocation, etc. Despite having great features, these hybrid applications lack in performance, speed

and optimization when compared to native apps, for example. As some design might change from platform to platform, there might be issues related to the inability to have the same look and feel in exactly same way on different platforms.

- Web Apps – use a browser to run and are usually written in HTML, JavaScript or CSS. They redirect a user to URL and offer “install” option by simply creating a bookmark to their page on the user’s application menu. Web applications often require minimum of device memory. Although, as an internet connection is required to get data from the servers, the use of these types of apps with a poor connection can result in bad user experience. A clear stumbling block is that there are not many APIs for developers, with exception of geolocation a few others.

2.2. Statistical information regarding Mobile Applications

Statistically speaking, the market for Mobile Apps is in an incredible revenue rate, forecasting an almost trillion (935.2 billion) U.S. dollars in 2023, doubling, in 4 years, this year forecasted revenue of about 461.7 billion U.S. dollars, represented in figure 1:

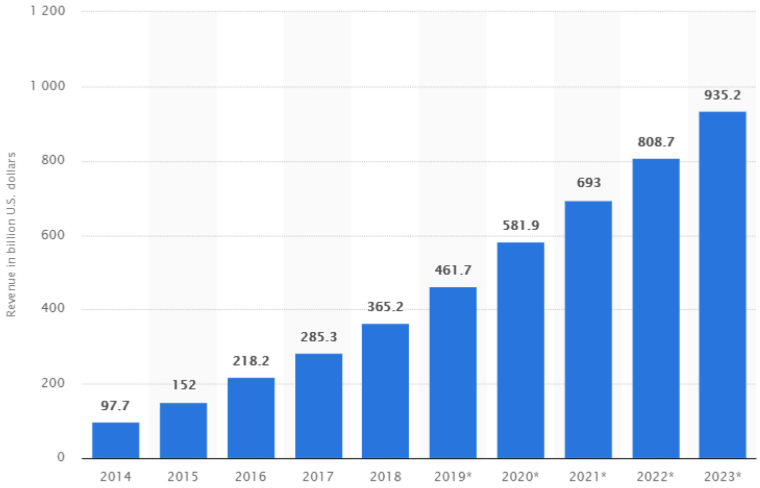


Figure 1-Worldwide mobile app revenues in 2014 to 2023 (in billion U.S. dollars). https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/ on June 8th, 2020

Consumers all over the world are also spending higher money figures as years have progressed, and we would have a forecasted total of 150 billion U.S. dollars spent by 2022, represented in figure 2:

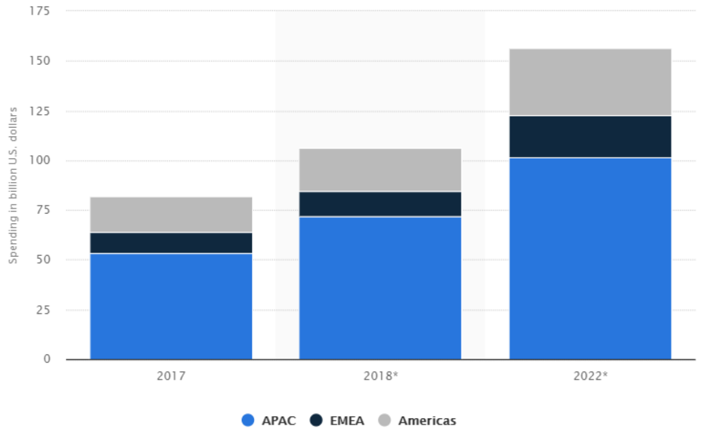


Figure 2-Worldwide consumer spending on mobile apps in 2017, 2018 and 2022, by region (in billion U.S. dollars). https://www.statista.com/statistics/695104/worldwide-mobile-app-consumer-spend-by-region/ on July 8th, 2020

Generating revenue is what drives businesses forward all over the world. Having that into consideration, knowing how many apps are being downloaded each year is also important to understand if the market is headed into the right direction. That being said, the download of apps continues to escalate, registering almost 200 billion downloads in 2019, a growth of 25% in just 3 years, represented by figure 3:

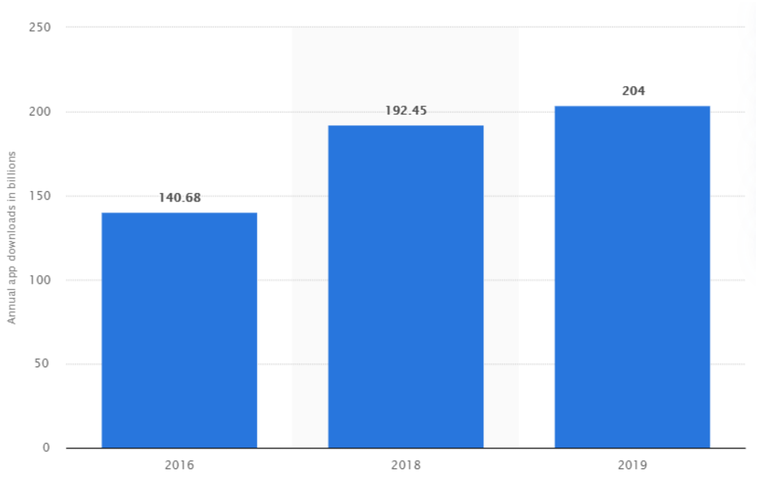


Figure 3-Number of mobile app downloads worldwide from 2016 to 2019 (in billions). https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/ on July 8th, 2020

2.3. State of the art for mobile application usability

Nielsen and Loranger (Nielsen & Loranger, 2006) describe usability is a quality attribute relating to how easy something is to use. More specifically, it refers to how quickly people can learn to use something, how efficient they are while using it, how memorable it is, how error-prone it is, and how much users like using it. If people can't or won't use a feature, it might as well not exist.

Mobile devices and their applications provide significant advantages to their users, not only in terms of portability but also in accessibility and location awareness. Through the years there has been a strong development in the hardware and software which led to a significant expansion of the mobile and mobile related markets. Therefore, leading into huge number of mobile applications being idealised and developed over the past years (Nayebi, Abran, & Deshamais, 2012).

This vast and increasing number of mobile apps has challenged developers to up the quality of their developed apps in order to compete. There are many important aspects to the quality of mobile apps, one being usability. Furthermore, the architecture of these applications must consider a number of design constraints, such as limited resources, connectivity issues, data entry models, and varying display of resolutions of mobile devices (Nayebi, Abran, & Deshamais, 2012).

Nayebi and colleagues (Nayebi, Abran, & Deshamais, 2012). also mentions that, in the past, the usability of software systems was evaluated subjectively, and therefore improperly defined. Researchers would select the aspects of usability to evaluate and measure what they considered important. At the same time, usability measurement and analysis methods and methodologies were being developed. Lab experiments, field studies, and hands-on measurement are some of methodologies most often applied by researchers.

ISO 9241-11 (ISO, 2020) defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use". With the fast arrival and rapid deployment of mobile technologies, a number of additional studies have focused on the usability of mobile devices (Ryu, 2005; Gafini, 2009): "Problems caused by physical restrictions of mobile devices and wireless networks imply that while designing and conducting usability studies for mobile applications, these issues must be carefully examined in order to select an appropriate research methodology and minimize the potential effect of contextual factors on perceived usability when they are not the focus of studies" (Zhang & Adipat, 2005). For Zhang et al. mobile usability includes some of the new mobility-related challenges we often come across: Mobile Context, connectivity, small screen size, different display resolutions, limited processing capability and power and Data Entry Methods.

Nayebi et al. states that a field study methodology - which is a general method for collecting data about users, user needs, and product requirements that involves observation and interviews, is required. This means using a questionnaire form that considers "new mobile operating system needs such as interaction with a multi-touch screen, displays of different resolutions and dimensions, device orientation changes, and gestures like tap, flick, and pinch". It is also stated that a hands-on measurement methodology is needed so that it can be applied to the same application development process as considered in the field study method (Hussain & Kutar, 2009). Hussain et al. defines a usability metric framework for mobile phone apps and uses the Goal Question Metric approach to link usability goals, such as simplicity, accuracy, and safety, to questions and related metrics.

2.4. Mobile Application Accessibility

Worldwide, there are more than 750 million people with disabilities. As move towards a highly connected world, it is critical that the web be usable by anyone, regardless of individual capabilities and disabilities – accessibility – including the deaf, blind, physically challenged, and cognitive or visually impaired (Paciello, 2000).

Today's world is surrounded by information as it is more available to us that it has ever been in our human history. Despite smartphones having a higher reach every day that passes, the challenge of making it accessible and usable by all users, despite what their limitations may be, is still one difficult to overcome. It is not limited to the big companies that produce mobile devices, but it also matters to all the actors in the chain, from the content producer to the final user, network providers, service deliverer, software developers and even search-engine titans (Billi, et al.).

Accessibility has been a great problem solver for people with disabilities, but those are not the only group that could be at risk of being excluded from the use of Information Technologies: the elderly. As people age, some problems start to appear and the decline of a wide variety of abilities (vision, hearing, mobility and cognition) starts to take place, impacting on many aspects of their normal life.

The evolution of mobile technologies has caused an enormous social change. Although, this change has not been possible or available for all groups. Disabled people and elderly people experience several difficulties such as:

- Devices were not properly designed for older people;
- Developers didn't address accessibility issues on their application designs and so on.

In (Abascal & Civit, 2001), Abascal et al. defines the requirements that mobile communication systems for disabled and older people should meet can be classified as the following categories:

- Personal communication: for users with restricted movement, mobile technology enhances their chances of personal communication.
- Security: situations of illness, home accidents and so on, require a quick communication channel.
- Social Integration, Access to education and labour market: Services like tele-working and tele-education contributes to social inclusion and autonomy of user with disabilities.

- Autonomy: the combination of personal communication, security and access to integrative services gives to people with disabilities and older people more opportunities to carry out an independent way of life.

As seen, addressing accessibility could improve the quality of life of not only the disabled but also our elderly.

2.5. Mobile Application Accessibility Guidelines

There are several known resources for Mobile Accessibility Guidelines (for both Native Apps and Web-Apps), some of them are generic resources and others are platform specific (Android, iOS, etc).

2.5.1. World Wide Web Consortium (W3C) guidelines

Beginning with the launch of the Web Accessibility Initiative (WAI) in April 1997, it became clear that build and redesigning the Web to be accessible to people with disabilities would become an important directive of the World Wide Web Consortium (W3C) (Paciello, 2000).

The W3C collects a set of barriers common to mobile device users and people with disabilities that are focused in four different contexts that deserve accessibility considerations (Patch, Spellman, & Wahlbin, 2020):

- Perceivable – small screen size, zoom/magnification and contrast.
- Operable – keyboard control for touchscreen devices, touch target size and spacing, touchscreen gestures, device manipulation gestures and placing buttons where they are easy to access;
- Understandable – changing screen orientation, consistent layout, positioning important page elements before the page scroll, grouping operable elements that perform the same action, provide clear indication that elements are actionable and provide instructions for custom touchscreen and device manipulation gestures.
- Robust – set the virtual keyboard to the type of data entry required, provide easy methods for data entry and support the characteristics properties of the platform.

