*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

Project Title: Exploring the Unconscious

**Team Members: Ruben Valdes Jr., Daniel Antonio**

**Product Owner(s)**: Mark Mohr

**Mentor(s)**: Mario Eraso

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*This document presents the information necessary to gain a good understanding of Exploring the Unconscious. The purpose of this project is to provide a platform for users to connect to their unconscious. We implemented a java solution to solve this problem and serve that information to a front-end designed in javascript. Using an algorithm developed by mark mohr rooted in numerology, we were able to develop the 1.0 version of an application that decodes words and phrases to a 3 digit code to test for compatibility. Testing yielded some success and revealed some bugs in our programing that we hope will be fixed for version 2.0*

**Table of Contents**

**Introduction** …………………………………………………………………………………………………………………………….. 5

Current System ……….…………………………………………………………………………………………………………... 5

Purpose of New System …………………………………………………..…………………………………………………... 5

**User Stories**

Implemented User Stories ………………………………………….………………………………………………………..7

Pending User Stories …………………………………………………………………...………………………………..….. 10

**Project Plan**

Hardware and Software Resources ……………………………………………………...……………………….… 12

Sprints Plan ……………………………………………………………..………………………………………………………. 13

*Sprint 1*  …………………………………………………………………………………………………………………………... 13

*Sprint 2*  …………………………………………………………………………………………………………………………... 13

*Sprint 3*  …………………………………………………………………………………………………………………………... 14

*Sprint 4*  …………………………………………………………………………………………………………………………... 15

*Sprint 5*  …………………………………………………………………………………………………………………………... 16

*Sprint 6*  …………………………………………………………………………………………………………………………... 17

*Sprint 7*  …………………………………………………………………………………………………………………………... 18

**System Design**

Architectural Patterns ………………………………………………………………………………………………….. 20

System and Subsystem Decomposition …………………………………………………………………………….… 21

Deployment Diagram …………………………………………………………………………………………………….…... 22

Design Patterns ………………………………………………………………………………………………………….….... 22

**System Validation**  …………………………………………………………………………………………………………………….23

**Glossary**  ………………………………………………………………………………………………………………………………….37

**Appendix**  ………………………………………………………………………………………………………………………………….38

Appendix A - UML Diagrams ………………………………………………………………………………………………. 38

*Static UML Diagrams*  ……………………………………………………………………………………………………….38

*Dynamic UML Diagrams*  …………………………………………………………………………………………………..40

Appendix B - User Interface Design …………………………………………………………………………….…... 52

Appendix C - Sprint Review Reports …………………………………………………………………………...…… 69

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents …………………………………………………………………….…………… 74

**References** ……………………………………………………………………………………………………………………...………...80

# Introduction

The purpose of this document is intended to help guide the development of our project, Exploring the Unconscious. The document will provide developers an insight into the work that went to developing this project. Also the document facilitates communication and understanding of the system by providing several different views of the system design.

## 

## Current System

As we all know, when we have a personal question we ponder what is the best answer for our question. Whether it is a question as simple as “what restaurant should I take my boss to eat” or “what colored shirt should I wear today” it is natural for us to have personal questions. In turn, it is natural that we will come up with possible answers to these questions. The struggle lies in deciding which answer best suits our question. It can be argued that many of us already know which solution we most desired but for many of us it is difficult to discover that solution. This is because, more often than not, the solution we most desire is locked within our unconscious mind. Stress and prolonged periods of time often makes it difficult for our conscious mind to properly sync to our unconscious mind. This in turn leads people to make rash decisions and choose answers to their questions that may not be as desirable as one may think.

## Purpose of New System

The purpose of our system is to provide a platform with which a user may connect with their unconscious mind. The system will allow the user to compare the “compatibility” of words and phrases in order to determine which answers best suit their questions. As the user utilizes the system, they will need to think harder and more thoroughly about how best to write their inputs so as to acquire compatibility. In doing so, the user will be better able to sync their thoughts with their unconscious minds and thus find the exact answer or answers that they desire.

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the Exploring the Unconscious project. These user stories served as the basis for the implementation of the project’s features. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

**Use Case**Name**: Making an account**Actor: Guest  
Preconditions: As a user I must be able to create an account by clicking the register button  
Description <Flow of events>:  
Guest selects “Create an Account”  
System displays an account setup page that asks for First Name, Last Name, E-Mail  
 Address, E-Mail confirmation, Password, and Password confirmation.  
Guest fills out requested information and presses “Create Account”  
System directs Guest back to the login screen.  
  
