

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**TOWARDS AN ENGAGING SOCIAL TV ON A
SECOND SCREEN**

Vítor Hugo Gonçalves Martins

TRABALHO DE PROJETO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

Trabalho de Projeto orientado pelo Prof. Doutor Francisco José Moreira Couto
e co-orientado pelo Prof. Doutor Paulo Ricardo Pacheco Rodrigues Trezentos

2015

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**TOWARDS AN ENGAGING SOCIAL TV ON A
SECOND SCREEN**

Vítor Hugo Gonçalves Martins

TRABALHO DE PROJETO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

Trabalho de Projeto orientado pelo Prof. Doutor Francisco José Moreira Couto
e co-orientado pelo Prof. Doutor Paulo Ricardo Pacheco Rodrigues Trezentos

2015

Acknowledgments

First and foremost I would like to thank my supervisors, Prof. Dr. Francisco José Moreira and Prof. Dr. Paulo Ricardo Pacheco Rodrigues Trezentos for the opportunity, for guiding me through this master thesis, for their patience and for their availability. A special thanks to Tiago Guerreiro, who in addition of being my teacher during this master degree, helped me with brainstorming and advices besides being available to perform an usability evaluation. Moreover, I thank all my teachers at Faculdade de Ciências da Universidade de Lisboa for their teachings, not only the lesson plan but also their human lessons. Furthermore, I thank all my co-workers and friends in University for their companionship.

I would also like to thank all people at Caixa Mágica Software for the way they received me and who also helped me with the trial. A special thanks to Ana Elias and Vasco Silva who helped my quick integration in the company.

Finally I would like to thank my family, friends and girlfriend for their love and patient along my academic path.

To my parents and brother

Resumo

Este trabalho debruça-se sobre a integração das redes sociais com conteúdos televisivos e na experiência de visualização dos consumidores de televisão.

Os sistemas de informação de hoje em dia estão cada vez mais orientados ao utilizador e às redes sociais que estes usam. As empresas estão obrigadas a utilizar estas ferramentas para as suas estratégias de marketing e muitas delas têm parcerias com redes sociais como o Facebook ou o Twitter para compreender os interesses e desejos dos seus clientes. Desta forma, as empresas produtoras de conteúdos televisivos têm estado bastante atentas a este fenómeno.

Recentemente, temos vindo a assistir ao aparecimento de aplicações de redes sociais inseridas em *Smart TVs*, *set-top-boxes* e outros sistemas de televisão. Na sua maioria, oferecem as mesmas rotinas de interação que os seus clientes têm nas redes sociais, no conforto dos seus sofás e através da televisão. Muitas destas têm vindo a esforçar-se por deixarem de replicar somente as funcionalidades já existentes. Nesse sentido, têm vindo a oferecer funcionalidades para os espectadores poderem partilhar os seus hábitos de consumo televisivo.

As aplicações que integram redes sociais com os conteúdos televisivos têm vindo a ser categorizadas de várias formas na literatura existente. O termo televisão social é o mais comum para descrever este tipo de fenómeno. A televisão social é, na verdade, um conceito antigo. Há algum tempo atrás, os espectadores assistiam a um programa de televisão e no dia seguinte discutiam-no ao pequeno almoço em sua casa, no trabalho ou num bar com os amigos. Agora, essa discussão foi transposta para as redes sociais, durante a emissão em directo. Formalmente, a televisão social pode ser definida como o conjunto de tecnologias que dão suporte às interações sociais entre os espectadores. Com a ajuda da Internet é possível obter a mesma experiência de TV com pessoas de todo o mundo, como se estas estivessem a fazer-nos companhia no sofá de nossa casa. A simples partilha do programa que está a ser visto é, muitas vezes, o ponto de partida para discussões sobre programas. Desde há muitas décadas, que é comum para os espectadores ver televisão e falar sobre actores, programas, moda ou música presente na TV. Estes tem vindo a fazê-lo *online*, de forma natural e confortável, desde o aparecimento das redes sociais, sobretudo o Facebook e o Twitter. Estas permitem uma comunicação bidireccional no sentido em que qualquer pessoa pode partilhar a sua opinião e simultaneamente ter conhecimento

da opinião do restante universo de espectadores, bem como de conteúdos providenciados pelas produtoras. Para os telespectadores, a televisão social permite uma experiência envolvente que lhes dá a possibilidade de troca de opinião televisiva com qualquer parte do mundo. Além disto, aqui é lhes dado o poder de dizer aos produtores de conteúdos quais os seus interesses e desejos. Estes ganham conectividade, interactividade e melhores conteúdos. O objectivo da televisão social passa por enriquecer a experiência de ver televisão através das redes sociais e, simultaneamente, fazer crescer as audiências televisivas.

Nos últimos anos, tem-se vindo a captar uma tendência crescente em torno de outro assunto integrado neste tema, o segundo ecrã. Este é um conceito muitas vezes relacionado e confundido com a televisão social.

O Segundo Ecrã, pode ser definido como o conjunto de tecnologias, onde o utilizador pode receber dados em tempo real ou informação extra sobre o conteúdo, que está a ser emitido na televisão, em directo. Este segundo ecrã é, na sua origem, uma companhia para o telespectador e um complemento ao primeiro ecrã, a televisão.

Tornou-se um assunto popular nos últimos anos e teve um destaque mais acentuado com a consolidação da utilização de *smartphones* e *tablets*, especialmente enquanto se vê televisão. Este destaque está relacionado com o aparecimento de aplicações que sincronizam o conteúdo da TV com os dispositivos móveis. Com o lançamento do iPad e outros *tablets*, surgiu uma nova era de experiências *lean-back*¹ e no início de 2011 começaram a aparecer as primeiras aplicações *second screen*. Desta forma, os produtores de conteúdos de televisão têm investido em criar este tipo de aplicações, fornecendo ao espectador mais informação sobre o que está ver na televisão, através de um segundo ecrã. Os típicos casos de aplicação são os programas de desporto, séries, *reality-shows* ou outro tipo de concurso. Este conceito de *second screen* foi crescendo também com a necessidade de fazer os espectadores interagirem com os programas em tempo real e dar-lhes uma voz activa. Assim, este tipo de aplicações permite muitas vezes ao espectador votar em algo e influenciar o desenrolar do programa, em directo. Na literatura pode-se encontrar uma categorização do tipo de aplicações *second screen*. A primeira agrupa as aplicações baseadas em tecnologia capaz de reconhecer o programa que está a ser visto pelo utilizador. A segunda diz respeito as que oferecem a experiência do realizador, permitindo o acesso às várias câmaras da emissão. Por último, várias fazem integração com o Facebook e Twitter, mostrando os últimos comentários nas redes sociais. A expansão desta categoria vive da necessidade de reconhecer os interesses dos espectadores.

Embora os sistemas televisivos apresentem actualmente uma larga oferta de produtos que fazem algum tipo de integração da televisão com as redes sociais, estes não têm conseguido envolver os espectadores de forma eficaz e que os leve a serem utilizados frequentemente. Nos E.U.A., o país com maior oferta de aplicações nas categorias da televisão

¹experiência onde o espectador senta-se, relaxadamente reclinado para trás

social e do segundo ecrã, 86% das pessoas não têm estas aplicações instaladas nos seus dispositivos móveis. No entanto, apenas 24% destas não são donas de um *smartphone* ou *tablet*. Assim, este trabalho de mestrado detecta as falhas e sucessos destas aplicações e apresenta uma solução que facilita e incentiva a integração social dos conteúdos televisivos. Desta forma, esta tese apresenta o estado da arte nas áreas da televisão social e do segundo ecrã, investigando os estudos e os inquéritos realizados nestas áreas. Assim, explora-se o que tem vindo a falhar nestas aplicações, bem como os avanços tecnológicos que têm envolvido os consumidores de televisão. Uma vez feito o diagnóstico, é proposta uma solução para a integração das redes sociais com a televisão num segundo ecrã, sincronizada com uma *set-top-box*. Esta solução procura explorar as virtudes de alguns produtos já existentes, eliminando as barreiras para a sua usabilidade e criando novos incentivos para a interação social através da televisão. O facto de reconhecer automaticamente o programa que está sintonizado na *set-top-box* do espectador irá mostrar como um *check-in* numa aplicação de segundo ecrã pode ser transparente para o utilizador. Alinhado com esta ideia, a aplicação mostra recomendações de programas feitas por amigos do Facebook que ao ser carregadas transportam automaticamente o espectador para a recomendação. De forma a guiar o espectador no que pode ver na sua televisão, informação sobre os programas mais quentes a nível social é dada em forma de recomendação. Assim, esta aplicação visa melhorar a usabilidade e o envolvimento dos espectadores sem os distrair do primeiro ecrã, a TV. No fim, é feita uma avaliação à usabilidade da aplicação e à satisfação dos espectadores de televisão através de uma amostra, de forma a estimar o sucesso da mesma. Aqui, os resultados mostram que os utilizadores, no geral, classificaram a aplicação como tendo uma usabilidade satisfatória e que apresenta um design apelativo. Esta avaliação revela também as descobertas feitas em relação às funcionalidades que os participantes acharam mais interessantes.

Palavras-chave: Televisão, Redes Sociais, Segundo Ecrã, Interatividade, Usabilidade

Abstract

Social television can be defined as the technologies supporting social interactions among viewers. Second screen became a trendy topic in the last years and had a buzz with the uprise of smartphones usage, specially while watching TV. This is a technology in which the user gets real time data about a live TV show on a mobile device.

In the last years, many social TV applications appeared on television systems but, in general, they replicate the features from the existing social networks. Besides that, this environment is not suited to them due to its usability and design space and, as so, they tend to be integrated in second screen products. In spite of the existing offer, statistics reveal that users are not engaged with these applications. In the U.S.A., 86% of people do not have this type of applications installed on their devices, where 24% of them do not have a mobile device.

This thesis addresses the integration of social networks in the television content and viewers experience. It studies the previous research and products on Social TV and Second Screen areas and proposes a new solution.

Furthermore, it detects the flaws and successes of previous applications and then presents the design and development of a new solution. The solution aims to offer an appealing social integration with TV content, by automatic synchronization between the mobile device and the set-top-box, offering social features focused on TV content. Thus, it allows TV recommendation between friends and can be used as a social TV guide. Lastly, it presents an evaluation of its usability, design and utility, where it shown that is expected to be even more useful to who consume more TV. Moreover, achievements on usability, main findings about the most engaging features and interesting comments made by the participants are presented.

Keywords: Television, Social Networks, Second Screen, Interactivity, Usability

Contents

List of Figures	xviii
List of Tables	xxi
1 Introduction	1
1.1 Motivation	3
1.2 Goals	5
1.3 Planning	8
1.4 Contributions	8
1.5 Structure of the document	9
2 Related Work	13
2.1 Social TV	13
2.2 Second Screen Flaws	15
2.3 Time-Shifting	15
2.4 Implementations	16
2.4.1 Prototypes	16
2.4.2 Products	17
2.5 Android	25
2.5.1 Material Design	25
2.5.2 Libraries	26
2.6 Graph Data Bases	27
2.6.1 Neo4J	28
3 Social Zapping	31
3.1 Requirements	31
3.1.1 Non Functional Requirements	31
3.1.2 Functional Requirements	31
3.2 Design	32
3.2.1 1st Prototype	33
3.2.2 2nd Prototype	37
3.2.3 Navigation Flow Diagram	40

3.2.4	Output	41
3.2.5	Preliminary Experts Test	45
3.3	Implementation	47
3.3.1	Back-End Database	47
3.3.2	Back-End Services	51
3.3.3	Mobile Application	52
3.3.4	Ranking Points	65
4	Evaluation	69
4.1	Trial Setup	70
4.2	Data Collection	70
4.2.1	Auxiliary Application	71
4.3	Participants' Characterization	71
4.4	Results	73
4.4.1	Usability	73
4.4.2	Overall Opinion	74
4.4.3	Engagement	77
4.4.4	Observations, Comments and Suggestions	79
4.5	Discussion	81
4.5.1	Usability	81
4.5.2	Overall Opinion	81
4.5.3	Engagement	82
5	Conclusion	85
5.1	Achievements	85
5.1.1	Design and Usability	85
5.1.2	Focus on the TV content	86
5.1.3	Social TV engagement	87
5.1.4	Evaluation	88
5.2	Future Work	89
A	Cypher Queries	91
B	Evaluation Guidelines	99
C	Neo4J Database Access in Android	101
	Bibliography	106

List of Figures

2.1	CaboVisão - Facebook	19
2.2	Viggle - Mobile application with Second Screen and Social TV features	20
2.3	State of the Art - Social TV Products	24
2.4	Graph expression representation in a diagram	29
3.1	1st Prototype - Home (Comments) and Minimized Menu	34
3.2	1st Prototype - Comment TV Show on Twitter	34
3.3	1st Prototype - My Recommendations	35
3.4	1st Prototype - Friends Recommendations	35
3.5	1st Prototype - Buzz Screen and Show Buzz Detail	36
3.6	1st Prototype - Recommendation	36
3.7	Navigation Flow Diagram	40
3.8	2nd Prototype - Highlights	41
3.9	2nd Prototype - Side Menu	41
3.10	2nd Prototype - Recommend Button	42
3.11	2nd Prototype - Choose Friend To Recommend	42
3.12	2nd Prototype - Detail Of A TV Show	42
3.13	2nd Prototype - Detail Of A Friend	43
3.14	2nd Prototype - What I Recommended	43
3.15	2nd Prototype - Friends Recommendations	44
3.16	2nd Prototype - Social TV Guide	44
3.17	2nd Prototype - What Friends Are Watching	44
3.18	Neo4j Graph View - My Social Graph on the application	49
3.19	Neo4j Graph View - My Likes on Facebook pages related to TV content	50
3.20	Neo4j Graph View - The recommendations I have made and received from friends	51
3.21	Social Zapping - Login Screen	53
3.22	Social Zapping - Side Menu	54
3.23	Social Zapping - Highlights Screen (Top)	54
3.24	Social Zapping - Highlights Screen (Bottom)	55
3.25	Social Zapping - Recommend Button	56
3.26	Friends Available For Recommendation	57

3.27	Social Zapping - Insert Text On Recommendation	57
3.28	Social Zapping - TV Show Details	58
3.29	Social Zapping - Friend Detail	59
3.30	Social Zapping - Friend Detail (With Recommendations To Follow)	59
3.31	Social Zapping - Recommendations I Made	61
3.32	Social Zapping - Recommendations From Friends	62
3.33	Social Zapping - TV Guide (EPG Ordered By Social Buzz)	63
3.34	Social Zapping - What My Friends Are Watching	64
3.35	Precision formula	65
3.36	Recall formula	65
3.37	F1 Measure formula	65
3.38	Mapping of Precision formula for TV show's recommendations	66
3.39	Mapping of Recall formula for TV show's recommendations	66
3.40	Mapping of F-Measure formula for TV show and user ranking	66
3.41	Hybrid formula for TV shows Ranking	67
4.1	Social Zapping Simulator - Channel switch during trial	71
4.2	Participants' Characterization - Gender and Age(top), Education and TV Consumption(bottom)	72
4.3	How hard was for you to accomplish this task?	74
4.4	How much fun do you had using this application?	75
4.5	Fun vs TV Consumption	75
4.6	How much do you like the application's design?	76
4.7	How hard was to navigate on the application?	76
4.8	How useful would this application be to you if it was a commercial product?	77
4.9	Utility vs TV Consumption	77
4.10	Evaluate how interesting is each feature for you?	78

List of Tables

1.1	Planning	8
3.1	Non Functional Requirements (NFR)	31
3.2	Functional Requirements (FR)	32
3.3	Tasks Performed On Experts Test	45
3.4	Results of Experts Evaluation	46
3.5	Functional requirements satisfied in each screen	67
4.1	List of tasks to perform by the participants	73
4.2	List of features to evaluate by the participants	78

Chapter 1

Introduction

Work Environment

This project takes place on a professional environment, at Caixa Mágica Software. Caixa Mágica is a company that provides robust technological solutions based on Open Source software to the domestic and enterprise markets. It has strong partnerships with the Portuguese Public Administration, the Portuguese Army, large hardware providers such as JP Inspiring Knowledge, Intel and Insys, it is involved in big European Projects (leader in some of them), and as such is now a global player. Recently, Caixa Mágica developed a new application for interactive TV of NOS and the 2014 World Cup App. Before that, it developed the application about the Portuguese Soccer League, also for interactive TV of NOS. Some of the main Business Units of this company are the Professional Services and the Research and Development units. As so, there's an experience at the innovation area, at analyzing and creating new products. As so, the proposal for this project is the development of a new solution for the television market, integrated in the Research and Development unit of Caixa Mágica. The project is encapsulated on this intern unit but aims to be a proof of concept and to be proposed for the enterprise market. Due to the research nature of this project, complete implementations on commercial TV systems are not expected and as so, assumptions on some technical details are made.

The project addresses the areas of Internet Protocol Television (IPTV) Systems, social Networks and mobile applications, with focus on the Social TV and Second Screen topics.

Social TV

A few time ago, people watched television and talked about the show in the next day during breakfast at home, on work or even in a bar. Now that discussion has been taken to the social networks, during the live show. Social television is often a misunderstood concept. The name, itself, points to the related topics: television and social media. It can

be defined as the technologies supporting social interactions among viewers. Social networks are part of our daily lives and social TV leverage these powerful resources to make people discuss a TV show, while watching it. With the help of the Internet is possible to get the same TV experience with people from all over the world as if they were in your couch.

Quite often, this term is mixed up with Internet TV, Internet Protocol TV or Web TV. Internet TV refers to streaming contents on a media player or to download content. Internet Protocol TV uses the Internet instead of satellite or cable to deliver TV services. Web TV allows you to watch web-based TV shows online.

The typical use case for a viewer is tweet about a favorite show or a show that he is watching, start a Facebook page or follow an existing page about a TV show, enter a thread to discuss about the TV show on air or simply share to the world that he's watching his favorite show. Since 1939 that most Americans are watching television and it's natural to them to want to talk about actors, shows, fashion and music in TV shows. They have been doing this online in a natural and comfortable way since the arrive of Facebook and specially Twitter. It allows a two way communication, they can share their opinion and check also everybody's opinion about that show and contents from official shows pages. For the viewers, social TV allows an engaging experience that gives them the power of telling content producers what their tastes and interests. With social TV they gain connectivity, interactivity and better content. On the other side, content producers gain access to richer data about customers. This way, they can know what's trending, what's liked or not by viewers, routing them to most suited contents. The goal of social TV is making the concept of watching television a richer experience by social networking and grow TV audiences that way.

Second Screen

Second screen became a trendy topic in the last years and had a buzz with the uprise of smartphones usage, specially while watching TV. This buzz is related to increase of applications synchronizing TV contents with mobile devices. With the iPad launch and other tablet devices, a new era of lean-back experiences arrived and the first second screen applications appeared in the beginning of 2011.

This concept is many times interchanged with Social TV but they are not the same. Second Screen is a technology where the user gets real time data and extra information about a live TV show on a mobile device. This way, the mobile device works as a second screen as the TV continues to be the main screen. In the last years, TV content producers invested on creating second screen applications, giving to viewers a complementary information about what they are seeing live on TV. The typical genres involved on this investment are series and sport games. This way, it is possible for the viewer to get extra information in real time about what he's seeing like trivia about characters, the fashion brand that a

character is using, a soundtrack or statistics about a football match.

Across that, the second screen idea expanded to the remote control. Nowadays, there are many second screen applications integrated with remote control features, assuming the role of the remote and leaving it out of the couch.

There are three essential genres of second screen applications. First, there are applications using audio and digital fingerprints recognition to know when to display the contents on the second screen. Second, there are the “director’s experience”. These type of applications offer live-only features like different cameras on the second screen or access to the backstage areas. Lastly, there are lots of them integrated with Facebook or Twitter, showing the last comments on social networks. This can be feeds from the cast, feeds from an official channel of show page or, more often, discussions with other viewers.

The expansion of this concept continues with the need of recognizing viewers tastes and providing them accurate contents.

Back to the concepts interchanging, Social TV is a subset of second screen. The definition and typical applications of the second screen presented above don’t have anything social about them.

1.1 Motivation

Along with the proposal for this project, Caixa Mágica presented the initial motivation for this work as follows. Nowadays, many information systems are user-driven and oriented to the social networks they belong. Although the existing integration of social networks in TV systems, its usage through the remote control it is not workable and leads them to desert these products. Furthermore, this integration should be explored to guide the viewers on what to watch on TV.

Usability and Display Space

In the last years, many social TV applications have been developed, specially in U.S.A. The majority have been implemented on television systems like Smart TVs and Set-Top-Boxes. The first flaw we can see on these applications is the lack of concern about the design space and the TV purpose. In general, a social application like Facebook opens on top of the main image like a sidebar or, as in many cases, occupying the whole screen. Viewers do not want to stop watching television while they use social networks. If that was true, they would simply turn the TV off and browse on social networks. Thus, the first lesson is to remember the nature of TV viewers and as so, they sit on their couch to get comfortable and to enjoy the big screen. This way, TV has to be used for its main purpose, a big screen that displays TV contents in front of a comfortable couch. In these lines, another main problem about applications incorporated on TV systems remains on the interaction models. The traditional remote control is not a handful way to interact within

a social network. Today, a social network has many levels of functionality, therefore there are too many buttons to access by remote navigation. The arrow navigation on menus, and specially the text typing with a T9 keyboard, are not efficient with the remote control and leads viewers not to use these applications. Some research have been done in this domain [7] [4], and we can use new models of interaction, evicting barriers to the usage of such applications. In these studies, we can verify that users find a conflict in these applications, when they occupy the same display space as the main image, and that a second screen offers the possibility to remove the need of showing UI elements on the main screen. In addition, it increases the performance by reducing the UI latency, and faster reaction, comparing with the television systems. In terms of usability issues, the remote control often assumes the main cause of problems and some authors say that it can soon disappear of our living rooms [7] [4]. Unlike the infrared communication of the remote, the second screen has the advantage of not needing to be pointed directly at the television set, to work. Hence, it was already obvious in 2007 [7] that the remote control could not be used for interactive TV and specially for social applications in television systems. Then, in 2014 [4], interaction models as tablet, freehand gestures and remote control have been compared and has been stated that a mirrored TV screen on a tablet is the most promising alternative to interact with the TV. Thus, social applications interacting with TV content cannot be competing for the display space with live shows and cannot be controlled by a remote control. That way, a solution for social TV in a second screen is needed.

Social Integration with TV content

As stated by a second screen application CEO (Anthony Rose, Zeebox), one of the keys factors that consumers need in a second screen application is Social. He says that in their research they found that “Consumers need to be able to connect and share with friends during and about TV shows”¹. Across that, another problem about current social applications on TV, is the fact that they are replicating the original social networks. They give you access to news feed, allow to check friends profile and timelines, follow people, make new friends and find trends. Users already do this on the original social applications like Facebook or Twitter and they will not use another application to have the same features with no added value.

