

System Maintenance Document

Team Freeware

**Staff Engagement, Visualisation & Experience Engine (Steve)**

Team members

Ali Mohammed

Alex Mccoy

Duc Hung Tran

Hamza Shaikh

James Worcester

**Lecturer:** Scott Mann

Table of Contents

[Index 3](#_Toc117506176)

[1.0 Introduction 4](#_Toc117506177)

[1.1 System Overview 4](#_Toc117506178)

[1.2 Audience Description 4](#_Toc117506179)

[1.3 Applicability Statement 4](#_Toc117506180)

[1.4 Purpose Statement 4](#_Toc117506181)

[1.5 Document Usage Description 4](#_Toc117506182)

[1.6 Conventions 5](#_Toc117506183)

[2.0 Software Design Scope 5](#_Toc117506184)

[2.1 Major Software Functions 5](#_Toc117506185)

[2.2 Major Design Constraints 6](#_Toc117506186)

[2.3 Major Design Limitations 7](#_Toc117506187)

[3.0 Reference Documents 7](#_Toc117506188)

[3.1 Existing Software Documentation 7](#_Toc117506189)

[3.2 System Documentation 7](#_Toc117506190)

[3.3 Vendor Documentation 8](#_Toc117506191)

[4.0 User Stories 8](#_Toc117506192)

[4.1 User Story Dictionary 8](#_Toc117506193)

[4.2 Iterative User Story Documents 9](#_Toc117506194)

[4.2.1 User Story Definition 9](#_Toc117506195)

[4.2.2 Flow of Interaction Diagram 11](#_Toc117506196)

[4.2.3 Wireframes 14](#_Toc117506197)

[5.0 Object-Oriented Design 19](#_Toc117506198)

[5.1 High Level Architecture Diagram 19](#_Toc117506199)

[5.2 High Level Package Diagram and Components 20](#_Toc117506200)

[5.3 Domain Model and Class Diagram 21](#_Toc117506201)

[5.5 Entity Relationship Diagram and Database Scheme Design 21](#_Toc117506202)

[5.4 Entity Relationship Diagram/Database Scheme Design 22](#_Toc117506203)

[5.5 Sequence Diagram 23](#_Toc117506204)

[5.5.1 View Survey Results 23](#_Toc117506205)

[5.5.2 User Search 24](#_Toc117506206)

[5.5.3 Login 25](#_Toc117506207)

[5.6 Object Dictionary 25](#_Toc117506208)

[6.0 Testing 27](#_Toc117506209)

[6.1 User Story Testing 27](#_Toc117506210)

[6.2 System Test Report 31](#_Toc117506211)

[Glossary 33](#_Toc117506212)

[Appendix 35](#_Toc117506213)

## Index

A

*API* 24

E

entities 25

Entity Relationship Diagram 25

U

**Use Case Diagram** 22

**Use Cases** 23

# Introduction

## 1.1 System Overview

The developed iOS & Android app has been designed to be used as a system for Planit employees to create, administer, answer and view the results of custom surveys assigned to internal teams, and have those results visualised in the form of bar and line graphs. As custom surveys can be created within the app, survey content can be tailored to suit any need, and utilise short answer responses or responses on a scale from strongly agree to strongly disagree. The app also features an employee directory and employee profile screens, as well as a chat channel system where employees can create multi-user chat channels and post to those channels. The app can be accessed worldwide by employees and managers through the internet.

## Audience Description

The app is to be used by employees, managers and administrators to provide survey feedback to Planit, utilise chat channels, create teams and search for employees. Users should have a basic knowledge of how to navigate through apps and run and install apps on iOS or Android devices.

The various users of the system are:

**Administrators**

This includes personnel with the technical knowledge required to maintain and update the system

**Managers and Employees**

These individuals can administer surveys and utilise chat channels and the team and user directories.

## Applicability Statement

This system is designed to be run on any phone that operates with iOS or Android.

## Purpose Statement

The main purpose of this project has been to create an internal app for Planit Testing with the above-mentioned feature set. It assists managers in collecting information from staff, tracks staff experience, and further enhances staff usability and knowledge

## Document Usage Description

**Introduction**

Provides a brief introduction to the project and its purpose for the intended users.

**Software Design Scope**

Covers the major software functions of the program as well as major design constraints and limitations

**Reference Documents**

Consists of supporting existing software documentation, system documentation and vendor documentation.

**User Stories**

This section describes the set of completed user stories in relation to the operation of the project. It includes a user story dictionary, wireframes for the entire app, user story definitions and flows of interaction diagrams.

**Object Orientated Design**

This section describes the fundamental implementation of the concepts derived for the program. It includes: a high-level architecture design diagram, a high-level package diagram, a high-level domain model and class diagram, an entity relationship diagram, high level sequence diagrams and a detailed object dictionary.

**Software Testing**

This section documents user story testing, testing methods, effectiveness and coverage, as well as a usability test report documenting testing methods, effectiveness and coverage.

## Conventions

For this System Maintenance Document (SMD), we have utilized the blue colour font for headings to differentiate between each section. This will help the reader identify each section. Furthermore, a table of contents Is provided to allow easy navigation within this document.

Most of the document is written in size 11 font with Calibri (Body) whilst most headings are written in Calibri Light (Headings).

# 2.0 Software Design Scope

## 2.1 Major Software Functions

**Please note: Due to time constraints, the final build of the app does not utilise different levels of authorisation and access for different users. As such, each user can create, administer, assign, answer and view the results of surveys. They can also create chat channels, view all channels, and post to all channels.**

**User:**

A **user** can:

* Account Management
  + Log in
  + Log out
  + Sign up for an account
  + Confirm their email address
  + Send a password request to reset their password
  + Enter a new password
  + Enter in more details about themselves when logging in for the first time
* Teams
  + Create teams
  + View a list of teams
  + View a list of members of each team
  + Add a user to any team
* Surveys
  + Create custom surveys with five custom survey questions (short answer or on a scale from Strongly Disagree to Strongly Agree)
  + Review created surveys
  + Assign surveys to teams
  + View a list of all surveys
  + Review the contents of any survey
  + View a list of surveys assigned to the currently logged in user
  + Answer assigned surveys
  + View survey results from any number of users for any answered assigned surveys
  + View visualisations of survey results using bar and line graphs for viewing scalar question results
* Chat Channels & Noticeboards
  + Create multi-user chat channels or noticeboards
  + View a list of chat channels/noticeboards
  + View all the posts in each chat channel/noticeboard
  + Create a new post in any chat channel/noticeboard
  + See the name of each user next to each of their posts
* User Directory
  + View a list of all users
  + Click on any user to view their public profile
  + Click on the Profile dashboard item to view your private profile

**System Functions:**

The system should be able to:

* Be reliable
* Function without unhandled errors or crashing
* Create new accounts for users, adding them to Cognito and into the SQL RDS database
* Log in and authenticate users
* Save all created and assigned surveys
* Save each user’s answer to every survey
* Visualise survey question results in graphical form with bar and line graphs
* Save each post to each chat channel and noticeboard

## 2.2 Major Design Constraints

The application is:

* Compatible with iOS and Android
* Configurable
* Engaging
* Fun and easy to use
* Visually appealing
* Able to handle multiple simultaneous users
* Allowing administrators database access
* Allowing administrators the ability to easily migrate the database
* Identifying outliers in survey question answers
* Enabling employees to be happier and more content in the work at Planit, to have a better experience
* Providing a screen/report to view the results of each survey
* Including graphs on the survey reports

## 2.3 Major Design Limitations

The application does not:

* Feature different levels of authorisation and access for different users
* Feature images/photographs for users/teams/surveys/channels or noticeboards
* Allow users to be created/updated/archived or deleted by administrators
* Allow users to be assigned various roles
* Allow users to be assigned to a department
* Feature any form of gamification
* Allow users to subscribe/unsubscribe to channels or noticeboards
* Allow users to become moderators for channels or noticeboards
* Allow users to customise settings in the app
* Provide notifications to users
* Provide a client directory or the ability to create/view/update or delete clients
* Allow users or clients to be assigned an address
* Allow the creation of projects or project metrics
* Provide filters on survey reports to allow the data to be filtered by various characteristics
* Allow survey data to be exported (e.g. CSV/Excel)
* Collect project metrics/metadata from client projects
* Show employee current region, whether they are working from home or the office, or hybrid
* Allow identification of metrics that suggest projects are at risk
* Allow surveys with more or less than five questions to be created
* Allow the use of multiple choice or numerical questions (other then the Agree/Disagree scale)
* Allow the deletion/archiving of surveys
* Allow survey’s to be assigned at a certain frequency (daily, weekly, monthly, quarterly)
* Allow surveys to rotate between specific groups of questions each week or quarter
* Allow anonymous surveys
* Change the name of the app to a gender-neutral name

# 3.0 Reference Documents

## 3.1 Existing Software Documentation

Existing software documentation and mock data was provided by the client in the document ‘La Trobe Uni – Capstone Project 2022 – Steve – Mock Data v1.0.xlsx’. This document demonstrates the desired system requirements and example data.

