NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SCSE18-0315

# Person-centred care for dementia patients (Front end Android App)

## Final Year Project Report

Author: Chong Jun Cheng

Matriculation Number: U1620826L

Supervisor: Associate Professor Chan Syin

Examiner: Associate Professor Yu Han

School of Computer Science and Engineering

Academic Year 2018/1

# Abstract

As Singapore's population continues to age, the number of dementia patients is expected to rise [2]. For the guardian of dementia patients, taking good care of a dementia patient is a major concern. Alzheimer's Association in their 2018 Dementia Care Practice Recommendations named Person-centred care as the core of providing quality care to dementia patients [1].

PEAR system was developed to assist care centres and caregivers in providing Person-centred care to dementia patients. The objective of the project was to maintain and improve the existing Android application in PEAR system. The project focused on improving user experience (UX) of existing PEAR Android application. To provide a great user experience (UX), it is important to have a good User Interface (UI) design and good performance.

The project revamped the User Interface of the entire application. The author had implemented a total of 8 pages based on the new User Interface design. The performance of the application was significantly improved with the use of API and only fetching the necessary information for a page. Finally, the responsiveness of the application was also improved with the use of the Android AsyncTask class.

# Acknowledgements

The final year project would not be successful without the support of my supervisor, and my teammates.

I would like to extend my deepest gratitude to my supervisor, Associate Professor Chan Syin, for her constant guidance throughout the course of the project. Her suggestions and ideas on the project have constantly encouraged me to improve my implementations.

I would like to thank fellow final year project front-end team members, Hui Ming, Jia Hui, Poh Yen and Yai Wen for their cooperation and efforts in improving the application.

I would also like to thank the fellow final year project back-end team members, Hui Ying, Jun Yi, Mindy and Saifullah for working very hard to provide API to me and the front-end team.

Finally, I would like to thank fellow final year project seniors, Ashley and Zhen Hui, for aiding me in understanding the project and supported me in the project.

# Contents

# List of Figures

# List of Tables

# 1. Introduction

## 1.1 Background

According to a 2015 study by Institute of Mental Health (IMH), 1 in 10 people aged 60 and above in Singapore has dementia [2]. As Singapore's population continues to age, the number of dementia patients is expected to rise [3]. For the guardian of dementia patients, taking good care of a dementia patient is a major concern.

Alzheimer's Association in their 2018 Dementia Care Practice Recommendations named Person-centred care as the core of providing quality care to dementia patients [1]. Person-centred care, developed by Tom Kitwood at the University of Bradford, England, is an ideology of care focus on the needs of the individual where they become the centre of the caring process [4].

To provide Person-centred care to dementia patient, PEAR system was developed to assist care centre and caregiver. PEAR which started its development by prior final year project students since a few years ago, allows each patient in a care centre to have a personalised schedule. PEAR comprises 3 parts, web application, mobile application and games (based on Android platform). Although there are many care centre or caregiving systems out there, there was no system that is so complete and focused on providing Person-centred care.

## 1.2 Objective

The objective of the project is to maintain and improve the existing Android application in PEAR system. The PEAR Android application helps the caregiver to manage patients schedule and allow them to retrieve a patient's past and current information easily. In addition, the PEAR mobile application supports monitoring of the patient's game play result which helps doctors and game therapists in their therapy planning.

The project will focus on improving user experience (UX) of existing PEAR Android application. To provide a great user experience (UX), it is important to have a good User Interface (UI) design and good performance. The project will review and redesign existing PEAR UI to provide the user with a familiar UI with the least learning curve. As for the performance, the project will examine the current application and try to minimise any potential overhead.

## 1.3 System Overview

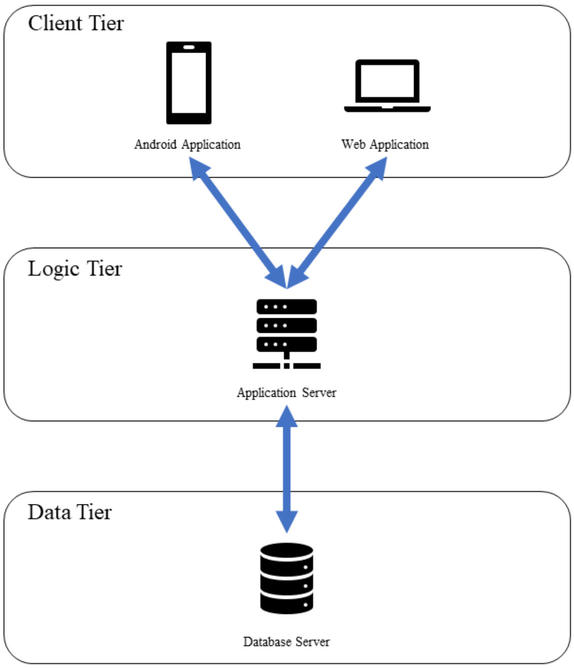


*Figure 1: PEAR System Architecture*

Figure 1 above shows the system architecture of the PEAR system. PEAR implements the 3-Tier Architecture which is a client-server architecture where each part and each layer were developed and maintained independently on different platforms. The 3 tiers are Client Tier, Logic Tier and Data Tier. PEAR Android games, an important part of PEAR was not included in the system architecture as they were developed and run independently without direct support of the application server.

## 1.3.1 Client Tier

The Client Tier served as the presentation layer or user interface to access the information in the PEAR system. The Client Tier comprises Android Application and Web Application.

**Android Application**

The Android application will run on Android tablets and phones. Care Centre Supervisor, Caregiver and Doctor will use the Android application to view and manage the patient's information. The supervisor and caregiver will also use the application to view the auto-generated schedule for the patient in the Care Centre. The Android application will have a subset of features as opposed to the web application.

**Web Application**

The web application will run in a web browser. Care Centre Supervisors, Doctor, Game Therapists, Patient's Guardians and System Administrator will use the web application. The web application will have the complete set of features.

## 1.3.2 Logic Tier

The Logic Tier served as the application layer to provide business logic to clients using an Application Server.

### 1.3.3 Data Tier

The Data Tier using a database server, served as the data layer, to provide access to PEAR application data.

## 1.4 Team Structure

Figure 2 below shows the project team organisation of PEAR which consists of 3 sub-teams, front-end, back-end and games team.

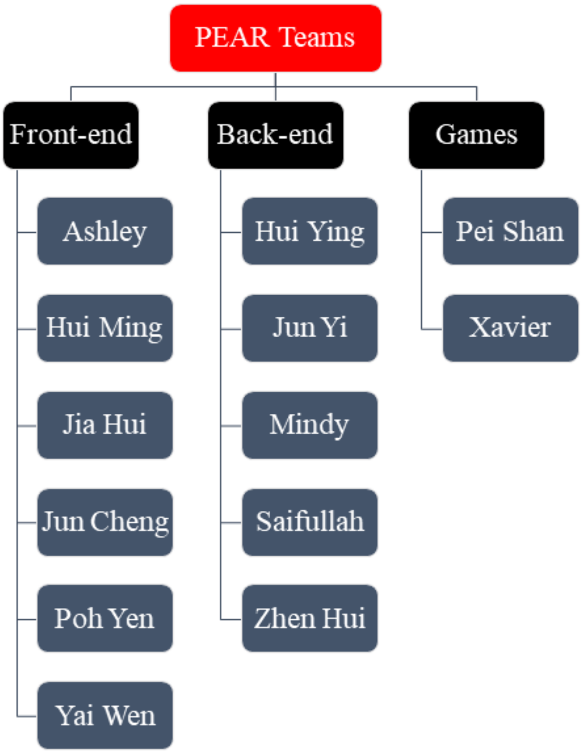


*Figure 2: Project Team Organisation*

### 1.4.1 Front-end Team

The front-end team works on the Android Application. The front-end team worked closely with back-end team to retrieve information from the database server. The retrieving of information was done via Application Programming Interface (API) which was built by the back-end team. Therefore, the front-end team must tell back-end team specifically what information is needed for each API.

### 1.4.2 Back-end Team

The back-end team works on the web application, application logic and database management. In addition, as mentioned above, back-end team also built API for Android Application.

### 1.4.3 Games Team

The games team builds mobile games for dementia patients. The games team designed the games to improve metal health of the dementia patient through cognitive stimulation.

## 1.5 Scope

The scope of the project includes designing User Interface (UI), implementing new UI with RESTful API calls, and last but not least Testing.

### 1.5.1 User Interface (UI) Design

As the existing PEAR application was developed over the years, it is necessary to update the look and feel of the mobile application. The UI element commonly used in the past might be dated now. The mobile application UI need to synchronise with the current user's expectation, therefore, we have to use updated and popular UI elements or UI practices that mobile application users are familiar with.

### 1.5.2 Implementation with RESTful API

One of the common practices of retrieving data from a server in a Client-Server environment is to use Application programming interface (API). In this project, the mobile application will use RESTful API provided by the project's back-end team to retrieve, create, update or delete data from the database server.

