



**CHALMERS**

---

# CONSATSAT FINDER

## CONSATSAT FINDER

A Service to Simplify the Consultant Recruitment Process

Bachelor's Thesis in Computer Science and Engineering

RITHIKA KABILAN

---

Department of Computer Science and Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Gothenburg, Sweden 2019



BACHELOR'S THESIS

## **Consat Finder**

A Service to Simplify the Consultant Recruitment Process

**RITHIKA KABILAN**

Department of Computer Science and Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Gothenburg, Sweden 2019

## **Consat Finder**

A Service to Simplify the Consultant Recruitment Process

RITHIKA KABILAN

© RITHIKA KABILAN, 2019

Examiner: JONAS DUREGÅRD

Department of Computer Science and Engineering

Chalmers University of Technology / University of Gothenburg

SE-412 96 Göteborg

Sweden

Telephone: +46 (0)31-772 1000

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Cover:

Consat Finder application logo.

Department of Computer Science and Engineering

Göteborg, Sweden 2019

## Foreword

The bachelor's thesis is part of the Computer Science program (180 HP) at Chalmers University of Technology and is comprised of 15 HP.

The work has been conducted at Consat Engineering AB's office located at Lindholmen, Gothenburg. I would like to thank Consat Engineering AB and its employees for the inviting and inspiring environment. I would also like to aim a big thank you to my supervisors for giving me the opportunity to do my thesis with Consat and for their constant support:

- ❖ Titi Oyedeji, Business Area Manager IT & Mobilitet
- ❖ Jonas Williamsson, In-House Projects
- ❖ Robin Grönlund, Consat Engineering AB
- ❖ Annika Brännström, Consat Engineering AB

I would also like thank my Chalmers supervisor, Sakib Sisteck, for his constant support and advise throughout the course of the project.

## **Consat Finder**

A Service to Simplify the Consultant Recruitment Process

RITHIKA KABILAN

*Department of Computer Science and Engineering, Chalmers University of Technology*

Bachelor's Thesis

### **ABSTRACT**

In today's corporate world, consultants are sought after by many companies. Clients must exert a lot of time and efforts in order to find a potential consultant. The current consultancy recruitment process is time consuming and exhausting, this thesis addresses the simplification of this process through the use of an application called Consat Finder. It is a user friendly mobile app for the clients to use for searching and browsing consultant profiles matching their preferences. Clients can book an interview and ask questions about a consultant they are interested in. As a result of this thesis, a functional frontend-based version of Consat Finder has been developed and delivered to Consat who are satisfied with the outcome and will continue its further developments.

**Keywords:** clients, consultants, Consat, CV, profiles, recruitment process

## Abbreviations

API – Application Programming Interface

HCI – Human-Computer Interaction

IDE – Integrated Development Environment

MVP – Minimum Viable Product

OS – Operating System

PO – Product Owner

RFP – Request for Proposal

UI – User Interface

UX – User Experience

## Table of Contents

<b>Foreword .....</b>	<b>iv</b>
<b>1. Introduction .....</b>	<b>1</b>
1.1. Background.....	1
1.2. Purpose.....	1
1.3. Goal.....	1
1.4. Scope.....	2
<b>2. Technical Background .....</b>	<b>3</b>
2.1. The Current Consultant Recruitment Process .....	3
2.2. Requirements Engineering.....	3
2.3. Prototyping.....	5
2.4. React Native.....	5
2.5. User Experience.....	6
2.6. Scrum.....	6
<b>3. Method .....</b>	<b>8</b>
3.1. Pilot Study.....	8
3.2. Scrum and Planning.....	8
3.3. Work Environment.....	9
3.4. Prototype.....	9
3.5. Build Application.....	9
3.6. User Testing.....	10
<b>4. System Construction.....</b>	<b>11</b>
4.1. Establishing the Environment .....	11
4.2. Requirements Gathering .....	11
4.3. Prototyping.....	14
4.4. Base Wireframe.....	23
4.5. Browsing Screen .....	25
4.6. Read CV Screen.....	28

4.7.	<i>Navigating Between Screens</i> .....	30
4.8.	<i>Filter Feature</i> .....	32
4.9.	<i>Interactable Settings Menu</i> .....	36
4.10.	<i>Profile Counter</i> .....	37
<b>5.</b>	<b>Result</b> .....	<b>38</b>
<b>6.</b>	<b>Conclusion</b> .....	<b>39</b>
6.1.	<i>Other Solutions</i> .....	41
6.2.	<i>Environmental and Ethical Aspects</i> .....	41
6.3.	<i>Suggestions for Further Improvements</i> .....	42
	<b>References</b> .....	<b>44</b>
	<b>Appendix A</b> .....	<b>47</b>

# **1. Introduction**

## **1.1. Background**

Since its foundation in 1985, Consat Engineering AB has been branded as a mechanical company. Throughout the years, software development has slowly become an integral part of Consat and is something that needs to be clarified to their clients. Consat's aim is to refurbish and strengthen their brand as a software company. In order to do so, they devised an idea where software development is integrated with client consultancy.

## **1.2. Purpose**

Today's situation requires clients to call Consat, or any other consultant agency, and have a deep dialogue about what kind of consultants they wish to hire. This is quite taxing and takes up a lot of time on Consat's part. One way to make this process more effective, as well as promote Consat's software capabilities, is to create an application where clients are able to browse through the Consat consultants' profiles and pick a consultant who meets their needs.

## **1.3. Goal**

The goal was to create the frontend base of the Consat Finder application for Consat. Consat looked for a user friendly, mobile app for their clients to use for searching and browsing consultant profiles matching the clients' selected preferences or filters. For the above-mentioned goal, Consat further defined expected criteria for design, build of a functional prototype. A fully functional application (Consat Finder) would encompass the full stack of backend, application logic, APIs and the user interface design. However, as stated above this project's goal is to build the frontend for Consat Finder. Given the primary focus on user friendliness, user satisfaction, Consat expected the use of human-computer interaction design processes to be used. Thus the primary focus for Consat Finder is the front end design following human computer interaction design processes. The MVP application thus uses a pre-existing database or a temporary data construct as the backend to hold dummy test data.

## 1.4. Scope

The initial scope for the Consat Finder project was to develop an application that confirmed to the MVP that covered the following:

- ❖ Focus on the end clients interaction with the proposed mobile application considering the user experience, user friendliness
- ❖ Ensure that the mobile app used Consat's brand color theme and logo
- ❖ While the mobile app was expected to be aesthetically pleasing, key software application performance requirements are also to be considered. That is, maintain a balance between light weight front end design architecture and the response times.
- ❖ The backend database is to be designed to store the key aspects of a consultants profiles , namely key technical skills , short description. The key features would be used to filter and browse. The short CV description would be used as the teaser in the display
- ❖ Clients who will use the mobile app should be able to login using valid login credentials. However the prototype focuses on the core functionality but has the possibility for Consat to extend, integrate with their existing user login and validation application for a seamless user experience. Thus, only the wireframe (frontend) design for the login functionality is in scope.
- ❖ Clients should be able to filter the consultant profiles based on their preferences and the Client can bookmark as 'favorite' a profile to review later.

## **2. Technical Background**

### **2.1. The Current Consultant Recruitment Process**

What is a consultant? A consultant is someone who can offer their knowledge or expert advice within a certain field to their client [1]. Clients can either be individuals or organizations. Clients predominantly hire consultants for their knowledge on areas outside the clients' scope of expertise, however there are other reasons as well. Consultants may also offer an object view on a project, give unbiased feedback or simply work as a set of extra hands [2].

Unlike a regular recruitment process where companies simply post a job advertisement on their home page or on LinkedIn, when in need of a consultant they either need to reach out to the consulting company/agency themselves or use a middleman agency to find a consultant on their behalf [3]. When a firm or individual is contacted, the client informs them of the task as well as lists the capabilities they are looking for in the consultant through the use of a RFQ. Interested parties are expected to submit a proposal consisting of their qualifications, fee and contact information. The client will then filter through the proposals and interview the most promising ones, after which a consultant will be chosen [2]. It should be noted that this written procedure is more of a guideline than a rule. Clients may do things differently, such as interview all the applicants before filtering and choosing a consultant.

### **2.2. Requirements Engineering**

In order to know the purpose of as well as what services should be provided by the application, information regarding the application must be gathered and processed thoroughly [4]. Requirements within a set scope are found through an iterative process of specification, validation and elicitation.

#### **2.2.1. Requirement Specification**

The first step is to specify the scope of the project and the stakeholder's (the PO) requirements, thereafter specifying user requirements [5]. User requirements depict the services provided by the application and are created during the elicitation phase. Requirement specification entails elaborating on user requirements [4], such as using natural language or defining the scope if it is

yet to be set [6]. Natural language is an umbrella term for techniques used to specify system and user requirements by issuing informal descriptions of the requirements from the users' standpoint, not the application's. While there are many natural language styles, the only relevant style is the one called 'User Stories'. A user story is a short description of a feature or function of the application from the user's perspective [7]. The general syntax for creating user stories is:

*“ As an X, I want Y so that Z. ”*

Where X stands for the type of user (e.g. 'As a student'), Y represents the feature or task to be performed (e.g. 'I want to register myself for an exam') and Z states the purpose of Y (e.g. 'so that I am eligible to take the exam').

### 2.2.2. Requirement Validation

It is a process where the requirements are examined to find improvements and ensure they align with the project's objectives, both from the user's perspective as well as the PO's proposals [8]. This is to correct conflicting and flawed requirements in conjunction with maintaining realistic and verifiable requirements. This can be achieved by conducting a feasibility check early on in the iterative process and then prototyping and testing requirements as the iteration increments. A feasibility check reflects the project's value from different perspectives and sets an acceptable and achievable MVP [9]. There are four feasibility types:

- I. Technical Feasibility: what technical expertise and technologies are required?
- II. Schedule Feasibility: what can be achieved during the project's time constraint?
- III. Economic Feasibility: what is the project's budget?
- IV. Operational Feasibility: what do the stakeholders value? Will it work?

Prototyping is used to demonstrate a model of the final product to the users and stakeholders to validate that it conforms to their needs [10]. It is usually a quick design of the application and its requirements and should be easily modifiable. Requirements can be tested by running user tests, i.e having the users interact with a prototype following a given list of instructions. Their interaction and reception of the prototype will be closely monitored and then compared to the developer's assumptions and expectations on the user's feedback.