2.5.2. Android Accessibility Practices

For the android operating system, Google has made available some guidelines to help developers with their practices towards building better apps for everyone and improve their app’s accessibility (Google, 2020). These guidelines are as it follows:

- Follow the practices recommended by Material Design, which allows every user, including the ones that may have some disability, to navigate and interact more easily with the app;

- While creating the app, follow the basic principles of accessibility that make a big difference for your users, like content classification, a large target area, the contrast of colours and the visualization attributes.
- Integrate accessibility in your testing cycle, incorporating manual and automatic accessibility tests.

When accessing the Google developer specific documentation for app accessibility, Google offers three main bullet points to improve accessibility: increase text visibility, use large, simple controls and describe each UI element (Google, 2020).

2.5.3. iOS accessibility Practices

Apple has also made available, through their developer centric website, their vision of a more Inclusive Design, which gives more people the opportunity to enjoy your app by ensuring that everyone can use and understand it (Apple, 2020). Apple considers the three following best practices to be taken into consideration when creating a more inclusive application:

- Design with Accessibility in Mind – through enabling familiar, consistent interactions that make it simple and straightforward to perform any kind of task, complex or not – Simplicity – and through making sure that all content can be perceived whether people are using sight, hearing, or touch – Perceivability.
- Support Personalization – by using standard controls during the implantation of your app's UI. These allow text and interface elements to automatically adapt to several accessibility preferences, such as Bold Text, Larger Text, Invert Colours, and Increase Contrast.
- Audit and Test your app for Accessibility – When auditing, it examines every element in your app and give you a comprehensive list of issues to fix. Trough testing with accessibility features turned on, it helps you ensure that all users can complete the most important tasks in your app, no matter how they interact with their devices.

2.6. UX on mobile applications

Mobile applications are a considered to be a specific domain when talking about user experience (UX). The user satisfaction and its value towards an app is critical for the mobile app value chain, which is why it is understanding to develop apps in a way that accentuates the user experience (Barnes, 2002).

One of the most important design aspects during the implementation of a mobile application is the consistency of user experience. It can be affected by a variety of design choices taken during the development of applications. Taking the users into account and understanding the technical limitations and opportunities associated with the application in hands is mandatory (Kuusinen & Mikkonen, 2014).

A common problem in designing these apps is that users want their actions rapid and focused, instead of what they are used to on computers where can maybe take a higher percentage of time to execute a similar action. Therefore, actions must be simple and single yet focused, and must be accomplished with ease by using a minimal number of keystrokes. A generic workflow has been defined for the development of mobile applications (Kuusinen & Mikkonen, 2014):

1) *Scoping*: Before starting the design of a mobile application, the main goal of the application must be defined. Both what the application can or cannot do.

2) *Performance considerations*: responsiveness on mobile applications is essential for its user's experience, so assuring it throughout the app is required. Specific metrics should be created for the most important user scenarios.
3) *User interface design*: It is important to study key use cases and features that characterize the application before the technical design. End-user productivity – common actions easily and rapidly carried out – and responsiveness – feeling of control by the user – are considered as the most important principles of user interface (UI).
4) *Data model and memory concerns*: The way data is shown to the user has an impact on how it is located in the memory, how the system behaves in peak conditions and how the application disposes data. This may affect the UX – for example, operations that need extensive computations should never freeze the UI.
5) *Communications and I/O*: How the application handles local and remote resources has a major effect on UX. The way local resources are usually accessed is faster that communicating with a remote resource.

2.7. Healthcare mobile applications

Currently, there is an unprecedented level of disruption for healthcare providers as sixty-three percent of have faced severe due to reasons including internal organizational changes, cost pressure, regulation and compliance, funding or shifting consumer demands. In a study made by Gartner, only fit organizations can thrive and fragile one's collapse (Gartner, 2020).

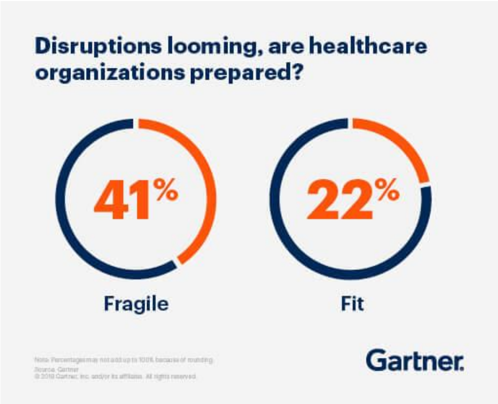


Figure 4- Percentage of collapse by fragile and fit organizations. https://www.gartner.com/en/industries/healthcare-providers-digital-transformation on July 8th, 2020

Digital and innovation initiatives are being made by the best healthcare companies to enable new business models, address the challenges of increasing demand and escalating costs. Digital transformation in healthcare is the positive impact of technology in healthcare. Technologies like artificial intelligence (AI)-enabled medical devices, blockchain electronic health records and telemedicine are just a few concrete examples of digital transformation in healthcare (Reddy, 2020). As Reddy also mentions, innovation is the key, with the main goal of streamlining physicians' work, optimizing systems, improving patient outcomes, reducing human error, and lowering costs through better web and mobile experiences.

The use of mobile phone and other technologies in the medical industry is also referred as mHealth (mobile health). The current primary uses are to inform patients and healthcare professionals about preventive healthcare measures, treatment support, tracking current progress and assistance in clinical trials (Bora, 2020). Both consumers and providers show increasing interest in healthcare apps, Bora also mentions, with as much as 71% of patients preferring their doctors using healthcare mobile apps and 70% of medical school students reported using at least one medical app regularly (Ventola, 2020).

The industry of digital health market is on an upward trend having reached over USD 106 million in 2019, with an estimated growth being calculated at 28,5% CAGR (Compound Annual Growth Rate) through 2026 (Ugalmugle & Swain, 2020).

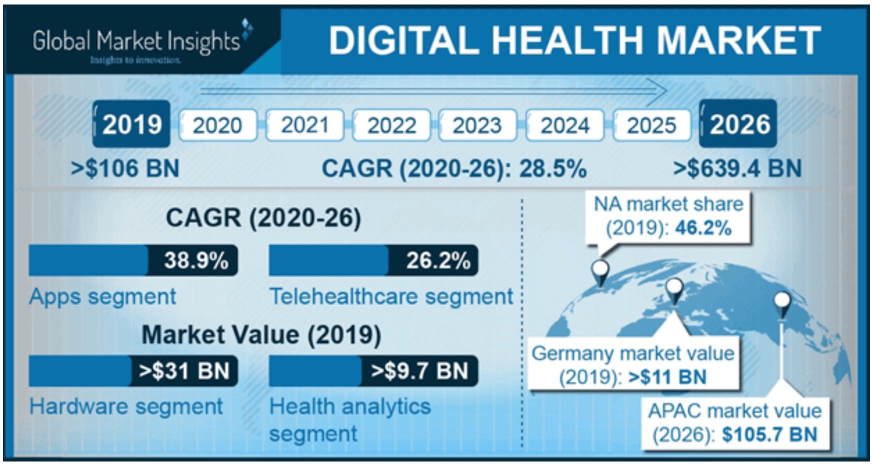


Figure 5 - Digital Health Market Industry trend from 2019 through 2026. https://www.gminsights.com/industry-analysis/digital-health-market on July 8th, 2020

We can also differentiate between healthcare apps used both by professional and patients with professional apps being used for medical instruction, access to healthcare records and diagnoses, prescriptions for medicine, and other internal matters (Lastovetska, 2020).

Professional healthcare apps can be divided in these types:

- Healthcare reference and database apps
- Professional networking apps
- Patient health tracking apps
- Doctor appointment and clinical assistance apps
- Telehealth mobile apps (doctor-on-demand apps)

Patient focused mobile health apps can also be divided in these types:

- Patient healthcare education apps (patient portals)
- Reminder apps
- Diagnosis apps

- Healthy lifestyle apps
- Monitoring apps
- Mental health apps
- Dieting apps
- Women’s health apps

2.8. Healthcare mobile applications UX

Taking UX in consideration is especially important with healthcare mobile applications as there may be patients with possible impairments that limit navigation. And with patients being the primary focus, there may be more users with these impairments than with other types of applications.

A healthcare mobile app can sometimes be confusing to the user when in need of an answer. Therefore, giving detailed and customised feedback is a great way to address that. Despite having the need to detail the information, keeping the interface clean with excellent legibility does also have a positive on UX, making it easier to navigate through the app (Gomes, 2020).

2.9. The concept of an Expert System

A study made by Truven Health Analytics, an IBM Watson Health company, revealed that 71% of visits to the emergency department in hospitals are unnecessary, and therefore avoidable. From these 71%, 24 percent did not require immediate attention and 48 percent did, but could have been avoided with appropriate primary care. (Becker's, 2020) In a study published in July 24th, 2019 by the United States’ largest insurer, it is shown that these unnecessary hospital visits are costing the U.S. health care system an extra 32 billion dollars a year (Deam, 2020).

With proper and easy access to basic healthcare information, these circumstances could be prevented. Giving people the option to quickly search possible symptoms and solutions to those symptoms could lower the high percentage of unnecessary hospital visits. Having that knowledge available to all is important but it’s also important that it is relevant and correct. The best way to guarantee it, is to gather knowledge from experts of the field in question.

Expert systems (ES) are a branch of applied artificial intelligence (AI). The basic idea behind ES is simply that expertise, which is the vast body of task-specific knowledge, is transferred from a human to a computer. This knowledge would then be stored in the computer and users can call upon the computer for specific advice as needed. This means the computer can make inferences and arrive at a specific conclusion, give advices and explanations, if necessary, like a human consultant. (Liao, 2004).

An expert system is divided into two subsystems, the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts and deduces new facts (Wikipedia, 2020).

2.10. Expert Systems in Healthcare

Expert systems have been used in a variety of fields, including medicine, space and business. (McCauley, N., & Ala, 1992; Johnston, Langton, Haynes, & Mathieu, 1994; Velicer & Prochaska, 1999; Leonard, 1995; Combining real-time monitoring and knowledge-based, 1993). For example, in an article published in 2012, the expert system would be used to automatically generate suitable patient instructions. This would mean the ES would be essentially disparate from existing clinical decision support tools available to healthcare providers for diagnosis and management because it would act independently of the clinician with automated alerts being sent directly to the patient and to the clinician. (Seto, et al., 2012)

Ultimately, the trial conducted with the ES mentioned would associate the developed rule set with improved clinical management, self-care, and quality of life. Furthermore, opening up the possibility to apply the lessons learned and development process to create rule sets for other clinical settings and for a range of other health applications (Seto, et al., 2012).

3. Information Architecture

One of the main problems that need to be tackled is misinformation and after understanding the state of the art for usability and accessibility in mobile applications it is also imperative being able to understand how users use and navigate these applications.

As mentioned previously, an abundant amount of information is available to everyone today. With the use of smartphones, smart bands, smart watches, tablets and other devices that use internet, users are provided with new and faster ways of interacting with information.

This abundance and diffusion of information make our lives easier and better in many aspects but could also introduce new challenges. With so much information available, sometimes it is hard to focus and find the necessary information and understand it when found.

The Information Architecture (IA) is a design discipline that is focused in making the information accessible and comprehensible. The IA enables us to think about problems in two important perspectives:

- Information about products and services are perceived by people as information sites;
- These environments of information can be organized for better visibility and comprehensibility.

IA has no official multifunctional definition that it stands by, but it can be defined by, the structural design of environments with shared information or, the art and science of shaping information and experiences about products which support usability, searchability and comprehensibility.