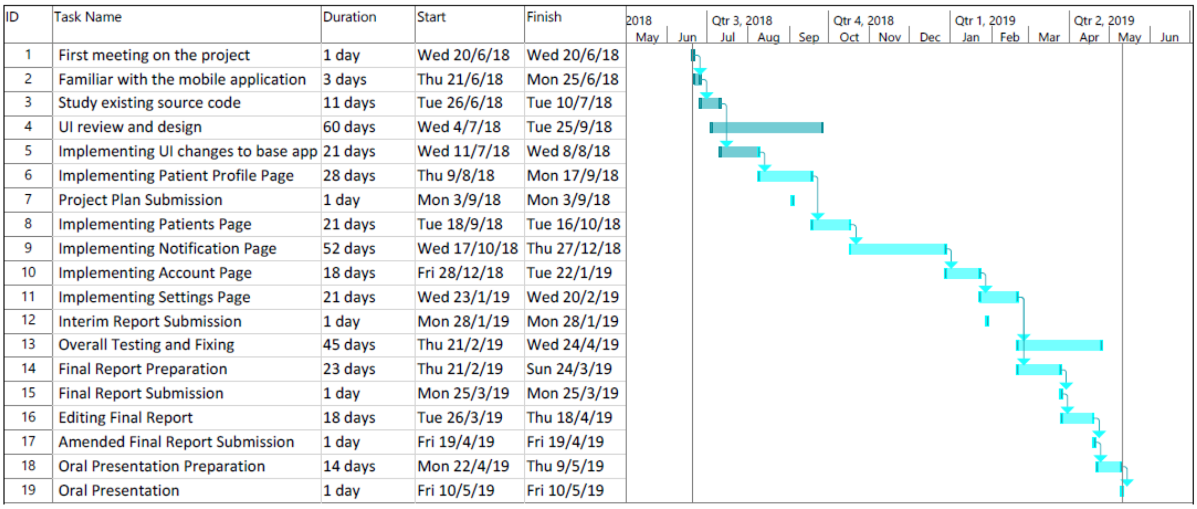
After discussing with the team, the author was in charge to implement the following pages in the mobile application:

- Notification
- Notification Details
- Patients
- Patient Profile
- Account
- User Information
- Settings
- About

### 1.5.3 Testing

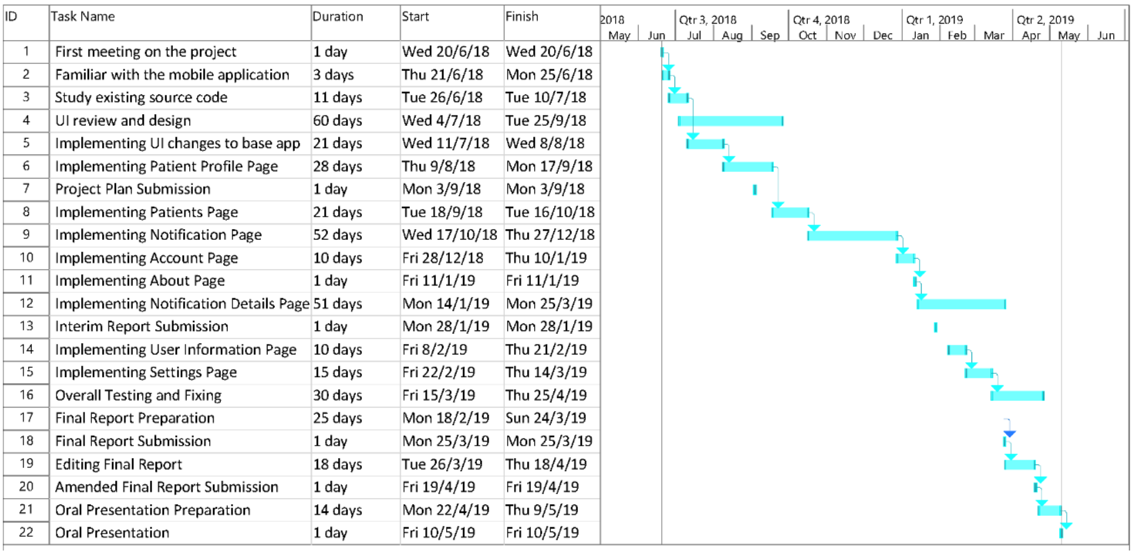
It important to do testing after implementation as it is to ensure the functionality works as intended. As multiple students are working on the same mobile application, merging of our code might break functionality or cause issue, therefore, it is necessary to do an overall testing and maintenance towards the end of the project.

## 1.6 Project Schedule



| ID | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | First meeting on the project | 1 day | Wed 20/6/18 | Wed 20/6/18 |
| 2 | Familiar with the mobile application | 3 days | Thu 21/6/18 | Mon 25/6/18 |
| 3 | Study existing source code | 11 days | Tue 26/6/18 | Tue 10/7/18 |
| 4 | UI review and design | 60 days | Wed 4/7/18 | Tue 25/9/18 |
| 5 | Implementing UI changes to base app | 21 days | Wed 11/7/18 | Wed 8/8/18 |
| 6 | Implementing Patient Profile Page | 28 days | Thu 9/8/18 | Mon 17/9/18 |
| 7 | Project Plan Submission | 1 day | Mon 3/9/18 | Mon 3/9/18 |
| 8 | Implementing Patients Page | 21 days | Tue 18/9/18 | Tue 16/10/18 |
| 9 | Implementing Notification Page | 52 days | Wed 17/10/18 | Thu 27/12/18 |
| 10 | Implementing Account Page | 18 days | Fri 28/12/18 | Tue 22/1/19 |
| 11 | Implementing Settings Page | 21 days | Wed 23/1/19 | Wed 20/2/19 |
| 12 | Interim Report Submission | 1 day | Mon 28/1/19 | Mon 28/1/19 |
| 13 | Overall Testing and Fixing | 45 days | Thu 21/2/19 | Wed 24/4/19 |
| 14 | Final Report Preparation | 23 days | Thu 21/2/19 | Sun 24/3/19 |
| 15 | Final Report Submission | 1 day | Mon 25/3/19 | Mon 25/3/19 |
| 16 | Editing Final Report | 18 days | Tue 26/3/19 | Thu 18/4/19 |
| 17 | Amended Final Report Submission | 1 day | Fri 19/4/19 | Fri 19/4/19 |
| 18 | Oral Presentation Preparation | 14 days | Mon 22/4/19 | Thu 9/5/19 |
| 19 | Oral Presentation | 1 day | Fri 10/5/19 | Fri 10/5/19 |

*Figure 3: Initial Project Schedule Gantt Chart*



| ID | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | First meeting on the project | 1 day | Wed 20/6/18 | Wed 20/6/18 |
| 2 | Familiar with the mobile application | 3 days | Thu 21/6/18 | Mon 25/6/18 |
| 3 | Study existing source code | 11 days | Tue 26/6/18 | Tue 10/7/18 |
| 4 | UI review and design | 60 days | Wed 4/7/18 | Tue 25/9/18 |
| 5 | Implementing UI changes to base app | 21 days | Wed 11/7/18 | Wed 8/8/18 |
| 6 | Implementing Patient Profile Page | 28 days | Thu 9/8/18 | Mon 17/9/18 |
| 7 | Project Plan Submission | 1 day | Mon 3/9/18 | Mon 3/9/18 |
| 8 | Implementing Patients Page | 21 days | Tue 18/9/18 | Tue 16/10/18 |
| 9 | Implementing Notification Page | 52 days | Wed 17/10/18 | Thu 27/12/18 |
| 10 | Implementing Account Page | 10 days | Fri 28/12/18 | Thu 10/1/19 |
| 11 | Implementing About Page | 1 day | Fri 11/1/19 | Fri 11/1/19 |
| 12 | Implementing Notification Details Page | 51 days | Mon 14/1/19 | Mon 25/3/19 |
| 13 | Interim Report Submission | 1 day | Mon 28/1/19 | Mon 28/1/19 |
| 14 | Implementing User Information Page | 10 days | Fri 8/2/19 | Thu 21/2/19 |
| 15 | Implementing Settings Page | 15 days | Fri 22/2/19 | Thu 14/3/19 |
| 16 | Overall Testing and Fixing | 30 days | Fri 15/3/19 | Thu 25/4/19 |
| 17 | Final Report Preparation | 25 days | Mon 18/2/19 | Sun 24/3/19 |
| 18 | Final Report Submission | 1 day | Mon 25/3/19 | Mon 25/3/19 |
| 19 | Editing Final Report | 18 days | Tue 26/3/19 | Thu 18/4/19 |
| 20 | Amended Final Report Submission | 1 day | Fri 19/4/19 | Fri 19/4/19 |
| 21 | Oral Presentation Preparation | 14 days | Mon 22/4/19 | Thu 9/5/19 |
| 22 | Oral Presentation | 1 day | Fri 10/5/19 | Fri 10/5/19 |

*Figure 4: Actual Project Schedule Gantt Chart*

Figure 3 shows the planned project schedule Gantt chart and figure 4 shows the actual project schedule Gantt chart. The reason for the discrepancy is some pages has sub-pages which the author did not consider in the initial project planning. For example, the account page portion contains 3 more sub-pages, the User Information page, the Settings page and the About page.

The other difference is the notification page portion of the application took a much longer time than expected to implement. The first reason being the notification details page is very complicated, and the author underestimated the difficult of implementing the page. The other reason being the back-end team took a long time to produce the API due to the complexity of the notification system. Nonetheless, all the pages stated were completed within the timeframe of the final year project.

# 2. Literature Review

## 2.1 Dementia

Dementia is a term that describes a group of symptoms affecting memory, thinking and social abilities severely enough leading to difficulties in performing everyday activities [5].

### 2.1.1 Symptoms of Dementia

The symptoms of dementia vary greatly but the common signs and symptoms includes [6]:

- Memory Loss
- Difficulty in communication and language
- Difficulty in paying attention
- Difficulty reasoning or problem-solving
- Problem with visual perception

### 2.1.2 Common Dementia Types

The most common type of dementia is Alzheimer's disease which accounts for 60 to 80 percent of cases [6]. The second most common type of dementia is Vascular dementia, which occurs after suffering from a stroke [6].

## 2.2 Person-centred Care

As mentioned in the introduction, person-centred care is an ideology of care focus on the needs of the individual where they become the centre of the caring process [4].