Thus, the solution must enroll social applications within the TV context, and find ways to offer a centralized experience, where some specific social features can engage people to TV content and give them a better experience, instead of distracting them to other environments. Social features like TV shows hashtags and trends on Twitter, status updates about TV shows on Facebook or backstage photos on Instagram are a good example of social features related to TV. Right now, most of these applications are disappointing viewers

¹<http://www.digitaltrends.com/home-theater/the-difference-between-second-screen-and-social-tv-apps/>.

because they are only throwing a Twitter stream on the sidebar and including a “share” button. Social interactions like making a tweet or posting a like in a TV show should have some feedback. These actions need to be observed, used and replied by others to be social and give some emotion to who made those actions. TweetTV founder and CEO, Bradley Markham said: “Simply tweeting to say ‘I’m watching this football game,’ is no more social than being stranded on an island and throwing out a message in a bottle saying ‘I’m stuck on this island.’ That message wouldn’t be social until someone actually responded.”¹. Thus, this social interaction feedback is required toward enriching the viewing experience through social networking about TV content.

These observations reveal the main challenges to be addressed in this work, and argue the need for a reflection on the way these applications are trying to engage the audience. Across that, there is the opportunity to develop a solution in a country (Portugal) where the TV consumption is one of the most sophisticated in the world and where the TV services are well developed, regarding the technology².

1.2 Goals

Regarding the motivation presented in the last section, the objective of this project is the design, implementation and evaluation of an effective social TV application on a second screen, integrating in a Internet Protocol television (IPTV) system. The main goals are categorized in the following topics.

Design and Usability

This application has to offer an engaging and perceptive interface, for the sake of viewers usage. The user experience has to be easy and natural. Hence, there must be an effort to make the interface reactions quick and smooth. It must accomplish the traditional design and usability principles. Across that, there is intent to confirm the advantage of a second screen by not having the main screen space reduced because of a social application and, at the same time, have better graphics and faster responses. The time spent to make any social interaction on the second screen must be farther less than as if it is done on a Set-Top-Box(STB) application. The issues surrounding STB applications is eliminated with the development of a mobile application on the Android environment, having the second screen purpose for the social integration with TV content. For the design and usability accomplishment, several patterns are checked³ and time complexity is optimized

²<http://economico.sapo.pt/noticias/consumo-de-tv-em-portugal-ao-nivel-dos-paises-mais-desenvolvido>

³<https://developer.android.com/design/index.html>.

by direct communication with the set-top-box. The final evaluation with participants tests the usability accomplishment.

Focus on the TV content

This solution should not make the same mistakes like other previous works, as in distraction is concerned. The application should not offer exaggerated information and too much screen elements to be functional. In addition, it must focus on maximize the automation of social tasks and TV controls, for watching and social networking about the TV content, synchronously. This way, there must be a direct communication with the TV system, in order to make an efficient check in, by the recognition of the tuned channel on the STB. In the initial investigation and study about the state of the art, many features are identified and analyzed regarding their impact on the attention given to the TV content. Their donation for social integration with TV content is also considered. Hence, features with too much information, not related to the TV content and time consuming are identified and avoided. As referred in the previous point, the direct communication decreases the time spent on syncing the TV with the application and, therefore, the distraction from the TV is reduced. Therefore, the application has to keep the viewers eyes on the main screen, for the most of the time, through smooth automatic check in and synchronization with the set-top-box.

Social TV Engagement

The main challenge for this type of solution is making the final consumers to use them. That way, one of the main goals is to offer a better TV experience through social networking. The solution must engage the viewers by offering some added value, regarding the social networks integration with TV. This way, it must incite emotions on the viewers. The actions of sharing the TV experience with friends, recommend shows and discuss with strangers have to be encouraged, natural and enjoyable to the user. To bring about the engagement and emotional feedback, a search for the reasons to share is made and social features are adjusted to TV content. While the viewer is watching the content, a “recommend” button is available. When pressed, this button allows to choose a Facebook friend and post on his Facebook timeline, recommending the show. Furthermore, the second screen presents its own Timeline, with entries from the friends recommendations. Here, each recommendation entry in the timeline have a button to tune in the recommended show directly on the STB, with no need to use the remote. This can work as a “Social Zapping”. To engage the recommendation action, users are informed of which of their friends followed their recommendation and whose recommendations were the most followed. The actions of comment and recommend a program also have an impact on the show popularity. This popularity is used as a recommendation tool, helping the viewer

on deciding what to watch. The show popularity is presented to the user as a graphic, showing the top live shows. Each bar have also a link to switch to that show.

Evaluation

After the development of the mobile application, a concept evaluation has to be done. This way, and for sake of evaluation, a field trial must be done. The trial must evaluate all the previous topics by taking notes, counting the time spent in each task and in each screen, and collecting participants opinion. This aimed to do an approximation of a traditional living room environment: a comfortable couch and a big screen. This evaluation is performed by a inviting a set of participants, a sample of the TV consumers universe. It was required to define a set of tasks to perform by the participants in order to test the previous objectives. The main questions to be answered were: Is easy to understand the interface? Does it have an appealing interface? Does it have quick reactions? How much time it takes to perform a task? Are the viewers intending to use the solution to talk about the show, instead of using Facebook or Twitter applications? Does it engage the viewers in the TV content or distract them? Does it offer a better TV experience? The questions are answered by notes taken on the observation of the participants behavior and by a final survey to the participants. The time consumed is considered an annotated also.

1.3 Planning

Table 1.1 shows this work planning.

Start Date	End Date	Duration	Phase	Description
25/09/14	24/11/14	8 weeks	Analysis and Writing	- Definition of project's goals and requirements. - State of the Art Analysis. - Preliminary report writing.
25/11/14	16/12/14	3 weeks	Design	- Storyboard sketches design. - Detailed design and specification of the solution's architecture. - Back-end implementation.
17/12/15	29/04/15	19 weeks	Development	- Analysis of the communication between the Second Screen (Android device) and the STB. - Implementation of automatic check in and channel switch. - Exploration of the Facebook API to authenticate users and collect and store(back-end) the social graph. - Develop crawlers to collect Twitter streams and Facebook posts and store them on the back-end. - TV show's popularity calculation. - Android application development integrating all previous implementation outputs.
30/04/15	21/05/15	3 weeks	Testing	- Functional tests definition and execution. - Global tests to the system. - Bug fixes and upgrades
22/05/15	05/06/15	2 weeks	Evaluation	- Definition of evaluation questions. - Preparation of the field trial. - Observation and annotation of participants opinion and behavior. - Suggestions analysis.
07/06/15	25/06/15	3 weeks	Writing	- Final thesis report writing.

Table 1.1: Planning

1.4 Contributions

With the motivation and the goals defined on the previous sections, these are the main contributions of this work:

- Revision of the related work in the Social TV and Second Screen fields of study, besides scientific studies also commercial products.

- A design proposal, following Material Design (2.5.1), Google design specification for Web and Mobile applications emerged recently. As so, this design proposal follows the last design trends and is consistent with international standards about design. The design proposal is the result of some iterations on creating high-level mockups, experts heuristic evaluation and brainstorming with experts on Information Systems.
- Social Zapping, a mobile application making a proof of concept of an engaging and usable solution for integrating social networks with TV content. Here, recommendation rules are important. A user has to be watching a show in order to be able to recommend the show. The recommendation is considered followed only when a user sees the recommendation, by tuning it on his TV via Social Zapping. To accomplish this, automatic synchronization with STB is needed as strong emotional feedback about recommendations made also as receiving trustworthy friends recommendations, given that they had to watch it before recommending me.
- An evaluation of the mobile application, where is possible to check how useful Social Zapping can be, being expected to be even more useful to viewers who consume more TV. Besides that, it is showed that Social Zapping achieved a reasonable usability level and had a good feedback by who experimented it. Main findings about the most engaging features are presented. Here is also interesting to check the participants comments and suggestions. This application is also prepared to be integrated with TV cable operators with little effort.
- Social Zapping Simulator, an Android application, serving as an evaluation tool. This tool helps the person who is guiding the trial to avoid invading the participant's space and reduce the need to peek the device on the participant's hands, by showing which channel the user picked, highlighting the user who switched the channel. This can also help on sorting recommendations for live shows to simulate the social trends on show's popularity in real time.

1.5 Structure of the document

This document is organised as follows:

- Chapter 2: Provides a background for this work by a brief description of some of the relevant products and technologies in this area. It also refer their flaws and strengths. Tries to find reasons for the low usage of such products and get some inspiration in existing and successful features.
- Chapter 3 - Describes how the work was done until reach the final product, a mobile application called Social Zapping. Here, is detailed how the design evolved and all

the decisions taken during the process. Moreover, it is explained how the mobile application was developed fulfilling the design.

- Chapter 4 - Is about the evaluation of Social Zapping by users. It reveals the preparation and setup of the evaluation environment, the results and analysis.

Chapter 2

Related Work

2.1 Social TV

The research in the area of social TV have met different approaches and it is important to take a close look on the previous work. Some research focus on doing simple field trials with low-level prototypes and disregarding the technical aspects of the technology used. They are interested on observing viewers behaviors and get insights about that, adding the viewers opinion and reactions. Ducheneaut *et al.* studied how people interact in front of a TV set and proposed guidelines for future social TV applications [8]. They found that viewers use TV as an excuse to invite someone to come over their homes and socialize. The program genre has impact on encouraging comments about the show [8] [6]. It depends on the rhythm and if it has redundant content or if it is people-centered. They also state that poor quality movies can be a reason to foster social interaction. As so, the selected content has strong impact on how much comments are made about a show. Across that, people behave similarly whether collocated or not. As expected, the results show that viewers enjoy communicating with each other during a show. Here, it was also observed that the social interactions occur in the gaps and are ruled by short sentences. In these gaps, newcomers can catch up what they miss and ask someone to do a quick recapitulation. It also confirmed that group viewing creates a pressure on leaving the TV tuned in the same channel. This study categorizes social comments, where the more impacting ones are content-based, context-based and non-sequitur and phatic. The latter is very simple but vital to the social atmosphere in the room, whether the non-sequitur negatively affect the viewing experience and distract them form the show context. Thus, phatic and comments related to the show are the type of comments suited for social TV use. As guidelines, support for the timing of social interaction and avoiding distractions form the TV screen are proposed. Years later, Harboe *et al.* tested the effect of audio communication in parallel with TV watching [10]. Here, they found that most of the participants would prefer audio communication instead of text typing or video calls to talk about a show. Here, it was observed that viewers do not feel alone anymore when having

this type of experience. It also gives them opportunity to comment about commercials and help to eliminate boredom. This communication is perceived as a source of extra content, as viewers can help each other sharing program facts. A threat to these systems is the intrusion of outliers in the family environment. Most research in social TV is about analyzing the state of the art and proposing new research agenda or trying to predict the future trends. Some describe the features from the existing products and others simplify the problem in minor problems by categorization. Cesar and Geerts [6] presented four major activity categories: Content selection and sharing, communication, community building and status update. In 2008 it was already perceived the need to use new interactions models [5], such as gestures, paper, voice, everyday objects and second screen. The remote is inadequate and there is a need for the use of physical artifacts. However, there were stated some concerns about the privacy and security of social TV, regarding the information about viewing habits. An interesting insight is that different studies have different results and opinions about the disturbing factor of voice communication. Natural ways to interact, hand-held devices as remote controls and sharing fragments were predicted as the future topics to research. Thus, sharing TV moments were considered important to the success of social TV, along with presence awareness. On the latter topic, ambient displays have been tested [11], using an electronic object to give information about the presence of friends on social TV with a light scheme. The Ambient Orb is a color changing lamp configured to give information about the number of buddies on the social TV network, that are watching television. Researchers asked participants to choose any room of their house to place the object. Besides this information, this object had the intent of attract viewers to use the system, specially when a friend invited them to watch a show together. They did this by a pulsing light scheme. As far as results are concerned, users understood the technology and were attracted to the display and most of the invitations were made to people who were already watching TV. Some viewers stated that the orb had few information and it was not engaging because people could not know who were the people online and what they were watching. Therefore, the object needed to have a strong link to the first screen. Here, the acceptance of a friend's invite to watch a show leads to an interesting automatic channel switch. A good insight about the acceptance of such social presence awareness objects is that participants reported a friendly feeling because they felt like someone were at home with them. Recently, research showed that TV viewing habits are changing and that a second screen increases the Twitter activity during live transmissions [13], but that the introduction of social media features has to concern about using sharable content that is relevant to people that are not watching the show [9]. Across that, there must be a balance between engagement and distraction [9].

2.2 Second Screen Flaws

With the buzz of second screen applications, many argued that the second screen would capture the attention from the main screen. Some proper research have been made [12] and discovered that about 30% of the total visual attention was to the second screen, whilst watching TV at the same time. This alerts to the need of simplifying the UIs, even regarding the multitasking generations. Besides that, in 2013, Jason Ary, director of strategy at Double Encore, wrote “The 5 things Keeping Second Screen From the Masses”¹. The first one points out the purpose of second screen. Second screen applications were born out of boredom and defined by viewers and, as so, these applications should focus on eliminate boredom while interacting with TV. Further that, features like social networking or send emails are already a second screen habit that viewers enjoy. The second reason refers to the exaggeration of information in the second screen. This requires constant focus and “brings a conflicting lean-forward user experience to a lean-back environment”. The next tone says that the fragmentation of this technologies is a big issue and it’s getting worse. Jason says that “every show seemingly has its own second screen application these days, but no consumer is going to learn/use 20 different apps for a good Second Screen experience” Hence, is important to have applications that can be used to all TV content with one UI and one experience. In addition, “there are few offerings like this and they are very reliant on ACR technologies hat are not overly user-friendly”. Here, Jason finishes by stating that these applications “need to talk directly with set-top boxes to figure out what the user is watching consistently and without user input”. The fourth one relies on the lack of cable providers involvement. Lastly, is said that there is a need for more tablets in homes. Challenging this last statement, surveys are showing that smartphones and tablets usage, whilst watching TV, is quickly increasing². This way, there is an opportunity to learn from the errors made on previous second screen products. A lean-back and less distracting experience are required, in order to focus the viewer attention on the main screen, having the company of a complementary tool to socialize about the show. Furthermore, a direct communication with the set-top-box is required, in order to get a smooth recognition of the show being watched.

2.3 Time-Shifting

Recent studies are showing that the concept of time-shifting on TV is getting more and more usage^{3 4}. Time shifting can be defined as a viewer pulling back in time a recorded

¹<http://http://www.doubleencore.com/2013/04/5-reasons-why-second-screen-has-yet-to-go-mainstream>

²http://www.ericsson.com/res/docs/2012/consumerlab/tv_video_consumerlab_report.pdf

³http://economico.sapo.pt/noticias/consumo-de-tv-em-portugal-ao-nivel-dos-paises-mais-desenvolvidos-do-mundo_200962

⁴<http://www.ericsson.com/res/docs/2014/consumerlab/tv-media-2014-ericsson-consumerlab.pdf>.

show to watch on TV, at a time more convenient to the consumer. Technologies that allow this technology, even when users forgot to record, are getting more and more usage and social interactions for them are also needed, in order avoid leaving aside a big slice of TV viewers. If it's possible to follow comments of a show in real time, it should be possible to have that experience also on time-shift mode. It's already possible to observe this feature integrated with online content providers (Tomorrowish) but there very few exploits of this feature for TV systems. Recent research says that second screen applications should not only offer synchronized experiences but also when watching delayed [9].

2.4 Implementations

2.4.1 Prototypes

WeOnTv [1]

WeOnTv is a social application project tested on a popular IPTV System in Portugal. It supports communication around TV content based on Instant Messaging services. It also allows to know what other users are watching and recommend a channel. Once a recommendation is accepted, the system switches the channel. The messages can be sent using the remote or a mobile phone and there are message models with genre categorization. The application offers two modes. On the TV Mode the TV content continue to be the focus but some notifications about the IM service are discretely displayed on the screen bottom. On the chat mode, the purpose is to discuss the show with other users and manage social settings.

FanFeeds [3]

FanFeeds is a second screen application prototype to author and consume synchronized extra content around TV shows. Here, the Facebook social graph is retrieved to get user's friends and to discover them on the app. After a manual check-in on episode page, the user can view comments, like or reply comments and author new comments. A time ticker is presented representing the shows timeline. When the user is authoring a comment, it can attach an external piece of content through a built-in Google and Youtube search console or an image URL copy paste. The user can choose if they want to share with everyone or just friends. Comments can be sorted by absolute time, number of likes and be filtered by friends or everyone. It shows a "All Feeds" page to present all type of activity (likes, comments, check-in, replies) from all user on all shows, in order to get a feel for trending shows. The study recognizes the need to integrate with existing social networks and not try to be a new one in place to avoid the burden of adoption. Even so, field trial participants said that the show-centric feature was different and better than Facebook or Twitter. On the trial results, researchers found that most frequent activities and consumptions were

the comments and replies. Four important insights are presented from the field trial. 1. Users prefer to consume more information rather than create information for their friends. It's important to know who else is on the application and gamification was meaningful. 2. People experienced the time-shift social feature on their will and enjoyed. 3. Creation of posts was still viewed as distraction. 4. auto-synchronization is need between the second screen and the primary screen.

Television Meets Facebook[2]

This project consists on an application connecting a Digital Video Recorder(DVR) with the Facebook social network. The user can automatically watch the shows liked by friends and transmit viewing data to the social network. The application grabs the user's friends, their TV interests and viewing data. It also grabs TV guide schedule and rating from the Web. Here, the interface shows the setup data for the DVR, the listing of friend's favorite TV shows and movies and the listing of shows previously recorded and the ones to be recorded. In addition, there is another component on the DVR, allowing to record and rate TV shows.

2.4.2 Products

NOS - Facebook na Iris (Portugal) ⁵

Connects a Facebook account to the user profile in NOS STB. The association set up is made by website. In the STB, the user enters the profile PIN. If the Facebook account is linked to the profile and the profile is active, social network features will appear. On surfing the VOD films or TV show menu, Facebook like and recommend buttons will appear and let the user post a like or recommendation of a show to Facebook.

NOS - Iris Remote (Portugal) ⁶

NOS mobile application for remote control of NOS STB. Offers full menu navigation by swipes, fast zapping by channel icon list, numeric keyboard and TV shows search button. Search automatically for STBs on the Wi-Fi network. No need to interact directly with STB.

⁵<http://nos.pt/particulares/ajuda/equipamentos-servicos/televisao/iris/Pages/Facebook-e-perfil-Iris.aspx>

⁶<http://nos.pt/particulares/ajuda/equipamentos-servicos/televisao/iris/aplicacoes/Pages/Iris-remote.aspx>

NOS- Iris Share (Portugal) ⁷

NOS mobile and STB application. Allows to display pictures stored in the mobile the device, on the STB. Searches automatically for STBs on Wi-fi network. Requires direct interaction with STB to launch the IRIS Share application on the STB.

Meo - MYCONTENTS@TV – Facebook (Portugal) ⁸

STB application. Share interests and follow up your Facebook friends. Allows to follow the user's news feed, wall and friend's profile. Show photos in slide-show mode. "Like" feature. Share favorite TV shows with friends, publishing on Facebook with "What I'm watching" and "I like this movie" features. When a friend sees these posts it can click and tune in the channel(TV use) or access the MEO MAGAZINE, a TV Guide application on Facebook(Web use). The instructions for associate the Facebook account to the STB are sent by email and it's required a PIN to access the application on STB(can be memorized). It's possible to associate other Facebook accounts to the same STB. The account association it's valid for all STB on the client's contract. The publishing is only allowed on live TV shows and not on recorded shows or VOD. It has a session timeout of 2 hours where it's asks again for the PIN. Has a limit of 10 photo albums that can be accessed.

Meo - MYCONTENTS@TV - Twitter (Portugal) ⁹

STB app. Shows up on the TV the tweets and mentions. Allows to access followers and follows, search for trends, write and answer tweets, retweet e add to favorites. It also has a TV Mode to follow Twitter activity while watching TV. Steps to associate account: 1. Go to the STB application and get a PIN of the service activation. 2. Still on the TV, the user will be redirected to the Twitter's official website where it has to authenticate and give permissions to the application. 3. From there, the Twitter account it will be activated on the STB.

Vodafone – Canais Interactivos (Portugal) ¹⁰

STB application. When watching a show, press a red button on remote control and sidebar will appear with many options, including the Facebook option. Entering the Facebook option, the Facebook page of that TV channel will appear on the STB screen with the last 10 posts.

⁷<http://www.nos.pt/particulares/ajuda/equipamentos-servicos/televisao/iris/aplicacoes/Pages/Iris-share.aspx>

⁸<http://www.meo.pt/tv/experiencia-tv/aplicacoes-interativas#facebook>

⁹<http://www.meo.pt/tv/experiencia-tv/aplicacoes-interativas#twitter>

¹⁰<http://www.vodafone.pt/main/particulares/tv-net-voz/televisao/funcionalidades/canais-interactivos.html>

Vodafone – StartApps (Portugal) ¹¹

STB application to access Facebook, Twitter and Picassa contents, displayed on a sidebar.

CaboVisão – Twitter (Portugal) ¹²

STB application. Twitter account association by QR code or email. Follow tweets (last 20 written in Portuguese and sent in the TV application) and tweet about the channel in a side window next to the channel emission, share a screenshot of the TV show, add favorite hashtags, set the hashtags to follow, retweet. Figure 2.1 shows an example of how current cable TV operators integrate social networks with TV content.

CaboVisão – Facebook Integration on a STB (Portugal) ¹³

STB application. Facebook account association by QR code or email. Allows to post free text comment, post about the TV show the user is watching, view comments about that user's publications in a side window (continuing to display the TV content), share a screenshot (not available for adult content channels).

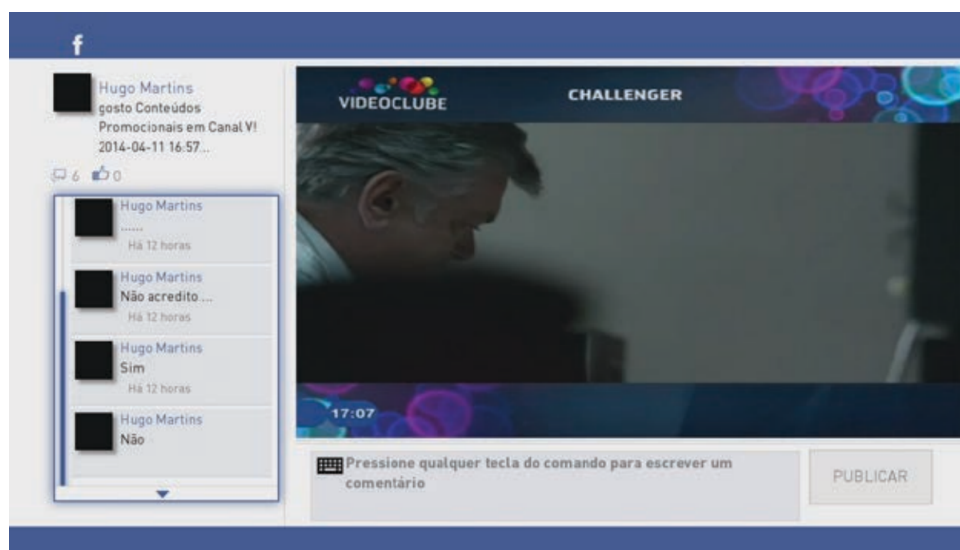


Figure 2.1: CaboVisão - Facebook

Tvtag(former GetGlue) (U.S.A.) ¹⁴

Web/mobile application with social network to manually check-in TV shows, lead a discussion, get recommendations, like, share and follow TV shows schedules. Users get

¹¹<http://www.vodafone.pt/main/particulares/tv-net-voz/televisao/funcionalidades/startapps.html>

¹²http://www.cabovisao.pt/resources/onebox_manual_facebook.pdf

¹³http://www.cabovisao.pt/resources/onebox_manual_twitter.pdf

¹⁴<http://tvtag.com>

stickers to reward and engage them. It shows trending shows (most watched). Collects movies and television interests on Facebook account and use them to personalize the feed filters. Gives status to users, according to their fan level. On the mobile application, it's possible to take a screenshot, make a drawable annotation and share it with friends. The user has some action to adapt his recommendations: like/unlike recommendation, check-in, save, like, “not for me” and “add review” for each episode of a show.