According to Peter Morville, the purpose of IA is to help users understand where they are, what they have found, what is around, and what to expect. As a result, your IA informs the content strategy – focus on the planning, creation, delivery, and governance of content – through identifying word choice as well as informing user interface design and interaction design through playing a role in the wireframing and prototyping processes (usability.gov, 2020).

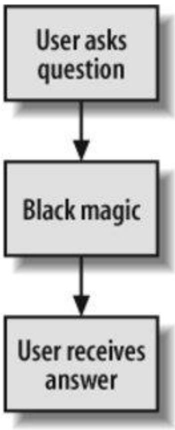
Information architecture starts with users and the reason they come to a site in the first place: they have an information need. Information needs can vary widely, and each type of information need causes users to exhibit specific information-seeking behaviours. Those needs and behaviours have to be understood, and designs should correspond accordingly. There is no goal more important to designing information architecture that to satisfy user' needs (Morville & Rosenfeld, 2006).

Seeking something you know is there, like your colleague's phone number, is quite a different information need that learning about a topic, like small-cap mutual funds, and your site's information architecture should be designed with those differences in mind. These needs are examples of information-seeking behaviours and, not surprisingly, searching for something you know is a very different behaviour than browsing for the unknown. Distinguishing between these needs and behaviours and determining which are your users' highest priorities is an extremely valuable pursuit helps you determine where to invest your efforts, resources, time, and money as you design your architecture.

3.1. The importance of User Needs and Behaviours

There are different models of what happens when users look for information. Modelling User's needs and behaviours forces us to ask useful questions about what kind of information the user wants, how much information is enough, and how the user truly interacts with the architecture.

The "too-simple" is the most common information model, and it's also the most problematic:



Or, expressed as a simple algorithm:

1. User asks a question;
2. Something happens (i.e., searching or browsing);
3. User receives the answer;
4. Fin.

This can be problematic, because it rarely happens this way. There are exceptions – for example, when users know what they're looking for, they know where to find the answer, they know how to stat the question, and they know how to use the site to do so. But users don't always know exactly what they want.

This model can also be problematic because it narrowly focuses on what happens while the user is interacting with the information architecture. By oversimplifying, this model cedes so many great opportunities to understand what goes on in users' heads and observe the richness of what happens during their interactions with an information architecture.

When a user comes to a website to find something, what does she really want? In the "too-simple" model, she wants the "right answer" to her question.

Although, websites store much more than highly structure data. Not surprisingly, text is the most common type of data stored, and text itself is made up of ambiguous, messy ideas and concepts.

There can be defined four different information needs:

- When you are hoping to make a perfect search, you usually know what you're looking for, what to call it, and where you'll find it – this is called *known-item seeking*;
- When you are hoping to find a few useful items in your searches, you are doing something called *exploratory seeking*. In this case, the user is not exactly sure what he's looking for. In fact, whether he realizes it or not, he is looking to learn something from the process of searching and browsing.
- When you want everything, you are performing *exhaustive research*. The user is looking for everything on a particular topic, hoping to leave no stone unturned. In this case the user often has many ways to express what she's looking for and may have the patience to construct her search using all those varied terms.
- Finally, our failing memories and busy schedules continually force us to engage in *re-finding* a piece of useful information that we've happened upon before. For example, if you find something at work, you might not read it at work, you'll *re-find* it later instead.

These are not the only information needs available but represent a large portion of users which fall into.

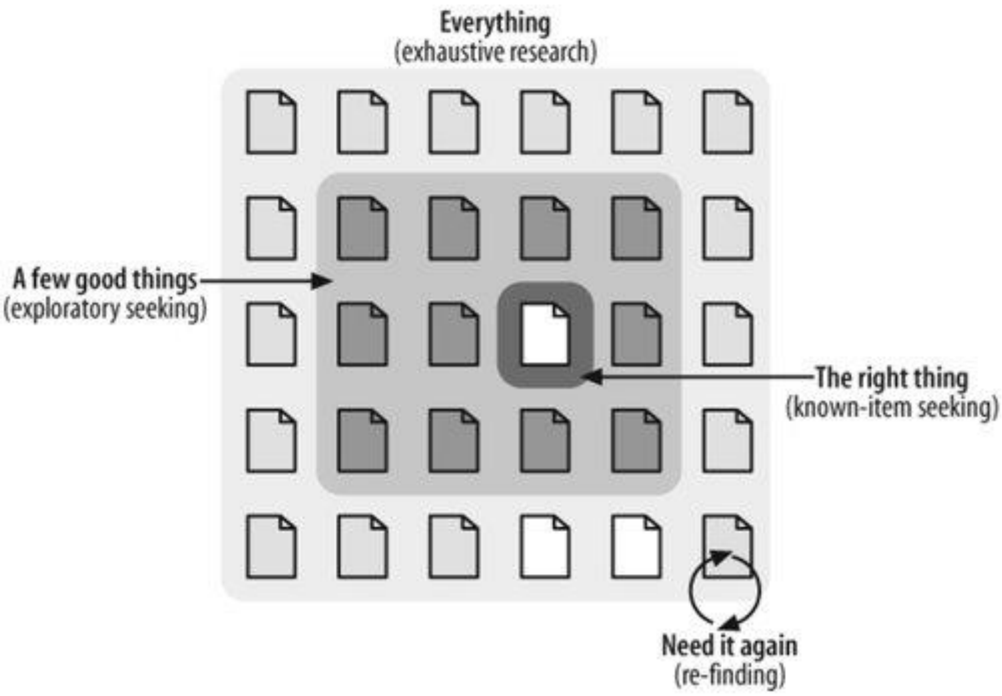


Figure 6- Four common information needs

And how do website users find information? They enter queries in search systems, browse from link to link, and ask humans for help (through email, chat interfaces, and so forth). Searching, browsing, and asking are all methods for finding, and are the basic building blocks of information-seeking behaviour.

Petter Morville and Louis Rosenfeld recommend, when learning about users' information needs and seeking behaviours, using a pair of research methods:

- **Search analytics** – involves reviewing the most common search queries on your site (usually stored in your engines logfiles) as a way of diagnosing problems with search performance, metadata, navigation, and content. This provides a sense of what users usually seek, and may help inform your understanding of their information needs and seeking behaviours;
- **Contextual inquiry** – a user's research method with roots in ethnography, is a great complement to search analytics because it allows you to observe how users interact with information in their "natural" settings and, in that context, ask them why they're doing what they're doing.

Other research methods that can be used are task analysis, surveys, and, with proper care, focus groups.

3.2. Understanding Information

Morville and Rosenfeld (Morville & Rosenfeld, 2006) also say the structure of information environments influences more than the way we find things, it also changes how we understand them. How we understand an information environment – the context in which information is inserted – also influences how we find information and vice-versa.

The Information Architects must create usable and comprehensible environments by human beings, where they can grow and adapt over time, to attend to their user and organizational needs.

3.3. Basic Principals of Information Architecture

It is important being able to visualize the IA but, paradoxically, a good projected Information Architecture is invisible to the users.

When we analyse a front page of a website, we observe many aspects of design, like colours, font types, photos, but we can also see how many columns there are and their width that may or may not change while scrolling down the page. If searching carefully, we could also see design aspects related to the interaction of the website, like using a cursor on the main menu options.

The information on websites is presented to us in various ways:

- **Navigation systems** – help users to go through the content, with a custom organization. There are usually represented by the individual and suspended menus on the main navigation bar.

- **Search systems** – allow users to search for content. Usually, when a user starts typing on a search bar, a list of suggestions is shown with a possible list of options related to the term searched.
- **Labelling systems** – describe categories, options and links in languages that are significant to the users.

3.3.1. "Top-down" – Information Architecture

Categories are used to group pages and applications on the whole website. Labels represent systematically the content of the website and the navigation systems as well as the search systems can be used to go through the site.

The main pages must anticipate the most common information needs of their user base. The users must determine the most common questions and project on the website, attending to those needs.

The most common questions the users have when using a "top-down" information architecture is:

1. Where am I? (1)
2. I know what I am looking how, how do I search for it? (2)
3. How do I navigate through this site? (3)
4. What is important and unique in this organization? (4)
5. What is available in this website? (5)
6. What is happening in this environment? (6)
7. How to I communicate with them through the various most popular digital channels? (7)
8. How do I enter in contact with a person? (8)
9. What is their address? (9)
10. How to access my account? (10)

3.3.2. "Bottom-up" – Information Architecture

When offering support to search and navigation, the inherent content structure allows that the answers to the questions of the users "arrive" to the surface.

This "bottom-up" architecture, content structure, sequence and marking help answer questions like:

- Where am I?
- What's here?
- Where can I go?

This architecture is suggested and inherent to the system content. It's important because users are more and more prone to ignore the "top-down" IA of the system. Nowadays, while searching for a site, users use tools that search the whole web, like a Google search, they click in advertisement and click links while they read content, or through social networks like Facebook or Twitter.

Once there, users jump to another relevant content without learning the "top-down" structure. A good information architecture is projected to anticipate this type of behaviour, the simple and practical "Navigation stress test" of Keith Instone is an optimal way to evaluate the "bottom-up" Information Architecture of a website (Instone, s.d.).

While answering the questions bellow, you can see if your website passes the navigation test. The process starts by picking a random low-level page, then printing it black and white, pretend to enter this website for the first time and the mark-up the piece of paper with what you think the answers are.

Navigation Question	**Mark Up on the Paper**
What is this page about?	Draw a rectangle around the title of the page or write it on the paper yourself
What site is this?	Circle the site name, or write it on the paper yourself
What are the major sections of this site?	Label with X
What major section is this page in?	Draw a triangle around the X
What is "up" 1 level from here?	Label with U
How do I get to the home page of this site?	Label with H
How do I get to the top of this section of the site?	Label with T
What does each group of links represent?	Circle the major groups of links and label. D: More details, sub-pages of this one N: Nearby pages, within same section as this page S: Pages on same site, but not as near O: Off-site pages

Table 1 Keith Instone's Navigation stress test questions

3.3.3. "Invisible" – Information Architecture

A common information architecture is often invisible (Morville & Rosenfeld, 2006). Morville and Rosenfeld describe that in a general manner, every search result is executed by a software – a search engine – that the users never see. This search engine is configured to index and search certain parts of the website, show particular types of information in each result and manipulate the searches in determined ways, like removing stop words (i.e., "a", "the", "is", "at").

Every decision made related to the configuration of this search system is unknown by the users and are integral aspects to the architecture design. Information Architecture is much more that simple plants that represent navigation routes and wireframes that inform about the visual design.

3.3.4. Information Architecture Components

It is hard to know what components make an information architecture. Users may interact directly with some of those components, but they often do not know the existence of others.

The information architecture components mentioned previously arise in the following four categories:

1. Organization Systems – How the information is organized (i.e., by topic or chronologically);
2. Labelling Systems – How the information is labelled (i.e., using scientific terminology);
3. Navigation Systems – How to navigation and move to obtain information (i.e., clicking through a hierarchy);
4. Search systems – How to search information (i.e., executing a search query on and index).

4. Analysis of Similar Apps

Through a search on the two main app stores (Google Play, and App Store), four apps were found to be providing the same services/issues discussed in the present document. These apps were recommended for analysis by an expert. By analysing and studying these different apps, it represents a steppingstone for making a better and more suitable mobile app for the public and specifically for patients and their families. This analysis was made in May, 2020.