Exit condition: Guest clicks “register after creating their profile  
Exceptions:  
User enters an E-mail confirmation that does not match original E-Mail.  
System displays notice saying E-Mails do not match.  
User enters a password confirmation that does not match original   
 password. System displays notice saying passwords do not match.

**Use Case**  
Name: **Navigate to Explore Page**  
Actor: User  
Preconditions: None   
Description <Flow of events>:  
  
 User clicks Explore from home page  
 App navigates to the Explore page  
  
Exit condition: App successfully navigates to Explore page  
Exceptions: User clicks a different button.

**Use Case**  
Name: **Login to system**  
Actor: Guest  
Preconditions: As a user I must be able to Login to my account by clicking the login button  
Description <Flow of events>:  
Guest selects “Login”  
System displays a log in screen, Email and Password.  
  
Exit condition: Guest clicks “login” after filling out the information.  
Exceptions:  
User enters an E-mail confirmation that does not match original E-Mail.  
System displays notice saying E-Mails do not match.  
User enters a password confirmation that does not match original   
 password. Incorrect password.

**Use Case**

Name: **Compatibility**  
Actor: User  
Preconditions: As a user I must be able to check the compatibility of my question and answer through my own input  
Description <Flow of events>:  
User is prompted as to whether they wish to check that the question and answers they have inputted are compatible.  
User inputs “Y”.   
System outputs whether the question and answers are compatible.  
User is prompted whether they wish to test another question and answers.  
  
Exit condition: User types in “N” that they do not wish to input anymore questions or answers

**Use Case**  
Name: **Input**  
Actor: User  
Preconditions: As a user I must have fields to input my desired question and answer  
Description <Flow of events>:  
User is prompted that they must have a minimum of three answers before using this program.  
User is prompted to input a single question.  
User inputs their question.   
User is prompted to input there 1st answer.   
User inputs their answer.  
User is prompted to input their 2nd answer.  
User inputs their answer.  
User is prompted to input their final answer.  
User inputs their answer.  
User is prompted whether they wish to test another question and answers.  
  
Exit condition: User types in “N” or “Y.  
Exceptions:  
User inputs a question or answer with a numerical value in it.   
System informs user that they cannot input numerical values directly, they must write them out. Program is terminated.  
  
**Use Case**  
Name: **Continuation**  
Actor: User  
Preconditions: As a user I must have the option to quit using the system  
Description <Flow of events>:  
User is prompted whether they wish to test another question and answers.  
User inputs “Y”  
User inputs the next set of question and answers  
  
Exit condition: User types in “N”.

## Pending User Stories

**Use Case  
Name: Logout**Actor: User  
Preconditions: User must be logged in  
Description <Flow of events>:  
  
User clicks on logout button  
System signs the user out and takes the user to the Home screen.  
  
  
Exit condition: User is logged out  
Exceptions: User is not logged in.

**Use Case**

Name: **Previous Inputs**

Actor: User

Preconditions: User must have inputted something prior

Description <Flow of events>:

User clicks on previous inputs button

System sends user to previous inputs page

User can view all their previous inputs and what was their compatibility

# Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

## Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

* **NetBeans**
* **Git Hub**
* **Android Studio**
* **Mingle**
* **Vim**
* **Firebase**
* **React native**
* **Stack Overflow**
* **Node Js**

## 

## 

## Sprints Plan

### Sprint 2

Attendees: <Daniel Antonio, Ruben Valdes>

Start time: <10:00am>

End time: <11:00am>

After discussion, the velocity of the team were estimated to be <2 weeks>.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story #674 Compatibility & #677 Access Prior Input
* User Story #668 User login & #671 Make Account

The team members indicated their willingness to work on the following user stories.

* Daniel Antonio
* User Story #668 Login to system
* User Story # 671 Make Account
* Ruben Valdes
  + User Story #677 Access Prior Input
  + User Story #674 Compatibility

### Sprint 3

Attendees: <Daniel Antonio, Ruben Valdes>

Start time: <10:00am>

End time: <11:00am>

After discussion, the velocity of the team were estimated to be <2 weeks>.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story #674 Compatibility & #673 Input
* User Story #672 Logout & #679 Navigate To Explore Page

The team members indicated their willingness to work on the following user stories.

* Daniel Antonio
* User Story #672 Logout
* User Story #679 Navigate To Explore Page
* Ruben Valdes
  + User Story #673 Input
  + User Story #674 Compatibility

### Sprint 4

Attendees: <Daniel Antonio, Ruben Valdes>

Start time: <10:00am>

End time: <11:00am>

After discussion, the velocity of the team were estimated to be <2 weeks>.

The product owner chose was satisfied with the user stories we had completed and requested no further features for the product.