Miso SideShows (currently merged with Viggle Inc) (U.S.A.)

Mobile application to get content trivia, facts about actors, clothes and songs. The user checks in the show he's watching either on the application or in the browser. To do this, it has to pick the show and the episode being watched and press check-in button. In addition, this check-in can be published on Facebook with a comment. The user can do a generic comment about the show also. A timeline is displayed with all check-ins from other users and their comments. The user is rewarded for his social interactions with points and leader boards are announced. Miso allows trusted users to edit and curate show information. It has a table with currently trending top for the most viewed shows.

Viggle Inc (U.S.A.)¹⁵

Mobile application. Audio recognition for TV show check-in. Users get points for watch TV shows that can be redeemed for music downloads, gift cards and others. Engaging with TV shows by RT quizzes, polls and trivia. Football predicting games. Awards for loyalty to Series and Genres. Entertainment Marketing and Rewards Company. It integrates show pages and network applications where they offer social media feed (Facebook, Twitter, Tumblr, Reddit, Pinterest) and show related pages (IMDB, Wikipedia). On the left side of Figure 2.2 is shown how Viggle shows the Twitter Stream. On the right side of the same image, royal points are presented, also as the check-in button.

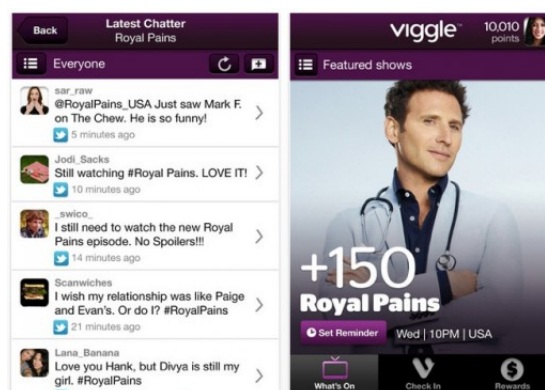


Figure 2.2: Viggle - Mobile application with Second Screen and Social TV features

¹⁵<http://www.vigglestore.com/tv>

Beamly(former Zeebox) (U.K. And U.S.A.) ¹⁶

Web/mobile application for social networking television topics. The user signs up and give his interests like favorite tv genres, tv shows and celebrities. This info is used to show personalized feeds. Allows chat rooms about a show and have news feed about shows. It's possible to follow celebrities and play trivia games. Shows a guide with twitter buzz and audience meter. The Twitter stream has cast activity.

ChatterBox (Thailand) ¹⁷

Mobile application for social and interactive TV. Allows to connect with strangers and talk about the show in chat rooms, follows last tweets, get alerts for favorite tv shows, play games and get rewarded for that as for social interactions. It also has an interactive component for real time polls.

Fan TV (U.S.A.) ¹⁸

Gets Facebook movie interests and add them to a "I am a fan" list. Allows to create a watch-list and get recommendation for movies. It's integrated with a touch remote control and a fancy interface.

CutX – TV Memo (China) ¹⁹

This application allows users to grab a screenshot or a clip from the TV and share them on social networks with comments.

Peel (U.S.A.) ²⁰

Smart Remote on mobile device. Browses TV shows info by genre. Tags favorite shows. Share with friends what we're watching on Facebook and Twitter. Find and Follow friends to see their Favorites, Guilty Pleasures and recommendations. Posts have a link that allows your friends to click and automatically tune in the show you're watching. Peel allows to remote control the TV through the application but it requires an initial configuration.

¹⁶<http://us.beamly.com>

¹⁷<http://www.chatterbox.mobi>

¹⁸<https://www.fan.tv>

¹⁹<http://cutx.catchpo.tw>

²⁰<http://peel.com>

Relay TV (U.S.A.) ²¹

Mobile application. Shows the latest Tweets. Shows a timeline with the shows that your friends are watching and clicking on each line, you can enter a discussion room with them. In the chat, they can use voice.

Sidecastr (U.S.A.) ²²

Shows Tweets in real time. The tweets are curated and separated in categories. In sports, it has categories for players, plays and the game. The sync is made by audio recognition. To fully work, it's request to not talk during the sync and not to sync during commercials or theme music that repeats on multiple episodes. The tweets are showed in a 'sidecast'. A sidecast is a top level comment with the tweet, the user photo from Twitter and a background screenshot. The categories are their secret sauce as they say. You can remove unwanted topics and prioritize others. They also introduce Smart Categories like My Sidecasts, My Thumbs Up, My Starred and, in the future, My Friends Sidecasts. When a sidecast is clicked, it's possible to see like, unlike, flag as inappropriate. Sidecastr offers auto scrollable comments list. When you post a sidecast, you have to pick you category of comment. It's also possible to go back in time of the show and check earlier sidecasts.

Snappy TV (U.S.A.) ²³

Allows to grab a clip from a live show and share it in Facebook, Twitter ad Google+ and SnappyTV site with a comment. On the SnappyTV site, other users can see the snap and view the number of views for each highlight as share them in social networks.

Tomorowish (U.S.A.) ²⁴

Social media integration and time-delayed playback. Tomorowish has a special sauce in their machine for comments curation. They do semantic analysis of the conversation that is relevant, removing the clutter of check-ins and similar comments, in order to engage the audience. This curation also measure popularity, language, sentiment and use blacklisting and white-listing.

Tunerfish (U.S.A.) ²⁵

Social network about television. Each show/movie has it's page and you can click say that you're watching it and reveal that you are a fan. These actions can be combined with

²¹<http://www.relaytv.com>

²²<http://www.sidecastr.com>

²³<http://www.snappytv.com>

²⁴<http://www.tomorrowish.com>

²⁵<http://www.tunerfish.com>

textual comment and shared on Facebook and Twitter. A sticker reward mode is used. In each show page is possible to check recent activity for all users in the application, check their interests and follow them. On the home page, we can see the shows with more fans, a trending indicator.

Nantmobile - IDTV (former TVTak) (U.S.A.) ²⁶

Second screen application for automatic video recognition and display of polls, trivia questions, and shopping. The recognition is based on a video capture, on which the user points the camera to the TV.

Tweek (Germany) ²⁷

iOS application for TV content discovery and sharing. TV guide recommendation. The recommendation algorithm is based on the social graph. Hence, you can check what your friends recommend, either through like button or a direct tweet. The Home Screen shows the activity for the friends you're following.

Yap TV (U.S.A.) ²⁸

Social TV Guide. Allows to get instant TV recommendations, personalized TV news feed, photos and videos for any show and check what friends watch. Shows the Twitter stream. Presents polls. Allows to chat in private rooms and check what your friends are watching.

Summary

After a close look at the current offer in the social TV and second screen areas, it is possible to get an idea of the typical features in this type of applications. Table 2.3 resumes the main products and its features related to the topics on this work. Some insights are described in the following lines. First of all, it can be observed that the mobile environment is the leader platform for this type of solution, mainly because of its lean-back experience. As so, social networks integration with the TV content as a STB application is overdued, but remain as the current offer by the main IPTV providers in Portugal. As mentioned before, the usability of this type of systems is quite complex and removes the viewer from the TV content. In the STB applications group, CaboVisão is the one who offers a deeper social integration. In the pursuit of alternatives to support this type of applications, Web environment is not the main and only option. Currently, these applications are provided as a Web application to complement or replicate the mobile one. As features is concerned,

²⁶<http://nantmobile.com>

²⁷<http://www.tweek.tv>

²⁸<http://yapstudios.com>

	Products																		
	NOS	MEO	Vodafone	CaboVisão	Viggle	Tvrag	Beamly	ChatterBox	Fan TV	CutX	Peel	Relay TV	Sidecastr	Snappy TV	Tomorrowish	Tunerfish	TVTak – iDTV	Tweek	Yap TV
Platform																			
Set-Top-Box	X	X	X	X															
Web						X	X		X					X	X	X			
Mobile					X	X	X	X	X	X	X	X	X	X			X	X	X
Features																			
Facebook integration	X	X	X	X	X	X			X		X	X		X		X			
Twitter integration		X	X	X	X		X	X			X	X	X	X	X				
Publish what is being watched	X	X		X	X	X					X					X			
Text Comment		X		X		X	X			X	X		X		X	X		X	
Social media feed filtered by show			X	X	X	X	X	X				X		X	X				X
Social network feed with time delayed													X		X				
Friends activity						X					X	X				X		X	X
Chat on TV show room				X	X	X	X	X				X		X					X
Collect interests on Facebook						X			X										
Check in (Manual)		X	X	X		X	X								X	X			
Check in (Auto)					X								X				X		
Recommendations	X					X			X			X						X	
Share screenshot				X		X				X									
Share TV clip														X					
Trivia Quizes / RT Polls					X		X	X									X		X
Reminders					X		X	X	X							X			
Rewards for social activity					X			X											
Tune in a TV channel through the app											X	X							
Stickers						X										X			
Favorite shows feed						X	X												
Show pages curated by fans					X								X						
Web info retrieval (IMDB, Wikipedia)							X												
Exchange points for gifts					X														

Figure 2.3: State of the Art - Social TV Products

to comment a TV show, or enter a chat to discuss it, is the most popular feature because it is the essence of being social. These comments can be integrated with a social network like Facebook or Twitter as a social feed or try to create its own social network, which restricts the range of people who “hears about” the application. Along these lines, publishing to these social networks what the user is watching it is also quite common, as people need to share this information to express themselves or just for bringing someone to watch the show with them. The check-in feature refers to the action of the application registering the show you are watching. It is a simple but key feature for simplifying the user experience. This is normally done by a manual action of picking a show from a list, but there are some technologies allowing to this automatically by audio(Audio Content Recognition(ACR)) or video recognition(Video Content Recognition(VCR)), using the device microphone and the camera shooter, respectively. There is no application getting this information directly from the main source, the TV system. Along with the common features, trivia quizzes, real time polls and point rewards are used to evict boredom and used as the main features to engage the user. This type of gaming creates more complexity for the user, and can be seen as the main reason for people spend more time looking at the second screen instead of watching the TV content. It is also possible to observe that sharing TV clips it is an interesting feature, that would be important to engage users, but it is restricted to the content copyright and, as so, there is little supply of that feature.

2.5 Android

2.5.1 Material Design ²⁹

In 2014, Google has revealed a new design specification for Android, Chrome OS and the Web, which they called “Material Design”. The main goal is “to create a visual language for our users that synthesizes the classic principles of good design”. Material design brings an interface in line with reality, emphasizing elements, transitions, and animations as if in real life. Moreover, “material” is accentuated with vibrant neon colors and large bold squares. Google wants to create a design standard, making developers to create applications with less disparate styles and unifying and consistent style. As so, from this point, Google has been slowly revealing tools and guidelines for developers to create applications in line with their style, but until this date, there is little native support for it.

Floating Action Button ³⁰

The floating action button is one of the novelties that Material Design brought. This consists in a round button, normally with a vibrant color and a white plus sign inside

²⁹<http://www.google.com/design/spec/>

³⁰<http://www.google.com/design/spec/components/buttons-floating-action-button.html>

which turns into a white cross when pressed. The important is that this button is used to promote an action. As so, this button is recommended to be used when a developer wants to highlight the primary action of an Android application. This can be also used in a contacts context.

2.5.2 Libraries

Nowadays, every developer have to leverage the open source community in projects like Github ³¹ and that was done on the implementation phase. Given the Material Design specification referred in the previous section, I had to search for libraries to support such specification, some official and, in most cases, non-official libraries. As so, the following libraries were important to support the development of the mobile application:

- v4 Support ³² - As mentioned in the official documentation: "This library is designed to be used with Android 1.6 (API level 4) and higher. It includes the largest set of APIs compared to the other libraries, including support for application components, user interface features, accessibility, data handling, network connectivity, and programming utilities."
- v7 Appcompat ³² - As mentioned in the official documentation: "This library adds support for the Action Bar user interface design pattern. This library includes support for material design user interface implementations." This library depend of the v4 Support Library.
- v7 Cardview ³² - As mentioned in the official documentation: "This library adds support for the CardView widget, which lets you show information inside cards that have a consistent look on any application. These cards are useful for material design implementations, and are used extensively in layouts for TV applications."
- v7 Recyclerview ³² - As mentioned in the official documentation: "The recyclerview library adds the RecyclerView class. This class provides support for the RecyclerView widget, a view for efficiently displaying large data sets by providing a limited window of data items."
- Android Asynchronous Http Client ³³ - Essentially, it is a callback-based HTTP client library for Android.
- Facebook SDK for Android ³⁴ - This SDK is fundamental to integrate any Android application with Facebook. It allows to login with Facebook, share content from

³¹<https://github.com>

³²<https://developer.android.com/tools/support-library/features.html>

³³<https://github.com/loopj/android-async-http>

³⁴<https://developers.facebook.com/docs/android>

the application to Facebook, read and write to the Graph API, the primary way to get and set data on Facebook's social graph.

- Picasso³⁵ - A library for a simple and agile image downloading, caching and transformation.
- MaterialDesign³⁶ - A library with UI components of Android L, meeting the Material Design standard.
- Materialish-progress³⁷ - A progress wheel with the Material Design style.
- android-floating-action-button - A project with the implementation of the Floating Action Button for Android based on the Material Design.
- HeaderListView³⁸ - An implementation of a ListView component "with a cool iOS-like "sticky" section headers".
- Superlistview³⁹ - An easier way to use a ListView, specially when implementing UI elements for content loading and refresh.
- Horizontallistview⁴⁰ - This library include is based on the Google code for the ListView component and implements an horizontal scrollable list.

2.6 Graph Data Bases

A graph is essentially a set of nodes and the relationships that connect them. A graph database is defined as an online database management system with CRUD methods exposing a graph data model. The graph data model is described with the followings characteristics: it contains nodes and relationships; nodes have properties (key-value pairs); relationships are labeled, can have properties, are always directed and connect a start to an end node.

The main advantages of this type of database are Performance, Flexibility and Agility. Comparing with traditional relational databases or NOSQL stores, there is a strong increase of performance when using this type of database when dealing with connected data. This difference is evident when the data-set turns large and the intensive join queries performance get worst. This happens because the execution time is proportional to the size of the portion of graph used to satisfy the query, rather than the size of the entire

³⁵<http://square.github.io/picasso/>

³⁶<https://github.com/navasmdc/MaterialDesignLibrary>

³⁷<https://github.com/pnikosis/materialish-progress>

³⁸<http://applidium.github.io/HeaderListView>

³⁹<https://github.com/dommerq/SuperListview>

⁴⁰<https://github.com/sephiroth74/HorizontalVariableListView>

graph. Besides that, graphs are naturally addictive, which gives this model a strong Flexibility. It is possible to add new kinds of nodes and relationships to an existing structure without making changes to previous structure and disturbing the application functionality. This reduces the project's risk and increases productivity because developers do not have to model the entire domain in detail ahead in time. Another power of graph databases is Agility. The schema-free nature of the graph data model allows the developer to be aligned with today's incremental and iterative software delivery practices, as with agile and test-driven development practices.

2.6.1 Neo4J

Neo4J is the most popular graph database management system, accordingly to the DB-Engines Ranking of Graph DBMS ⁴¹. This may explained with the fact being the only transactional database with the required combination of performance, trustability and native graph storage with native graph processing, graph scalability and high availability. Neo4J is an open source graph database that features native graph storage and processing for fast reads and writes with ACID compliance. Neo4j is accessible from most programming languages using its RESTful Web API interface and comes with a Web-based administration tool that includes a visual node-link graph explorer. Some typical applications of Neo4J are real-time recommendations, social graphs, fraud detections, graph-based search and network and IT operations. It has well known customers as eBay, Walmart, Cisco, HP or TomTom.

The core features of Neo4J are ACID transactions, high availability, scalability, high speed querying and a declarative graph query language.

Cypher Query Language

Cypher Query language is the declarative graph query language introduced in Neo4j. It allows expressive and efficient querying and updating of the graph store. Cypher is designed to be simple (more than traditional query languages like SQL) but still very powerful, allowing to express very complicated queries. This language it optimized for reading and not for writing, making query optimization an implementation detail because do not require traversal updates just because the physical database as changed. It is easy to use because it matches the intuitive way of describing graphs through diagrams.

A simple example of these diagrams is presented in the Figure 2.4 where its equivalent ASCII art representation in Cypher is (a)-[:KNOWS]->(b)-[:KNOWS]->(c), (a)-[:KNOWS]->(c).

⁴¹<http://db-engines.com/en/ranking/graph+dbms>

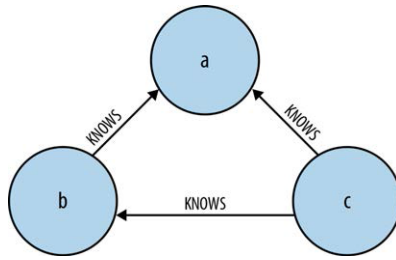


Figure 2.4: Graph expression representation in a diagram

Cypher is inspired in SQL, given that queries are built using various clauses. Clauses are chained together and they feed intermediate results between them.

Considering the related work, it is clear to verify that there is the need of a new solution to integrate social networks with TV content in a way that users create engagement. Integration of social networks on TV or STB shall be dismissed due to usability and design space issues, yielding to mobile applications on a second screen, specially tablets. Social networks features must not be replicated in and only some of them are TV related. Hence, focus on the main screen, the TV, must be achieved.

As so, the next Chapter introduces Social Zapping, a mobile application which aims to create more user engagement using a Second Screen application with Social TV features in a way where usability, design and features are appreciated by TV viewers.

Chapter 3

Social Zapping

After reviewing the related work and analysing the current studies and commercial solutions I started to build the proof of concept that is possible to increase users engagement in applications which integrate social networks with TV content.

This chapter describes how this proof of concept application was analysed, designed and developed.

3.1 Requirements

The main purpose of this application is to engage the TV viewers in a second screen application, offering them a simple integration of their Facebook account with TV content and a social recommendation system. To accomplish that I have defined the following requirements for the application:

3.1.1 Non Functional Requirements

Table 3.1.1 shows the non functional requirements for this application.

Requirement	Description
NFR 1	Effectiveness
NFR 2	Emotional factors (e.g. fun)
NFR 3	Performance / Response time
NFR 4	Robustness (no crashes)
NFR 5	Testability (easy to evaluate)
NFR 6	Usability by target users
NFR 7	User Friendliness

Table 3.1: Non Functional Requirements (NFR)

3.1.2 Functional Requirements

Table 3.2 lists the functional requirements for this application.

Requirement	Description
FR 1	Login with Facebook
FR 2	Collect Facebook interests in TV content
FR 3	Auto-Recognition of the TV show tuned on the STB
FR 4	Display of the current show on the first screen
FR 5	Recommend a show to a friend, if and only if the user is watching the show
FR 6	Follow a recommendation from a friend, only and only if the user started to watch the TV content recommended
FR 7	Present feedback about the impact of the user recommendations with top followers, best recommendations and information about all the recommendations done
FR 8	Present the friends' recommendations in a timeline where each entry has a play button to immediately start watching the recommendation and set the recommendation to the "Followed" status
FR 9	Maintain the information about the TV show tuned in the STB always present in every screen
FR 10	Give access to the recommendation function right on the first screen
FR 11	When showing the list of friends available to recommend, filter them by groups like "Most recently recommended" to speed up the process of picking up a friend to recommend
FR 12	Display what user's friends are watching in the moment
FR 13	Display the trends in TV shows popularity like an EPG, considering the live shows and guiding the TV viewer on what to watch
FR 14	Manage a ranking system for TV shows and users who contribute to the recommendation system
FR 15	Display recommended a specific show
FR 16	Display a simple friend profile, showing what he is watching, his ranking and the historic of recommendations made and received from that friend

Table 3.2: Functional Requirements (FR)

3.2 Design

In this phase, the goal was to start drawing sketches for the second screen application in order to get a visual support of what would be the final product. More, this sketches helped to think about the information to display. As so, the physical design of the back-end starts with ideas raised from prototyping. First of all, the interfaces for the application screens have been designed on paper just to get a quick visualization of the idea. Here, there was no need to have a good sketch. It is just a simple and unaffected way of storing the idea. Later, with the a more clear and organized idea, considering the navigation flow and layout aspects, I've build a prototype. Here, there was a large concern about the design

itself. Text fonts, buttons size, shapes, color patterns and data disposition on the display were addressed. On that way, I have used an online tool, FluidUI ¹, for drawing mobile prototypes, based on HTML. This kind of prototype can be considered as a medium-high fidelity prototype. This prototype used realistic data. Fields like the name of the live show, top shows, channel names, comments about the show, Facebook friends and Twitter usernames were real. Using realistic data is important to avoid users distraction and confusion when testing the prototype. In the beginning of this design phase, there was a need to drink some inspiration from others. Hence, the design process started with a search for the Android Design trends at the moment and looking at the most famous and inspirational examples ^{1 2 3 4}.

3.2.1 1st Prototype

The main concern was to offer the most efficient way to use the core social TV features. As so, the first concern was the navigation flow and constant awareness of the core features on the main screen. Also, there was a concern to maintain the name of show tuned on TV along the screens. The interface has buttons to access all the features at any screen, being them large or minimized. When the buttons are large, an image of the show is displayed on the back of the menu buttons in a translucent effect. Furthermore, a bar with the name of show and channel is always visible. Here, I've decided that was important to associate each feature to an icon all the time, in order to make the user memorize where he can access each feature, even when there is no text near the access button. This is specially important due to the decision of allowing the user to minimize the menu, with a swipe, to a top bar on the screen, named the "Navigation Bar". The advantage is to get more display space to present the content. See Figure 3.1 to get a feeling about the core features access. The access to the core features is persistent on the main menu. The options on this menu are:

- "Comentários" - Comments on Twitter about the live show tuned on the STB. See Figure 3.1. To Tweet there is a blue button on the bottom leading to the screen on Figure 3.2.

¹<https://www.fluidui.com/>

¹<https://developer.android.com/design/material/index.html>

²<http://android.inspired-ui.com/>

³<http://androidniceties.tumblr.com/>

⁴<http://www.android-app-patterns.com/>

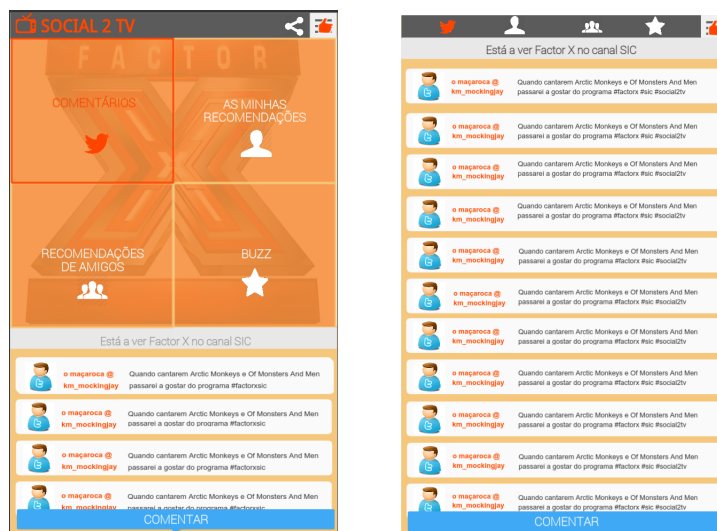


Figure 3.1: 1st Prototype - Home (Comments) and Minimized Menu



Figure 3.2: 1st Prototype - Comment TV Show on Twitter

- "As Minhas Recomendações" - It shows the influence of the TV recommendations made by the user on his friends. See Figure 3.3.