All other previously produced documents such as the old SMD, previous presentations and previously produced diagrams are eclipsed by this final SMD, final User Guide, final Source Code submission and final presentation.

## 3.2 System Documentation

Additional system documentation is available and provided in the submitted User Manual, submitted Source Code (including readme) and Final Presentation.

## 3.3 Vendor Documentation

* **AWS** - Create, Install, and Operate Websites, Apps, or Services on AWS's Scalable Infrastructure - Link: https://docs.aws.amazon.com/
* **AWS AppSync Developer Guide** – Accelerate application development with serverless GraphQL and Pub/Sub APIs - Link: https://docs.aws.amazon.com/appsync/latest/devguide/appsync-dg.pdf
* **GraphQL** – A query language for your API – Link: https://spec.graphql.org/
* **React Native** – Create native apps for Android and iOS using React - Link: https://reactnative.dev/docs/getting-started/
* **Node.js** – A JavaScript based asynchronous back-end runtime environment - Link: https://nodejs.org/en/docs/
* **MySQL** - The most popular database management solution. Free and open source - Link: https://dev.mysql.com/doc/

# 4.0 User Stories

## 4.1 User Story Dictionary

* **Aus-1: As a user, I want to be able to sign in the App or create a new account if I have not got one yet so that I can successfully log in and use the App**.
* This user story describes the log in screen that is displayed after users launch the App. Users will be able to see login screen and can type in their registered account and password or they can create a new account by using Email and create a new password and then users can successfully sign in.
* **Aus-2: As a user, I want to be able to change the password in case they forgot the current password so that users can keep the account safe.**
* This user story describes the situation in which users forgot their password of the account and want to change to a new one by using their registered email. After successfully changing the password, users can use their new password to smoothly log in to the App.
* **Aus-3: As an admin, I want to be able to create a new Team and see the list of created Teams so that I can manage and organize better**
* This user story refers to the Admin tab screen which is displayed when users want to create a new Team or view the list of created Teams. On the screen, users will see Create Team and View Teams button, if user click on the Create Team button and then type in all the details, they can easily create a new Team and then they can see the created Team appears on the list of Teams when they click on the View Teams button.
* **Aus-4: As an admin, I want to be able to add new members into a Team so that I can manage the workforce effectively.**
* This user story describes how an admin could add new team members to any Team. They can do that by navigating to the list of created Teams and clicking the add new member button to see the list of members and then add any member on the list to that Team.
* **Aus-5: As a user, I want to be able to create a new survey and assign it to any created Team so that all the members in that Team can receive the survey.**
* This user story describes the way a survey can be created by a user. It includes all screens for creating a survey, fill in all the questions and the user also can review the survey and then assign it to any Teams.
* **Aus-6: As a user, I want to be able to see the list of Assigned Surveys so that I can review the completed one or finish any uncompleted.**
* This user story refers to a screen which displays the list of Assigned Surveys. It includes all the details of any survey and users can click on it to see whether that survey was completed or not
* **Aus-7: As a user, I want to be able to view the list of chat channels and access any channel to view the messages so that I can interact effectively with everyone on the chat.**
* This user story describes how users navigate to the Chat tab and then they can see all the chat channel they are involved in. After users click on a channel, the screen will display all the content in that channel.
* **Aus-8: As a user, I want to be able to type and send my messages onto any chat channel so that everyone in that channel can see my messages.**
* This user story refers to the screen which displays the area for typing in message and sending it to the chat channel so everyone can see all the message content.

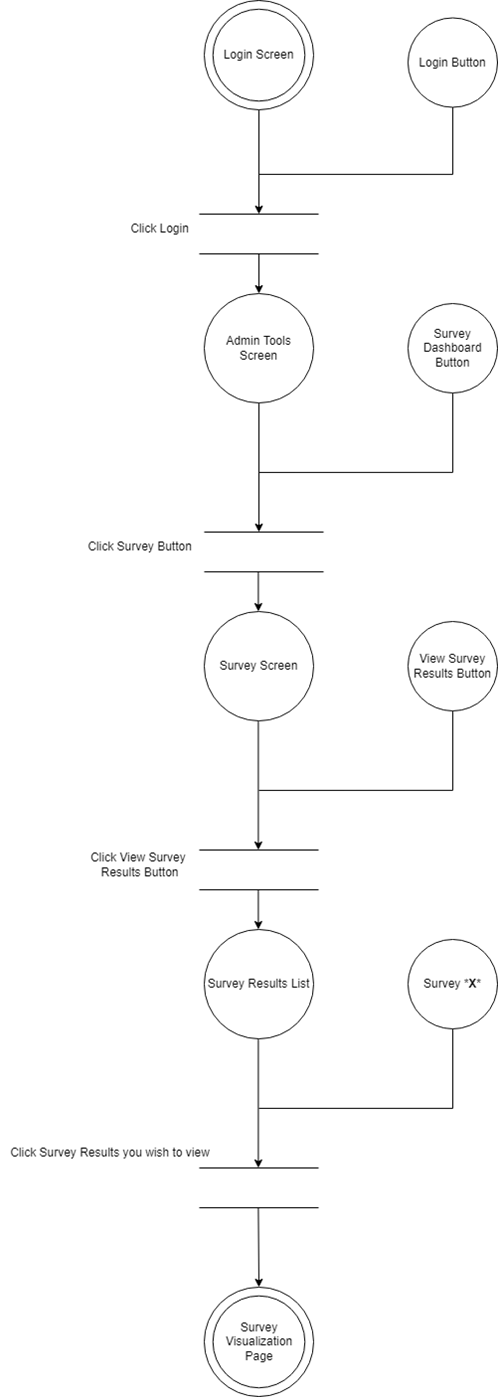
## 4.2 Iterative User Story Documents

### 4.2.1 User Story Definition

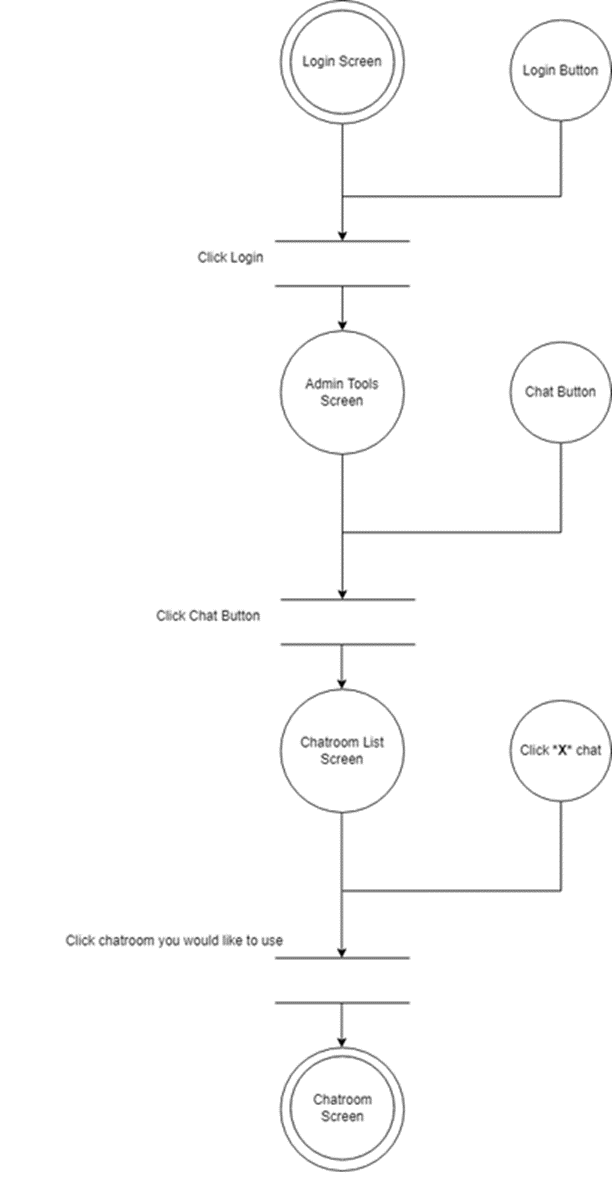
|  |  |  |  |
| --- | --- | --- | --- |
| User Story ID | User Story | Acceptance Criteria | Story Point |
| Aus-1 | As a user, I want to be able to sign in the App or create a new account if I have not got one yet so that I can successfully log in and use the App. | -Users can clearly see the Sign in screen after launching the App  - Users can successfully log in the App by typing in correct details  - Users can navigate to Sign Up screen if they click on Create new account | 6 |
| Aus-2 | As a user, I want to be able to change the password in case they forgot the current password so that users can keep the account safe. | -Users can easily create a new password by using their Email and receive the verification code | 3 |
| Aus-3 | As an admin, I want to be able to create a new Team and see the list of created Teams so that I can manage and organize better | -The screen which displays the list of created Team successfully pops up after user click on the View Team button  - User can smoothly create a new Team by adding members to it | 9 |
| Aus-4 | As an admin, I want to be able to add new member into a Team so that I can manage the workforce effectively. | -Users can click on any Team on the list of created Team to see the list of added members and can click the add new member to add a new member | 6 |
| Aus-5 | As a user, I want to be able to create a new survey and assign it to any created Team so that all the members in that Team can receive the survey. | -Users can navigate smoothly to create a new survey by clicking on the Create new survey button  -Users could choose the type of the question and type in the content of the question  -Users can review the whole survey and assign it to any Team they want | 12 |
| Aus-6 | As a user, I want to be able to see the list of Assigned Surveys so that I can review the completed one or finish any uncompleted | -After navigating to home tab, users can see all the assigned survey appear nicely  -Users could click on any survey to complete it smoothly. | 9 |
| Aus-7 | As a user, I want to be able to view the list of chat channels and access any channel to view the messages so that I can interact effectively with everyone on the chat. | -After navigating to Chat tab, users can view the list of all chat channels they are involved in.  -Users can click on any chat channel to see all the content inside that. | 9 |
| Aus-8 | As a user, I want to be able to type and send my messages onto any chat channel so that everyone in that channel can see my messages | -After choosing a channel, users can clearly see the space for typing new message and click to type on it  -Users can smoothly send the message and the message will be displayed immediately on the chat screen | 9 |

