**Data student:**
- **Family name, initials:** Pramadianti, A. D.
- **Student number:** 2184771
- **Project period:** 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:** [Signature Image]
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 ........................................................................................................................................... i

Glossary ........................................................................................................................................... ii

1. Introduction .................................................................................................................................. 1

2. Company ...................................................................................................................................... 2

3. Project Overview .......................................................................................................................... 4
   3.1. Initial Situation ......................................................................................................................... 4
   3.2. Project Objective ...................................................................................................................... 4
   3.3. Project Description .................................................................................................................... 5
   3.4. Method of Approach ................................................................................................................ 8

4. Project Process ............................................................................................................................. 9
   4.1. Research ................................................................................................................................ 9
       4.1.1. Hybrid Frameworks .......................................................................................................... 9
       4.1.2. Features for Initial Plan of Elsevier’s Store Mobile App .................................................... 11
       4.1.3. What Makes an Application Sticky / Successful? ............................................................... 13
       4.1.4. Data Storage ..................................................................................................................... 14
       4.1.5. Hybrid VS Native Programming ......................................................................................... 14
       4.1.6. Target Platform ................................................................................................................ 17
       4.1.7. Dependency Manager for Android ..................................................................................... 18
       4.1.8. 3rd Party Services .............................................................................................................. 20
       4.1.9. Minimum Viable Product (MVP) ....................................................................................... 25

   4.2. Design ................................................................................................................................... 27
       4.2.1. Action State Diagram ....................................................................................................... 27
       4.2.2. Wireframe .......................................................................................................................... 28
       4.2.3. First Prototype and Screen Flow ....................................................................................... 28
       4.2.4. Second Prototype and Screen Flow .................................................................................. 29
       4.2.5. Third Prototype and Screen Flow ..................................................................................... 29

   4.3. Implementation ...................................................................................................................... 30
       4.3.1. Version 1 ............................................................................................................................ 30
       4.3.2. Version 2 ............................................................................................................................ 30
       4.3.3. Version 3 ............................................................................................................................ 31
       4.3.4. Version 4 ............................................................................................................................ 32
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 widespread 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.

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.

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.

![Figure 4: Elsevier's store data analytics](image)

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.
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.

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:

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

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)
   b. Elsevier’s Mobile App:
      i. Elsevier eLibrary (Singapore)
      ii. Elsevier NL (News)
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

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:

<table>
<thead>
<tr>
<th>Data Storage</th>
<th>SQL/NoSQL</th>
<th>Native</th>
<th>Hybrid</th>
<th>Android</th>
<th>iOS</th>
<th>Online/offline access</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSON</td>
<td>NoSQL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Both</td>
</tr>
<tr>
<td>SQLite</td>
<td>SQL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>Both</td>
</tr>
<tr>
<td>MySQL</td>
<td>SQL</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>Online</td>
</tr>
<tr>
<td>iCloud</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Online</td>
</tr>
<tr>
<td>Core Data</td>
<td>NoSQL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML5</td>
<td>NoSQL</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>Offline</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>App Features</th>
<th>Native</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics</td>
<td>Native APIs</td>
<td>HTML, Canvas, SVG</td>
</tr>
<tr>
<td>Performance</td>
<td>Fast</td>
<td>Slow</td>
</tr>
<tr>
<td>Native look and feel</td>
<td>Native</td>
<td>Emulated</td>
</tr>
<tr>
<td>Distribution</td>
<td>App Store</td>
<td>App Store</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Access</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Notifications</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Contacts, calendar</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offline storage</td>
<td>Secure file storage</td>
<td>Secure file system, shared SQL</td>
</tr>
<tr>
<td>Geolocation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
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.

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.