Figure 3.3: 1st Prototype - My Recommendations

- "Recomendações de Amigos" - It presents a Feed of show recommendations made by friends. See Figure 3.4.



Figure 3.4: 1st Prototype - Friends Recommendations

- "Buzz" - Guide the TV viewer on picking a TV content to watch, showing the top shows regarding their social buzz in the viewer's social graph. See Figure 3.5.

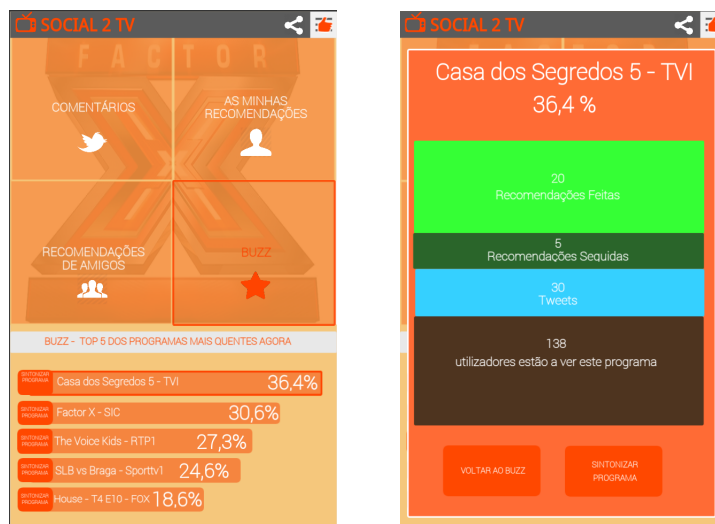


Figure 3.5: 1st Prototype - Buzz Screen and Show Buzz Detail

These features are presented with four large buttons on a rectangular shape, occupying half of the screen but can be minimized to the navigation bar with a simple swipe up. The other critical feature is to recommend TV shows to friends. The concern was to offer access to this feature with a maximum distance of two clicks, trying to preserve the focus on the main screen. Thus, there is always a button on the right top of the screen to access the friends list, creating a sidebar with the list of friends and their status. Here, I had to decide how to show the user which friends are offline, online (watching other show) and online and watching the same show (See Figure 3.6).



Figure 3.6: 1st Prototype - Recommendation

The first decision was about how to deal with friends that are watching the same TV content. It makes no sense to recommend a show to a friend who is already watching it

because does not have added value unless you want to tell what you are watching, but this was not on the scope of friends recommendation. That way, the list of friends available to recommend a show should not include the friends who are already watching the show. But how to know that my friend is watching the same content that me? The feature of telling and find what friends are watching was not ignored but designed to be less evasive. Here, I have thought on two solutions. The first is to show the friends watching the same show together with the offline and online friends. Here, the area corresponding to a friend who is watching the same content is not clickable. Moreover, an icon is displayed to symbolize that he is watching the same show. The other solution is to exclude these friends from the sidebar list and show their names on a moving footer. This way, the sidebar can be dedicated to the friends available to recommend and the footer to check who is with you already. This decision is clearly a situation where an usability interview is helpful and, as so, it was postponed to after that interview. Regarding the friends status, a color scheme was addressed like on Facebook. Here, a fluorescent green symbolize online friends available to recommend, an opaque green represents friends watching the same content and the grey means offline.

3.2.2 2nd Prototype

Once the first prototype was finished, a quick demo was made to some Professors on my University, Francisco Couto and Tiago Guerreiro, experts on Information Systems. Using the first prototype as support, I have gathered their opinion and tested my navigation and content design. Here, a brainstorming as emerged and all the ideas were analysed and mashed up in a new solution for the application design. Hence, a second prototype was build. After the experience with the FluidUI on the first prototype, I have decided to use another prototype tool. FluidUI was great to build a prototype with few screens and low complexity on the navigation but I have realized the problems that my professors had to understand how to use the prototype. The main problem was that I have created clickable areas, with links to other screens, only on buttons needed to complete an usability test task. Even with the concern on showing notes pointing the clickable areas and highlighting them, it was hard to find where to click and how to use it. Hence, I have noticed that on a second prototype I had to make all the visible buttons linked to some screen, allowing totally free and exploratory navigation. Although being a quick way to build a basic prototype, FluidUI became to be hard to use when I started to add more detailed elements and complexity on the interface, specially on selecting elements near to others and editing their properties. Moreover, I have noticed some lack of consistency on the service and, as so, I decided to search for another tool. This time, I have selected a free trial of a prototyping tool with well known customers like Adobe or Google. Its name is Justinmind⁵. Justinmind caught my attention because it offers a desktop application with

⁵<http://www.justinmind.com/>

an interface similar to Adobe Photoshop, a tool that I have already used. Moreover, it has widgets for Android design on phone and tablet. Here, it was great to find that this Android libraries presented components with Google's Material Design specification. The latter was the key factor to choose this tool. The brainstorming output was a list of navigation and content design issues, ideas to solve them and general opinions to improve the solution. The insights from this demo are presented as follows.

- on the main menu, there was low contrast between the selected icon and its background
- the menu with large buttons occupies half of the screen, drives out the user's attention and the swipe up gesture to minimize it is not understood or useful
- the minimized menu needed icon labels because the user could not remember the feature related to each icon
- 'As minhas recomendações' - My recommendations Top indication feature was not predictable.
- the application icon on the navigation bar was not persistent along the screens transitions and represented a navigation problem
- the four features on the main menu were not all related to the show being watched on the TV and as so, it was noticed that should be designed a separation between what is the general statistic information, what is information related to the show being watched and what are the main actions for the user
- the idea of using an auto-scrollable footer displaying the friends who are watching the same show does not work because people do not want to wait for checking if a specific friend is watching the same TV content
- the word "Buzz" is a bad terminology because it leads to a trivia game with this name and, as so, users may think that they will enter in a trivia game about the TV show.
- content related to the show that is tuned, the most popular shows and what friends are watching should gain the main focus on the initial screen of the application
- nowadays, applications display content or a preview of it in the first screen. Users give very few seconds of their attention on the first usage after installing the application. The effort to engage should start right away with content exposure.
- it should be created a filter on the statistic information, allowing to pick if the results are based on the activity of favorites, all friends or overall users activity

- the initial screen could present a dashboard with cards representing TV shows or friends, the main concepts for the application. This also leverages the patterns defined on the Android Material Design: As referred in the related work chapter(2.5.2), "These cards are useful for material design implementations, and are used extensively in layouts for TV apps."
- features not related the live content like general statistics should be moved to a menu on the navigation bar or a side bar
- the recommend button should be easily available when the user is informed that a friend is watching some other show

Along with the issues and improvements I have just enumerated, I started do sketch the new ideas on paper. The paper sketches, like on the first prototype process, helped to quickly register the visual idea running on the designer head. After these sketches, I started to prototype with the Justinmind tool. Here, the libraries for Android were very helpful since they already had the Android Material Design components. Such components, specially the cards, were useful to speed up the prototyping and were perfect to get a close feeling of the final appearance.

3.2.3 Navigation Flow Diagram

Before starting to prototype with the referred tools and redefined ideas for the design. I have defined the navigation flow as can be observed in the Figure 3.7.

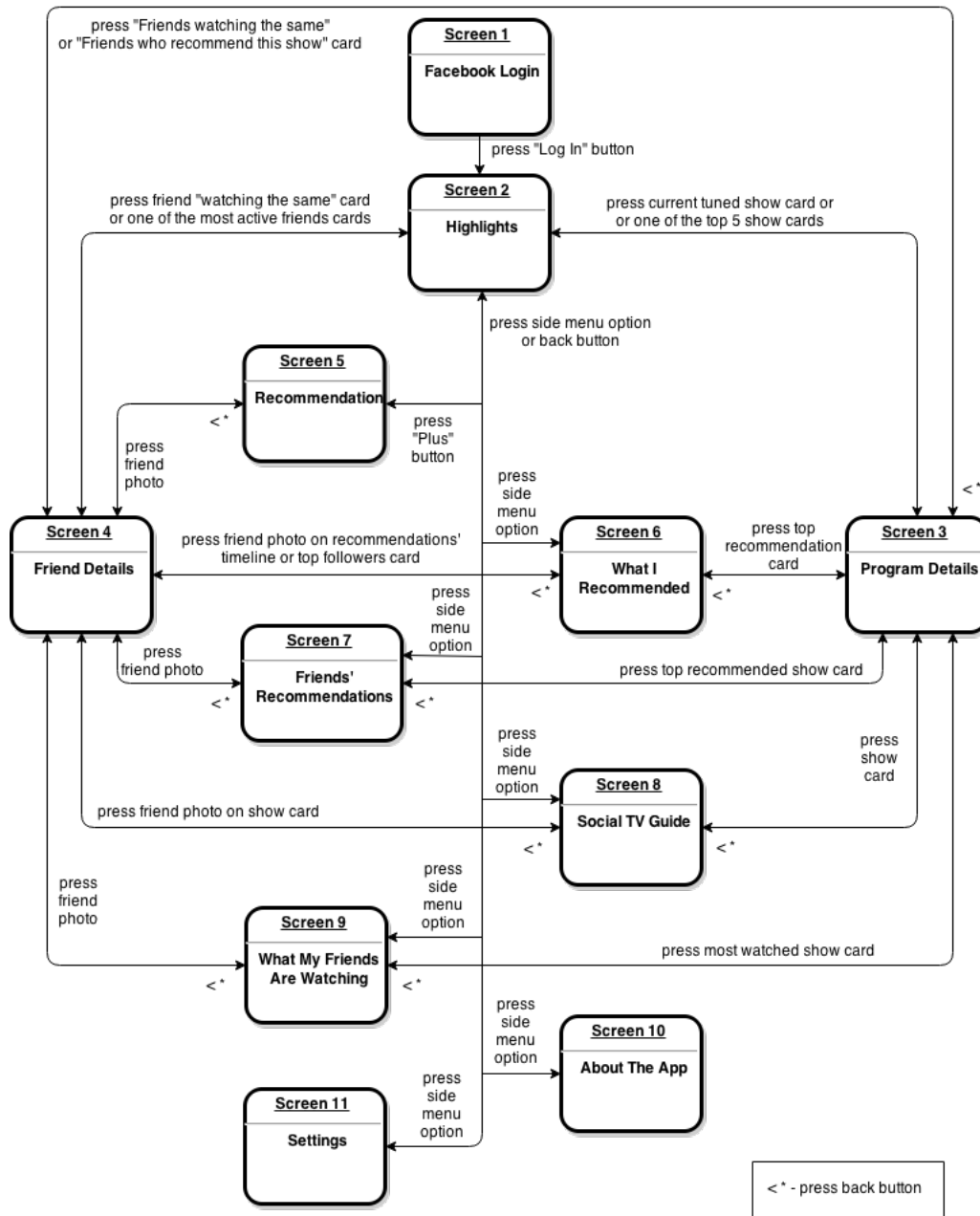


Figure 3.7: Navigation Flow Diagram

3.2.4 Output

The following images show the output of the final design mockup. This mockup is online (<http://socialtvprototype.alojamentogratico.com>) for anyone interested in interact with it and capture ideas from this design proposal.

Figure 3.8 shows the first screen after the login. Here, it is highlighted what the viewer is watching, what friends are watching and the top 5 most popular shows. Furthermore, there is an orange floating action button to perform recommendation.

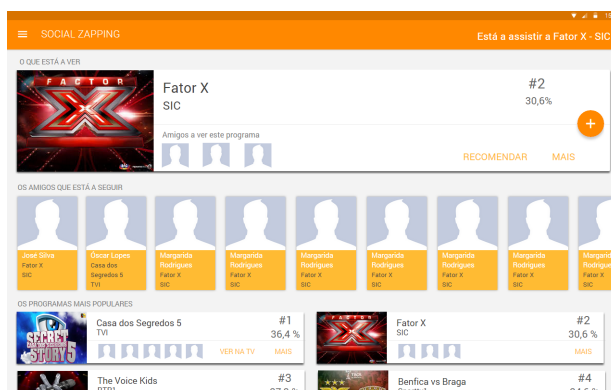


Figure 3.8: 2nd Prototype - Highlights

Figure 3.9 shows the side menu with its options: "Highlights", "My Recommendations", "Friends Recommendations", "TV Guide", "What my friends are watching".

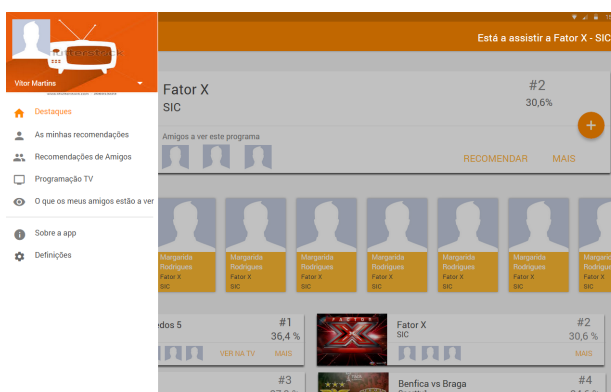


Figure 3.9: 2nd Prototype - Side Menu

Figure 3.10 presents how the floating action button expands when pressed.

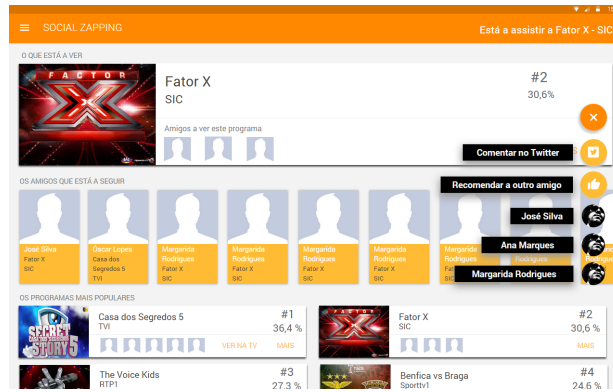


Figure 3.10: 2nd Prototype - Recommend Button

Figure 3.11 illustrates how the user pick a friend to recommend.

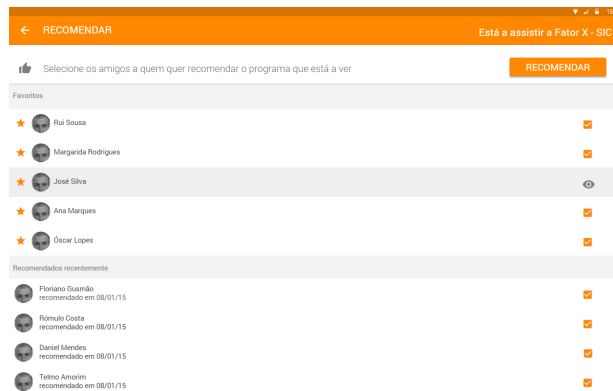


Figure 3.11: 2nd Prototype - Choose Friend To Recommend

Figure 3.12 displays TV show details along with information about friends watching and recommending the show.

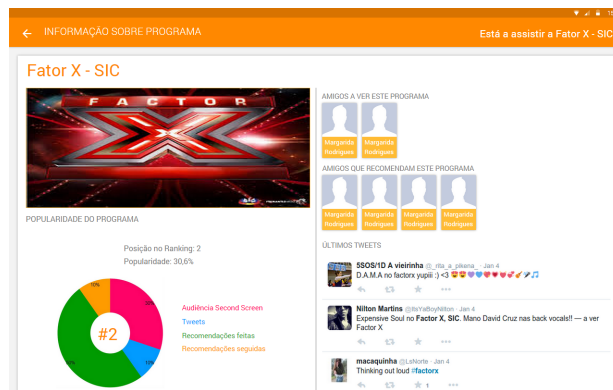


Figure 3.12: 2nd Prototype - Detail Of A TV Show

Figure 3.13 reveals how the friend profile was designed.

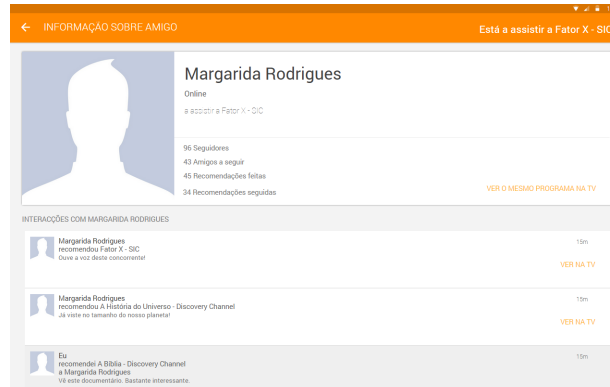


Figure 3.13: 2nd Prototype - Detail Of A Friend

Figure 3.14 shows how the recommendations made by the user are presented, highlighting top recommendations and top followers as user statistics.

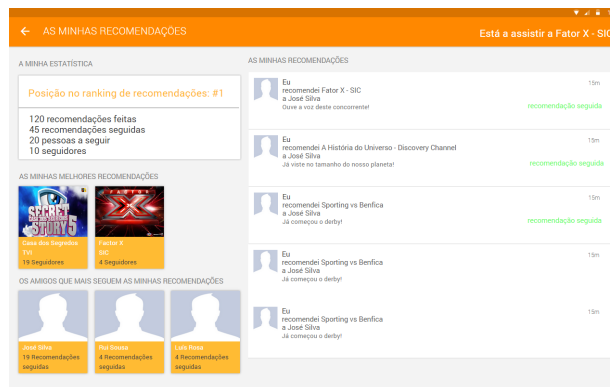


Figure 3.14: 2nd Prototype - What I Recommended

Figure 3.15 shows the friends recommendations, highlighting the top ones.

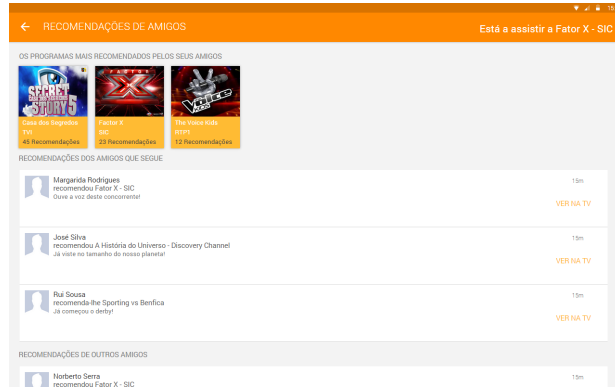


Figure 3.15: 2nd Prototype - Friends Recommendations

Figure 3.16 shows how the EPG, ordered by TV show popularity, was designed.

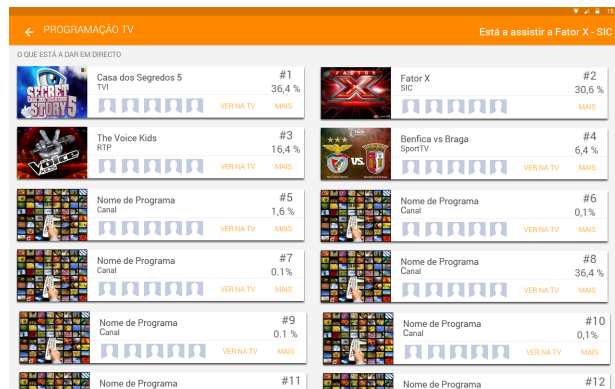


Figure 3.16: 2nd Prototype - Social TV Guide

Lastly, Figure 3.17 presents information about what friends are watching, through cards.

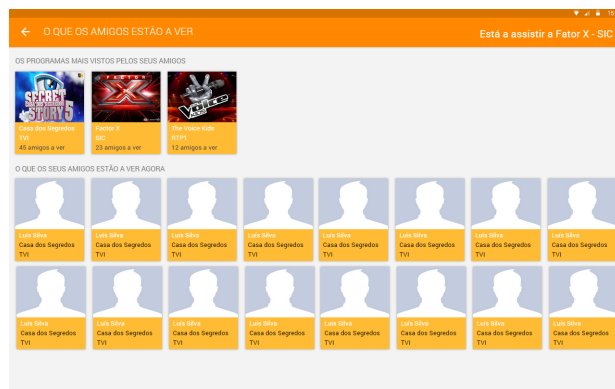


Figure 3.17: 2nd Prototype - What Friends Are Watching

On Section Implementation 3.3.3, details about these screens are presented.

3.2.5 Preliminary Experts Test

In order to assure a reasonable engaging and proficient design for the application, I have requested an expert to evaluate my design mockup facing Nielsen's usability heuristics ². This expert is Tiago Guerreiro, an Assistant Professor at the Informatics Department (DI) of the Faculty of Sciences, University of Lisboa (FCUL) and a researcher at the Human-Computer Interaction and Multimedia (HCIM) Research Team at LaSIGE, a research unit of the same department. The mockup was published on a public URL which was delivered to the expert with a text giving the mockup context and tasks listed on table 3.2.5.

Task	Description
1	Identify the popularity of the show on TV.
2	Identify the most popular show and start to watching it.
3	Identify the popularity of 5th most popular show.
4	Identify how many friends are watching the same show.
5	Identify what a friend (Luís Silva) is watching now.
6	Identify the last recommendation made or received by your friend Margarida Rodrigues.
7	Switch to the channel of the last show recommend by one of the friends your are following.
8	Recommend the show you are watching to a friend you are following.
9	Recommend the show you are watching to a group of friends.
10	Identify the meaning of the icon related to the friend José Silva on the screen showing the list of friends available to recommend.
11	Identify the friend who had followed more of your recommendations.
12	Identify the best recommendation made.
13	Identify if the last recommendation made was followed.
14	Identify what is your position in ranking of users who recommend TV shows.
15	Identify what is your position in ranking of users who recommend TV shows.
16	Send a comment on Twitter about what you are watching.

Table 3.3: Tasks Performed On Experts Test

Results

Table 3.4 show the problems found by the expert his test, matching with the missed heuristic.

²<http://www.nngroup.com/articles/ten-usability-heuristics/>.

Error	Heuristic
Popularity indicator is not clear.	H6. Recognition better than recall H2. Match between system and real world
Need to scroll in order to see all content in the first screen.	H8. Aesthetic and minimalist design
A click on a show's card returns always the same show details (Factor X - SIC). It should navigate to the details of the clicked show.	H3. User control and freedom H4. Consistency and standards H7. Flexibility and efficiency of use
In the menu, the "Highlights" has the home icon. This icon should be consistent with the label.	H4. Consistency and standards
It is not possible to click on the current tuned show indicator "A assistir a Factor X - SIC" on the friend's screen. It should be possible to click the label and start watching the same show instead of having a separated button or, at least, have both.	H7. Flexibility and efficiency of use
The menu becomes unaccessible when on the details screens (e.g. Friends, Program).	H3. User control and freedom
Absence of help and hints of how to use the application	H10. Help and Documentation

Table 3.4: Results of Experts Evaluation

Although not having changed this second prototype correspondingly to the experts results, this results were critical and their impact is verified on the implementation phase description. The reason why I did not modified the prototype was the fact that in this phase no more evaluations by experts or users were expected. Hence, that changes were applied directly on the development of Social Zapping.