### 2.2.1 Key ideas of Person-centred Care

The key ideas behind person-centred care for dementia are as follows [7]:

- Treating the person with dignity and respect

- understanding their history, lifestyle, culture and preferences, including their likes, dislikes, hobbies and interests
- looking at situations from the point of view of the person with dementia
- providing opportunities for the person to have conversations and relationships with other people
- ensuring the person has the chance to try new things or take part in activities they enjoy.

### 2.2.2 Benefits of Person-centred Care

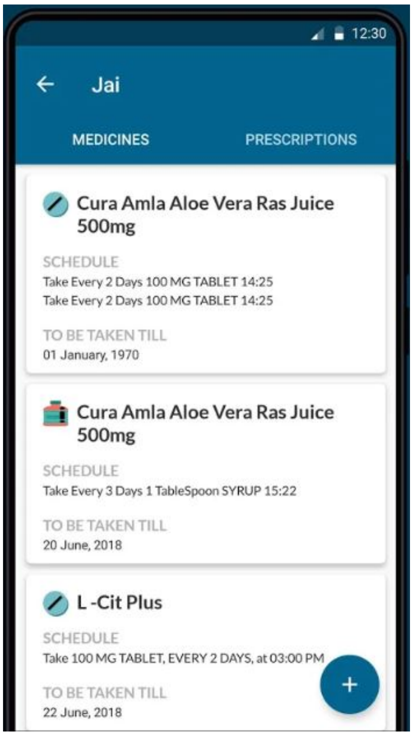
The first benefit of person-centred care is it allows a person with dementia to take part in the doing things they enjoy [7]. The other benefit is that it can be an effective way of preventing and managing behavioural and psychological symptoms of dementia [7].

## 2.3 Existing Caregiving System

**Caregiver** is a health monitoring Android application developed by Madhavbaug Clinic in India. The application allows the caregiver to provide great care and personalised services to patients in the hospital.

**Caregiver** application provides the following features:

- Add Patient
- Add Event / Alert / Appointment
- Assign Diet
- View / Add Health Vitals
- View / Add Medicines
- View / Add Vital Limits
- Call/chat with care team members and patients



Cura Amla Aloe Vera Ras Juice 500mg

SCHEDULE
Take Every 2 Days 100 MG TABLET 14:25
Take Every 2 Days 100 MG TABLET 14:25

TO BE TAKEN TILL
01 January, 1970

Cura Amla Aloe Vera Ras Juice 500mg

SCHEDULE
Take Every 3 Days 1 TableSpoon SYRUP 15:22

TO BE TAKEN TILL
20 June, 2018

L -Cit Plus

SCHEDULE
Take 100 MG TABLET, EVERY 2 DAYS, at 03:00 PM

TO BE TAKEN TILL
22 June, 2018

*Figure 5: Screenshot of Caregiver app*

Figure 5 on the right shows a screenshot of the Caregiver application.

# 3. Requirement Specification and Analysis

## 3.1 Use Case Diagram

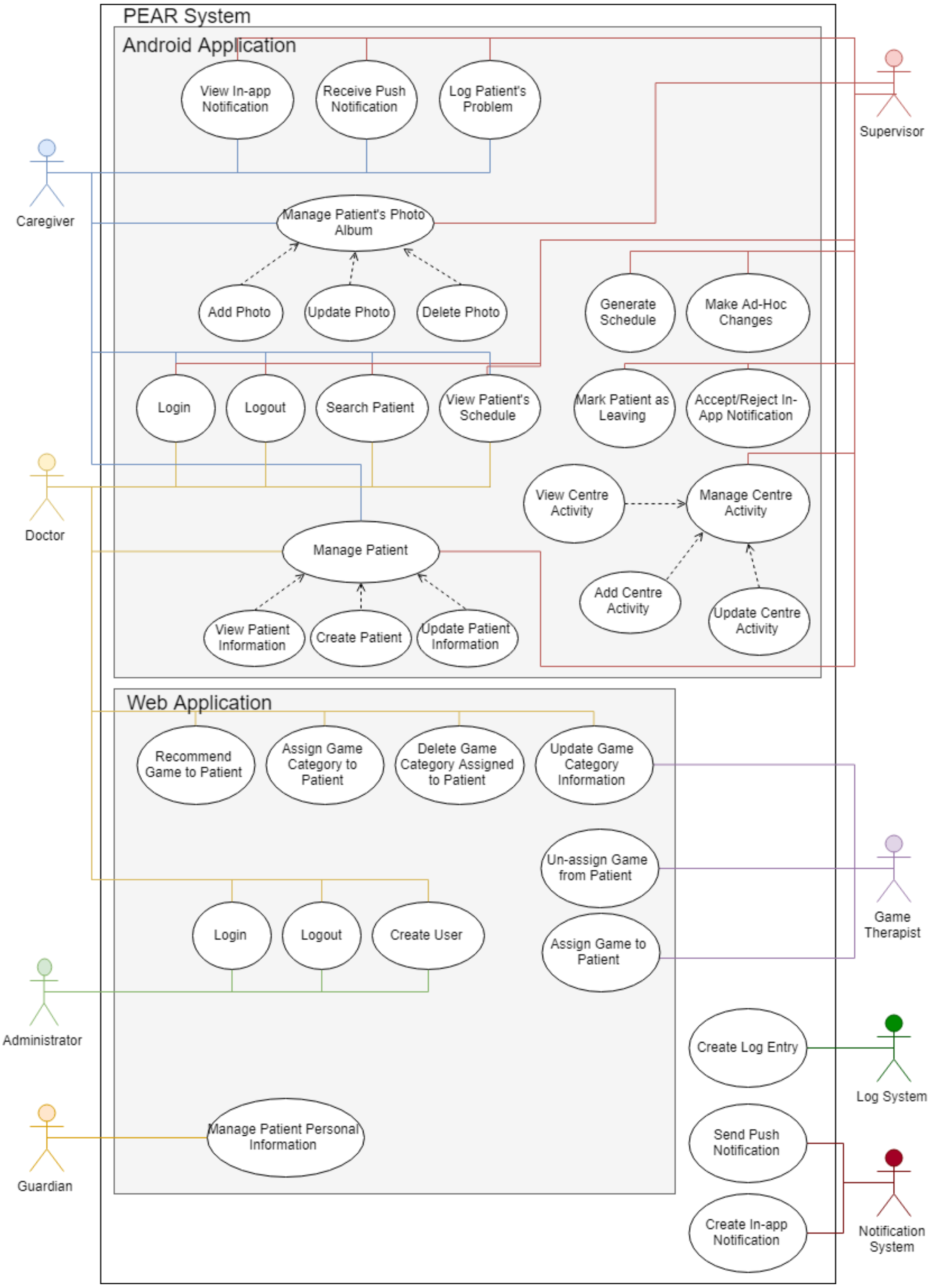


*Figure 6: Use Case diagram of PEAR system*

## 3.2 User Roles and Responsibilities

Figure 6 above shows the use case diagram of PEAR system. There are 6 types of users who will use the PEAR system. The roles and their responsibilities are:

### 3.2.1 Caregivers

A caregiver oversees taking care of patients in a care centre. A caregiver can have many patients under his/her care. The caregiver will facilitate patient in day-to-day activities in the care centre such as bringing the patient to the right location based on the auto-generated schedule, key in the patient's vital information into the system, and change patient activity preferences all done via the Android application.

### 3.2.2 Supervisor

The supervisor does what a caregiver does, but he/she is also in charge of all the caregivers. Whenever a caregiver changes any patient's information, it will have to be approved by the supervisor first before it gets to appear in the system. A dedicated page with more features was implemented in the Android App to support the supervisor extra responsibilities such as regenerating patients schedule and manage the available activities for patients.

### 3.2.3 Guardians

The guardian is the next-of-kin or contact person of the patient. The guardian can use the web application to upload photos of the patients and other information to help caregivers understand the patient better.

### 3.2.4 Game Therapists

The game therapist is responsible in game treatment for the patient. They can assign games to a patient or remove games assigned to a patient.

### 3.2.5 Doctors

The doctor concerns with the patient's mental and physical health. They can recommend or not recommend an activity to a patient based on the patient's mental and physical capability. They can also put notes in the system so that the caregiver can keep a look out of the patient's well-being.

### 3.2.6 System Administrators

The system administrator maintains the entire PEAR system. One of the tasks of the system administrator is to create a new user in the system.

## 3.2 Functional Requirements

The following are partial functional requirements (based on pages that author implemented) for the Android application:

1. The application must allow the user to login to the system.

    1.1 The application must allow the user to enter their user name and password.

    1.2 The application must verify with the application server on the user name and password.

2. The application must allow the user to view their notifications.

    2.1 The notifications must be listed in chronological order.

    2.2 The unread notification must be highlighted in light orange.

    2.3 The application must allow the user to receive push notification.

3. The application must allow the user to view notification details.

    3.1 The application must allow the supervisor to accept/reject the notification.

    3.2 The application must allow the caregiver to view the status of the request made.

    3.3 The application must allow the supervisor to enter reject reason when doing a reject of the notification request.

    3.3 The application must allow the user to see the changes made.

4. The application must allow the user to view a list of all patients in the care centre.

4.1 The application must list patients in alphabetical order.

4.2 The application must allow the user to search for a patient.

4.3 The application must allow the user to do an alphabetical fast scroll.

5. The application must allow the user to view patient profile.

5.1 The application must show the patient information.

5.2 The application must show the patient's schedule of the day.

6. The application must allow the user to access the account page.

6.1 The application must show the name of user and role of the logged in user.

7. The application must allow the user to view their personal information.

7.1 The application must allow the user to edit their personal information.

8. The application must allow the user to change in-app notification ringtone in application settings.

9. The application must allow the user to view the about page.

10. The application must allow the user to log out of the application.

## 3.3 Non-Functional Requirements

- **Flexibility**

  The Android application must be flexible to allow adding of new features easily without breaking other parts of the application.

