6/15/2015

REED
ELSEVIER

ELSEVIER'S STORE MOBILE APP

Fontys
University of Applied Sciences

GRADUATION REPORT

FONTYS UNIVERSITY OF APPLIED SCIENCES

HBO-ICT: English Stream

Data student:	
Family name , initials:	**Pramadianti, A. D.**
Student number:	**2184771**
project period: (from – till)	**3 February 2015 – 3 July 2015**
Data company:	
Name company/institution:	**Elsevier B.V.**
Department:	**E-Business Department**
Address:	**Radarweg 29, 1043 NX, Amsterdam**
Company tutor:	
Family name, initials:	**Corbeau, Maxime**
Position:	**Manager Web Analytics & SEO**
University tutor:	
Family name , initials:	**Veugen, Agnes**
Final report:	
Title:	**Elsevier's Store Mobile App**
Date:	**June 15th, 2015**

Approved and signed by the company tutor:

Date: June 15, 2015

Signature:

Preface

This report represents what I have done in the last 5 months being a graduation intern in Elsevier BV, a publishing company in terms of Science, Health, and Technology. The project is to explore mobile application as a way to drive acquisition, increase conversion, and improve retention for Elsevier's web store (http://store.elsevier.com/).

During this internship, I've been receiving a lot of help from people around me and therefore I would like to thank them all for their support, kindness, encouragement, and guidance to me. Especially, I thank my company's mentor, Maxime Corbeau, for guiding me through the whole process of the internship, for encouraging me to build a great product, for giving me more knowledge about analytics, e-Business, and technical implementation itself, and mainly for believing in me to finish the project.

I'm also thankful for having Mrs. Agnes Veugen as my school mentor. Thank you so much Mrs. Veugen, for guiding me through the whole process of the internship, especially for giving me advice, feedback, and insight to continue carrying on this internship.

For Elsevier's store team, Fabian Kersten and David Toborek, thank you for the help in defining the business requirements of the project and deciding what solutions needed for Elsevier's web store.

And for people around me, I thank you all for being there, supporting and encouraging me to carry on this internship and graduate.

Table of Contents

Summary

Elsevier B.V. is a world-leading academic publishing company in terms of Science, Health, and Technology. Its headquarters is located in Amsterdam, Netherlands. It employs more than 7,000 people across 24 countries and publishes 2,000 journals and 20,000 books. It reported a profit margin of 36% on revenues of US$3.2 billion. One of so many ways that Elsevier does to generate its revenue is by selling books, journals, and solutions through its web store (store.elsevier.com). Having new ideas to grow the web store, e-Business department of Elsevier, who generates revenue from its web store, would like to explore a mobile app as a way to drive acquisition, increase conversion, and improve retention for Elsevier's store. Therefore it hires an intern to do a project regarding the mobile application for Elsevier's store.

In exploring the mobile application as a way to bring more customers, get people to buy often, and get people to come back more often, the intern has to do research to identify the necessary requirements and technology and implement the application. The development process follows lean start-up methodology in order to define the minimum viable product of the app, publish the app to app store in devices, gather the data about user's behaviour towards the app, and learn from the data if the app is useful for the customers and reaching the goal.

The process of the project includes research, design, implementation, testing, and submission and publication. The research is done to define to what kind of app that would like to be developed, which programming language and framework to use, what target platform of the app, which data storage needs to be used to store the data of the app, the features for minimum viable product, 3rd party services that need to be integrated with the app including for analytics, notifications, sending e-mail, etc., and how to build a successful app.

The design mainly covers wireframe, user interface design, and screen flow of the app to give more visualization about how the app works and looks like.

After some research and design are finished, the implementation starts. The implementation of the app is done weekly, and for each week, there is a version published by the team to be tested if it works properly and effectively. When the application is more or less 60% developed, the app becomes Beta Testing version and is published to app store to be tested by public.

Testing the app mainly is done following usability testing method, where the app is tested to humans to get feedback directly. However, to have better qualified application, unit testing also should be done to validate if the app works as intended and properly.

Unfortunately, the app is not finished yet due to the deadline of the app is longer than the deadline of this report. However, the intern is still finishing the app, so it will be finished and published before the internship ending.

Glossary

API

Application programming interface

A set of routines, protocols, and tools for building software applications.

App

Application

B2C

Business to consumer

Type of commerce transactions between business and customers

Github

A web-based Git repository hosting service, which offers all of the distributed revision control and source code management functionality.

IDE

Integrated development environment

Software application that provides facilities to computer programmers for software development.

JSON

Javascript object notation

A light-weight data-interchange format to be easy to read and write for humans, and to parse and generate for machines.

MVP

Minimum Viable Product

In the product development, it is the product with the core features that allow the product to be deployed.

1. Introduction

Smart phones and tablets are wide spread all over the world. Inside those smart gadgets, there are millions – even billions – mobile applications that are being used daily by most people in this world. People rely on their mobile devices for everything from health to gaming; they are hooked to applications inside the devices without they realize it. Because of that, mobile application market is growing so fast, it was worth $53 billion in 2012 and is forecasted to grow to $143 billion in 2016.

Knowing that mobile app market matures, e-Business department of Elsevier BV wants to explore mobile application as a way to drive acquisition, increase conversion, and improve retention. Even though there are many mobile applications published by Elsevier BV, but none of them is published by its e-Business department. E-Business department of Elsevier generates revenue from Elsevier's web store (http//store.elsevier.com), therefore the department would like to have a mobile application for Elsevier's web store (a B2C value proposition making several million $ yearly).

To find out if a mobile application could bring more customers, get people to buy more often, and get people to come back more often, it requires further discussion about business requirements, research and identifying the necessary technology, defining the technical requirements, and implementing the application itself.

In this report, further details of how the exploration about mobile application going is discussed. This covers the information about the company (Chapter 2), the depth description about the project (Chapter 3), the research discussion and result (Chapter 4.1), the design of the application (Chapter 4.2), how the implementation is done (Chapter 4.3), how to test the application (Chapter 4.4), the process of submission and publication of the app (Chapter 4.5), and the conclusion of this project and the recommendation for improvisation (Chapter 5).

2. Company

Elsevier BV is an academic publishing company that publishes medical and scientific literature. It provides information solutions that enhance the performance of science, health, and technology professionals, empowering them to make better decisions, deliver better care, and make groundbreaking discoveries, that advance the boundaries of knowledge and human progress.

Founded in 1880, Elsevier took its name from the Dutch publishing house Elzevir, booksellers and publisher in Netherlands since 1580. Elsevier is the oldest and largest company dating from that time. However, it doesn't have any connection with Elzevir itself.

Figure 1: Amsterdam Headquarters of Reed Elsevier

Elsevier BV is a part of the Reed Elsevier group, an Anglo-Dutch multination publishing and information company co-headquartered in London, UK and Amsterdam, Netherlands. The Reed Elsevier group is a dual-listed company consisting of Reed Elsevier PLC (listed in London and New York) and Reed Elsevier NV (listed in Amsterdam and New York). It's a merger between Reed International – a British trade book and magazine publisher, and Elsevier – the Dutch science publisher.

Elsevier BV employs more than 7,000 people in over 70 offices across 24 countries, and publishes 2,000 journals and 20,000 books. It reported a profit margin of 36% on revenues of US$3.2 billion. One of so many ways that Elsevier does to generate its revenue is by selling books, journals, and solutions through its web store (store.Elsevier.com). In order to generate more revenue, Elsevier is starting to have more employee recruitment and creating new and challenging projects for students, one of the projects is exploring mobile apps for Elsevier's store. This project is part of their aim to grow the company.

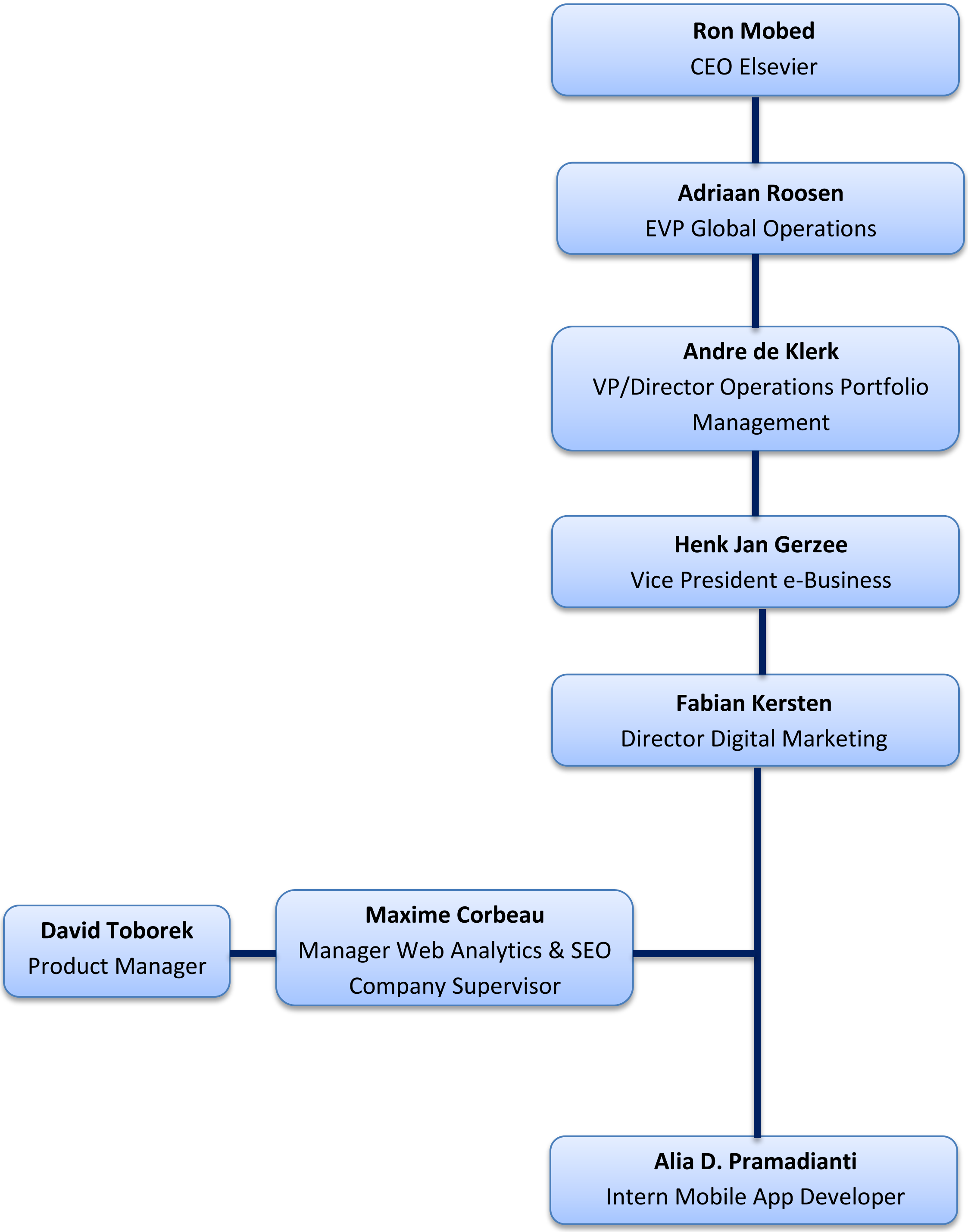


Figure 2: Elsevier BV Organization Flow

3. Project Overview

3.1. Initial Situation

Mobile applications are starting to dominate the mobile world; it is pretty much a hot trend nowadays. It has been steadily growing in terms of revenues and jobs created. The mobile app economy was worth $53 billion in 2012, and the forecast for 2016 is that it will grow to $143 billion. The main trigger behind rocketing mobile app usage is the growing sales of tablets, smartphones and other mobile devices.

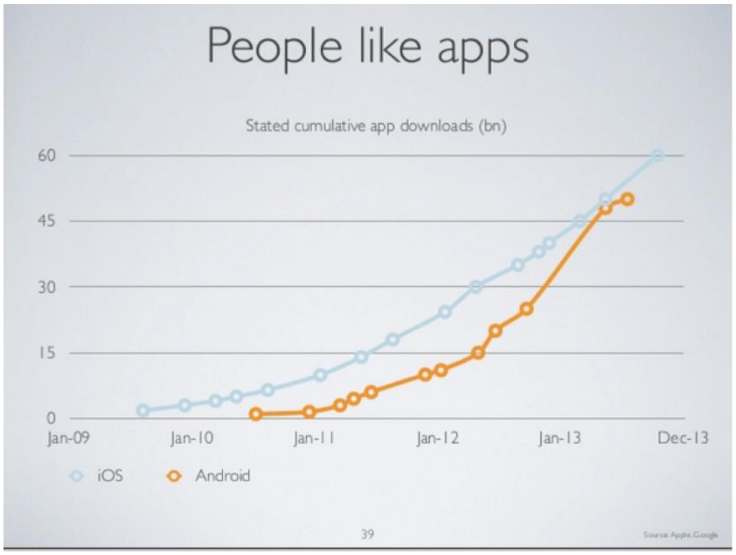


Figure 3: The cumulative chart of app downloaded

Knowing that the mobile app market grows so fast, E-Business department of Elsevier BV considers exploring mobile apps as a way to drive acquisition, increase conversion, and improving retention. There are a lot of mobile applications that are published by Elsevier BV, but none of them is for Elsevier's Store. Therefore, E-Business department, who generates revenue from its web store (a B2C value proposition making several million $ yearly) would like to have a mobile app for its web store.

3.2. Project Objective

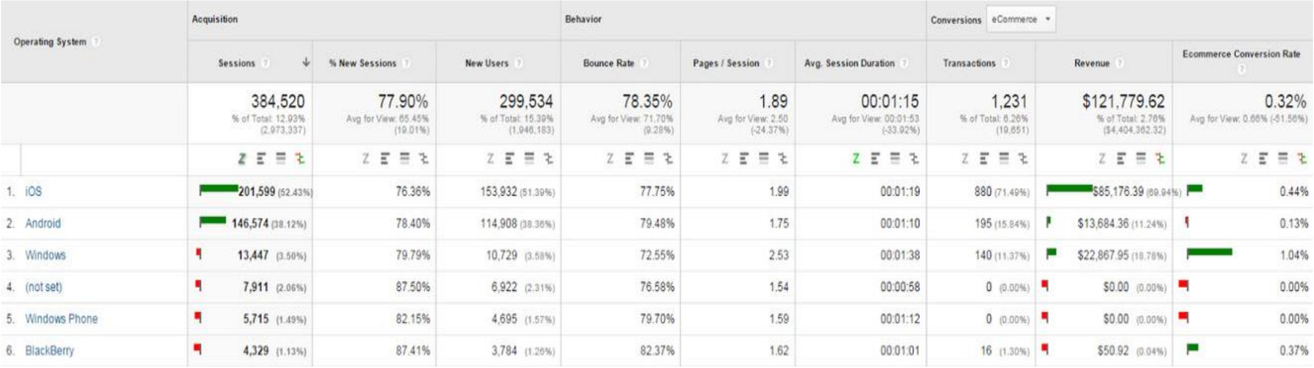
The main goal of the project is exploring mobile apps for Elsevier's store as a way to bring more customers, get people to buy more often, and get people to come back more often. Along with that, the project also aims to give solution to e-Business department of Elsevier in generating more revenues, have additional features that can be useful for Elsevier's customers, and solve the pain points of the web store. The final application will be invested in digital marketing and user feedback for further promotion and improvisation.

3.3. Project Description

To develop a qualified mobile application, it requires further discussion about business requirements, research and identifying the necessary technology, the technical requirements, and the implementation of the application.

Mobile application for Elsevier’s store project is done by a team from Elsevier’s e-Business department. Business requirements are discussed within the team, and the results are decided together. Research and identifying the necessary technology, defining the technical requirements, and implementing the application itself are done by the trainee.

Based on the data taken from analytics of the web store (Figure 4), most of the customers of the web store use iOS, which gives the idea to the team to build a mobile application for iOS.



Operating System	Acquisition			Behavior			Conversions eCommerce		
	Sessions	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Session Duration	Transactions	Revenue	Ecommerce Conversion Rate
	384,520 % of Total: 12.93% (2,973,337)	77.90% Avg for View: 65.45% (19.01%)	299,534 % of Total: 15.39% (1,946,183)	78.35% Avg for View: 71.70% (9.28%)	1.89 Avg for View: 2.50 (-24.37%)	00:01:15 Avg for View: 00:01:53 (-33.92%)	1,231 % of Total: 6.26% (19,651)	$121,779.62 % of Total: 2.76% ($4,404,362.32)	0.32% Avg for View: 0.66% (-51.56%)
1. iOS	201,599 (52.43%)	76.36%	153,932 (51.39%)	77.75%	1.99	00:01:19	880 (71.49%)	$85,176.39 (69.94%)	0.44%
2. Android	146,574 (38.12%)	78.40%	114,908 (38.36%)	79.48%	1.75	00:01:10	195 (15.84%)	$13,684.36 (11.24%)	0.13%
3. Windows	13,447 (3.50%)	79.79%	10,729 (3.58%)	72.55%	2.53	00:01:38	140 (11.37%)	$22,867.95 (18.78%)	1.04%
4. (not set)	7,911 (2.06%)	87.50%	6,922 (2.31%)	76.58%	1.54	00:00:58	0 (0.00%)	$0.00 (0.00%)	0.00%
5. Windows Phone	5,715 (1.49%)	82.15%	4,695 (1.57%)	79.70%	1.59	00:01:12	0 (0.00%)	$0.00 (0.00%)	0.00%
6. BlackBerry	4,329 (1.13%)	87.41%	3,784 (1.26%)	82.37%	1.62	00:01:01	16 (1.30%)	$50.92 (0.04%)	0.37%

Figure 4: Elsevier's store data analytics

And to make the process more flexible, the programming chosen is Hybrid programming, where the developer use website language to build the app, and wrapping them in native container, so the Apple Store or Google Store (stores where mobile users can free download or buy the app from their phone) can accept the app to be in the market. Hybrid also can be used in multiple platforms, which could give the team to have an app for both iOS and Android just by developing using one language.

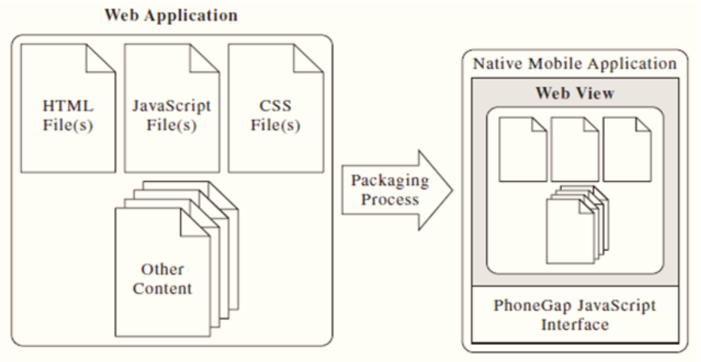


Figure 5: Example of how PhoneGap (one of hybrid frameworks) works

And for the application itself, the team wanted to have a mobile application where the users can buy e-books and read the e-book directly from the app. The users are able to register through the app or login with their data in web store database, search the e-book(s) or browse the e-book(s) by particular subject(s), having a library where they can store their e-book(s), account setting where the users are able to see their profiles, and the payment method integration so the users are able to buy e-book(s) directly. The application also should be able to be integrated with 3rd parties company for the analytics, A/B testing, notifications, etc.

After some research has been done (look at research part to have more details), there are some factors that prevent the team to have Plan A going. Those factors are:

1. The company is not able to provide Macbook for the trainee to build the app, because the development for iOS should be done in Macbook even though it using Hybrid programming.
2. The e-book format should use digital watermarking of the company. Because of this technical security, the implementation of putting e-books in the mobile app and downloading e-book becomes technically complicated, which isn't effective for this project.
3. Payment method integration is not an option where the process is too complicated, which makes it impossible to be implemented for mobile app.
4. Not all 3rd parties companies that the team thinks could be integrated with the app, for the success of the app, have an option to integrate with Hybrid programming. Most of them can just work with Native programming.
5. A duplicate application of the web store itself doesn't focus in solving the pain points of the web store.

Based on those factors, Plan B is created. The mobile application is now developed for Android users, because (as seen in Figure 4), Android users are the second highest in the rank of data analytics of the web store. 3rd party companies and some advices from the others (again, see research for more details) are the reasons that the programming language is changed to Native programming, where the mobile application is developed using its own language. And since the app is now developed for Android users, the development language is Java.

The type of application itself is totally different from the initial plan. Based on pain points found in web store (see appendix B for the details), the team decided to solve one of them. The problem is a bad user experience to find and use the coupon available to buy books and journals in the web store. With the current web store, it's difficult for the user to look for specific coupon codes to use in buying the books and journals, the user will be directed to the same pages as a loop while going to see the details of the code and if the user wants to use the code. In other words, there is no clear state of the flow of the website regarding the coupons. That kind of problem is confusing for the user, and it would give more pains while opening those pages in mobile's web browser. And also, while the user wants to look at special offers given by Elsevier, there are not many available coupons shown in the website when in fact there are more than 50 coupons available for the user. Fixing the website now is not an option because Elsevier's web store in not maintained by e-Business department and such problem needs to be solved soon enough.

Thus, to solve the problem, there will be an app that will serve the users to find Elsevier coupons to be used while buying books and journals in the web store. The team names it as a coupon app, which the contents of the app will be pertinent with the coupons code available from Elsevier. After some discussion, the team realize that the coupon app could be a way to improve retention where the users are going back to the web store to buy something because they have discount from the coupon code, increase conversion where the users buy more because the books' prices are cheaper than normal, and provide better user experience where the users could easily find something that are useful to them.

3.4. Method of Approach

To have a great application which is useful for the customers, there is a method that has been used in approaching the goal of the mobile app for Elsevier's store: Lean Start-up Methodology.

Lean start-up is a methodology based on lean manufacturing Japan 1951. It seeks to eliminate wasteful practices and increase value producing practices during the product development phase. The customer feedback during the product development is integral to lean start-up process, so the producer doesn't invest time designing features or services that consumers do not want.