3.3 Implementation

Towards the implementation of this project, these were the technologies used and built from scratch.

3.3.1 Back-End Database

In this phase, it was fundamental to analyse the requirements and to decide what storage solution to use. I have decided for a graph database. Neo4j was the one I picked because it was the most popular³ and simple to start working. The decision for a graph database is obviously based on the relationship-centered characteristics of a social graph. Regarding the requirement of having a social graph to store, this option was natural. Besides that, Cypher Query Language is very powerful where it is possible to write complex queries like "Who are the friends of my friends which are not my friend already?" in a simple way. An extra advantage is the fact of Neo4J having a useful REST API which vanishes the need of building Web Services for client access. Another interesting application of this type of database are recommendations. In this project, this was not an important aspect in the decision of the database because this work does not implement any recommendation algorithm. Recommendations in such algorithms are based on the relationships between people and things. With a graph database, that was a very simple to integrate in the existing model and can be explored in future work.

Graph Database Model

The following sections describe the Graph Database Model:

Node labels

FacebookUser

This label is applied when a node representing a user who logged in the application via his Facebook account is created. It has the following properties:

- facebook_id - The Facebook Id for the user's account.
- username - The first and last name of the user on Facebook.
- birthday - The user's birthday, if set on Facebook.
- location - The user's location if set on Facebook.
- link - The URL to user's Facebook page.

³<http://db-engines.com/en/ranking/graph+dbms>

- `photoUrl` - The photo used by the user in his Facebook profile. This image is applied in the application menu header.
- `coverPhotoUrl` - The cover image present in Facebook if set. This image is applied in the application menu header.
- `watching` - The channel position of the TV show being watched by this user.
- `recommendations` - The total number of recommendations made by the user.
- `recommendations_followed` - The total number of recommendations followed by the user.

FacebookInterest

This is the representation of "Like" in a Facebook page related to TV content. As so, the user "likes" are filtered by their category. It has following properties:

- `facebook_id` - The Id of the Facebook page.
- `category` - The category of the Facebook page. Regarding the purpose of this application, the possible values are: Tv, Tv show, Movie, Actor/director or Public figure.
- `interest_name` - The name of the page the user likes.
- `created_time` - The time when the user has "Liked" the page.

Program

The Program label represents a TV show present in the EPG. It has following properties:

- `eventId` - The TV show id.
- `title` - The name of the TV show.
- `description` - A text describing the context of the show.
- `duration` - The duration of the TV show in minutes.
- `startDate` - The time predicted in the EPG for the show to start.
- `endDate` - The time predicted in the EPG for the show to end.
- `imageUrl` - An image of the TV show.
- `channelPosition` - The position of the TV Show in the EPG.

- channelName - The name of the channel where the TV show is exhibited.
- channelImageUrl - The channel logo.
- recommendations - The total number of recommendations made for this TV show
- recommendations_followed - The total number of recommendations of this show that were followed.

Relationship Types

FRIEND

As expected, it represents the friendship between two Facebook Users, registered in Facebook. It has no properties.



Figure 3.18: Neo4j Graph View - My Social Graph on the application

LIKES

This relationship represents a Facebook user's interest in a TV Content. It has no properties.



Figure 3.19: Neo4j Graph View - My Likes on Facebook pages related to TV content

RECOMMENDS

It is the main action of the app, i.e., the recommendation of a TV show to a friend. Like that, it represents a recommendation made by FacebookUser A to his friend, FacebookUser B. It has the following properties:

- eventId - The TV Show Id
- title - The TV Show title
- imageUrl - The TV Show image
- channelName - The name of the channel where the show was transmitted
- date - The time when the recommendation was made
- followDate - The time when the recommendation was followed. (This property is only present when it the recommendation is effectively followed)
- msg - The text of the recommendation

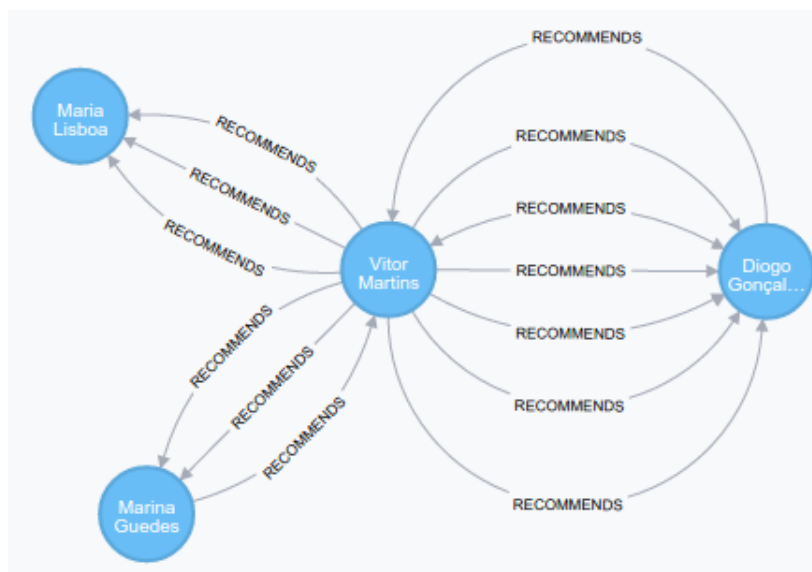


Figure 3.20: Neo4j Graph View - The recommendations I have made and received from friends

3.3.2 Back-End Services

In the analysis phase, I have decided to use RESTful Web Services to access the content in the database. To perform this, I used Neo4j REST API. With this, I have reduced the effort of creating RESTful Web Services for each content I would retrieve from the database. As so, the client (mobile device) could request data from database directly from these services. To ensure the access to the database, the REST API supports authorization and authentication via HTTP Basic Auth. To do this, the following HTTP headers are sent in every HTTP request to this API:

- Accept: application/json; charset=UTF-8
- Authorization: Basic <payload>

Here, <payload> is a base64 encoded string of "username:password".

This API allows querying with Cypher. Besides that, it supports using parameters submitted as JSON. This is an example of how to request database content through the Neo4j REST API:

- POST http://localhost:7474/db/data/cypher
- Accept: application/json; charset=UTF-8
- Content-Type: application/json

```
{
  "query" : "MATCH (x {name: {startName}})-[r]-(friend)
  WHERE friend.name = {name} RETURN TYPE(r)",
```

```
"params" :{
  "startName" : "I",
  "name" : "you"
}
}
```

And the corresponding response:

- 200: OK
 - Content-Type: application/json; charset=UTF-8
-

```
{
  "columns" : [ "TYPE(r)" ],
  "data" : [ [ "know" ] ]
}
```

Regarding the format of the response, "The results are returned as a list of string headers (columns), and a data part, consisting of a list of all rows, every row consisting of a list of REST representations of the field value — Node, Relationship, Path or any simple value like String."⁴

3.3.3 Mobile Application

The mobile application was developed for Android devices, specially Tablets due to the reasons found on the related work. In place to develop the application, Android Studio, the official Android IDE, was used. To keep a backup copy and to have a version control system, I used the Github plug-in. Another fundamental tool was Genymotion, a Android emulator much more fast that the native emulator.

Here, I have developed 11 packages grouping a total of 27 Java classes. Through the development of this application I have also integrated a bunch of external libraries in order to fulfill the Material Design specification and to implement an usable and appealing design. The development was guided by the design mockup previously created, starting from the login screen and following the navigation flow diagram (3.2.3). In the following paragraphs it is revealed the important aspects in the implementation of each screen. Moreover, here is possible to understand the real aspect of the product and find where the functional requirements are implemented.

Login Screen

Here is where the user starts to use Social Zapping. He is presented with a Facebook Login Button and his Facebook's profile picture when is already logged in. After inserting

⁴<http://neo4j.com/docs/stable/rest-api-cypher.html>.

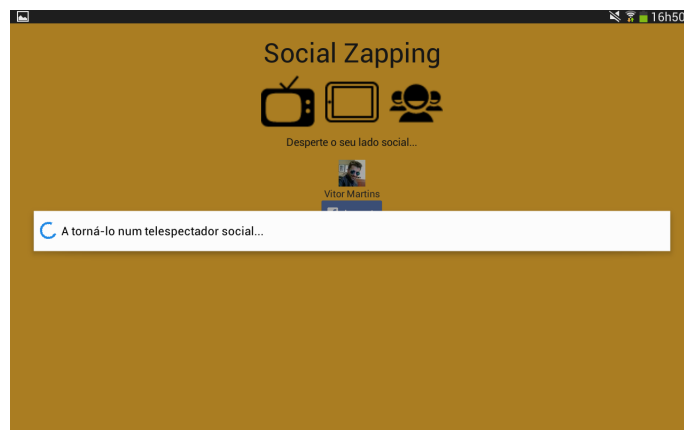


Figure 3.21: Social Zapping - Login Screen

the credential of his Facebook account, the application stores user's public information, interests in movies, television, likes and friends who use this application. Here, likes are filtered by the categories TV show, Actor/Director and Public Figure. All this is done by the integration of the Facebook SDK for Android and its Graph API. As so, after the login, the database is populated/updated with a user, Facebook interests and Facebook friends nodes all connected with relationships like LIKES and FRIEND. Figure 3.21 shows the progress dialog during the login with the message "Turning you into a social TV viewer...".

External Libraries Used (2.5.2)

- Facebook SDK for Android: Login Button and Graph API

Cypher Queries Applied

- Create/Update FacebookUser node (A.1)
- Create/Update FacebookInterest (A.2)
- Create/Update FRIEND relationship (A.4)
- Create/Update LIKES relationship (A.3)

Highlights Screen

After successfully logged in, the user is presented with the default and main screen of the application. This is a dashboard presenting a navigation bar with an "hamburger" button to access the side menu and the information about the current TV show tuned in the STB. The first highlight is a card showing the resumed information about the current show being watched: title, channel, image and who are the friends watching the same. In this

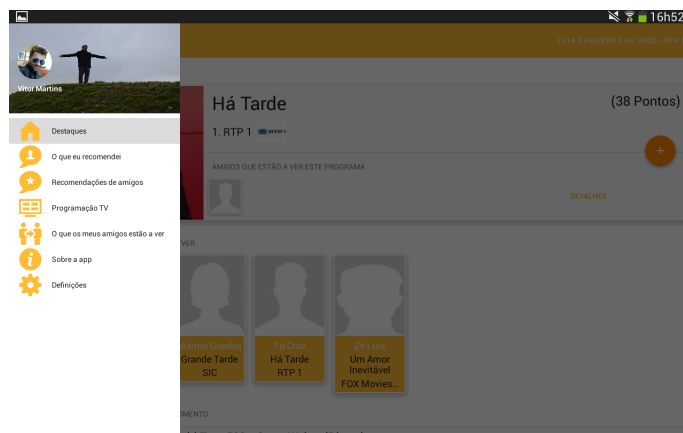


Figure 3.22: Social Zapping - Side Menu

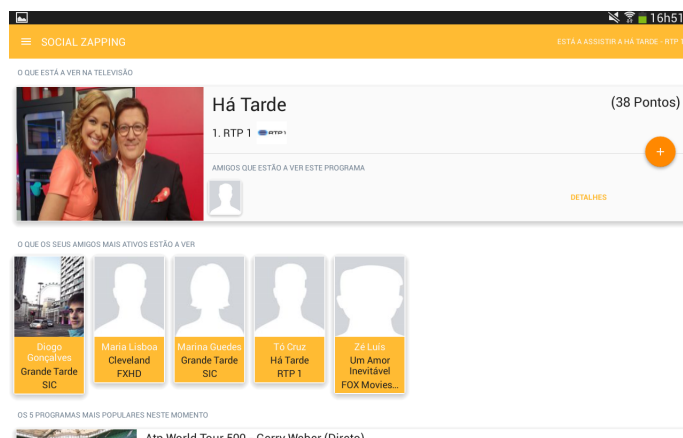


Figure 3.23: Social Zapping - Highlights Screen (Top)

card is presented the main action of Social Zapping, the recommendation of the current show, in the form of a floating action button (round orange button with a white plus sign). This button gives access to recommend any friend and offers a shortcut for the 3 most recommended friends. Right below, is showed a highlight of what friends are watching in the moment. In the bottom, is previewed the top 5 live shows, considering the application's popularity mechanism. Figure 3.22 shows the options in the menu: "Highlights", "What I recommended", "Friends' recommendations", "TV Guide", "What my friends are watching", "About" and "Settings". Figures 3.23 and 3.24 show the contents of this screen with the following 3 sections (grey colored headers): "What you are watching on TV", "What my most active friends are watching" and "The top 5 most popular shows in this moment". Figure 3.25 shows the state of the Floating Action Button when pressed, expanding to new buttons: "Recommend to another friend" and 3 shortcut buttons to recommend one of the top 3 most recommended friends.

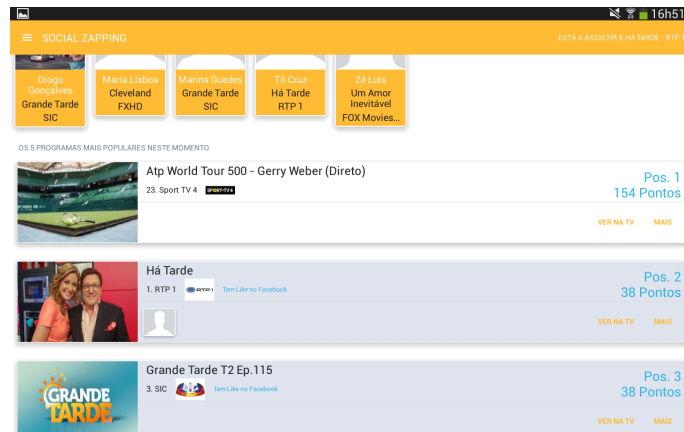


Figure 3.24: Social Zapping - Highlights Screen (Bottom)

External Libraries Used (2.5.2)

- CardView
- Floating Action Button
- Material-Design
- Material-Design-Drawer
- Picasso
- Android Asynchronous Http Client

Cypher Queries Applied

- Match the show being watched (A.23)
- Update the show being watched (A.24)
- Match the top 3 most recommended friends (A.7)
- Match what friends are watching (A.6 A.23)
- Match the top 5 live shows (A.8)
- Check match between TV Show and Facebook's interests (A.10)

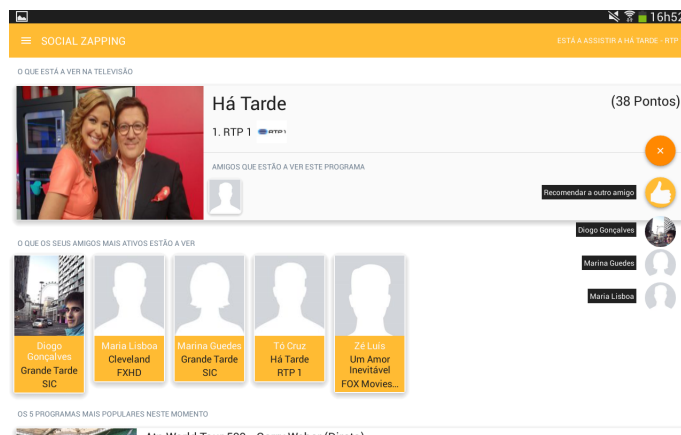


Figure 3.25: Social Zapping - Recommend Button

Choice Of A Friend To Recommend Screen

This screen displays lists of friends available to recommend the current show. Besides to complete list of friends in the bottom, it is displayed other lists filtering the complete list, in order to help the selection of the most relevant friends to recommend. As so, list with the most recently recommended friends, favorite(most recommended) friends and friends with common interests are presented. The latter is based on the match between the user's stored Facebook interests on TV content and the registers for his friends. In each row there is the recommend button, which provides the effective recommendation. Figure 3.26 shows the screen where is possible to select a friends among the following groups: "Favorites", "Recently recommended", "Friends with common interests" and "All". Figure 3.27 shows the dialog where the user writes the recommendation text comment.

External Libraries Used (2.5.2)

- Material Design
- Picasso
- Round Profile Picture
- Header List View
- Android Asynchronous Http Client

Cypher Queries Applied

- Match most recommended friends (A.7)
- Match most recently recommended friends (A.11)

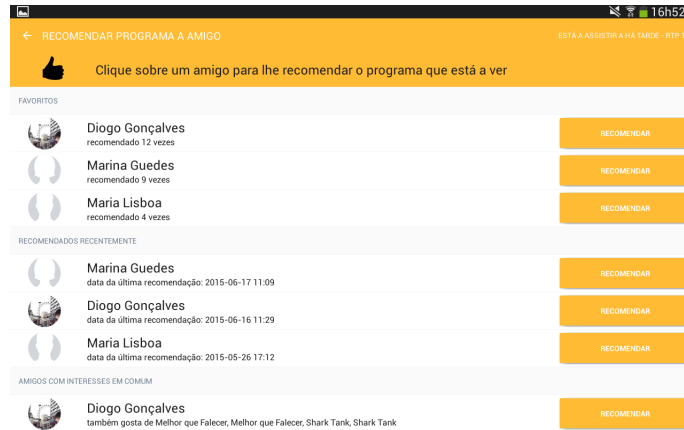


Figure 3.26: Friends Available For Recommendation

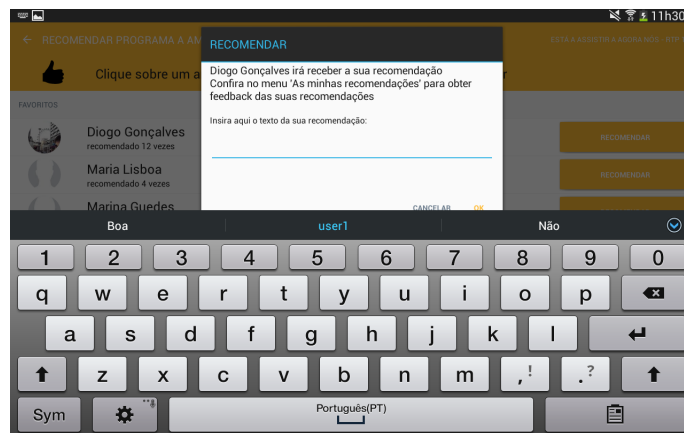


Figure 3.27: Social Zapping - Insert Text On Recommendation

- Match friends with common interests (A.12)
- Match all friends (A.6)
- Create RECOMMENDS relationship (A.5)

TV Show Detail Screen

Here, details about a TV show are presented, like the image, title, channel, description, start and end dates. This information is often found on the EPG and, as so, the interesting point here is the presentation of the show's ranking, find who are the friends who are watching the show and who recommended it. Figure 3.28 show the screen where the user find TV show details. On the left side is possible to check start and end dates as the description, in the right side the user checks the recommendations' statistics and the friends who are watching the and who recommended it.

External Libraries Used (2.5.2)

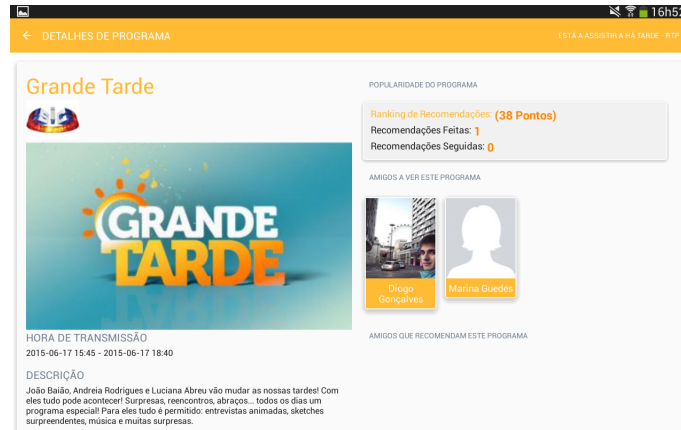


Figure 3.28: Social Zapping - TV Show Details

- Card View
- Picasso
- Material Design
- Android Asynchronous Http Client

Cypher Queries Applied

- Match TV show (A.9)
- Match friends watching a show (A.6)
- Match friends who recomend a show (A.22)
- Match TV show ranking (A.20)

Friend Detail Screen

This is not just a user profile screen. Besides that the friend's name and photo, information about the TV content being watched by him is showed. Moreover, all the recommendation interactions with that friend is presented in the form of a list, keeping track of the recommendations historic with that friend. In the Figure 3.29 is possible to check in the bottom half of the screen the recommendation I made to my friend and in the Figure 3.30 is showed the recommendation received from a friend.

External Libraries Used (2.5.2)

- Card View

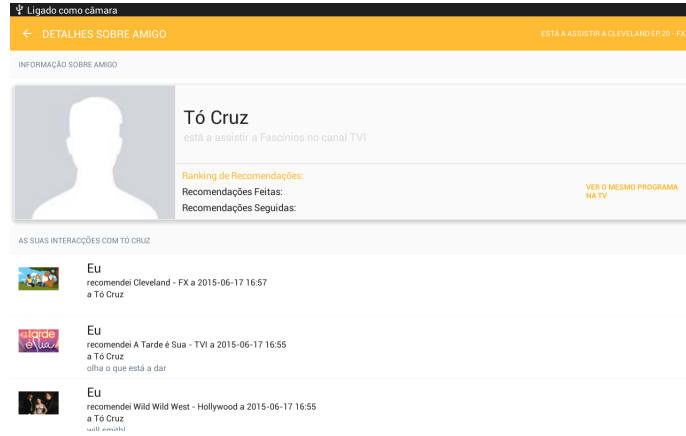


Figure 3.29: Social Zapping - Friend Detail

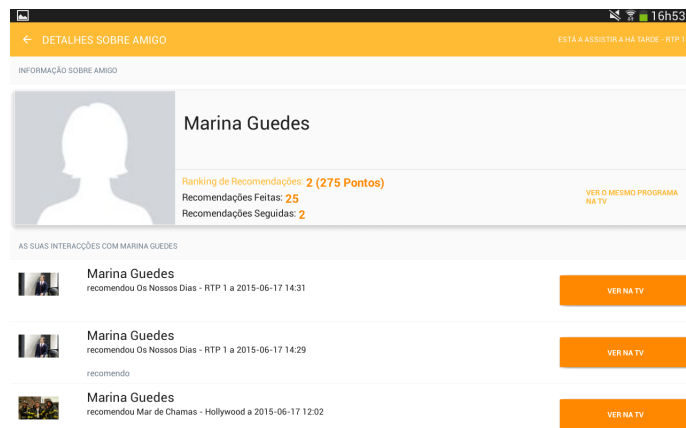


Figure 3.30: Social Zapping - Friend Detail (With Recommendations To Follow)

- Material Design
- Header List View
- Picasso
- Round Profile Picture
- Android Asynchronous Http Client

Cypher Queries Applied

- Match TV content that friend is watching (A.23)
- Match all recommendations sent and received to and from that friend (A.21)

Recommendations Made Screen

This is where the user gets his recommendations feedback. All the recommendations made by the user can be checked here, including if they were already followed. Besides that, the user is presented with his recommendations statistics, his top followers also as the recommendations most followed by his friends. On the left side of Figure 3.31 it is displayed the information about: "Your statistics", "Your best recommendation", "The friends who follow more of your recommendations". On the right side it is listed the historic of the recommendations made.