In this chapter, a high-level analysis will be made for these different mobile applications:

- PatientAider
- AHRQuestionBuilder
- TeamSTEPPS 2.0
- Healthily

The main goal is to explore and analyse three main topics discussed in this document, accessibility, UX and UI.

4.1. PatientAider

The PatientAider is an informational mobile application meant to educate and inform a patient and family members/friends during and hospital stay. As it is state and as this is an informational app, it is not meant to substitute the professional advice from a medical professional.

This application is presented in a simple way to the user, firstly asking where the user/patient physically is in order to curate the information and topics available to whichever option the user selects.

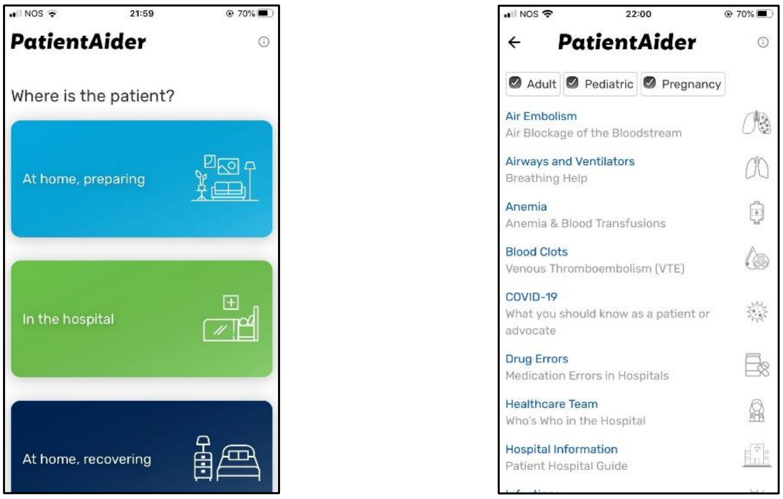


Figure 7 PatientAider Navigation

When the user selects an option, a vast page of with information about it is loaded, even making it possible to view videos related to the topic.

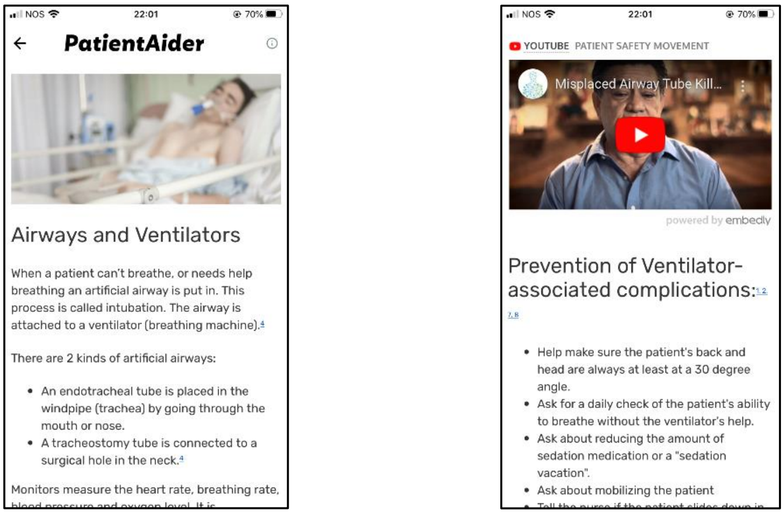


Figure 8 PatientAider page content

Despite being clearly a light application, it still has some flaws and bugs, not only concerning the general state of the app development and usability, but it also raises concerns regarding accessibility:

- The app information/feedback button on the top right of the application is very small and therefore hard to press.
- After pressing one of the initial three buttons which are clear and big (making it easy to press), it's only possible to select a topic by pressing the text, which makes it hard to reach. There is a misuse of phone real estate that could help this issue, as show on figure 9.
- There is no information about what topic you have selected, or where your patient is. This can lead to easy loss of track while scrolling through a topic page.
- Due to lack of general topics, the ones presented to the user vary little depending on which option the users selects in main page. Filtering the topics also seem to have no impact.
- While scrolling through a topic, if you want to go back in the app, you must scroll back up in order to press the corresponding button. Also, the phone's "back" button does not seem to have any effect on the application.

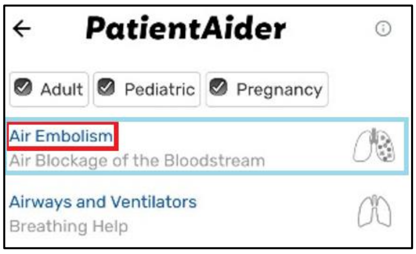


Figure 9 PatientAider's bad usability

4.2. AHRQuestionBuilder

This mobile application is purposed to prepare the user to his/hers next medical appointment. It provides the tools to set a date and time, information about the healthcare provider and finally prepare a set of possible questions the user might want to ask during the appointment. There are also two other tabs which the user can access to watch videos or access resources.

The app home page is used to prepare the medical visit, where the user can fill all the information necessary.

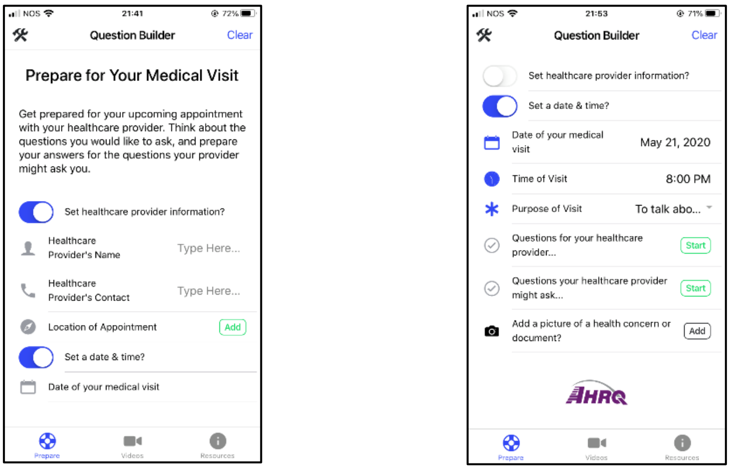
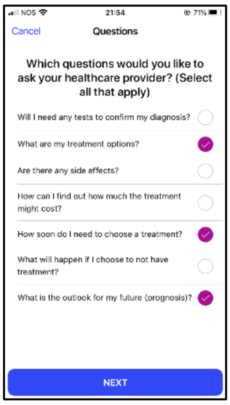


Figure 10 AHRQuestionBuilder prepare hospital visit

A purpose of visit option is made available to the user where a template of possible questions is made available depending on what the user wants to address at the appointment, even allowing custom questions to be written. It is also possible to prepare a set of responses to the questions that may be asked.



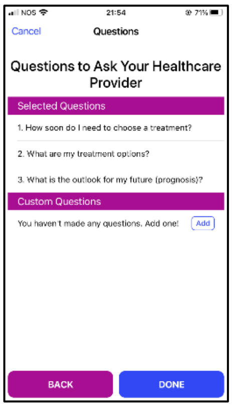


Figure 11 AHRQuestionBuilder question preparation

When the user fills up what he feels it's necessary, the application provides a simple review page where everything is written down on a blank page, which can be emailed or added to the calendar.

Overall, the application does not appear to have clear usability flaws as major tasks can be done with little to no problem and a feedback button is provided in case something unusual arises. Although, some changes could be done to improve the overall usability and accessibility of the application:

- This is clearly an English based application, limiting the user if he does not know the language.
- Although 2 additional tabs with videos and resources are provided, there is no easy way to filter them. The overall content is very limited.

4.3. TeamSTEPPS 2.0

This app is overall like PatientAider, but instead focused on the medical staff. In TeamSTEPPS there are numerous guidelines, best practices, and other information to help the medical staff in different areas like team structure, communication, leadership, situation monitoring, mutual support, etc.

Navigating through the app is easy, as in how you navigate it. There is a main page, with two available buttons, one to open the menu and the other to jump straight to the first content page. Then, if you want to go to another page, either you open the menu again to choose a different page or you press the bottom right button, which depends on the context of the page.

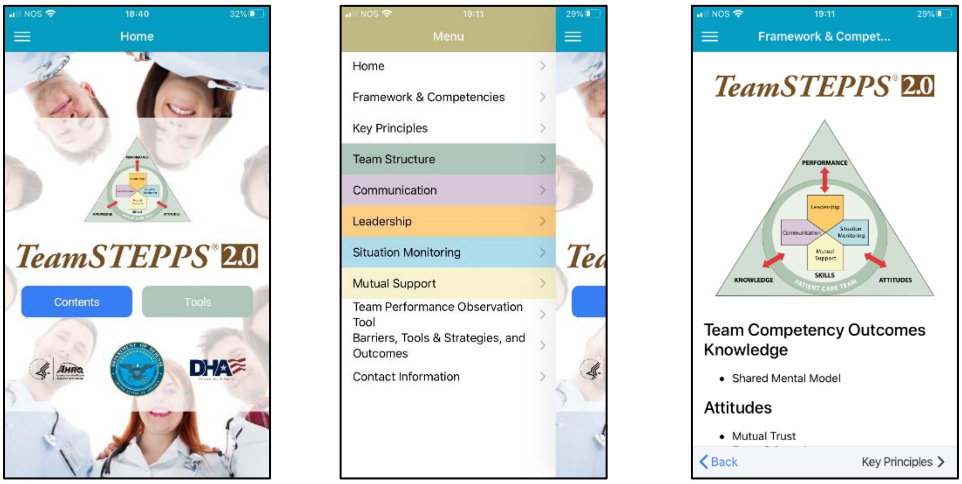


Figure 12 TeamSTEPPS 2.0 Navigation

TeamSTEPPS is not only like PatientAider in terms of the informational nature of the app, but it also lacks in the overall usability and accessibility of the application:

- The entire app is simple in navigation almost feeling like navigation through a slide show, which may be intended, but leaving the impression as if this app was made for conference purposes.
- There are pages with options which can be selected but when navigating, those options can be lost.
- Navigation also seems like a rainbow show, with some pages having a poor background colour, making it hard to read the text.
- Another English based application, which can't be adapted to other countries and hospitals who can benefit from the information.

4.4. Healthily

Healthily is a more complete mobile application than the others mentioned above. This is a selfcare app that allows users to check trusted information, take notes, set trackers and quick check symptoms.

The different ways to interact with the app are well laid out. You have two main areas, one where you can ask Healthily almost everything or check symptoms, and other one that is your personal journal where you can take daily notes about your health, both mentally and physically, and keep your tracker updated.