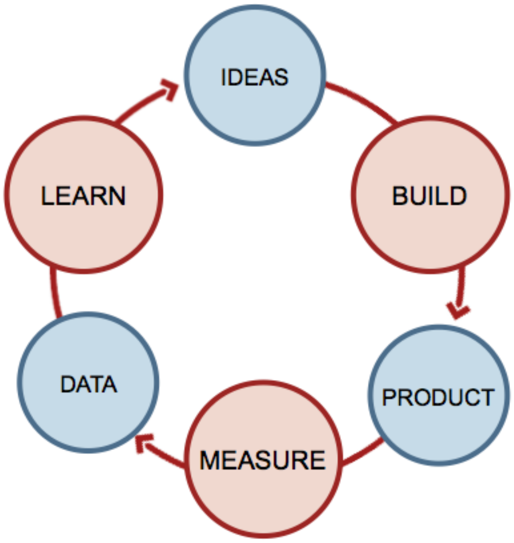


Figure 6: Lean Start-up Process

The core part of lean start-up methodology is Build-Measure-Learn process. It's explaining what the producer should between the phases:

Idea -> Build -> Product -> Measure -> Data -> Learn

In other words, it's a loop process of turning ideas into products, measuring costumers' reactions and behaviour against built products, and then learning whether to persevere the idea; the process is repeated over and over again.

In this project, this method is used to defining the Minimum Viable Product of the application. It's used by implementing each feature (or each product), measuring data based on the time spent to research and build each feature, and learning if the feature is easy to build and useful for the customers.

The method is also used when the app (as a product) is launched and the team could gather data based on the data analytics of the app, and learning if it's useful for the customer and a way to drive acquisition, increase conversion, and improve retention.

4. Project Process

The process of the project is divided into five phases: research, design, implementation, testing, and submission and publication. Research phase is answering the questions about why and how the process is carried. Design phase is about what is designed, what tools needed for the design and why they are chosen, and the explanation of what, how, why the final design is carried. Implementation phase is about the explanation about how the app is implemented, what is implemented, and what tools needed and why they are chosen. Testing phase is explaining what methods are used and how they are done. Submission and publication explains the process of publishing the app.

4.1. Research

The research part is carried using DOT Framework, a research strategy framework consisting five strategies: Field, Library, Workshop, Lab(oratory), and Showroom. Field is used when the research takes place by looking at, searching and investigating the applied area. Library is used to investigate related or available work, such as reading books, internet, blogs, related literature, etc. Workshop is used when 'hands-on' experience happens, such as implementing part of application, designing database, etc. Lab(oratory) is used when testing the aspect of the project against the aspect of Field or Library. And last but not least, Showroom is used when positioning the project against other current available projects.

4.1.1. Hybrid Frameworks

As seen in the project overview, the plan is changed from Hybrid programming to Native programming. But before it's changed, a research to determine which Hybrid framework would be used to develop the app is done. Hybrid frameworks that have been researched:

1. Cordova
2. Ionic
3. Phonegap
4. Mobile Angular UI
5. Software Intel
6. Appcelerator
7. Sencha Touch
8. Telerik Kendo UI

The research context of those frameworks is including how to install, how it works, its target platforms, its ease of use, and how its perpetuity in Github doing.

1. Library

   In this DOT Framework method, research is done by browsing all over internet about the context. Based on the research, the data gotten is:

No	Framework	Platforms				Ease of use	Github Stars
		iOS	Android	Windows	Blackberry		
1	http://cordova.apache.org/	x	x	x	x	Usually used together, and many people use them	856
2	http://ionicframework.com/	x	x	x	x		13293
3	http://phonegap.com/	x	x	x	x	Can be used together with cordova + ionic, or with mobile angular UI	1915
4	http://mobileangularui.com/	x	x	x	x	Its components almost similar with cordova, ionic, and phonegap	1374
5	https://software.intel.com/en-us/html5/tools	x	x	x	x	Integrate with cordova, and use XDK app to build mobile app using templates available	70
6	http://www.appcelerator.com/titanium/	x	x	x	x	Use titanium studio app to create new mobile app project	1763
7	http://www.sencha.com/products/touch/	x	x	x	x		104
8	http://www.telerik.com/kendo-ui	x	x	x			1301

Table 1: Hybrid frameworks with their target platforms and ease of use

Based on the table above, Github stars indicate the number of people showing their appreciations of the frameworks, and so many people use ionic frameworks to build their mobile app. And it's known that most hybrid frameworks can be used in most mobile app platforms, so it gives the team flexibility to choose which platform(s) that the mobile app would be developed in later on.

2. Workshop

   Workshop is used by trying to use all frameworks, installing them and make a simple 'Hello World' project. This helps to determine the ease of use of the framework. Since the development is using Windows laptop, so the workshop method is only done for Android platform.

Based on the research above, the decision of the framework to use might fall to Ionic framework together with Cordova, but actually the final decision hasn't really made due to the changing programming method.

4.1.2. Features for Initial Plan of Elsevier's Store Mobile App

To define the features that could be implemented in the Elsevier's store mobile app, the research is done by looking for existing books mobile app from the competitors' point of view and existing mobile app from Elsevier's point of view.

1. Library

   Library is done by browsing through App Store, Google Play and their websites. Based on the research, the apps found are:

   a. Competitors' Mobile App:

      i. Kindle Amazon
      ii. Kobo
      iii. Barnes and Noble College
      iv. NOOK Barnes and Noble
      v. Springer Link
      vi. Wiley (Blackwell)
      vii. Google Play Books
      viii. iBook
      ix. Wattpad
      x. Aldiko
      xi. Scribd
      xii. eBook reader

   b. Elsevier's Mobile App:

      i. Elsevier eLibrary (Singapore)
      ii. Elsevier NL (News)
      iii. Research Highlights
      iv. Clinics Review Articles

v. The Lancet

vi. Poster in my Pocket

vii. E-volution (Portuguese)

viii. Journal Viewer

2. Workshop

Workshop is done by downloading the applications from App Store and Google Store, and trying them all to get the information they have. They all have similar features, but there are some features that only are in particular app. Summary of the features they have:

i. Register / Login

ii. Books store

iii. Free search for filtering data

iv. Sync

v. Library to keep the books

vi. Account setting

vii. Info / About page, to display the information of the company

viii. Feedback page

ix. Help page, usually display FAQ page

x. Sorting the list

xi. List view or block view of the content

xii. Browse by categories

xiii. Rating of the books

xiv. Table of contents

xv. Content data included books, journals, and articles

xvi. Contact us page, to contact the company

xvii. Shopping cart

xviii. Bookmark of the book

xix. Make something as favourite

xx. Share social media

xxi. Payment method

xxii. Subscribing

xxiii. E-mail the article if the user wants it

Based on the research above, in order to develop a good mobile application, the Elsevier's store Mobile App would like to have the features such as:

i. Social Login / Registration / Logout

ii. Books store

iii. Browse by categories (or subjects and imprints)

iv. Free search (keyword, title, ISBN, ISSN)

v. Coupon code

vi. Library, sorting, list view / block view

vii. Download e-book

viii. Account setting (contact, login details, billing address)

ix. Payment method

x. Share social media

Unfortunately, the features couldn't be implemented due to some limitations from Elsevier such as:

a) Digital Watermarking

   Digital Watermarking is kind of marker to identify ownership of the copyright of Elsevier's books, e-book, and journals. Elsevier's e-books have this because Elsevier is afraid that there will be piracy of its e-books. Because of this technical security, the implementation of putting e-books in the mobile app and downloading e-book becomes technically complicated. Due to this, books store feature and download e-book are not able to be implemented in the end.

b) Payment method

   The payment method for Elsevier's store is too complicated, which makes it impossible to be implemented for mobile app.

4.1.3. What Makes an Application Sticky / Successful?

A library research has been done to know exactly what should be done by Elsevier's store team to have a successful mobile application. This is done by browsing all over the internet and read some articles about it. The ideas gotten from the articles are defined into 5 phases of the business: Acquisition, Activation, Retention, Referral / Virality, and Revenue. Some important ideas that could be taken by Elsevier's store team for the app:

1. Track and measure everything with tracking tools
2. Usability testing
3. Use native code
4. Good marketing
5. Easy to install, access, and use
6. Social media share
7. Free
8. Design according to how the users think about things

For the whole data that can be useful for the team can be found in Appendix C. And based on those data, the Elsevier's store team then have some ideas about how to make a successful app.

4.1.4. Data Storage

In order to help the team to decide which programming method to use, researching about data storage for Hybrid or Native programming is also done. The library method is used to carry this research. The result is:

Data Storage	SQL/NoSQL	Native	Hybrid	Android	iOS	Online/offline access
JSON	NoSQL	X	X	X	X	Both
SQLite	SQL	X	X	X	X	Both
MySQL	SQL	X		X	X	Online
iCloud		X			X	Online
Core Data	NoSQL	X			X	
HTML5 local storage			X	X	X	Offline

Table 2: Comparison data storage to use in hybrid or native programming

Based on the data above, the data storage used is JSON where works well for both Native – Hybrid and Android – iOS. In addition, JSON is also a light data storage which getting request from or posting data into the file is easier and faster.

4.1.5. Hybrid VS Native Programming

To decide between hybrid and native programming to use to program the app, there are some strategies of DOT framework used:

1. Field

   Field is used by having interviews with some people to know which between Hybrid and Native programming is better to use to develop the app for Elsevier's store:

   a. Interview a friend of the trainee who has experience in developing mobile application using Hybrid programming. In the interview, she said that Hybrid programming is too slow for the performance, so she advises that it's better to use Native programming to develop a mobile application. Even though Hybrid programming is using HTML, Javascript, and CSS to build an app, the developer still needs to understand Java (Android native language) and Objective-C (iOS native language) to fix bugs (or errors), because sometimes the errors occurred only can be fixed in native programming. And she also advises that if the app will be connected to API / server, it's better to use native programming because it will load faster.

b. Interview Filippos Koulyras, a product manager of Research Highlights and Journal Viewer – two of many products developed by Elsevier, who is responsible for Research Highlights and Journal Viewer mobile app. In the interview, he said that Journal Viewer is developed using Hybrid programming, which is not continually managed nowadays, because its performance is bad and slow, and the content is difficult to be updated. Using hybrid programming also means having fewer features to be developed. On the other hand, Research Highlights is developed using Native programming, both for iOS and Android, and he's happy with the results because it performs fast and is having better looks and feels of the app. He also said that native application is having better chance to be approved by App Store to be in the market.

c. Interview Optimizely employees, who are responsible for A/B testing for mobile application. Optimizely is one of the third parties that Elsevier works together with. In the interview, they said it would require more works to integrate Optimizely with mobile application using Hybrid programming. They also said that they don't exactly know how to integrate it with Hybrid app, but maybe it would work with Optimizely for the website. In conclusion, if Elsevier wants to integrate the app with Optimizely, the app has to be developed using native language.

2. Library

Library method is used by browsing all over internet just to find the right answers between hybrid and native programming. One of the comparisons taken from the internet is:

	Native	**Hybrid**
App Features		
Graphics	Native APIs	HTML, Canvas, SVG
Performance	Fast	Slow
Native look and feel	Native	Emulated
Distribution	App Store	App Store
Device Access		
Camera	Yes	Yes
Notifications	Yes	Yes
Contacts, calendar	Yes	Yes
Offline storage	Secure file storage	Secure file system, shared SQL
Geolocation	Yes	Yes

Gestures		
Swipe	Yes	Yes
Pinch, spread	Yes	Yes
Connectivity	Online and offline	Online and offline
Development skills	Objective-C, Java	HTML5, CSS, Javascript

Table 3: Native VS Hybrid Programming

Another thing that is taken as consideration to decide is the experiences by well-known companies such as Facebook, Netflix, and Banana Republic. As the co-founder and CEO of Facebook, Mark Zuckerberg, said “The biggest mistake we’ve made as a company is betting on HTML5 over native” at TechCrunch Disrupt conference in San Francisco, regarding the original HTML5-powered Facebook mobile apps that had problems with slowdowns and other issues. According to him, as soon as the hybrid app is replaced with native iOS app, there is a lot of improvement including the double increase of feed stories number consumed by its customers. He also said that everyone could do much better work with native apps.

Figure 7: Mark Zuckerberg interview in TechCrunch

On the other side, Netflix and Banana Republic are using hybrid programming for their apps. Netflix uses same code for its apps on all devices: tablets, phones, smart TVs, DVD players, etc. Its performance of decoding and streaming videos is delegated to the native layer, so it’s fast and seemingly native app, that provides best of both native and hybrid worlds. And Banana Republic has same exact designs for both iOS and Android app.

As can be seen in Figure 8 and Figure 9 below, the tab bottom of the page works better in the iOS app than in Android app, which makes it clearly seen that it’s non-native Android. It happens with the back button in the top bar too, that the back button in Android is useless since Android phone always provide physical back button.

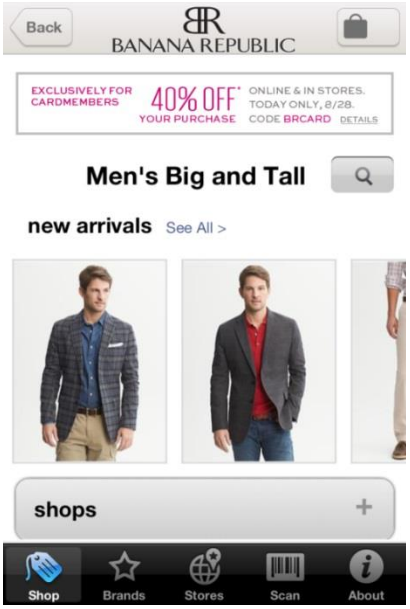


Figure 9: Banana Republic app for iPhone

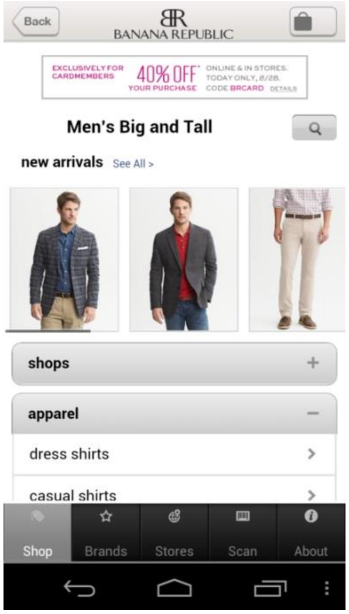


Figure 8: Banana Republic app for Android

Based on the data gotten from the research, the final decision is to use native programming to build the mobile application for Elsevier's store because it would integrate well with 3rd party companies and hopefully will work fast just like most people said about it.

4.1.6. Target Platform

As stated in the subchapter before this – Hybrid VS Native Programming, the programming method that is used for developing Elsevier's store mobile app is Native programming. Therefore, the decision is to develop for only one platform, because native programming can't be for multi platforms.

And as seen in Figure 4 in Project Description chapter, most of Elsevier's web store's users use iOS to buy books / journals. Therefore, the initial decision of target platform is iOS, because most of Elsevier's customers are iOS users. Developing iOS mobile app only can be done in Apple OS, which only can be done either in Apple laptop (Macbook) or Apple computer (iMac), therefore the trainee needs Macbook to develop the app. But, because of the limitations of the company that the company can't provide Macbook for the trainee, the target platform is changed to Android. So, now, the decision is to develop a mobile application for Android users. Another additional reason why Android is chosen is because as seen in Figure 4 again, Android users are the second highest of Elsevier's web store's users rank.

4.1.7. Dependency Manager for Android

In the project development, dependency management is the section to centralize dependency information. The integration of 3rd party companies can be done using dependency manager, so it works properly and fast. An Optimizely employee has ever said that to integrate iOS app with Optimizely, the integration can be done through CocoaPods, a dependency manager especially for iOS and Mac projects. Thus, in order to integrate Android app with 3rd party companies properly, the research regarding Android's dependency manager has been done.

1. Library

   This is done by browsing all over internet to find the best dependency manager for Android. There are two great dependency managers for Android: Gradle and Maven. Both works for any other 3rd party companies such as: Optimizely, Mixpanel, etc.

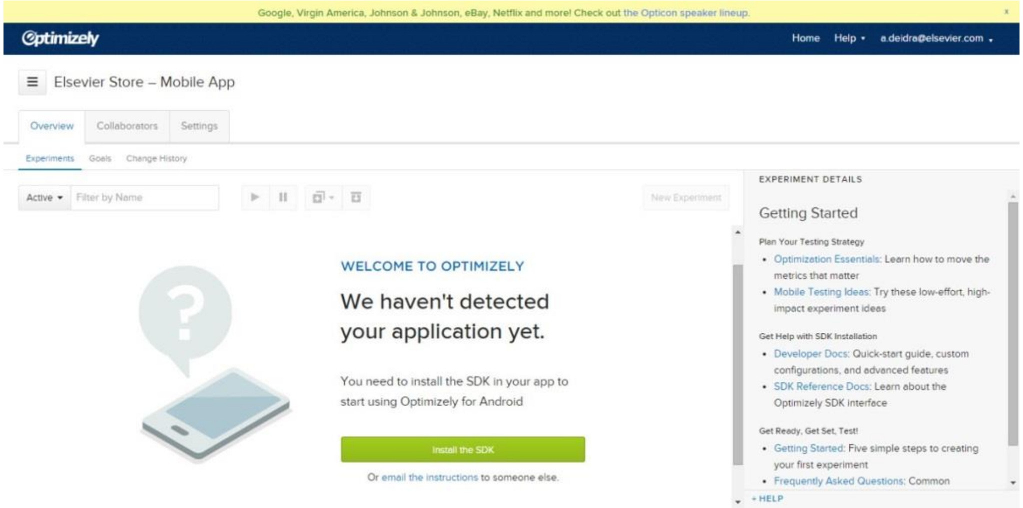
2. Lab(oratory)

   Lab method is done by testing Gradle and Maven in integrating 3rd party companies with Elsevier's store mobile App. In order to use Gradle and Maven, their plugin must be installed in the IDE first, and then install them in the project.

   - Optimizely Integration

     The integration of Optimizely with the app using Gradle is done by following steps written in Optimizely website. When the steps are completely gone through, the app is run to register itself to Optimizely. But unfortunately, the app doesn't work properly and is not registered to Optimizely.

     So, another way is chosen to test the integration again. This time, it's using Maven, and also follows steps written in Optimizely website to integrate them properly. While the steps are done followed, the app is run to register the app to Optimizely. Exactly the same as what happened in integrating the app using Gradle, while the app is running it's supposed to register the app to Optimizely, but it's still not working properly.



WELCOME TO OPTIMIZELY

We haven't detected your application yet.

You need to install the SDK in your app to start using Optimizely for Android

Or email the instructions to someone else.

EXPERIMENT DETAILS

Getting Started

Plan Your Testing Strategy

- Optimization Essentials: Learn how to move the metrics that matter
- Mobile Testing Ideas: Try these low-effort, high-impact experiment ideas

Get Help with SDK Installation

- Developer Docs: Quick-start guide, custom configurations, and advanced features
- SDK Reference Docs: Learn about the Optimizely SDK interface

Get Ready, Get Set, Test!

- Getting Started: Five simple steps to creating your first experiment
- Frequently Asked Questions: Common

Figure 10: Optimizely can't work properly

As seen in Figure 10, even though integrating the app using both Gradle and Maven are done, the app still isn't registered to Optimizely. And it turns out, it doesn't work properly not because Gradle or Maven, but because there is miscommunication between Elsevier and Optimizely regarding the payment to use Optimizely's services. So, it's still not known if it works properly or not using Gradle or Maven.

- Mixpanel Integration

  To integrate the app with Mixpanel, the integration can be done by using Gradle. To integrate Mixpanel properly, following steps written in Mixpanel website is done. While all the steps are gone through, like integrating Optimizely to the app, the app should be run to register itself to Mixpanel.

  Unfortunately, the app is not registered to Mixpanel. It turns out that Mixpanel has its own way for each IDE used for Android. And for IDE that Elsevier's store mobile app use for development, Mixpanel integration couldn't be done using dependency manager, the integration has to be done manually by installing the Mixpanel library to the project. Therefore, the dependency manager here is not really useful.

Based on those researches, Gradle and Maven are not really properly worked. Therefore, Elsevier's store mobile application doesn't use Gradle or Maven for managing dependency information.

4.1.8. 3rd Party Services

Developing a great application doesn't mean building everything by ourselves from the start. There must be some aspects that can't be implemented just because there are more effective ways to do that, one of them is leveraging 3rd party services by integrating them with the app. Thus, in order to give best approaches in maintaining the app, the integration with 3rd party services is done.

Since there are a lot of 3rd party services that can be useful for Elsevier's store mobile app, it doesn't mean that the app has to be integrated with all of them. Therefore, the research regarding 3rd party services is carried.

1. Push Notification

   Push Notification is important for the application because it will help Elsevier's store team to send notification to the user regarding news of the app or anything that's relevant to Elsevier. This feature gives the team options to either build its own server or use 3rd party server to do the transaction between the app and the team. Since developing its own server is going to take more time than leveraging 3rd party services, the team chose to use 3rd party services.

   a. Library

      Library is done by browsing all over internet to know which 3rd party provide push notification service. This gives the options to use Pushbots and Mixpanel. After reading the information about them, it's said that both work well to push the notification to the users.

   b. Lab(oratory)

      To know exactly if the information gotten from the internet is true, testing them is the best option.

      - PushBots

        Testing PushBots is done by making a new Android project and following steps in YouTube video: 'Send GCM Android Push Notifications in less than 7 mins with PushBots' published by PushBots.

        In turns out that PushBots works so well. The lead time between the notification is sent and when it arrives is also fast, which gives additional point to PushBots.

- Mixpanel

  Testing Mixpanel is done by making a new Android project and following steps in Mixpanel tutorial website which gives instruction how to integrate it to the app and how to enable push notification.

