InPower Admin Portal and Mobile Application

TECHNICAL REPORT

Start Date: **24/04/2024**

Authors : **Lucas Ting, s3934181**
Robert Kennedy, s3841073
Dinura Kularatne, s3901134
Youthtakak Mam, s3851384
Charles Blyton, s3407778

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This report provides a comprehensive overview of our development of two of InPower's projects, namely the admin portal of their website developed using Laravel – a PHP-based web framework – and the development of their social media app on both Android and iOS systems.

The report highlights how we started development on the admin portal of the client's website, which included adding in new features such as managing users, managing groups, and other features. To do this implementation, we first had to get the existing application and its functionalities all up and running, which took a bit of time considering our lack of experience in Laravel. Once we had it running, we implemented some dummy data to see and use the existing functionalities to build upon but decided that the scope of the project seemed too large, and the client was happy to change our project to the social media application.

The social media application involved us splitting into separate groups to work on the Android and iOS sides, where two people worked on the iOS and three on the Android. There was also another student team working on them too, where they tackled some other screens to develop.

This report goes through the functional and non-functional requirements that were essential for both phases of the project. It elaborates on the architectural design of both the Laravel and social media applications, highlighting both the architectural designs of the iOS and Android apps. Also highlighting the technical frameworks of these.
The report also highlights the implementation process for each stage, and any challenges and solutions that we came up with along the way, through the testing process.

In conclusion, this report serves as a detailed account of the project's lifecycle, from initial planning and requirement gathering to implementation, testing, and deployment, providing valuable insights to the client, and the RMIT team to reflect on.

1 INTRODUCTION

The initial objective of this project was to enhance the functionalities of InPower's existing administrative portal using a web-based programming language. However, due to technical challenges, we shifted our focus to developing new features for InPower's web application. We were tasked with implementing the designs for pages that the client had ready for us on Figma. These pages include sign-up verification, pronoun selection, profile picture selection, and selfie verification. There were also other pages included like the discovery page, edit profile page, following page, and community group pages. During this process, we encountered difficulties in getting the project code to run, which required several days of troubleshooting with InPower's technical team.

This was a problem as we were already off to a late start due to misaligned breaks and holidays between our student team and the client's main technical team as well as the change in scope of our work. The main stakeholder (our client) was notably a little anxious about our late start and made efforts in bringing us up to speed as quickly as possible so that we could start working on the pages. The other stakeholder (the client's technical team) though already had their share of things to work on graciously helped us through the troubleshooting and setting up of the relevant apps so that we could start working on the pages our client had requested.

2 REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS SPECIFICATION

2.1.1 Android/iOS Project Functional Requirements

Req. ID	Description	Priority
1	Design a seamless sign-up interface where users can input their phone number, email, birthday, and password.	High
2	Create a visually engaging verification process that sends a text to the user's phone number for validation.	Medium
3	Develop an intuitive login page allowing users to access their account using their phone number and password, with a user-friendly password reset option.	High
4	Design an aesthetically pleasing profile page where users can view all their posts and saved content.	Medium
5	Craft a visually appealing "My Anonymous Posts" section for users to access their anonymous contributions.	Medium
6	Design an interactive post display feature allowing users to click on a post for detailed viewing.	Medium
7	Create a visually appealing feed page as the default landing page upon login.	High
8	Develop a visually appealing prompt for push notification permissions.	Medium
9	Design user input fields for first and last name, ensuring an intuitive and accessible experience.	High
10	Implement a user-friendly interface for users to select their preferred pronouns.	High

11	Design an intuitive interface for users to upload a profile photo that aligns with the upcoming self-verification process.	High
12	Create a visually engaging "Selfie Time" feature where users can take and retake photos for profile verification.	High
13	Design an intuitive page for blocked users and include an unblock button.	Low

2.2 NON FUNCTIONAL REQUIREMENTS SPECIFICATION

2.2.1 Android/iOS Project Non-Functional Requirements

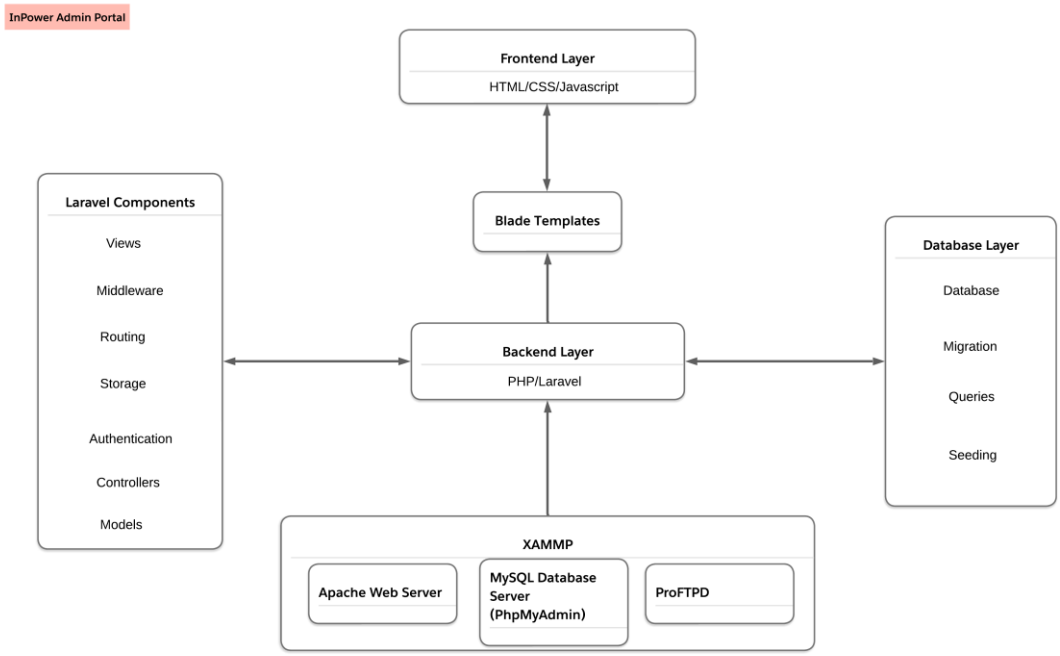
Req. ID	Description	Priority
1	Ensure that the app displays the correct font according to the client's requests.	High
2	Ensure that the pages are accessible by both Android and IOS users.	High
3	Ensure that the labels for the items used in the pages are clear and intuitive for the back-end designers to use.	Low
4	Ensure appropriate use of placeholder texts and pictures.	Low
5	Ensure code used does not clutter.	Medium
6	Prompt for answering member questions to join certain groups with restricted access.	Medium

2.2.2 Laravel Project - Non-Functional Requirements

Req. ID	Description	Priority
1	Student team should set deadlines and allocate dedicated time slots for language and system learning activities to ensure regular progress made.	Medium
2	Student team should maintain organized and comprehensive documentation of learning progress including notes, code snippets, tutorials and references for future review and reference.	Low
3	Student team should consistently engage in peer collaboration and knowledge sharing both within the team and with the client.	High
4	Student team should ensure that all code and project files are properly organized, documented and in the right version.	High

3 ARCHITECTURE

3.1 LARAVEL PROJECT ARCHITECTURE:



3.1.1 Development Environment

- Visual Studio Code (VSCode)
- XAMPP – Includes Apache, MySQL, PHP, and PHPMyAdmin

3.1.2 Frontend layer

- HTML, CSS, and JavaScript language comprises the frontend layer of the project. The HTML code is written inside PHP files to handle the website structure, CSS handles styling, and JavaScript handles the client-side functionality. Framework like Bootstrap was used to created seamless styling by providing CSS class reusability.
- Blade templating engine was used to present data on the web through Views, embedding PHP code within HTML.