# System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

## 

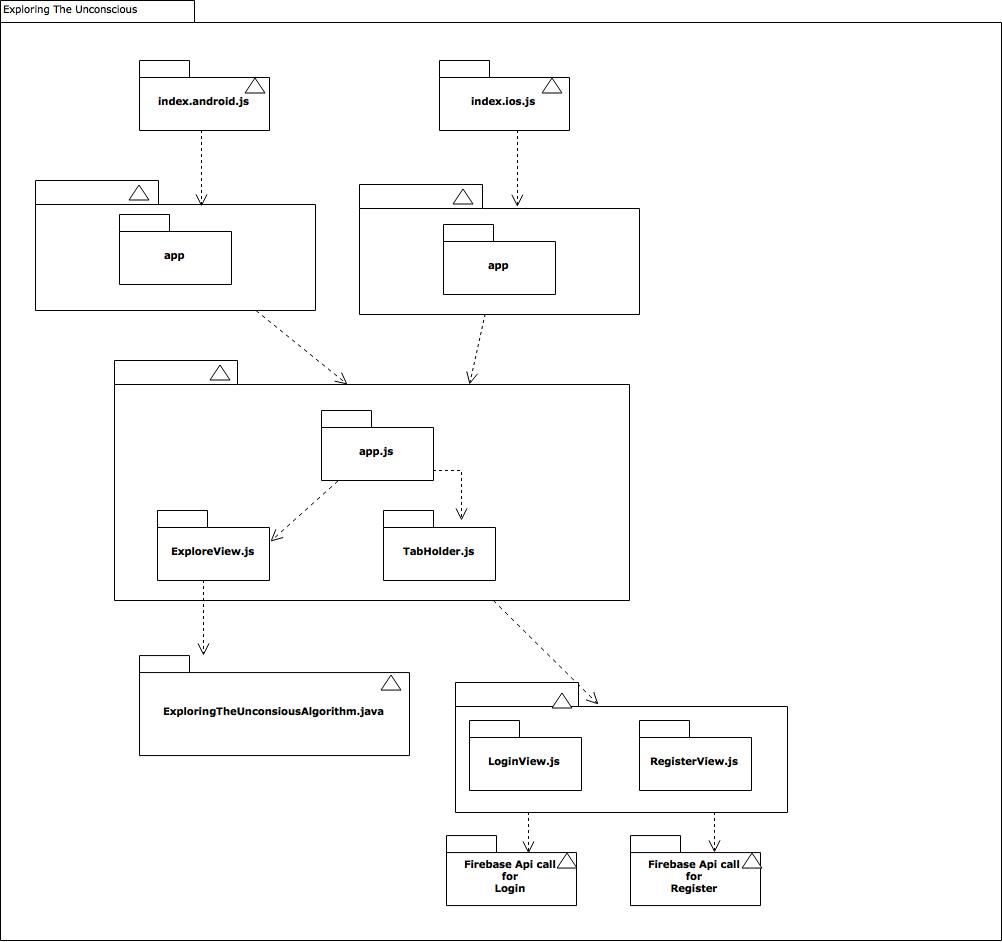
## Architectural Patterns

The primary architectural design we went with was Model-View-Controller (MVC). This was seen as the best design because we had user data as well as the question data to model for our database. We had to use a controller to access the fields in the model and serve them to the view as a service. MVC was methodology that facilitated this the best due to the fact that we had all parts the model, the view, and the controller that went into building the application.

## System and Subsystem Decomposition

## The application files systems hierarchy is broken down into files that have navigators to other files that can make api calls. Both the android and ios index files call the app.js file that resides within the app folder., which compiles natively to their respective operating systems. The app.js file is the home landing page of the app which contains a navigation stack that can navigate to either the explore page--ExploreView.js--or the TabHolder page--TabHolder.js. The ExploreView will in the future make an api call to the ExploringTheUnconsiousAlthrithm.java file and return the results to the page. The LoginView and RegisterView are in a tabular format switchable with a swipe to the left or the right which allows a user to create an account or to register, the user object is stored into the database.

## Deployment Diagram



## Design Patterns

In terms of creational patterns, for the coding of the algorithm the factory method was primarily used. This is because the code relied heavily on the inheritance of user data which would then delegate the creation of objects to the various subsystems within the project.

For the front end code the creation pattern that was used was the builder pattern. The front-end was designed to be a wrapper for the methods in the algorithm and encapsulates the code to provide the service for the end user.

In terms of structural patterns, the bridge method was used for the algorithm. This is because the actual data within the system can vary between classes, thus it is useful to think of the classes as an abstraction.