<table>
<thead>
<tr>
<th>Gestures</th>
<th>Yes</th>
<th>Yes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Swipe</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Pinch, spread</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Connectivity</td>
<td>Online and offline</td>
<td>Online and offline</td>
<td></td>
</tr>
<tr>
<td>Development skills</td>
<td>Objective-C, Java</td>
<td>HTML5, CSS, Javascript</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Native VS Hybrid Programming

Figure 7: Mark Zuckerberg interview in TechCrunch
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.
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/](http://www.elsevier.com/) and [http://store.elsevier.com/](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.
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.
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.

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.

Figure 13: Example of New Relic Report
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.

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.

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](https://marvelapp.com/)

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/](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

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.

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.

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.
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.

The features available in this version:

- Find list of all coupons
- Browse the coupons by its subjects
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
Figure 28: Coupon details page
Figure 29: Browse by subject with a favorite
Figure 30: Push notification received
Figure 31: Check a specific coupon code
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.

![Figure 32: Sorting ascending](image1.png)
![Figure 33: Sorting by number of offers](image2.png)

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

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

### 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 ........................................................................................................................................45

2. Methodology ....................................................................................................................................45

3. Project Description ............................................................................................................................46
   3.1. Background ..................................................................................................................................46
   3.2. Research Questions .......................................................................................................................46
   3.3. Research Sub-Questions ...............................................................................................................46
   3.4. Project Boundaries .......................................................................................................................47
   3.5. Objectives .....................................................................................................................................47
   3.6. Research and Reporting Methods .................................................................................................47

4. Project Charter ..................................................................................................................................47
   4.1. Current Situation / Problem Statement .........................................................................................47
   4.2. Project Justification / Goal Statement .........................................................................................47
   4.3. Project Product / Scope In or Out .................................................................................................48
   4.4. Constraints .....................................................................................................................................48
   4.5. Risk and Solutions .......................................................................................................................48
   4.7. Stakeholders ...................................................................................................................................49
      4.7.1. Supervisor .................................................................................................................................49
      4.7.2. Organization ..............................................................................................................................50
1. 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.

Elsevier BV is a part of the Reed Elsevier group, an Anglo-Dutch multinational 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
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.

<table>
<thead>
<tr>
<th>Scope In</th>
<th>Scope Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Minutes meeting</td>
</tr>
<tr>
<td>User manual</td>
<td>Training</td>
</tr>
<tr>
<td>Developer manual</td>
<td>[optional]</td>
</tr>
<tr>
<td>Project plan</td>
<td></td>
</tr>
<tr>
<td>Requirements document</td>
<td>[optional]</td>
</tr>
<tr>
<td>Design document</td>
<td>[optional]</td>
</tr>
<tr>
<td>Test plan [optional]</td>
<td></td>
</tr>
<tr>
<td>Project report</td>
<td></td>
</tr>
<tr>
<td>Presentations</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>No.</th>
<th>Risk</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Miscommunication</td>
<td>• Weekly meeting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reported problem on the documentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Review</td>
</tr>
<tr>
<td>2.</td>
<td>Lack of technical knowledge</td>
<td>• Consult expert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Research technical solutions (internet and media source)</td>
</tr>
<tr>
<td>3.</td>
<td>Errors and Bugs</td>
<td>• Debug by testing software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Consult test engineers</td>
</tr>
<tr>
<td>4.</td>
<td>Insufficient working time</td>
<td>• Extra working hour before normal arrival</td>
</tr>
</tbody>
</table>
f. Project Phase and Time

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

<table>
<thead>
<tr>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>i. Supervisor</strong></td>
</tr>
<tr>
<td><strong>Company</strong></td>
</tr>
<tr>
<td>Name : Maxime Corbeau</td>
</tr>
<tr>
<td>Position : Web Analytics and SEO Manager</td>
</tr>
<tr>
<td>E-mail : <a href="mailto:m.corbeau@elsevier.com">m.corbeau@elsevier.com</a></td>
</tr>
</tbody>
</table>
ii. Organization