- **Security**

  The Android application must ensure no unauthorised access to the system. The connection to the application server must have end-to-end encryption.

- **Response Time**

  The Android application must response to the user action no longer than 5 seconds.

- **Usability**

  The Android application must be easy to learn and use. The user should be able to learn how to use the application within 3 days.

## 3.4 System Assumptions

The following PEAR system assumptions are set by the predecessor teams [8]:

- Only one user can be assigned as supervisor.
- Each patient can only be assigned to one caregiver.
- Each patient can only be assigned to one doctor.
- The generated patient’s schedule is fixed to the operating hours of care centre from 9am to 5pm on weekdays (Monday to Friday).
- There are no overlapping events for a patient’s schedule.

## 3.5 Patient’s Information

Comprehensive information of the patient is essential to provide person-centred care. The following information of the patient are stored in the PEAR system [8]:

*Table 1: Patient's Information*

| **Category** | **Information** | |
|---|---|---|
| *Activity Preferences* | For each activity:<br>• Activity name<br>• Doctor recommendation (recommend/not recommend/neutral)<br>• Patient rating (like/dislike/neutral) | |
| *Allergy* | For each allergy:<br>• Allergy name<br>• Reaction<br>• Notes | |
| *Holiday Experience* | For each holiday:<br>• Country<br>• Start date and end date<br>• Description<br>• Photos | |
| *Personal Information* | • First and last name<br>• NRIC<br>• Profile photo<br>• Gender | • Education level<br>• Occupation<br>• Religion<br>• Exercise frequency |

| | | |
|---|---|---|
| | • Date of birth<br>• Age<br>• Home number<br>• Mobile number<br>• Home address<br>• Preferred name<br>• Preferred language<br>• Notes<br>• Living with | • Diet<br>• Pet<br>• Sexual status<br>• Drugs use<br>• Caffeine consumption<br>• Alcohol consumption<br>• Tobacco use<br>• Guardian’s name<br>• Guardian’s mobile number |
| *Photo Album* | • Photo albums<br>• Photos | |
| *Prescription* | For each prescription:<br>• Drug name<br>• Before/after meal<br>• Start date and end date<br>• Frequency<br>• Dosage<br>• Notes | |
| *Problem Log* | • Date of creation<br>• Problem category<br>• Notes | |
| *Routine* | • Routine name<br>• Repetition<br>• Start date and end date<br>• Start time and end time | |
| *Vital* | • Date Taken<br>• Time Taken<br>• Before/After Meal<br>• Blood Pressure (Diastolic)<br>• Blood Pressure (Systolic)<br>• Height (cm)<br>• Weight (kg)<br>• Temperature (°C)<br>• Notes | |

# 4. Project Environment and Structure

## 4.1 Project Environment

The following applications and services were used to develop the Android applications:

- **Figma**

  Figma is a cloud-based tool for interface design. The high-fidelity prototype of the android application was created using Figma.

- **GitHub**

  GitHub is used to host the android application project for the purposes of version control and ease of integration.

- **Android Studio**

  Android Studio is the integrated development environment (IDE) used to code and run the Android application.

- **Restlet Client**

  API testing was done in the Restlet Client web-based tool.

The following software and services were used for project management and communication among the team members:

- **Dropbox**

  Dropbox was used to store project deliverables and important documentations.

- **Google Drive**

  All the documents used to communicate between the front-end team and back-end team were stored in Google Drive.

- **Trello**

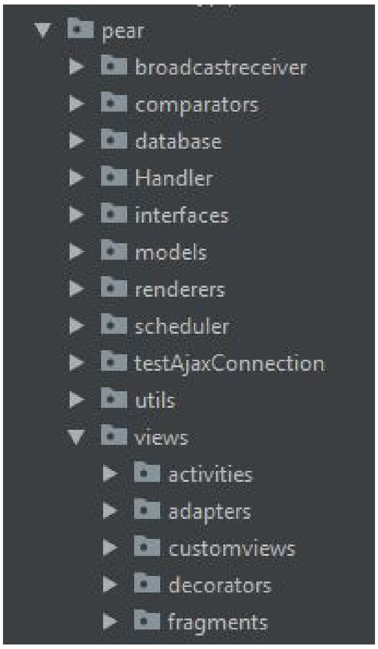
  Trello was the platform to update and communicate with one another on the progress and problems or any bugs faced.

## 4.2 Project Structure

The source files of the Android application are organised as on the figure 7 on the right.

The most used folders are **activities**, **fragments** and **database** folder. Depends on the page that is implementing, the java class for the page can be stored in either **activities** or **fragments** folder**.** The **database** folder is used to store all the classes which interacts with the Application programming interface (API).

The other packages and files they contain are described in table 2.



```
▼ pear
    ▶ broadcastreceiver
    ▶ comparators
    ▶ database
    ▶ Handler
    ▶ interfaces
    ▶ models
    ▶ renderers
    ▶ scheduler
    ▶ testAjaxConnection
    ▶ utils
    ▼ views
        ▶ activities
        ▶ adapters
        ▶ customviews
        ▶ decorators
        ▶ fragments
```

*Figure 7: Android Application Source Code Organisation*

*Table 2: Android Packages Name and Description*

| ***Package Name*** | **Description** |
|---|---|
| *broadcastreceiver* | Classes that are used to implement the notification system. |
| *comparators* | Classes used to contain methods for doing sorting or comparison operation. |
| *interfaces* | Interfaces that can be reused. |
| *models* | Entity classes of the application. |
| *renderers* | Classes that binds data to the user interface. |
| *scheduler* | Classes that helps to generate a patient's schedule. |
| *utils* | Utility classes that contain global variables or global reusable methods. |

# 5. System Design

## 5.1 General User Interface Design

To provide a great user experience, it is essential to have a good user interface design. The user interface of PEAR Android application had gone through a major revamp for a cleaner and more modern look. The new design takes inspiration from latest Google Material Design 2 which is used in Android 9 Pie and many Google-made applications (Figure 8).

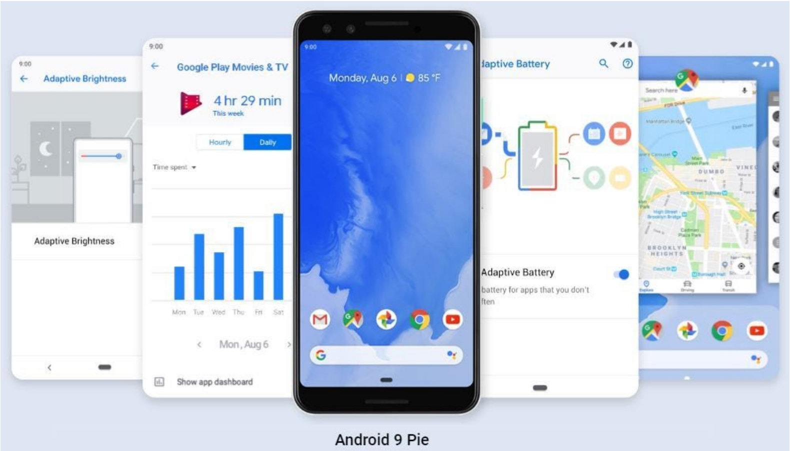


*Figure 8: Screenshot of Android 9 Pie and Google Apps [9]*

### 5.1.1 Rounded corners and Circle

The application now features large radius rounded corners for all the rectangular buttons and uses the circle for the picture in replace of the square picture. The rounded corners make information easier to process as they allow the viewer to put focus inside the rectangle [10] (Figure 9). It is also inline with Android 9 Pie user interface and newer phones and tablet which was made with rounded corners display.

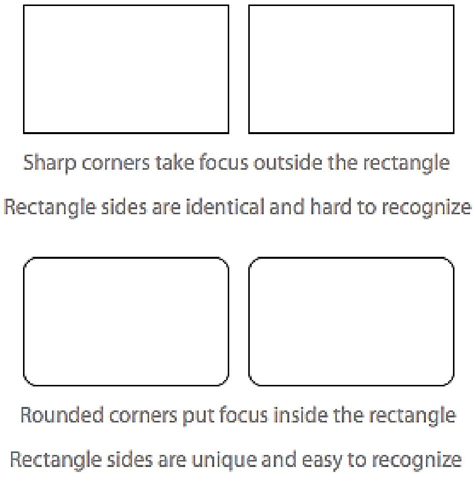

Sharp corners take focus outside the rectangle
Rectangle sides are identical and hard to recognize
Rounded corners put focus inside the rectangle
Rectangle sides are unique and easy to recognize

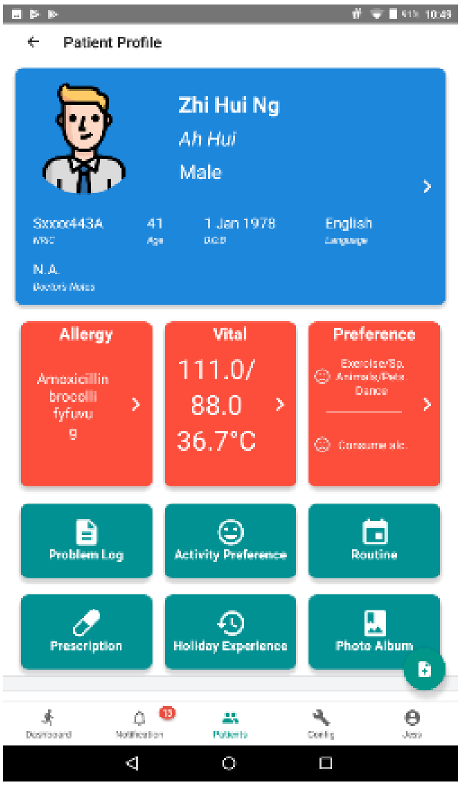
*Figure 9: Sharp corners vs rounded corners [10]*

### 5.1.2 Cards

The application uses cards to segment information on a page. Cards divide information into meaningful sections which aids the user in scanning and skimming information. The use of cards can be seen in the Patient Profile page (Figure 10).