3.1.3 Backend layer

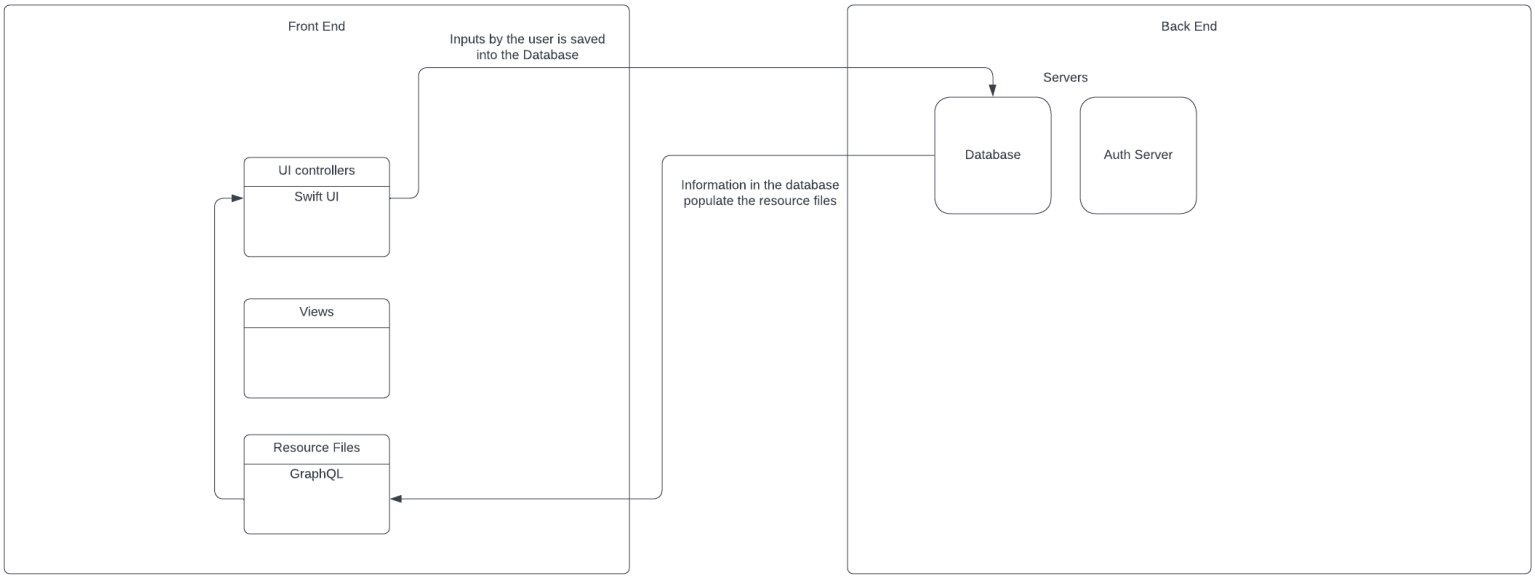
- PHP – The main programming language used for Laravel. Laravel is a PHP framework providing controllers, database, models, views, authentication, and routing features.

3.1.4 Database layer

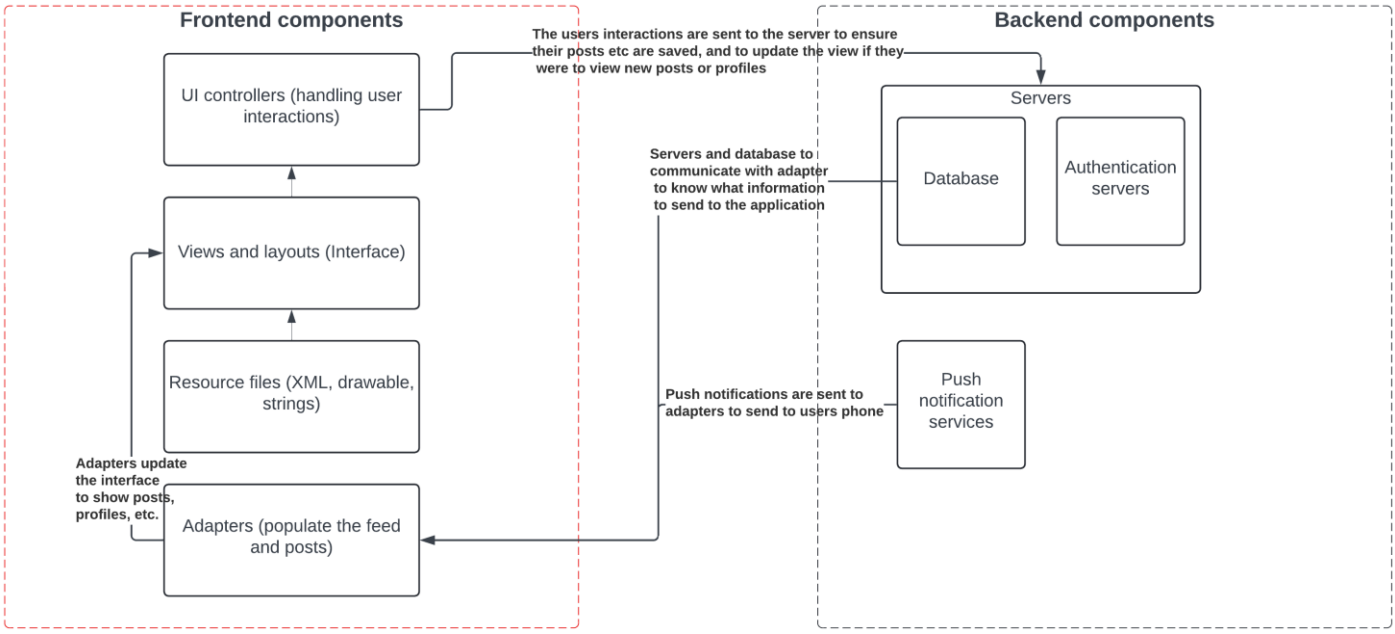
- A database administration tool like PHPMyAdmin was used to store data on the web, like users, badges. Groups, compliance, selfies, and category.
- Database Migrations and Seeds – Populating MySQL database with test data.

3.2 ANDROID/IOS PROJECT ARCHITECTURE:

3.3.1 iOS application architecture:



3.3.2 Android application architecture:



Highlighted in red, the frontend components are the components that we were working on. The backend components were not provided in the project but are typically what would be used on an android application such as the InPower social media application.

4 TECHNICAL FRAMEWORK

4.1 LARAVEL PROJECT TECHNICAL FRAMEWORK:

In stage 1 for the admin portal, we used a Laravel environment for the programming language. This was locally hosted on our computers using Apache and MySQL from XAMPP. The Laravel admin portal application was already implemented, albeit barebones, so we had to create some dummy data using Laravel's model factories and seeders.

4.2 ANDROID/IOS PROJECT TECHNICAL FRAMEWORK:

4.2.1 IOS application framework:

For the iOS project, we will be using swift and SwiftUI using Xcode, in addition we will be using Firebase as the backend of the app. We were given pre-existing swift code of the app to give a face lift, the team must create new pages and adding in features.

4.2.2 Android application framework:

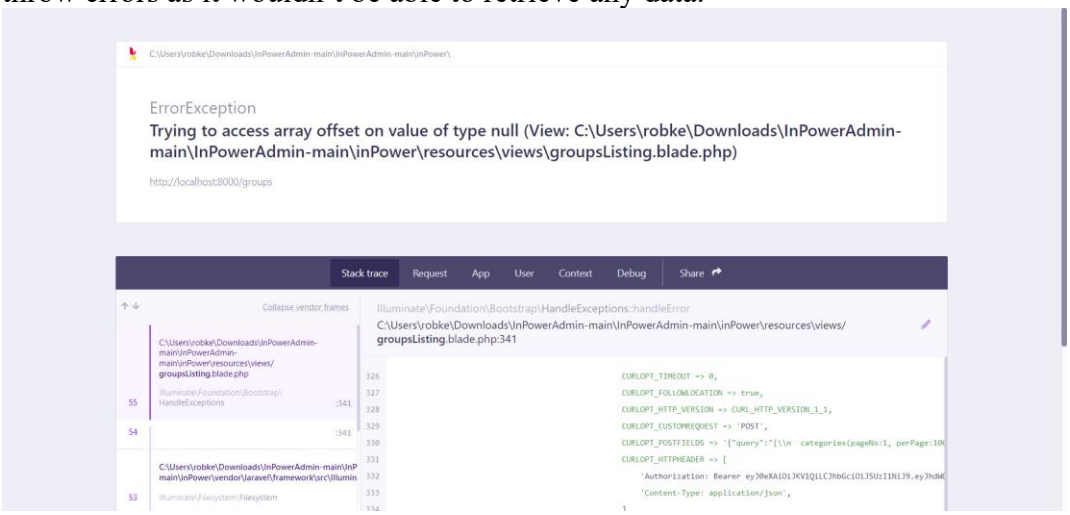
For the Android project, we will be using Kotlin for the base application, and Firebase as the backend of the application. Like the iOS project, we were given pre-existing XML files of the front end to give a face lift, including creating new pages and adding in features.