The adapter pattern was used in the implementation of the front-end as this is what the user sees. This ties in with the architectural pattern we used which was MVC; the model is decomposed when it arrives to the front end for the user to physically see.

In terms of behavioral patterns, the algorithm’s code is a mix of chain-of-responsibility and command pattern. The classes operation is dependent on the classes that preceded it and each class is expected to follow commands as response to data that it receives. This is also true for the front-end design as the user input will be treated as object to be handled eventually with an api call and the response will be decomposed which is the command pattern. In addition, some user requests can be treated as a state change, similar to the logout feature which should change the state of the app, enabling certain features and disabling others.

# System Validation

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Purpose** | **Results** |
| **Test 1** | **Test whether the system runs the compatibility function.** | **PASS** |
| **Test 2** | **Test that the compatibility function will run for more than one answer.** | **PASS** |
| **Test 3** | **Test that the computations performed by the compatibility function are mathematically correct based on the algorithm provided.** | **PASS** |
| **Test 4** | **Test whether the system prevents the user from inputting numeric values.** | **PASS** |
| **Test 5** | **Test whether the user can continue using the system after the initial input.** | **PASS** |
| **Test 6** | **Test the compatibility of the phrase “large dog” with “wolf”, “dog” and “cat”.** | **PASS** |
| **Test 7** | **Test the compatibility of two names “Mario Eraso” and “Mark Mohr”.** | **PASS** |
| **Test 8** | **Test the compatibility of the word “table” with itself.** | **PASS** |
| **Test 9** | **Test whether the system will output a “Not Compatible” for the words “bus” and “car”.** | **FAIL** |
| **Test 10** | **Test revised code for whether it outputs a “Not Compatible” for the words “bus” and “car”.** | **PASS** |
| **Test 11** | **Test to make sure all variables are reset after each iteration of the program.** | **PASS** |

# 

# 

# 

# 

# Glossary

|  |  |
| --- | --- |
| Keyword | Definition |
| **Compatibility** | There is at least one digit in the 3 digit encoding of one word or phrase that matches at least one digit in the encoding of the other word or phrase. |
| **Navigation** | Movement through the UI |
| **Redux** | State management tool for react-native |
| **Database(non-relational)** | Collection of stored data |
| **Sequence Diagram** | Describes the interaction processes showing the how and what order. |
| **MVC** | Model, View, Controller. Methodology that we used for developement |