### 5.1.3 Colour

The primary colour of the application is white which gives a very clean look. The application also used colour which stands out such as orange, blue and green to call for user attentions. One example of this can be seen from the login page of the application where the login button is in orange (Figure 11). The other example is Patient Profile page where the different information is being divided with different colours and the orange cards contains the most important information (Figure 10).



Patient Profile

Zhi Hui Ng
*Ah Hui*
Male

Sxxxx443A — NRIC
41 — Age
1 Jan 1978 — DOB
English — Language
N.A. — Doctor's Notes

Allergy: Amoxicillin broccolli fyfuvu g

Vital: 111.0/ 88.0 36.7°C

Preference: Exercise/Sp. Animals/Pets. Dance; Consume alc.

*Figure 10: New Patient Profile Page*

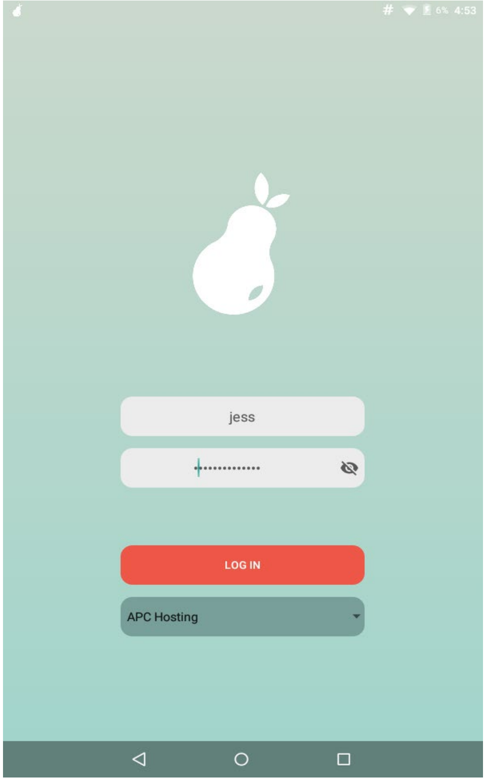


*Figure 11: New Login Page*

## 5.2 Prototype

To find out the user experience of the new user interface (UI), the team had developed a high-fidelity (Hi-Fi) prototype of entire android application (Figure 12). It was created in Figma, a web-based tool for creating interactive UI. During the development, every design choice was made based on the needs of the user. Many UI considerations such as colour of the UI elements, placement of buttons, displaying of the right information were debated and discussed among the team and with the supervisor.

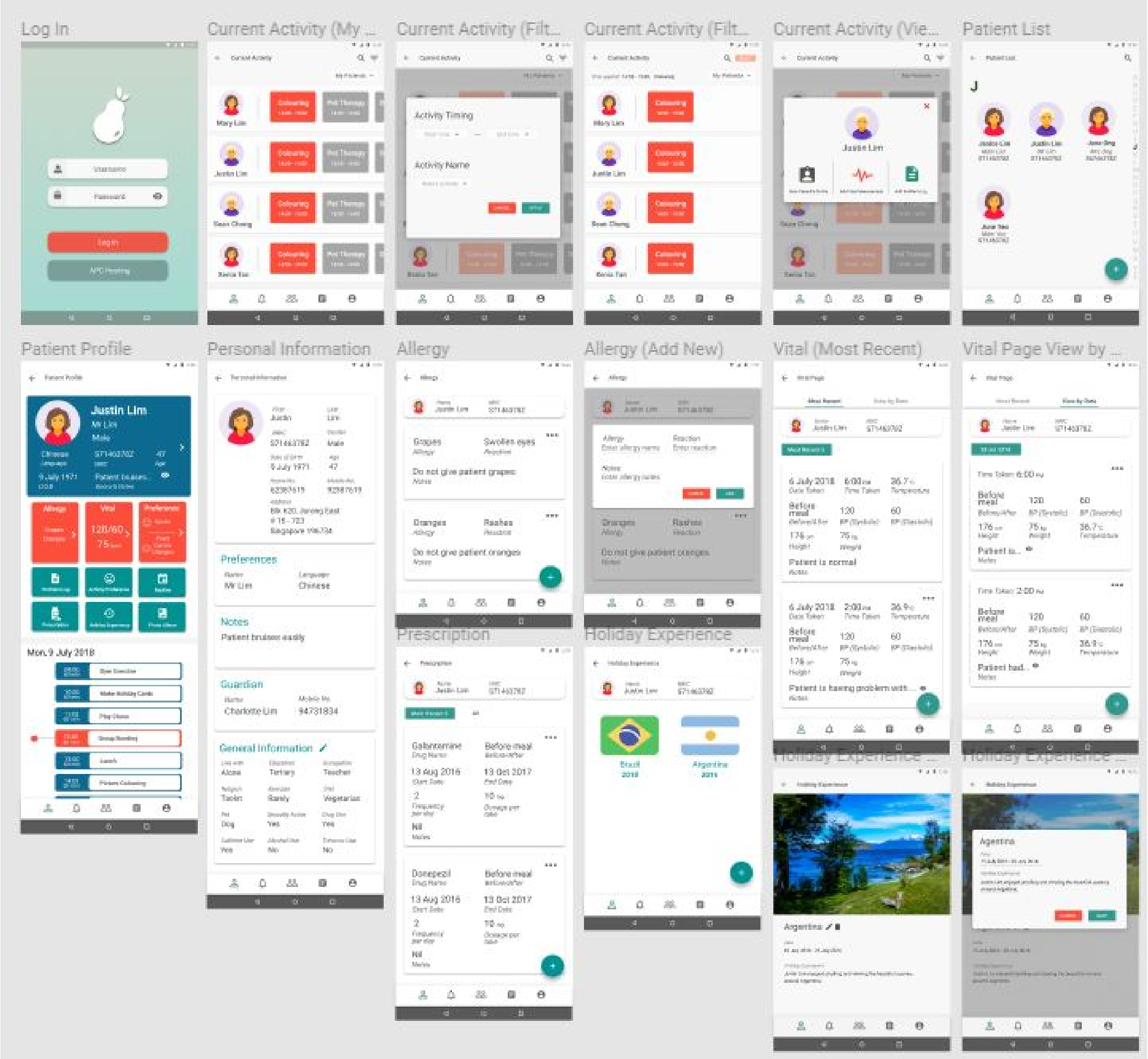


*Figure 12: Screenshot of partial Hi-Fi prototype in Figma*

# 6. System Implementation

## 6.1 RESTful API

The Android application was originally designed to interact with the database server directly. It costs a lot of overheads because for every (Structured Query Language) SQL query request made to the database server, it will need to open and close the connection. Consequently, the process of retrieving data or inserting data into the database server is very slow. Therefore, the previous batch of final year project students started migrating the application to use RESTful API to retrieve data. RESTful API is based on Representational State Transfer software architectural style which is efficient and uses little bandwidth. The back-end team will produce API which will output data in JavaScript Object Notation (JSON) format. The front-end team will then parse the JSON and display it out in the application. As a result, the time taken to retrieve data was significantly improved with the use of API. However, the loading time for some pages in the application was still not ideal because we later realised all information pertaining to a patient was retrieved at one go. To further improve the performance of data retrieval, thus new APIs were created by the back-end team to only fetch information needed for the displaying of a page. There will be at least one API for each application page. All the pages the author implemented were using RESTful API, retrieve information using GET request, add new information using POST request, or update information using PUT request.

## 6.2 ActionBar

The action bar is the top bar in an Android application's activity window. The ActionBar is used to display page title and buttons for functions such as search or filter. In the old design, the ActionBar was thick and takes up many of the precious screen estate. To leave more space for the page, the ActionBar is now made to be thinner. Figure 13 below shows the difference between the old and new ActionBar.

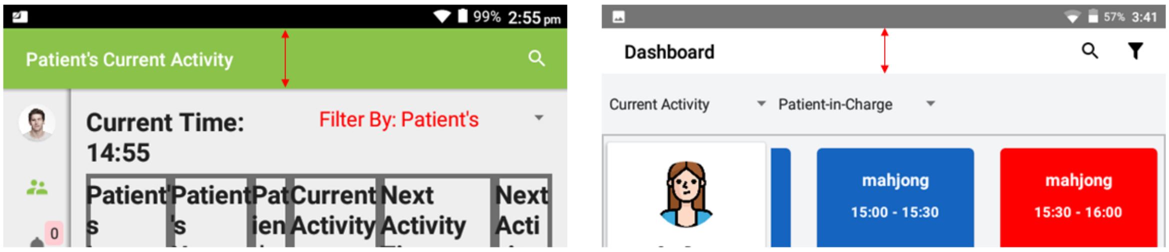


*Figure 13: The old vs new ActionBar*

## 6.3 Navigation bar

In the old design of the application, the navigation bar was a vertical bar on the left side of the screen. The redesigned navigation bar is now a horizontal bar at the bottom of the screen. The decision to shift the navigation bar is because of two factors, user familiarity and screen aspect ratio. For user familiarity, most of the Android applications nowadays have a bottom navigation bar. The other factor is screen aspect ratio, most of the Android tablets or phones in the market has a 16:9 or 16:10 aspect ratio screen. Therefore, to get the most out the horizontal screen space, it is better to use a bottom navigation bar as the application is expected to be mostly used in portrait mode. Figure 14 below shows the difference between old and new navigation bar.