### 4.2.2 Flow of Interaction Diagram

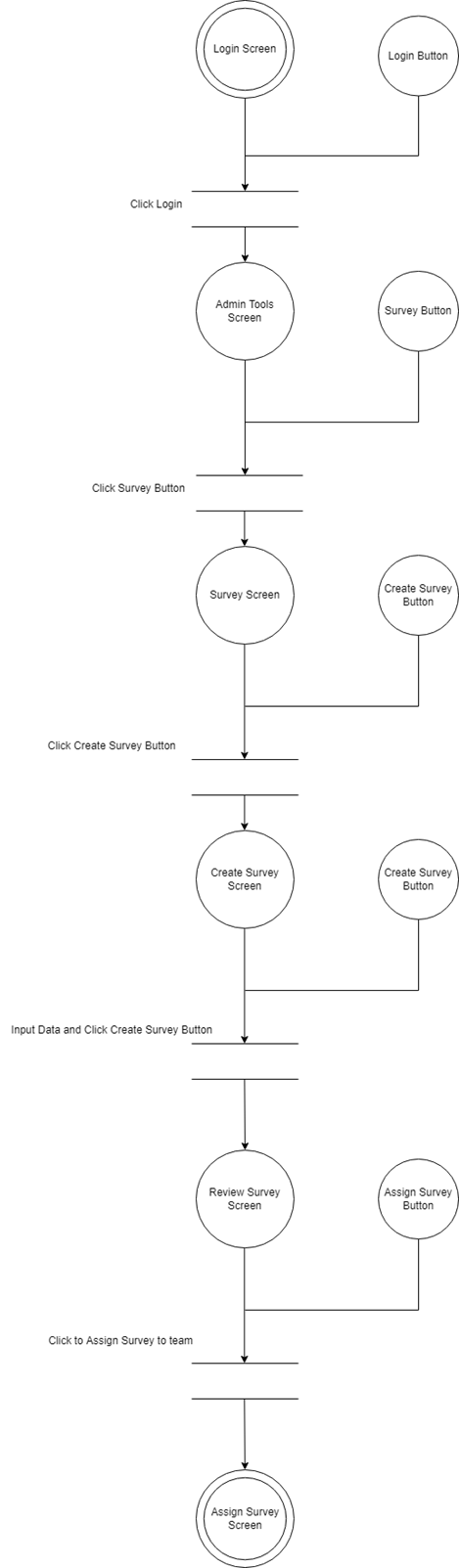
**Viewing Survey Visualizations**



**Entering Chatroom for Messaging**

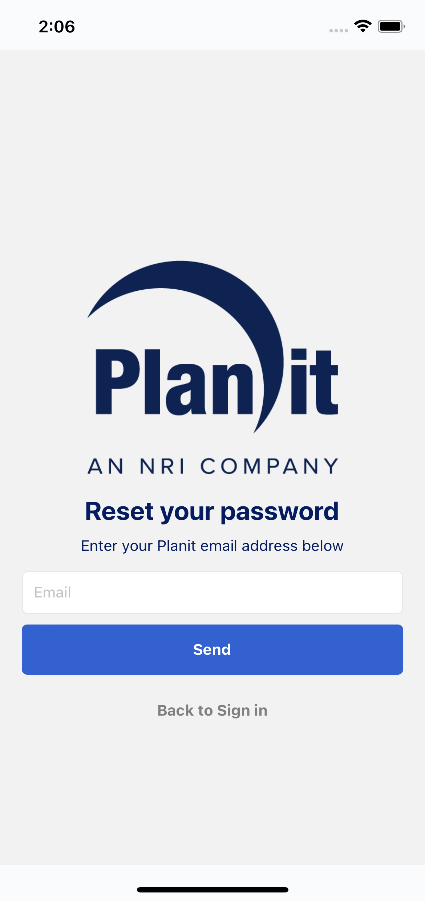
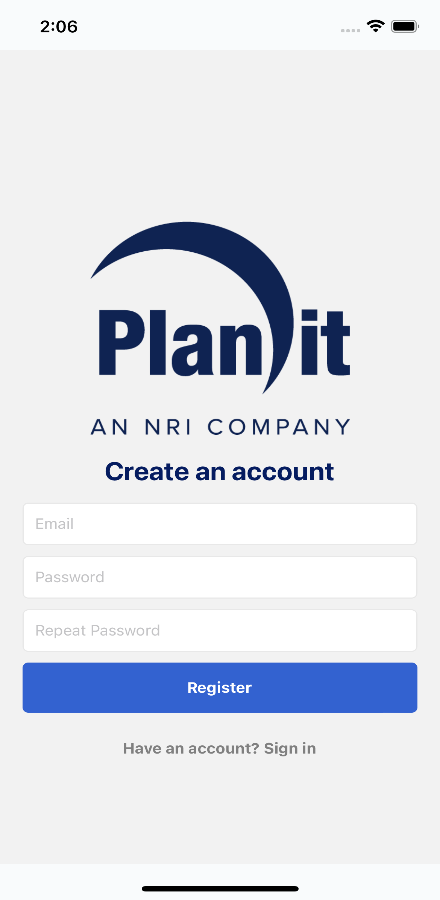
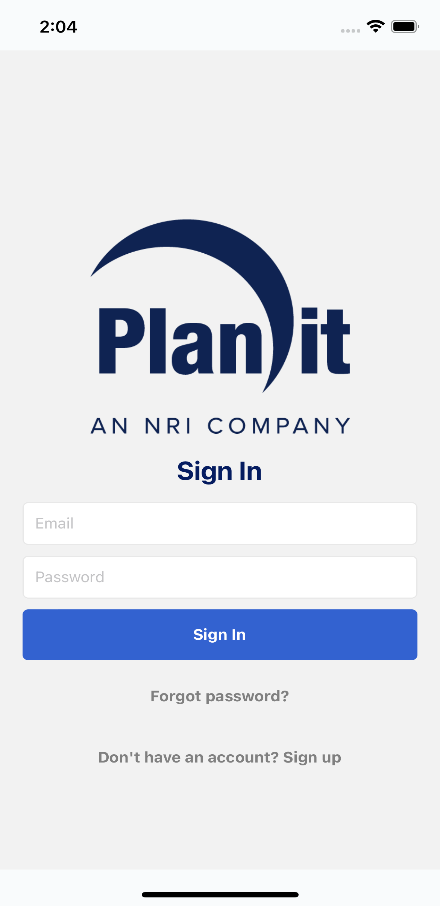