5 IMPLEMENTATION

5.1 Laravel Project Implementation

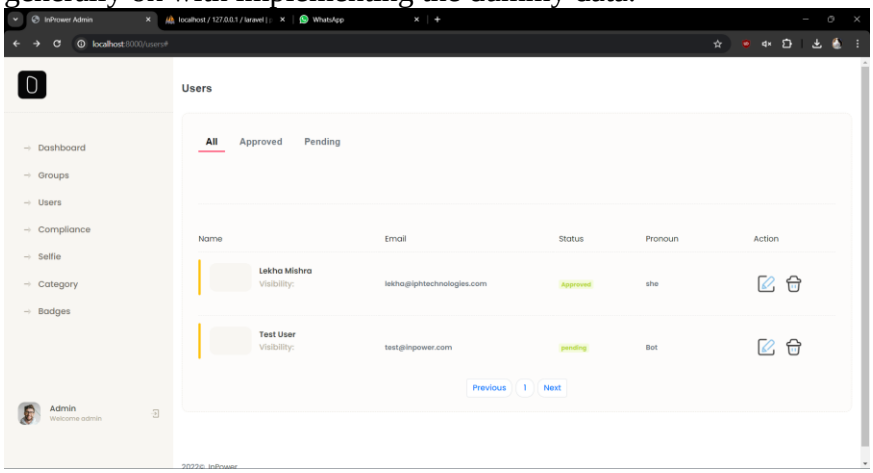
For the implementation of the Laravel application Admin portal website, we were tasked with using previous code and implementing some dummy data to get the application running. From here we were supposed to implement some new features and add data to ensure that the new features worked as well.

Below is a screenshot of the application before we implemented the dummy data. It would throw errors as it wouldn't be able to retrieve any data.



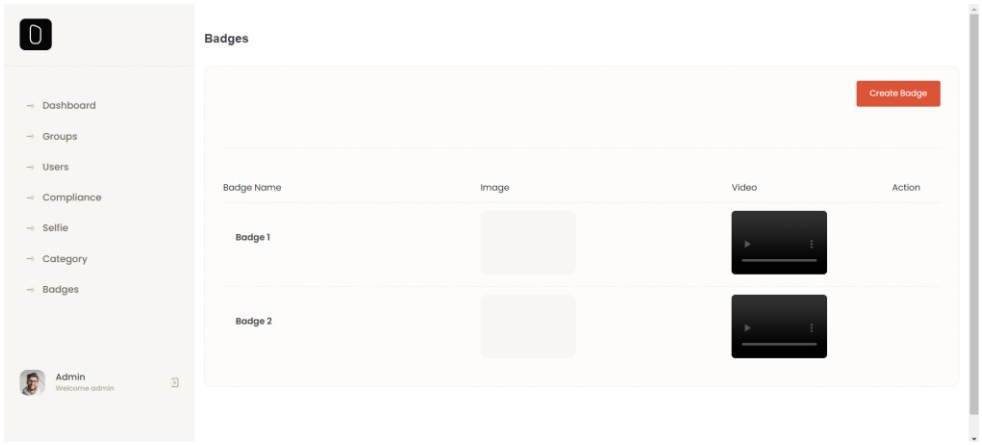
5.1.1 Implementing dummy data for the user's table

In the admin portal, there were several different tabs relating to different tables within the database. For the user's table, we had to implement some dummy users as shown in the screenshot below. This included their name, email, and pronouns. To implement dummy data in Laravel, they use a process called seeding, where you define seeders within a database to generate data. Understanding this process took some time but once we got it down it was generally ok with implementing the dummy data.



5.1.2 Implementing dummy data for the badges table

Like the user's table, we had to add some data for the badges table. This one was a bit more complicated as we had to add a video, but the concepts were similar. Although for this table, it had a button to create a badge if the admin was wanting to add another badge, so that they don't have to manually add it in a database.



After we had the dummy data implemented the project shifted from the Laravel project to the mobile application project as the scope of the Laravel project seemed too large and out of our comfort zone to continue.

5.2 Android/iOS Project Implementation

For the implementation of the mobile application, our group was split up into two different groups consisting of the iOS application and the android application. From here, we were tasked with creating and updating front end pages of an existing mobile application with regards to a new design. Which was provided to our group via a Figma document from the client. The assigned Figma screens were divided between the InPower-Admin group, and another group also working with the client.

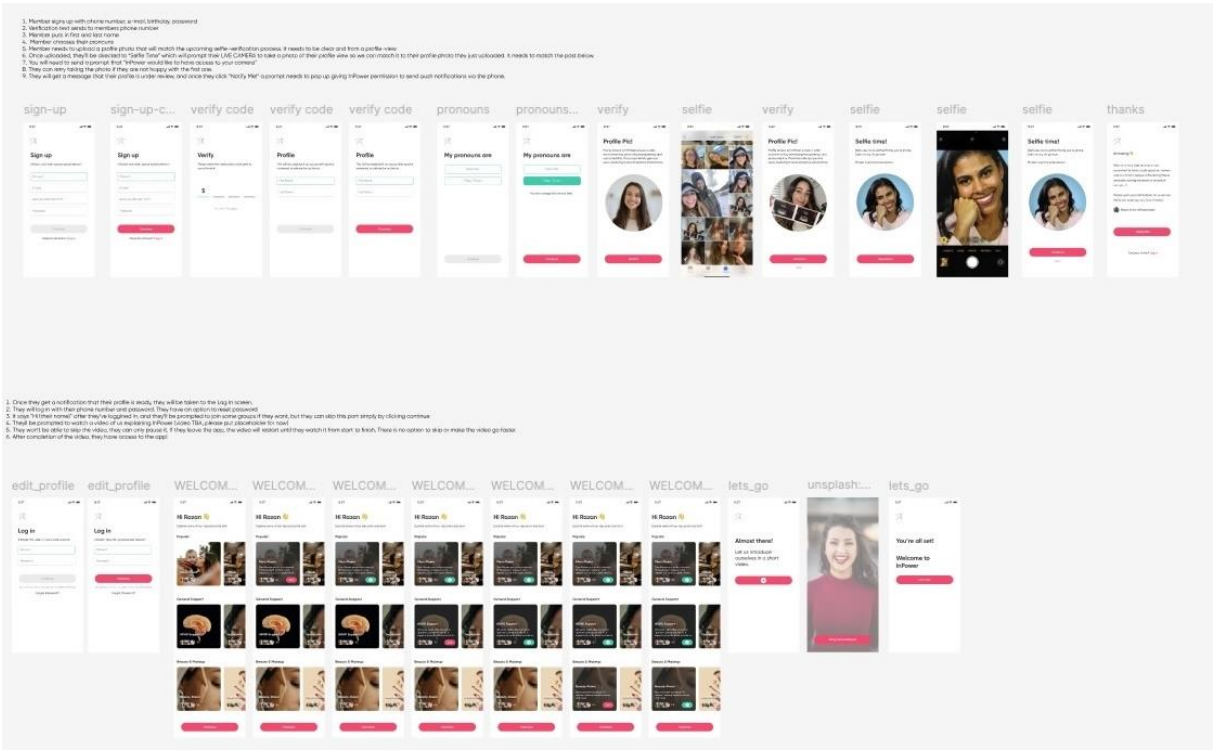
5.2.1 Implementation of the iOS project:

An existing repository was provided to InPower-Admin by the client. This repository had been created by a previous client of InPower with various technologies with the earliest files dating to 2018. Upon inspection of the repository, it was realised that many of the technologies were outdated and needed to be updated to run on recent version of Xcode and on iOS 17. The interface for the previous app had been constructed using Storyboards and UIKit within Xcode, this has mostly been depreciated by Apple and into the future will eventually be entirely replaced by SwiftUI. This fact was discussed with the client, and it was decided to implement the updated interface in SwiftUI and completely replace the project's existing interface.