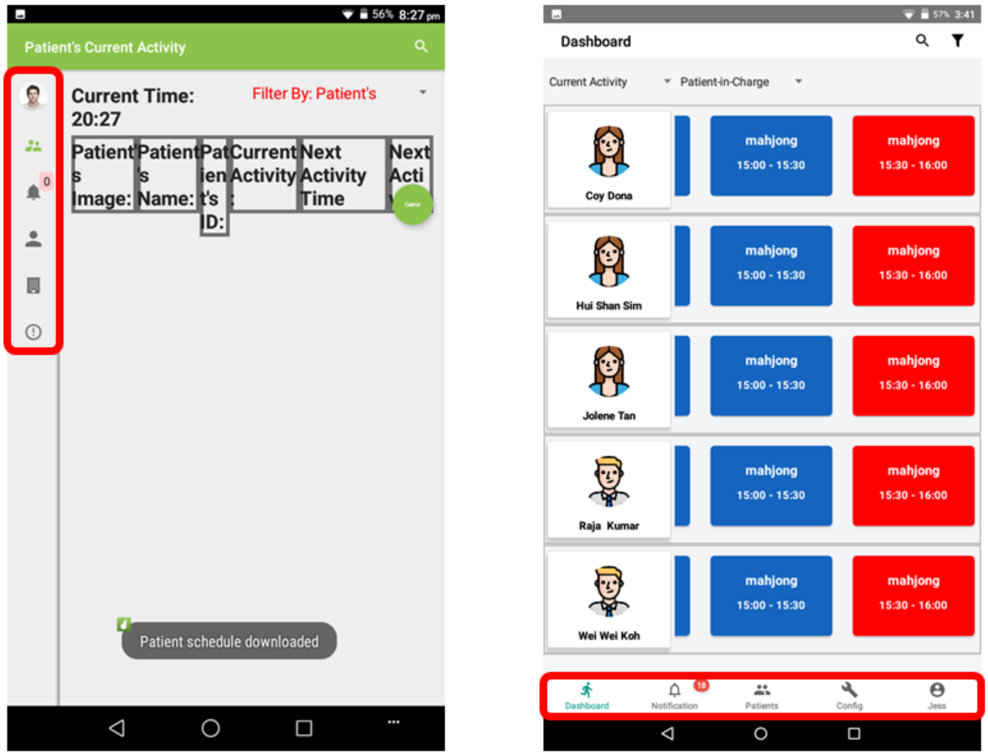


*Figure 14: Old vs New Navigation Bar*

### 6.3.1 Navigation bar destinations

As shown in Figure 14, the top-level page destinations on the navigation bar are:

1. Dashboard
2. Notifications
3. Patients
4. Care Centre Configuration (only visible to Supervisor)
5. Account (indicated by the user name of the user)

The Account page is a new destination which replaces the About page on the old navigation bar. There will be more details on the new Account page under the Account page section later in the report.

### 6.3.2 Navigation Bar Implementation Challenges

As the Android application was developed over the years by different students, a mixture of Android Activity and Fragment classes were used. It became a challenge to implement the new bottom navigation bar as it should appear on all pages of entire application (other than the Settings page). The implementation for those pages using fragment class such as Dashboard, Notifications, and Care Centre Configuration page is a none issue as the new bottom navigation bar can be simply added to the MainActivity class. As for all other pages using Activity class, the navigation bar cannot be simply duplicated to multiply Activity class as there are too many pages and it will be difficult to control the destinations. Therefore, the solution was to create another Activity class called BaseNavigationActivity. All the pages using Activity class will just need to inherit the BaseNavigationActivity class to add the navigation bar. The user flow of destinations is simple because there are only two set of navigation bar. The result is two Activity classes with the navigation bar are switching back and forth when any of the navigation destination button is selected.

### 6.3.3 Notification Badge

Inside the notification destination button, the red badge (Figure 15) is used to show unread notification count. The badge will be hidden if the user has none unread notification.

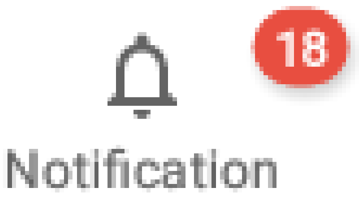


*Figure 15: Notification Badge*

As there are two sets of navigation bar, one in MainActivity and one in BaseNavigationActivity, hence the method to retrieve the unread notification count was created in a separate class. Both the MainActivity and BaseNavigationActivity will call the same method to get the unread notification count.

The method to retrieve unread notification count will only be called when the user opened a new page or when the user relaunched a page. Therefore, a broadcast service was created to update the unread notification count whenever a new notification arrives, used when the application is idle.

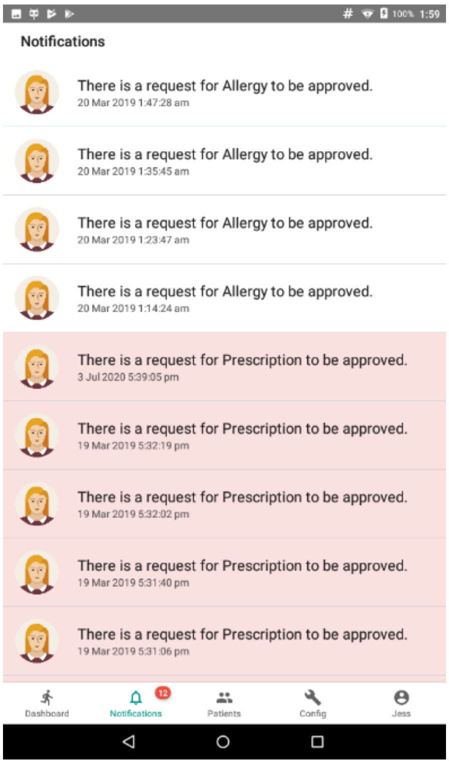
## 6.4 Notifications

Whenever a caregiver changes information on a patient, a notification will be sent to supervisor tablet or phone to inform the supervisor to approve the request. A notification will also be sent to the caregiver to assure the request had been sent and he/she will be able to keep track of it later. Vice versa, when a supervisor accepted or rejected a request, the caregiver who had made the request will receive a notification. Notifications has two parts, in-app notifications page and push notification.

### 6.4.1 In-app notifications page

The notifications page is used to display a list of notifications of the user. Figure 16 shows the new notifications page.

The notification page was a total revamped from the old design. In the old design, the list of notifications is grouped by patient name. The list of notifications is now in chronological order which most of the applications are using today.  Those notifications which are unread, meaning have not been accepted or rejected will be highlighted in light orange. Each of the notification is selectable, upon selecting on it, the user will be brought to the Notification Details page.



Notifications

There is a request for Allergy to be approved.
20 Mar 2019 1:47:28 am

There is a request for Allergy to be approved.
20 Mar 2019 1:35:45 am

There is a request for Allergy to be approved.
20 Mar 2019 1:23:47 am

There is a request for Allergy to be approved.
20 Mar 2019 1:14:24 am

There is a request for Prescription to be approved.
3 Jul 2020 5:39:05 pm

There is a request for Prescription to be approved.
19 Mar 2019 5:32:19 pm

There is a request for Prescription to be approved.
19 Mar 2019 5:32:02 pm

There is a request for Prescription to be approved.
19 Mar 2019 5:31:40 pm

There is a request for Prescription to be approved.
19 Mar 2019 5:31:06 pm

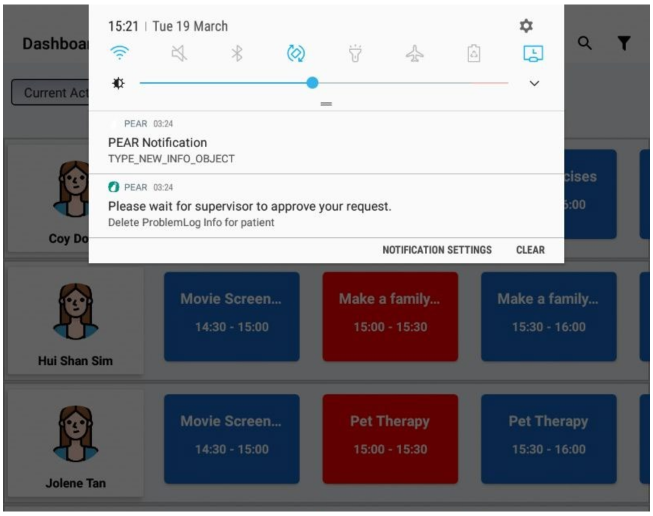
*Figure 16: New Notifications Page*

For each notification, the photo of the user who initiate the notification will be shown on the left. However, as the current database does not support storing of user profile photo thus for now, it is just a placeholder. Followed by on the right side of the photo, it will be the notification message and below it is the notification creation time.

### 6.4.2 Push Notification

The push notification will be displayed in the user's phone or tablet notification panel. In the application, the push notification was originally implemented with Google Cloud Messaging (GCM) which had been since deprecated. The application was updated with the new Firebase Cloud Messaging (FCM) implementation. The device registration token was used to send notification by the application server one by one. Device registration token is a token generated by FCM Software development kit (SDK) on the initial startup of the application. In order to make the back-end send push notification to the correct individual device, when the user login to the application, device registration token was captured and uploaded to the back-end sever. The device registration token will be stored in the Firebase Real-time Database. The back-end logic will then link the device token to the user logged in.

Once the user received the notification, he/she will be able to see the push notification in notification panel (Figure 17). By selecting on the push notification, the user will be brought to the Notification Details page.