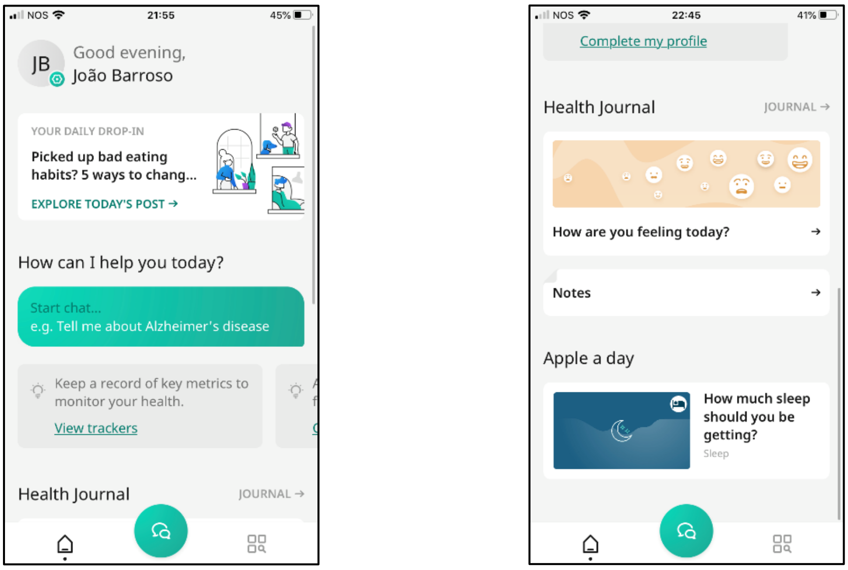


Figure 13 Healthily navigation and content

Healthily UX and accessibility is far richer that the other mobile apps previously mentioned. The text is generally big and easily readable. Some content can be viewed on the web and if the user wants to get a better view, the option is also available.

Figure 14 Healthily – web content accessibility options

4.5. Comparative Board

The analysis done for the apps can now be made into a comparative board. This way the most crucial points can be addressed and compared.

For the comparative board there will be indicators that each app will be classified for, therefore making it possible to rank the mobile applications. These indicators must take into consideration the different areas mentioned already in this article that directly help either medical staff or patients. The method of classification is based on if the application is built with features that handle well the different indicators, divided into two different classifications, “Yes” and “No”.

Indicators	PatientAider	AHRQuestionBuilder	TeamSTEPPS 2.0	Healthily
UX	No	Yes	No	Yes
UI	No	Yes	No	Yes
Accessibility	No	Yes	No	Yes
AI	No	No	No	Yes

Table 2 Similar Apps Comparative Board

Unfortunately, most of the apps suffer from a lack of accessibility and UX/UI features. As previously mentioned, these are essential features that will be addressed in the mobile application that is going to be built.

The healthily app, on the other hand, has a perfect score of features, and this shows a great interest for the general user experience on the app. As AI features are also present, this makes it the most complete app of the ones addressed and a great point of reference.

4.5.1. In-depth analysis of the comparative board

For UX/UI, PatientAider and TeamSTEPPS struggle which sets them apart from the rest of the apps.

PatientAider has several issues such as small areas – text line – for users to click, or when on a large page of content, needing to go back up in order to back out of the page is clearly not thought out, and degrades user experience.

TeamSTEPPS is like a colourful slide show. Having multiple many colours on the pages, and some of them making it difficult to read the content. There are pages where the user can select some options, but when navigating to other pages, the options selected reset. It is a bug, and it is bad for user experience.

AHRQuestionBuilder and Healthily, however, provide clear actions the user can make with big buttons and text that is easily read.

Accessibility wise, AHRQuestionBuilder and Healthily the only apps that demonstrate concern and implement features to accommodate accessibility, with the others having the following issues:

- PatientAider – very small buttons to press, low contrast text with background.
- TeamSTEPPS – low contrast between content and background on some pages

Regarding artificial intelligence, as previously mentioned, Healthily offers an AI feature, which stands out as the only app to do so. This feature allows the user to ask several different things that can range from symptom checking, health tests or asking a simple question.

5. Expert System

Providing users valid information is the core for resolving the main issue of misinformation. In this chapter expert systems will be addressed, demonstrating how they can gather from experts and serve users the most reliable information.

5.1. Architecture of knowledge-based expert systems

A knowledge base ES contains domain-specific and high-quality knowledge. For any ES to have success, it majorly depends upon the collection of its highly accurate and precise knowledge. This knowledge is required to exhibit intelligence. (tutorialspoint, 2020)

As it is mentioned above, knowledge-based expert systems (KB-ES) receives expert knowledge coded into facts, rules, heuristics, and procedures. A general architecture of KB-ES is show in Figure 7. The database possesses information in terms of fact or heuristics based on user interest of specific problem domain. Knowledge base possesses domain knowledge expressed in terms of mathematical logic. Accordingly, it is classified as a Rule-Based System (RBS) based on the “if-then” else condition and will execute certain tasks, once the condition is satisfied. (Kumar, 2017)

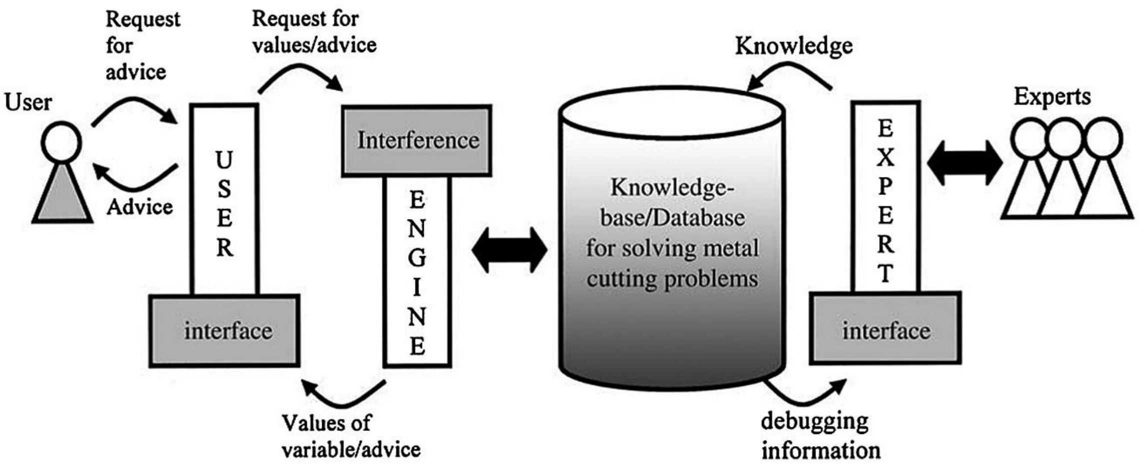


Figure 15 – An example of a knowledge-based expert system *(Cakir & Cavdar, 2006)*

Inference engine acts as a regulating environment and interacts with the user. It receives input from the user about the problem and takes additional information as desired. It understands the knowledge base to formulate inferences and provides expert advice.

5.2. Collecting and Presenting knowledge

Knowledge is a theoretical or practical understanding of a subject or domain. Knowledge is also the sum of what is currently known. Those who possess knowledge are called experts. (Hussein, 2020)

Anyone can be considered a domain expert if he/she has deep knowledge (of both fact and rules) and strong practical experience in a particular domain. The area of the domain can be limited and generally, an expert is a skilful person who can do things other people cannot.

There are several techniques/models for knowledge representation such as rules, frames and semantic nets. The term rule in AI can be defined as an IF-THEN structure that release given information or facts in the IF part to some action in the THEN part. Any rule consists of two parts: the IF part, called the antecedent (premise or condition) and the THEN part called consequent (conclusion or action):

- IF <condition> THEN <action>
- A rule can have multiple antecedents joined by the keywords AND (conjunction), OR (disjunction) or a combination of both. For example, IF <condition1> AND <condution2> OR <condition3> THEN <action>

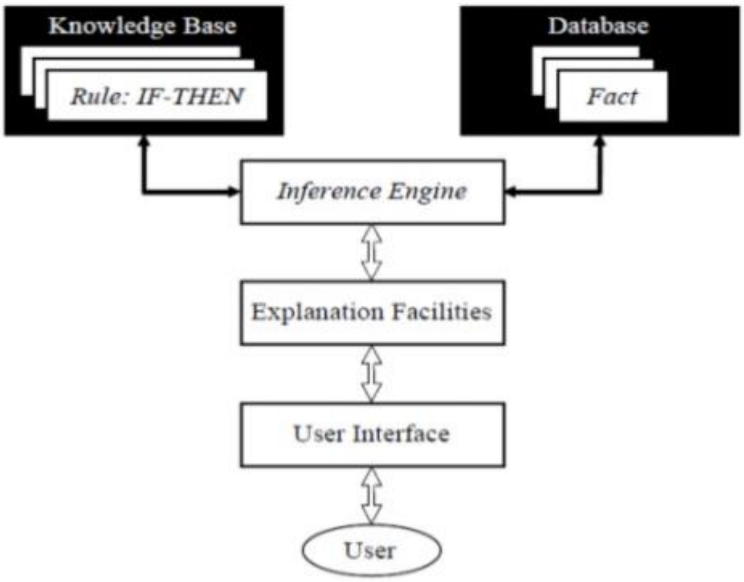


Figure 16 - Basic structure of a rule-base expert system

Frames are also useful for representing knowledge. As frames allow nodes to have structures they can be regarded as representations of knowledge. A frame is similar to a record structure and corresponding to the fields and values are slots and slot fillers.

Semantic net (or semantic network) is a knowledge representation technique used for propositional information. It is also called as propositional net. Semantic nets convey meaning.

Mathematically a semantic net can be defined as a labelled directed graph. Semantic nets consist of nodes, links (edges) and link labels.

5.3. Forward chaining and backward chaining

As previously mentioned, in rule-based expert systems, when the IF (condition) part of the rule matches a fact, the rule is fired and its THEN (action) part is executed. The matching of the rule IF parts to the facts produces inference chains. The inference chain indicates how an expert system applies the rules to reach a conclusion.

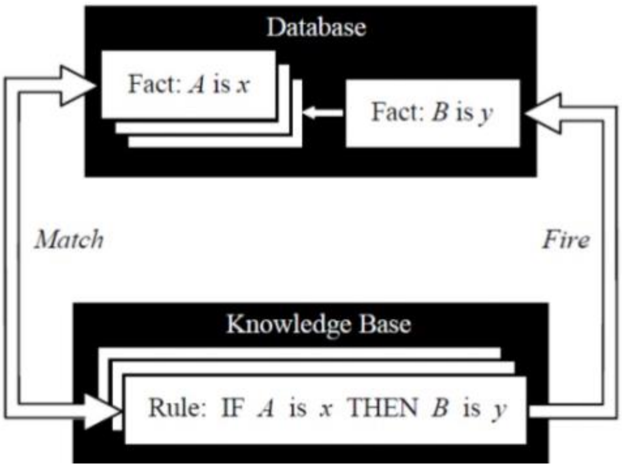


Figure 17- Inference engine cycle via a match-fire produce *(Hussein, 2020)*

Forward chaining is the data-driven reasoning (Hussein, 2020). It is also known as a forward deduction or forward reasoning method when using an inference engine. Forward chaining is a form of reasoning which start with atomic sentences in the knowledge and applies inference rules in the forward direction to extract more data until a goal is reached. The Forward-chaining algorithm starts from known facts, triggers all rules whose premises are satisfied, and add their conclusion to the known facts. This process is repeated until the problem is solved. (JavaTPoint, 2020)

Backward-chaining is also known as a backward deduction or backward reasoning method when using an inference engine. A backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts that support the goal – goal-driven reasoning (JavaTPoint, 2020).

5.4. Advantages and Disadvantages of rule-base expert systems

The use of rule-base expert systems has advantages such as (Hussein, 2020):

- Natural knowledge representation, as when an expert usually explains the problem-solving procedure with such expressions as this “In such situations, I do this”, these expressions can be represented naturally as IF-THEN production rules;
- It provides a uniform structure, as each rule is an independent piece of knowledge;
- Separation of knowledge from its processing. This makes it possible to develop different applications using the same expert system shell;
- Dealing with incomplete and uncertain knowledge, as most rule-based expert systems are capable of representing and reasoning with incomplete and uncertain knowledge.