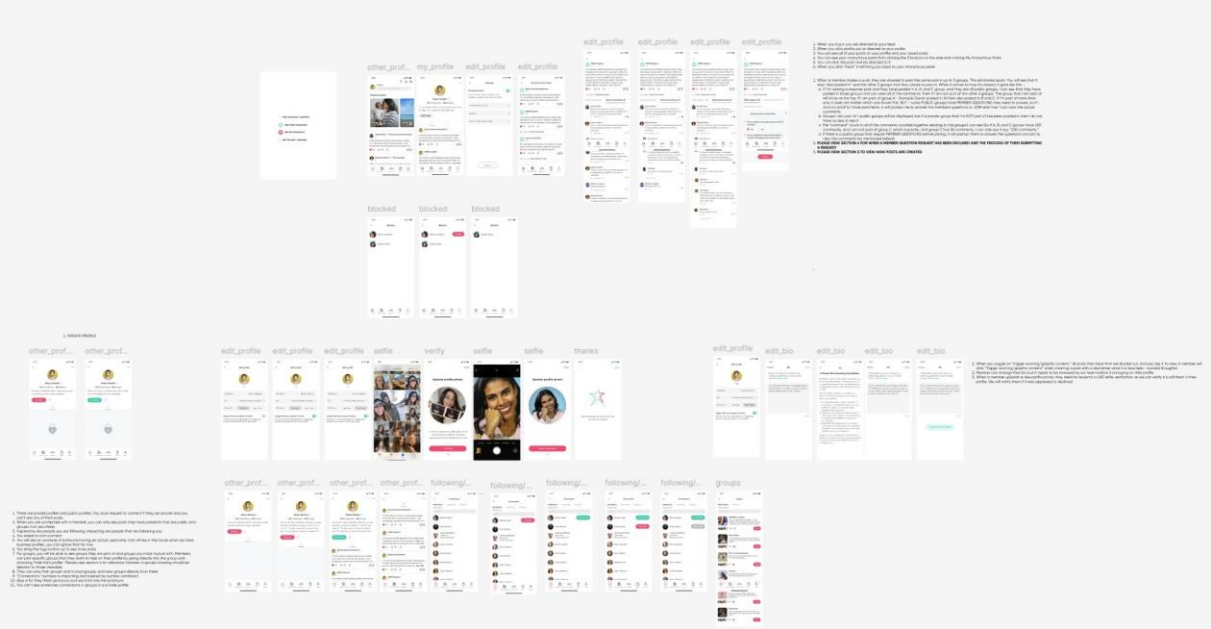
Screenshots of the provided Figma designs provided by the client are included below. 'Sections' were allocated to individual members of the InPower-Admin team to complete, time estimates were not made at this early stage.

InPower-Admin was tasked with constructing SwiftUI files or pages to match the below Figma design provided by the client InPower. The pages we not to be connected to the Firebase back end during this project. Colours, formatting, and functionality of buttons in order to move to the next page were within the scope of this project. The newly created pages were to be integrated into the existing project.

Section 1: Login and Welcome Pages



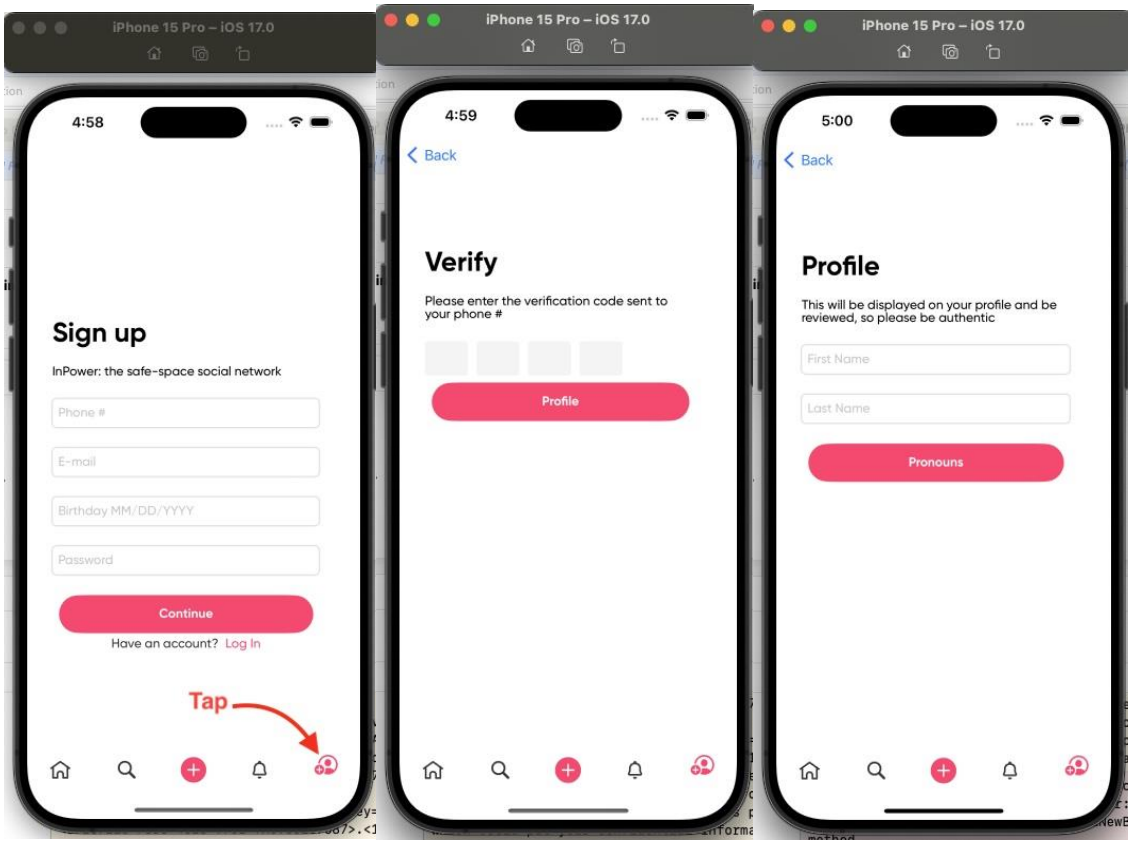
Section 2: Profile, Edit Profile, Edit Bio, Other Profile, /Following Pages