  It turns out that Mixpanel works good as well. The lead time between the notification is sent and when it arrives is also fast, which is the same as PushBots. But another interesting point is that using Mixpanel, the push notification can be tracked if it's opened or not, because Mixpanel is also advanced analytics platform for mobile and website which also could help in analysing the users' actions in the app.

Based on the research above, since Mixpanel excels one point in tracking the push notification, then 3rd party service chosen for push notification is Mixpanel.

2. Analytics

   Since this mobile application is to be invested in e-Business department, according to them data is important to know how the business going. So, the integration between the app and the analytics is done. But, to decide which analytics service that's going to be used for Elsevier's store mobile app, the research has to be done, the result is:

   - Google Analytics

     Elsevier's e-Business department already has experience with Google Analytics. Thus, it's really important to them to integrate the app with Google Analytics.

   - Google Tag Manager

     Google tag manager is new for Elsevier's e-Business department. But integrating Google Tag Manager could help the team to maintain the app without going deeper to technical part of the development. Using google tag manager, the team could integrate it to google analytics and define events that happening with the app so the team could analyse the data pushed to google analytics from the app.

     Other than that, Google Tag Manager also could help the team to define the variable needed in the app such as application settings, styles, and characters.

   - Mixpanel

As stated before that Mixpanel is an advanced analytics platform for mobile and website. And it also specialize in people analytics, which means the user data can be sent to Mixpanel to be tracked, which is useful for the team to send notification based on user's preference. With Mixpanel, the team also can have analytics of events happening in the app, for example like tracking the notification open rate and whether people come back or not.

Based on the data acquired, the team then use all of them, because Google Tag Manager can be used together with Google Analytics, thus integrating both of them is like kill two birds with one stone, and Mixpanel is useful for analysing the user.

3. A/B Testing

   A/B testing is used by Elsevier to test its websites. A/B testing is an experience with two variants, A and B, to identify changes to web pages that increase or maximize an outcome of interest (e.g. click-through rate for button or banner advertisement). Since it's been useful for http://www.elsevier.com/ and http://store.elsevier.com/, Elsevier's store team would like to have it as well in the mobile application. Thus, to decide which 3rd party that's going to be integrated with the app, the research must be done.

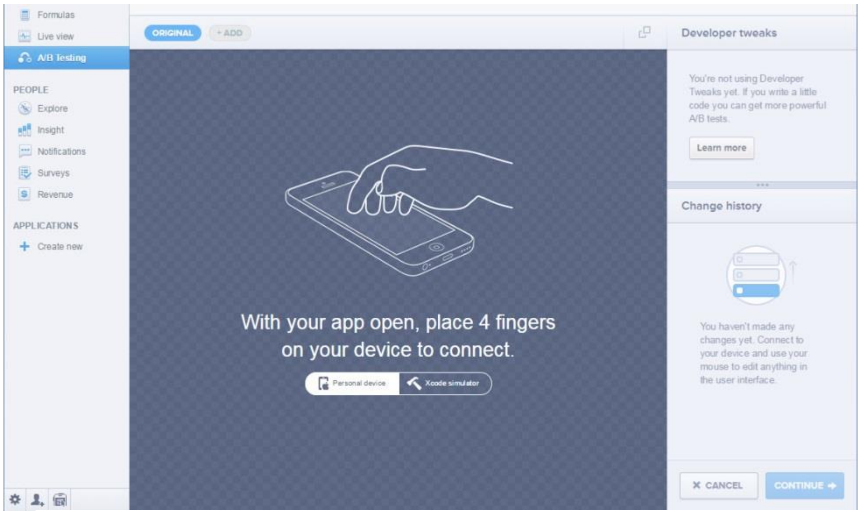
   a. Field

      This is done by asking company's mentor which 3rd party that Elsevier would like to integrate with. His answer is Optimizely, because they have been Optimizely's customers for such long time, and they trust Optimizely.

      But, as written in subchapter above – Dependency Manager for Android, Elsevier has some misunderstanding with Optimizely regarding the payment, thus integrating Optimizely with the app is not possible.

   b. Library

      While exploring Mixpanel for push notification, it turns out that Mixpanel also has A/B testing option for mobile application. After further research, unfortunately A/B testing using Mixpanel only can be done for iOS.



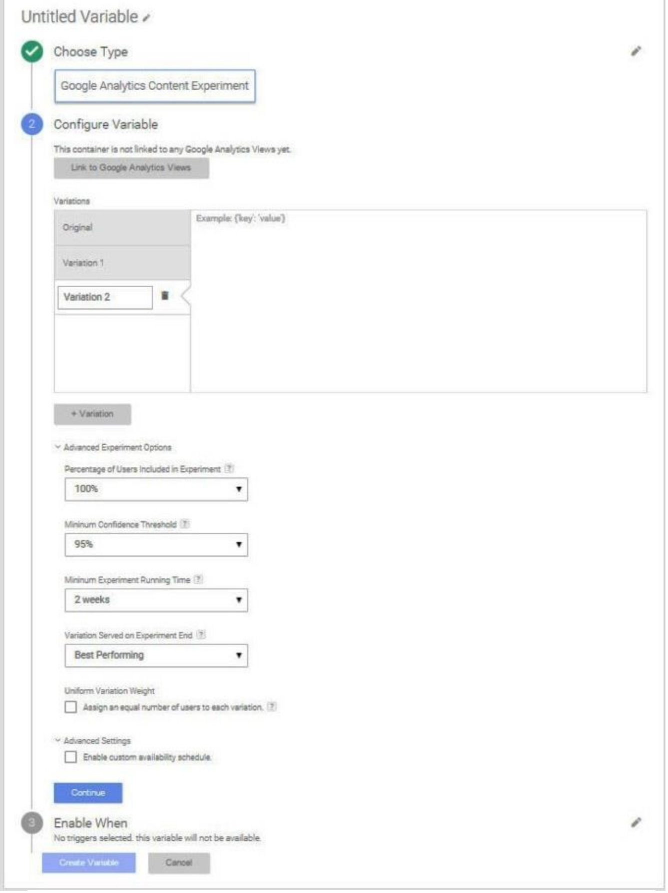
With your app open, place 4 fingers on your device to connect.

You're not using Developer Tweaks yet. If you write a little code you can get more powerful A/B tests.

You haven't made any changes yet. Connect to your device and use your mouse to edit anything in the user interface.

Figure 11: Mixpanel A/B testing only for xCode and Apple device

Other than Mixpanel, Google Tag Manager also can be useful for A/B testing. Even though it's not written as testing platform, but it provides a feature to experiment the content variable of the app on the user and send the data to google analytics to be analysed.



Untitled Variable

Choose Type

Google Analytics Content Experiment

2 Configure Variable

This container is not linked to any Google Analytics Views yet.

Variations

Original

Variation 1

Variation 2

Example: {'key': 'value'}

Advanced Experiment Options

Percentage of Users Included in Experiment: 100%

Mininum Confidence Threshold: 95%

Mininum Experiment Running Time: 2 weeks

Variation Served on Experiment End: Best Performing

Uniform Variation Weight

□ Assign an equal number of users to each variation.

Advanced Settings

□ Enable custom availability schedule.

3 Enable When

No triggers selected. this variable will not be available.

Figure 12: Web interface to define experiment

As seen in figure 12, the team could define variations of variables in the form of JSON String and define the percentage of targeted users as the team wants as well as the period that the experiment would be going. Therefore, doing experiment of some variants on the user is also like using A/B testing, just with different name.

Based on the data above, Google Tag Manager could be used for A/B testing when the app is launched.

4. Crash and Bugs Monitoring and Reporting

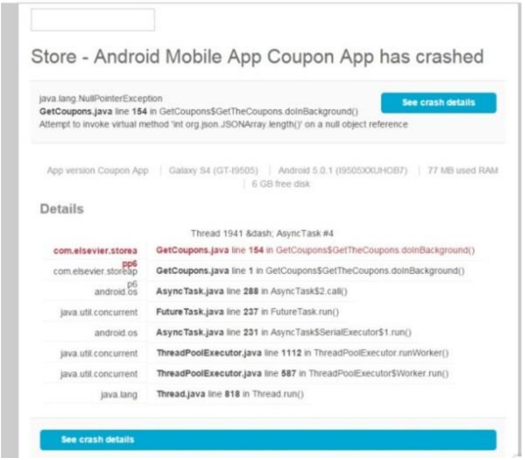
Who knows that an app can be broken all of sudden. Thus, in order to be able to keep monitoring how the app works, Elsevier's store mobile application needs crashes or bugs monitoring and reporting. But there are many 3rd party services that offer the tools to help building high-performance and stable mobile application. So the research is done to know which one is very useful for Elsevier's store mobile app.

a. Library

This is done by browsing all over internet regarding the crash monitoring and reporting. It gives two results: Crashlytics and New Relic.

b. Lab(oratory)

To know which tool is better, the laboratory method is done by firstly integrating the app with New Relic. The integration is done by following steps written in New Relic website. The testing is done by giving a temporary error, so the team could see if it's working properly. And the result is it works properly.



Store - Android Mobile App Coupon App has crashed

```
java.lang.NullPointerException
GetCoupons.java line 154 in GetCoupons$GetTheCoupons.doInBackground()
Attempt to invoke virtual method 'int org.json.JSONArray.length()' on a null object reference
```

App version Coupon App | Galaxy S4 (GT-I9505) | Android 5.0.1 (I9505XXUHOB7) | 77 MB used RAM | 6 GB free disk

Details

```
                         Thread 1941 &dash; AsyncTask #4
com.elsevier.storeapp6   GetCoupons.java line 154 in GetCoupons$GetTheCoupons.doInBackground()
com.elsevier.storeapp6   GetCoupons.java line 1 in GetCoupons$GetTheCoupons.doInBackground()
            android.os   AsyncTask.java line 288 in AsyncTask$2.call()
  java.util.concurrent   FutureTask.java line 237 in FutureTask.run()
            android.os   AsyncTask.java line 231 in AsyncTask$SerialExecutor$1.run()
  java.util.concurrent   ThreadPoolExecutor.java line 1112 in ThreadPoolExecutor.runWorker()
  java.util.concurrent   ThreadPoolExecutor.java line 587 in ThreadPoolExecutor$Worker.run()
             java.lang   Thread.java line 818 in Thread.run()
```

Figure 13: Example of New Relic Report

Secondly, the app should be integrated with Crashlytics, but unfortunately it hasn't done because it turns out that Crashlytics is still in Beta Version, which means the Crashlytics is still in testing version and can't be used by the public.

Based on the research, because New Relic is successfully integrated, the decision then is made to use New Relic for Crash Monitoring and Reporting tool.

4.1.9. Minimum Viable Product (MVP)

Minimum Viable Product is the term used in Lean Startup methodology. As the team approaching the project using that method, defining the MVP for launch premier has to be done. But to know what is best for the customers, the team also do the research.

4.1.9.1. *Brainswarming*

Before changing the app, the team tried Brainswarming method to produce more ideas about the app such as: what kind of app that Elsevier's store team would like to have and its features. Brainswarming is a changing method of brainstorming that was never worked for most people. Brainswarming is done by defining a goal and sub-goals that need to achieve, and listing all resources of the company. When the right resources area found, the team comes up with the solution. To see the details of the goals that Elsevier's store team would like to achieve, please see Appendix C. And based on this brainswarming method too, the type of the Elsevier's store mobile app is changed to 'Coupon App'.

4.1.9.2. *Hook Model*

Other than Brainswarming, MVP is also decided when the team would like to build a habit-forming product by following Hook Model, a four phase framework to forms habits, created by Nir Eyal. The framework includes four steps: trigger (internal and external), action, variable rewards, and investment.

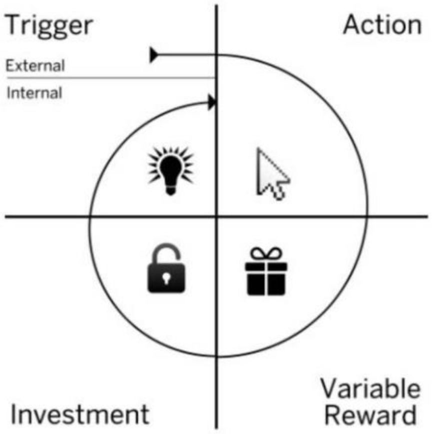


Figure 14: Hook model

Internal and external triggers are the part which activate user to do something. Following the triggers, the action must be done to form a better habit – otherwise the trigger is useless, in this phase the user also has to have motivation to do the action. In doing an action, the user is usually anticipating the reward, therefore variable rewards must be provided for the user to solve user's problem, reinforcing their motivation of the action. There are three variable rewards written in this framework: reward for search of social connections with others, reward for search of information, and reward for search of self-competence and completion. Before the automatic behaviours of the user are activated, the user must invest something into the product, therefore the investment phase is critical for the framework.

In order to build a habit-forming product, therefore Elsevier's store team also think about the features of the app that can trigger the user to do something with the app to solve user's problems, give rewards to the user so the user wanting more of it, and ask the user implicitly to invest something to the app. Some features created based on this framework:

1. Triggers
    a. Push notification and e-mail notification, which will trigger the user to come back to the app.
2. Action
    a. Sharing a coupon to social media, so the user get special coupon just for that user.
    b. Send the coupon to user's own e-mail so the user can use it directly when buying books in the browser.
3. Variable Rewards
    a. Looking for a specific coupon code if it's available or still valid, so the user feels like having a reward that the user can find coupon that is still valid.
    b. Rewards received by sharing a coupon code to social media, so the user feels the completion of something.
4. Investment
    a. Put user's e-mail to send the coupon data.
    b. Enabling the push notification and e-mail notification, so the user can receive any information related to coupons and Elsevier.
    c. Make subject(s) as favourite, so the user can browse or get notification about specific coupons.

Not all the features above are decided to become the MVP of the app. The features are not implemented for MVP:

1. Sharing a coupon to social media; this feature has been tried and tested several times in the app, but it still doesn't work properly and it will take more time to continue implementing it, therefore it's decided to not be implemented at all for now.
2. Rewards received by sharing a coupon code to social media; this feature is not going to be part of MVP because the sharing coupon features is not implemented which makes it not really useful right now.

Based on the research above, the final features for MVP are then decided. To see the complete list of MVP, please take a look at Appendix E.

4.2. Design

4.2.1. Action State Diagram

First, the design phase starts by designing the action state diagram of the application, from the initial plan, to give the ideas how Elsevier's store mobile application would work. Action state diagram defines what actions that the user can do in the application, such as: from homepage, the user can click browse so the user browse books by specific options, after that the user have three options to browse books by subjects, browse books by imprints, or browse books by authors. And for each browse option, it has further actions that the user can do to get to the intent point.

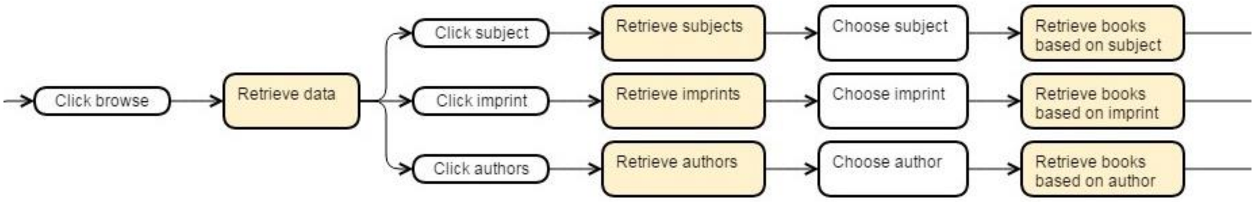


Figure 15: Example of specific action state diagram (Browse option) for Elsevier's store mobile app (Initial Plan)

The action state diagram is designed in https://www.gliffy.com/, a free-tool website to create flowcharts, wireframes, diagrams, etc. The final action state diagram can be found in Appendix E.

4.2.2. Wireframe

To give more visualization of the app, wireframes are design on paper, so people understand more about the action diagram designed before. Wireframe is a visual guide that represents the skeletal framework of the app, such as in Figure 16 below.

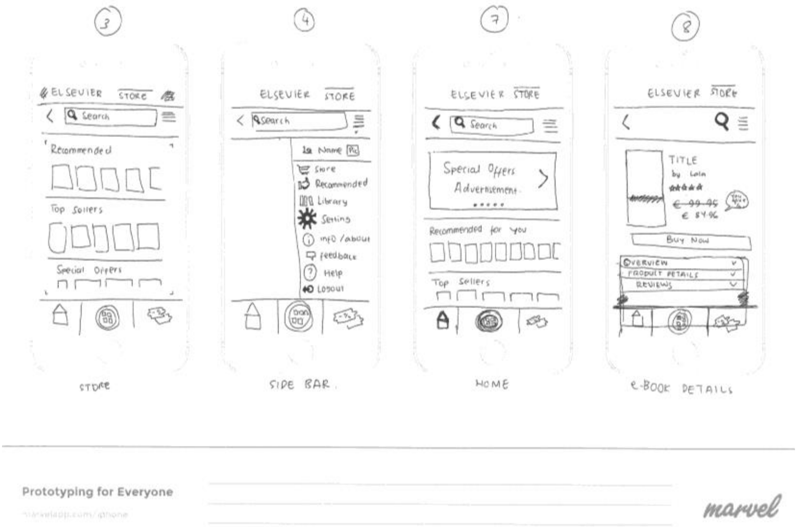


Figure 16: Example of some wireframe sketched on paper

The design is also for initial plan where the app should be built for iOS users and the user could download e-book from the app. Wireframe is also prepared because it's easier to do than designing the UI or prototype. The template used is gotten from https://marvelapp.com/, a free mobile and web prototyping for everyone. Please see Appendix F to see wireframe designed for Elsevier's store mobile app.

4.2.3. First Prototype and Screen Flow

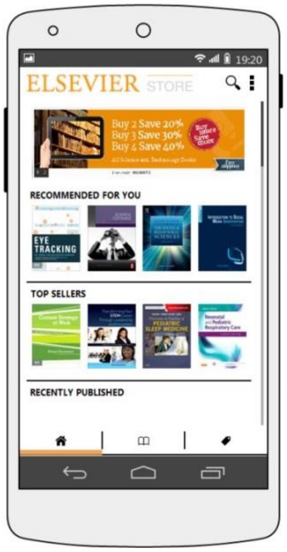


Figure 17: First homepage design

The design of first prototype and screen flow is done after the type of the app is changed to coupon app. The design is included the list of books, list of coupons, login, and registration page, browse by subject and imprint, etc.

This is done because at first thought, even though it would be coupon app, the books details would still be there so the user still can browse books available in Elsevier's store.

The tool used to design the first prototype is WireframeSketcher, a desktop application and a plug-in for any Eclipse IDE. The tool is chosen because it's easy to use, provide Android template, and provide nice designs which is easier for the team to design the prototype. Even though it's not free, it provides a month free trial period which is useful to design screen flows above. To see the full of screen flow, please see Appendix G.

4.2.4. Second Prototype and Screen Flow

After some discussion with the company mentor regarding the first screen flow, the trainee got feedback that it turns out the app that team wanted is the coupon app where it really focuses on the coupon available for Elsevier's store. So in this case, there will not be a list of books, it's just going to be something that relevant to coupons. Therefore, the design is changed to be more focus to coupons and look simpler.

For this second prototype, the screen flows that have been designed are the home screen, browse coupons by books subjects, and search or look up for a specific coupon code. Please see Appendix G to see more details of the screen flows.

The tool used to designs the second screen flows is draw.io, a free online diagram tool for making flow charts, process diagram, etc. It's chosen because WireframeSketcher is not free and its trial period has been expired. Thus, for this second screen flow, online tool is researched and discovered. The features are similar with WireframeSketcher too, such as: easy to use, provide Android templates and nice designs.

4.2.5. Third Prototype and Screen Flow

This third prototype and screen flow is supposed to be continuing the screen flows that have been designed in second part. The third screen flow designed includes all coupons and account settings of the user. Even though all coupons flow has been implemented, but it will be changed due to a feature of displaying the new coupons available for the user after the app is launched. Account settings is designed only to give a visualization about how each settings will be put in order, because it's not going to be complicated and only showing the options to enable/disable notifications and edit user's e-mail. Please see Appendix G to see all screens flow. The tool used to design the third screen flow is the same as second screen flow – draw.io.

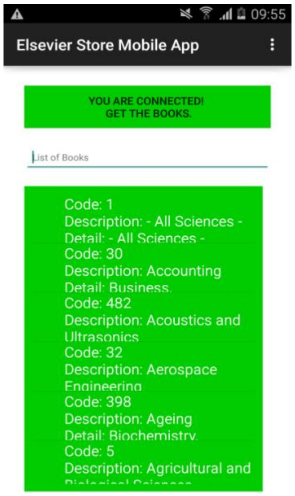
4.3. Implementation

The implementation of the application starts in the middle of the project. It's done parallel with designing the first screen flow and by releasing new version of the app weekly. Defining the whole features (or MVP) that have to be implemented is done by the trainee together with company's mentor, but defining the features for each version is done by trainee herself so the trainee also learn to organize the app better.

The IDE used to implement the app is Eclipse, an open source IDE mostly provided for Java language. It's chosen because the trainee has a little bit experience using Eclipse, and learning how to use new IDE takes more time which doesn't really relevant to this project. Eclipse is also easy to use and was an official IDE for Android when the trainee started the internship – now it's changed to Android Studio. Two Eclipse's versions are used for this project, because there is limitation while using 3rd party service. Eclipse Luna is used for version 1, 2, 3, and Eclipse Kepler is used for version 4 until the app is done. Further details for each version are explained below.

4.3.1. Version 1

This version starts by trying to get data from a web server and displaying it in the application. The data is including the list of books available in Elsevier. Getting the list of books from the server works properly, therefore the implementation can continue to the next step.



09:55

Elsevier Store Mobile App

YOU ARE CONNECTED!
GET THE BOOKS.