Although, there are also several disadvantages:

- Opaque relations between rules. Although the individual production rules are relatively simple and self-documented, their logical interactions within the large set of rules may be opaque.
- Ineffective search strategy as expert systems with large set of rules (over one hundred rules) can be slow, and these large rule-based systems can be unsuitable for real-time applications.
- Inability to learn. Unlike a human expert, who knows when to adjust or adapt to situations, an expert system cannot automatically modify its knowledge base.

6. Conceptual basis

To provide a mobile application that is accessible and of easy use but also providing credible information to users is essential. In this chapter it will be demonstrated the concepts behind achieving such application.

A basic user workflow from the start of a symptom search to its conclusion can be divided into two main parts, the user search – user interface – and the system responses to symptoms/questions – knowledge representation.

6.1. Knowledge representation

As mentioned before, the knowledge must be gathered from domain expert(s), i.e. registered nurses, physicians or therapists. Without this knowledge there cannot be a viable expert system.

In the beginning of development of an application which encompasses a KB-ES, the knowledge required to be used by the system is non-existent. This represents a "cold-start" problem which is related to the sparsity of information available (Lika, Kolomvatsos, & Hadjiefthymiades, 2014). This means that even when considerable information has been collected, there are still possible scenarios where a user cannot get a viable response.

After collecting knowledge from experts as represented in Annex 1, the next step is rule definition. Rules link possible symptoms to responses, following the IF-THEN structure, i.e. IF user has headache THEN take aspirin. Through the user's inputs/data provided the system will determine the appropriate response based on the rules defined. By users giving information to achieve their goal, i.e. symptom search and possible treatment, the inference engine follows the forward-chaining method.

6.2. User Interface

It is important that the users are able to use the application in any situation. This means providing users with an accessible and easy to use application.

The best way to make an app accessible is to follow and apply WAI-ARIA (Accessible Rich Internet Application) technical specification. This provides a framework to improve the accessibility and interoperability of web content and applications (Diggs, McCarron, Cooper, Schwerdtfeger, & Craig, 2020). It is also important to follow accessibility guidelines of the application's intended operating system that can be either or both android and iOS (Google and Apple).

To develop an easy-to-use application is to develop with the user in mind, giving important to UX. As previously mentioned, users want their actions rapid and focused. This means actions must be simple and single yet focused, and easily accomplished with minimal number of touches.

7. Built Prototype

A prototype was built to demonstrate the use of a system to emulate a rule-based ES, as well as the use of accessibility options and focus on UX/UI this way making the application easy to use and inclusive to all.

7.1. Languages and Frameworks

With the mobile environment in mind and the inclusion of the two most prominent mobile operative systems currently in use, it was chosen to build the application's UI through Expo. This is both a framework and a platform for React applications. As it has a set of tools and services around React Native and native platforms, this means the development, building and deployment can be quickly iterated on iOS, Android and web applications from the same JavaScript/TypeScript code (Expo, 2020).

As mentioned before, it was opted to build a mobile app with React Native which is the most used cross-platform mobile framework with 42 percent of software developers adopting it on a worldwide scale, followed by a rising Flutter usage with 39 percent (Statista, 2020). By using React Native it is possible to build an application for many platforms as React primitives render to native platform UI. It also comes with ease of use if developers are already accustomed to React as there is little to no learning curve transitioning to this framework. The development of applications also benefits from JavaScript's fast refresh.

For the application's backend it was decided to be implemented in Flask, a lightweight WSGI (Web Server Gateway Interface) web application framework that uses Python language. It makes the starting process quick and easy, but it also offers the ability to scale up to complex applications. This is how the API was made to service the front end of the mobile application.

7.2. Interface

The prototypes' interface was made to comply with the main goals of being easy to use and accessible. There are two main screens and in these the user can experiment with the emulated rule-based ES and test accessibility options.

To make the prototype easy to use an emphasis was put on app legibility and making navigation actions simple with large texts, labels and buttons. There are four main screens:

- Homepage – represented by Annex 6.
- Symptom search screen – represented by Annex 7.
- Settings – represented by Annex 8.
- Rule Tree (admin screen) – represented by Annex 9.

Users can select, on the symptom search screen, the option to search for a disease based on selectable symptoms. The option to write and search for a symptom is also available. These symptoms are those gathered from the expert systems and populated into the rule-based system.

Regarding accessibility, several features were implemented that respect the W3C guidelines for accessibility, some of them even configurable by the user. There is an option that allows the user to change the main theme colour to their type of colour blindness and other option to invert colours to create more contrast. The code was also implemented with accessibility in mind as there are ARIA labels for every text, button or image that allow the user to navigate and better grasp the contents of the prototype.

7.3. Collecting knowledge from Experts

As mentioned previously there is a fourth section on the app that allows any authorized user, such as experts or admins, to access a visual representation of the rule-based expert system. The rules are represented by a tree using the JSON in which those are defined. This makes it quick and simple for the authorized user to add, remove, or edit any element of that JSON tree.

7.4. Emulated Rule-based Expert System

The prototypes' backbone is its emulated rule-based ES. For the prototype it was chosen to use two main topics, symptoms and diseases. Since symptoms can lead to diseases, we can simulate a rule-based system based on them, with the goal of suggesting possible diseases that they can lead to.

To represent these symptoms and diseases a database was created with five different symptoms and diseases. To link and associate each symptom to a possible disease, a rule tree was created, represented by the following example in annex 5.

When a user writes or selects a symptom, an API request is made that searches this rule tree, returning consequent possible symptoms the user might have or experience later. It also returns disease(s) associated with the symptom(s) the user selected. If there are no more symptoms or diseases associated with the symptom selected, the response simply returns an empty list.

Written searches are processed, using various methods and python functions, to find the symptom the user was looking for. First, the text is tokenized, then it is split and stop words – unwanted and irrelevant words - are removed. Then the commoner morphological and inflexional endings from the words are removed using the Porter Stemmer method. Next, the various inflicted forms of a word are grouped and analysed as a single item using lemmatization and lastly, searches the filtered words from the written search, return the symptom it recognized from the global symptom list

The API request and their logic functions are represented in annex 3 and annex 4.

8. Conclusion

To tackle the many different problems that come with health misinformation online it was proposed a mobile app for everyone that has a system providing reliable information.

For everyone means being accessible and easy to use and for this a study was conducted to understand what the app ecosystem is like and what we can take from other mobile applications to create something better.

Being a health expert, it is not a simple task, so it only made sense to gather health-based information from these professionals. It is reliable information from those who care for others. A knowledge-based expert system based on rules was adopted making the process of providing reliable information an easy task.

Overall, this study was made to make a mobile application that could help improve everyone's lives and a stepping-stone to that was reached.

A prototype was built to test the different necessary features this way providing an initial look of what a solution to solve the problems mentioned can be. However, it is only a prototype, not a deliverable application and a lot can still be done and improved in the future.

Throughout the studying, analysis and development process I was able not only to learn and study about interesting and engaging health-related topics but also improve on subjects and programming languages that were taught during my master's degree.

8.1. Future Work

The development of this prototype around health literacy and education opens the opportunity to develop a range of new features and/or enhance those already implemented.

Since this is an app made specially for everyone, user testing is mandatory with inquiries and usability testing. A large number of people need to be included in this testing. This testing user base needs to range from people with no physical conditions to people who display physical conditions.

On the topic of testing, load and stress testing also needs to be done to ensure users can reliably use the application.

By using web crawling and web scraping it is possible to fetch the most reliable information on the web, creating informational pages relative to diseases and symptoms and to worldwide news. This way we can deliver any amount of information to our users from trusted sources.

It would also be important to compare and study different results from the application and compare them with experts to see if they are reliable therefore trust proofing the app. Being able to compare results with experts also means having more complex predictions, therefore, enhancing the knowledge system with better algorithms is mandatory.

To enhance even further the knowledge system, a symptom relevance attribute could be added to the data collected. This means that when collecting data and creating rules, the experts could also indicate a level of relevancy for the possible subsequent symptoms and diseases. The most relevant symptoms and diseases would be presented to the user higher up the symptom/disease list.

Another way the app can be improved is by adding some sort of assistance or chat system that helps the user. This system can guide users around the app and help them executing the actions they want. Using the chat feature, and by connecting to a certified health professional, it could also provide assistance regarding any health-specific issue.

Furthermore, the ability to offer a vast number of accessibility options is also that needs to be taken into considerations as we want to bridge the gap to everyone who needs these options to have a greater experience using the application.

Additionally, implementing an authentication system is also important. This way the application can provide, for example, an history page based on the user's activity on the app and offer actions or solutions based on that history, offering a more personalized experience.

References

Abascal, J., & Civit, A. (2001). Mobile communication for older people: new opportunities for autonomous life. *Proceedings of EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly*.

Apple. (2020, July 10). *Best Practices for Inclusive Design*. Retrieved from developer.apple: https://developer.apple.com/design/human-interface-guidelines/accessibility/overview/best-practices/

Barnes, S. J. (2002). The mobile commerce value chain: Analysis and future developments. *International Journal of Information Management*, 91-108.

Becker's, H. R. (2020, 09 29). *Study: 71% of ED Visits Unnecessary, Avoidable*. Retrieved from Becker's Hospital Review: https://www.beckershospitalreview.com/patient-flow/study-71-of-ed-visits-unnecessary-avoidable.html

Billi, M., Burzagli, L., Catarci, T., Santucci, G., Bertini, E., Gabbanini, F., & Palchetti, E. (n.d.). A unified methodology for the evaluation of accessibility. *Univ Access Inf Soc*, pp. 337-356.

Bora, P. (2020, July 10). *The Ultimate Guide to App Development for Healthcare*. Retrieved from Digital Authority Partners: https://www.digitalauthority.me/resources/healthcare-app-development-ultimate-guide/

Cakir, M. C., & Cavdar, K. (2006). Development of a knowledge-based expert system for solving metal cutting problems. *Materials & Design*, pp. 1027-1034.

Combining real-time monitoring and knowledge-based. (1993). *SpaceOps 1992, in: Proceedings of the Second International Symposium on Ground Data Systems for Space Mission Operations*, pp. 453-458.

Deam, J. (2020, Outubro 1). *Study: Avoidable ER visits costing U.S. health care system $32 billion per year*. Retrieved from Houston Chronicle: https://www.houstonchronicle.com/business/article/Study-Avoidable-ER-visits-costing-U-S-health-14124489.php

Diggs, J., McCarron, S., Cooper, M., Schwerdtfeger, R., & Craig, J. (2020, November 03). *Accessible Rich Internet Applications (WAI-ARIA)*. Retrieved from W3C: https://www.w3.org/TR/wai-aria/#intro_ria_accessibility

Expo. (2020, December 1). *Introduction to Expo*. Retrieved from Expo: https://docs.expo.io/

Gafini, R. (2009). Informing Science and Information. *Usability issues in mobile-wireless information*, pp. 755-769.

Gartner. (2020, July 8). *Healthcare Digital Transformation*. Retrieved from Gartner: https://www.gartner.com/en/industries/healthcare-providers-digital-transformation

Gomes, S. (2020, July 11). *3 HEALTHCARE APPS WITH THE BEST UI/UX DESIGN*. Retrieved from Imaginary Cloud: https://www.imaginarycloud.com/blog/3-healthcare-apps-with-the-best-ui-ux-design/

Google. (2020, July 10). *Build more accessible apps*. Retrieved from developer.android: https://developer.android.com/guide/topics/ui/accessibility

Hussain, A., & Kutar, M. (2009, June 22-23). Usability Metric Framework for Mobile Phone Application. *10th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting*.