**Creating a Survey to Assign it to a team**

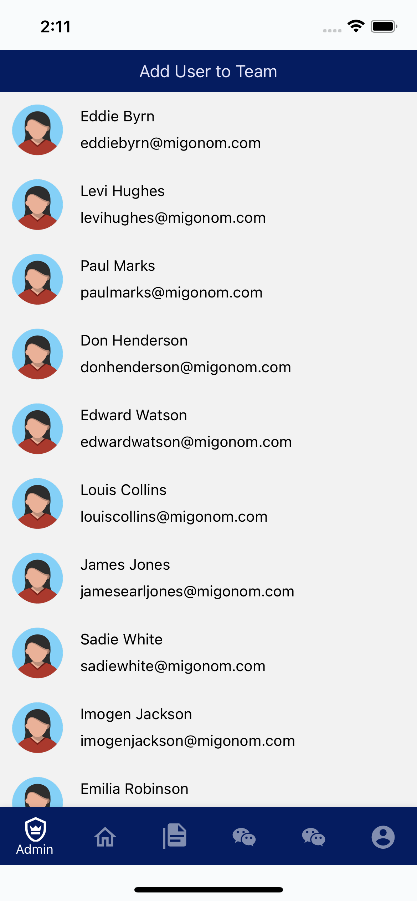
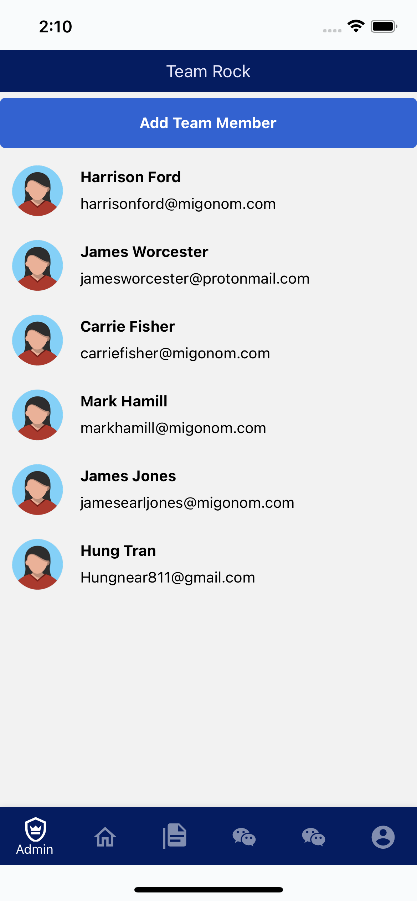
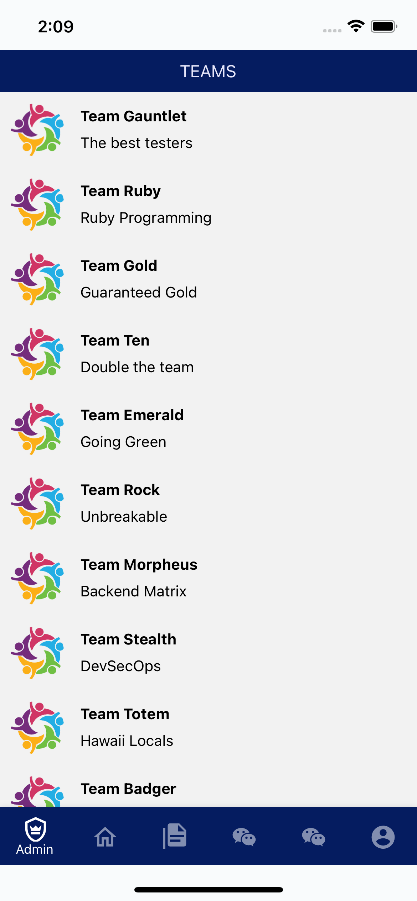
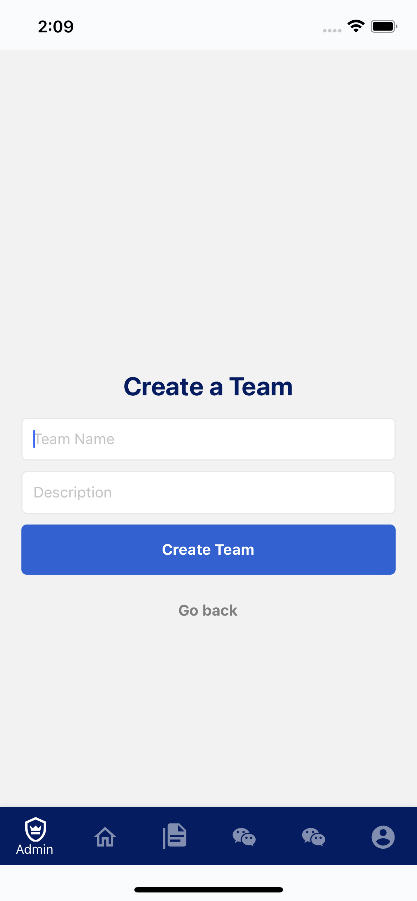
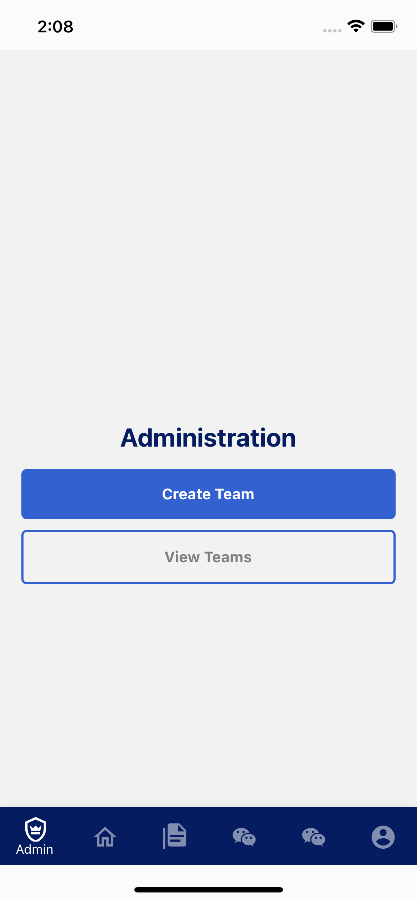


### 4.2.3 Wireframes

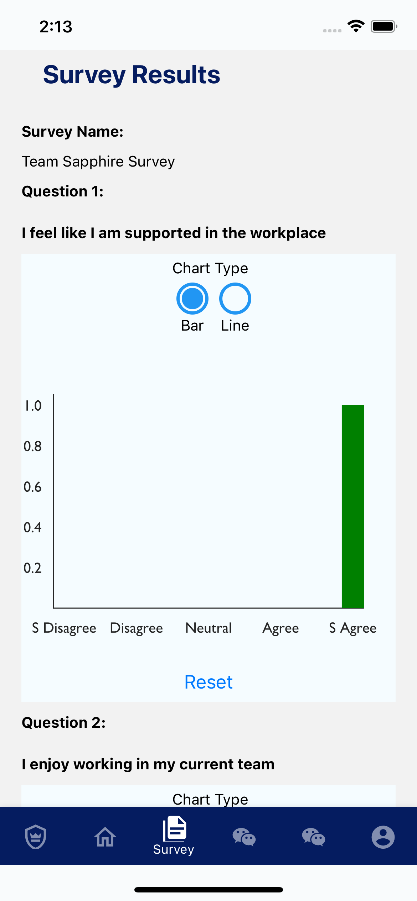
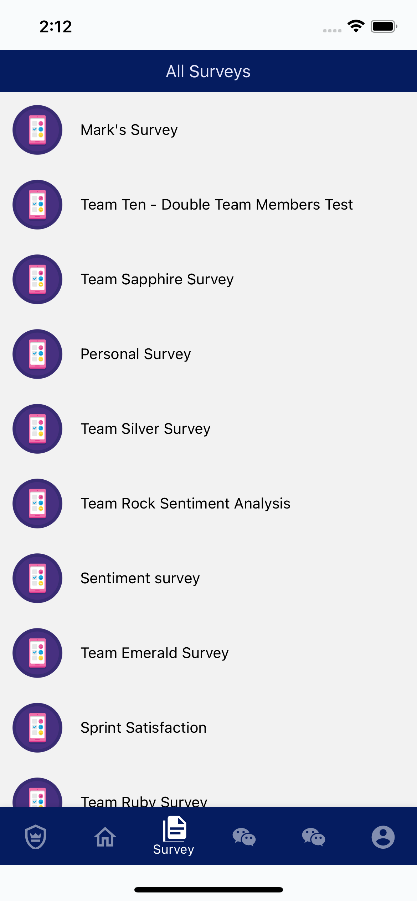
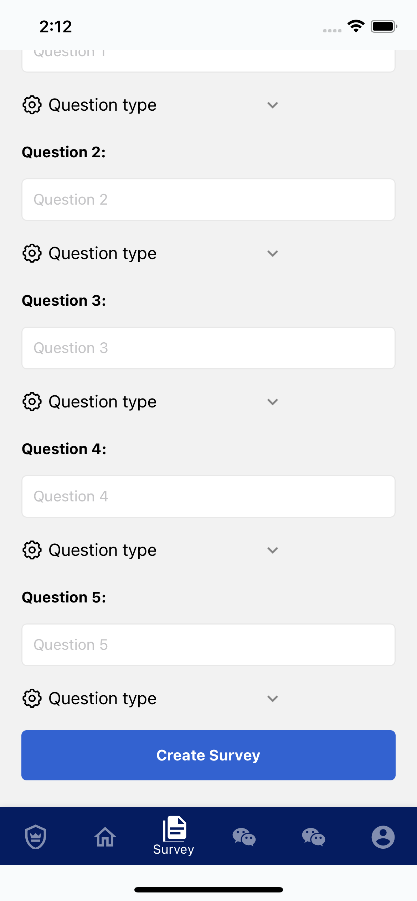
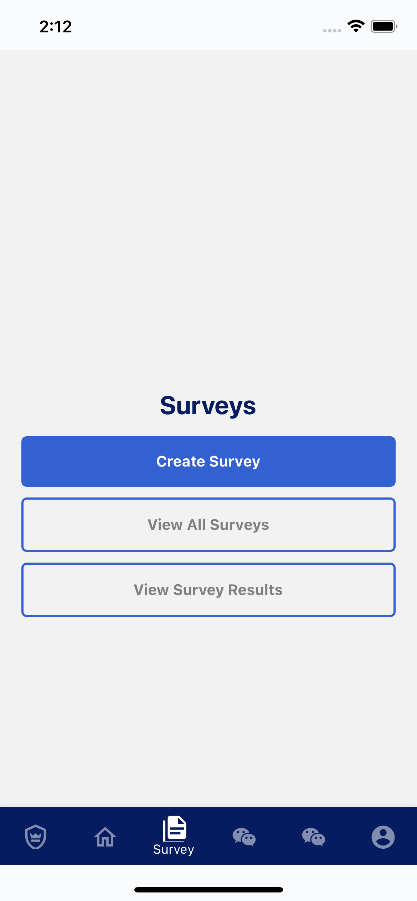
**Login Function Screens:**