- Information

<table>
<thead>
<tr>
<th></th>
<th>David (Company Supervisor)</th>
<th>Max (Manager Web Analytics &amp; SEO)</th>
<th>Alia</th>
<th>Fabian</th>
<th>Teodor</th>
<th>Mrs. Veugen (University Supervisor)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Delivery</td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Requirements</td>
<td>C</td>
<td>C</td>
<td>I</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>IT</td>
<td>R</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td>R</td>
<td>I</td>
<td>I</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>C</td>
<td>C</td>
<td>R</td>
<td>C</td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>Development</td>
<td>I</td>
<td>R</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>Ap</td>
</tr>
<tr>
<td>Project Planning</td>
<td>I</td>
<td>R</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>Ap</td>
</tr>
<tr>
<td>Weekly Update Meetings</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

✓ Goals for the Better Elsevier’s Web Store

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

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

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

*Table 5: Acquisition strategy*

#### Referral / Virality

<table>
<thead>
<tr>
<th>Idea</th>
<th>Example app/company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorporate viral mechanisms with core functionality</td>
<td></td>
</tr>
<tr>
<td>Attract network of new users</td>
<td></td>
</tr>
<tr>
<td>Engage the users to invite friends</td>
<td></td>
</tr>
<tr>
<td>No-nonsense interface</td>
<td>Clear</td>
</tr>
<tr>
<td>Unique</td>
<td></td>
</tr>
<tr>
<td>Provide true mobility, ability to run offline mode</td>
<td></td>
</tr>
<tr>
<td>Provide help / FAQ page</td>
<td></td>
</tr>
</tbody>
</table>

*Table 6: Referral or Virality strategy*
## Retention

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

Table 7: Retention strategy

## Revenue

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

Table 8: Revenue strategy
<table>
<thead>
<tr>
<th>Idea</th>
<th>Example app/company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consider small markets, such as Asia, Latin America</td>
<td></td>
</tr>
<tr>
<td>Marketing is key, and in advance</td>
<td></td>
</tr>
<tr>
<td>Articles in TechCrunch</td>
<td>Instagram</td>
</tr>
<tr>
<td>Closed beta</td>
<td>Evernote, Clear</td>
</tr>
<tr>
<td>In all app store launches on Day One</td>
<td>Evernote</td>
</tr>
<tr>
<td>Many options of the books, and those books are not available in other stores</td>
<td>Kindle Amazon</td>
</tr>
<tr>
<td>Video demo in Vimeo</td>
<td>Clear</td>
</tr>
<tr>
<td>Platform appropriate, compatible and functional for all brands and devices</td>
<td></td>
</tr>
<tr>
<td>Easy to install, access, and use</td>
<td></td>
</tr>
<tr>
<td>Spontaneous -&gt; accessible for short bursts of user activity</td>
<td></td>
</tr>
<tr>
<td>Huge launches based on intense marketing campaigns</td>
<td></td>
</tr>
<tr>
<td>Appeal to almost everybody</td>
<td></td>
</tr>
<tr>
<td>Pay attention to UI Guidelines IOS / Android</td>
<td></td>
</tr>
<tr>
<td>Make it run offline</td>
<td></td>
</tr>
</tbody>
</table>

*Table 9: Activation strategy*
## Appendix D: Minimum Viable Product

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

Figure 37: First Prototype and Screen Flow
Figure 38: Homepage screen flow

Figure 39: Browse by Subject screen flow
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

Figure 47: All coupons list

Figure 46: All coupons filtered

Figure 45: Subject list sorted ascending

Figure 44: Subject list sorted descending

Figure 43: Subject list sorted by number of offers
Figure 51: Notification options for future offers when first time making a subject as favorite

Figure 49: Subject list with some favorites subjects

Figure 50: Favorite page

Figure 52: List of coupon of a specific subject

Figure 53: Coupon details page

Figure 54: Put e-mail options when clicking 'email me' button
Figure 55: Check a specific coupon code page

Figure 56: List of coupons given after checking the code

Figure 57: Push notification received