### 2.2.3. Requirement Elicitation

While there are many requirement elicitation techniques available for use, there is only one which pertains to this thesis: use cases [5]. Use cases describe interactions between the application and

the user, sometimes even external systems if those are involved. They are modeled using use case diagrams to describe the application's system. Requirements are extracted by brainstorming and deconstructing application features within the chosen scope.

### **2.3. Prototyping**

Before an application is constructed it must be designed and tested to check its technical feasibility, much like how architects create sketches and miniature models before constructing a building. This is achieved by creating prototypes of the application, that is to say, the experimental manifestation of incomplete designs of the application which can be interacted with [11]. Prototypes can be as simple as a paper sketch or as complex as a three-dimensional mockups and wireframe software.

Prototyping, also known as the process of developing a prototype, is an essential part of the iterative user-centered design procedure [11]. A user-centered design procedure is where the users are involved in the design process [12]. The users test a prototype, evaluate it and give feedback so that further improvements can be made, thereafter which the redesigned prototype is user tested again. Using this method, the application's purpose becomes clearer, its layout more refined and not to mention new ideas can be implemented.

Prototyping can be divided into two categories: low-fidelity prototyping and high-fidelity prototyping [11]. Low-fidelity prototypes lack functionality and often times do not resemble the final product since they are usually simple sketches or cardboard designs. However they are simple and quick to design and modify, which is an important part of developing the conceptual design where ideas are being formed. High-fidelity prototypes almost mirror the final product and are quite functional to the point where users can perform tasks on them as if they were the real product. They most often are computerized and interactable wireframes made using existing software, such as Adobe XD. However, creating and modifying a high-fidelity prototype takes a longer time than a simple low-fidelity prototype.

### **2.4. React Native**

React Native is defined as a JavaScript framework used to develop mobile applications for Android and iOS devices [13]. In other words, a software library written in JavaScript used to provide

structure to the development of mobile applications. React Native is built upon React, a framework developed by Facebook [14]. However React is used to produce UI for web browsers whereas React Native is used to create applications for mobile platforms.

React Native applications can be simulated using two different OS environments: Android or iOS [13]. An Android simulation requires Android Studios to be downloaded and can be set up on any computer whereas an iOS simulation can only be done using Xcode, which is a development tool exclusive to Apple Mac computers. Re-rendering of screens within the application occur when the value of either `props` [15] or `state` [16] (two data types which control a component, the former immutable and the latter mutable) changes [17]. React Native is integrated with strong developer and debugging tools to enable quick reloadings of the project while changes are being made. However, it is important to note that React Native is a relatively young project – it was first launched in 2015 – and therefore still has minor bugs and lacks a few key functionalities found in modern day mobile operating systems [18]. Most of the missing functionalities can be accosted for by implementing support from various open source APIs.

## 2.5. User Experience

User experience refers to how the end-user feels when interacting with a product, taking into account their satisfaction levels, overall emotions upon using the product as well as the usability of the product [12]. Usability of a product is reflected in the quality and structure of the UI, taking into account the learnability of the product, how efficient it is and if it is pleasurable to use [19]. The product should also be aesthetically pleasing and not annoy the user. The product should be satisfactory and easy to use, in other words meeting the needs of the user without compromising the quality of the product.

## 2.6. Scrum

Scrum [20] is the technique of processing a big project into several smaller components iteratively and is primarily used to oversee the development of a project or product. The main lifecycle of Scrum can be condensed to the following:

- I. Establish a Product Backlog, i.e list of items that need to be completed within the project [21].

- II. Sprint Planning – plan Sprints (a set period of time within which the assigned work must be completed [22]) within the selected scope of the project.
- III. Conduct the work assigned to each Sprint.
- IV. Reflect upon the outcome at the end of each Sprint and adapt the Product Backlog and upcoming Sprints accordingly, a so-called Sprint Review [23].
- V. Repeat steps II – IV until the project requirements are satisfied.

The outcome of each sprint should result in some feature of the product being ready or near completion. If the product is analogous with a cake, the idea is to create one slice of cake (which is composed of several different layers) rather than to create one layer of the cake at a time [24]. Each 'layer' corresponds to a level of the product whereas a 'slice' is equivalent to a feature. For example, if the 'cake' is a car parking ticket application, some layers of that application could include databases, UI and security. Whereas a slice of the application could be the registering the car's number plate upon entry into the garage. A slice of cake has more value to the end-user than individual layers of a cake that has yet to be assembled. The reason Scrum is designed in such a fashion is to account for time delays in the event the project is not completed in time, in which case the team at least has a couple of functional features to demonstrate to the end-user or client.

There are three main roles associated with Scrum:

- I. The PO's role is to manage the product backlog so that the development team know what is to be produced and can achieve the desired outcome [19].
- II. The Scrum Master is responsible for ensuring the team delivers according to the agile values and adhere to the practices agreed upon by the team.
- III. The Development Team are the people who deliver the tasks assigned to each sprint. Their main duty is to assure that the outcome of the sprint is valuable to the product.

### **3. Method**

Prior to the start of the project, an estimated plan was constructed as to have a guideline on how the thesis was expected to proceed. The plan included various project management methods, expectations, predictions and presets.

#### **3.1. Pilot Study**

In order to evaluate the amount of work required to construct the application, some research on JavaScript and React Native will be required. Only after gaining a thorough understanding of the programming language and environment can accurate estimates be made. Since both JavaScript and React Native are subjects which offer a wide spectrum of knowledge, the scope of the research will need to be narrowed to only the basics.

#### **3.2. Scrum and Planning**

The first step of Scrum will be to set a Product Backlog, followed by a Sprint Backlog. A Product Backlog will be created based on the requirements discerned from a discussion with Consat. A simplified Sprint Planning will result in a Gantt schedule consisting of multiple Product Backlog items with various Sprint lengths, some lasting a week and half while others are longer. The schedule will be planned in such a manner that certain items could run in parallel, such as reading theory while constructing a certain feature of the application, while others could be conducted iteratively throughout the project time span, such as user testing and prototype redesigning. Each sprint will deal with one main feature of the application. The idea will be to build one feature at a time rather than to work on a little of everything at the same time. This way each Sprint will have an operational feature as an outcome, which has a higher value to the client since it is interactable and a part of the final product. It is planned that user testing, reflecting and redesigning are tasks that will repeat iteratively every few Sprints or so as a way to receive feedback on the application design and flow.

There are several different roles associated with Scrum which are usually divided amongst the group members, however this concept will not be applied to this particular project on the grounds that this team consists of one person. Instead, Consat will take the role of PO while the remaining two roles, the Scrum Master and the Delivery Team, will be merged into a single role.

### **3.3. Work Environment**

In preparation for prototyping and building the application, certain tools will have to be downloaded and set up. Since the initial plan is to create a cross-platform application, the programming will be conducted on a Macbook. According to the Scrum plan, the first step will be to create a prototype of the application. Once the prototype has been finalized and user tested, the application can be built. The prototype will be designed on Adobe XD while the application will either be built using Visual Studio Code (VS Code) and simulated using Xcode.

### **3.4. Prototype**

The project will start off by making paper prototypes which encompass the MVP and the extra features. This will be done in order to stimulate creative designs in a time-efficient manner as well as to gain an overview of what the application could look like. Once a satisfactory paper prototype has been made, it will be recreated digitally on Adobe XD with color and screen transitions.

### **3.5. Build Application**

The outcome of the project will be an application called 'Consat Finder' which is a fully functional mobile application with the basic features mentioned in the MVP. The user should be able to browse through various anonymous profiles and have the option to view a profile in detail before deciding whether to pursue further actions with the chosen profile. Using the VS Code IDE, the application will be built using a framework called React Native where the programming language will be JavaScript.

### **3.6. User Testing**

The purpose of the project is to develop an application which simplifies a currently arduous routine. Therefore it is only befitting that user tests will be conducted at regular intervals to measure a user's satisfaction with the application. User feedback will allow the application to be evaluated from multiple non-biased perspectives, thereby encouraging reflections and eventual redesigns to the application.

## **4. System Construction**

### **4.1. Establishing the Environment**

The first step was to decide what tools would be used to build Consat Finder on. The type of tool used to build an application is a deciding factor since the final product varies depending upon what environment it was produced in. In this particular case, it was between creating a pure Java application from scratch or using the React Native framework. While developing a Java application from scratch would be a massive task, it was something I was knowledgeable in and had worked with before. In contrast, I was unfamiliar with JavaScript and had only heard of React Native in passing. That being said, React Native's built-in functions, support and good reviews ultimately led to it being chosen as the development environment for the application. Another reason was that React Native was ideal for creating cross-platform applications (iOS and Android) using the same project. A Java-based application would have required two separate projects, each accommodating one of the operating systems.

### **4.2. Requirements Gathering**

Before the application could be built, it was important to understand the purpose and objectives of the project and to know exactly what was to be built. A thorough discussion with the head supervisor at Consat was held at the start of the project to review their expectations and visions for the application. Fig.1 depicts the requirements engineering process implemented in this project. It is an iterative process where requirements are added, modified and removed throughout the course of a project, however in this particular project a decision was made that there would be no change in the requirements after the prototyping phase.

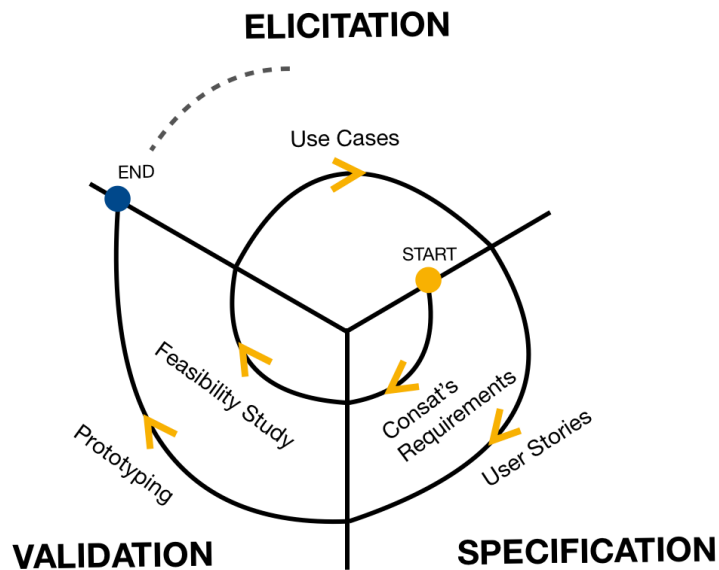


Fig.1 Requirements Engineering Process of Finding and Gathering Requirements

#### 4.2.1. Consat's Vision

The vision for the fully developed and complete Consat Finder is for it to represent Consat and simplify the consultant hiring process for both them as well as their clients. It would feature a database of profiles belonging to Consat consultants which the clients can browse through and read.