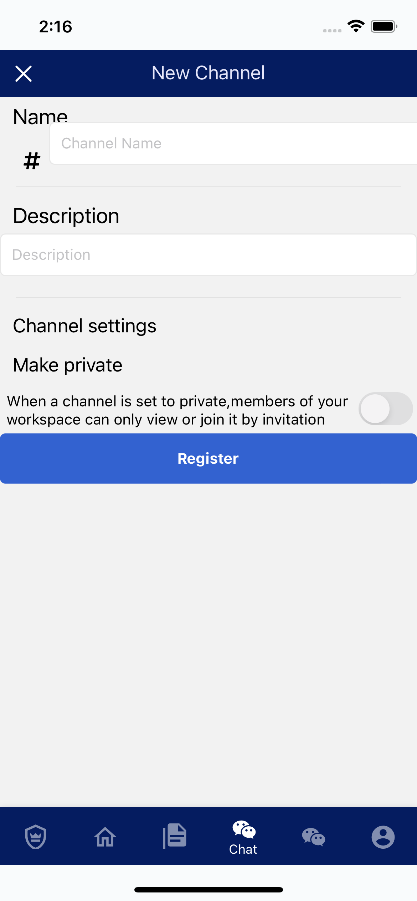
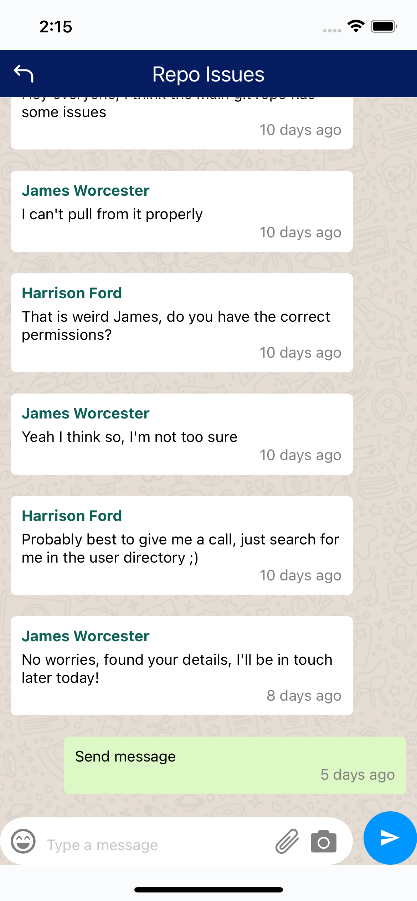
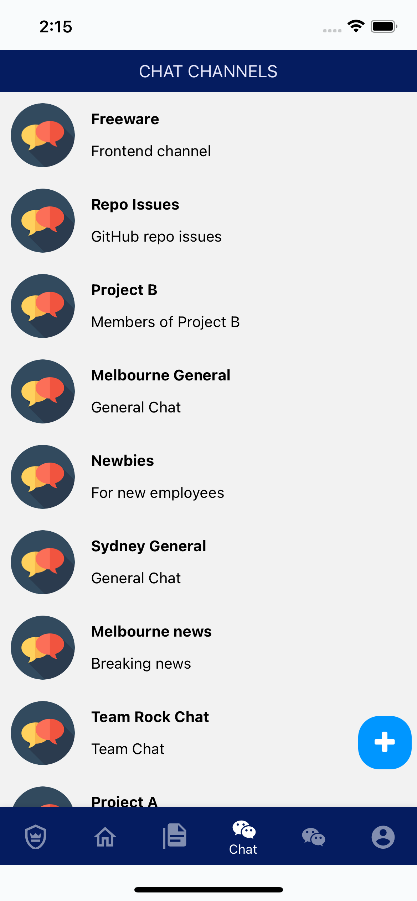
**Team Screens : Create a new team, view all the created teams and add members to team.**



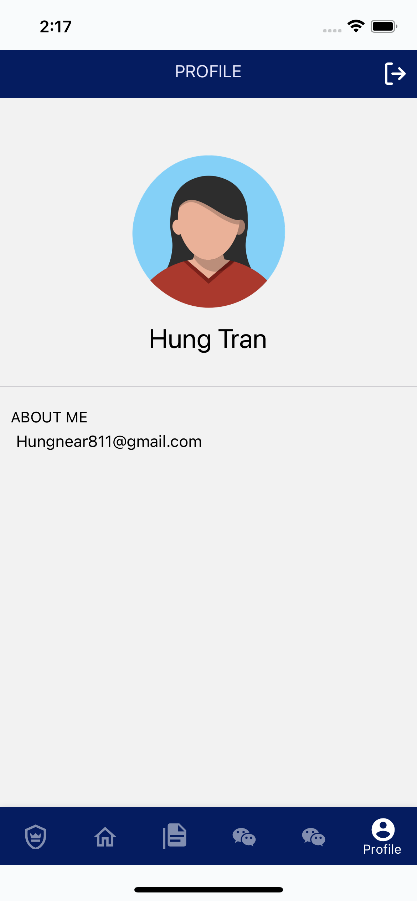
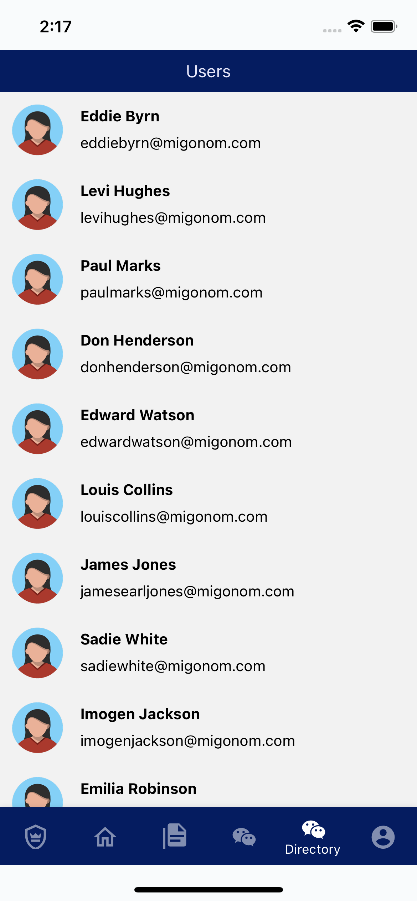
**Survey Screens:**