External Libraries Used (2.5.2)

- Card View
- Material Design
- Round Profile Picture
- Picasso
- Header List View
- Android Asynchronous Http Client
- Horizontal List View

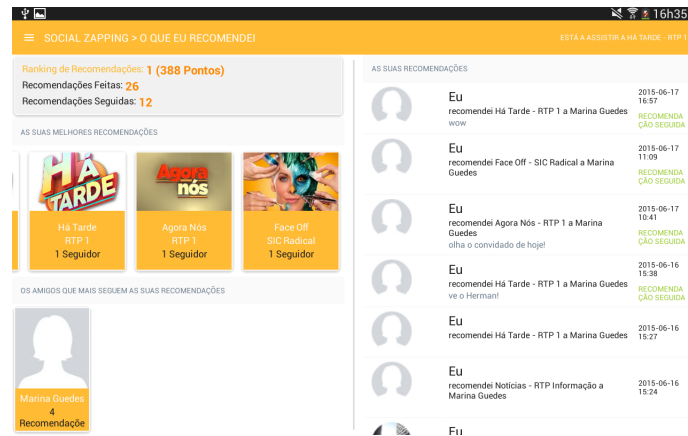


Figure 3.31: Social Zapping - Recommendations I Made

Cypher Queries Applied

- Match a user ranking (A.19)
- Match most followed recommendations (A.17)
- Match top followers (A.18)
- Match all the recommendations made by a user (A.16)

Recommendations Received Screen

As expected when we talk about social recommendations, there is a screen where we find a timeline with all the TV shows recommended by friends. Moreover, here we find highlights about the most recommended shows on the top of the screen. Figure 3.32 show cards with the most recommended shows at top and the historic of recommendations received at the bottom. The orange button is where the user follows the recommendations, playing it on TV.

External Libraries Used (2.5.2)

- Card View
- Material Design
- Round Profile Picture
- Picasso
- Header List View
- Horizontal List View
- Android Asynchronous Http Client

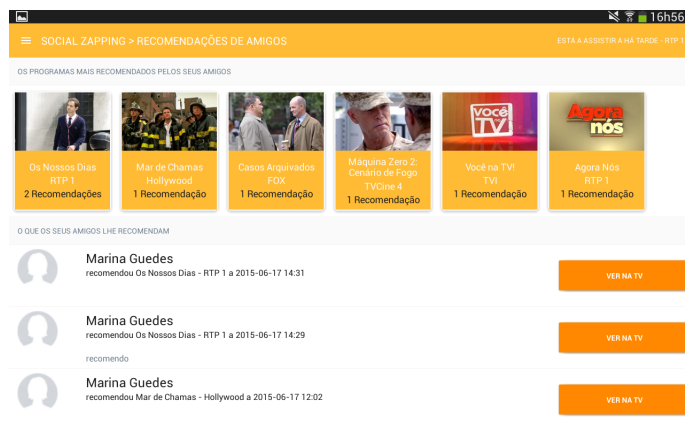


Figure 3.32: Social Zapping - Recommendations From Friends

Cypher Queries Applied

- Match most recommended shows to a user (A.13)
- Match all recommendations made to a user (A.14)
- Update RECOMMENDS relationship with the date when it was followed (A.15)

TV Guide Screen

Like on a traditional EPG, a list of TV shows is presented. Here, with the cards representing each show, the list is ordered by popularity, presenting the ranking position and its point as the traditional basic information about the show. The EPG information was stored on database running a Javascript on NodeJS, on a daily basis, that collected public information about the EPG. Extra information about the friends who are watching each show is also highlighted in each card. Another highlight provided is when there is a match between a user's Facebook interest and the title of a show. In that case, the card turns blue and a message is displayed to enhance the interest on the show. In the Figure 3.33 is showed the TV Guide screen with some blue cards representing likes on TV shows at Facebook.

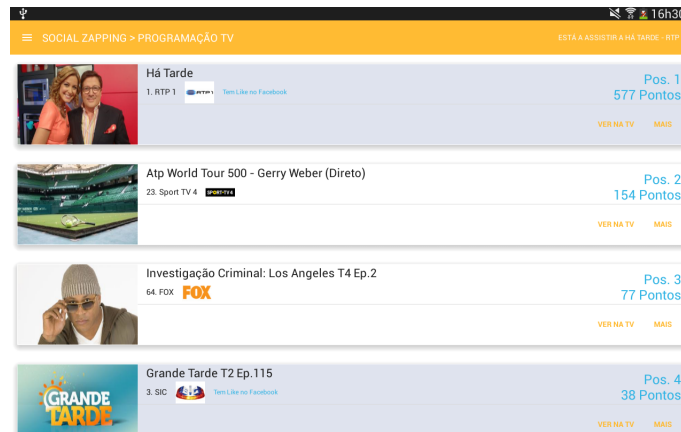


Figure 3.33: Social Zapping - TV Guide (EPG Ordered By Social Buzz)

External Libraries Used (2.5.2)

- Card View
- Material Design
- Super List View
- Picasso
- Android Asynchronous Http Client

Cypher Queries Applied

- Match EPG (A.8)
- Match TV show ranking (A.20)
- Match friends watching a show (A.6 A.23)
- Check match between TV Show and Facebook's interests (A.10)

"What Friends Are Watching" Screen

The screen that everybody likes but that has an ethical struggle about TV viewers privacy. Here, the user can check what all his friends are watching at the moment. In the Figure 3.34, the shows with more friends watching are observable at the top and at the bottom is showed a list of friends and what they are watching.

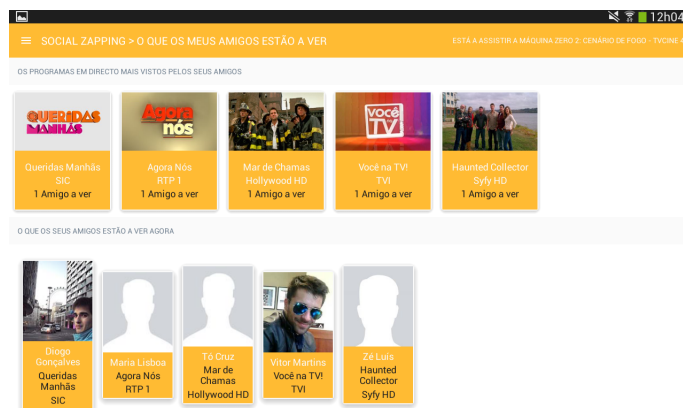


Figure 3.34: Social Zapping - What My Friends Are Watching

External Libraries Used (2.5.2)

- Card View
- Material Design
- Header List View
- Picasso
- Android Asynchronous Http Client

Cypher Queries Applied

- Match what friends are watching (A.6 A.23)
- Match TV show (A.9)

Database Access

As any application with some amount of data, it was required to have a way for the client, in this case, the mobile application, to query database. This was done with a very efficient union of available technology. Considering the use of Neo4J Database, I have leveraged their REST API and an external library ²⁸. The request was explained in 3.3.2. On Appendix C there is an example of how this request is done in Android.

With this API I had the build of web services for free and as so, there was no need to develop a server side component, connected with the database, to receive client requests and retrieve query results. This decision was made taken in count the experimental condition of this application. Despite the fact of the neo4j REST API having server authentication and authorization, all the queries remained on the source code.

3.3.4 Ranking Points

One of the functional requirements of the application was to maintain a ranking of TV shows in place to order them by recommendations points and, as so, show popularity trends in the live shows universe. Besides TV shows, users also collect points and can compete between them.

The recommendation points depend on two actions: recommend a show to a friend and follow a recommendation made from a friend. Here, we have to remember that is only possible to recommend the show tuned in the STB and auto-recognized by the application and that the following action is only considered when a friend effectively start to watch the TV content linked to the recommendation.

Hence, I analysed what formula to apply to calculate the ranking of shows and users.

Precision and Recall Mapping

First, I considered the Precision and Recall formulas. These formulas are usually applied on measuring information retrieval effectiveness. Precision is the fraction of retrieved documents that are relevant and recall is the fraction of relevant documents that are retrieved. If we join them, we have F-Measure. This is the weighted harmonic mean of precision and recall, also known as the F1 measure when precision and recall have the same weight.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Figure 3.35: Precision formula

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Figure 3.36: Recall formula

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$$

Figure 3.37: F1 Measure formula

Here, the first approach was to map these concepts to the social recommendation context. Thus, documents were mapped to recommendations, retrieved documents to number of recommendations and relevant documents to recommendations that were followed.

This mapping resulted in the following formulas:

$$Precision_s = \frac{RF_s}{R_s} \quad Precision_u = \frac{RF_u}{R_u}$$

Figure 3.38: Mapping of Precision formula for TV show's recommendations

$$Recall_s = \frac{RF_s}{RF_t} \quad Recall_u = \frac{RF_u}{RF_t}$$

Figure 3.39: Mapping of Recall formula for TV show's recommendations

$$Ranking_s = \frac{2.Precision_s.Recall_s}{Precision_s + Recall_s} \quad Ranking_u = \frac{2.Precision_u.Recall_u}{Precision_u + Recall_u}$$

Figure 3.40: Mapping of F-Measure formula for TV show and user ranking

R_t - Total of recommendations

RF_t - Total of recommendations that were followed

R_s - Total of recommendations for a show

RF_s - Total of recommendations for a show that were followed

R_u - Total of recommendations made by a user

RF_u - Total of recommendations made by a user that were followed

After this mapping, these formulas were inserted in the Android application and tested, making some recommendations for distinct shows with distinct users. When testing, the first action was to recommend a show to a set of friends and then check for changes on the show's ranking. As it was clear, nothing was changed and the show continued to have 0 points in the ranking, even with some recommendations related with it. After this small test, it was not satisfying that a show or a user started to have points only after getting a recommendation followed, due to the mapping of the intersection of relevant and retrieved documents to the concept of recommendations followed.

Hybrid Formula

Here it was decided to apply a hybrid formula to give points for shows and users. Inspired in the previous attempt, it is a weighted harmonic mean of the contribution for the total number of followed recommendations (left term of the equation) with the contribution for the total number of recommendations (right term of equation). k is a double between 0.0 and 1.0 which represents the weight percentage given to the followed recommendations contribution. This way, it is possible to evaluate if the application should reward more

who recommends or who follow recommendations. For this proof of concept, k was set to 0.5, meaning that the two terms have the same weight.

For presentation reasons, I added p factor to multiply the decimal value for points and transform it to a simple big number of points. Hence, p was set to 1000.

$$Ranking_s = p \left(k \cdot \frac{RF_s}{RF_t} + (1 - k) \cdot \frac{R_s}{R_t} \right) \quad Ranking_u = p \left(k \cdot \frac{RF_u}{RF_t} + (1 - k) \cdot \frac{R_u}{R_t} \right)$$

Figure 3.41: Hybrid formula for TV shows Ranking

Summary

The implementation of Social Zapping turned to be simple by leveraging existing technologies like Neo4J and its REST API and Android external libraries and projects.

Table 3.5 shows that every functional requirement defined on Section 3.1.2 was satisfied. Here it is possible to observe that the Highlights Screen satisfies many of the functional requirements.

Screen	Functional Requirements Satisfied
Login	FR1; FR2; FR9
Highlights	FR3; FR4; FR5; FR9; FR10; FR12; FR13; FR14
Choice Of A Friend To Recommend	FR5; FR9; FR11
TV Show Detail	FR9; FR12; FR13; FR14; FR15
Friend Detail	FR9; F12; F14; FR16
Recommendations Made	FR7; FR9; FR14
Recommendations Received	FR6; FR9
TV Guide	FR9; F12; FR13; FR14
What Friends Are Watching	FR9; F12

Table 3.5: Functional requirements satisfied in each screen

Chapter 4

Evaluation

After developing the proof of concept application, it was time to execute an usability test. There are various types of usability testing, according to the development phase. In a phase like this, an Exploratory Test suited the needs. The goal was to explore user's mental model for the tasks that the application is trying to ease. This way, exploratory test evaluates whether the user can distinguish functional elements of an interface and if he values the features presented. It also tries to verify if the intuitiveness of the design. This kind of test is helpful when the designers are still finding what features to include in the final product, but for this work it will help to identify the problems of this proposal. As Nielsen stated, "Despite many weaknesses, interviews are a valuable method for exploratory user research"¹. As so, some research about how to plan, execute and report an exploratory usability interview had to be made. Here, it was important to understand what were the up and down sides of interview about usability. The first consideration is to avoid asking participants to remember details of past experiences or speculate on future use of the system. Regarding the latter, in cases of doubt, I've decided to prototype different alternatives because users cannot predict a solution they have not seen. Interviews do not answer to specific design questions like the color of a button, position of an element or the number of navigation levels. These dilemmas cannot be directly asked to participants. They are resolved by observation of user's interaction with the application and by spontaneous comments. The person who dialogues with the users have to be aware about the query effect. The number of queries has to be minimum considering that people make up opinion about anything if they are asked. Then, their opinions will not reflect their real preferences. Getting to the good side, interviews are a good approach to explore users' general attitudes and how they think about a problem. After that, it is time to design features to address that problem. Another good approach is to use the critical incident method. This resides on the idea of asking the user to remember particularly difficult cases or, in the other hand, cases where they were specially surprised and something worked particularly well. Another benefit of an one-to-one interview is that what is being

¹<http://www.nngroup.com/articles/interviewing-users>

discussed can be addressed to participant's individual concerns and misunderstandings can be cleared up right away.

4.1 Trial Setup

In order to prepare this interview, research on guidelines was done^{2 3}. On Appendix B, I summarize what I found to be the most relevant tips.

The trial environment setup was simple. I have replicated the living room environment as much as possible at the office, sitting the participants on a comfortable chair in front of 42 inches full HD LCD TV and giving them a tablet with 10.1 inches.

Before giving the tablet to the participant, I logged in the application with the Facebook account of a test user I have created for this purpose. With another tablet I logged in the application with my personal Facebook account and made some recommendations to that test user, who I was related as a friend. After that, I ensured that the TV was tuned in the channel position registered in the database for the test user. During the trial I used an auxiliary application to know when the user switched channel. This auxiliary application is explained at the Section above. When that happened, I inconspicuously used the remote to switch the channel to simulate the channel switching through the application.

4.2 Data Collection

The trial was conducted only by one person, myself. During the trial, I have observed the participant's reactions and listened to their comments and suggestions. The registration of such data was on paper, taking quick notes while the participants toss ideas around.

The main registration tool was an online survey using Google Forms, saving the responses in a calculus sheet. During the trial, I have registered the time required to complete all the tasks. Here, I asked for the participants to complete a total of 22 tasks using Social Zapping and evaluate how hard was to perform each one of them in a Likert scale where 1 was "Impossible" and 10 was "Perfectly Possible". Moreover, I asked about the utility and interest on using the application if they could use it as a product at their homes, using the same scale. The tasks were distributed with a random order, in place to avoid the influence of their sequence because some of them are relative to the same screen. To enforce this, I asked for the participants to go to the "About" screen after completing each task in place of losing previous context and isolate the usability of each task.

²<http://www.webleadable.com/blog-reports/web-usability/usability-testing.shtml>

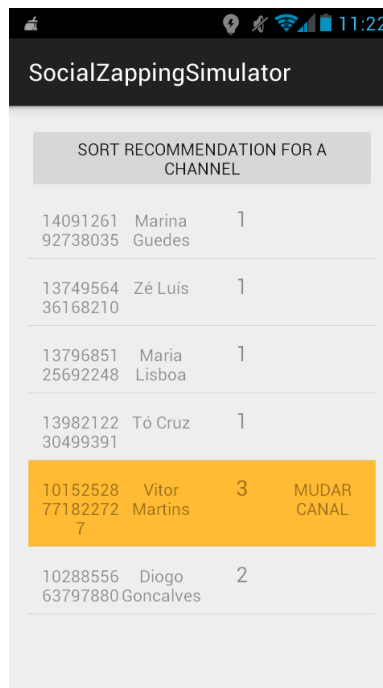
³<http://www.gv.com/lib/get-better-data-from-user-studies-16interviewing-tips>

4.2.1 Auxiliary Application

In this type of evaluation it is important to replicate the target environment as much as possible. With that thought in mind, I have developed an auxiliary Android application, Social Zapping Simulator. This application aimed to avoid disturbing the participant's experience with constant peeking of his usage and specially avoid asking when there was an attempt to switch the channel through the application. This is important to give a private and comfortable environment to the user, without intimidating him.

Thus, Social Zapping Simulator is an evaluation tool I have developed for this evaluation purpose and used only by the person who is guided the trial, myself.

This application access the database as Social Zapping does and provides a few simple features: show the current tuned channels of each user, highlight a user's row when he switches the channel via Social Zapping and sort a live show to have an increment of one recommendation, simulating a real time popularity trend change on the social TV guide.



The screenshot shows the SocialZappingSimulator app interface. At the top, there is a header bar with the title "SocialZappingSimulator". Below the header, there is a button labeled "SORT RECOMMENDATION FOR A CHANNEL". The main content is a list of users with their IDs, names, and recommendation counts. The user "Vitor Martins" is highlighted in orange, indicating a channel switch. The recommendation count for Vitor Martins is 3, and there is a "MUDAR CANAL" button next to it. The other users and their recommendation counts are: Marina Guedes (1), Zé Luis (1), Maria Lisboa (1), Tó Cruz (1), and Diogo Goncalves (2).

ID	Name	Recommendation Count	Action
14091261 92738035	Marina Guedes	1	
13749564 36168210	Zé Luis	1	
13796851 25692248	Maria Lisboa	1	
13982122 30499391	Tó Cruz	1	
10152528 77182272 7	Vitor Martins	3	MUDAR CANAL
10288556 63797880	Diogo Goncalves	2	

Figure 4.1: Social Zapping Simulator - Channel switch during trial

4.3 Participants' Characterization

The user study was mostly conducted at the Caixa Mágica Software headquarters, having been performed also at my mentor's office in my University, with a total of 11 different participants. Most of them were males, having 8 males and 3 females. The age of the participants tended towards the age intervals between 25 and 34 and between 35 and 44 years old, both with 4 participants. The youngest user was 12 years old and the oldest

had 42 years old. About their education, the lowest education level was the 7th grade, the highest was Advanced Graduate work or Ph.D and most of them had a Bachelor's or Master's degree. All of the participants watch some TV at home, where 6 of them consume TV only 1 hour per day, 3 consume between 2 and 5 hours and 2 of them consume between 2 and 5 hours.

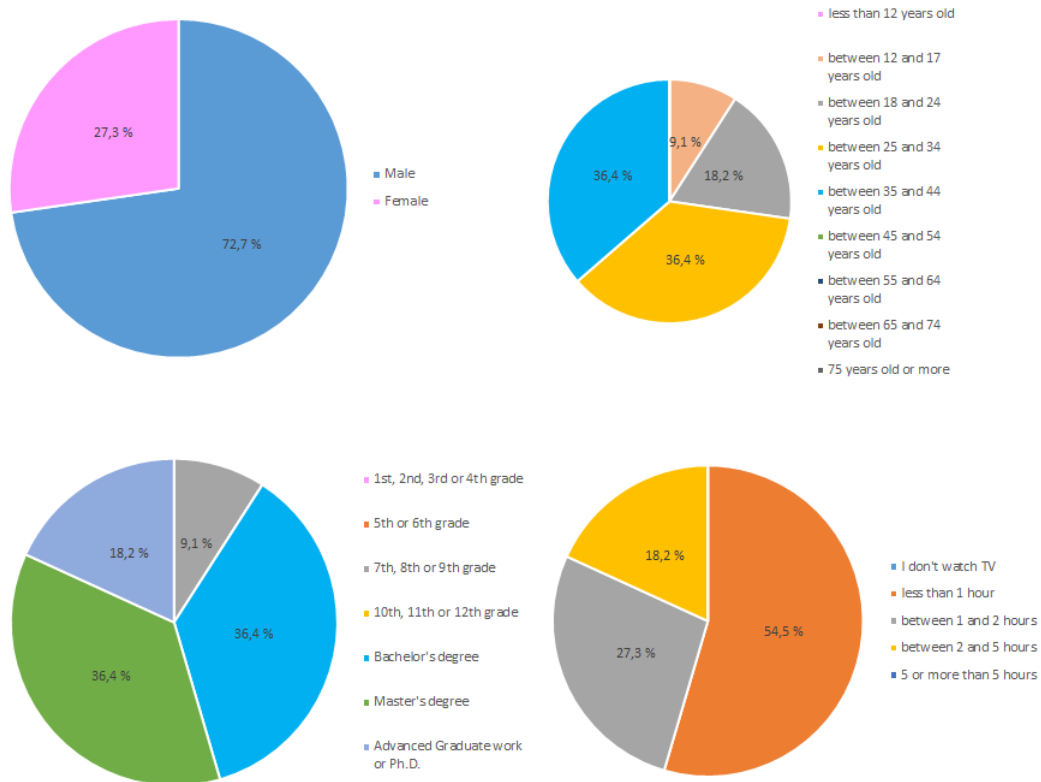


Figure 4.2: Participants' Characterization - Gender and Age(top), Education and TV Consumption(bottom)

4.4 Results

4.4.1 Usability

Part of the trial consisted in accomplishing tasks in the application and evaluating how hard was to do them. This has taken 36 minutes on average, considering pauses for comments and suggestions.

In the Figure 4.3 it is possible to observe how hard were the tasks to perform. The box-and-whisker plots are ordered by average. The hardest task to accomplish was number 17, which goal was to recommend a show that the participant was not watching. Task number 1, which was also related to the recommendation action and was the second more difficult task to perform. In almost all tasks there was at least one participant considering the task perfectly possible, but there were 2 tasks considered impossible to realize. The average of all tasks easiness to perform was 8.6.