Inspired by Tinder, the browsing feature allows the client to flip through summarized versions of the consultants CVs which contain nothing more than their main capabilities and a few lines of text. Should the client wish to read the CV in depth, they simply select the summarized profile to read the full-length CV. From here on, I shall refer to the actual CV as 'CV' and the summarized CV as a 'profile'. When reading a CV, the client should then be able to either book an interview with the consultant or be able to message an inquiry. One of Consat's requirements was that the consultants should be anonymous, meaning that only their capabilities, work experience and similar topics may appear in their Consat Finder CVs.

It would also be possible for the client to filter the profiles based on what capabilities they are searching, e.g. if they are searching for a senior Java expert there should be a filter options which allows the client to view profiles matching the queried description. Clients would also have the possibility to personalize their filter searches by creating their own 'favorite filters' list in their own profile pages.

The application will be exclusive in the sense that only Consat's clients may use the application. Upon downloading the application, the client will be taken to a login page. The first login page requires the client to enter a pin code they would have received from Consat. Thereafter they can proceed to confirm their account and set up their profile page. When confirming an account, the client will need to check that the information stated is correct. This autofill function will be made possible due to the fact each code is personal. Therefore information such as their name, contact information and the organization or company they are associated with will already be in the system.

The consultants themselves will not have access to their profiles or CVs displayed on Consat Finder, only the admin will be allowed to make direct changes. If the consultant needs to update their CV, they will need to either send an updated CV to the admin or update their LinkedIn profile. The database from which the CV data will be retrieved will either be LinkedIn or a custom file-based system.

The admin will be able to control which profiles are visible to the clients and which are not. As soon as a consultant is booked, their profile is taken off the browsing shelf until they are available again. The admin will also be able to control which clients have access to the application.

#### **4.2.2. Feasibility**

It became quickly apparent that Consat Finder could not be fully developed and ready to be launched within the allocated time frame of three months, hence a scope needed to be set. There were several aspects that needed to be taken into consideration when setting the scope, namely the technical, schedule and operational feasibility types. Since the time frame for the project was three months, the time frame to not only build a version of the application which had some value to the stakeholders (Consat), but to also make prototypes and do user tests was a little less than that. Moreover, I also had to learn JavaScript from the ground up. Consequently, the scope set included only the core functions of the application:

- ❖ A browsing view – browse through the available profiles.
- ❖ Read a CV – view a profile in depth by reading the associated CV.
- ❖ Select a consultant – either book an interview or write a message after reading a CV that satisfies the user's requirements.

From this point onwards, a 'user' is someone who interacts with the application or the prototypes, regardless if they are a client or an admin. The MVP for this project was set to the last three functions from the list of core functions mentioned above. This is since they were more valuable to Consat and its clients than the login feature which can easily be implemented from an open source API. If there was to be time over, then the filter feature was to be implemented.

#### **4.2.3. Use Cases & User Stories**

Upon further elicitation for requirements through brainstorming sessions and breaking down the core functions, another feature was discovered. If the client favors a few profiles and wishes to view and compare them after browsing through all the other profiles, the client can simply favorite a profile to view later. Each profile will have a star icon which can be activated and deactivated. Once activated, the profile can be viewed in the favorites tab. This feature was to be implemented only if there was time left over.

After compiling the requirements found so far, they were simplified into use cases to provide a more structured and explicit overview of what was to be included in Consat Finder. The use cases for the Consat Finder project are noted below:

- ❖ Browse through anonymous profiles.
- ❖ Filter the profiles.
- ❖ Favorite profiles to view later.
- ❖ Read the CV of an anonymous profile.
- ❖ Book an interview with a consultant.
- ❖ Write an inquiry about a consultant.

The use cases were also used to write user stories in order to elicit more requirements from the user's perspective, as seen in Appendix A, however there was no new requirement as an outcome. It did however offer a concrete explanation on the purpose of each use case.

### **4.3. Prototyping**

The final step before building the Consat Finder application was to create prototypes and have users test them. Users would interact with them by following a given set of test situations and are asked for feedback.

### 4.3.1. Low-Fidelity Prototyping

The first prototype made was a quick sketch outlining a general idea of what the different screens in the application could look like, as seen in fig.2. Its purpose was to confirm with Consat that the ideas and features had been communicated correctly between us. Small adjustments were made, such as the layout of the filter options changed from being 'a list of filter options' to 'clickable filter option buttons'. Another change was the layout of the browsing screen. Instead of using left and right arrows to flip through the profiles, just like in Tinder, the client just needs to swipe left and right. Nonetheless, unlike Tinder the profiles do not disappear if the client swipes to the left, it is simply a carousel effect.

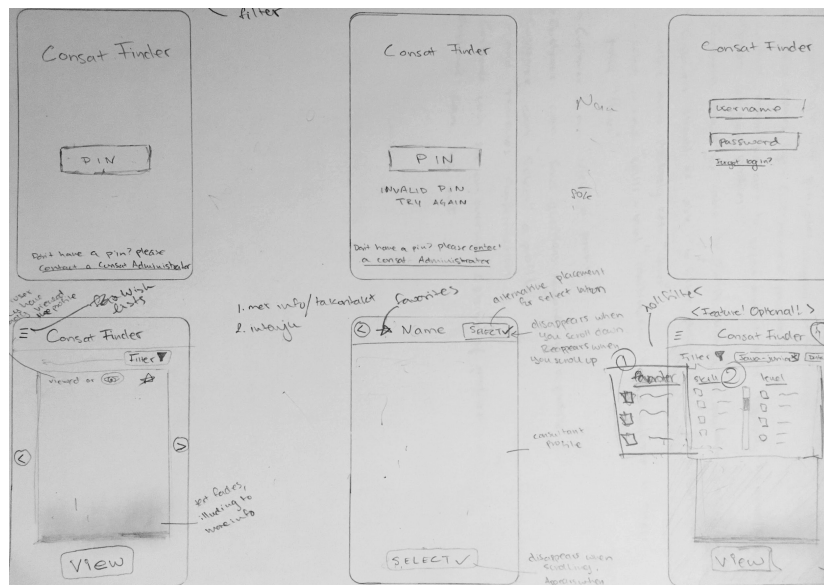


Fig.2 Paper Prototype of Consat Finder

### 4.3.2. High-Fidelity Prototyping

The prototypes designed with Adobe XD offered an authentic experience of what the actual application would feel like. Other than being able to flip through the profiles, the prototype was interatable and reacted to the users' inputs. There were a total of three iterations with the high-fidelity prototypes where they were user tested and redesigned as a result of the feedback.

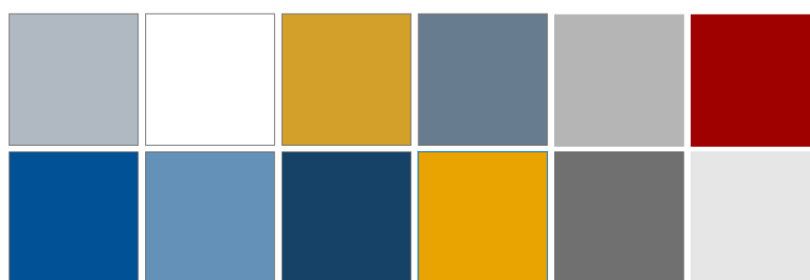
#### 4.3.2.1. User Testing Procedure

Within each iteration of the prototype, user tests were conducted on users with various backgrounds. Some were college students within different programs who have no programming knowledge while others had years of experience within the field. The reason for this was that the application is supposed to cater to a wide demographic. Depending upon the iteration, the number of users who participated during the user tests varied between six to eight users. This was due to scheduling conflicts since not all users could participate in each test.

The users were instructed to conduct a series of tasks, such as ‘find a consultant who knows CAD’. They were also asked for live feedback while they were performing the various tasks required of them. Once all the tasks were completed they were then asked to fill out a survey which contained both open and closed questions, in other words they would first rank certain aspects on a scale of one to five and then answer open-ended questions.

#### 4.3.2.2. First Iteration

Before the wireframes were built, a color scheme had to be decided. Since Consat’s representative colours are white and shades of blue, those would be the base colors for the applications to promote Consat’s presence. A contrast color was necessary in order to highlight or outline important items, hence a couple shades of orange were included since orange is the contrast color to blue. The complete color palette can be seen in fig.3.