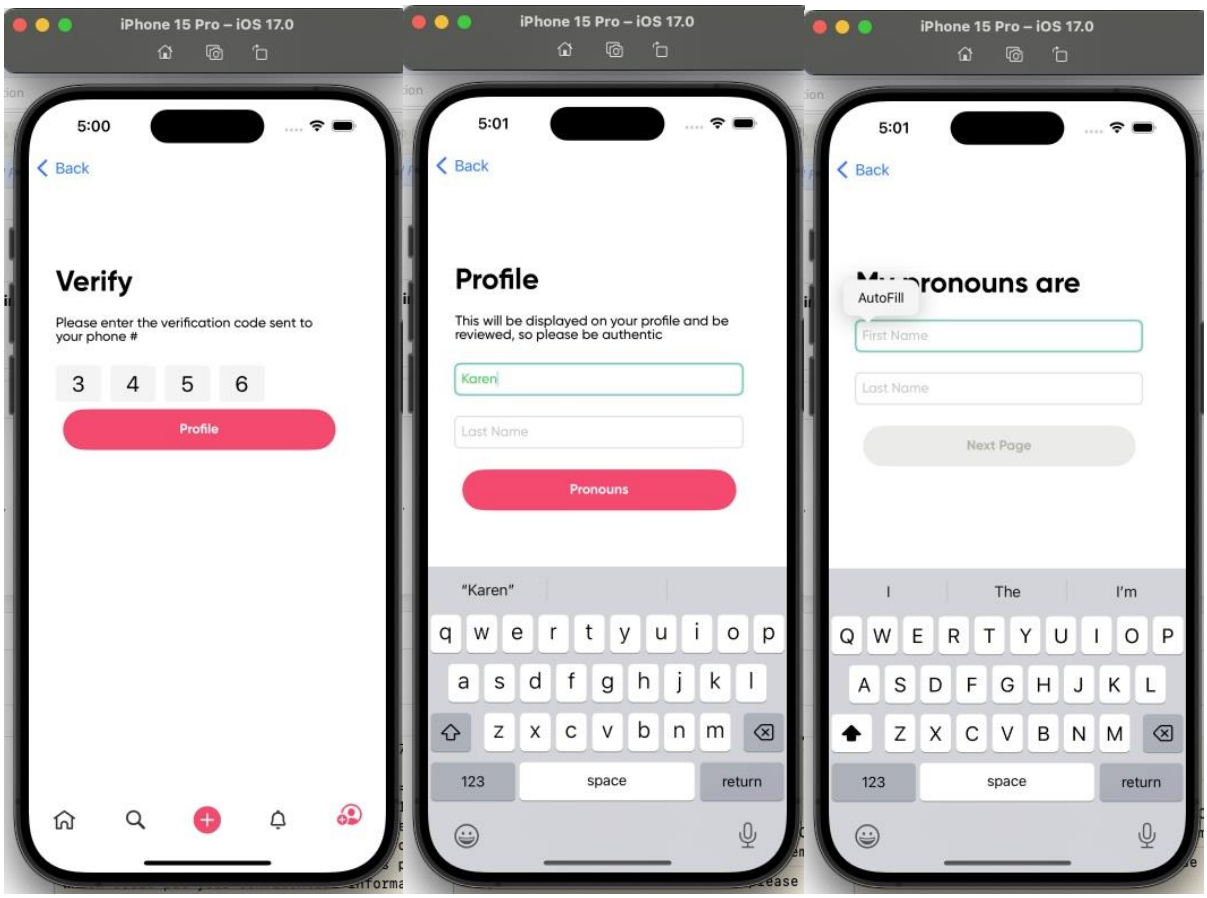


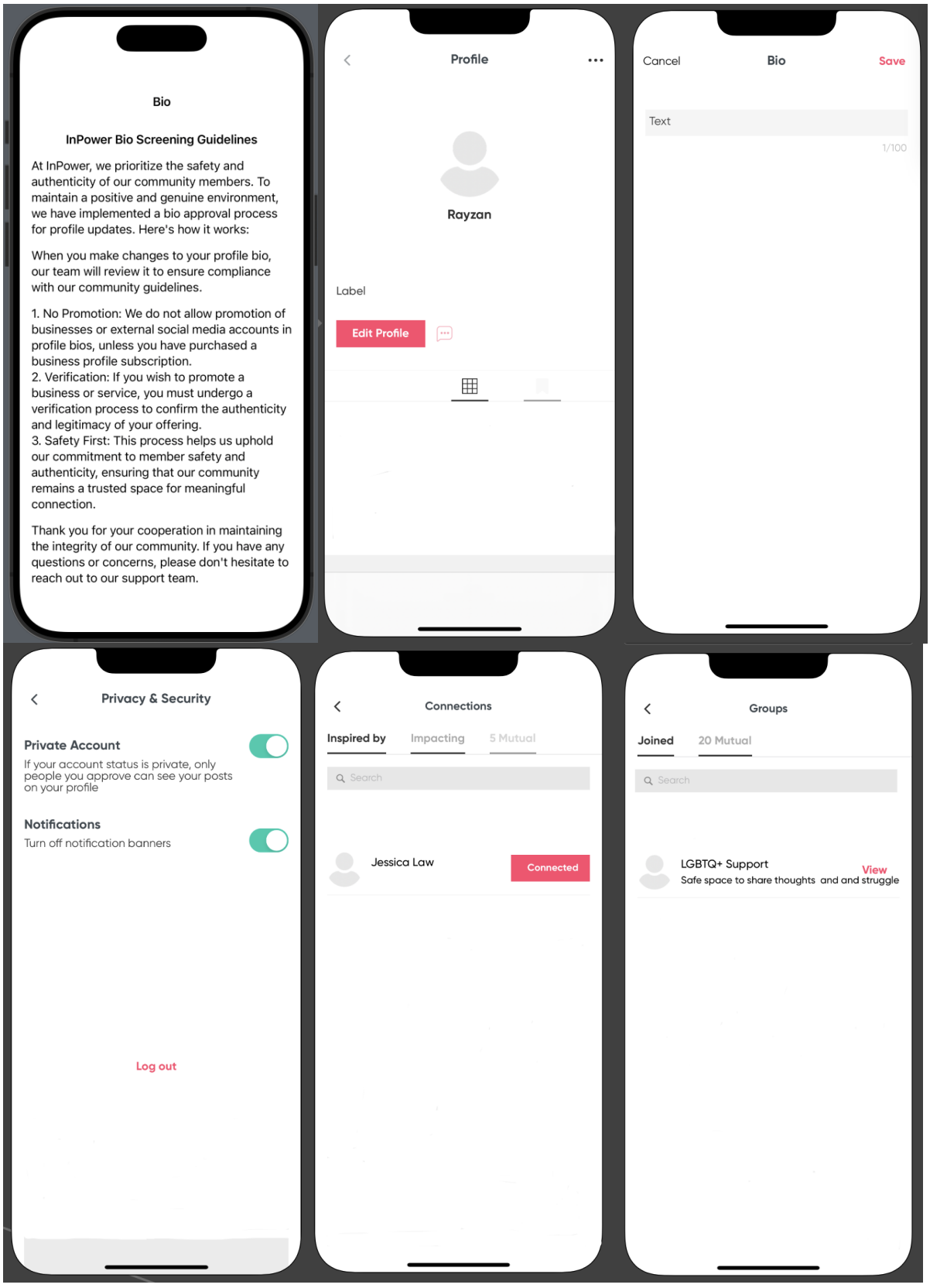
InPower-Admin did not have experience using Storyboards or UIKit, or the integration of SwiftUI and UIKit Storyboards. It was decided to implement the new interface in an entirely blank project, to provide the client with a visual implementation of work which was being done. Research into integrating the new interface with the old codebase was conducted alongside this work.

Completed Work

Login and Welcome Pages




iPhone 15 Pro – iOS 17.0
5:00
Back
Verify
Please enter the verification code sent to your phone #
3 4 5 6
Profile
iPhone 15 Pro – iOS 17.0
5:01
Back
Profile
This will be displayed on your profile and be reviewed, so please be authentic
Karen
Last Name
Pronouns
"Karen"
iPhone 15 Pro – iOS 17.0
5:01
Back
AutoFill
First Name
Last Name
Next Page
I
The
I'm