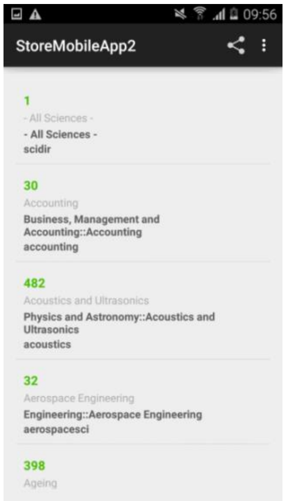
List of Books

Code: 1
Description: - All Sciences -
Detail: - All Sciences -
Code: 30
Description: Accounting
Detail: Business.
Code: 482
Description: Acoustics and
Ultrasonics
Code: 32
Description: Aerospace
Engineering
Code: 398
Description: Ageing
Detail: Biochemistry.
Code: 5
Description: Agricultural and
Biological Sciences

Figure 18: Version 1 application screen shot

4.3.2. Version 2

As seen in Figure 17, the view of the list doesn't look good, so in this version fixing the view of the list is done which resulting in better list view (see Figure 19). In this version, the user is also able to click the list item to see the book's details and share the book's details to Twitter by using a specific feature provided by Android itself. But the problem with this feature is that the owner of the app can't have the analytics if the user is really sharing it or not. Thus, in this version, there is also sharing to twitter feature by connecting the app to Twitter API itself. But, since connecting to Twitter API is not as easy as it seems, the implementation of this feature is not 100% done.



1
- All Sciences -
- All Sciences -
scidir

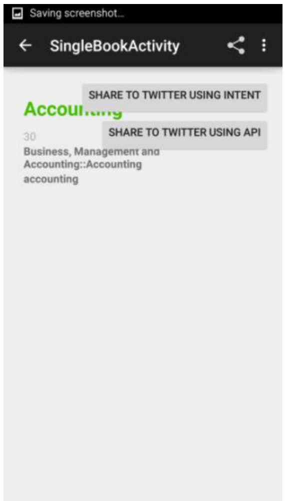
30
Accounting
Business, Management and Accounting::Accounting
accounting

482
Acoustics and Ultrasonics
Physics and Astronomy::Acoustics and Ultrasonics
acoustics

32
Aerospace Engineering
Engineering::Aerospace Engineering
aerospacesci

398
Ageing

Figure 20: Homescreen version with better listview



Accounting
30
Business, Management and Accounting::Accounting
accounting

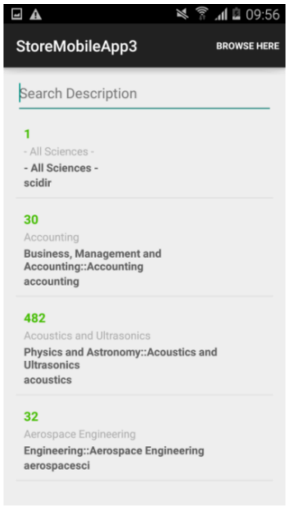
Figure 19: Books details with the option share to twitter if the share icon is clicked

4.3.3. Version 3

In this version, new functionalities are added:

- Free search of subject, where the user just type anything in a textbox and the list of books will be filtered based on what the user types.
- Browse the book by its subjects, where the user is able to choose a subject and then find list of books based on the subject that the user chooses.

Within this version, the user is still able to find the functionalities in the previous versions. Free search and browse by subject are done because it’s crucial for the MVP of the app; therefore it’s better to do it now than later.



Search Description

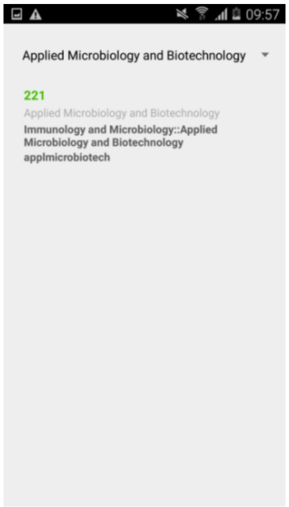
1
- All Sciences -
- All Sciences -
scidir

30
Accounting
Business, Management and Accounting::Accounting
accounting

482
Acoustics and Ultrasonics
Physics and Astronomy::Acoustics and Ultrasonics
acoustics

32
Aerospace Engineering
Engineering::Aerospace Engineering
aerospacesci

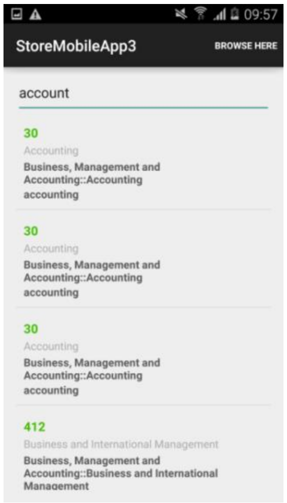
Figure 23: Homepage version 3 with a textbox to free search of the list



Applied Microbiology and Biotechnology

221
Applied Microbiology and Biotechnology
Immunology and Microbiology::Applied Microbiology and Biotechnology
applmicrobiotech

Figure 22: Browse by subject page



account

30
Accounting
Business, Management and Accounting::Accounting
accounting

30
Accounting
Business, Management and Accounting::Accounting
accounting

30
Accounting
Business, Management and Accounting::Accounting
accounting

412
Business and International Management
Business, Management and Accounting::Business and International Management

Figure 21: Free search example

4.3.4. Version 4

In version 4, there is not much additional change of the app. The integration of New Relic with the app is done in this version to know exactly if New Relic works properly.

In this version, changing IDE is also done. As mentioned before, Eclipse Luna is used for version 1, 2, 3, but it changes since this version because of the integration with New Relic. It turns out New Relic doesn't have its library for new version of Eclipse (Luna), therefore the implementation of Elsevier's app must be done in Eclipse's versions before Luna. But, to not switch the version too far, the earlier version before Luna is used: Eclipse Kepler.

4.3.5. Version 5

Starting this version, the implementation of the application is changed. It turns out that getting books data from the server is only a trial to know exactly if it's working properly, because actually the app will just work like that: getting data from the server and displaying them in the app.

Within this version, there is no book list and its details again. It's totally changed to the list of coupons. So, instead of continuing the version 4, the app is newly created. One thing that is not changed is the integration with New Relic, so the team still could see if there is crash in the app. The implementation is also starting to look like the design in the prototype and screen flow.

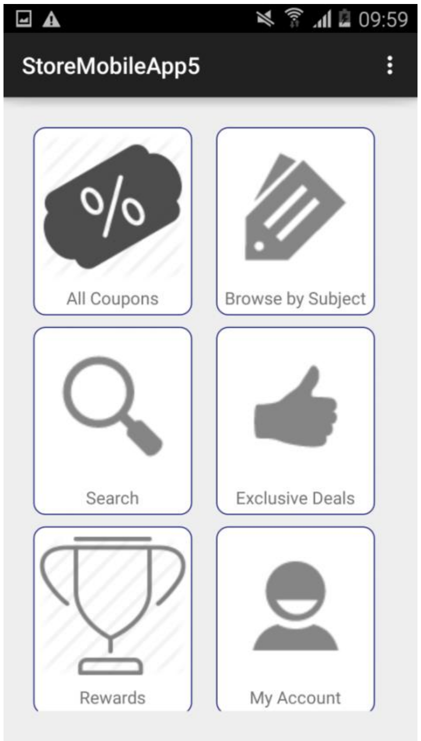
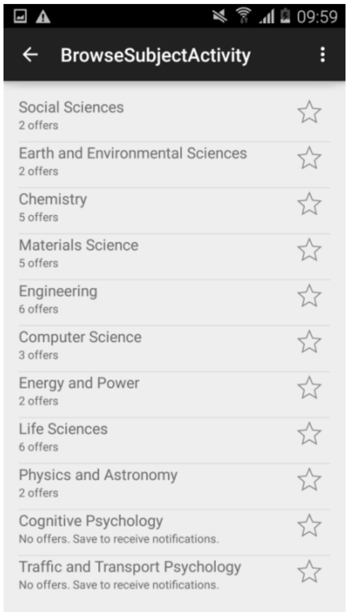


Figure 24: Homepage version 5

The features available in this version:

- Find list of all coupons
- Browse the coupons by its subjects



09:59

BrowseSubjectActivity

Social Sciences
2 offers

Earth and Environmental Sciences
2 offers

Chemistry
5 offers

Materials Science
5 offers

Engineering
6 offers

Computer Science
3 offers

Energy and Power
2 offers

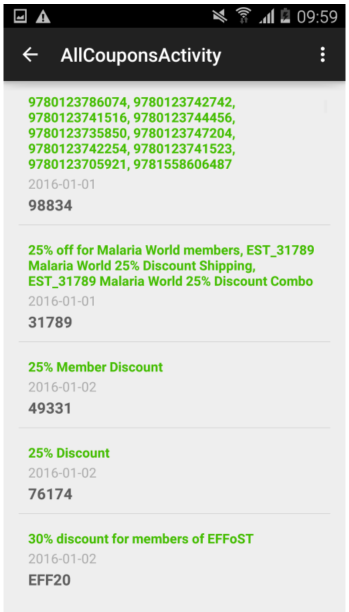
Life Sciences
6 offers

Physics and Astronomy
2 offers

Cognitive Psychology
No offers. Save to receive notifications.

Traffic and Transport Psychology
No offers. Save to receive notifications.

Figure 26: Browse by Subject page version 5



09:59

AllCouponsActivity

9780123786074, 9780123742742, 9780123741516, 9780123744456, 9780123735850, 9780123747204, 9780123742254, 9780123741523, 9780123705921, 9781558606487
2016-01-01
98834

25% off for Malaria World members, EST_31789 Malaria World 25% Discount Shipping, EST_31789 Malaria World 25% Discount Combo
2016-01-01
31789

25% Member Discount
2016-01-02
49331

25% Discount
2016-01-02
76174

30% discount for members of EFFoST
2016-01-02
EFF20

Figure 25: All coupons page version 5

4.3.6. Version 6

There are a lot of additional functionalities in this version. And not like other versions, the implementation for version 6 takes about 2 weeks – where the others released weekly, because there are two demos that trainee has to do in these two weeks, therefore there must be solid mobile application that can be shown to people that this is a great product. One demo is done for the VPs and director of e-Business department that happens in the first week – of this version, and another one happens in the second week is for all employees of e-Business department and some people in US and UK quarters.

The functionalities that can be found in this version:

- New homepage icons design
- New view of list of all coupons
- Browse coupons by books subject
- List of coupons for a specific subject
- Check a specific coupon code if it's available or valid
- Coupon's details
- E-mail a specific coupon code to user's e-mail, so the user can use the code directly in the web store
- Favourite a specific subject
- See only favourites subject(s)

- Receive push notification

The integration with 3rd party services for specific functionalities also done in this version:

- Integration with New Relic, for crash reports to the team
- Integration with Mixpanel, to send the push notification to the user
- Integration with Mandrill, to send the e-mail containing the coupon code to the user

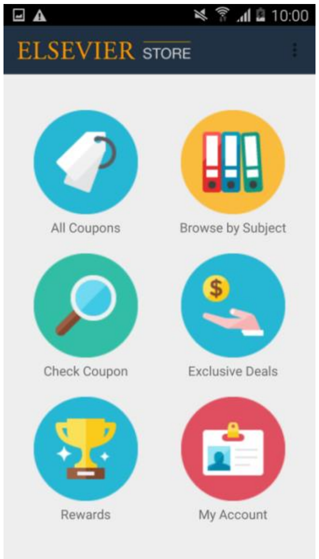
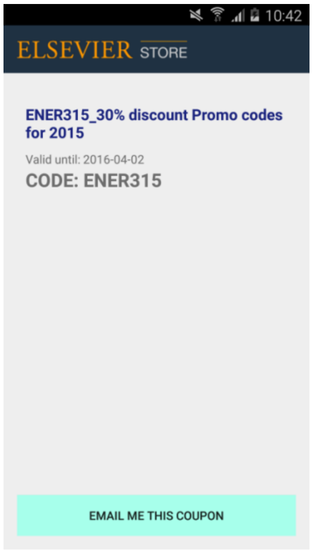


Figure 27: Homepage version 6

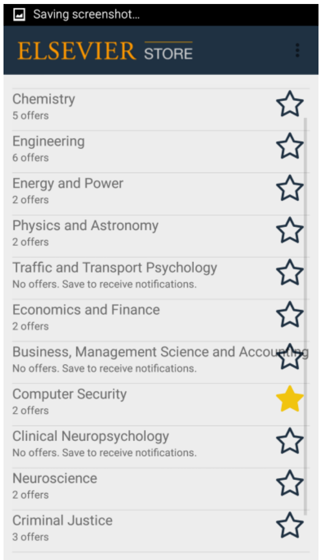


ENER315_30% discount Promo codes for 2015

Valid until: 2016-04-02

CODE: ENER315

Figure 28: Coupon details page



Chemistry
5 offers

Engineering
6 offers

Energy and Power
2 offers

Physics and Astronomy
2 offers

Traffic and Transport Psychology
No offers. Save to receive notifications.

Economics and Finance
2 offers

Business, Management Science and Accounting
No offers. Save to receive notifications.

Computer Security
2 offers

Clinical Neuropsychology
No offers. Save to receive notifications.

Neuroscience
2 offers

Criminal Justice
3 offers

Figure 29: Browse by subject with a favorite

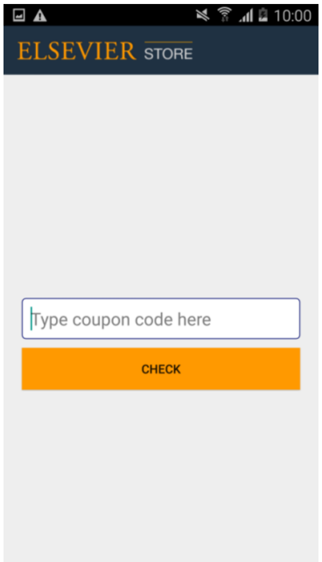
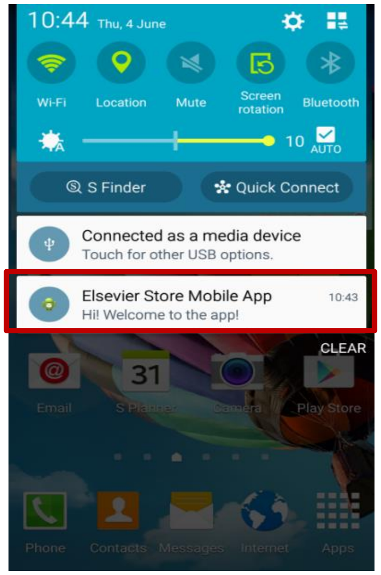


Figure 31: Check a specific coupon code



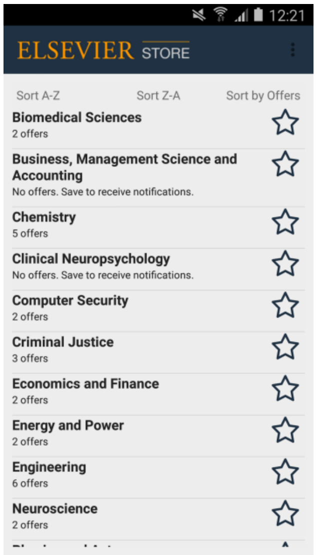
Connected as a media device
Touch for other USB options.

Elsevier Store Mobile App 10:43
Hi! Welcome to the app!

Figure 30: Push notification received

4.3.7. Version 7

Before this version started, defining final MVP is done. Therefore, starting this version, the implementation is done adjusting the MVP (for list of MVP can be found in Appendix E). There are not many changes in this version; the changes are only done for the look, because this version wanted to be sent to the agency for marketing purpose. But there are additional functionalities that are implemented: sorting the subject ascending and descending, and sorting the subject by its numbers of offers.



Biomedical Sciences
2 offers

Business, Management Science and Accounting
No offers. Save to receive notifications.

Chemistry
5 offers

Clinical Neuropsychology
No offers. Save to receive notifications.

Computer Security
2 offers

Criminal Justice
3 offers

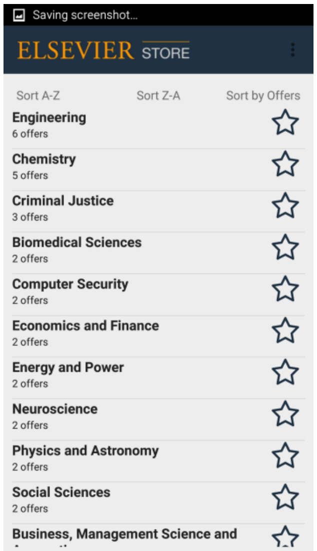
Economics and Finance
2 offers

Energy and Power
2 offers

Engineering
6 offers

Neuroscience
2 offers

Figure 32: Sorting ascending



Engineering
6 offers

Chemistry
5 offers

Criminal Justice
3 offers

Biomedical Sciences
2 offers

Computer Security
2 offers

Economics and Finance
2 offers

Energy and Power
2 offers

Neuroscience
2 offers

Physics and Astronomy
2 offers

Social Sciences
2 offers

Business, Management Science and

Figure 33: Sorting by number of offers

4.3.8. Version 8

In this version, the development continues with some features that are easiest and most possible implemented, such as:

- Free search in all coupons page to filter the coupon.
- Having an option to choose if the user want to get notification when it's the first time user favourites a subject or coupon.
- The rewards part is changed to 'My Favourite', thus the favourite page is re-implemented in that flow instead of in browse by subject flow.

For the other features, including exclusive deals and account settings flow, they will be implemented when the list of exclusive coupons is ready and in the next version after this. The integration with the other 3rd party services, including google tag manager and google analytics, will be done once all features already implemented. To see how the application looks like until this part, please take a look at Appendix H.

4.4. Testing

Since the application is not finished yet, testing part is also not completed. What has been done so far is usability testing. Usability testing is done by asking some people to use the app after each version is published, and by having the demos to show to people, including (as written above) VPs, director, and all employees of e-Business department, and some internal employees of Elsevier that residence in US and UK quarters; feedback received is used for improving the application.

However, to have good quality of the app in the end, some testing method will be done by the trainee following Android Testing Framework that is stated in Android Development website. Android Testing Framework includes: create test case, user interface testing, integration (unit) testing, and functional validation testing.

1. User Interface Testing

   This testing is done by testing the components of the user interface (such as: buttons, textbox, checkbox, etc.) to know if it works as intended.
2. Integration (unit) Testing

   This testing is done by testing the activities happened in the app and their interactions with other components to check if such activity has correct intention to do.
3. Functional Validation Testing

   This testing is done after testing the integration (unit) of the app, to verify if such activity works properly. In this test, build an activity monitor is done to display the data that has been tested.

4.5. Submission and Publish

The application must be published to Google Store to test if this project is successful or not, therefore the submission and launch is done.

Before releasing the production application, apparently Google Store provides a way for its customers to publish the Beta Testing application file, so the application can be tested by testers defined by Elsevier. Thus, the team is currently publishing Beta Testing version in Google Store to get feedback by public. The version used for Beta Testing is version 7 (see implementation above for more details).

Once the whole application is done, the Production version is published.

5. Conclusions and Recommendation

Elsevier B.V. is a world-leading publishing company that publishes medical and scientific literature. E-Business department of Elsevier, who generates revenue from its web store (a B2C value proposition making several million $ yearly), is considering to explore mobile applications as a way to drive acquisition, increase conversion, and improve retention of the web store. The department knows that mobile application market has been steadily growing (mobile app market economy was worth $53 billion in 2012 and it's forecasted to grow to $143 billion in 2016), therefore e-Business department would like to have a mobile app for its web store.

To have a qualified mobile application that can reach the goal, it requires further discussion about business requirements, research, identifying the necessary technology and the technical requirements, and the implementation of the application. Lean startup methodology also has been carried to minimize the time in developing the product so Elsevier's store team doesn't invest time designing features or services that consumers do not want.

The methodology is used to define the minimum viable product of the app, and is more useful when the app is launched so the team could gather data based on the analytics and learn which features are useful for the customers, which are not, and if the app is a success in reaching the goal.

Research is done by the trainee to define the necessary requirements, which results in having a coupon app for Android users, where the users are able to look up the list of all coupons available from Elsevier to be used in buying Elsevier's books, e-Books, and journal. The app is also integrated with 3rd party services to minimize the time in developing all aspects and to easily maintain the app without going deeply to technical part.

To give more visualization of how the app looks like and how the flow of the app is going, the design is done. After having some ideas about the design of the app and requirements gathered in research phase, the implementation started. In the implementation phase, the app is developed and published weekly which result in one version in one week. The app is currently in Beta Testing version and is already submitted to Google to get feedback of the content.

Unfortunately, the production app isn't finished yet due to the deadline of the report is earlier than the app itself. However, it certainly will be finished by the end of June 2015 and published soon after that.

The recommendation for further promotion and improvisation will be:

- ✓ Buying e-book from the app directly and downloading e-book to the device so the user is able to read the e-book directly, should be implemented.
- ✓ Several features that are not able to be implemented before such as sharing the coupons to social media also should be implemented, so the user can get rewards which, hopefully, the app will improve retention.
- ✓ The app should also be developed for iOS user, because as described in the content above, that most of Elsevier's store customers use iOS devices.

Evaluation

Doing an internship in a big company such as Elsevier is a pleasant experience for me. I learned that you'll be working together with a lot of people, because the organization is so big that you can't do everything just by yourself or your team. Beside, working in business environment where most of people are not interested enough on technical part taught me how to communicate to non-technical people about a product.

Being an intern as a mobile app developer is quite thrilling for me, it's not easy neither is difficult, but it's really complicated. I had a really small amount of knowledge about developing mobile application before I started the internship, yet I chose the project. The process has become a challenge, but it's quite entertaining, especially when we (me and even the team) had no idea how the app would be and which target platform the app would be in.

When knowing that I had to build an app for iOS users, I got really excited because it's really new for me – I never had any experience to develop any applications for Apple OS. But unfortunately the plan is changed so I had to develop the app for Android users. It's still exciting too because Android is new for me, even though I use it for several years now, I just never thought that I'd build anything of it.