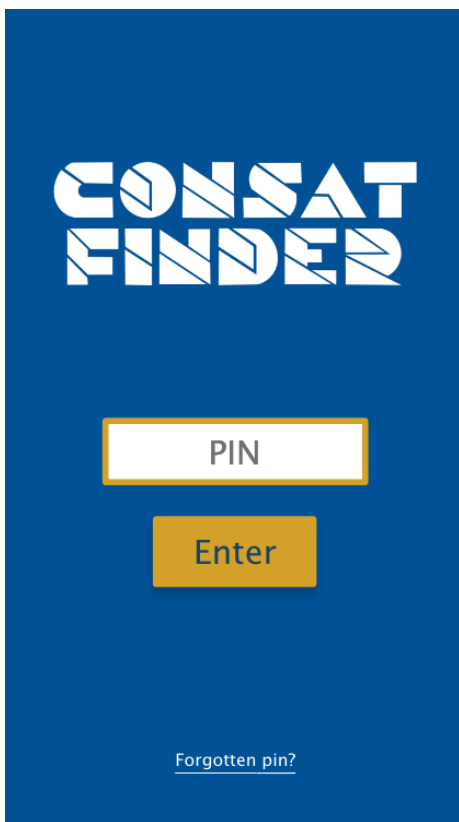


*Fig.3 Color Palette for Consat Finder, First Iteration*

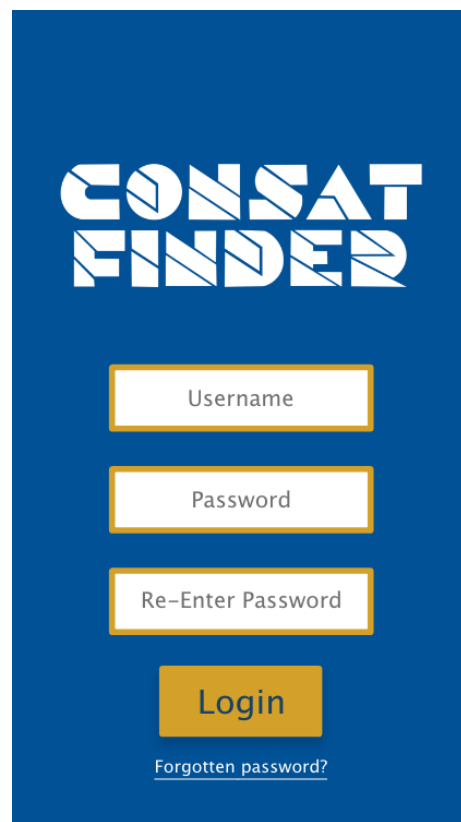
The wireframes were constructed next, a light blueprint of the different views that were to be designed. Once the rough outline was complete, the logo and icons were created using the Pages application on Mac. The icons themselves (e.g. user profile, filter and setting icons) were relatively simple to create, however it took a while to create the Consat Finder logo. Although there exists a logo and typography for ‘Consat’, the ‘Finder’ part had to be designed. After studying and

experimenting the art style used to design the official Consat logo, I was able to create the 'Finder' part. Even though there is an official Consat logo, there was not one in white with a transparent background available. Hence, I had to design that as well before assembling all the components to form the Consat Finder logo.

The first screens to be completed were the login screens for entering the pin code for first time users as well as logging in as regular users, see fig.4 and fig.5. The colors were experimented with quite a bit by testing multiple shades, not only on these screens but on the others as well, until a visually pleasing combination was constructed.

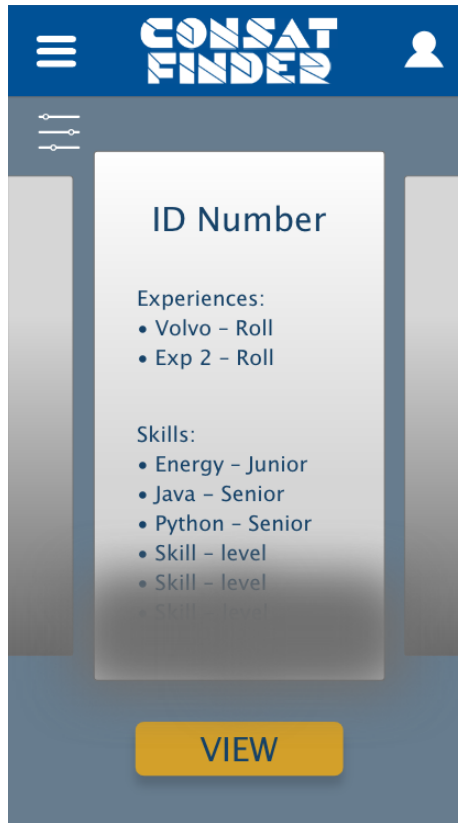


*Fig. 4 Pin Code Screen – Prototype*



*Fig.5 Returning User Login Screen – Prototype*

The browsing screen was initially designed with the idea that the text becomes blurred and the profile background gradually becomes darker at the foot of each profile, see fig. 6. Furthermore, the Consat Finder logo was to be placed at the center of the header, with the user profile button to the right and a menu button to the left. Due to lack of space in the header, the filter button had to be positioned below the menu button, outside the header. While I was aware that this was not a feasible solution, it had to do for the time being.



*Fig.6 Browsing Screen, First Iteration – Prototype*

When the filter button is pressed, it triggers a pop-up menu to appear where the user can specify the the period in which they require a consultant and can select filter options (see fig.7). Once a filter option is selected, another small pop-up menu materializes as it asks the user to specify what level of experience of the chosen capability they seek (see fig. 8).

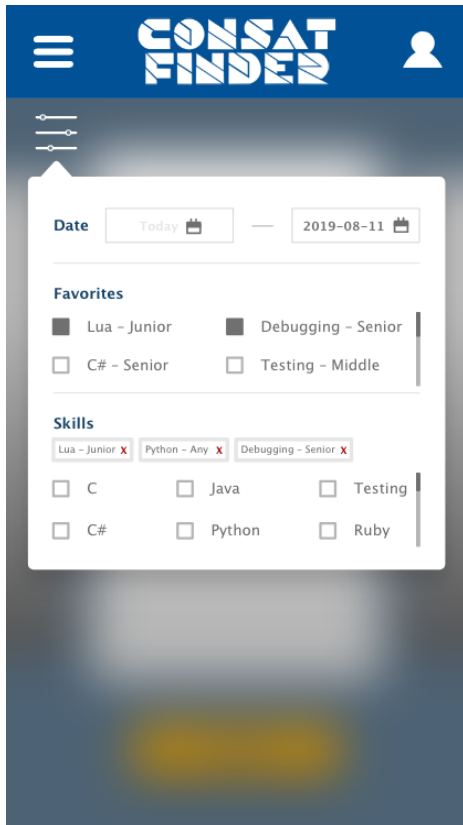


Fig.7 Filter Feature – Prototype

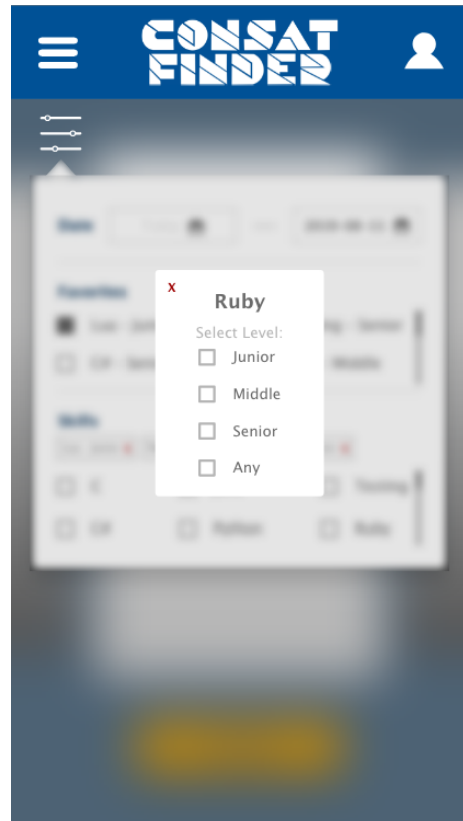
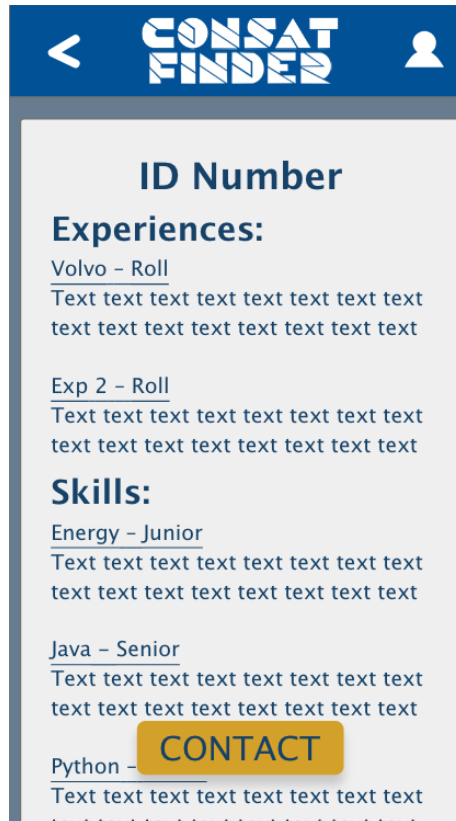


Fig.8 Filter Option Experience Level –

Prototype

The final screen to be constructed was the 'read CV' screen. Its header construction was similar to that of the browsing screen, except there is a back button replacing the menu icon on the left side (see fig.9). The user could scroll through the CV to read it and they had the option to press the 'Select' button, which would trigger a pop-up where the user is given three alternatives: book an interview, ask a question or cancel.



*Fig.9 Read CV Screen – Prototype*

The user testing revealed that many users could easily navigate themselves on the screen and interpreted the different views quickly. As I suspected, there were many who felt unsettled by the filter icon's position. One user pointed out that adding a gradient to the profiles makes it harder to read the text.

#### 4.3.2.3. Second Iteration

As a result of the feedback received from the users in the previous iteration, the browsing screen and some parts of the read CV screen were redesigned.

The header was redesigned so that the filter icon replaced the menu button and a settings icon replaced the user profile button (see fig.10). The menu and user profile buttons were removed because they were outside the scope and were therefore irrelevant to the project. A settings icon was placed in the right corner to substitute the lack of a menu icon. All applications must have a settings menu and since the menu button was removed, it had to be implemented some other way. However, since it is not part of the scope, it will remain non interactable.



Fig.10: Browsing Screen, Second Iteration – Prototype

Another addition was the bottom navigation bar. This served as a more modern approach to navigating between different screens (i.e the browsing, messaging, favorite and review screens) rather than having to first press a menu button to open a hamburger menu and then navigate to a screen. It was simpler and more aesthetically pleasing. The 'profile' icon represented the browsing screen and was the default screen. The 'starred' icon was connected to the favorites screen where users can browse through their personal favorites. Only if there was time left would this screen be developed further. The 'mail' and 'review' icons were connected to the messages screen and the review screen respectively. In the messages screen users would view the messages they sent and received concerning consultants. The review screen was there for the users to be able to review the consultants they selected previously. However it is important to note that neither of these last two screens were further developed in project since they were outside the project scope.