Hussein, A. (2020, October 18). *Expert System With Python -1*. Retrieved from slideshare: https://www.slideshare.net/ahmadhussein45/expert-system-with-python-1

Instone, K. (n.d.). *Navigation stress test*. Retrieved December 1, 2020

ISO. (2020, July 18). *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability*. Retrieved from ISO: https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-1:v1:en

JavaTPoint. (2020, October 18). *Forward Chaining and backward chaining in AI*. Retrieved from Java T Point: https://www.javatpoint.com/forward-chaining-and-backward-chaining-in-ai

Johnston, M. E., Langton, K. B., Haynes, R. B., & Mathieu, A. (1994, January 15). Effects of computer-based clinical decision support systems on clinician performance and patient outcome. A critital appraisal of research. *Ann. Intern. Med. 120 (2)*, pp. 135-142.

Kumar, S. P. (2017, December 30). International Journal of Production Research. *Knowledge-based expert system in manufacturing planning: state-of-the-art review*, pp. 4766-4790.

Kuusinen, K., & Mikkonen, T. (2014). On Designing UX for Mobile Enterprise Apps. *Euromicro Conference on Software Engineering and Advanced Applications*.

Lastovetska, A. (2020, July 10). *Healthcare Mobile App Development: Types, Trends, & Features*. Retrieved from MLSDev: https://mlsdev.com/blog/healthcare-mobile-app-development

Leonard, K. J. (1995). The development of an expert system for fraud alert in consumer credit. *Eur. J. Oper. Res. 80 (2)*, pp. 350-356.

Liao, S.-H. (2004). Expert system methodologies and applications. *Expert system methodologies and applications*.

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014, March). Facing the cold start problem in recommender systems. *Expert Systems with Applications*, pp. 2065-2073.

McCauley, N., & Ala, M. (1992). Inform. Manage. 22. *The use of expert systems in the healthcare industry*, pp. 227-235.

Morville, P., & Rosenfeld, L. (2006). *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites.* O'Reilly.

Nayebi, F., Abran, A., & Deshamais, J.-M. (2012, May). Canadian Conference on Electrical and Computer Engineering. *The State of the Art of Mobile Application Usability Evaluation*.

Patch, K., Spellman, J., & Wahlbin, K. (2020, July 9). *Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile*. Retrieved from W3C: https://www.w3.org/TR/mobile-accessibility-mapping/

Reddy, M. (2020, July 8). *Digital Transformation in Healthcare in 2020: 7 Key Trends*. Retrieved from Digital Authority Partners: https://www.digitalauthority.me/resources/state-of-digital-transformation-healthcare/

Rosul, D. (2020, July 6). *What are the popular types and categories of apps*. Retrieved from ThinkMobiles: https://thinkmobiles.com/blog/popular-types-of-apps/

Ryu, Y. S. (2005). Development of usability questionnaires for electronic mobile products and decision making methods.

Seto, E., Leonard, K. J., Cafazzo, J. A., Barnsley, J., Masino, C., & Ross, H. J. (2012). Developing healthcare rule-based expert systems: Case. *International Journal of Medical Informatics 81*, pp. 556–565.

Statista. (2020, December 1). *Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020*. Retrieved from Statista: https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/#:~:text=React%20Native%20is%20the%20most,software%20developers%20used%20React%20Native.

tutorialspoint. (2020, Outubro 8). *Artificial Intelligence - Expert Systems*. Retrieved from TutorialsPoint: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_expert_systems.htm

Ugalmugle, S., & Swain, R. (2020, July 10). *DIGITAL HEALTH MARKET SIZE BY TECHNOLOGY*. Retrieved from Global Market Insights: https://www.gminsights.com/industry-analysis/digital-health-market

usability.gov. (2020, July 18). *Information Architecture Basics*. Retrieved from usability.gov: https://www.usability.gov/what-and-why/information-architecture.html

Velicer, W., & Prochaska, J. O. (1999, February). An expert system intervention for smoking cessation. *Patient Educ. Couns. 36 (2)*, pp. 119-129.

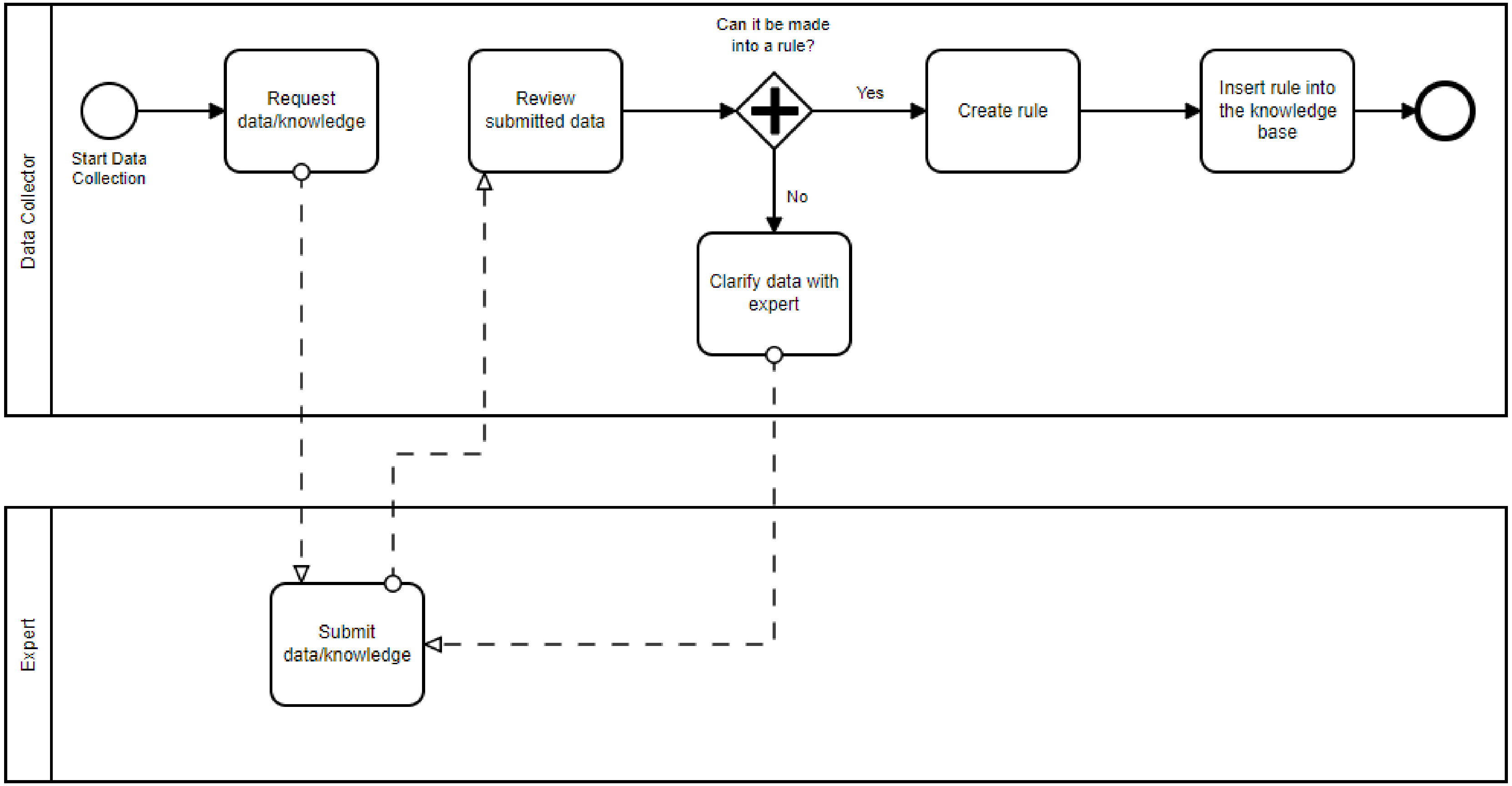
Ventola, L. C. (2020, July 10). *Mobile Devices and Apps for Health Care Professionals: Uses and Benefits*. Retrieved from National Library of Medicine: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4029126/#b1-ptj3905356

Wikipedia. (2020, Outubro 8). *Expert system*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Expert_system

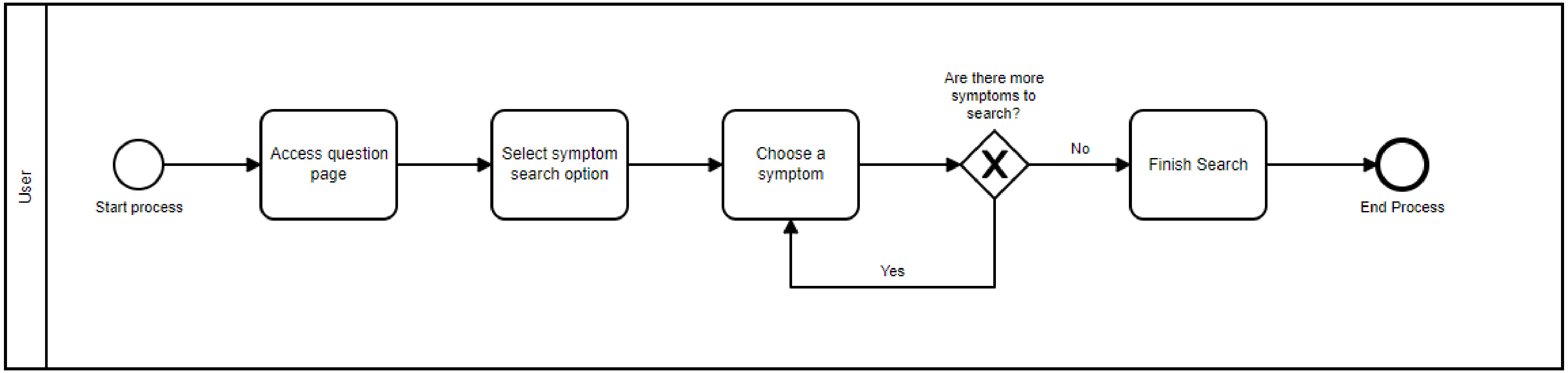
Zhang, D., & Adipat, B. (2005). Challenges, methodologies, and issues in the usability testing of mobile applications. *International Journal of Human-Computer Interaction*, pp. 293-308.