Also, I was a little bit disappointed when knowing the app that the user could download e-book couldn't be pursued because I thought it would be really exciting to build that kind an app. But now, I feel thankful that the plan is changed, because developing this coupon app is already complicated enough, even though it looks simple.

Anyway, this project went quite smoothly, even though it involves a lot of people, research, and some problems especially in developing some features. And I learned that I have to ask for help when I'm stuck with something, and thanks to my mentor who always telling me that it's really okay to ask for help.

Overall, I feel really pleased that I'm given this internship opportunity and I learned a lot from it. I'm sure that I will remember the good things of it and leverage them in the near future.

References

1. Elsevier. (n.d.). Retrieved from http://en.wikipedia.org/wiki/Elsevier
2. Sacha Zinevych. (August 26, 2014). The Overview of Mobile App Market: Why You Should Enter Now. Retrieved from http://publ.com/blog/2014/08/26/the-overview-of-mobile-apps-market-why-you-should-enter-now/
3. Elsevier. (n.d.). At a Glance. Retrieved from http://www.elsevier.com/about/at-a-glance
4. Vikash. (December 4, 2014). How PhoneGap Works. Retrieved from http://dotnet-guru.com/phonegap-works/
5. Marlo Korf and Eugene Oksman. (n.d.). Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options. Retrieved from https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options
6. Drew Olanoff. (September 11, 2012). Mark Zuckenberg: Our Biggest Mistake Was Betting Too Much on HTML5. Retrieved from http://techcrunch.com/2012/09/11/mark-zuckerberg-our-biggest-mistake-with-mobile-was-betting-too-much-on-html5/
7. Devindra Hardawar. (September 11, 2012). Facebook's Zuckenberg: 'The Biggest mistake we've made as a company is betting on HTML5 over native.'. Retrieved from http://venturebeat.com/2012/09/11/facebooks-zuckerberg-the-biggest-mistake-weve-made-as-a-company-is-betting-on-html5-over-native/
8. Apps. (n.d.). Retrieved from https://play.google.com/store/search?q=elsevier
9. The World's 60 Largest Book Publishers, 2013. (July 19, 2013). Retrieved from http://www.publishersweekly.com/pw/by-topic/industry-news/financial-reporting/article/58211-the-global-60-the-world-s-largest-book-publishers-2013.html
10. Jane. (February 6, 2011). Best Mobile Reading Apps. Retrieved from http://dearauthor.com/ebooks/dear-jane-ebooks/best-mobile-reading-apps/
11. Apps For Book Lovers. (n.d.). Retrieved from http://appadvice.com/applists/show/apps-for-book-lovers
12. Avid Larizadeh. (July 29, 2013). Eight Tips For A Successful App. Retrieved from http://www.forbes.com/sites/avidlarizadeh/2013/07/19/eight-tips-for-a-successful-app/
13. Rahul Varshneya. (December 12, 2013). The Secrets of The World's Most Successful Mobile Apps. Retrieved from http://www.entrepreneur.com/article/230160
14. Jose Vasquez. (December 31, 2013). 10 Qualities of Successful Mobile Apps in Startup Tech Companies. Retrieved from http://www.huffingtonpost.com/jose-vasquez/10-qualities-of-successfu_b_4391550.html
15. Victor Balasa. (n.d.). Guide To Starting A Successful App Business: Tips and Resources. Retrieved from http://www.hongkiat.com/blog/app-business-tips-resources/
16. Tom Lowery. (December 20, 2013). 12 Tips for Creating a Successful App. Retrieved from http://www.cmswire.com/cms/mobile-enterprise/12-tips-for-creating-a-successful-app-023589.php
17. Jeremy Olsen. (November 7, 2012). How To Succeed With Your Mobile App. Retrieved from http://www.smashingmagazine.com/2012/11/07/succeed-with-your-app/

18. Mark Hoelzel and Emily Adler. (December 5, 2013). Inside Amazon's Kindle Tablet Ecosystem, Arguably The Most Underrated Mobile Platform. Retrieved from http://www.businessinsider.com/a-look-at-amazons-kindle-ecosystem-2013-12?IR=T
19. Jason Lengstorf. (n.d.). JSON: What It Is, How It Works, & How to Use It. Retrieved from http://www.copterlabs.com/blog/json-what-it-is-how-it-works-how-to-use-it/
20. Francisco Nieves. (November 5, 2013). How to Create a Database Mobile App with SQLite and Xamarin Studio. Retrieved from http://www.itexico.com/blog/bid/97885/How-to-Create-a-Database-Mobile-App-with-SQLite-and-Xamarin-Studio
21. Ashley Menhennett. (June 16, 2014). Building HTML5 Mobile Apps with Local Storage and Topcoat. Retrieved from http://www.binpress.com/tutorial/local-storage/106
22. What is Cocoapods. (n.d.). Retrieved from https://cocoapods.org/
23. Mixpanel. (n.d.). Android API Integration. Retrieved from https://mixpanel.com/help/reference/android
24. Mixpanel. (n.d.). Installing the Android library for Eclipse. Retrieved from https://mixpanel.com/help/reference/android-eclipse/
25. A/B testing. (June 4, 2015). Retrieved from http://en.wikipedia.org/wiki/A/B_testing
26. Website wireframe. (April 13, 2015). Retrieved from http://en.wikipedia.org/wiki/Website_wireframe
27. Beta version. (April 18, 2015). Retrieved from http://en.wiktionary.org/wiki/beta_version
28. Crashlytics. (n.d.). Retrieved from https://get.fabric.io/crashlytics
29. Newrelic. (n.d.). Retrieved from http://newrelic.com/
30. Pushbots. (n.d.). Retrieved from https://pushbots.com/
31. Pushbots [PushBots]. (May 12, 2013). Send GCM Android Push Notifications in less than 7 mins with PushBots [Tutorial]. Retrieved from https://www.youtube.com/watch?v=faop6vBBe3E
32. Nathan Esquenazi. (May 31, 2014). Re: Is there an equivalent of cocoapod for Android for dependency management? [Github issues comment]. Retrieved from https://github.com/codepath/android_guides/issues/33
33. Matthew. (August 2, 2013). Re: Something similar to Cocoa Pods and Bundler for Android (fro dependencies)? [Stackoverflow comment]. Retrieved from http://stackoverflow.com/questions/13771375/something-similar-to-cocoa-pods-and-bundler-for-android-for-dependencies
34. Ddewaele. (December 14, 2014). Integrating maven in your Android Eclipse projects. Retrieved from http://blog.doityourselfandroid.com/2010/12/14/integrating-maven-android-eclipseprojects/
35. 5. Maven Dependency Management. (n.d.). Retrieved from http://docs.spring.io/spring-android/docs/current/reference/html/spring-android-maven.html
36. Ben Ellingston. (April 28, 2014). Gradle: The New Android Build System. Retrieved from http://gradle.org/gradle-the-new-android-build-system/
37. Build System Overview. (n.d.). Retrieved from https://developer.android.com/sdk/installing/studio-build.html

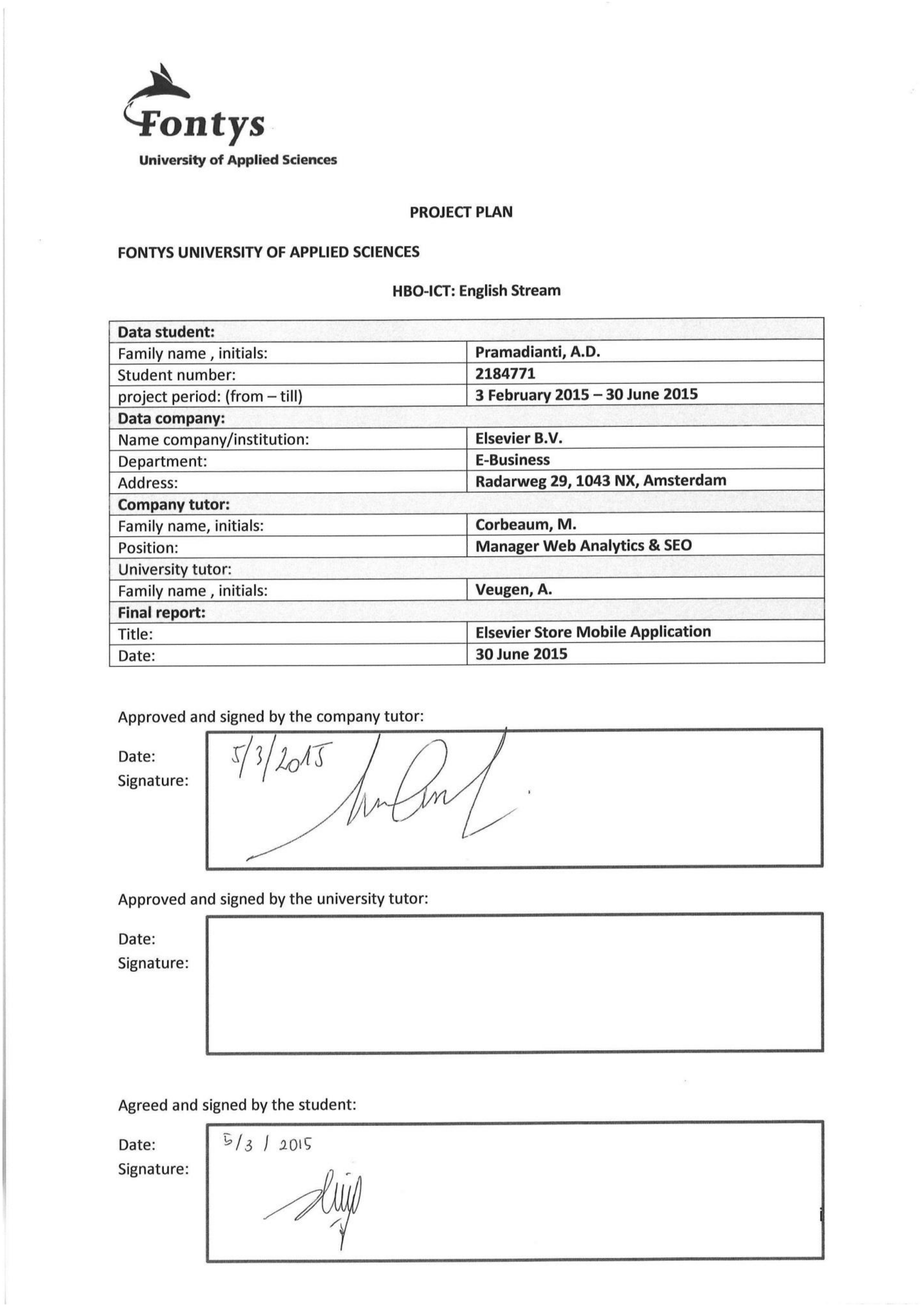
38. Johnsmart. (November 1, 2010). Android Development with Maven – a primer. Retrieved from https://weblogs.java.net/blog/johnsmart/archive/2010/11/01/android-development-maven-primer
39. Optimizely. (n.d.). Installing the Android SDK. Retrieved from https://app.optimizely.com/mobile/android/install?project_id=2645390022
40. Dr. Tony McCaffrey. (March 24, 2014). Brainswarming: Because Brainstorming Doesn't Work [Video file]. Retrieved from https://hbr.org/video/3373616535001/brainswarming-because-brainstorming-doesnt-work
41. Tony McCaffrey. (March 25, 2014). Why You Should Stop Brainstorming. Retrieved from https://hbr.org/2014/03/why-you-should-stop-brainstorming/
42. Nir Eyal. (2014). Hooked: How to Build Habit-Forming Products. Location: //To be continued
43. Google. (May 28, 2015). Google Analytics SDK v4 for Android – Getting Started. Retrieved from https://developers.google.com/analytics/devguides/collection/android/v4/
44. Kristoffer Olofsson. (December, 2013). A Guide to Google Tag Manager for Mobile Apps. Retrieved from http://online-behavior.com/analytics/mobile-apps-tag-manager
45. Google. (April 21, 2015). Android v4 – Getting Started. Retrieved from https://developers.google.com/tag-manager/android/v4/
46. Android Developers. (n.d.). Setting Up Your Test Environment. Retrieved from https://developer.android.com/training/activity-testing/preparing-activity-testing.html
47. Android Developers. (n.d.). Creating and Running a Test Case. Retrieved from https://developer.android.com/training/activity-testing/activity-basic-testing.html
48. Android Developers. (n.d.). Testing UI Components. Retrieved from https://developer.android.com/training/activity-testing/activity-ui-testing.html
49. Android Developers. (n.d.). Creating Unit Tests. Retrieved from https://developer.android.com/training/activity-testing/activity-unit-testing.html

Appendix A: Project Plan

Elsevier’s Store Mobile App

Elsevier’s Web Store Mobile App Development

Intern:	Alia Deidranita Pramadianti
University:	Fontys University of Applied Science
Period:	3 Feb 2015 – 30 June 2015
Company Supervisor:	Maxime Corbeau
University Supervisor:	Agnes Veugen



Fontys
University of Applied Sciences

PROJECT PLAN

FONTYS UNIVERSITY OF APPLIED SCIENCES

HBO-ICT: English Stream

Data student:	
Family name , initials:	**Pramadianti, A.D.**
Student number:	**2184771**
project period: (from – till)	**3 February 2015 – 30 June 2015**
Data company:	
Name company/institution:	**Elsevier B.V.**
Department:	**E-Business**
Address:	**Radarweg 29, 1043 NX, Amsterdam**
Company tutor:	
Family name, initials:	**Corbeaum, M.**
Position:	**Manager Web Analytics & SEO**
University tutor:	
Family name , initials:	**Veugen, A.**
Final report:	
Title:	**Elsevier Store Mobile Application**
Date:	**30 June 2015**

Approved and signed by the company tutor:

Date: 5/3/2015

Signature:

Approved and signed by the university tutor:

Date:

Signature:

Agreed and signed by the student:

Date: 5/3 / 2015

Signature:

Table of Contents

1. Company

Figure 34: Amsterdam headquarters of Reed Elsevier

Elsevier BV is an academic publishing company that publishes medical and scientific literature. It provides information solutions that enhance the performance of science, health, and technology professionals, empowering them to make better decisions, deliver better care, and make groundbreaking discoveries, that advance the boundaries of knowledge and human progress.

Elsevier BV is a part of the Reed Elsevier group, an Anglo-Dutch multination publishing and information company co-headquartered in London, UK and Amsterdam, Netherlands.

Elsevier BV employs more than 7,000 people in over 70 offices across 24 countries, and publishes 2,000 journals and 20,000 books. It reported a profit margin of 36% on revenues of US$3.2 billion. One of so many ways that Elsevier does to generate its revenue is by selling books, journals, and solutions through its web store (store.Elsevier.com). In order to generate more revenue, Elsevier is starting to have more employee recruitment and creating new and challenging projects for students, one of the projects is exploring mobile apps for Elsevier's store. This project is part of their aim to grow the company.

2. Methodology

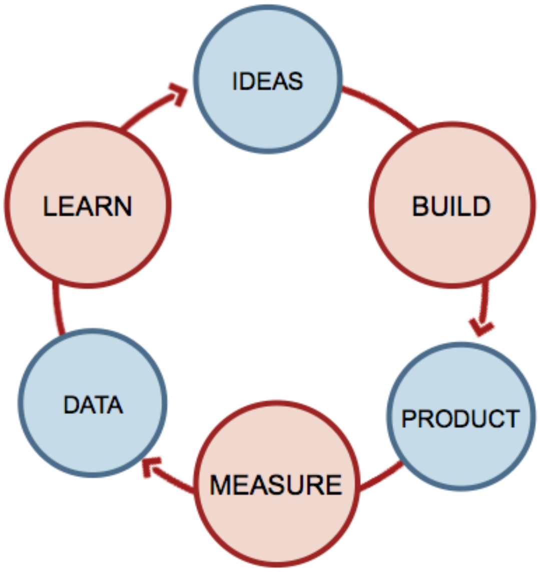


Figure 35: Lean Process

In the project, as the requirements for the product is not all clear at the start, a decision was made to use lean startup methodology for the development.

Lean startup is based on lean manufacturing, the streamline production philosophy pioneered by Taiichi Ohno by combining flow principles used by Henry Ford starting in 1906 and the TWI (Training Within Industry) programs introduced in Japan 1951. It seeks to eliminate wasteful practices and increase value producing practices during the product development phase. The customer feedback during product development is integral to lean startup process, so the producer doesn’t invest time designing features or services that consumers do not want.

The core part of the lean startup methodology is Build-Measure-Learn process. It’s explaining what the producer should do between the phases: Idea -> Build -> Product (or Code in Application Development) -> Measure -> Data -> Learn. In other words, it’s a loop process of turning ideas into products, measuring costumers’ reactions and behaviors against built products, and then learning whether to persevere the idea; the process is repeated over and over again.

3. Project Description

a. Background

Mobile applications are starting to dominate mobile world. It has been steadily growing in terms of revenues and jobs created. There are a lot of mobile applications that are published by Elsevier BV, but none of them is for Elsevier’s store. E-Business department, the one who generates revenue from its web store (a B2C value proposition making several million $ yearly), is considering to explore mobile apps as a way to drive acquisition, increase conversion, and improve retention.

Thus, to create a mobile application for Elsevier’s store, it requires further discussion about business requirements, research and identifying the necessary technology, defining the technical requirements, and implementing the application itself. The application will be invested in digital marketing and user feedback to further promotion and improvisation.

b. Research Questions

Will a mobile application be one way to drive acquisition, increase conversion, and improve retention for Elsevier’s Web Store?

c. Research Sub-Questions

- What is the target audience of the application?
- What are the business requirements of the application?
- What is the target platform of the application?
- What frameworks need to be used?
- What tools needed to develop the application?
- What functionality (features) that will be in the application, besides the ones that in the Store?

- What kind of product catalog that will be in the application?
- Will the application run offline too, or will it just run online?
- What is back-end solution for the application? API connectivity or any other way?
- What is analytics framework to use, except Google Analytics?

d. Project Boundaries

The boundaries of this project are the research on the development of mobile application for Elsevier's store and the implementation of the prototype of the application. All other things are out of scope.

e. Objectives

Main objectives:

To explore a mobile application as a way to bring more customers, get people to buy more often, and get people to come back more often.

Sub-objectives:

- To learn how to implement a mobile application for Elsevier's store.
- To give solution to e-Business department of Elsevier in generating more revenues.
- To suggest additional features that can be useful for Elsevier's customers.

f. Research and Reporting Methods

In order to do a proper research and make an adequate report, there are some strategies that are needed to be approached:

- Read articles from several sources (books and internet).
- Complete specifications indicated from company and school.
- Consult for review with the company and school mentor weekly.
- Improve and revise items based on feedback and comments from the mentors.
- Report the test results and analyses the outputs.

4. Project Charter

a. Current Situation / Problem Statement

The e-Business team of Elsevier generates revenue from its Web Store (store.elsevier.com). In the moment, e-Business team is considering to explore mobile apps as a way to drive acquisition (bring more customers), increase conversion (get people to buy more often), and improve retention (get people to come back more often).

b. Project Justification / Goal Statement

In respond to the aforementioned situation, this project was started. The underlying concept of this project is that there are many people who read e-books, and sometimes it's better to have books in their tablet and phone, so they can read it wherever they are without bringing many books in their bag. Therefore, there is a need to help them and provide them a media to buy e-books and read it directly from the app.

c. Project Product / Scope In or Out

The final product of this project is represented by a mobile application for Elsevier's Web Store. Through this application, users are able to buy and download e-books using their mobile(s), and also read the e-books directly.

Elsevier's store mobile application is intended for day-to-day use, ranging from study related to work related problems, and covers science, health, and technology topics. The target audiences of the application are therefore the professors, students, librarians, and other practitioners who fit the situations.