**Chat Function Screens:**



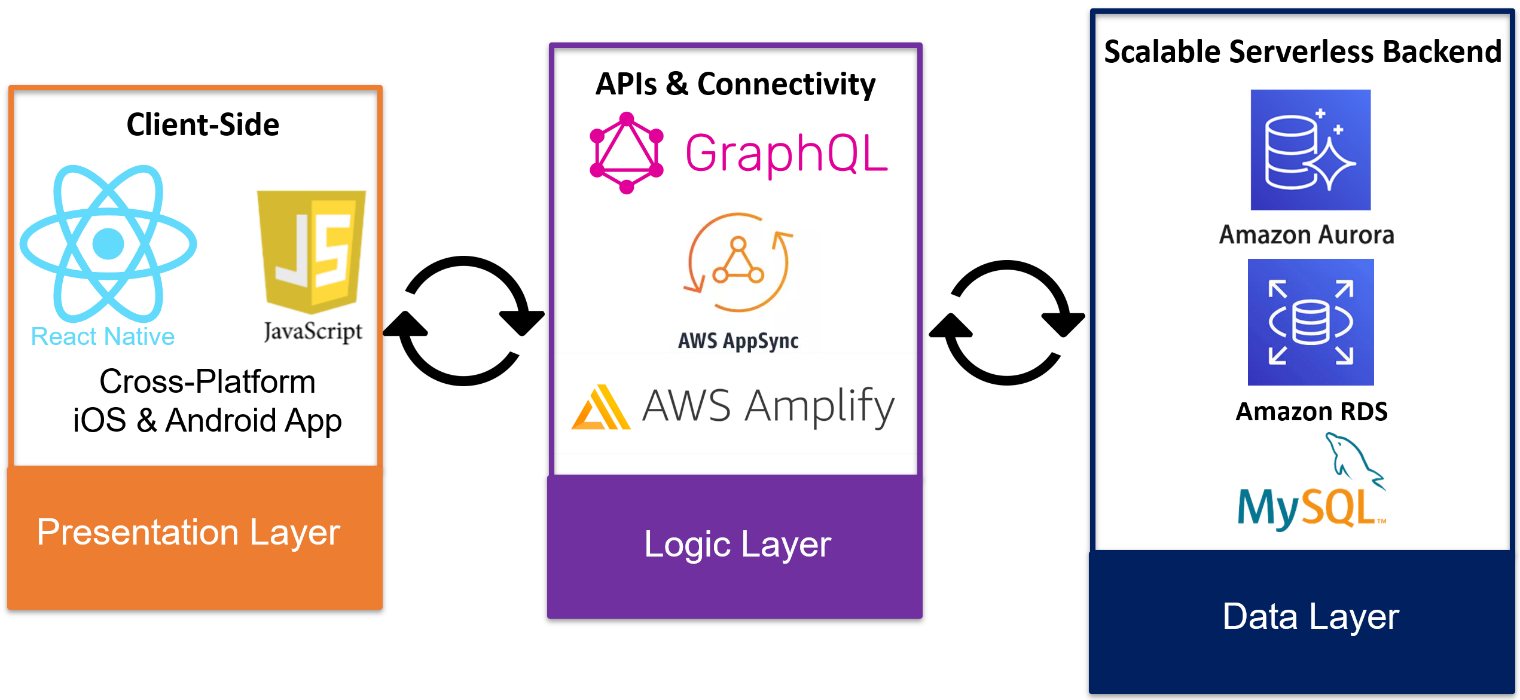
**Users Directory and Profile:**



# 5.0 Object-Oriented Design

## 5.1 High Level Architecture Diagram

**Note: a larger copy of this diagram has been attached with the SMD submission**



## 5.2 High Level Package Diagram and Components

**Note: A larger copy of this diagram has been attached with the SMD submission**

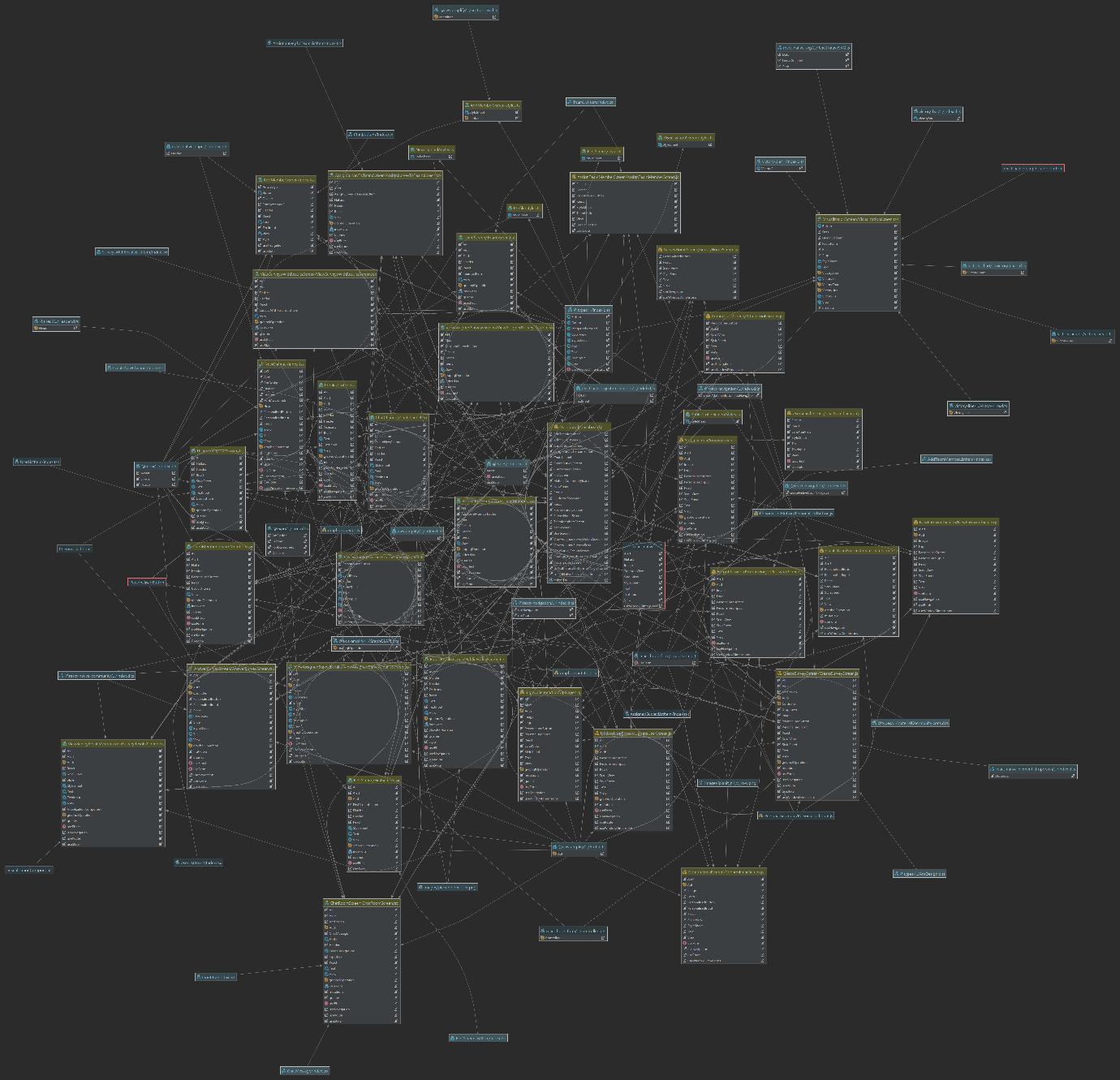
Diagram

Description automatically generated

## 5.3 Domain Model and Class Diagram

A class diagram is a form of static structural diagram that depicts the organization of a system by displaying its classes, properties, methods, and object connections. This diagram demonstrates all classes and their relationships in the /src/screens directory of the project.

**Note: A larger copy of this diagram has been attached with the SMD submission**



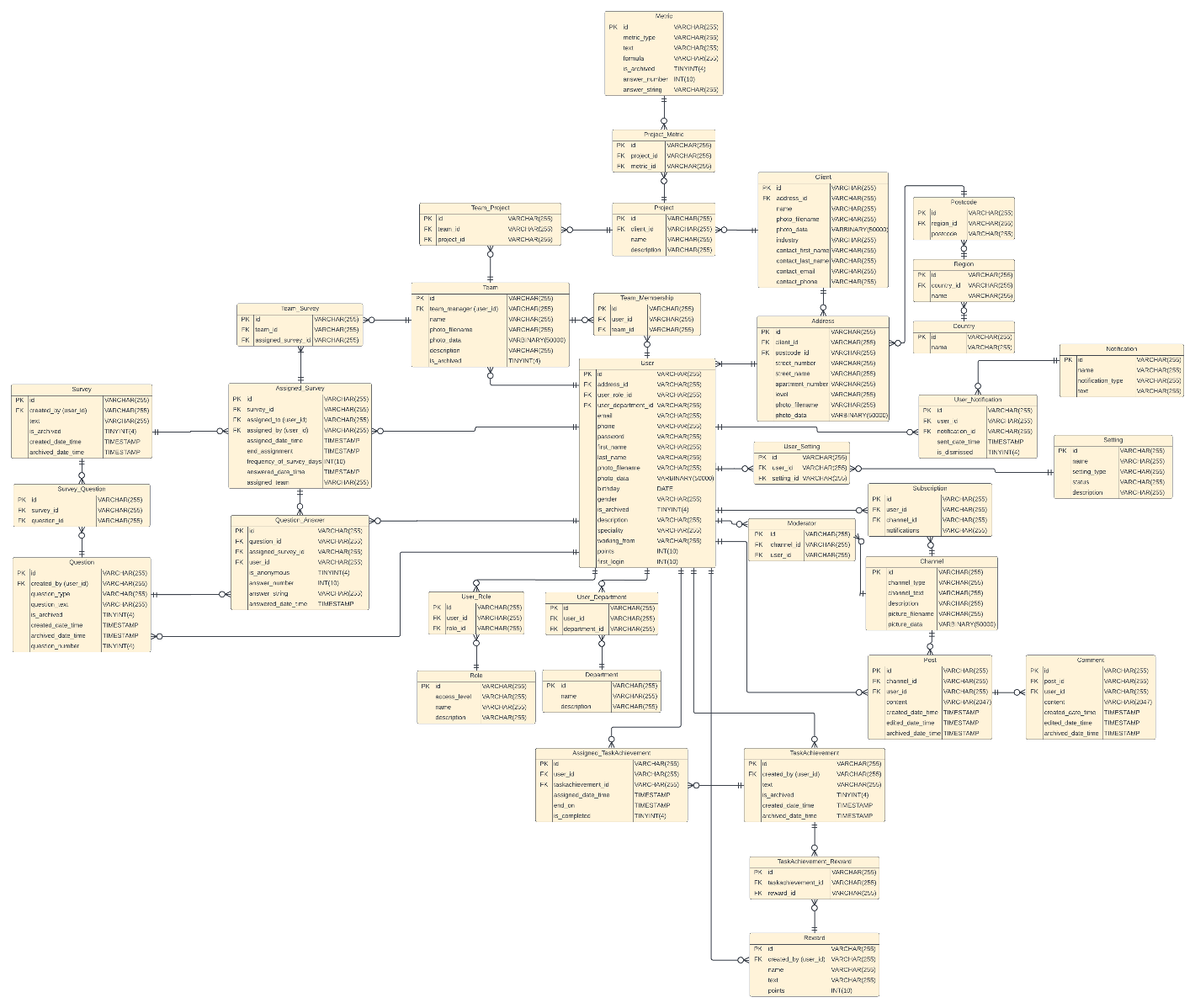
## 5.5 Entity Relationship Diagram and Database Scheme Design

A Physical Database Entity Relationship Diagram (ERD) is a way of modelling and describing the entities, attributes, data types and relationships between entities to directly implement them in a relational database using SQL.