15:21 | Tue 19 March

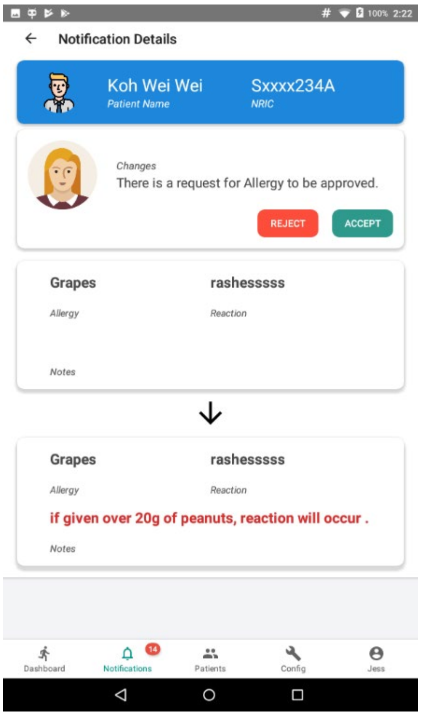
PEAR 03:24
PEAR Notification
TYPE_NEW_INFO_OBJECT

PEAR 03:24
Please wait for supervisor to approve your request.
Delete ProblemLog Info for patient

Figure 17: Push Notification in Device Notification Panel

## 6.5 Notification Details

The notification details page is used to display details of the request made. On the same page, the supervisor will accept or reject a request. The caregiver will use the page to see the status of the request.



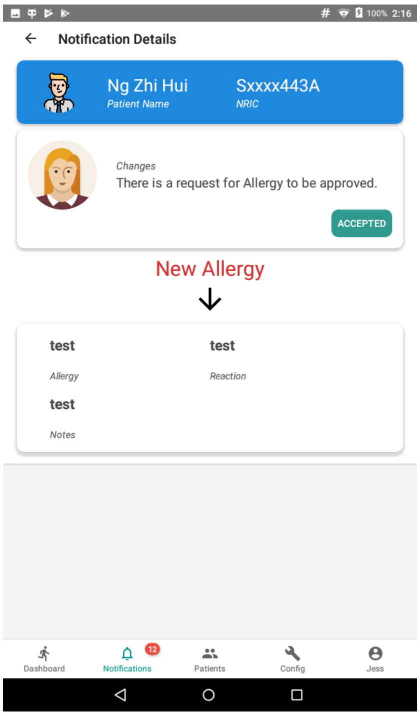
Koh Wei Wei
Patient Name
Sxxxx234A
NRIC

Changes
There is a request for Allergy to be approved.

Grapes
Allergy
rashesssss
Reaction
Notes

Grapes
Allergy
rashesssss
Reaction
if given over 20g of peanuts, reaction will occur .
Notes

*Figure 18: New Notification Details (show changes made)*



Ng Zhi Hui
Patient Name
Sxxxx443A
NRIC

Changes
There is a request for Allergy to be approved.

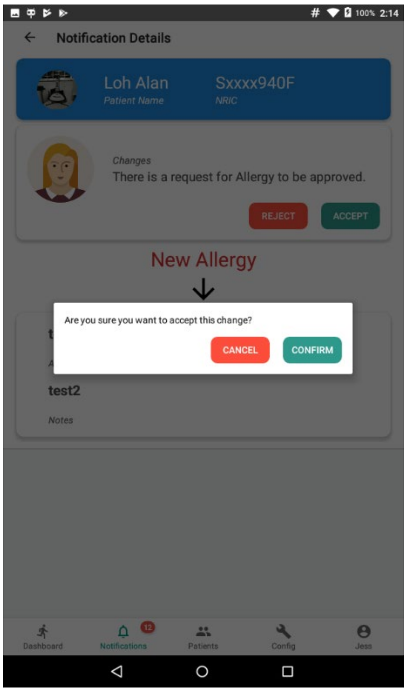
test
Allergy
test
Reaction
test
Notes

*Figure 19: New Notification Details (new information)*

In the new design, consistent with other pages in the application, the top card will show the patient photo, patient's name and masked NRIC. It's followed by a card which show the notification message and the photo of the user who initiated the request. In the card, if the user is supervisor and if the request is still pending, it will show two buttons Accept and Reject. If the user is a caregiver, he/she will not see the buttons but instead he/she will see the status of the request.

Below the card is either changes made (Figure 18) or if it's a new information, it will be labelled as new (Figure 19). The challenge of implementing this page is that there are many categories of the patient's information, the page needs to display the correct information layout accordingly. The application currently only support displaying of changes for allergy information category as the current database log format did not keep a full record of changes made and what was unchanged for other information categories.

If the supervisor selects the Accept button, a prompt will be shown to ask the supervisor to confirm (Figure 20). If the Reject button is being selected, a prompt with reason input box will be shown to ask the supervisor to input a reason as it is mandatory to include a reason if it is a reject (Figure 21).



Notification Details

Patient Name: Loh Alan
NRIC: Sxxxx940F

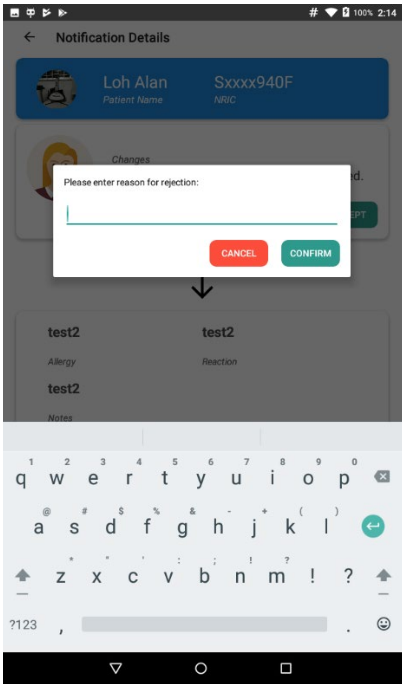
Changes: There is a request for Allergy to be approved.

Are you sure you want to accept this change?

Notes: test2

*Figure 20: New Notification Details*

*(When supervisor select Accept)*



Notification Details

Patient Name: Loh Alan
NRIC: Sxxxx940F

Please enter reason for rejection:

Allergy: test2
Reaction: test2
Notes: test2

*Figure 21: New Notification Details*

*(When supervisor select Reject)*

## 6.6 Patients

The Patients page is used to display a list of all patients in the care centre. Each patient in the list is selectable. Upon selecting any of the patient, it will bring the user to the Patient Profile page. The idea behind the new design is to maximise screen space and reduce white space. Therefore, each selectable patient item is now more compact, while keeping about the same amount of information except for preferred language.

Comparing the old design and the new design (Figure 22), the page can now show 6 instead of 2 patients with the same screen space. The other change is that the patient photo is now smaller and is in a circular shape instead of a square.

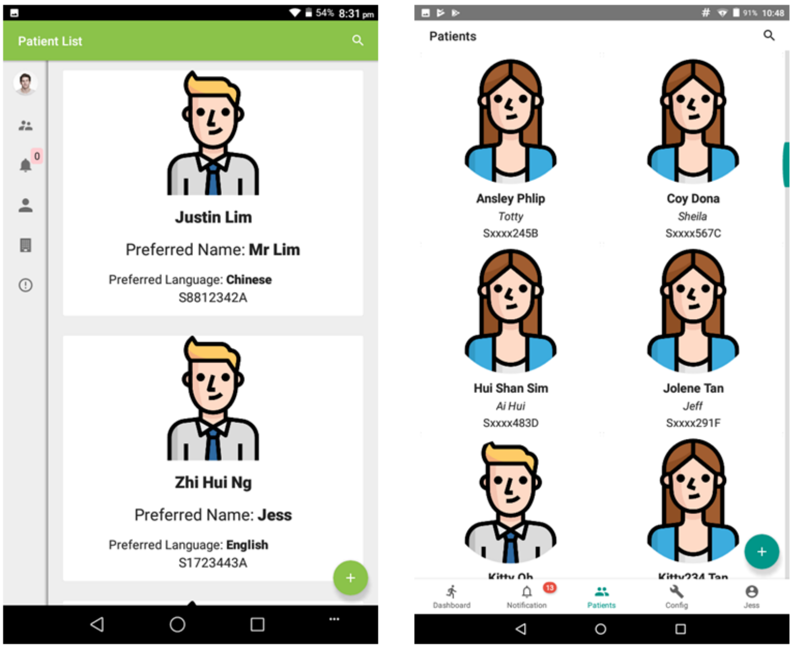


*Figure 22: Old vs New Patients*

A new feature which was implemented is the alphabetical fast scroll (Figure 23). It is useful when the list of patients is long, and the user want to find the patient based on the patient's name. To use the alphabetical fast scroll, the user must drag the scroll bar on the right.

However, even with the alphabetical fast scroll, the search function at the top right corner is still being kept because it is still a powerful and useful feature. No change to the old design is the add patient button at the bottom right corner.