# Appendix

## Appendix A - UML Diagrams

## 

## 

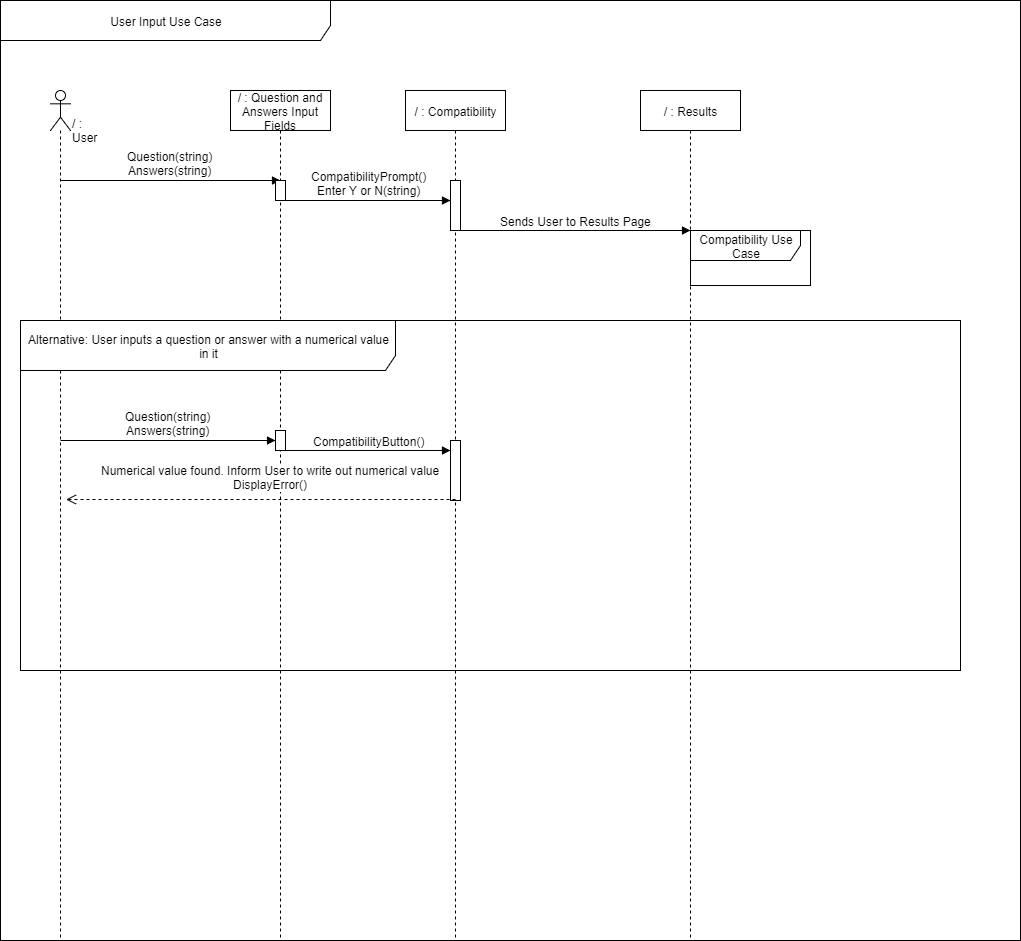
## 

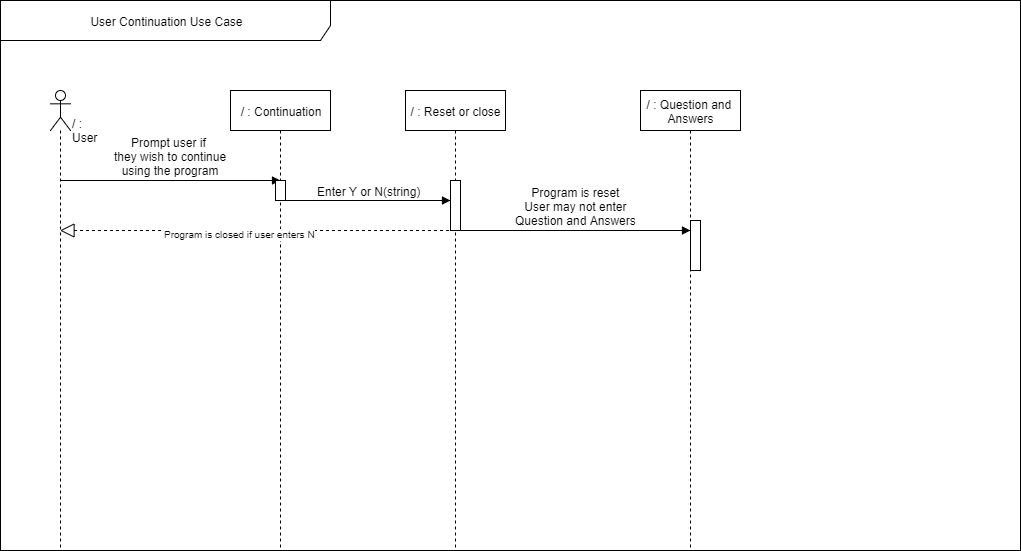
## 

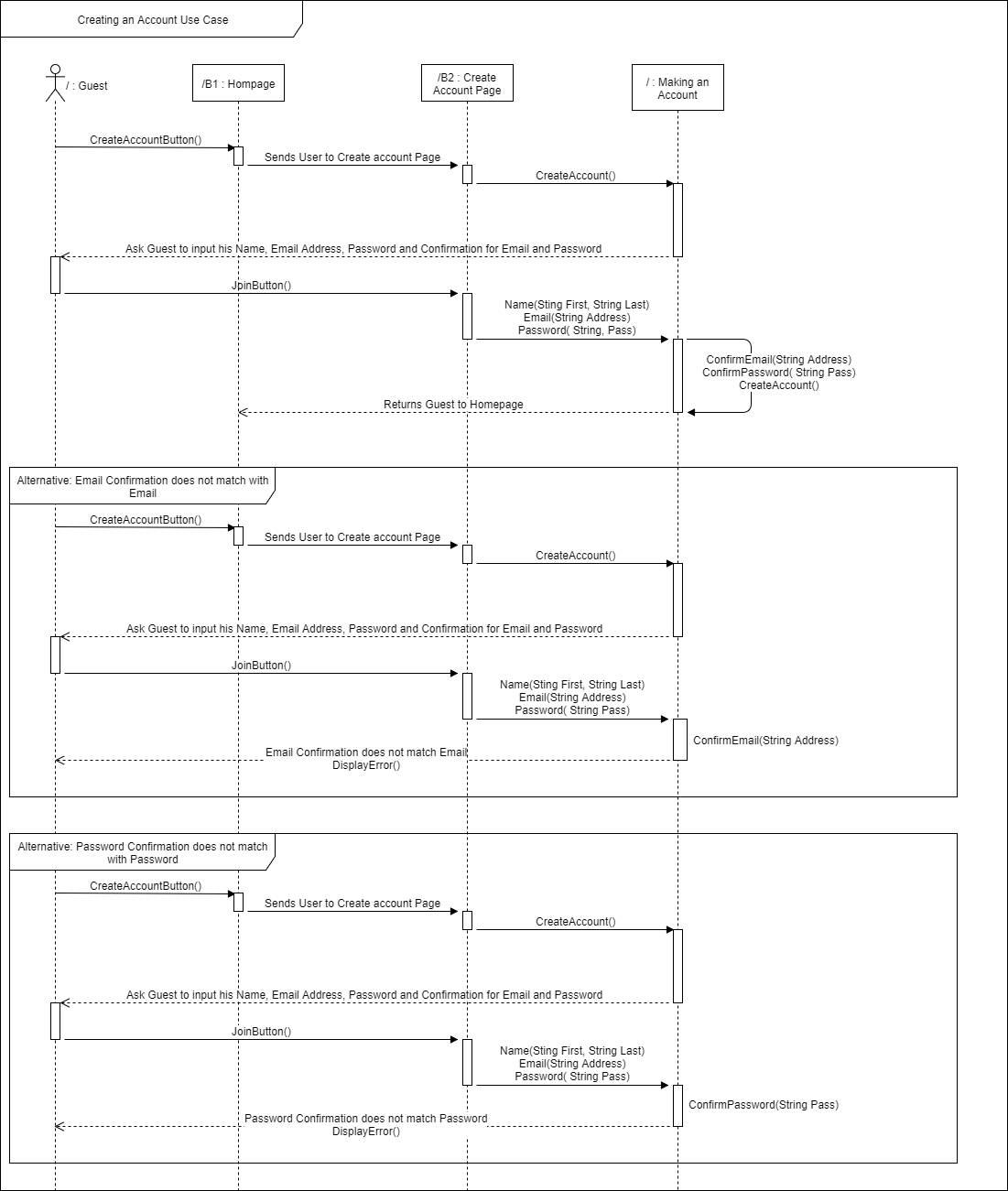
## 

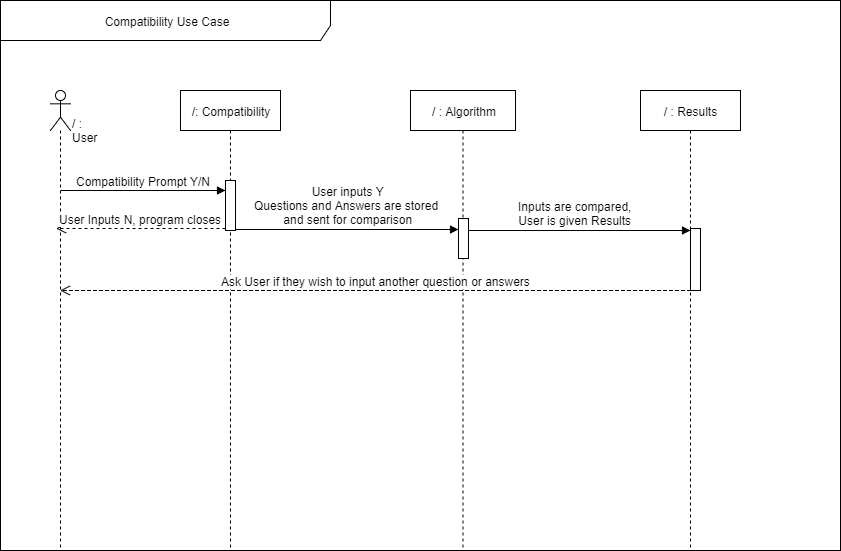
## 

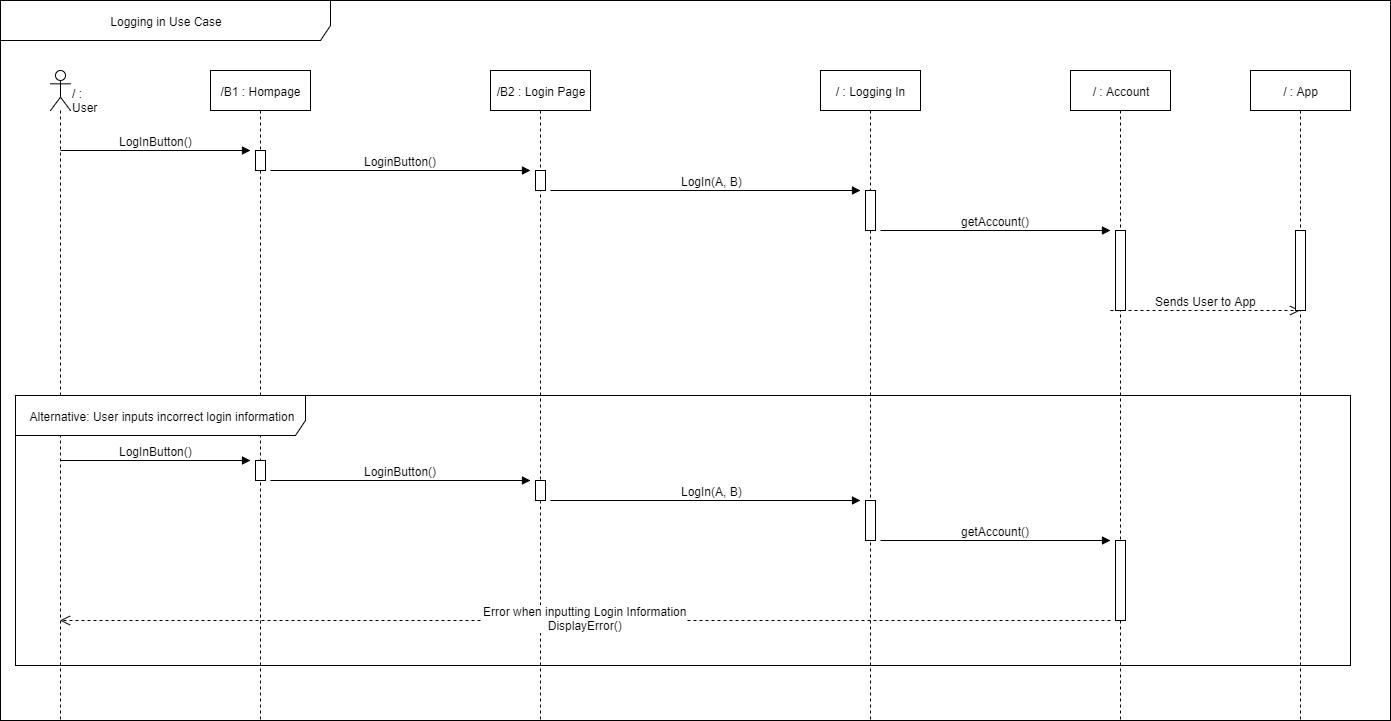
## Appendix B - User Interface Design

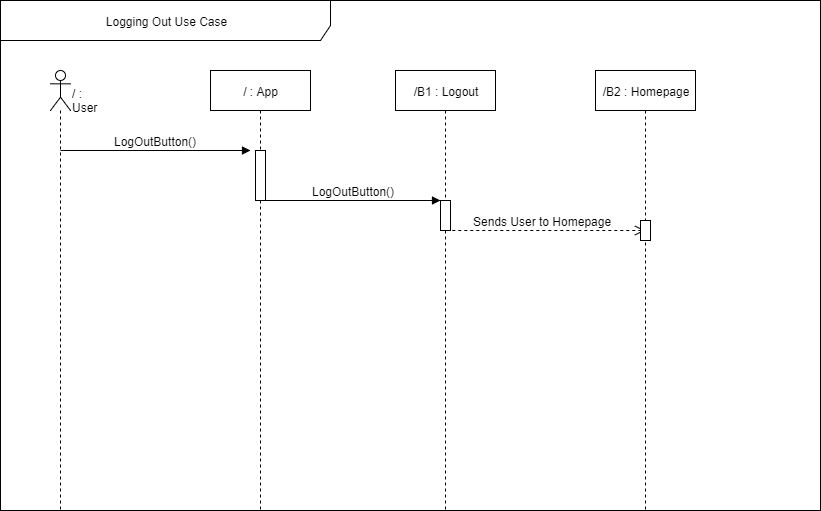


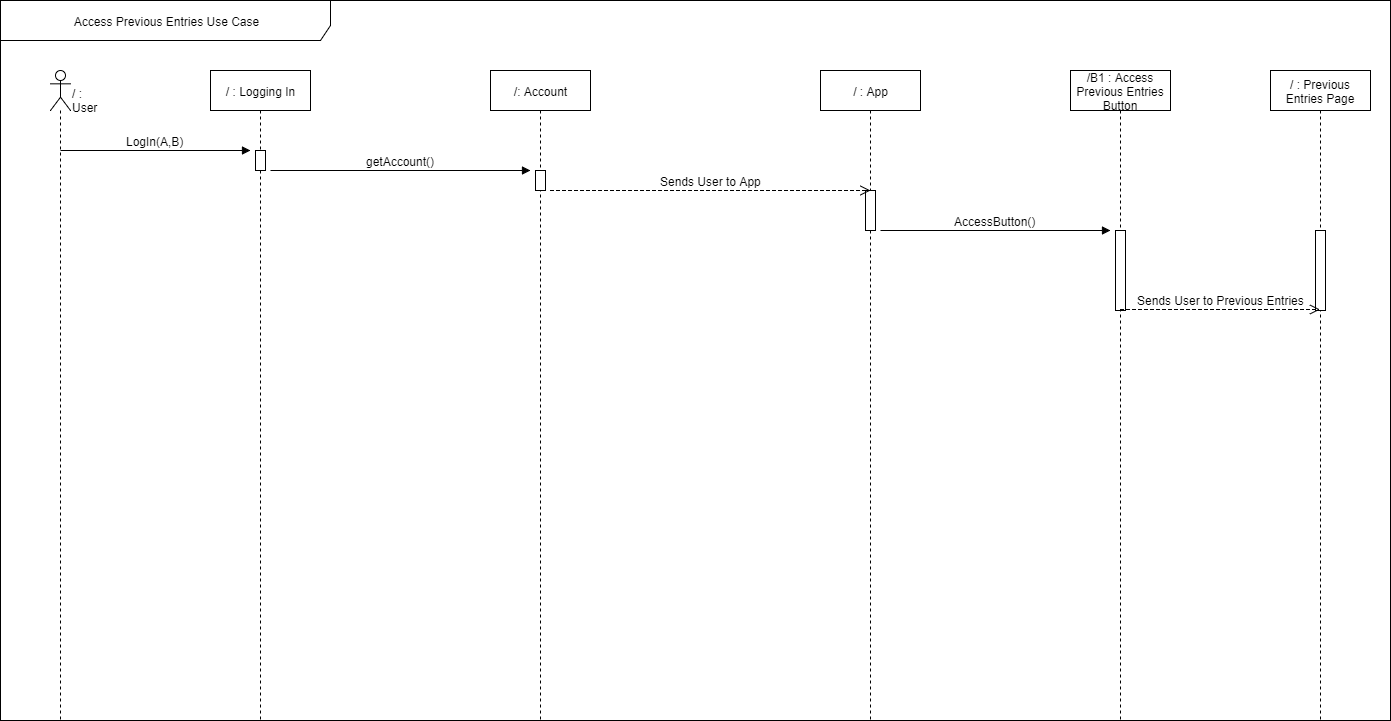












## Appendix C - Sprint Review Reports

**Sprint 2 Review** Meeting Minutes  
   
   
Attendees: Ruben Valdes Jr., Daniel Antonio   
Start time: 7:30  
End time: 8:30  
  
  
  
After presenting our user stories, the product owner accepted all of the following:  
User Story #668 – Login to System  