**Note: A larger copy of this diagram has been attached with the SMD submission**

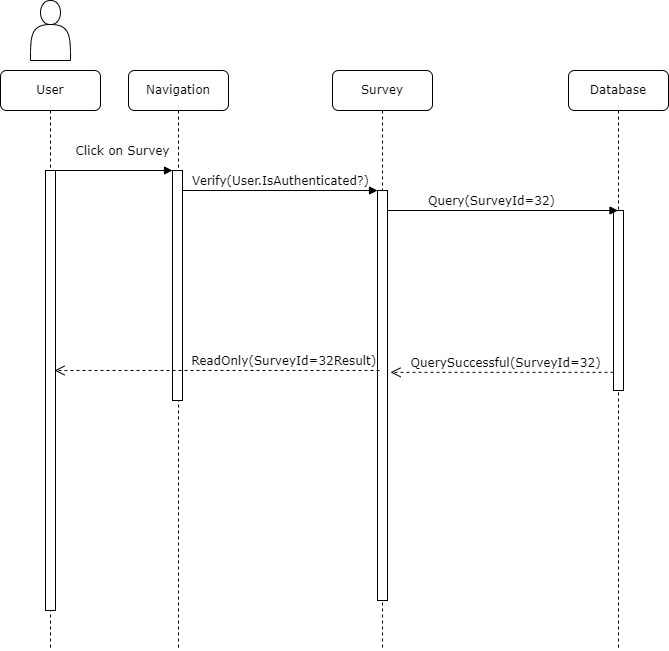
## 5.4 Entity Relationship Diagram/Database Scheme Design

**Note: A larger copy of this diagram has been attached with the SMD submission**



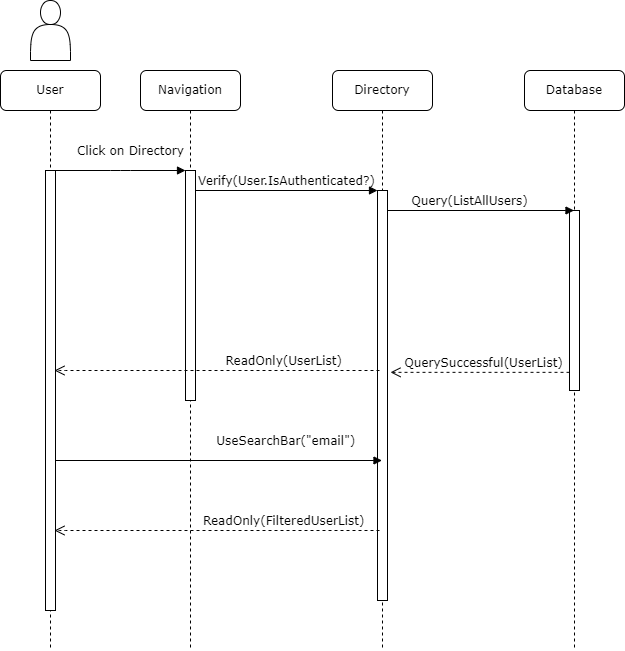
## 5.5 Sequence Diagram

### 5.5.1 View Survey Results



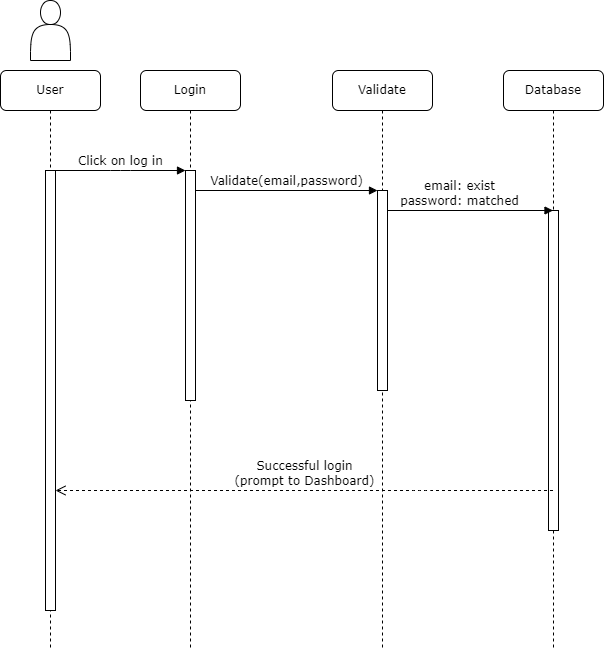
The diagram above shows how an authenticated user can view a survey results. The process starts when the user navigates to the survey page and select a survey from the given list. The program will check if the listed survey still exists in the database and make a query. Furthermore, the database will return the results to the survey page. In which, it will be displayed in a read only format to the user.

### 5.5.2 User Search



The diagram above shows how an authenticated user can search for a particular user in the directory. The process starts when the user navigates to the directory page. The program will check if the user is authenticated and will send a query to the database requesting a list of all users. The database will return the results to the directory page which will then display it for the user in a read only format. The user can use the search bar function to filter the list. This is important because the filtering process has nothing to do with the database, but it is done locally on the user device. The user needs to enter the email address to start the search function. The filtered list will be displayed to the user in a read only format.

### 5.5.3 Login



Unlike the above diagrams, this process starts when an unauthenticated user tries to log in to his or her account. The user clicks on log in which will send the users email and password to validate against the records in the database. If the user email exists and the password matches with the record. The user will be logged in successfully and will be prompt the dashboard page.

## 5.6 Object Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Object** | **Package** | **Type** | **Description** | **Signature** |
| onSignInPressed() | Users | User Management | Method | Sign in Authentication, retrieves/stores the User ID, does not allow if they have not signed up | onSignInPressed(getUser, username, current\_user) |
| onForgotPasswordPressed() | Users | User Management | Method | AWS Amplify is used to send password reset request to users email. | onForgotPasswordPressed(username, current\_user) |
| onSignUpPressed() | Users | User Management | Method | Navigates to Sign up Screen | onSignUpPressed() |
| onRegisterPressed() | User | User Management | Method | Tests database connection, AWS Amplify is then used to save the username and password | onRegisterPressed(userSub, userDetails) |
| createNativeStackNavigatior() | Navigation | Navigation | Attribute | Provides customizable navigation stack that holds all of our screens in one place | createNativeStackNavigation(AdminToolsStack, HomeScreenStack, SurveyHomeScreenStack, ChatChannelStack, UserScreenStack, navigation()) |
| tabBarIcon() | Navigation | Navigation | Attribute | Multiple icons for navigation around the dashboard | tabBarIcon(file-document-multiple, wechat, account-circle, shield-crown-outline, home-outline) |
| onViewTeamsPressed() | Navigation | Navigation | Method | Navigates to team page | onViewTeamsPressed() |
| onCreateTeamPressed() | Navigation | Navigation | Method | Navigates to team creation page | onCreateTeamPressed() |
| onCreateTeamPressed() | Teams | Team Management | Method | Creates team | onCreateTeamPressed(name, description, teamDetails, createTeamAdmin) |
| onViewTeamsPressed() | Navigation | Navigation | Method | Navigates to page to see what teams are created | onViewTeamPressed() |
| onBackPressed() | Navigation | Navigation | Method | Navigates to previous page | onBackPressed() |
| TeamScreen() | Teams | Team Management | Method | View list of teams in array | TeamScreen(teamData, getTeams, listTeams) |
| ViewAssignedSurveysScreen() | Survey | Survey Management | Method | View list of assigned surveys in an array | ViewAssignedSurveysScreen(uniqueSurveys, surveyData, surveyDataStripped, getSurvey, getSurveys) |
| onCreateSurveyPressed() | Survey | Survey Management | Method | Navigate to survey creation screen | onCreateSurveyPressed() |
| onViewSurveysWithResultsPressed() | Survey | Survey Management | Method | Navigate to survey results screen (visualization) | onViewSurveysWithResultsPressed() |
| onViewAssignedSurveysPressed() | Survey | Survey Management | Method | View all created surveys | onViewAssignedSurveysPressed() |
| onCreatePressed() | Survey | Survey Management | Method | Creates survey to assign to individuals | onCreatePressed(question1\_id, question1\_text, question1\_type \*\*repeated for other questions\*\*, createQuestion, username, createSurvey, q1selectedType \*\*repeated for all questions\*\*, survey\_id, survey\_text) |
| Chatchannel() | Chat | Chatroom Functionality | Method | Creation of functioning chatrooms that can be created, edited and joined. | Chatchannel(channel, setChannel, getChannels, channelData, listChannels) |
| ChatRoomScreen() | Chat | Chatroom Functionality | Method | Chat room screen | ChatRoomScreen(post, setPost, myId, setId, channel\_id, listPostsByChannel, postData, listPostsByChannelWithName) |
| createNewPost() | Chat | Chatroom Functionality | Method | Creates new message in chat room | createNewPost(channel\_id, user\_id, myId, content) |
| PublicProfileScreen() | User | User Management | Attribute | User’s profile page | PublicProfileScreen(userDetails, userFirstName, userLastName, userEmail, getUser) |
| backgroundColor() | Formatting | General | Attribute | Sets background color of any visual feature | backgroundColor() |
| fontSize() | Formatting | General | Attribute | Changes font size of text | fontSize() |
| keyExtractor() | General | General | Attribute | Assigns id for key’s across entire program instead of default property | keyExtractor() |