The profiles' background color was changed to its original one-tone color, the gradient now discarded in favor of a sleek look and being able to read. The original idea of blurring the words was also dropped.

The second round of user testing went smoother and faster than the first one since it had the users had done the previous one as well. They were happy with that the filter icon was moved to the header. Some commented that the browsing screen looks much more realistic now that the bottom navigation bar had been added.

The background color of the browsing screen was changed because it looked more grey than blue, and the color scheme was supposed to represent Consat's characteristic colors.

#### 4.3.2.4. Third Iteration

There were only some minor changes made to the prototype. The only thing that changed was the browsing screen's background color (see fig.11). Instead of the cold blue-grey shade it was replaced with the same dark navy blue used to highlight the 'profiles' icon in the bottom navigation bar.



*Fig.11 Browsing Screen, Third Iteration – Prototype*

The final color palette was also updated to match the various shades that have been incorporated in the prototype, as well as shades that might be used in the future (see fig.12).



*Fig.12 Color Palette for Consat Finder, Second Iteration*

The final user test revealed that a couple users thought that the bottom navigation bar icons would look better without the text underneath. Another thought the 'select' button in the read CV screen was in the way when reading since it stays in place and would have preferred that it disappeared while scrolling down but reappeared when scrolling up. Unfortunately Adobe XD does not support scrolling or any other in-screen animation that is not related to screen transitions.

#### **4.4. Base Wireframe**

Once the prototyping phase was complete the application could be built. According to the Sprint planning, the first step was to build the base wireframe on which the rest of the application would be based on. The base wireframe included a header with the Consat Finder logo, the background for the main interaction area, and the bottom navigation bar (see fig.13).



*Fig.13 Base Wireframe*

#### 4.4.1. Header

The header was designed so that its height is a bit taller than the total height of the Consat Finder logo and the status bar put together. It stretches along the entire width of the screen with the Consat Finder logo in the center of the header. The color of the logo was to be the official Consat blue and a streak of orange would act as the divider between the header and the main interaction area.

The header's shape, color and the alignments of its items were styled in its corresponding `<View>` tag . A `StyleSheet` component was used to structure all the different styles used in the application, starting with the layout of the header. The placement of the Consat Finder Logo calculated and positioned in the stylesheet.

There were some issues with the positioning of the Consat Finder logo using the `justifyContent` and `alignItems` layout properties (shortened 'props'). It positioned the

logo slightly off-center which caused it to look jarring on the screen. As a result, the logo's position had to be coded manually by specifying the pixel distance from the left side of the screen as well as its distance from the top of the screen.

A thin orange line was added to differentiate between the header area and the interactable area since their background colors are similar. This was achieved by styling its height, width and color properties similarly to the header style, only that its height and background colors differ.

#### 4.4.2. Background

The background was the simplest element of the base wireframe since it is the main `<View>` component that has been styled. Its styling included filling the entire space using `flex: 1` and setting its `backgroundColor` to a navy blue color.

#### 4.4.3. Bottom Navigation Bar

The bottom navigation bar is an interactive feature which allows the user to switch between different screens (e.g the favorites screen) as per their convenience. However, the base wireframe only designs the visual layout of the navigation without developing any of the technical functionality of the navigation bar. The reason for this was because the base wireframe is not meant to implement any of the functions offered by the different features. Its purpose is to be the common template which is to be reused for all the subsequent screens.

### 4.5. Browsing Screen

The browsing screen was the current home screen for the application since it was the main feature. Its design was inspired by Tinder's simple but quick feature of swiping left and right to examine profile pictures and being able to see the entire profile page of those the user deemed relevant. Instead of the profiles disappearing after the user has swiped left or right, the profiles were displayed using a carousel-inspired layout (see fig. 14). Each profile contains a consultant's anonymous ID as well as their capabilities. When the user finds a promising consultant, they could press the 'View' button to see the consultant's CV.



Fig 14 Browsing Screen

#### 4.5.1. Database

The use of an existing database or creating and managing a new database lay outside the project's scope. Therefore, a local database consisting of dummy data had to be created and used. The dummy data was stored in an array where each object represented a CV. Each object had three sections of data stored in them: ID number, a list of their capabilities and a short description.

The ID numbers were substitutes for the consultant's name. The list of capabilities was written as a single string where each capability was divided by a comma. A short description written by the consultants highlights their field of expertise, experience level and any qualifications they might hold.

While all three data sections were accessed by the read screen to display the full CV, both the ID number and capabilities were also accessed by the browsing screen. The browsing screen used the data for the summarized profiles.

#### 4.5.2. Browsing Function

The desired carousel effect of the the browsing function was attained by utilizing an open source API developed by Kurtis Kemple [25] called `react-native-sideswipe`. It was difficult to understand how it was to be implemented due to the lack of an instruction guide. However after viewing examples of other projects using the same API it was possible to reverse engineer its implementation.

In order to use the `<SideSwipe>` component in the main JavaScript file, there were a few requirements which had be met. The first requirement was that there was an array available to be supplied as data. This was already solved since an array of dummy data was stored in a local JavaScript file. The second requirement was to have a class which handled the layout and animation of an item so that it could be applied to each object in the array. Creating a class which objectively specified the layout of a profile was simple, however comprehending the characteristics of the `<Animated>` component was another matter. There were many different animation types and each type had their own properties. All in all, it was very confusing. In the end, I had managed to understand the basics of the `<Animated.View>` sub-component and was able to use that to design the remaining elements.

#### 4.5.3. View Button

To wrap up the browsing screen, a decently sized 'View' button was placed below the profile-scrolling section. It served as a semi-interactable `<TouchableOpacity>` button at the time of its construction because while it was pressable, it did nothing else. Its purpose was to navigate the user to the read CV screen, however that screen had not been constructed at the time.

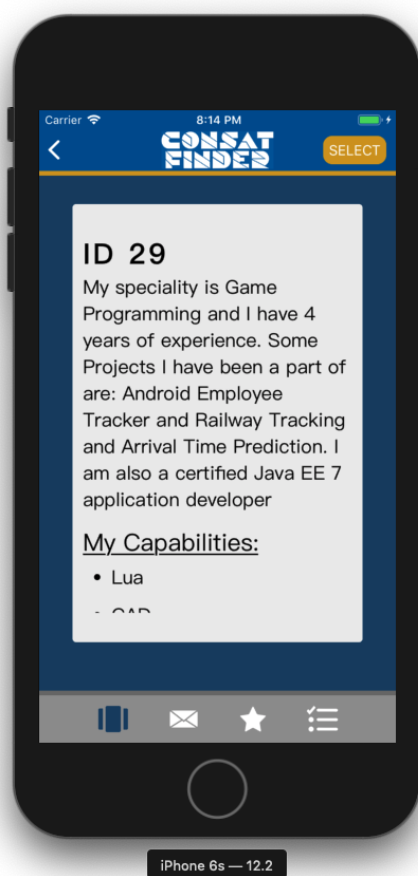
#### 4.5.4. Modification of Scope and MVP

The development of the browsing screen took longer than expected due to my inexperience with both JavaScript and the React Native framework. More time was spent on research and small-scale trials to understand the properties of React Native than initially planned. I would have had to spend time researching and learning every new component used as I proceeded development on the remaining screen. Thus I concluded the scope and MVP had to be modified.

The least valuable feature in the scope was the login screen. The reason it had less value than the other screen was because the technology already existed and could easily be applied through an API. Moreover, it would have required a substantial amount of time to create and develop the security aspect as well as the layout. Therefore, login screen was removed from the new scope while the rest remained. As such, the new MVP was to have a functional application where the user could browse through the profiles and then select one to read in depth.

#### 4.6. Read CV Screen

When a consultant's profile looked interesting, the user would then come to this screen to read the full CV. The screen had one purpose: to display a consultant's full CV using the data from the fake database (see fig. 15). Its design was simple: a scrollable 'page' containing a consultant's ID, experiences and capabilities. There was also a select button situated in the top right corner for the users to activate if they were considering hiring the consultant.



*Fig.15 Read CV Screen*

#### **4.6.1. CV Design**

The layout of the CV was as such that all the information resided inside the 'page'-styled `<View>` component. A `<SectionList>` component was used to render the different data sections (ID, description and capabilities) in the form of a scrollable list. There is another component called `<ScrollView>` which also renders information into a scrollable list, however the reason why a `<SectionList>` was used instead is that it offered section headers, which `<ScrollView>` did not.

Furthermore, a new class was created to help format the 'description' and 'capabilities' data sections. It provided styled how the consultant's description would appear on the screen, as well as structured the capabilities under a new subheading called 'My Capabilities'.

#### **4.6.2. Select Button**

The user can view all available profiles, read their corresponding CVs and then 'select' an action to book a meeting or to ask specific questions. This feature thus enhances the usefulness of the app, not just limited to browsing and viewing. In accordance with the feedback received from the third iteration of prototyping, the 'Select' button was not placed in its original position. Instead, it became a header icon located on the right side of the Consat Finder logo. Upon being pressed, a pop-up message appeared and asked the user whether they wish to interview the consultant or ask a question (see fig. 16). Once one of the options were selected, the user was rewarded with an alert message confirming their choice. This was not developed since the development of these features was not part of the scope. It is left for further development.

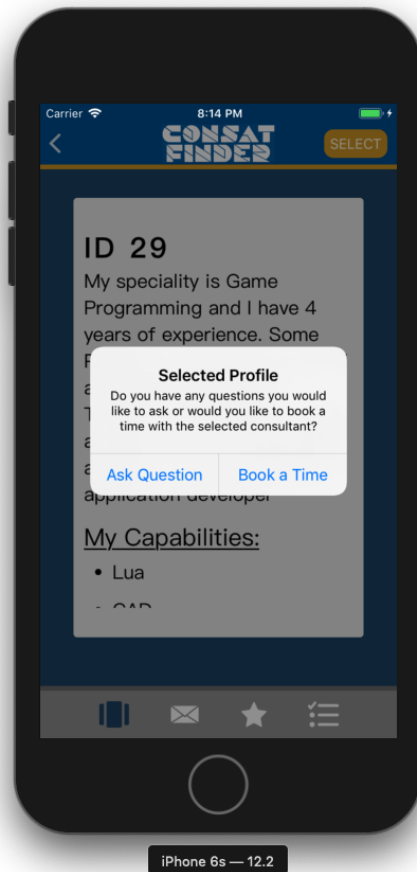


Fig 16 Select Options

## 4.7. Navigating Between Screens

So far, the main screens had been designed, however the application lacked functionality since there was no way to navigate from one screen to another. Through the use of React Native's own navigation API (which was not built in and needed to be imported into React Native), screen transitioning was smoothly incorporated into Consat Finder. Not only did the 'View' button finally take the user to the read CV screen, but the bottom navigation bar was also functional. The project's MVP was finally reached when the users could navigate between the different screens.