Task	Description
1	Recommend the current show being watched to all of your friends
2	Identify the show tuned on the TV and its popularity
3	Identify the friends watching the same show as you
4	Identify other shows being watched by your friends
5	Start watching the most popular TV show
6	Get more details about the current show
7	Find if any friend recommended the live show now on channel 1
8	Find if any friend is watching the live show now on channel 1
9	Find the description of the show you are watching
10	Find your position and points in the recommendations ranking
11	Find if the last recommendation made was followed
12	Identify who is the friend who followed more of your recommendations
13	Identify what was the recommendation more times followed by your friends
14	Identify what is the TV show more times recommended for you
15	Identify the last recommendation made for you
16	Identify the 6th TV show most popular in this moment
17	Recommend the live TV show now on FOX channel to a friend
18	Identify what your friend Vítor Martins is watching now
19	Find if you are ahead of a friend in the recommendations ranking
20	Find the oldest recommendation made to a friend
21	Identify how many recommendations has a TV show
22	Follow the last recommendation made for you watching it on your TV

Table 4.1: List of tasks to perform by the participants

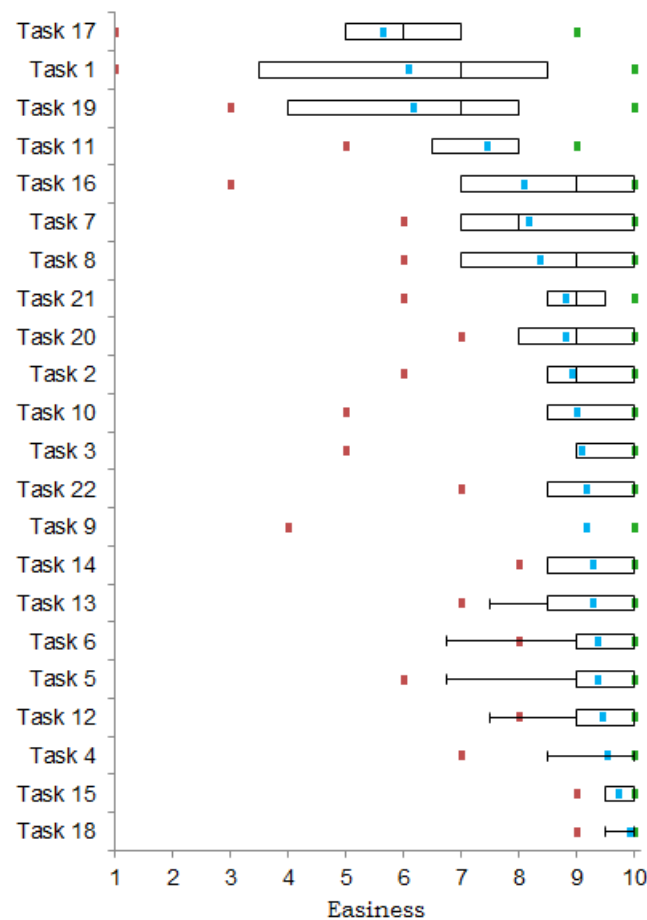


Figure 4.3: How hard was for you to accomplish this task?

4.4.2 Overall Opinion

After performing the tasks, I gave total freedom for the participants to explore and revisit the application if that helped them giving their opinion on the following 4 matters: Fun, Design, Navigation and Utility.

Fun

When asked about how fun they had experimenting Social Zapping, participants gave on average 8.2 on fun level, where 1 was completely boring and 10 represented exciting. Here, the minimum fun level that a participant committed was 4 and two participants gave the highest level, 10.

The following box-and-whisker plot summarizes the statistics of the participants' answers about fun.

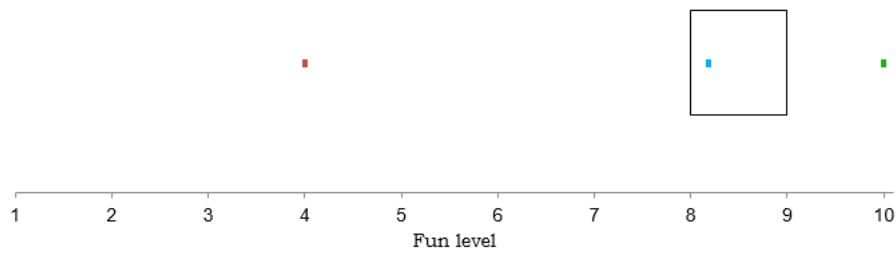


Figure 4.4: How much fun do you had using this application?

Giving that this sample of users was composed by people who spend little time watching TV, it was interesting to evaluate if this application was more fun for who consume more hours of TV per day. The following scatter graph crossed the level of fun given by the participants and the average of hours per day they consume TV. The graph shows little correlation about fun and TV consumption.

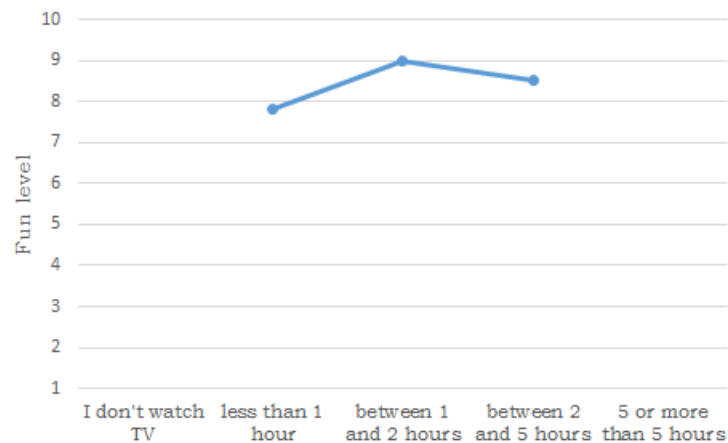


Figure 4.5: Fun vs TV Consumption

Design

One of the goals for this work was to propose an appealing design for the mobile application. Hence, it was asked how much did the participants liked the design, look and feel of Social Zapping.

The average was 8.1, some participants gave 9, the highest level received and there was who did not liked so much and gave 6, the lowest level.

The following box-and-whisker plot summarizes the statistics of the participants' answers about the design.

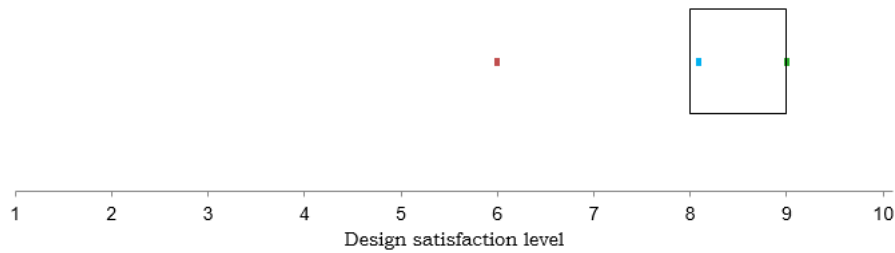


Figure 4.6: How much do you like the application's design?

Navigation

Although having some insights about usability with the tasks evaluation on Section 4.4.1, I found it was important to ask how hard was to navigate through the application after running a set of tasks involving all screens, as an overall opinion.

Here, where 1 was impossible to navigate and 10 was perfectly possible, the average answer was 5.7. The minimum score given was 3 and 9 was the highest.

The following box-and-whisker plot summarizes the statistics about the participants' answers on navigation matter.

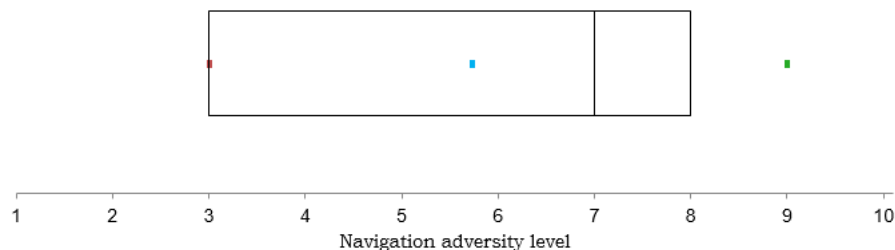


Figure 4.7: How hard was to navigate on the application?

Utility

One of the main goals about asking participant their overall opinion was also to get an insight about how useful Social Zapping could be to them, i.e., would they use it if it was a commercial product?

Here, 1 was absolutely not useful and 10 was very useful. In this prediction exercise, the average answer was 7.6. The lowest level given was 3 and some of them considered very useful, giving a score of 10.

The following box-and-whisker plot summarizes the statistics about the participants' answers about the utility.

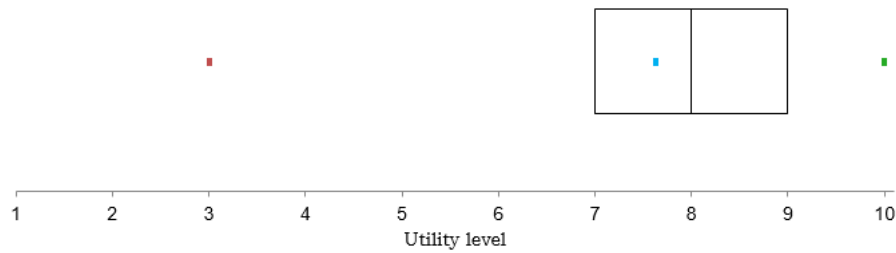


Figure 4.8: How useful would this application be to you if it was a commercial product?

As was done in Section 4.4.2, I related information about the number of hours the participants watched TV per day on average and their answers about the utility. In the Figure 4.9, it is possible to observe the level of utility increase as the TV consumption is higher.

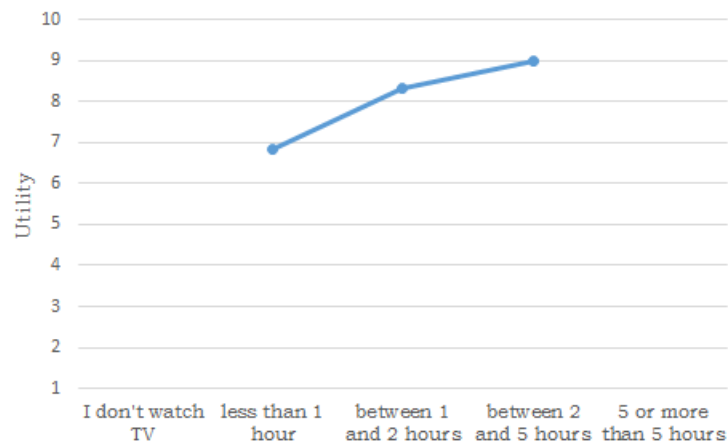


Figure 4.9: Utility vs TV Consumption

4.4.3 Engagement

Aiming to get an insight about the participant's engagement, I asked them to evaluate the interest on each of Social Zapping's features on a Likert scale, where 1 was not interesting at all and 10 was very interesting.

Figure 4.10 shows a box-and-whiskers plot for each task, ordered by average answer starting with the less interesting average.

Thus, " See what all my friends are watching on their TV" feature was the one with worst average(7.7) and " Click on a recommendation and automatically starts to play it on my TV" was the feature with best average(9.1).

Almost all features had answers bellow 6. The only features with all results above 6 were: " See if a recommendation I made was followed (if it was watched on friend's TV)", " Check details about the live TV show (image, description, start and end hour",

”Recommend a show to a friend”, ” See the TV show most recommended by my friend” and ” Click on a recommendation and automatically starts to play it on my TV”.

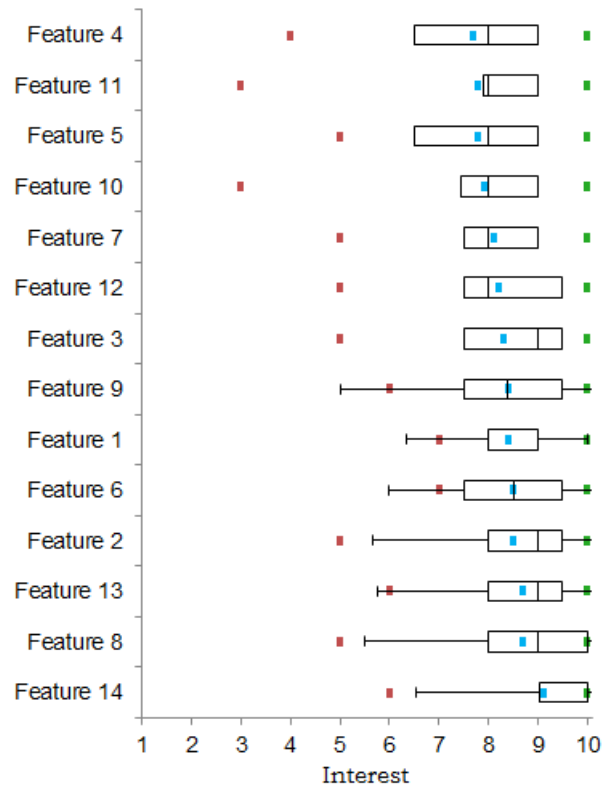


Figure 4.10: Evaluate how interesting is each feature for you?

Feature	Description
1	Check details about the live TV show (image, description, start and end hours)
2	See the friends who recommended a show
3	See the friends watching the same as me
4	See what all my friends are watching on their TVs
5	See the most popular live shows in real time, checking the popularity trends
6	Recommend a show to a friend
7	See what was my best recommendation (more times followed)
8	See who is the friend who followed more recommendations I made
9	See if a recommendation I made was followed (if it was watched on friend’s TV)
10	Users recommendations ranking
11	TV Shows recommendations ranking
12	See all recommendations interactions with a friend
13	See the TV show most recommended by my friends
14	Click on a recommendation and automatically starts to play it on my TV

Table 4.2: List of features to evaluate by the participants

4.4.4 Observations, Comments and Suggestions

Along the trial the think aloud method was extremely important to access users' navigation problems and their expectations on the application. With this method I registered their comments and suggestions. The following lines present their major navigation problems and their relevant comments and suggestions.

Regarding the recommendation task, participants had problems performing it. One participant started to look for a recommend option on the side menu and then looked on TV Guide option. Some of them noticed that in Highlights screen existed something about the live show. Here, some questioned "What does this big orange button with a plus sign does?", which means that the Floating Action Button grabbed the participants' attention, even if its purpose was not clear. One participant suggested switching the plus sign for a share or friend image and other who liked the button recommended to repeat the Floating Action Button on Show Details screen. The recommend button was expected on the Show Details screen. When the Floating Action was discovered, there was a clue about how to perform a recommendation but other struggles emerged. Participants expected a "Recommend to All Friends" button, right after pressing the floating action button to expand options. Reaching the screen where is possible to pick a friend to recommend, participants felt bored by having to recommend one friend at a time. The common action was to click on list headers like "Favorite Friends", "Most Recent Friends" or "All friends" in place to select a group of friends or all of them. In this screen, design was not satisfying. One participant said: "I do not like this repetition of big buttons in each row of the list, they are all the same and their presence is a little bit annoying". Other participant said: "It is boring to write a text comment for each friend when I want to recommend a group of friends". After finding how to recommend, some participants missed a confirmation dialog in place to have feedback about the success while others confirmed this in the Recommendations Sent screen. Even having previous instructions, most of the participants could not remember Social Zapping rules about recommendations. They forgot that it was required to watch a TV show in order to recommend it. As so, there were tries to recommend a TV show that was not being watched in that moment. For this action many screens were checked in order to find a clue about how to recommend, which clearly suggest improvements about the usability for this action. An interesting suggestion was to have constant access to a recommend button on the navigation bar, where the current show being watched is displayed. Other interesting comment about how to recommend was about its rule: "for recommendation about live shows is important that I have to be watching it but when I already know the TV show, which happens with movies or series, that could be recognized by Social Zapping by checking my previous recommendations and I could recommend it even if it is not the current show on my TV". Still in recommendations task, one participant said: "It was interesting to recommend negatively like a thumbs down"

On Highlights screen, one participant expected information about the TV shows that would be exhibited that day, in the future.

Checking if a recommendation was followed was many times very hard to do, regarding the lack of information when the recommendation was not already followed. Here, the users solved the problem when they scrolled and saw an example of a followed recommendation. One participant said: "only after some usage of the application is possible to distinct if a recommendation was followed". Here most of them tried to click on the recommendation row in order to get details but the row was not clickable. An interesting suggestion was to use same icons that applications like Facebook and Whatsapp use for giving feedback about seen messages. Therefore, two check signs could be used, one for when the recommendation is sent and other for when the recommendations is followed, turning green when their related actions happen.

In the side bar menu, recommendations icons were not suggestive. One participant recommended the use of a green arrow pointing right for recommendations sent and a red arrow pointing left for recommendations received, like call dialogs on mobile phones.

About the TV guide, there were problems finding a channel. Participants in most of the cases expected to have the EPG ordered by channel position and not by its popularity. The first comment was that having information about show's popularity and its position in ranking is a type of information expected to be on the left side of a view. In place to switch the order of the EPG, participants would appreciate to have buttons for switching between popularity criteria or channel position criteria, having the latter as default. A participant who watched less than 1 hour per day on average said: "For me, it would be interesting to ordered it alphabetically by channel name". Some of the participants also complained about how information about popularity and ranking position was presented in TV guide. Some of them did not understand that it was popularity information and other confused ranking position with channel position. This suggests improvements about popularity data presentation. Another confusion was that the TV guide was the most popular show in all time, missing the list header with "The Most Popular Live Shows" label.

Other problem that participant had was to find friends. Some of them understood that in "What my friends are watching" screen was possible to get it but most of them expected a menu option saying "Friends". On Friend Details screen, there was a big appreciation on the button for start watching the same show as friend. One participant smiled and expressed: "How cute! This is very funny". Moreover, this button had the most curious action on the whole trial. The youngest participant, a 12 year old boy, pressed this button and looked for the TV in place to check what a friend was watching. This proves that Social Zapping can be used in a natural way, keeping focus on the main screen.

In general, participants found some linguistic problems in headers labels. This result shows that the application's vocabulary should be reduced and offer more consistency on linguistic.

4.5 Discussion

4.5.1 Usability

The results about usability are positive, considering the total average of 8.6 regarding all tasks. That fact should not hide the strong problems participants had in some tasks. Some of them (17 and 1) were considered impossible or hard to perform by some participants and that tells me that they had usability problems. The reasons for some bad scores can be related to user's comments about design and/or linguistic problems.

Figure 4.3 shows that the tasks more hard to perform were related to the main action, make a recommendation, with tasks 1 and 17 on top of the hardest tasks to conclude.

Besides that, a good result was to find almost always a score of 10 in each task, by more than one participant in most cases, which reveals that some people considered Social Zapping natural to use.

4.5.2 Overall Opinion

Fun

In the overall opinion, answers about fun were positive with an average of 8.2, which means that people had entertainment with the application. Considering the participants who gave a low score, the graph in the Figure 4.5 shows that there is no relation between the fun they had and the number of hours they watch TV per day, as could be expected. This is considered a good result, considering that does not matter if the user watches much hours of TV per day or not, he always has a good time with Social Zapping.

Design

With an average of 8.1, the results about design show that participants enjoyed the Social Zapping design, look and feel. In fact, not having a single answer below 6 enhances this finding.

Navigation

After discussing the overall opinions about navigation, they should be confronted with usability results. In Section 4.4.1 the average was 8.6 where here the average opinion was 5.7. This shows that, when isolating how hard was to perform a task, there were many tasks easy to perform but it in general participants considered Social Zapping hard to use. Considering the comments, and the task found to be more difficult to perform, design decisions had a strong impact in this result, specially the floating action button used in Highlights screen (3.3.3).

The previous result about usability is richer because the evaluation was composed by a set of 22 task evaluations. As so, the hardest tasks scores were not so much to decrease

the average to a negative score. Giving that, a few tasks were impossible to navigate but that fact should not cause me to consider that Social Zapping is not so hard to navigate, even needing strong improvements in critical tasks.

Utility

These questions aimed to get an insight about participants interest in having Social Zapping as a commercial product. With an average of 7.6, Social Zapping would be useful to some people but not so much interesting as could be expected. This can be observed as a good result when aiming engagement. The relation found between TV consumption(4.5) and utility is very interesting, giving an insight that the more people watch TV the more interesting Social Zapping can be. Hence, the results would be expected to be better if the trial had recruited more people who consume more TV.

4.5.3 Engagement

When asked to evaluate how interesting were each of Social Zapping features, participants reveal a strong interest with the worst average being 7.1.

Here, it is possible that most of them were less interested in TV shows or users rankings. Moreover, it was confirmed that this type of social interaction integrating TV content can be interesting, giving that the top 5 more interesting features were "Click on a recommendation and automatically starts to play it on my TV", "See who is the friend who followed more recommendations I made", " See the TV show most recommended by my friend", "See the friends who recommended a show" and "Recommend a show to a friend".

Chapter 5

Conclusion

Gathering all the work done in this thesis, here I present a summary of my achievements, barriers, problems and the main findings.

5.1 Achievements

Remembering what this thesis aimed, the goals defined were Design and Usability, Focus on the TV content, Social TV engagement and Evaluation. The following Sections detail these goals and if they were achieved. In summary, the goal was to propose the design, implementation and evaluation of an effective social TV application, interacting with a with a Internet Protocol television (IPTV) system on a mobile device.

5.1.1 Design and Usability

In order to evaluate if these goals were obtained, results about Usability on Section 4.5.1, Navigation on Section 4.5.2 and Design Section 4.5.2 were obtained.

Considering the results obtained on users feedback about how hard was to perform each task, I can consider that the usability goal was obtained in general. I cannot consider it totally achieved giving that there were some tasks non-trivial to perform in a natural way. Moreover, when I asked what was the overall opinion about navigation the average was negative (5,7). This happens because participants considered that the main action of the application, recommend a TV show to a friend, was hard to perform.

An important finding to retain here is the design problem that the integration of the Floating Action Button, a Material Design' specification (See Section 2.5.1), brought. In fact, only people who had already saw this type of button associated to main actions in other applications (e.g. Google Inbox) quickly recognized its usage. Even recognizing it, there was a lack of flexibility here giving that participants expected to have this main button always present in the navigation bar or when entering the TV show' detail screen.

The other portion of this goal was to offer an appealing and perceptive interface. Analysing

if this goal was achieved, results about overall opinion on the design of Social Zapping show that the participants who experimented Social Zapping were very pleased with the look and feel of the application, giving the average answer of 8,1 on their evaluation. Considering some comments collected during the trial, participants appreciated the familiar design, i.e., a design that uses the last trends and standards specified by Google for Android applications. This was an important decision made during analysis and design phase and proved to be one of the main factors for people to engage in the application, regarding the few seconds of fame a mobile application has nowadays to catch the users attention.

Moreover, the design proposed for tablets screen size was well leveraged to present all the required information in each screen, and usability tests confirmed that in general, people found the information in a natural way. Here, I confirm that the issue surrounding the design space on STB application its eliminated by having a second screen design space to show complementary information about the main screen. Besides that, a satisfying performance for the application's response times was aimed, with participants having no complains about it or turned their eyes off the screen. Here, it is important to note that server response times are closely related to the Cypher queries response times on Neo4J database server. Giving this, a good performance was obtained, satisfying non-functional requirement NFR3 3.1.1, with these queries but the database size must be considered. For the trial, I used a server virtualized on a machine located at France, with 4 RAM memory, accessed by its public domain and requested from Caixa Mágica Software headquarters in Lisbon, Portugal. During the implementation, I stored the EPG for a day on a daily basis. When I reached a historic of a month and half of EPG information, I noticed that queries were taking much more time to execute. This lead to several improvements on the construction of such queries but even with those improvements, the performance was not reasonable. For the trial, I reduced the EPG historic to the current day only and that way, the response times were perfectly satisfying.

Thus, the Design goal was achieved with participants enjoying the look and feel of Social Zapping and the Usability goal was achieved in general, even with some barriers. Regarding the non-Functional requirements defined on 3.1.1, NFR1 and NFR6 were satisfied.

5.1.2 Focus on the TV content

Another important goal to achieve was to avoid distracting the viewer from the main screen when using Social Zapping. Aiming to achieve this, Social Zapping displays essential information on a clean and natural placement of elements.

Moreover, Social Zapping achieved the goal of making social tasks automatic with automatic recognition of the current show and automatic play of friends recommendations, enhancing focus on the TV content and avoiding long time overhead on manual synchronization with TV.

The goal was solved but the result required a specific evaluation, measuring the percentage of time that users focus on TV and the tablet, like Holmes, Josephson and Carney did on their study [12].

5.1.3 Social TV engagement

Social TV engagement was also critical for Social Zapping to address. Social Zapping should be a solution that engages viewers to have a better TV experience through social networking. This goal evaluation lacked on results from a trial on viewers homes, their natural environment of TV consumption. This would require a recruitment agency to provide participants with different TV consumption levels to be available to freely use Social Zapping, and to have counters of recommendations and their followings.