# 6.0 Testing

## 6.1 User Story Testing

Objectives of testing scenarios:

1. To detect errors/inconsistencies and design issues when the application is being used on the mobile device.

2. To be sure the application is running on both android and apple devices and to ensure it also meets the requirements that were stated in the project description.

Participants:

We conducted a testing day consisting of people randomly selected from the La Trobe University Agora. We will aim to select participants with different abilities and backgrounds regarding their understanding of technology. To the test the complexity of the app we will have tests taken from different ages and different degrees.

Metrics

Participants will undergo simple tasks that a user of the app would usually do. For the given user story, the application will be evaluated on how easy it was for the participant to achieve their goal, at what speed they achieved it, and the accuracy of the application's outcome is. Participants will also be asked to rate the application out of five regarding ease of use.

*Methods:*

Participants will be given a brief introduction into the application's purpose and will then be expected to use the application to execute some of the more important user stories that can be performed on the device.

Scenarios:

Scenario 1: sign up and create an account and then view the main login page

Step 1: Create an account.

Step 2: Login to the account that was created.

Step 3: Be able to view the home page.

Successful completion: User should be able to successfully create an account using their email and should successfully sign in.

Scenario 2: Create a team and view the team on the team’s page

Step 1; Press the “create team” button.

Step 2: Fill in the information for the “team name” and “description”

Step 3: Click on “create team” and then go back to home page and click “view teams” to see the The team you have created

Successful completion: User should be able to create a team of their desire. User should be able to pick which members they would like to join the team and should be able to pick the team’s name. User should be able to view their team on the team’s page.

On a scale of 1 to 5 from Strongly Disagree to Strongly Agree, can you answer the following questions:

|  |  |  |
| --- | --- | --- |
| #Q |  | Score |
| 1 | I think that I would like to use this app frequently | P1: 5  P2: 5  P3: -  P4: 3  P5: 4  P6: 4 |
| 2 | I found this app to be unnecessarily complex | P1: 2  P2: 1  P3: 2  P4: 2  P5: 2  P6: 1 |
| 3 | I thought this app was easy to use | P1: 5  P2: 5  P3: 4  P4: 4  P5: 5  P6: 4 |
| 4 | I think I that I would need the help of a support person to use this app | P1: 1  P2: 2  P3: 3  P4: 2  P5: 1  P6: 2 |
| 5 | I found the various functions in this app were well integrated | P1: 4  P2: 4  P3: 3  P4: 4  P5: 4  P6: 5 |
| 6 | I thought there was too much inconsistency | P1: 2  P2: 2  P3: 2  P4: 1  P5: 4  P6: 2 |
| 7 | I would imagine that most people would learn to use this app very quickly | P1: 5  P2: 5  P3: 3  P4: 4  P5: 4  P6: 5 |
| 8 | I found this app very cumbersome to use | P1: 3  P2: 1  P3: 2  P4: 1  P5: 2  P6: 1 |
| 9 | I felt very confident using this app | P1: 4  P2: 4  P3: 2  P4: 3  P5: 5  P6: 5 |
| 10 | I needed to learn a lot of things before I could get going with this app | P1: 1  P2: 2  P3: 2  P4: 2  P5: 1  P6: 2 |
| 11 | I would trust this app with my personal information | P1: 4  P2: 1  P3: 3  P4: 3  P5: 1  P6: 4 |
| 12 | This feels like a professionally developed app | P1: -  P2: 4  P3: 4  P4: 3  P5: 4  P6: 4 |
| 13 | I found the overall theme and colour of the app to be consistent and pleasing | P1: 4  P2: 4  P3: 2  P4: 3  P5: 3  P6: 5 |
| 14 | It feels like a lot of effort has gone into developing this | P1: 4  P2: 4  P3: 2  P4: 4  P5: 5  P6: 5 |
| 15 | The app feels familiar when compared to the design of other chat apps you use | P1: 4  P2: 4  P3: 1  P4: 3  P5: 5  P6:5 |

## 6.2 System Test Report

Introduction:

To ensure the app has no defects for user stories that are implemented, a report has been created that defines the testing procedures that have been taken to test the system.

Test Plan:

Below is a list of tests that are to be carried out to ensure the system is functioning as expected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test numbers** | **Test description** | **TEST DATE** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS / FAIL** | **ADDITIONAL NOTES/improvements** |
| 1.) | Put in my email credentials and try login into the app home page | 1/09/2022 | To be able to login and view the homepage of the app | Success. I was able to login securely and view the app homepage using my credentials | pass | -required letters for password is long and can be hard to remember  -option to save email and password for future login |
| 2.) | Creating an account and signing up for the app | 1/09/2022 | To be able to create an account using my email | Account was successfully created, and verification process was also included | pass | -two verifications codes were sent. First one did not work |
| 3.) | Answer the survey questions and view my results | 8/09/2022 | To be able to view my results and see what they mean | I was able to answer the questions however there was no actual question to be answered. Question had to be made up myself | Neutral | -No back button on survey page. Once survey is opened, I was stuck there and could not navigate back to home page |
| 4.) | Testing the usability of the survey page | 8/09/2022 | To be able to navigate smoothly in and out the survey home page | Was able to navigate in the survey page but not able to navigate out | neutral | - |
| 5.) | Testing chat channels and the functions in the channel | 10/09/2022 | To be able to type in the chat and clearly view who I am talking to | Was able to smoothly navigate in and out of chats. Chats also had a back button | pass | Chat channel looks very clean and precise. Seems like a real app  -no button to send attachments if user wishes to |
| 6.) | Testing chat channels and testing specific chats | 24/09/2022 | To be able to type messages and send them in the chat channel | was able to type the message but could not send them. Produced an error | fail | chat channel looks very clean however I was not able to send my message.  - When you press enter once it does not allow you to send your message |
| 7. | create a team's channel that I create | 24/09/2022 | to be able to create the team's channel with my desired team | i was able to create the team successfully and view it in the team's channel | pass | - no back button to get out of teams. |
| 8.) | Testing chat channels functionality | 24/09/2022 | to be able to press enter on the chat and go a line below and then send message | i was able to hit enter and go below In my chat but once I did that I was not able to send the message | fail | -no option to send message if I press enter.  - comes up with error |

## Glossary

**Android**

A mobile operating system based on the Linux kernel and created and licensed by Google for various devices.

**API**

In computer programming, an Application Programming Interface (API) is a connection between two computers or computer programs that enables applications to share and exchange data and functionality securely.

**Entity**

Regarding databases, an entity is a thing, person, place, unit, item or object that various data attributes can be collected and stored about. An example would be having a person as an entity and wanting to store their first name, last name, phone number and email address.

**Entity Relationship Diagram (ERD)**

Regarding databases, an entity relationship diagram (ERD) is a way of modelling and describing the entities, attributes, data types and relationships between entities so that they can be implemented into a relational database

**Gamification**

The adding of games and video game mechanics to things that don’t traditionally utilise them to encourage use and engagement, such as adding rewards, achievements and progression to an app, website or utility.

**iOS**

A mobile operating system created by Apple exclusively for their iPhone and iPod touch devices.

**Use Case**

A scenario-based technique that identifies actors and their interactions with a system and the sequence of events required to achieve a particular goal.

**Use Case Diagram**

A type of diagram in UML designed to visualise actors and all the required use cases and functionality that actor needs to interact with the system.

**User Story**

An informal one sentence description of features of a software system written in natural language (English) from the perspective of a user of the system. A typical user story format would be “As a <role> I want to be able to <capability>, so that I can <benefit>”

## Appendix

An end of project requirements analysis. All the sections highlighted in green were successfully completed and delivered upon, the sections highlighted in yellow could be completed relatively easily if the project were to continue, and the sections in white indicate more complex features that would take significant time to implement

**Note: A larger copy of this diagram has been attached with the SMD submission**