Annexes

Annex 1 - Knowledge collection process



Annex 2 - Symptom Search process



Annex 3 - Expert System API

Request	URL	Notes
get_allData	/allData	Returns all symptom/disease information
select_symptom	/selectSymptom	Returns the subsequent symptoms/diseases given symptom(s) the user selected. The returned data encompasses every symptom the user selected.

```
@app.route('/allData', methods=['GET', 'POST'])
    def get_allData():
        with open('database.json') as jsonfile:
            data = json.load(jsonfile)

    return data


@app.route('/selectSymptom', methods=['GET', 'POST'])
    def select_symptom():
        with open('database.json') as jsonfile:
            data = json.load(jsonfile)

    selected_symptom = json.loads(request.json['body'])
    engine = ExpertEngine(selected_symptom)
    val = jsonify(engine.getSymptoms(data))

    return val
```

Annex 4 - Expert System Functions

```
def getSymptoms(self, allData):
    selectedSymptom = self.val["selectedSymptom"]
    alreadySelected = self.val["alreadySelected"]
    text = self.val["symptomQuery"]
    rules = allData["rules"]
    currentRules = self.getCurrentRules(selectedSymptom, alreadySelected, allData["rules"])

    if text:
      selectedSymptom = self.getSymptomFromText(text, allData["symptoms"])

    diseases = []
    symptoms = []
    if selectedSymptom:
      if "diseases" in currentRules[selectedSymptom]:
            diseases = currentRules[selectedSymptom]["diseases"]

      if "symptoms" in currentRules[selectedSymptom]:
            symptoms = list(currentRules[selectedSymptom]["symptoms"].keys())

    return {"symptoms": symptoms, "diseases": diseases,"selectedSymptom": selectedSymptom}
```

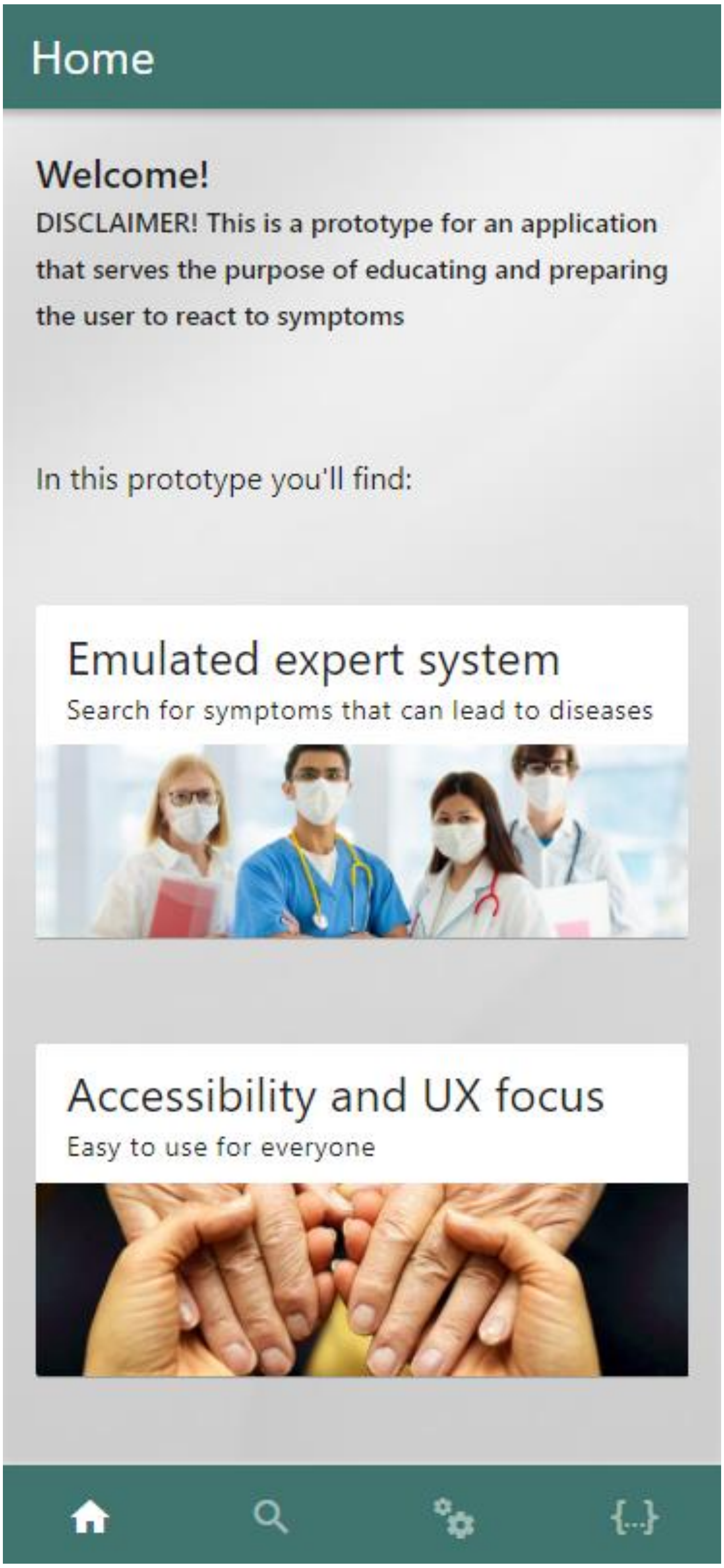
```
def getCurrentRules(self, newSymptom, selectedSymptoms, rules):
    rulesToReturn = rules
    if selectedSymptoms:
        for symp in selectedSymptoms:
        if newSymptom in rulesToReturn[symp]["symptoms"]:
            rulesToReturn = rulesToReturn[symp]["symptoms"]

    return rulesToReturn
```

```
def getSymptomFromText(self, text, symptoms):
        tokenized_text = sent_tokenize(text)
        stop_words = set(corpus.stopwords.words("english"))
        filtered_sent = []

    /*Splitting text and removing stopwords*/
    for tkn in tokenized_text:
            if tkn not in stop_words:
                filtered_sent.append(tkn)
        print(tokenized_text)

    /*Process for removing commoner morphological and inflexional endings from words in english*/
    ps = PorterStemmer()
        stemmed_words = []
        for flt in filtered_sent:
            stemmed_words.append(ps.stem(flt))

    print("Filtered Sentence:", filtered_sent)
    print("Stemmed Sentence:", stemmed_words)

    /*Process of grouping together the different inflicted forms of a word so they can be analysed as a single item*/
    lem = WordNetLemmatizer()
        lemmatized_words = []
        for stm in stemmed_words:
            lem.lemmatize(stm, "v")

    print("Lemmatized Words:", lemmatized_words)

    /*Searches the filtered words from the text, returning the symptom it recognized from the global symptom list*/
    symptom = ""
        for symp in symptoms:
            for word in lemmatized_words:
                if word in symp:
                    symptom = symp

    return symptom
```

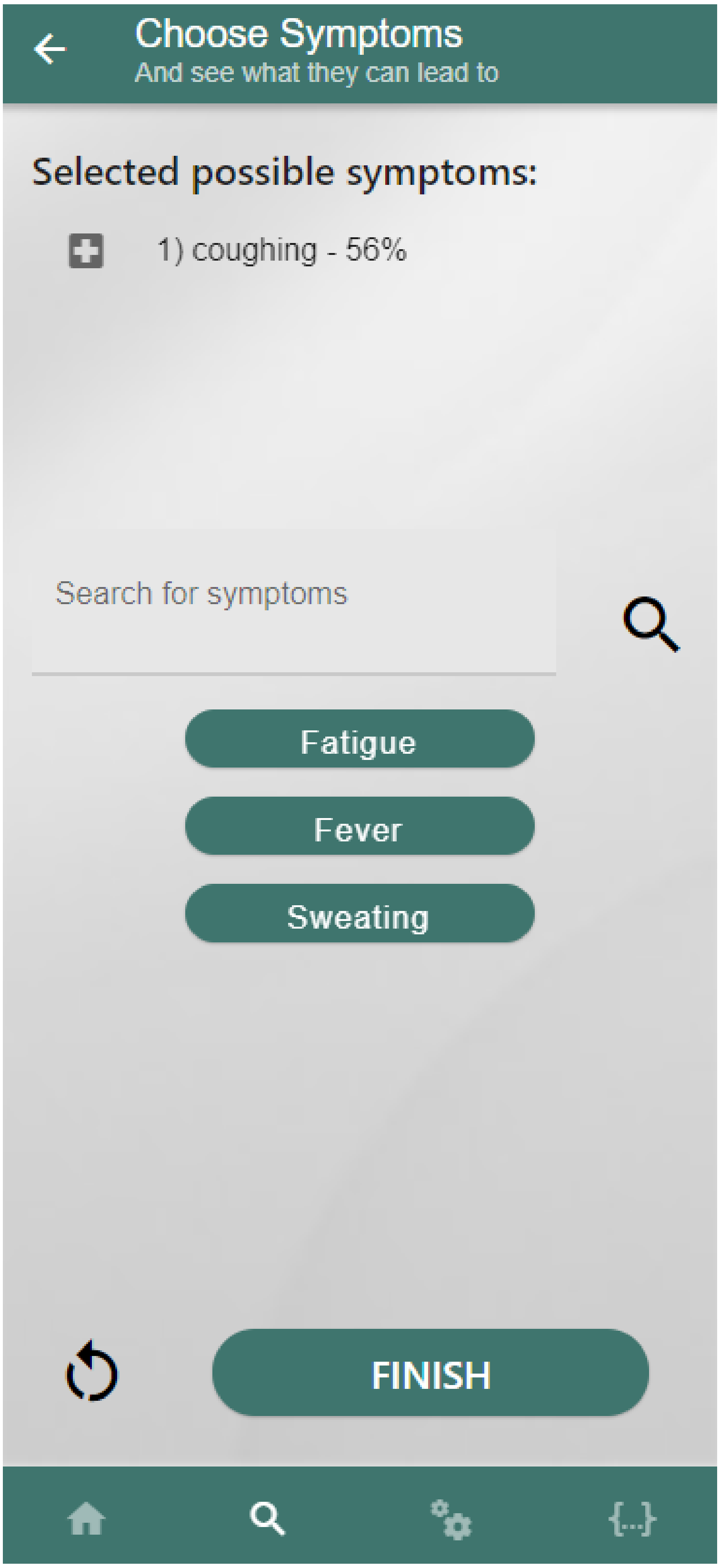
Annex 5 – JSON rule tree example


```
{
  "diseases": ["Cold", "Headache", "PneumoniaCoiso", "Salmonella", "Migraine"],
  "rules": {
    "coughing": {
      "diseases": ["Cold"],
      "symptoms": {
        "fatigue": {
          "symptoms": { "fever": { "diseases": ["Pneumonia"], "weight": 0.85 } }
        },
        "fever": {
          "symptoms": {
            "fatigue": { "diseases": ["Pneumonia"], "weight": 0.73 }
          }
        },
        "sweating": { "diseases": ["Headache"], "weight": 0.42 }
      },
      "weight": 0.56
    },
    "fatigue": {
      "symptoms": {
        "coughing": {
          "symptoms": { "fever": { "diseases": ["Pneumonia"], "weight": 0.49 } }
        },
        "fever": {
          "symptoms": {
            "coughing": { "diseases": ["Pneumonia"], "weight": 0.82 }
          }
        },
        "nausea": { "diseases": ["Salmonella"], "weight": 0.5 }
      }
    },
    "fever": {
      "symptoms": {
        "coughing": {
          "symptoms": {
            "fatigue": { "diseases": ["Pneumonia"], "weight": 0.24 }
          }
        },
        "fatigue": {
          "symptoms": {
            "coughing": { "diseases": ["Pneumonia"], "weight": 0.67 }
          }
        }
      }
    },
    "nausea": {
      "diseases": ["Migraine"],
      "symptoms": { "fatigue": { "diseases": ["Salmonella"], "weight": 0.69 } },
      "weight": 0.99
    },
    "sweating": {
      "symptoms": { "coughing": { "diseases": ["Headache"], "weight": 0.79 } }
    }
  },
  "symptoms": ["coughing", "sweating", "fever", "fatigue", "nausea"]
}
```

Annex 6 – Home Screen



Annex 7 – Symptom search Screen



Annex 8 – Settings Screen

Annex 9 – Admin Screen