### 4.7.1. Implementing Screen Navigation

Unfortunately screen transitioning was not a function or component already built into React Native. However they did offer an open source API called `react-native-navigation` [26]. This particular API created a navigation stack of the different screens available and kept track of

the current screen in use as well as previous screens. Each screen needed to be declared in stack and associated with a class which would render the screen. This part was simple to implement since the classes were already built (e.g `BrowseScreen`, `ReadCvScreen`, `FavoriteScreen`, `MessageScreen` and `ReviewScreen`).

One small issue that occurred was transferring data from one screen to another. The problem arose during the screen transition to the read CV screen from pressing the 'View' button in the browsing screen. The screen transition was not the issue, it was that the application had no way of knowing which profile it transitioned from. The solution was to pass the data along as a navigation parameters. However this led to yet another problem: how to access the current data array index, `currentIndex` inside the `<SideSwipe>` component from the outside? The `currentIndex` is used by the `<SideSwipe>` component to iterate through the data array containing the consultants' information. The reason it was difficult to access the `currentIndex` was because it was a local variable declared inside the `<SideSwipe>` component and therefore could not be accessed from the outside. Without access to the `currentIndex`, the read CV screen could not display the CV which corresponded with the chosen profile. It was solved by declaring a new global state, called `profileIndex`, in the constructor. Its state changed whenever the index value changed, thereby gaining the same value as `currentIndex`. The only difference between `currentIndex` and `profileIndex` was that the value of `profileIndex` could be accessed outside of the `<SideSwipe>` component while `currentIndex` could not. Through the use of `profileIndex`, the correct CV could be displayed in response to a user pressing the 'View' button.

#### 4.7.2. Bottom Navigation Bar

The bottom navigation bar was further developed by altering the icons' components from `<Image>` to `<TouchableOpacity>`. This enabled them to become interactable and thus clickable. The bottom navigation bar functioned by triggering the screen to navigate to the screen that corresponded with the icon that was pressed. The icon for the current screen will turn navy blue while the rest remain white (see fig.17).



*Fig.17 The Different States of the Bottom Navigation Bar*

Note that navigation to the messaging, favorites and review screens led to the base wireframe screen since they had not been developed yet due to them being outside the given scope. They were projects to be done in the future.

#### 4.7.3. Back Button

Since the navigation stack kept track of not only the current screen but also of the previous screens, an extra feature of the API was that it came with a built-in back button. Pressing it took the user to the previous screen. The only screen which implemented the back button was the read CV screen, as can be seen by the back icon on the left side of the header in fig.15.

#### 4.7.4. Default Header Option

Another extra feature of the navigation API was the default header option that was available. Essentially instead of hard coding a header and having to render it in every screen, the navigation stack offered a default header which could be styled and altered differently from screen to screen. While the result was the same as the hard coded header that had been designed in the base wireframe, it was much simpler to use since the icons could be positioned by assigning them to either `headerRight` or `headerLeft`.

### 4.8. Filter Feature

With the MVP achieved, the filter feature was the next item to be developed. Its function was to allow the user to filter profiles based on the capabilities they require. It was a feature available

only in the browsing screen at the time of development. It would eventually also be available in the favorites screen. The filter icon was placed on the left corner of the header in the browsing screen (see fig. 18), mirroring its position from the second iteration of prototyping. When pressed, a menu listing the applicable capabilities appeared.

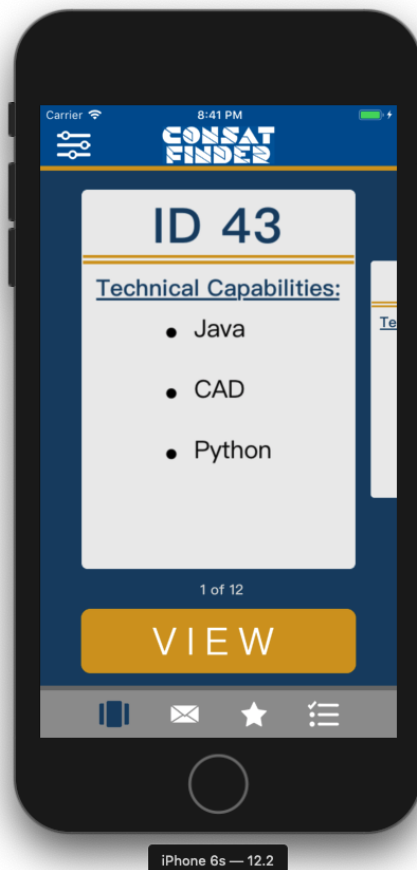


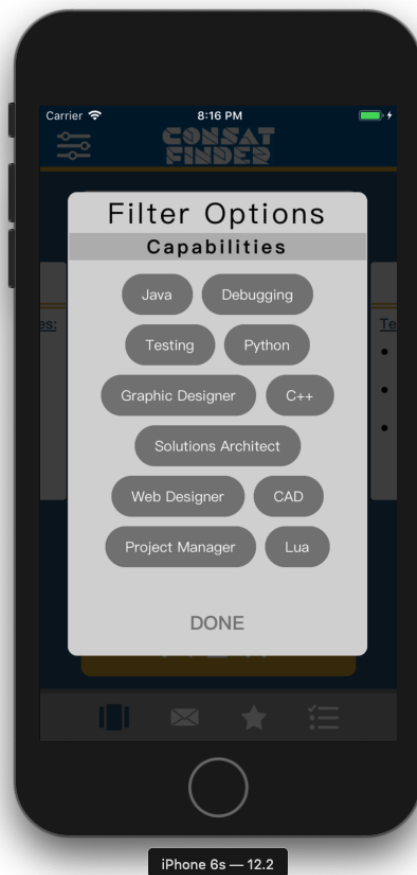
Fig. 18 Filter Icon

#### 4.8.1. Filter Design

In order to create a filter menu which slid up and dimmed the background, a modal component needed to be used. A modal component was a way to display content on top of a screen without having to redesign the screen layout itself. Pop-up menus are primary examples of modal components. While there was a built-in modal component in React Native, it was limited in terms of animations and styling properties. Therefore, another open source API, this time developed by the React Native community [27], called `react-native-modal` was chosen instead. It provided an expansion of the current `<Modal>` component in React Native by including new

animations and style customizations. It was chosen because it supported scrollable lists, which was a key characteristic of the filter menu.

The newly declared `<Modal>` component was used to create a simple menu with the title 'Filter Options'. It had one subheading called 'Capabilities' under which all the filter option buttons were gathered under (see fig.19). The reason for having a subheading at all was to leave room for the development of the 'favorite filter options' where users can select their favorite filter options in their user profiles.



*Fig.19 Filter Menu*

A class was developed to create filter option buttons based on what capabilities were available amongst the consultants. When inactive, the buttons remain dark grey, however once pressed they turn orange to signal they are active (see fig.20).

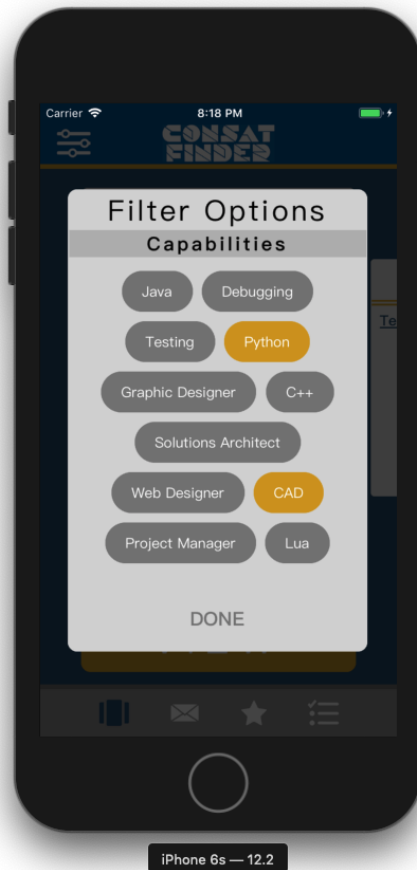


Fig.20 Active and Inactive Filter Options

#### 4.8.2. Filter Logic

The logic behind the filter feature was that once a filter option button was pressed, it would activate a function. The function would then compare the filter option to the capabilities of the profiles currently on the browsing screen. If it finds a match, it copies the consultant's data entry into a new array called `FilteredProfiles`. This was repeated until the function processed the last profile on the browsing screen, then it set a `state` variable to true and returned the `FilteredProfiles` array. The data in the `<SideSwipe>` component was then swapped with the `FilteredProfiles` array, which resulted in only those profiles that matched with the filter option being shown.

When a second filter option was chosen, the same process mentioned above was applied. Only this time the profiles currently on the browsing screen were already filtered once. The result was only profiles which matched both filter options remained. If there were no matches made, then

the pressed filter option button remain dark grey while the user is notified and is asked to try a different combination (see fig. 21).

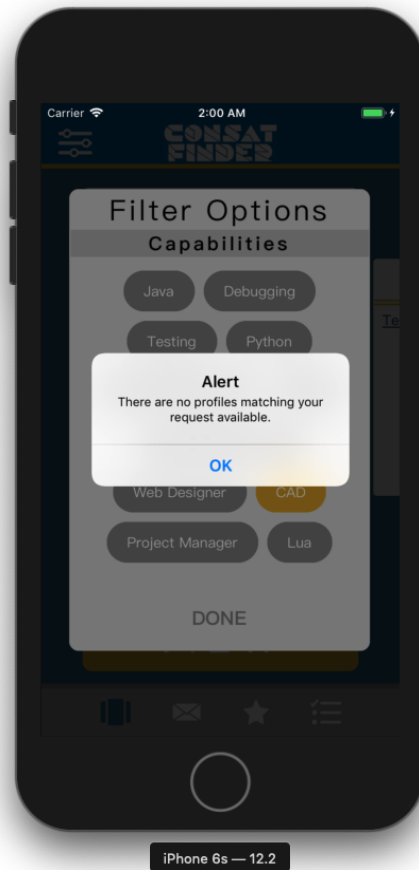


Fig.21 Filter Error Message

To deselect a filter option, the filter option need only be pressed again. When all the filter options were deselected, a local `state` was reset to `false` while the `<SideSwipe>` component returned to its default data containing all of the available profiles.