Scope In	Scope Out
Application	**Minutes meeting**
User manual [optional]	**Training**
Developer manual [optional]	
Project plan	
Requirements document [optional]	
Design document [optional]	
Test plan [optional]	
Project report	
Presentations	

d. Constraints

1. The deadline of delivering the end of product is 12 June 2015.
2. The intern should work 36 hours / week, being present during meeting(s).
3. The application should run on IOS and Mac OS.
4. Any absent permission should be informed to the supervisors through e-mail.
5. All documentations are enclosed only to the company and school supervisor. The documents are strictly prohibited for external publication.

e. Risk and Solutions

No.	Risk	Solution
1.	Miscommunication	• Weekly meeting • Reported problem on the documentation • Review
2.	Lack of technical knowledge	• Consult expert • Research technical solutions (internet and media source)
3.	Errors and Bugs	• Debug by testing software • Consult test engineers
4.	Insufficient working time	• Extra working hour before normal arrival

		time • Extra work in the weekend • Re-adjust project phasing
5.	Lack of documentation	• Documents are available in google drive, the intern enables the accessibility of the document(s) to the supervisor
6.	Failing deadlines	• Carefully scheduling activities

f. Project Phase and Time

Task	2-02-15	9-02-15	16-02-15	23-02-15	2-03-15	9-03-15	16-03-15	23-03-15	30-03-15	6-04-15	13-04-15	20-04-15	27-04-15	4-05-15	11-05-15	18-05-15	25-05-15	1-06-15	8-06-15	15-06-15	22-06-15	29-06-15
Company Supervisor Meeting	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
School Supervisor Meeting					■						■							■				■
Documentation	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Internship Data	■	■																				
Logbook and Milestone	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Project Plan	■	■	■	■																		
User Manual **																	■	■				
Developer Manual **																			■	■		
Requirements Documents **					■		■		■		■		■		■		■		■			
Design Documents **					■		■		■		■		■		■		■		■			
Test Plan **																■	■					
App Development Phase	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Research Dev Framework	■	■	■																			
Dev Frameworks Demo + Decision				■																		
Develop Generic Features					■	■	■	■	■	■												
Develop Specific Features											■	■	■	■	■	■	■					
Application Testing											■							■	■			
Go Live																				■		
Bug Fixing																					■	■
Final Report	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Introduction	■	■	■	■																		
Requirements and Design Specification					■		■		■		■			■								
Mobile Application Development								■						■			■		■		■	
Conclusion and Recommendation																				■	■	
Presentation Preparation																					■	■

- * The schedule plan is tentative; might be changed depends on the certain situation.
- ** Additional items

g. Stakeholders

i. Supervisor

Company

Name : Maxime Corbeau
Position : Web Analytics and SEO Manager
E-mail : m.corbeau@elsevier.com

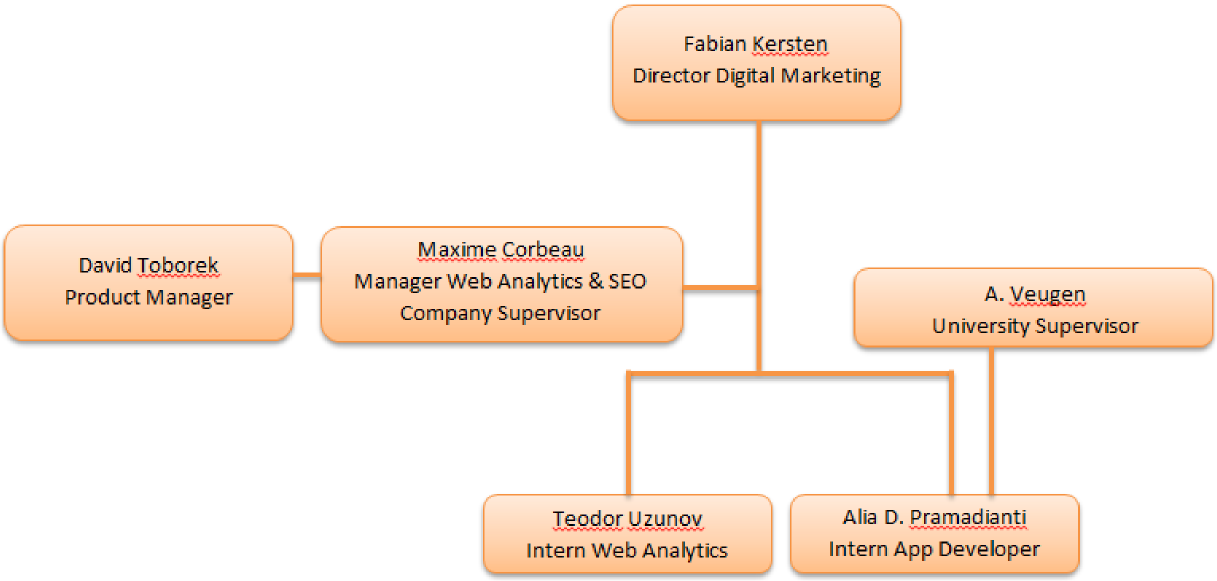
University

Name : Agnes Veugen

Position : Teacher at Fontys University of Applied Science

E-mail : a.veugen@fontys.nl

ii. Organization



- **Information**

	David	Max *(Company Supervisor)*	Alia	Fabian	Teodor	Mrs. Veugen *(University Supervisor)*
Overall Delivery				A		
Business Requirements	C	C	I	R	I	
IT	R	I		I		
Analytics		R		I	R	
Design	C	C	R	C		I
Development		I	R			I
Project Planning	I	R	I	I	I	Ap
Weekly Update Meetings			R			
Final Report		Ap	R			Ap

R = Responsible | A = Accountable | C = Consulted | I = Informed | Ap = Approve

Appendix B: Pain Points of & Goals for the Better Elsevier's Web Store

✓ Pain Points of Elsevier's Web Store

- Difficult to use / navigate the site
- Challenges in servicing customer (delivery)
- DRM formats for e-Books

✓ Goals for the Better Elsevier's Web Store

Acquisition	Conversion	Retention	UX	Goal	Count
?	x	x		Find deals/coupon codes (existing)	3
	x	x	x	Get recommendations	3
	x	x		Renew Journal / Subscriptions	2
	x	x		Suscribing for news (coupon, new books….)	2
	x		x	Find information on specific product	2
x			x	Discover new products	2
x		x		Write/Read book reviews	2
		x	x	Be updated on latest/forthcoming products	2
	x		x	Order offline	2
	x		x	Use iDeal or Paypal to pay	2
			x	Track order status	1
	x			Buy book	1
			x	Download eBook	1
	x			Use Coupon Codes	1
			x	Find product information on the go	1
			x	Buy products as guest (idea: more simple)	1
	x			Ask for a deal (request new, teacher, other personas)	1
		x		Rate a deal/promo	1
			x	Open book preview	1
	x			Compare products	1
			x	Provide feedback to Store team	1
x				Become affiliate to sell books	1
		x		Get a Desk Copy	1
			x	Access my (e)product immediately	1

Table 4: List of goals for better Elsevier's web store

Find Deals or Coupons

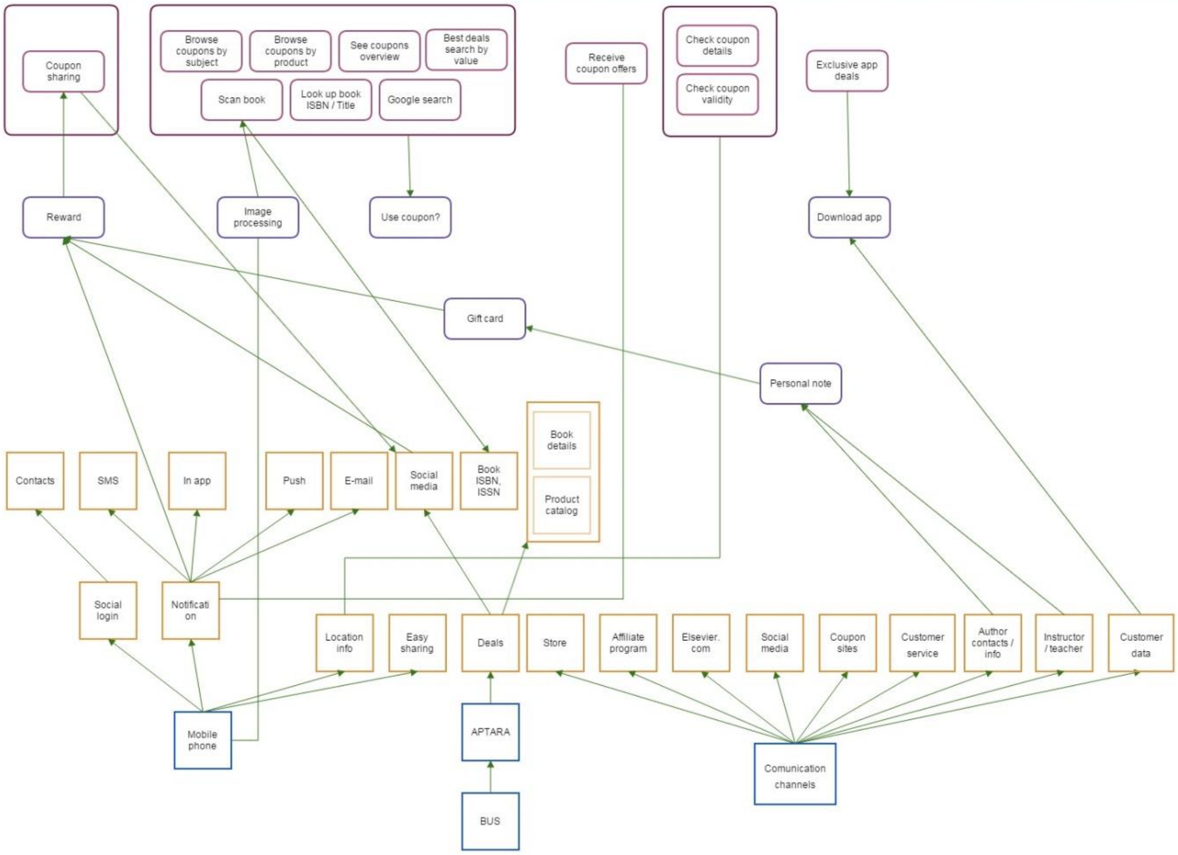


Figure 36: Example of Brainswarming graph based on the first goal that want to be achieved

Appendix C: Ideas to Make an App Sticky / Successful

Acquisition	
Idea	**Example app/company**
Track & measure everything with tracking tools	
Make the application free	
Be strategic about distribution: • Positive coverage in the media • Noticed and favoured by customers	
Let influential technology bloggers/readers to try a test version of the app	Instagram
Initial press coverage	Evernote
Make a connection with all publishers	Kindle Amazon
All out to get technology blog coverage	Clear
Demographic targeted	
High volume, low price (content)	
Usability testing	
Don’t neglect icon (first impression)	
Use native code • Avoid javascript (phonegap) • Try appcelerator • Choose technology based on design	
Start with iPhone, but never forget Android	

Table 5: Acquisition strategy

Referral / Virality	
Idea	**Example app/company**
Incorporate viral mechanisms with core functionality	
Attract network of new users	
Engage the users to invite friends	
No-nonsense interface	Clear
Unique	
Provide true mobility, ability to run offline mode	
Provide help / FAQ page	

Table 6: Referral or Virality strategy

Retention	
Idea	**Example app/company**
Original, improve, solve problem, entertain, well designed (visuals and user experience)	
Intuitive (offer fantastic experiences, free of bugs and crashes)	
Social media share	Instagram
Reasonable price, not that expensive for a book	Kindle Amazon
Pretty	
Memorable -> major impression	
Shareable	
Fun -> entertain the users	
Bring people together	
Design according to how the users thinks about things	
Provide good design of the application flow	
Visuals matter	
Spice up words for instructional copy, error message, notification	Sonicomobile
Animate with class	

Table 7: Retention strategy

Revenue	
Idea	**Example app/company**
Leave it to The Experts • Mobile traffic and marketing provides (highly fragmented, CPI) • Pay CPI only, and ensure that healthy share of what you buy is incentivized	
Low profit margin	Kindle Amazon
Profit gained from e-books selling, not hardware	
Ecosystem selling points	
Subscription	
From digital content streaming and downloading, app purchases, and advertising revenues	
Took only 2% share of global tablet shipments	
Offering little utility, tiny details elicit powerful emotional response	

Table 8: Revenue strategy

Activation	
Idea	**Example app/company**
Consider small markets, such as Asia, Latin America	
Marketing is key, and in advance	
Articles in TechCrunch	Instagram
Closed beta	Evernote, Clear
In all app store launches on Day One	Evernote
Many options of the books, and those books are not available in other stores	Kindle Amazon
Video demo in Vimeo	Clear
Platform appropriate, compatible and functional for all brands and devices	
Easy to install, access, and use	
Spontaneous -> accessible for short bursts of user activity	
Huge launches based on intense marketing campaigns	
Appeal to almost everybody	
Pay attention to UI Guidelines IOS / Android	
Make it run offline	

Table 9: Activation strategy

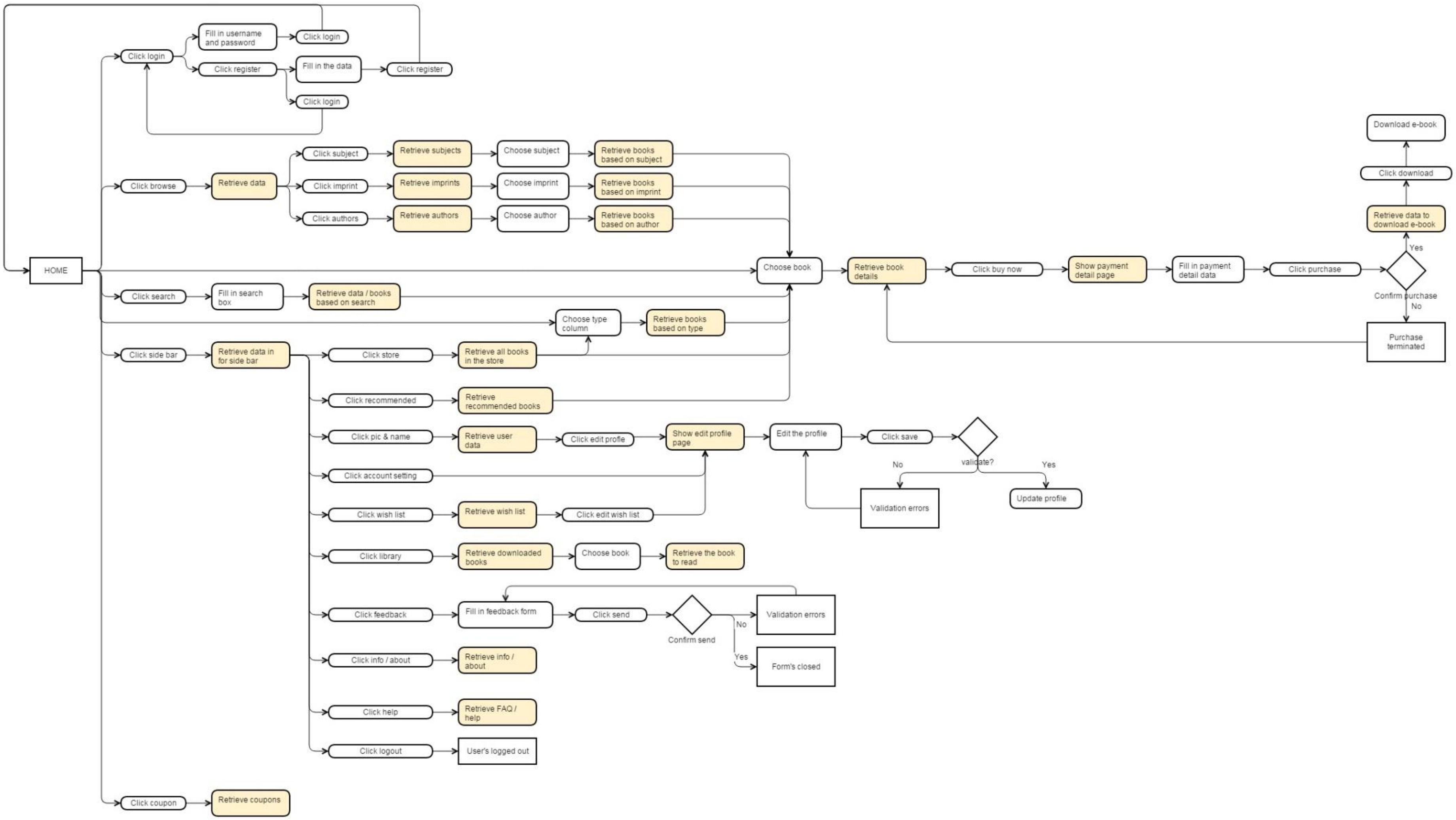
Appendix D: Minimum Viable Product

Category	Items
JSON	• GET request • GET request with parameter (e.g. coupon code) • POST request (e.g. Mandrill emails) • Based URL Configured via GTM
Crash reporting	Track issues / crashes
Push notifications	• Send • Track open rate • Open specific app screen after clicking the push notification
Quality assurance – App naming	• Define naming convention • Naming convention for Mixpanel, New Relic, Google Tag Manager, Spreadsheet
All	Add back button '<' in top bar of all screens
Homescreen	If API is down, display an error message instead of homescreen
All Coupons	• Give list of all coupons • Filter on "New" (must have "New" Flag from API) and NOT HAVE "SEEN" Flag from local storage • Click on coupon to see details • Exclude expired coupons • Free search (filter on anything)
Check coupon code	• Check exact coupon code • Check coupon code case insensitive
Browse by subject	• See list of subjects • See number of offers per subject • Put subject as favourite • Remove subject from favourite • See offers/coupons for a particular subject • Click on coupon to see details • Sort subjects by number of offers • Sort subjects by name ascending and descending • If first time favourite subject, bring subject to the option for notifications
Exclusive deals	• Give list of all exclusive coupons • Click on coupon to see details • When user sees an exclusive coupon, remember user has seen it • Homescreen: display red circle with number of exclusive deals user hasn't already seen
My favourites	• See only favourites items • Remove items from favourite
My account	• Notification options (push notification, email) • Add or edit e-mail • Save the data of notification, email, etc. in local storage or 3rd party services

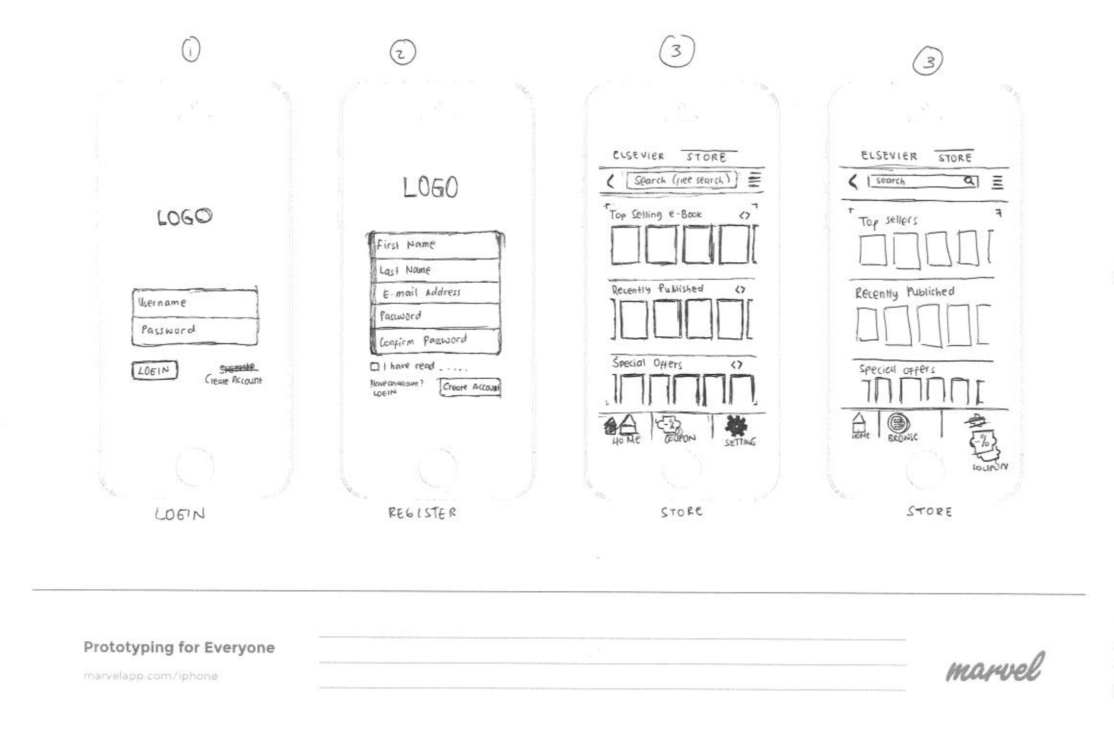
Coupon details	• Email me the coupon • Coupon code • Coupon validity • Coupon description
Email the coupon	Email to me (the data need to be integrated with my account)
Coupon validity	• Make it green when it's still valid • Make it orange when it's about to expired • Make it red when it's expired • Display "Only N days left", when it's about to expired
Admin/User accounts	Use admin Elsevier e-mail for Mixpanel, New Relic, GTM, GCM
Google Tag Manager - Variable	App settings include colors, strings, icons, coupon about to expire threshold, api URL, etc.
Analytics	Google Tag Manager, Mixpanel, Google Analytics

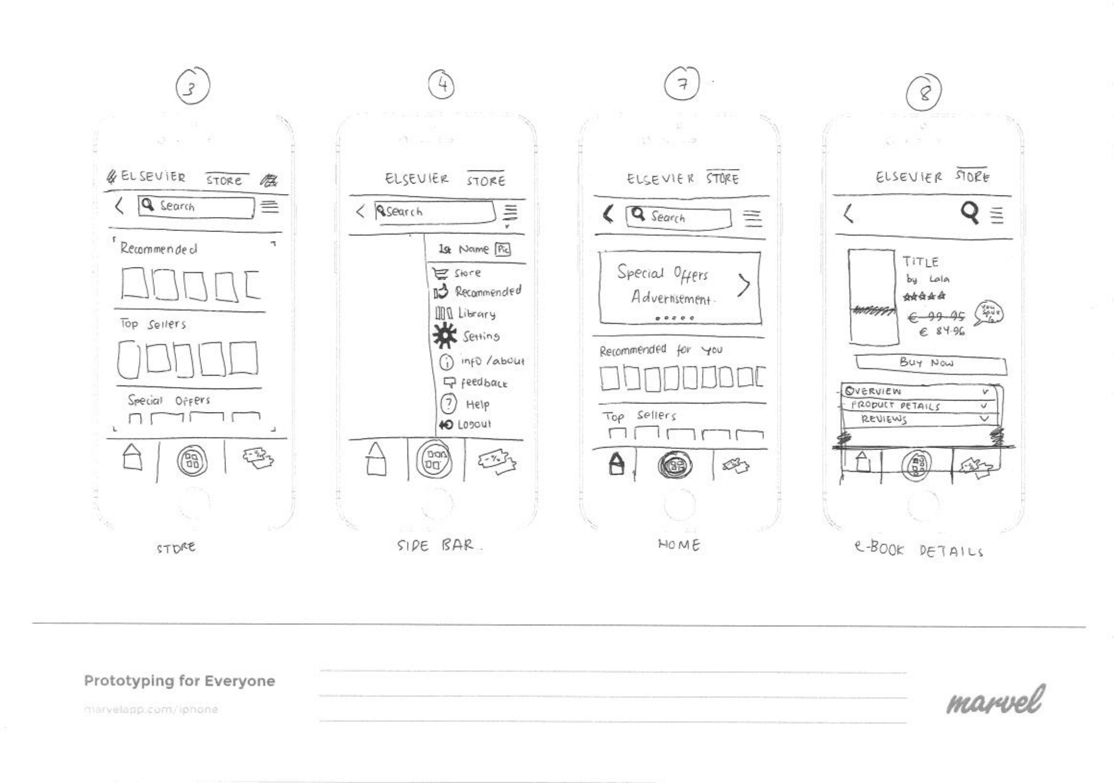
Table 10: Mininum Viable Product of Elsevier's store mobile app