User Story #670 – Compatibility   
User Story #671 – Creating an Account  
User Story #672 – Logout of System  
User Story #673 – Input   
User Story #677 – Access Prior Inputs   
  
  
  
None were rejected by the product owner and we may proceed with the implementation of said use cases.  
  
  
**Sprint 3** Review Meeting Minutes  
   
   
Attendees: Ruben Valdes Jr., Daniel Antonio   
Start time: 12:00pm  
End time: 1:00pm  
  
  
  
After presenting our user stories, the product owner accepted all of the following:  
User Story #677 Access Prior Input  
User Story #674 Compatibility  
User Story #668 Login to system  
User Story # 671 Make Account  
  
  
  
None were rejected by the product owner and we may proceed with the implementation of said use cases.  
   
   
   
   
   
**Sprint 4** Review Meeting Minutes  
   
   
Attendees: Ruben Valdes Jr., Daniel Antonio   
Start time: 11:00am  
End time: 12:00pm  
  
  
  
After presenting our user stories, the product owner accepted all of the following:  
User Story #681 Continuation  
  
  
  
None were rejected by the product owner and we may proceed with the implementation of said use cases.

## Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

**Shortcomings:**

* The api call
* The ability to logout of the system
* The User profile
* Statistical analysis of success to failure compatibility ratio

**Wishlist:**

* Use the java file to create a service able to be called from the front-end explore page
* Add user profile and populate the view with the data fields from the database
* Implement custom database in java using the java spring framework
* Add data field to user object that holds all questions and answers
* Give user option to put more answer options

# References

<https://www.slideshare.net/ManishKumar1387/design-pattern-in-software-engineering>

<https://www.slideshare.net/RaviYasas/example-for-sds-document-in-software-engineering>

<https://stackoverflow.com/>

<https://www.firebase.com/docs/>

<https://facebook.github.io/react-native/>

<https://github.com/react-community/>