Considering that this trial could not be done in the scope of this thesis, the trial I have executed with the users sample described on Section 4.3 aimed to evaluate their engagement by asking them to give a score of interest to each one of the features and to give their overall opinion about how useful Social Zapping could be to them if it was a commercial product. Hence, results discussed on Sections 4.5.3 and 4.5.2 give me a clue about the participants engagement. The results show that they were interested in using Social Zapping due to its interesting features. Here, I got important insights. First of them was finding that as more a participant consumed TV the more useful he found Social Zapping to be. This reveals that Social Zapping offers something not available on the market to those participants. Another interesting finding were the top 5 most useful features, accordingly to participants. Here, receiving and making recommendations engaged the users, but it only shows that social recommendation about TV content is required. A stronger insight was that among the top features were Social Zapping novelties: click on a friend recommendations and automatically start to watch it on TV, receive TV content recommendations about TV shows their friends actually saw and having total feedback about the following of recommendations. For the participants, this rules were extremely interesting, giving that they were always curious to know if a friend actually sees what they recommend and how often. Moreover, the rule that restrict recommendations to the tuned show was considered a little bit annoying for who wants to recommend but it was acceptable when they found that their friends recommendations also respect this rule. Thus, Social Zapping clearly offers added value to the users with its features and it is not just an attempt to create a new social network.

Here, engagement was not increased by the idea of contribution to the show's popularity, regarding the low interest on TV shows ranking system. In spite of that, participants appreciated the EPG order by popularity. Regarding the non-functional requirements defined on 3.1.1, NFR2 and NFR7 were satisfied.

Along these lines, Social Zapping achieved the engagement goal on the executed trial.

5.1.4 Evaluation

In place to have the evaluation goal aimed, some questions required answer:

- Q1: Is easy to understand the interface?
- Q2: Does it have an appealing interface?
- Q3: Does it have quick reactions?
- Q4: How much time it takes to perform a task?
- Q5: Will the viewers intend to use the solution to talk about the show, instead of using Facebook or Twitter applications? Does it engage the viewers in the TV content or distract them?
- Q6: Does it offer a better TV experience?

Regarding these questions, evaluation work allowed to answer Q1, Q2, and Q3 positively as discussed in the previous Sections (5.1.1 and 5.1.3). The remaining questions could not be answered by the evaluation work due the lack of a comparison procedure, between Social Zapping and previous solutions and measure of the time spent to perform each task. The latter was considered not relevant, given that no comparison method was done. Despite not answering all the questions, new questions arised and important insights were possible through the online survey. This online survey was extensive and proposed the completion of 22 tasks besides the overall opinion questions. In one hand, performing 22 tasks in a row may have bored the participants and influenced their opinion on fun. In the other hand, with such extensive trial, it was possible to perform an evaluation about design, usability, engagement and expectations and have detailed results, with high granularity detail, allowing to evaluate what this work aimed.

Hence, evaluation goal was achieved with satisfying results, regarding the sample used.

Social Zapping allowed to evaluate, through an Android application for tablets, how Social TV and Second Screen must be combined in order to improve the TV viewer experience, integrating social networks features with the TV content. Findings about users interest on having recommendations feedback and having rules that give more trustworthiness about the recommendations receive like who recommends having to be watching the show were the most relevant results. Concerns about usability and the design had strong impact on the positive feedback that Social Zapping had. The work done in the design phase, doing brainstorming along with information systems experts, many iterations of mock-ups creation, high-level prototyping, experts preliminary usability evaluation, pursuit of the latest design trends on mobile applications and implementation of design standards were

all worthwhile observing the evaluation results. Regarding the Non-Functional requirements defined on 3.1.1, NFR5 was satisfied.

In summary, all goals were achieved with satisfying results but they also showed that this work is not finished. During this thesis, a new product was launched in the social TV and second screen market on Portugal. NOS Iris ¹ reveals some Social Zapping beliefs. Here is possible to verify the assumption on automatic check-in, i.e., get the current channel being watched communicating directly with the STB. Moreover, channel switch along live shows and even previous shows in the last 7 days is possible. The following section presents future work.

5.2 Future Work

Considering participants comments and suggestions, Social Zapping must be improved in its design and usability, using evaluation results and users feedback. There must be a new design proposal for the recommend button, the main action. Many other details in design must be improved or corrected to eliminate the usability and flexibility flaws.

Along with these improvements, Social Zapping should be evaluated by users in their natural environment, the living room, setting up a trial on a period no less than a month, in order to measure its engagement.

In respect to the implementation, Social Zapping is a prototype and some components have to be improved. Database solution shall be verified in order to be scalable, keeping the current performance with at least the last 7 days of the EPG stored, regarding the trends of time shifting. Additionally, Web Services must be created in order to remove the direct querying from the mobile application and offer that as web services. This would assure more security and specially, more flexibility about future changes on queries.

Moreover, additional features should be integrated and evaluated in Social Zapping. Even not having participants expressing the need of Twitter integration, in some cases and with the right support for hashtags recognition, Twitter integration is obviously a feature that Social Zapping should have.

Besides that and accepting that IPTV operators revealed interest on providing clips sharing, recommending moments could be an additional feature to add on Social Zapping, having the same recommendation logic as an entire show. Here, the user could recommend a TV moment, not compromising the TV legal rights because the moment recommended would only be available on another TV. This is already possible in terms of available technology, but there must be also an opening of the TV content providers.

Another compelling idea to implement on Social Zapping are the findings described on Section 2.3 about Time Shifting. Social Zapping is prepared to offer time-shifting follow

¹<https://play.google.com/store/apps/details?id=pt.nos.flic>

of recommendations but it should offer synchronized experiences not only when a user is watching live shows but also when watching delayed. A feature as time-shift streaming of Tweets, like Tomorrowish (2.4.2) does, is an interesting idea.

Appendix A

Cypher Queries

```
MERGE (n:FacebookUser {facebook_id : {id}})
ON CREATE SET n.username = {name}, n.birthday = {birthday},
n.location = {location}, n.link = {link},
n.photoUrl = {photoUrl}, n.coverPhotoUrl = {coverPhotoUrl},
n.recommendations = 0, n.recommendations_followed = 0,
n.watching = 1
ON MATCH SET n.username = {name}, n.birthday = {birthday},
n.location = {location}, n.link = {link},
n.photoUrl = {photoUrl}, n.coverPhotoUrl = {coverPhotoUrl}
RETURN n
```

Listing A.1: Cypher query to add/update a node for a user who logged in the application via Facebook

```
MERGE (i:FacebookInterest { facebook_id : {id},
category: {category}, created_time: {created_time},
interest_name: {name}})
RETURN i
```

Listing A.2: Cypher query to add/update a Facebook interest

```
MATCH (u:FacebookUser {facebook_id:{facebook_id}}),
(i:FacebookInterest {facebook_id:{interest_id},
category:{category}, created_time:{created_time},
interest_name: {interest_name}})
MERGE (u)-[r:LIKES]->(i)
RETURN r
```

Listing A.3: Cypher query to create the relationship LIKES between a FacebookUser and a FacebookInterest

```
MATCH (u:FacebookUser {facebook_id:{facebook_id}}),
(tf:FacebookUser {facebook_id:{friend_id}})
MERGE (u)-[r:FRIEND]->(tf)
```

```
RETURN r
```

Listing A.4: Cypher query to create the relationship FRIEND between two FacebookUser nodes

```
MATCH (u:FacebookUser{facebook_id:{facebook_id}})-
[r:FRIEND]->(f:FacebookUser{facebook_id:{friend_id}}),
(p:Program{eventId:{eventId}})
MERGE (u)-[r2:RECOMMENDS{eventId:{eventId},
channelName:{channelName},
title:{title}, imageUrl:{imageUrl}, date:{date},
msg:{msg}}]->(f)
SET p.recommendations = toInt(p.recommendations)+1,
u.recommendations = toInt(u.recommendations)+1
RETURN r2
```

Listing A.5: Cypher query to get create a recommendation from a user to another

```
MATCH (u:FacebookUser{facebook_id:{facebook_id}})-
[r:FRIEND]->(f),
RETURN f.facebook_id, f.username, f.photoUrl
```

Listing A.6: Cypher query to get all friends

```
MATCH (u:FacebookUser{facebook_id:{facebook_id}})-
[r:FRIEND]->(f),
(u)-[r2:RECOMMENDS]->(f)
RETURN f.facebook_id, f.username, f.photoUrl, count(r2)
ORDER BY count(r2) DESC
LIMIT 3
```

Listing A.7: Cypher query to get the top 3 friends regarding the number of recommendations made to them

```
MATCH (p:Program)
WHERE p.startDate <= {date}
AND p.endDate > {date}
AND NOT p.channelName =~ '(?i).*hd.*'
WITH sum(toInt(p.recommendations)) as totalRecommendations,
sum(toInt(p.recommendations_followed)) as
totalRecommendationsFollowed
MATCH (p:Program)
WHERE p.startDate <= {date}
AND p.endDate > {date}
AND NOT p.channelName =~ '(?i).*hd.*'
WITH p as p, totalRecommendations as
totalRecommendations,
```

```

totalRecommendationsFollowed as
totalRecommendationsFollowed,
CASE WHEN totalRecommendations > 0
THEN toFloat(p.recommendations)/
toFloat(totalRecommendations)
ELSE 0 END as recommendationsRecall,
CASE WHEN totalRecommendationsFollowed > 0
THEN toFloat(p.recommendations_followed)/
toFloat(totalRecommendationsFollowed)
ELSE 0 END as recommendationsFollowedRecall
RETURN p,
    totalRecommendations,totalRecommendationsFollowed,
recommendationsRecall, recommendationsFollowedRecall,
(recommendationsRecall*0.5)+
(recommendationsFollowedRecall*0.5) as ranking
ORDER BY ranking DESC, p.channelPosition ASC

```

Listing A.8: Cypher query to return the live shows(except HD channels) ordered by their recommendation ranking

```

MATCH (n{eventId: {eventId}})
RETURN n

```

Listing A.9: Cypher query to get a TV show

```

MATCH (u:`FacebookUser`{facebook_id:<facebook_id>})-
[r:LIKES]->(i)
WHERE i.interest_name =~ '(?i).*<title>.*'
RETURN i

```

Listing A.10: Cypher query to check if a user has a Facebook interest in the title of a TV Show

```

MATCH (a{facebook_id:{facebook_id}})-[r:RECOMMENDS]->(b)
WITH b.facebook_id as bId, b.username as bUsername,
b.photoUrl as bPhoto, max(r.date) as rDate
RETURN bId, bUsername, bPhoto, rDate
ORDER BY rDate DESC
LIMIT 3

```

Listing A.11: Cypher query get 3 most recently recommended friends

```

MATCH (l1)<-[r11:LIKES]-(a{facebook_id:{facebook_id}})-
[r:FRIEND]->(b)-[r12:LIKES]->(l2)
WHERE l1.interest_name = l2.interest_name
RETURN b.facebook_id, b.username, b.photoUrl,
collect(l1.interest_name)

```

Listing A.12: Cypher query get friends with common Facebook interests in TV content

```

MATCH ()-[r:RECOMMENDS]->
(u:FacebookUser{facebook_id:{facebook_id}}),
(p:Program)
WHERE r.eventId = p.eventId
WITH distinct(r.eventId) as eventId,
p.startDate as startDate, p.title as title,
p.channelName as channelName, p.imageUrl as image,
count(r) as numberOfRecommendations
RETURN eventId, channelName, title, image,
numberOfRecommendations, startDate
ORDER BY numberOfRecommendations DESC, startDate DESC
LIMIT 10

```

Listing A.13: Cypher query to get the 10 most recommended TV shows for a user

```

MATCH (f)-[r:RECOMMENDS]->
(u:FacebookUser{facebook_id:{facebook_id}})
RETURN r.date, f.facebook_id, f.username,
r.eventId, r.channelName, r.title, f.photoUrl, ID(r), r.msg
ORDER BY r.date DESC

```

Listing A.14: Cypher query to get all recommendations for a user

```

MATCH (u)-[r:RECOMMENDS]->(), (p:Program)
WHERE ID(r) = {relationshipId}
AND r.eventId = p.eventId
SET r.followDate = {followDate},
p.recommendations_followed =
toInt(p.recommendations_followed)+1,
u.recommendations_followed =
toInt(u.recommendations_followed)+1
RETURN r

```

Listing A.15: Cypher query to register a recommendation follow

```

MATCH (f)<-[r:RECOMMENDS]-
(u:FacebookUser{facebook_id:{facebook_id}})
RETURN r.date, f.facebook_id, f.username, r.eventId,
r.channelName, r.title, f.photoUrl, ID(r), r.msg,
r.followDate
ORDER BY r.date DESC

```

Listing A.16: Cypher query to get all the recommendations made by a user

```

MATCH (u:FacebookUser{facebook_id:{facebook_id}})-
[r:RECOMMENDS]->(f)
WHERE HAS(r.followDate)
RETURN distinct(r.eventId) as pid, r.title as title,
r.channelName as channelName, r.imageUrl as imageUrl,
count(r) as recommendationsFollowed
ORDER BY count(r) DESC
LIMIT 10

```

Listing A.17: Cypher query to get a user most followed recommendations

```

MATCH (u:FacebookUser{facebook_id:{facebook_id}})-
[r:RECOMMENDS]->(f)
WHERE HAS(r.followDate)
RETURN distinct(f.facebook_id) as fid,
f.username as username, f.photoUrl,
count(r) as recommendationsFollowed
ORDER BY recommendationsFollowed DESC
LIMIT 10

```

Listing A.18: Cypher query to get the top 10 followers of recommendations made by a user

```

MATCH (u:FacebookUser)
WITH sum(toInt(u.recommendations)) as RFt,
sum(toInt(u.recommendations_followed)) as RFSt
MATCH (u:FacebookUser)
WITH distinct(u.facebook_id) as uid,
u.recommendations as RFu,
u.recommendations_followed as RFSu,
RFSt as RFSt, RFt as RFt,
CASE WHEN RFt > 0
THEN (toFloat(u.recommendations)/toFloat(RFt))
ELSE 0 END as recsContr,
CASE WHEN RFSt > 0
THEN
(0.5*toFloat(u.recommendations_followed)/toFloat(RFSt))
ELSE 0 END as followingsContr
RETURN uid,RFu,RFt,RFSu,RFSt,recsContr,followingsContr,
(0.5*recsContr)+(0.5*followingsContr) as ranking
ORDER BY ranking DESC

```

Listing A.19: Cypher query to get the users ranking

```

MATCH (p:Program)
WHERE p.startDate <= {date}

```

```

AND p.endDate > {date}
AND NOT p.channelName =~ '(?i).*hd.*'
WITH sum(toInt(p.recommendations)) as totalRecommendations,
sum(toInt(p.recommendations_followed)) as
totalRecommendationsFollowed
MATCH (p:Program{eventId:{eventId}})
WITH p as p, totalRecommendations as totalRecommendations,
totalRecommendationsFollowed as
totalRecommendationsFollowed,
CASE WHEN totalRecommendations > 0
THEN toFloat(p.recommendations)/
toFloat(totalRecommendations)
ELSE 0 END as recommendationsRecall,
CASE WHEN totalRecommendationsFollowed > 0
THEN toFloat(p.recommendations_followed)/
toFloat(totalRecommendationsFollowed)
ELSE 0 END as recommendationsFollowedRecall
RETURN p, totalRecommendations,
totalRecommendationsFollowed, recommendationsRecall,
recommendationsFollowedRecall,
(recommendationsRecall*0.5)+
(recommendationsFollowedRecall*0.5) as ranking
ORDER BY ranking DESC, p.channelPosition

```

Listing A.20: Cypher query to get a show ranking

```

MATCH (u)-[r:RECOMMENDS]->(f)
WHERE (u.facebook_id = {facebook_id}
AND f.facebook_id = {friend_id})
OR (u.facebook_id = {friend_id}
AND f.facebook_id = {facebook_id})
RETURN r.date, u.facebook_id as recommender_id,
u.username as recommenderName,
f.facebook_id as recommended_id,
f.username as recommendedName, r.eventId, r.channelName,
r.title, r.imageUrl, u.photoUrl, f.photoUrl, ID(r), r.msg,
r.followDate
ORDER BY r.date DESC

```

Listing A.21: Cypher query to get all recommendations made between two users

```

MATCH (u:FacebookUser{facebook_id:{facebook_id}})-
[r1:FRIEND]->(f)-[r2:RECOMMENDS{title:{title}}]->()
RETURN distinct(f.facebook_id), f.username, f.photoUrl,
count(r2) as recommendations
ORDER BY recommendations

```

Listing A.22: Cypher query to get friends who recommended a given show

```
MATCH (u:FacebookUser{facebook_id:{facebook_id}}),
      (p:Program)
WHERE toInt(u.watching) = toInt(p.channelPosition)
AND p.startDate <= {date}
AND p.endDate > {date}
RETURN p
```

Listing A.23: Cypher query to get the current TV show a user is watching

```
MATCH (u:FacebookUser{facebook_id:{facebook_id}}),
      (p:Program{eventId:{eventId}})
SET u.watching = p.channelPosition
```

Listing A.24: Cypher query to get the current TV show a user is watching

Appendix B

Evaluation Guidelines

- Choose the participants regarding the application target. The result will only be good as the people tested.
- Early in the design process, usability testing with a small number of users (approximately six) is sufficient to identify problems with the information architecture (navigation) and overall design issues
- Focus on the key goals and tasks for the application
- Make clear to the participants that they are not the ones being tested but the application
- During the interview, provide clear instructions, get the user familiar with the environment and ask feedback for about they expect or what they would like to have.
- Take notes about the first impression.
- Ask the user to suggest tasks. It may suggest new features or priorities
- Give each task at a time considering not to alter the user approach or intimidate them.
- Make the participant know that any feedback is valuable
- The only time you should speak is to help the participant to give an opinion.
- Always ask for suggestions. It can provide insight about how the application can be better
- Ask the participant what they remember about the application
- The use of familiar formatting and navigation schemes makes it easier for users to learn and remember the layout of an application. Use familiar conventions.

- During reporting, break up the text notes from each participant into a set of simple propositions and use their words as much as possible
- Be neutral and encouraging
- Don't judge or dismiss
- Build a rapport with the interviewee
- Make open-ended questions like "How would you do this?"
- Do not settle with the first answer. Make follow-up questions like "What is an example of that?"
- Answer questions with questions to avoid helping participants
- Do not try to convince the participant that the application is wonderful
- Remember that the goal is not to get an ego massage
- Watch facial expressions, body language and tone. Example: check if they are nervous, excited or bored

Appendix C

Neo4J Database Access in Android

Below there is an example of how the Neo4J Database Access in Android was done via REST API and with the Asynchronous HTTP client library:

```
static AsyncHttpClient client = new AsyncHttpClient();
static SimpleDateFormat sdfDate =
    new SimpleDateFormat("yyyy-MM-dd HH:mm");
static{
    client.addHeader("Authorization", "Basic
        <base64encodedparams>");
    client.addHeader("Accept", "application/json;
        charset=UTF-8");
    client.addHeader("Content-Type", "application/json;
        charset=UTF-8");
}

Date now = new Date();
String nowDateStr = sdfDate.format(now);
StringEntity entity = null;
JSONObject raw = new JSONObject();
JSONObject params = new JSONObject();
try {
    raw.put("query",
        Constant.CYPHER_QUERY_SET_FOLLOW_RECOMMENDATION);
    params.put("relationshipId", relationshipId);
    params.put("followDate", nowDateStr);
    raw.put("params", params);
    entity = new StringEntity(raw.toString(), "UTF-8");
} catch (JSONException e) {
    e.printStackTrace();
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}

client.post(context, Constant.NEO4JDB_BASE_URL_PUBLIC, entity,
```

```
"application/json; charset=UTF-8", new
JsonHttpResponseHandler() {
@Override
public void onSuccess ( int statusCode, Header[] headers,
    JSONObject response){

@Override
public void onFailure ( int statusCode, Header[] headers,
    Throwable e, JSONObject errorResponse){
    e.printStackTrace();
}

@Override
public void onFailure ( int statusCode, Header[] headers,
    String errorResponse, Throwable e){
    e.printStackTrace();
}
});
```

Bibliography

- [1] Jorge F. Abreu, Pedro Almeida, Ricardo Pinto, and Vanessa Nobre. Implementation of social features over regular IPTV stb. *Proceedings of the seventh european conference on European interactive television conference - EuroITV '09*, page 29, 2009.
- [2] Mariana Baca and Henry Holtzman. Television meets facebook: Social networking through consumer electronics, sep, 2008.
- [3] Santosh Basapur, Hiren Mandalia, Shirley Chaysinh, Young Lee, Narayanan Venkataraman, and Crysta Metcalf. FANFEEDS: Evaluation of Socially Generated Information) eed on Second Screen as a TV Show Companion. pages 87–96, 2012.
- [4] Jan Bobeth, Johann Schrammel, Stephanie Deutsch, Michael Klein, Mario Drobics, Christina Hochleitner, and Manfred Tscheligi. Tablet, gestures, remote control?: influence of age on performance and user experience with itv applications. In *TVX '14 Proceedings of the 2014 ACM international conference on Interactive experiences for TV and online video*, pages 139–146, Newcastle Upon Tyne, UK, jun 2014. TVX 2014, ACM.
- [5] Pablo Cesar, Konstantinos Chorianopoulos, and Jens F. Jensen. Social television and user interaction. *Computers in Entertainment*, 6(1):1, May 2008.
- [6] Pablo Cesar and David Geerts. Past, Present, and Future of Social TV: A Categorization. 2010.
- [7] L. Cruickshank, E. Tseklevs, R. Whithman, and A. Hill. Making interactive tv easier to use:interface design for a second screen approach. *The Design Journal*, 10(3), nov 2007.
- [8] Nicolas Ducheneaut, Robert J Moore, Lora Oehlberg, James D Thornton, Eric Nickell, and Palo Alto. Social tv: Designing for distributed, sociable television viewing. 2000.
- [9] David Geerts, Rinze Leenheer, and Dirk De Grooff. In Front of and Behind the Second Screen: Viewer and Producer Perspectives on a Companion App. pages 95–102, 2014.

- [10] Gunnar Harboe, Noel Massey, Crysta Metcalf, David Wheatley, and Guy Romano. The uses of social television. *Computers in Entertainment*, 6(1):1, May 2008.
- [11] Gunnar Harboe, Crysta J Metcalf, Frank Bentley, Joe Tullio, Noel Massey, Guy Romano, and Motorola Labs. Ambient Social TV: Drawing People into a Shared Experience. pages 1–10, 2008.
- [12] M. E. Holmes, S. Josephson, and R. E. Carney. Visual attention to television programs with a second-screen application. In *ETRA '12 Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 397–400, New York, NY, USA, 2012. ETRA '12, ACM.
- [13] Partha Mukherjee and Bernard J Jansen. LNCS 8393 - Social TV and the Social Soundtrack: Significance of Second Screen Interaction during Television Viewing. pages 317–324, 2014.
- [14] Jeffrey Rubin and Dana Chisnell. *Handbook of usability testing : how to plan, design, and conduct effective tests*. Indianapolis, IN : Wiley Pub., second edition, 2008. University of Washington Libraries. Available at <http://www.lib.washington.edu/usability/resources/guides/tests#section-1>.