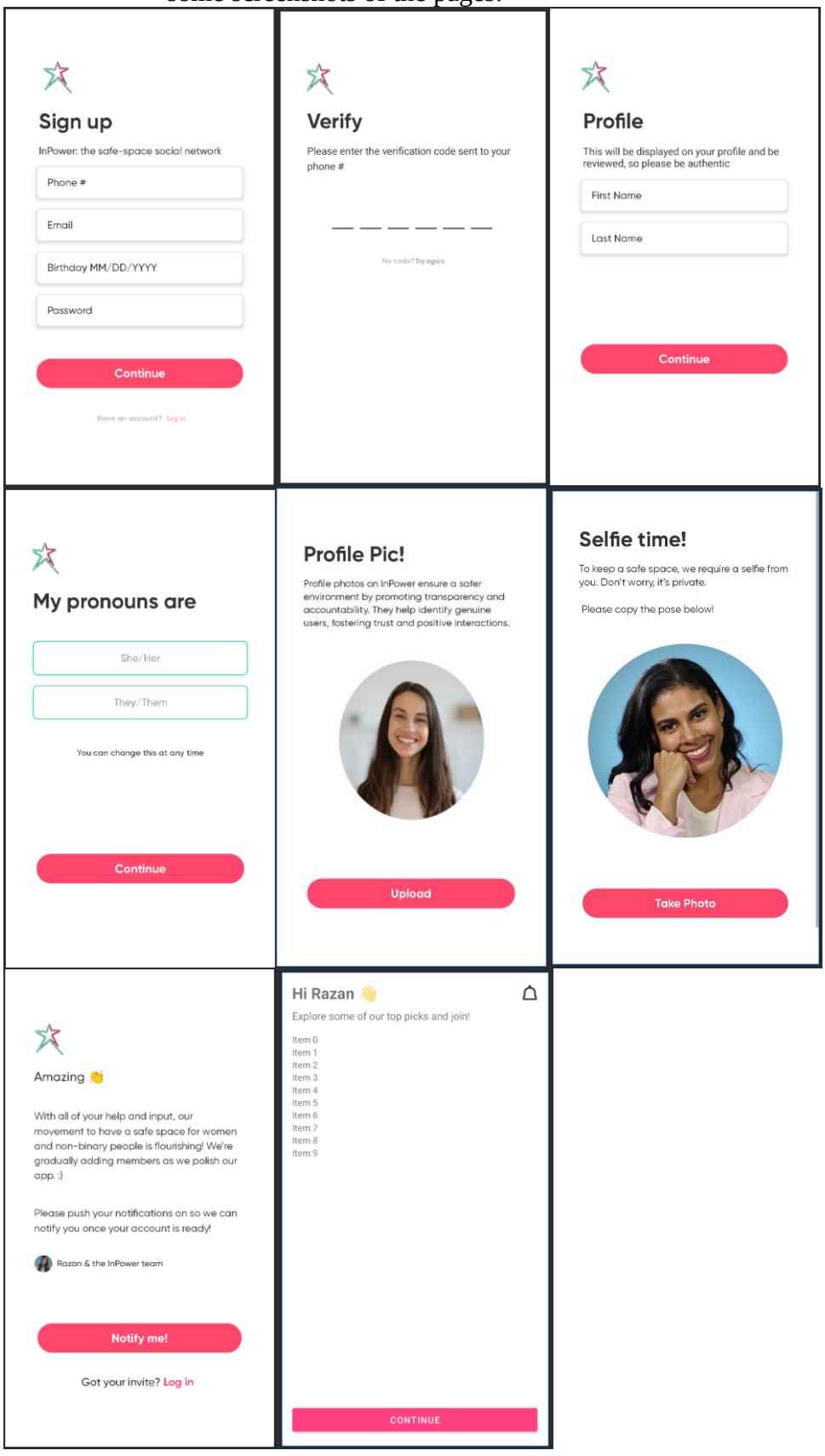

Bio
InPower Bio Screening Guidelines
At InPower, we prioritize the safety and authenticity of our community members. To maintain a positive and genuine environment, we have implemented a bio approval process for profile updates. Here's how it works:
When you make changes to your profile bio, our team will review it to ensure compliance with our community guidelines.
1. No Promotion: We do not allow promotion of businesses or external social media accounts in profile bios, unless you have purchased a business profile subscription.
2. Verification: If you wish to promote a business or service, you must undergo a verification process to confirm the authenticity and legitimacy of your offering.
3. Safety First: This process helps us uphold our commitment to member safety and authenticity, ensuring that our community remains a trusted space for meaningful connection.
Thank you for your cooperation in maintaining the integrity of our community. If you have any questions or concerns, please don't hesitate to reach out to our support team.
Profile
Rayzan
Label
Edit Profile
Cancel
Bio
Save
Text
1/100
Privacy & Security
Private Account
If your account status is private, only people you approve can see your posts on your profile
Notifications
Turn off notification banners
Log out
Connections
Inspired by
Impacting
5 Mutual
Search
Jessica Law
Connected
Groups
Joined
20 Mutual
Search
LGBTQ+ Support
Safe space to share thoughts and and struggle
View

5.2.2 Implementation of the Android project:

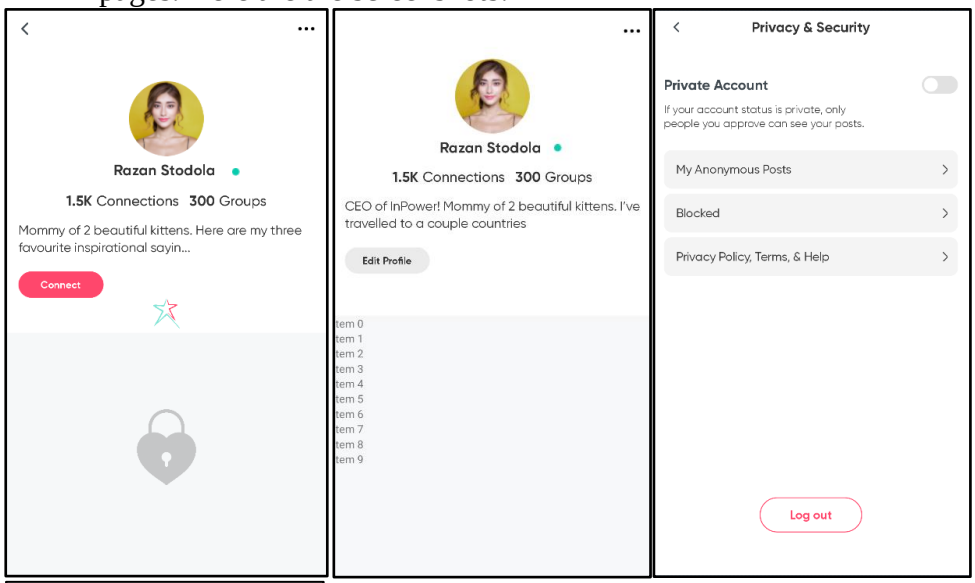
For the Android application, we were tasked with handling three different sections:

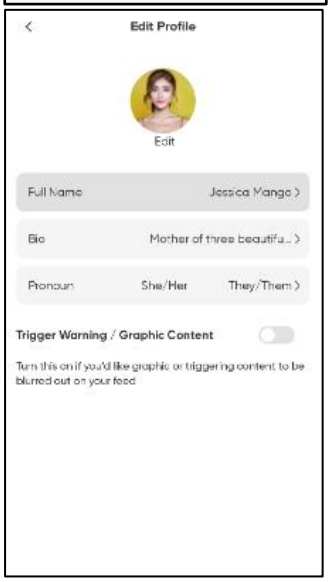
1. The sign-up verification and logins, with homepage

    Firstly, for the sign-up verification, we had to make the pages for the sign-up and verification, and then implement the verification into the sign-up pages. Alongside this, we had to implement the selfie and following pages. Here are some screenshots of the pages:



2. The profile related pages, including other profiles, and profile settings related pages.
   For the profile related pages, we had to update the existing profile pages to match the new looks, the other profiles, a locked profile and all the settings pages. Here are the screenshots:





3. The group related pages, with the implementation of quizzes
   For the group related pages, we had to implement the look of the new groups, joining groups, how the quizzes look and function, and the support for the groups. Youthtakak was unable to provide screenshots of his work which is why we've taken some points off in the contribution form.

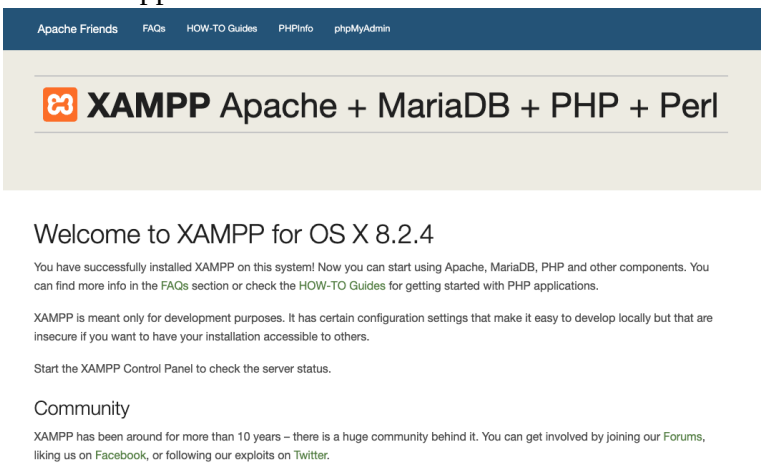
6 DEPLOYMENT INSTRUCTIONS

6.1 Laravel Project Deployment Instructions

Deploying the Laravel Project requires downloading XAMPP, and open source web server solution. In Xampp, Apache web server is required to run the admin portal on the web via localhost.
Download Apache from (https://www.apachefriends.org/download.html) and follow installing instructions and accept default settings.
After installing, open XAMPP.

Microsoft Windows:
In the XAMPP Control Pannel, click on "Start" button next to Apache and MySQL to start the servers
Check that XAMPP has been installed correctly by going to http://localhost

MacOS
When installing XAMPP, accept default configuration.
If prompted by GateKeeper, click "Open" to allow the installation to proceed.
After installing, open XAMPP
In the XAMPP window, go to Manage Servers tab, click on Start all at the bottom.
Check that XAMPP has been installed correctly by going to http://localhost
The below window should appear.



Installing dependency manager
The dependency manager Composer is required. Open command prompt or terminal locally

Microsoft Windows:
```
curl -sS https://getcomposer.org/installer | php
move composer.phar
C:\Users\YourUsername\AppData\Roaming\Composer\composer.phar
set PATH=%PATH%;C:\Users\YourUsername\AppData\Roaming\Composer
```
Replace `YourUsername` with your actual username.

MacOS:
```
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer
```

Configuring database settings for the Laravel Project
In the chosen IDE (preferably VS Code), update to the latest version.

Open a new terminal.
In the terminal, locate the directory of the main project.
Run the following command to create an env file,
`cp env.example .env`

- "env" file is required to configure database setting. The file is used to define environment specific configuration variables for database connection details to match MySQL database provided in XAMPP.

Open the .env file and configure the database settings according to your local environment.