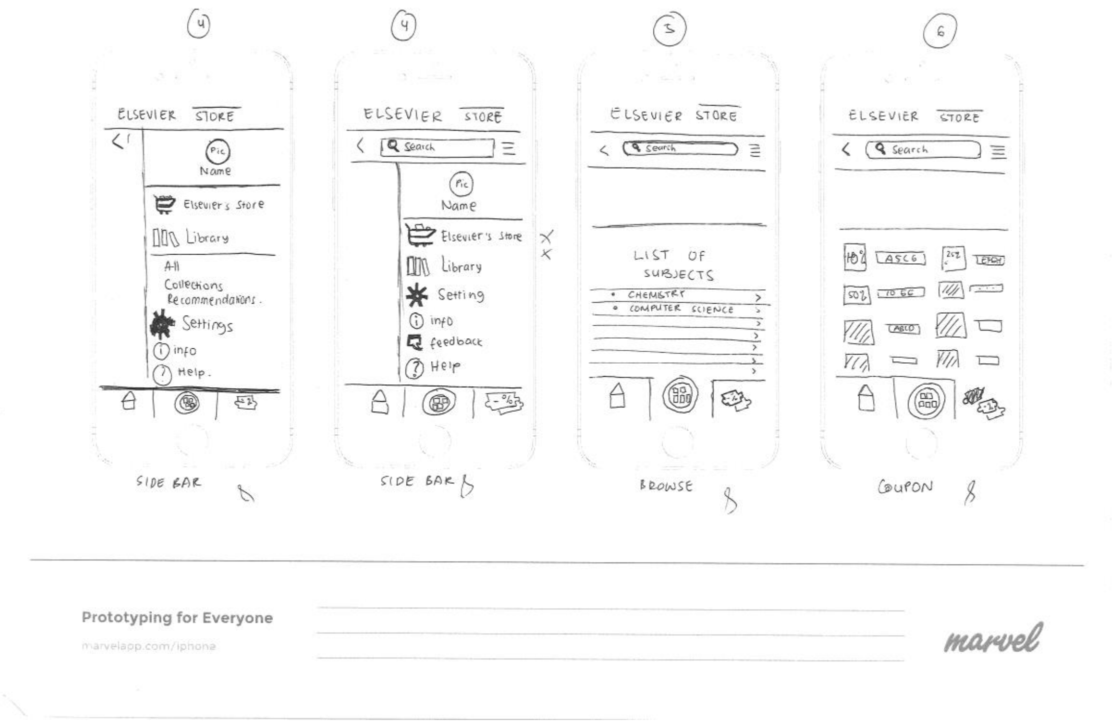
Appendix E: Action State Diagram of Initial Plan

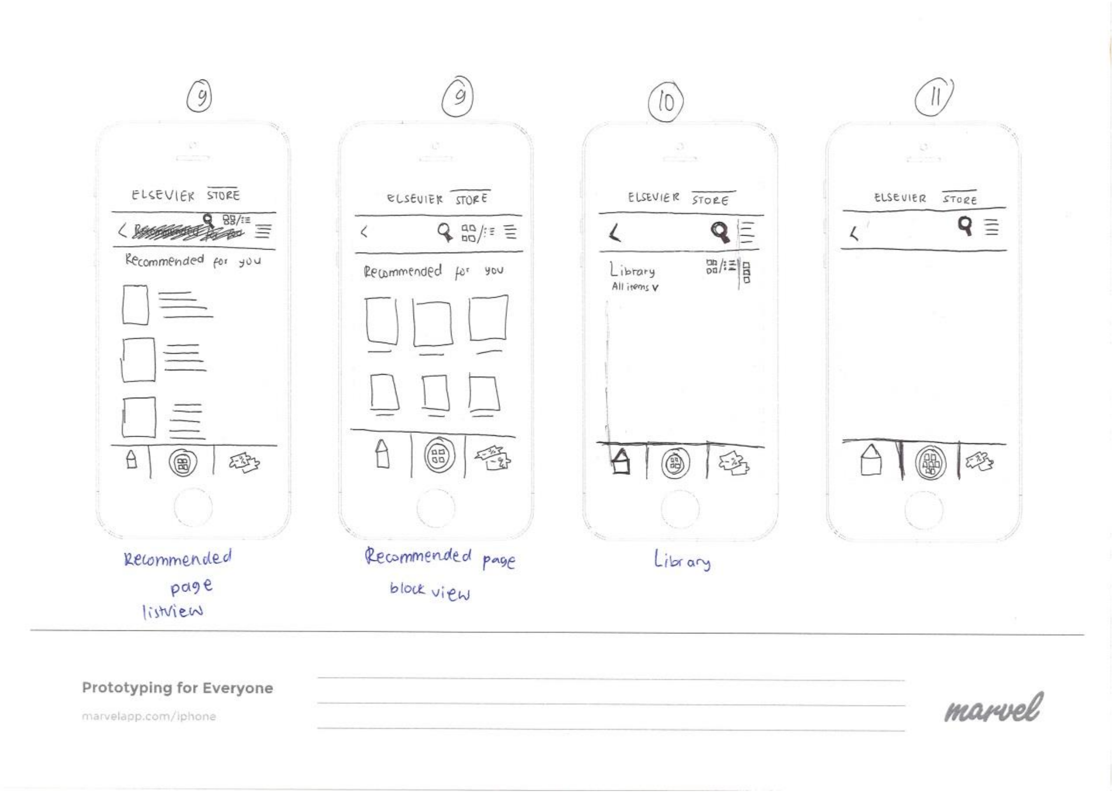


Appendix F: Wireframe of Elsevier’s Store Mobile App on Paper






4
ELSEVIER STORE
Pic
Name
Elsevier's Store
Library
All
Collections
Recommendations.
Settings
info
Help.
SIDE BAR
4
ELSEVIER STORE
Search
Pic
Name
Elsevier's store
Library
Setting
info
feedback
Help
SIDE BAR
5
ELSEVIER STORE
Search
LIST OF
SUBJECTS
CHEMISTRY
COMPUTER SCIENCE
BROWSE
6
ELSEVIER STORE
Search
10%
ASC6
50%
ABCD
COUPON
Prototyping for Everyone
marvelapp.com/iphone
marvel


9
ELSEVIER STORE
Recommended for you
Recommended
page
listview
9
ELSEVIER STORE
Recommended for you
Recommended page
block view
10
ELSEVIER STORE
Library
All items v
Library
11
ELSEVIER STORE
Prototyping for Everyone
marvelapp.com/iphone
marvel

Appendix G: Prototype and Screen Flow

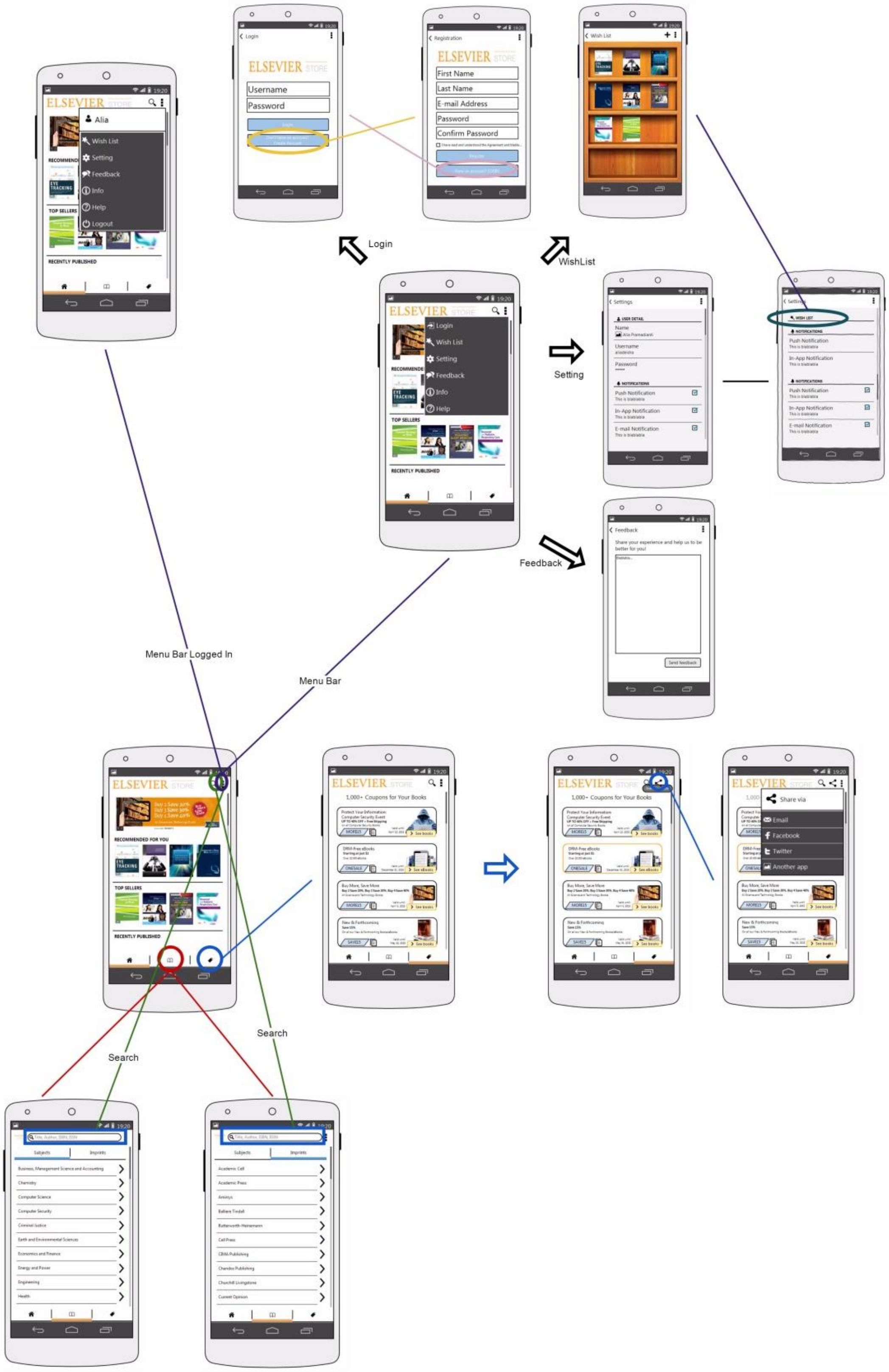


Figure 37: First Prototype and Screen Flow

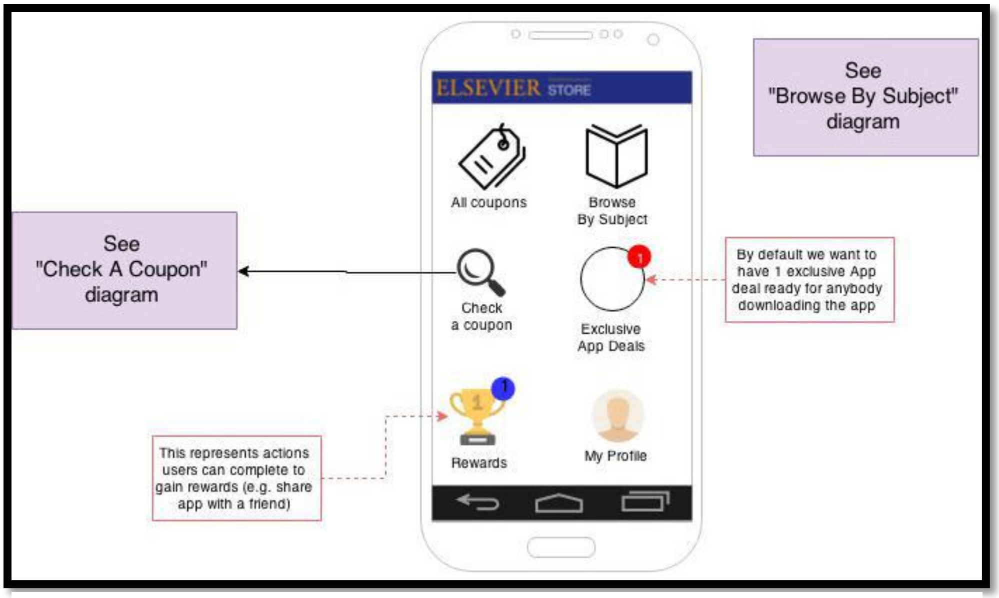

ELSEVIER STORE
All coupons
Browse By Subject
See "Browse By Subject" diagram
See "Check A Coupon" diagram
Check a coupon
1
Exclusive App Deals
By default we want to have 1 exclusive App deal ready for anybody downloading the app
1
Rewards
My Profile
This represents actions users can complete to gain rewards (e.g. share app with a friend)

Figure 38: Homepage screen flow

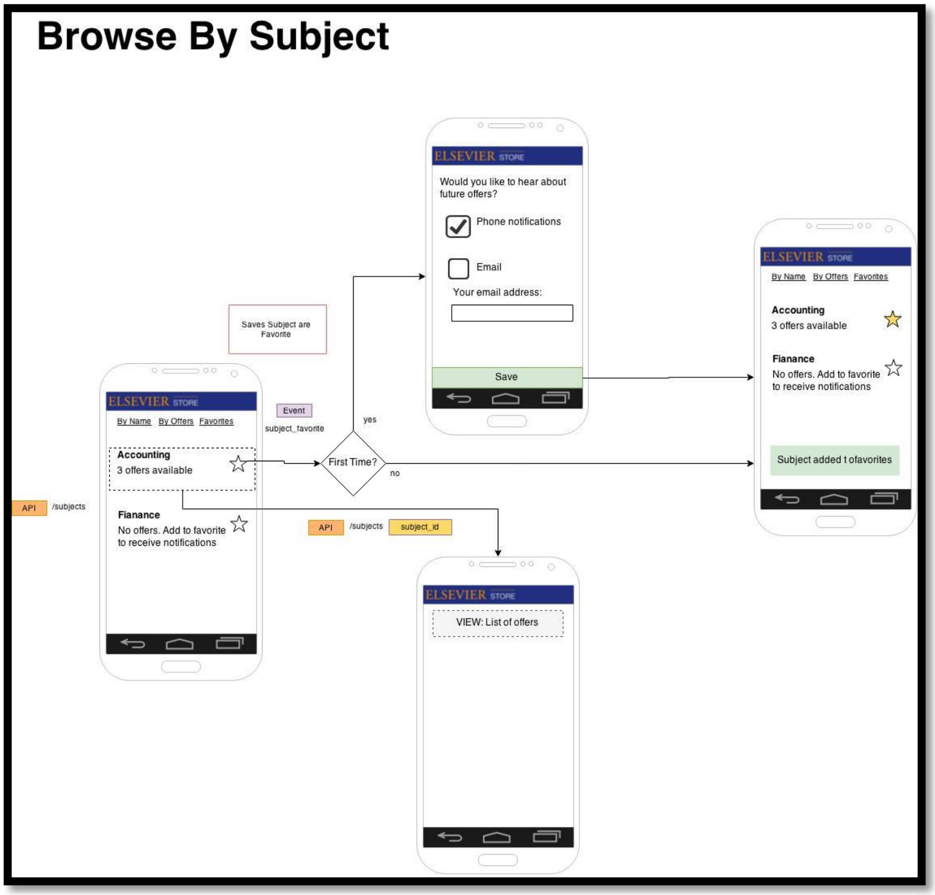


Figure 39: Browse by Subject screen flow

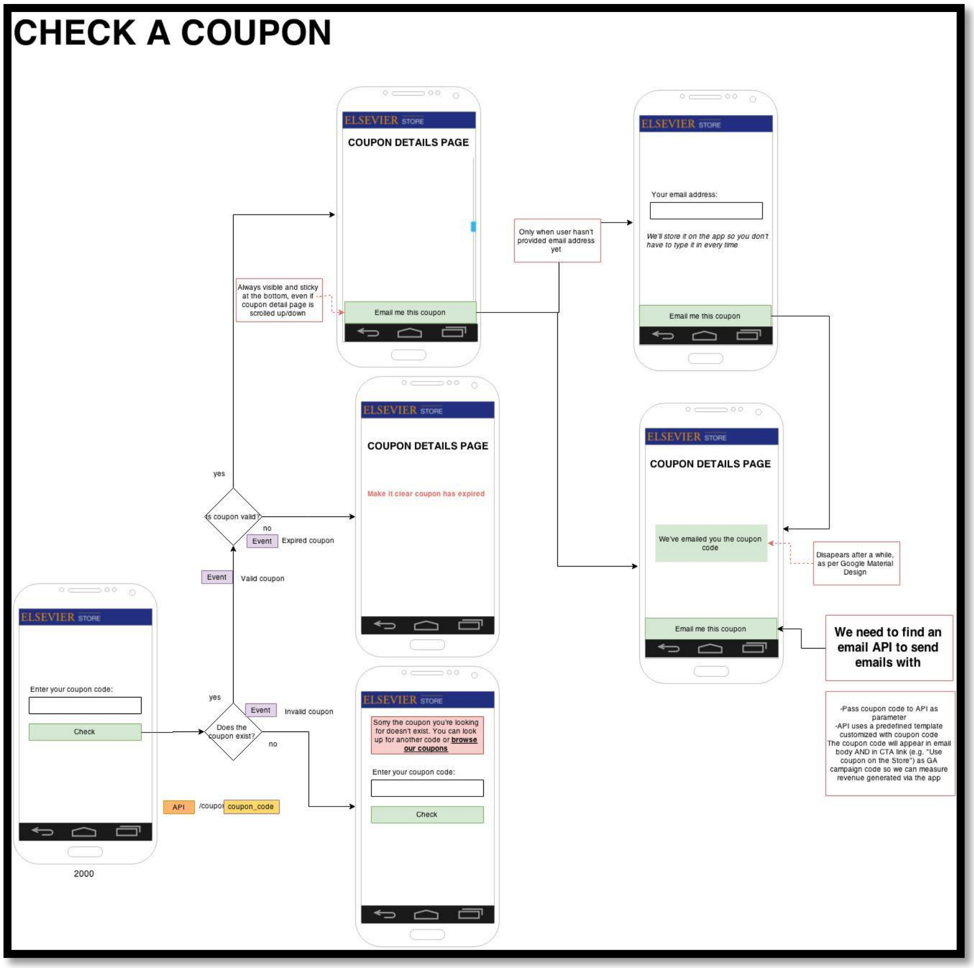

CHECK A COUPON
ELSEVIER STORE
Enter your coupon code:
Check
2000
API
/coupon
coupon_code
Does the coupon exist?
yes
no
Event
Invalid coupon
Event
Valid coupon
Is coupon valid?
yes
no
Event
Expired coupon
Always visible and sticky at the bottom, even if coupon detail page is scrolled up/down
COUPON DETAILS PAGE
Email me this coupon
Make it clear coupon has expired
Sorry the coupon you're looking for doesn't exist. You can look up for another code or browse our coupons
Enter your coupon code:
Check
Only when user hasn't provided email address yet
Your email address:
We'll store it on the app so you don't have to type it in every time
Email me this coupon
COUPON DETAILS PAGE
We've emailed you the coupon code
Disapears after a while, as per Google Material Design
Email me this coupon
We need to find an email API to send emails with
-Pass coupon code to API as parameter
-API uses a predefined template customized with coupon code
The coupon code will appear in email body AND in CTA link (e.g. "Use coupon on the Store") as GA campaign code so we can measure revenue generated via the app