#### 4.9. Interactable Settings Menu

The base to a settings menu feature was developed to add an icon on the right side of the header in the browsing screen (see fig.22). The modal API that was used to build the filter menu was also used to create the settings menu. When the settings icon was pressed, a small menu slides in (see fig.23). While there are several options, none of their functions were developed. Instead, a message was displayed that told the user that was it under development. This was because it was not part of the scope for this project.

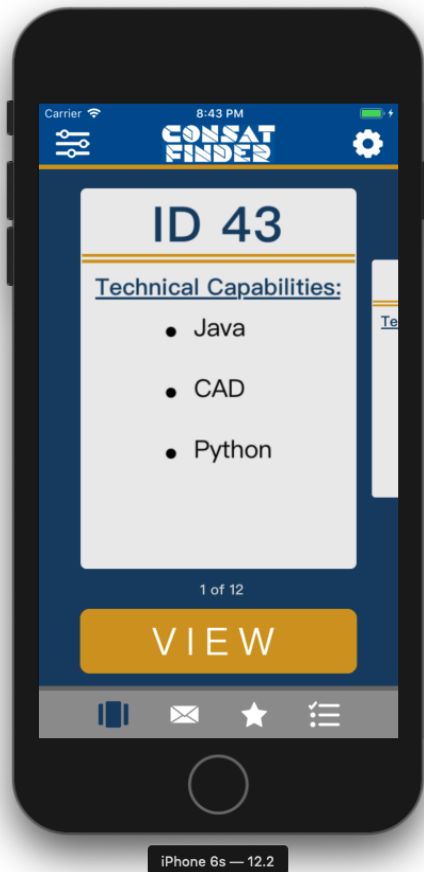


Fig.22 Settings Icon

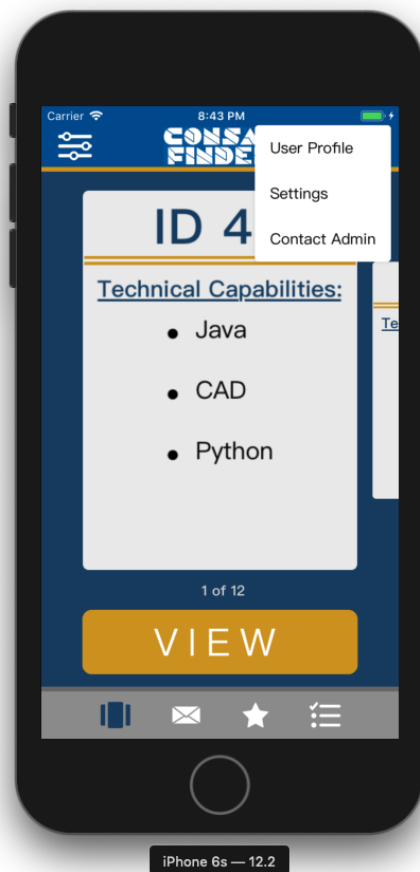
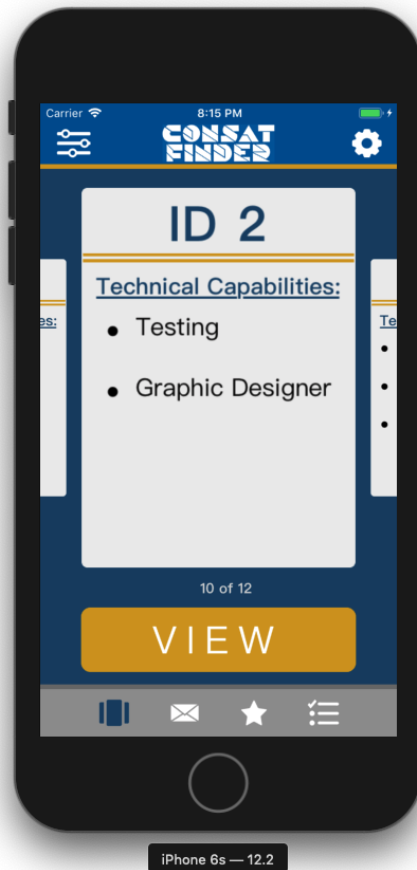


Fig.23 Settings Menu

#### 4.10. Profile Counter

It was easy to lose track of how many profiles there were with many profiles to browse through. Thus, a profile counter was implemented to help the user navigate themselves through the profiles. The counter displayed the current profile's index in relation to the total number of profiles available (see fig. 24). It was placed within the small gap between the current profile and the 'View' button.



*Fig.24 Profile Counter*

## 5. Result

The goal of constructing a frontend base for Consat where users could browse and filter through profiles and then read full length CVs was accomplished. While the scope and MVP were altered midway through the project, the original goal remained unchanged. The result was an interactable and functional application which met all the MVP requirements set, along with a few extra features. Its design and construction process were based on HCI-based processes such as requirement elicitation, specification and validation techniques. The prototypes were user tested both by peers knowledgeable within the field of computer engineering as well as peers without the technical background, and could therefore give detailed feedback from two perspectives. While the main profit was the development of Consat Finder itself, I also gained a lot knowledge on JavaScript and React Native as a result of it.

## 6. Conclusion

The application development of Consat Finder was the last phase of the Scrum process. Research into the current consultant recruitment process as well as discussions with Consat representatives are what led to the goals, scope, MVP and requirements being set. Thereafter started the iterative prototyping process of designing, user testing and then redesigning. Without prototyping not only would it have been more difficult to visualize the application's various screens, but there would not be any user tests, thereby losing the valuable feedback gained from the users. For example, the color scheme was altered quite a bit due to user feedback.

The topic of anonymity is still being debated within Consat. Presenting the clients with nameless and faceless profiles makes the whole consultant recruitment process feel disconnected and static. However the purpose of the anonymity was to ensure the consultants are examined purely on their technical qualities and not discriminated for personal reasons. A possible solution could be to add pictures to each profile, however it is unclear what kind of picture – whether it can be a picture of the the consultant or something else, it is something that needs to be decided by Consat since it is their requirement.

The original estimated effort and time plan proved to be erroneous. It took longer to design and develop the application than foreseen. One of the reason was that I had to learn React Native and Javascript programming, although I was familiar with Python, Java and other programming languages previously. Since I was already familiar with programming logic, I had assumed not much time would be required to learn JavaScript's syntax. This proved to be a misjudgment on my part since the syntax for JavaScript was unlike the other programming languages since it was comprised of components used for designing user interfaces. Due to my inexperience with both JavaScript and the React Native framework, everything had to be learned from scratch and tested. As a result, my Sprints became longer and I had to modify the scope and MVP after completing the browsing screen.

Another reason was that more time was spent on background research and gathering user requirements through various interviews than expected. Online information on how clients recruit consultants was scarce. There were very few trustworthy articles about the topic, and even then

they did not provide the specific information I was searching for. In the end, I was briefed on Consat's perspective of the process, of how clients call in or email them about recruiting one of Consat's consultants. While the consultant hiring process described in this thesis is in accordance with Consat's methods, it does not apply to all consulting companies.

Finally, some additional features were also included into the ConsatFinder. Since the login screen was removed from the scope, more time was available to be spent on developing features, such as the filter feature mentioned in the goal. The base functions for the settings menu, 'Select' button and the bottom navigation bar features were initiated but not fully developed. Those have been intentionally left partially-done as a future development opportunity.

The Consat Finder MVP prototype had been tested with peer students. The design and prototype has also been tested with other age demographic groups like teenagers under 18 who are rather picky about user interface interactivity. The objective was to observe how they felt about the ease of learning to use the app, how they reacted to the placement of the different buttons and so forth. Feedback and insights were also gathered from a senior architect from a global consultancy firm where similar use cases exist.

MVP and scope was updated after the browsing screen was developed. Due to time constraints, the user login screen had to be removed from the scope. The reason why that was removed was because there are plenty of APIs available to construct it since it isn't anything new. The other use cases hold more value to Consat's objective. No need to program something that already exists.

There were struggles when it came to understanding how to handle the different components and their properties. The first issue was on how to access the value of a local variable declared inside a component from another outside component. The solution was to create a global state variable, however it took a few hours to solve. The second issue was on how to access a certain value of an object located inside an array. This ended up being a simple matter of using the write index at the right location. The solution was to use two different indices: one to specify which object within the array, the second to specify which value was to be accessed.

Despite the difficulties at the beginning in understanding the environment and the language, as the development of Consat Finder progressed the more I understood and worked. In the end, a

functional frontend-based version of Consat Finder was developed and delivered to Consat who were satisfied with the outcome.

## 6.1. Other Solutions

Regarding the composition of Consat Finder, there were other possibilities looked into at the start of the project. During the initial brainstorming session of what type of application it could be, there were ideas about Consat Finder also being a web-application. Another one was about using LinkedIn as the CV database.

### **Using pre-existing open source solutions to create a web-based application, such as OpenCart:**

Development time can be reduced. Most of the features would just need to be configured. Focus then would be primarily on content that is to be published. However, then the flexibility to design own user interface layouts, elements and color scheme like has been done for Consat Finder would be lost.

### **LinkedIn:**

The decision on what kind of database would be used to store the consultants' profiles is still being debated. One option is to use Consat's current profile database, another idea is to use LinkedIn. It is a prevalent professional and social networking app. People build up their network of known acquaintances, follow thought leaders in their respective fields of interest or product vendors or solutions that are of interest. LinkedIn is available as open free option as well as a paid premium membership option. In addition to having a newsfeed channel that helps people to be updated on the latest news and innovations, it has the networking platform allowing users to comment, like, share. It also has the feature for personal message chatting option. This is commonly used by headhunters and recruiters who search for the profiles that are relevant and then contact the people using private messaging in LinkedIn. Therefore, it makes sense to connect the consultants' LinkedIn profiles with their corresponding profiles in Consat Finder.