Run the Laravel Project

In the code editor terminal and in the project directory, install the required dependencies.
`composer install`
Update dependency if error message shows "Your lock file does not contain a compatible set of packages. Please run composer update"
`composer update`

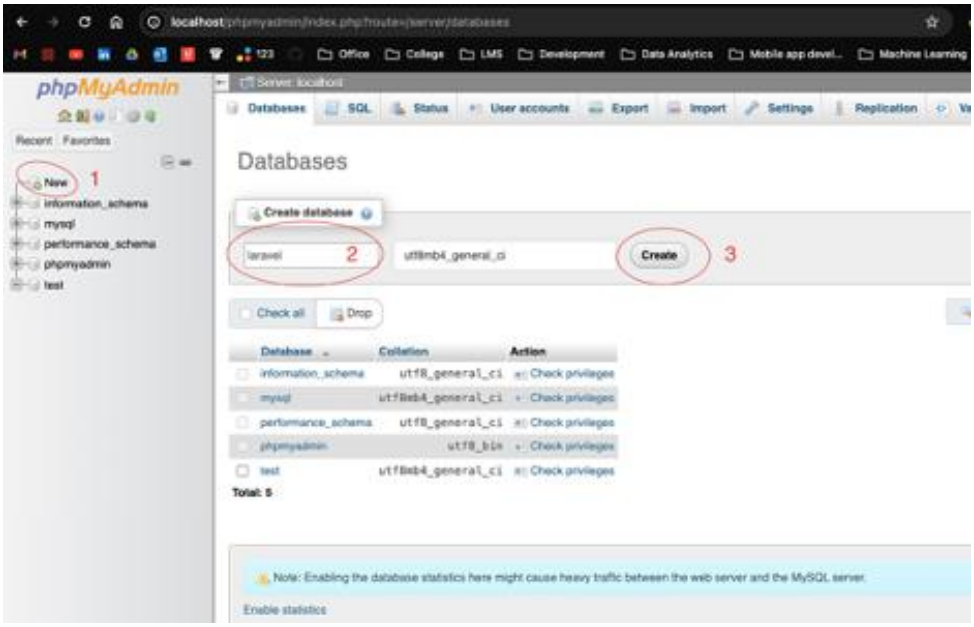
- Artisan CLI – Laravel's command like interface that provides commands for running Laravel-based code, database migrations, and executing unit tests.

In the same terminal, run the following command to generate an application key.
`php artisan key:generate`
Start Laravel development:
`php artisan serve`
Open your web browser and navigate to http://localhost:8000 to view the Laravel project.

Connecting database to Laravel Project

- Database server is MySQL. The database is accessed through PhpMyAdmin online database.

Assuming the Apache and MySQL server is running:
Go to http://localhost
Click on PHPMyAdmin in top right corner.
Create a new database named "Laravel".



After creating the database, go back to the code editor and go into a new terminal.
Enter the main project directory.
In the command line, type:
`php artisan migrate`
Then:
`php artisan db:seed`
Then:

```
php artisan serve
```

6.2 Android Application Project Deployment Instructions

Prerequisites:

1. Insalled Android Studio
2. A free Github personal account.
3. InPower must be contacted and must give your personal account access to the: "https://github.com/rtalebian/April29_New_Android/" repository.
4. Android 12.0 or later

Deployment instructions:

To deploy the Android Application, we first need to download the project from the GitHub. The project is structured in a way to be used with Android Studio. To be able to run the project, either Java 8 or Java 17 will be needed.

- If using Java 17, you may need to upgrade your gradle to version 7.0.2 (Tool >> AVG Upgrade Assistant >> Version 7.0.2 >> "Run selected steps")

After this, import the project into Android Studio and sync gradle in Android Studio to ensure everything included is up to date.

After this step, you will need to incorporate a virtual device to run the project on. To add a virtual device and run the project, follow these steps:

- Go to SDK Manager under the settings menu in the top right corner.
- Under Android SDK, in SDK platforms, select Android 11.0 ("R"), Android 12.0 ("S"), and Android 14 ("UpsideDownCake").
- Under Android SDK, in SDK tools, select Android SDK Build-Tools 35-rc3, Android Emulator, and Android SDK Platform-Tools.
- In the device manager in the right side of Android studio, click "Create Virtual Device" to add a device using these specifications, where the project can then be run.
- From here, the project should be able to be ran using the green arrow at the top of Android Studio.

6.3 iOS Application Project Deployment Instructions

Prerequisites:

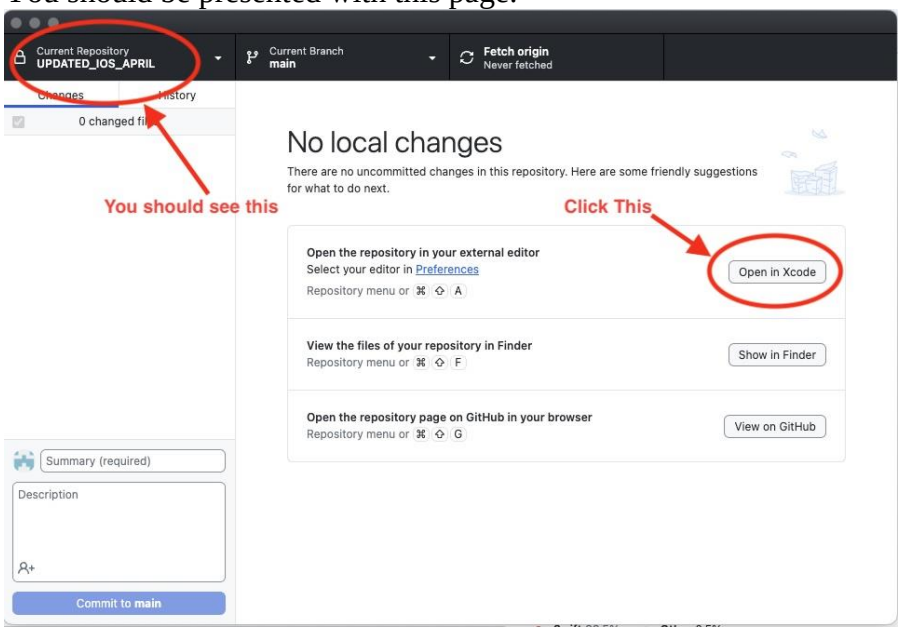
5. A Mac Computer running macOS Ventura 13.5 or later
6. Insalled Xcode 15 – at the time of writing, Xcode 15 is the most recent and only version of Xcode available on the Mac App store.
7. A free Github personal account.
8. Install most recent version of github desktop to mac computer
9. Xcode must be connected to your Github personal account with a personal access token. Instruction on how to do this may be found within Xcode>Preferences>Accounts> '+' > Github Account. Then follow instuctions within this wizard.
10. InPower must be contacted and must give your personal account access to the: "https://github.com/rtalebian/UPDATED_IOS_APRIL.git" repository.

Once these tasks have been completed you may continue to following steps:

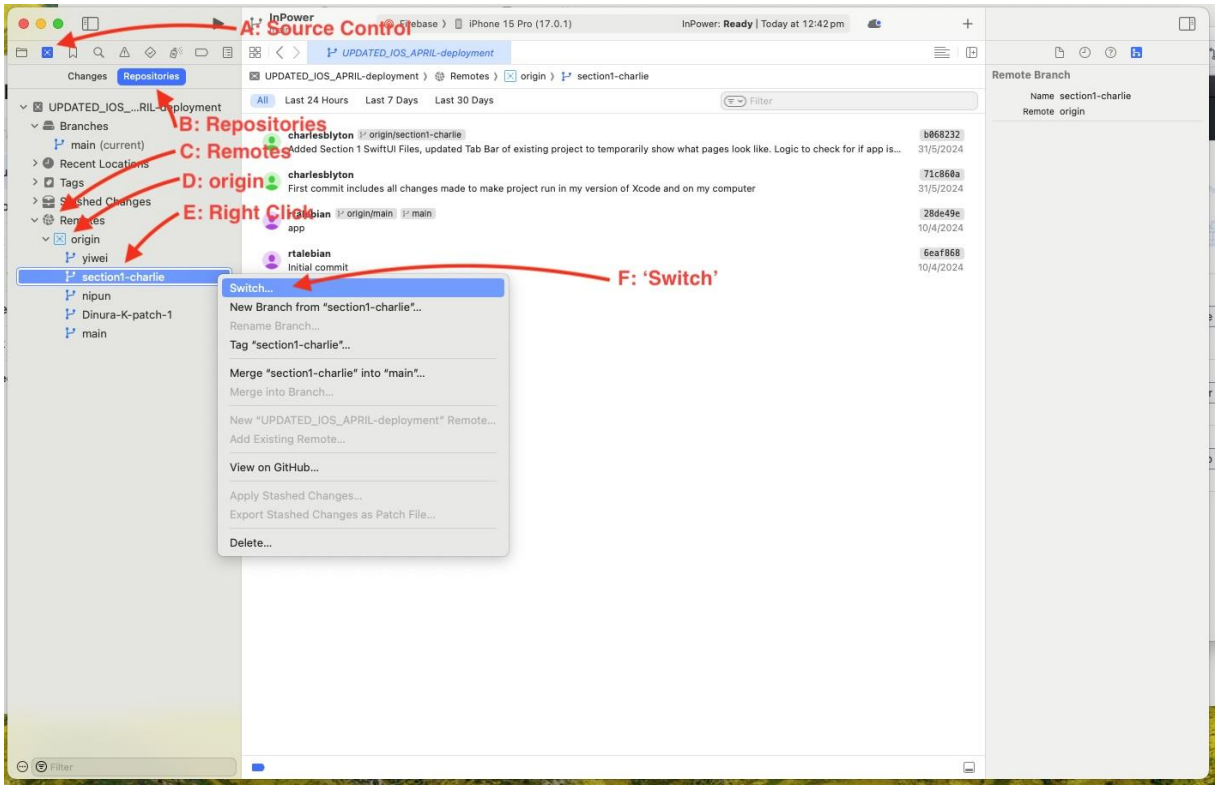
Github Desktop

1. With the github desktop application open, select File > Clone repository
2. From the list presented to you, under the 'rtalebian' heading, you should find a repository named 'rtalebian/UPDATED_IOS_APRIL'. Select this repository, then enter a local path where you would like to save the repository on your own machine. Give a different local name if desired.
3. Click 'Clone'.

4. You should be presented with this page:



5. Upon clicking ‘Open in Xcode’ the Xcode window will appear. Wait a few minutes until Xcode has processed all the files.
6. Follow the following procedure to switch to the required branch:
    a. Switch to the source control sidebar
    b. Click on repositories
    c. Open the remotes dropdown
    d. Open the origin dropdown
    e. Right click on the ‘section1-charlie’ option
    f. Select ‘switch’
    g. Press ok on the popup that appears



7. To View Charles’s work switch to section1-charlie' branch.
8. To view Dinura’s work switch to ‘Dinura-K-patch-1' branch.

7 TEST SPECIFICATIONS

Test Case ID	Req. covered	Test Objective	Preconditions	Steps	Test data	Expected result
FV1	Field validation	Ensure password meets complexity requirements (length)	1. The sign-up form is accessible. 2. The password field is visible and editable.	1. Navigate to the sign-up page. 2. Locate the password field. 3. Enter a password that does not meet complexity requirements (e.g., too short). 4. Submit the form. 5. Observe the error message. 6. Enter a password that meets the complexity requirements. 7. Submit the form. 8. Verify successful submission.	"12345" (too short) "abcdef" (too short)	Error message and button does not light up red
U1	Usability	Ensure confirmation buttons are clickable after different conditions on each page	1. The sign-up and login pages are fully loaded and accessible. 2. Confirmation buttons are present on the pages.	1. Navigate to the sign-up or login page. 2. Locate the confirmation button (e.g., "Sign Up," "Log In"). 3. Click the button without entering any data. 4. Observe the error messages. 5. Enter valid data in the form fields. 6. Click the confirmation button again. 7. Verify that the button responds appropriately and the user is taken to the next step.	Leave all fields empty and click "Sign Up" or "Log In".	Error message and button does not light up red

RD	Responsive Design	Verify the page layout adapts to the screen of the phone model our client wants to test it on	1. The page is loaded on the specific phone model for testing. 2. The device is connected to the internet and has a compatible browser.	1. Open the page on the specified phone model. 2. Check the layout for responsiveness (elements adjust correctly to screen size). 3. Interact with form fields and buttons. 4. Rotate the phone to landscape mode and verify the layout adjustment. 5. Verify that all elements are accessible and functional in different orientations.	Test on device specified by client (Google Pixel 8)	Pages load as with no distortion
CB1	Consistent Branding	Verify that colors, fonts, and logos align with InPower's brand guidelines.	1. The pages are fully designed and accessible. 2. Branding guidelines are available for comparison.	1. Navigate to the sign-up and login pages. 2. Compare the colors, fonts, and logos with the branding guidelines. 3. Verify that the branding elements are consistent across all pages. 4. Check for any deviations and document them.	Logo: Ensure correct and consistent use of the InPower logo.	Logo is present and visible in every page
FC1	Form Clarity	Ensure form fields are labeled clearly and concisely. Verify placeholder text provides additional guidance.	1. The sign-up and login forms are fully designed and accessible. 2. Placeholder text is implemented in the form fields.	1. Navigate to the sign-up and login pages. 2. Check that all form fields are clearly labeled. 3. Verify that placeholder text provides additional guidance where necessary. 4. Test the form by entering data to ensure labels and placeholders are helpful. 5. Observe the user experience and clarity of the form fields.	Full name Email Address Password	Form fields correctly display these placeholder texts

8 TESTING RESULTS

All test cases for the sign-up and login pages of the InPower application passed successfully. The password field validation ensured passwords met complexity requirements. Usability tests confirmed that confirmation buttons were clickable and functional under various conditions. Responsive design tests validated that the layout adapted well to specified device. Consistent branding was maintained across all pages, and form clarity was ensured with clear labels and helpful placeholder text. Overall, the pages are functional, user-friendly, and secure.

9 CYBER SECURITY

Due to the sensitive nature of the user data handled by InPower, comprehensive cybersecurity measures must be included during development. The Laravel administration portal, which handles backend management, must incorporate stringent security controls to protect administrative access and manage sensitive user data efficiently. This includes securing the portal with multi-factor authentication, role-based access controls, and regular monitoring for suspicious activities to ensure the integrity and confidentiality of the user data.

InPower users can make postings public or private inside their groups, allowing other users to engage and provide comments. As a result, securing personal information such as names, phone numbers, and photographs becomes critical. InPower prioritizes cybersecurity by using strong encryption techniques to protect data in transit and at rest, guaranteeing secure authentication processes to prevent unauthorized access, and constantly upgrading the software to address any vulnerabilities. Furthermore, adhering to privacy standards and completing regular security audits are critical for maintaining user confidence and protecting the platform from any cyber-attacks. As our team is in charge of creating the front end of the application, we have only implemented the mobile phone verification for the account creation.

10 IMPACT ON STAKEHOLDERS

In helping create a mobile application for InPower we can significantly enhance accessibility by providing users with easier access to resources and real-time updates through their mobile devices. This can lead to increased engagement and better user experiences due to a streamlined, user-friendly interface tailored to the community’s needs. Additionally, the app can offer improved privacy and security, giving users more control over their data compared to existing social media platforms. Over 84% of people aged over 18 are said to use social media (datareportal, 2018) – So by expanding its reach, InPower can attract a more diverse user base, fostering a stronger sense of social inclusion and support for women, non-binary, and genderqueer individuals. The app will also raise awareness about online harassment and healthy social interactions, contributing to a more empathetic and connected society.

11 OTHER CONSIDERATIONS

The client desires the InPower social media app to be available on both iOS and Android devices to maximize accessibility and reach a wider audience. Which was why our student team had two people working on the IOS pages and three people working on the android pages. Developing for both platforms ensures inclusivity, allowing users to access the app regardless of their device preference.
Since we were a temporary student team, we were also instructed by the tech team leads to have concise and intuitive IDs for the objects and classes that we were coding. This practice was crucial in helping them maintain code readability, ease of maintenance, and efficient collaboration among the many developers that the company will probably be working with. Clear and meaningful IDs help in quickly identifying the purpose and function of each element, reducing the likelihood of errors, and simplifying debugging processes. Adhering to these guidelines ensures that the code remains organized and understandable for both current and future development teams.

12 REFERENCES

The PHP framework for web artisans (no date) Laravel. Available at: https://laravel.com/docs/5.1/structure (Accessed: 12 April 2024).

Otwell, T. (no date) Laravel bootcamp - learn the PHP framework for web artisans, Laravel. Available at: https://bootcamp.laravel.com/introduction (Accessed: 16 April 2024).

Roos, P. (2023) Which web frontend architecture fits best, workingsoftware.dev. Available at: https://www.workingsoftware.dev/frontend-rendering-techniques/ (Accessed: 08 May 2024).

Global Social Media Statistics - DataReportal – global digital insights (no date) DataReportal. Available at: https://datareportal.com/social-media-users (Accessed: 24 May 2024).

A Guide to Functional Requirements (with Examples) (no date) A guide to functional requirements (with examples). Available at: https://www.nuclino.com/articles/functional-requirements (Accessed: 30 May 2024).