Figure 40: Check a Specific Coupon Code screen flow

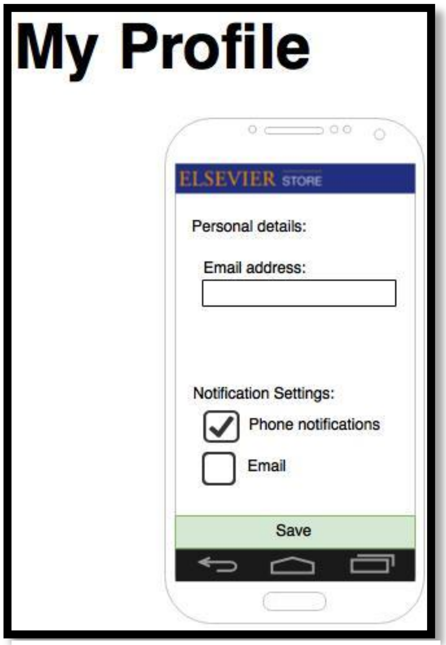


Figure 41: My account screen flow

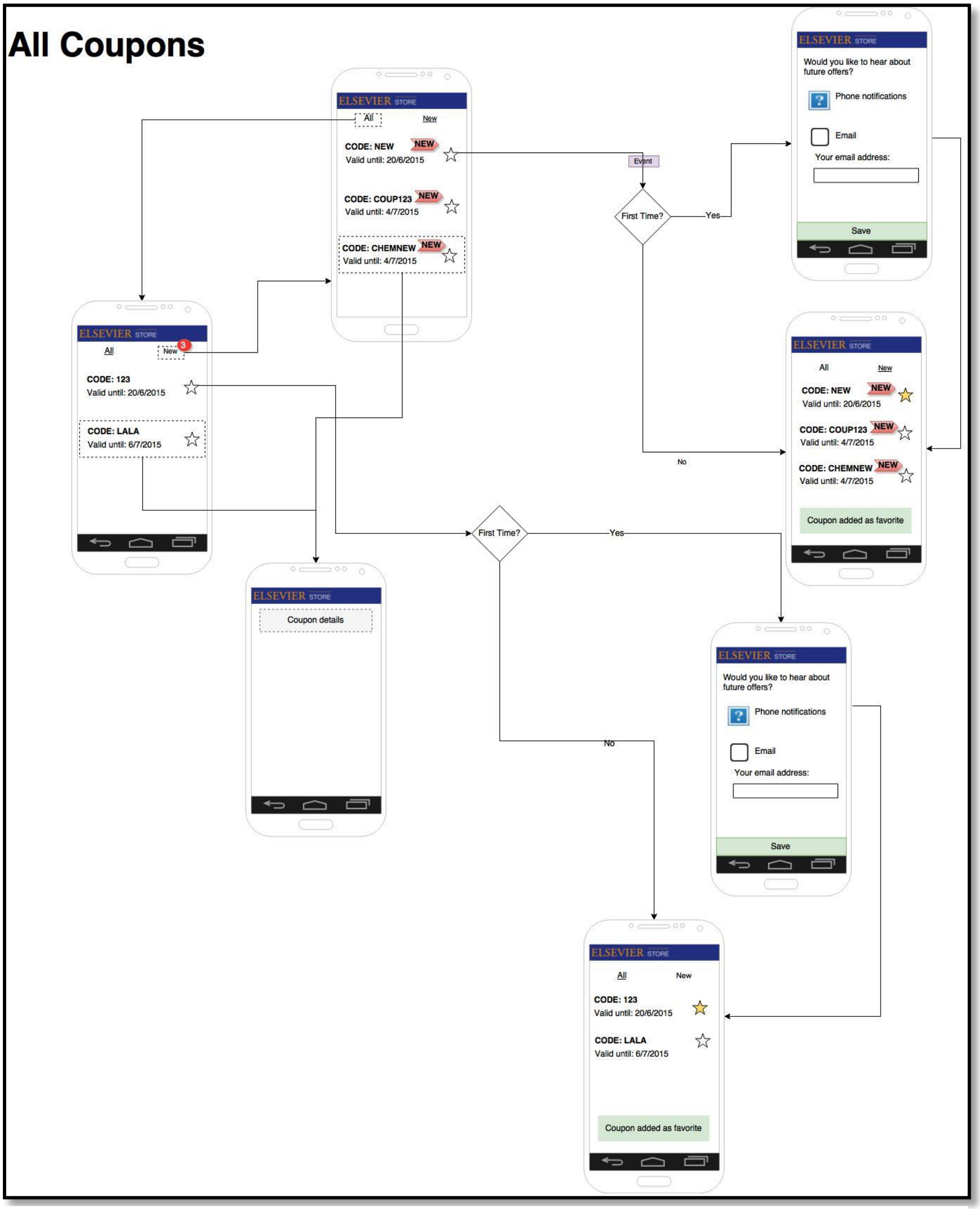


Figure 42: All coupons screen flow

Appendix H: Final Application Screen Shot

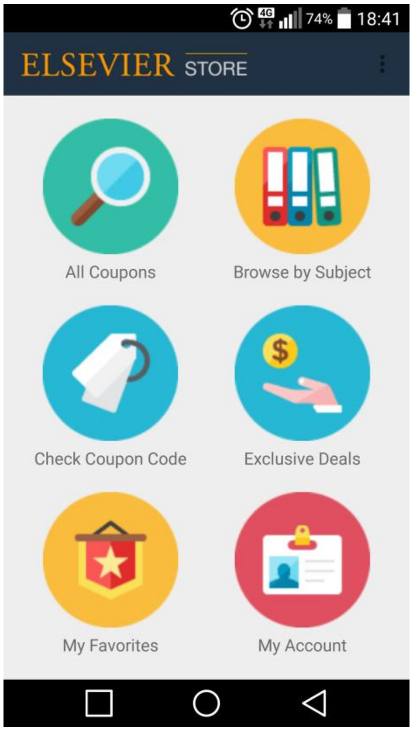
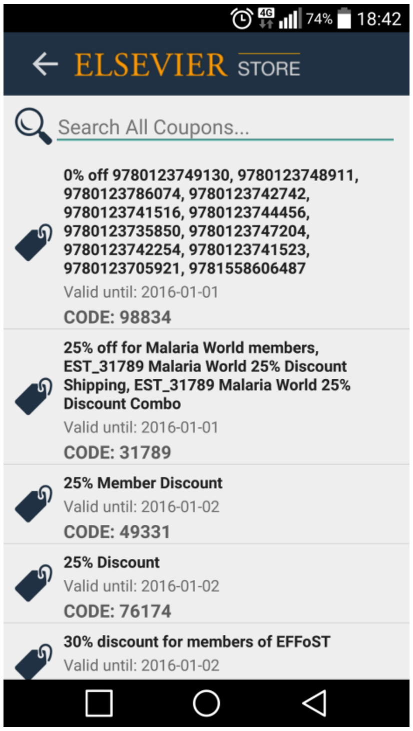


Figure 48: Homepage



0% off 9780123749130, 9780123748911, 9780123786074, 9780123742742, 9780123741516, 9780123744456, 9780123735850, 9780123747204, 9780123742254, 9780123741523, 9780123705921, 9781558606487
Valid until: 2016-01-01
CODE: 98834

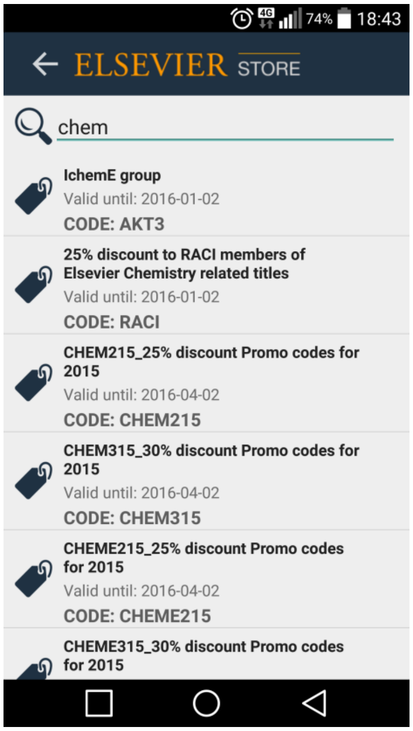
25% off for Malaria World members, EST_31789 Malaria World 25% Discount Shipping, EST_31789 Malaria World 25% Discount Combo
Valid until: 2016-01-01
CODE: 31789

25% Member Discount
Valid until: 2016-01-02
CODE: 49331

25% Discount
Valid until: 2016-01-02
CODE: 76174

30% discount for members of EFFoST
Valid until: 2016-01-02

Figure 47: All coupons list



IchemE group
Valid until: 2016-01-02
CODE: AKT3

25% discount to RACI members of Elsevier Chemistry related titles
Valid until: 2016-01-02
CODE: RACI

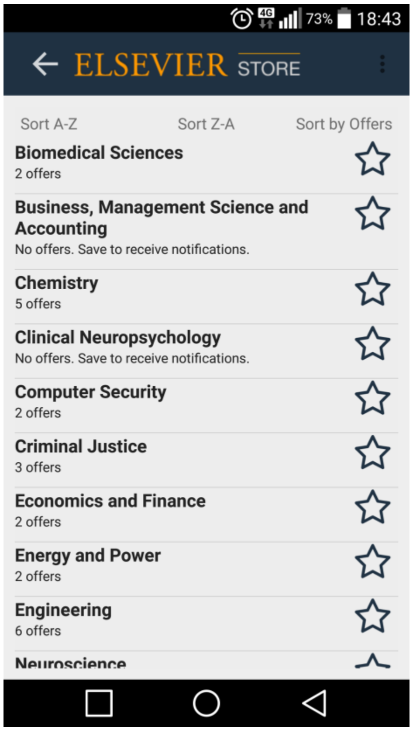
CHEM215_25% discount Promo codes for 2015
Valid until: 2016-04-02
CODE: CHEM215

CHEM315_30% discount Promo codes for 2015
Valid until: 2016-04-02
CODE: CHEM315

CHEME215_25% discount Promo codes for 2015
Valid until: 2016-04-02
CODE: CHEME215

CHEME315_30% discount Promo codes for 2015

Figure 46: All coupons filtered



Biomedical Sciences
2 offers

Business, Management Science and Accounting
No offers. Save to receive notifications.

Chemistry
5 offers

Clinical Neuropsychology
No offers. Save to receive notifications.

Computer Security
2 offers

Criminal Justice
3 offers

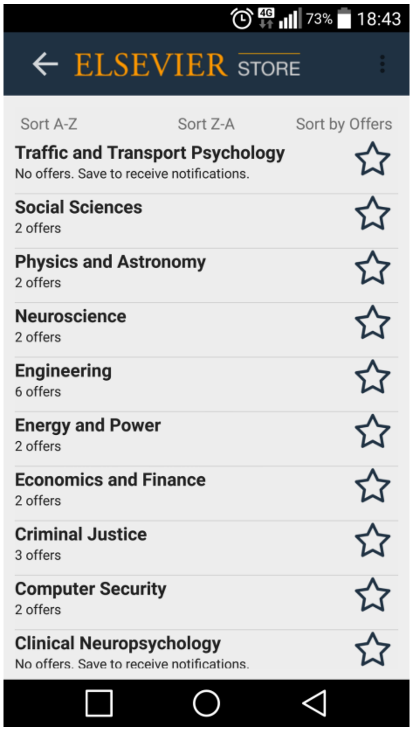
Economics and Finance
2 offers

Energy and Power
2 offers

Engineering
6 offers

Neuroscience

Figure 45: Subject list sorted ascending



Traffic and Transport Psychology
No offers. Save to receive notifications.

Social Sciences
2 offers

Physics and Astronomy
2 offers

Neuroscience
2 offers

Engineering
6 offers

Energy and Power
2 offers

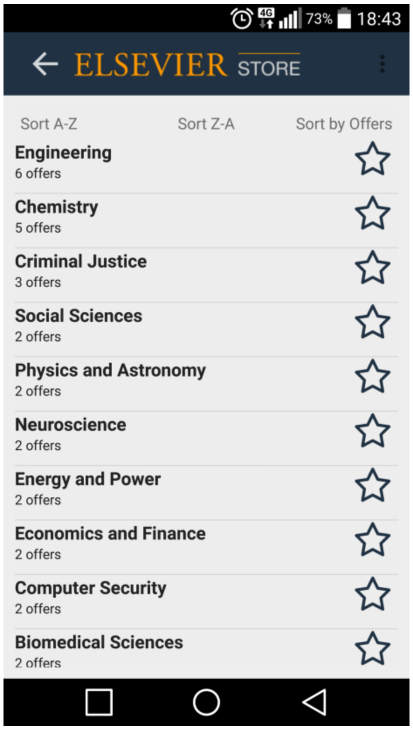
Economics and Finance
2 offers

Criminal Justice
3 offers

Computer Security
2 offers

Clinical Neuropsychology
No offers. Save to receive notifications.

Figure 44: Subject list sorted descending



Engineering
6 offers

Chemistry
5 offers

Criminal Justice
3 offers

Social Sciences
2 offers

Physics and Astronomy
2 offers

Neuroscience
2 offers

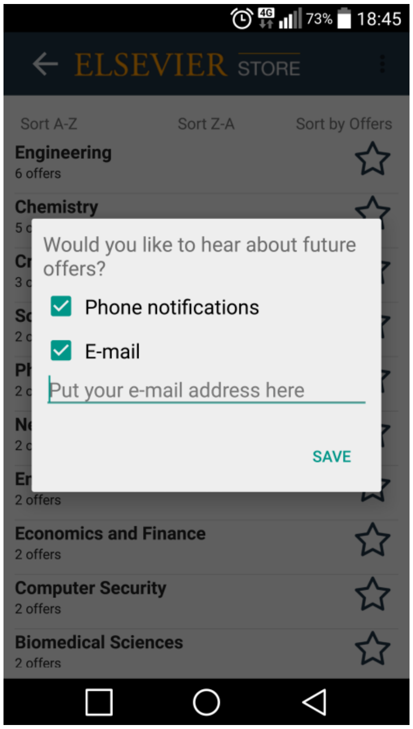
Energy and Power
2 offers

Economics and Finance
2 offers

Computer Security
2 offers

Biomedical Sciences
2 offers

Figure 43: Subject list sorted by number of offers



Engineering
6 offers

Chemistry

Would you like to hear about future offers?

☑ Phone notifications

☑ E-mail

Put your e-mail address here

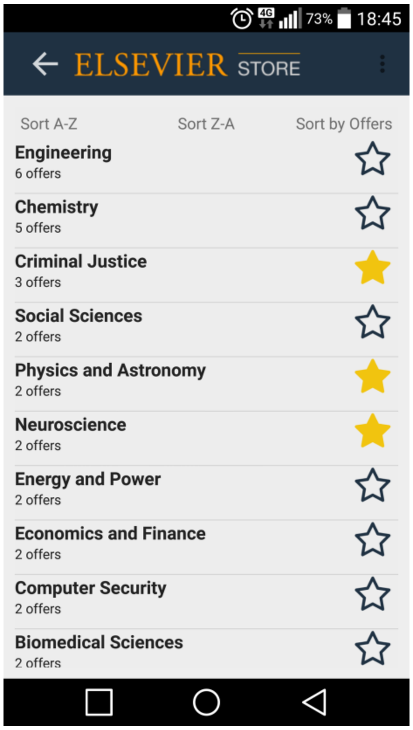
2 offers

Economics and Finance
2 offers

Computer Security
2 offers

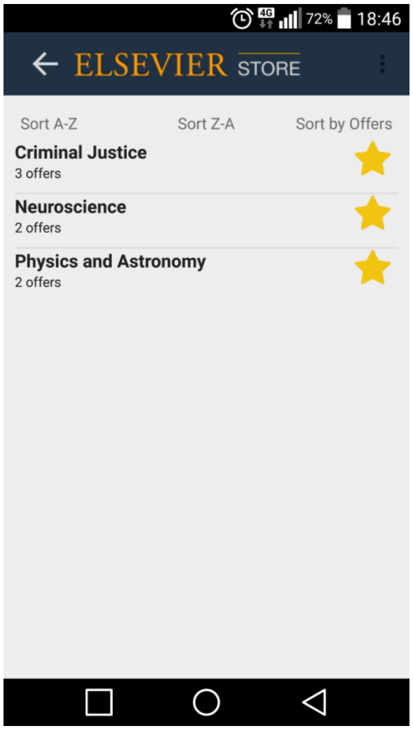
Biomedical Sciences
2 offers

Figure 51: Notification options for future offers when first time making a subject as favorite



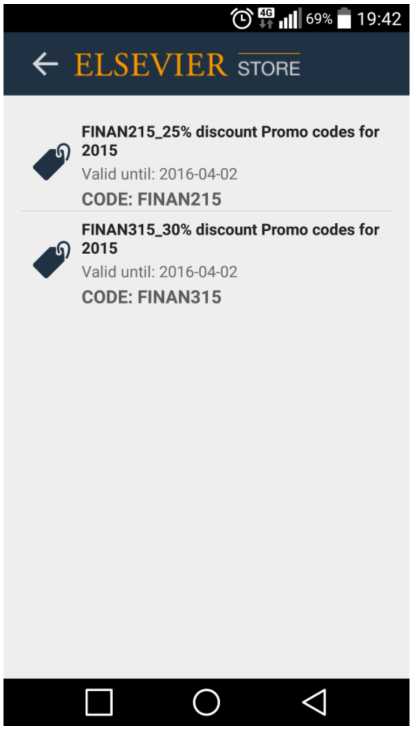
Engineering 6 offers	☆
Chemistry 5 offers	☆
Criminal Justice 3 offers	★
Social Sciences 2 offers	☆
Physics and Astronomy 2 offers	★
Neuroscience 2 offers	★
Energy and Power 2 offers	☆
Economics and Finance 2 offers	☆
Computer Security 2 offers	☆
Biomedical Sciences 2 offers	☆

Figure 49: Subject list with some favourites subjects



Criminal Justice 3 offers	★
Neuroscience 2 offers	★
Physics and Astronomy 2 offers	★

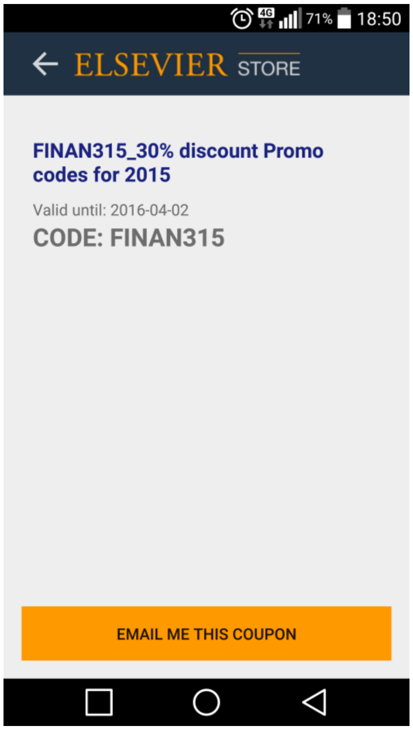
Figure 50: Favorite page



FINAN215_25% discount Promo codes for 2015
Valid until: 2016-04-02
CODE: FINAN215

FINAN315_30% discount Promo codes for 2015
Valid until: 2016-04-02
CODE: FINAN315

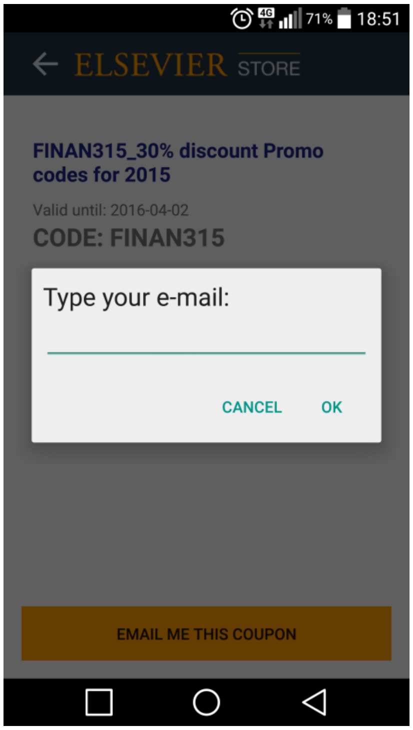
Figure 52: List of coupon of a specific subject



FINAN315_30% discount Promo codes for 2015

Valid until: 2016-04-02
CODE: FINAN315

Figure 53: Coupon details page

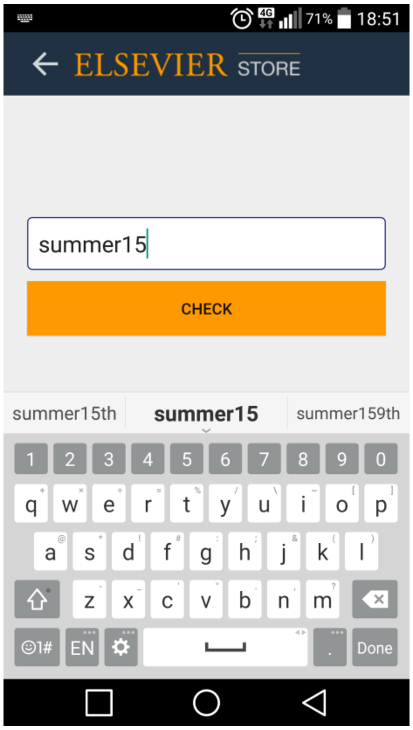


FINAN315_30% discount Promo codes for 2015

Valid until: 2016-04-02
CODE: FINAN315

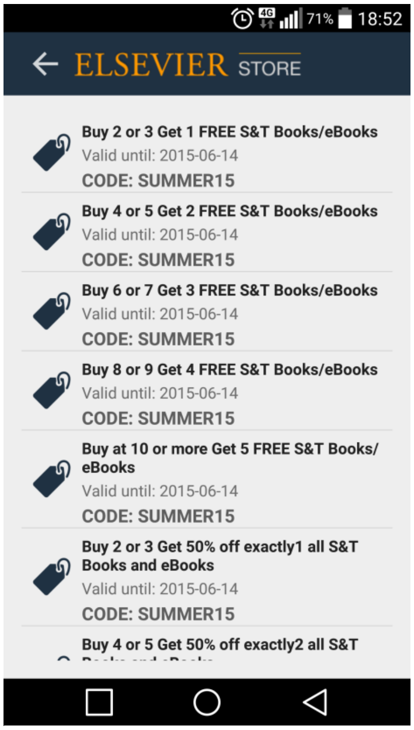
Type your e-mail:

Figure 54: Put e-mail options when clicking 'email me' button



summer15

Figure 55: Check a specific coupon code page



Buy 2 or 3 Get 1 FREE S&T Books/eBooks
Valid until: 2015-06-14
CODE: SUMMER15

Buy 4 or 5 Get 2 FREE S&T Books/eBooks
Valid until: 2015-06-14
CODE: SUMMER15

Buy 6 or 7 Get 3 FREE S&T Books/eBooks
Valid until: 2015-06-14
CODE: SUMMER15

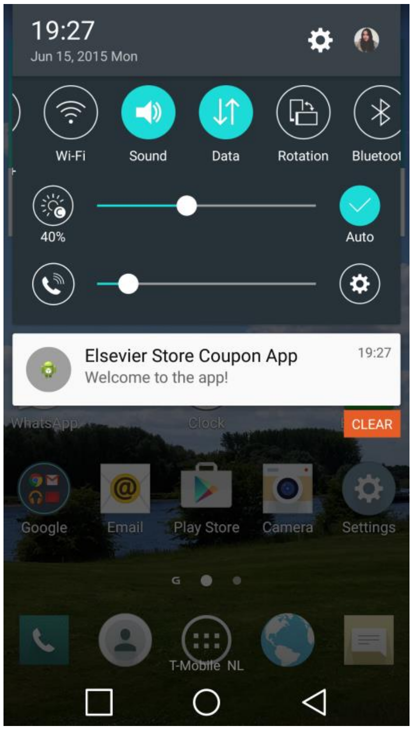
Buy 8 or 9 Get 4 FREE S&T Books/eBooks
Valid until: 2015-06-14
CODE: SUMMER15

Buy at 10 or more Get 5 FREE S&T Books/ eBooks
Valid until: 2015-06-14
CODE: SUMMER15

Buy 2 or 3 Get 50% off exactly1 all S&T Books and eBooks
Valid until: 2015-06-14
CODE: SUMMER15

Buy 4 or 5 Get 50% off exactly2 all S&T

Figure 56: List of coupons given after checking the code



Elsevier Store Coupon App 19:27
Welcome to the app!

Figure 57: Push notification received