## 6.2. Environmental and Ethical Aspects

The creation of the Consat Finder application makes the consultant recruitment process more efficient for both the clients as well as Consat. It is efficient in the sense that it saves time spent on back-and-forth calls and emails. This leads to a decrease in CO<sub>2</sub> emissions from transportation

vehicles used by the clients in transit to recruitment meetings while still process of picking a consultant. In the current recruitment process, clients and a representative of the consulting company may hold several meetings before a consultant is hired. With Consat Finder, all the searching is done by the client using the application. This way, the client will only be required to travel when there is an interview booked with a potential consultant to recruit.

From the ethical standpoint, the anonymity offered by the nameless profiles means that the consultants are chosen purely based on their capabilities, experience and other work-related qualities. Thereby, they cannot be discriminated by the client during the selection process since the client can only see their skills.

### **6.3. Suggestions for Further Improvements**

#### **Favorites Screen**

From the browsing screen, clients should be able to mark their favorite profiles by pressing a star icon attached to each profile. The star turns yellow when active and remains opaque when inactive. The starred profiles can then be browsed through in the favorites screen. The favorites screen uses the same scrolling functionality implemented in the browsing screen.

#### **Availability Details**

See availability period of each consultants and filter after availability. Introduce more details like date from when a consultant is potentially available. Then one of the filter search options could be based on available date.

#### **Location Details**

Another detail could be Location, that is where does the client require the consultant to come and work. For example are they requiring to find someone to help in Göteborg or in Stockholm, Karlstad etc. Then Location is also another filter condition.

It could also be possible for the user to select multiple filter categories that return search results which are a combination (example : capability = JAVA AND Location= Göteborg)

#### **End User Testing with Real Clients**

Next steps would include expanding the dummy data with more realistic, CV data and then testing with Consat's clients.

### **Build User Profile Interface**

Have the features for the user to create and update their own profile and preferences. Also update their preferred contact details , phone or email. The profile pages can also display the history of the last searches made by the user. It can also show the history of the last actions taken like meeting requests sent, or profiles downloaded.

### **Reset Current Filter Selections**

Once the client has made multiple filter selections, there should be a button to clear all the filters, that is reset. Currently the user has to remove each selection criteria one by one.

### **Be Able to Customize User Preferences**

A customized filter so that rather than having to pick from the general pool of filters, the user has a pool of custom-selected filters from which they can select from (quick access filters, basically). Users can customize this pool from their profile page]

### **Set Up RFP**

The client should be able to enter their RFP into their profile, which should also automatically filter based on the required skills a consultant must possess. This also means that when the admin (and consultant) receives a request for a consultant, the admin (and consultant) are able to view the RFP.

## References

- [1] Consultancy, “*What is a Consultant?*”. [Online]. Available: <https://www.consultancy.uk/career/what-is-a-consultant>. [Accessed: May. 18, 2019]
- [2] C. MacNamara, “*How to Successfully Hire and Work With an Excellent Consultant*”. [Online]. Available: <https://managementhelp.org/consulting/hiring-consultants.htm>. [Accessed: May. 18, 2019]
- [3] H. Tubman, “*Consultant Hiring Process*”. [Online]. Available: <http://www.nonprofitinclusiveness.org/consultant-hiring-process>. [Accessed: May. 7, 2019]
- [4] O. Elgaby, “*Requirements Engineering – Introduction (Part 1)*”, Sep. 13, 2016. [Online]. Available: <https://medium.com/omarelgabrys-blog/requirements-engineering-introduction-part-1-6d49.001526d3> [Accessed: May. 20, 2019]
- [5] J. Horkoff, DIT045, Class Lecture, Topic: “*Lecture 3 – RE: Requirements Modeling & Use Cases*”, Chalmers, Gothenburg, Sweden, 2018
- [6] T. Bures et al, “*Requirement Specifications Using Natural Language*”, Dec. 2012. [Online] Available: <https://pdfs.semanticscholar.org/b262/d775001635123fcde4f12b72830b17c01ea7.pdf>. [Accessed: May. 20, 2019]
- [7] J. Horkoff, DIT045, Class Lecture, Topic: “*Lecture 5 – RE: Scenarios, Personas & Creativity*”, Chalmers, Gothenburg, Sweden, 2018
- [8] O. Elgaby, “*Requirements Engineering – Requirements Validation (Part 4)*”, Sep. 14, 2016. [Online]. Available: <https://medium.com/omarelgabrys-blog/requirements-engineering-requirements-validation-part-6-29778d7bde24>. [Accessed: May. 20, 2019]

[9] J. Horkoff, DIT045, Class Lecture, Topic: “Lecture 2 – RE: Elicitation & Scoping & Risk”, Chalmers, Gothenburg, Sweden, 2018

[10] J. Horkoff, DIT045, Class Lecture, Topic: “Lecture 12 – RE & UX: Verification & Validation”, Chalmers, Gothenburg, Sweden, 2018

[11] J. Preece et al, *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, 2015, ch. 11. [E-book]. Available: <https://library-books24x7-com.proxy.lib.chalmers.se/assetviewer.aspx?bookid=142571&chunkid=803821435>. [Accessed: May. 18, 2019]

[12] J. Preece et al, *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, 2015, ch. 1. [E-book]. Available: <https://library-books24x7-com.proxy.lib.chalmers.se/assetviewer.aspx?bookid=142571&chunkid=803821435>. [Accessed: May. 18, 2019]

[13] Facebook Inc., “React Native”. [Online]. Available: <https://facebook.github.io/react-native/>. [Accessed: May. 19, 2019]

[14] Facebook Inc., “React”. [Online]. Available: <https://reactjs.org/>. [Accessed: May. 19, 2019]

[15] Facebook Inc., “Props”. [Online]. Available: <https://facebook.github.io/react-native/docs/props>. [Accessed: May. 19, 2019]

[16] Facebook Inc., “State”. [Online]. Available: <https://facebook.github.io/react-native/docs/state>. [Accessed: May. 19, 2019]

[17] A. Ravichandran, “Props and State in React Native Explained in Simple English”, Mar. 17, 2018. [Online]. Available: <https://codeburst.io/props-and-state-in-react-native-explained-in-simple-english-8ea73b1d224e>. [Accessed: May. 19, 2019]

- [18] B.Eisenman, “*Learning React Native*”, Dec. 2015. [Online]. Available: <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>. [Accessed: May. 19, 2019]
- [19] D. Norman, J.Nielsen, “*The Definition of User Experience (UX)*”, 2014. [Online]. Available: <https://www.nngroup.com/articles/definition-user-experience/>. [Accessed: May. 19, 2019]
- [20] Agile Alliance, “*Scrum*”. [Online]. Available: [https://www.agilealliance.org/glossary/scrum/#q=~\(infinite~false~filters~\(postType~\( '~page~'post~'aa book~'aa event session~'aa experience report~'aa glossary~'aa research paper~'aa v~ideo\)~tags~\( '~scrum\)\)~searchTerm~'~sort~false~sortDirection~'asc~page~1](https://www.agilealliance.org/glossary/scrum/#q=~(infinite~false~filters~(postType~( '~page~'post~'aa book~'aa event session~'aa experience report~'aa glossary~'aa research paper~'aa v~ideo)~tags~( '~scrum))~searchTerm~'~sort~false~sortDirection~'asc~page~1). [Accessed: May. 14, 2019]
- [21] Scrum Institute, “*The Scrum Product Backlog*”. [Online]. Available: [https://www.scrum-institute.org/The\\_Scrum\\_Product\\_Backlog.php](https://www.scrum-institute.org/The_Scrum_Product_Backlog.php). [Accessed: May. 14, 2019]
- [22] M. Rehkopf, “*Sprints*”. [Online]. Available: <https://www.atlassian.com/agile/scrum/sprints>. [Accessed: May. 14, 2019]
- [23] Scrum.org, “*What is a Sprint Review?*”. [Online]. Available: <https://www.scrum.org/resources/what-is-a-sprint-review>. [Accessed: May. 14, 2019]
- [24] J-P. Steghöfer, DAT256, Class Lecture, Topic: ‘*Lecture 6 – A Primer to Agile Requirements Management*’, Chalmers, Gothenburg, Sweden, 2018
- [25] K. Kemple, “*Introducing Sideswipe: a cross-platform carousel for React Native*”, Feb. 12, 2018. [Online]. Available: <https://blog.expo.io/introducing-sideswipe-a-cross-platform-carousel-for-react-native-8b9a0f18df53>. [Accessed: May. 14, 2019]
- [26] Facebook Inc., “*Navigating Between Screens*”. [Online]. Available: <https://facebook.github.io/react-native/docs/navigation>. [Accessed: May. 14, 2019]
- [27] React Native Community, “*react-native-modal*”, 2019. [Online]. Available: <https://github.com/react-native-community/react-native-modal>. [Accessed: May. 14, 2019]

## Appendix A

A table featuring the user stories brought forth during the requirements gathering stage of the project.

#	As a/an	I want...	So that...
1	client	to be able to browse through all the available consultants	I can see my options.
2	client	to be able to read a profile	I can see the consultant's capabilities and experience.
3	client	to be able to favourite/star potential candidates	I can view them later.

<b>4</b>	client	to be able to browse through my favourite profiles	It is easier to go back and review them.
<b>5</b>	client	to be able to filter through the profiles	I can find profiles that match my requirements.
<b>6</b>	client	to be able book an interview time with a consultant	I can meet them face to face to interview potential candidates.
<b>7</b>	client	to be able to ask follow up questions regarding a consultant	I can gain a better understanding.
<b>8</b>	client	to be able to create my own filter list	I can quickly access my most used filter options.
<b>9</b>	client	to be able to log into the application	I can find a consultant.
<b>10</b>	client	to be able to logout	nobody else can use my account when using my phone.
<b>11</b>	PO	to have anonymous profiles	the consultants can't get discriminated.
<b>12</b>	PO	only Consat approved clients to be able to use the application	the competition cannot gain entry.