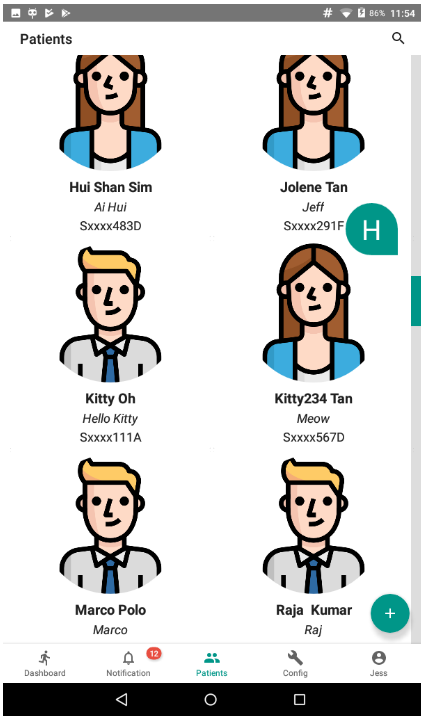
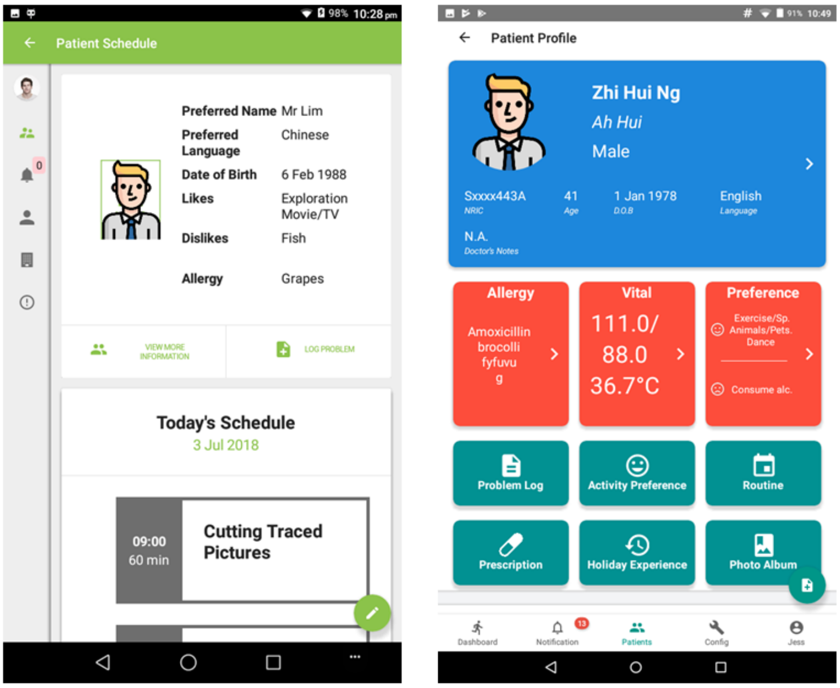


*Figure 23: Alphabetical Fast Scroll in Patients page*

## 6.7 Patient Profile

The patient profile page is used to display brief information of a patient and schedule of the day for the patient. In the old design (Figure 24), the page has little information and many unused spaces. The old design did not include the necessary information such as full name and NRIC of the patient. Therefore, in the new design (Figure 24), the unused space was reduced. The page was filled with more information and is further divided into cards. The idea with the new design is at a glance, the user should be able to find out all the basic and important information of the patient.



Patient Schedule

| | |
|---|---|
| Preferred Name | Mr Lim |
| Preferred Language | Chinese |
| Date of Birth | 6 Feb 1988 |
| Likes | Exploration<br>Movie/TV |
| Dislikes | Fish |
| Allergy | Grapes |

Today's Schedule
3 Jul 2018

09:00
60 min
Cutting Traced Pictures

Patient Profile

Zhi Hui Ng
*Ah Hui*
Male

| Sxxxx443A | 41 | 1 Jan 1978 | English |
|---|---|---|---|
| *NRIC* | *Age* | *D.O.B* | *Language* |

N.A.
*Doctor's Notes*

Allergy
Amoxicillin
brocolli
fyfuvu
g

Vital
111.0/
88.0
36.7°C

Preference
Exercise/Sp.
Animals/Pets.
Dance
Consume alc.

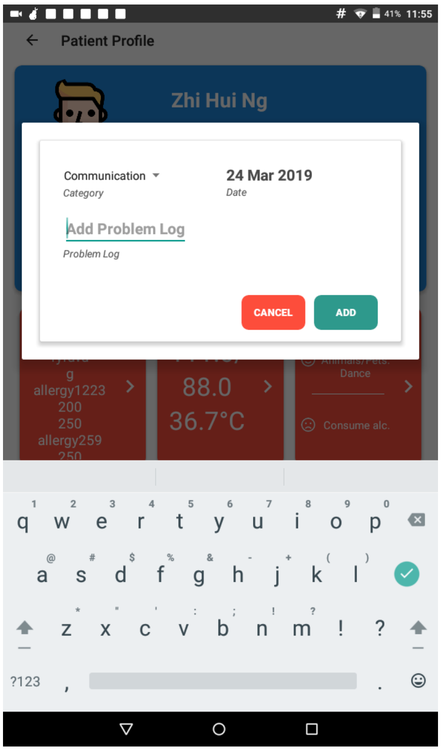
*Figure 24: Old vs New Patient Profile Page*

In the new design, the first card in blue contains some information of a patient. The patient's information are the full name, preferred name, gender, masked NRIC, age, date of birth, preferred language, and the doctor's note. The allergy card (in orange) is used to show a list of allergies that the patient has. The Vital card (in orange) shows the most recent patient's vital taken. The preference card (in orange) will show some likes and dislikes of the patient.

The next six cards deemed as less important, therefore their information is not displayed in the page. The six cards, problem log, activity preference, routine, prescription, holiday experience, and photo album were represented by icons. All the cards are selectable, upon selecting a card, the application will bring to the details page of the selected category.

Just as in the old design, scrolling down the same page will show the full patient's schedule of the day (Figure 25). However, changes were made to reduce the unused space for each event in the schedule. The current event will be highlighted in red.

By selecting on the circular button located at the bottom right corner, the user is able to access the add problem log feature on the same page (Figure 26).


Patient Profile

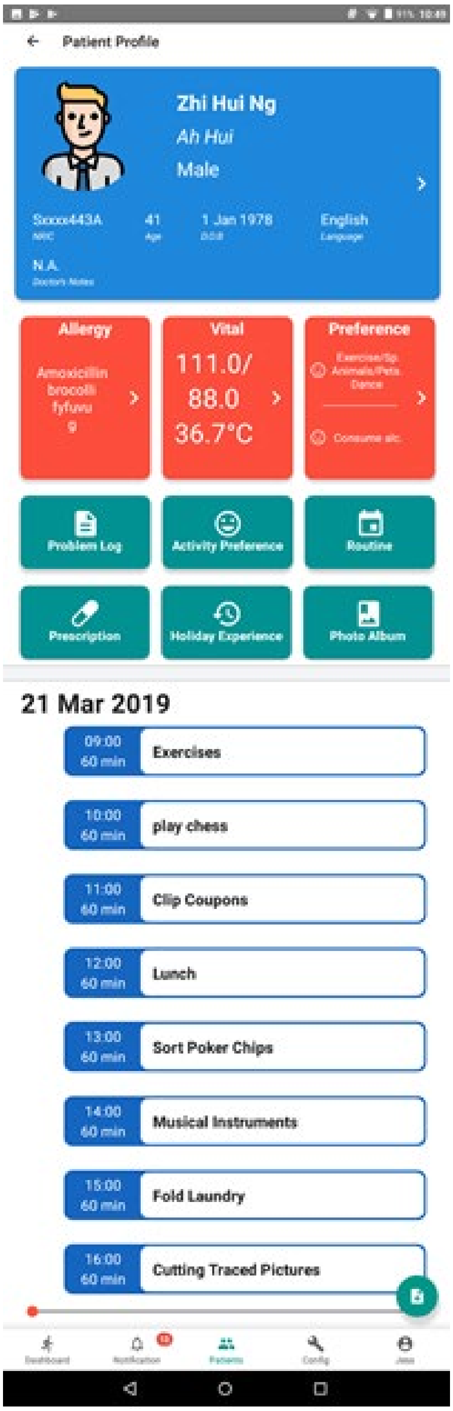
Zhi Hui Ng

Communication
Category

24 Mar 2019
Date

Add Problem Log
Problem Log

*Figure 26: Add problem log feature in Patient Profile Page*


Patient Profile

Zhi Hui Ng
Ah Hui
Male

Sxxxx443A
NRIC

41
Age

1 Jan 1978
DOB

English
Language

N.A.
Doctor's Notes

Allergy
Amoxicillin
brocolli
fyfuwu
g

Vital
111.0/
88.0
36.7°C

Preference
Exercise/Sp
Animals/Pets
Dance
Consume alc.

21 Mar 2019

| | |
|---|---|
| 09:00<br>60 min | Exercises |
| 10:00<br>60 min | play chess |
| 11:00<br>60 min | Clip Coupons |
| 12:00<br>60 min | Lunch |
| 13:00<br>60 min | Sort Poker Chips |
| 14:00<br>60 min | Musical Instruments |
| 15:00<br>60 min | Fold Laundry |
| 16:00<br>60 min | Cutting Traced Pictures |

*Figure 25: Entire new Patient Profile page*

## 6.8 Account

The Account page (Figure 27) is a new page dedicated for the user. It allows the user to know who is currently logged in to the application. From this page, the user can access the User Information page, Settings page, About page, or to log out of the application.

The first card is the brief user information card which contains the patient's first name, last name and role of the user logged in. The image was supposed to show profile photo of the user but currently it is just a placeholder because the current database does not support user profile photo.

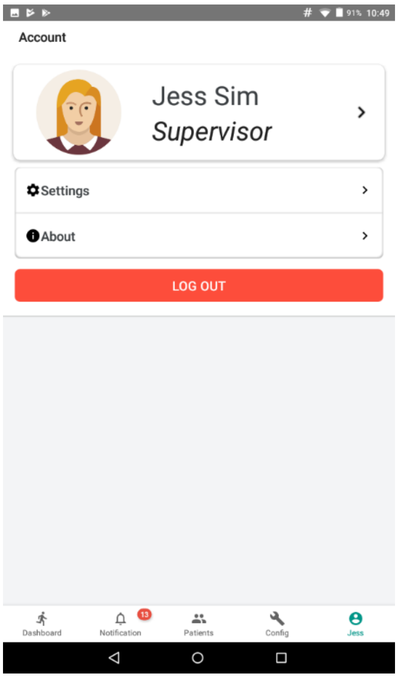


Figure 27: New Account Page

### 6.8.1 User information

To allow the user to view or edit his/her information, a new User Information page (Figure 28) was created.

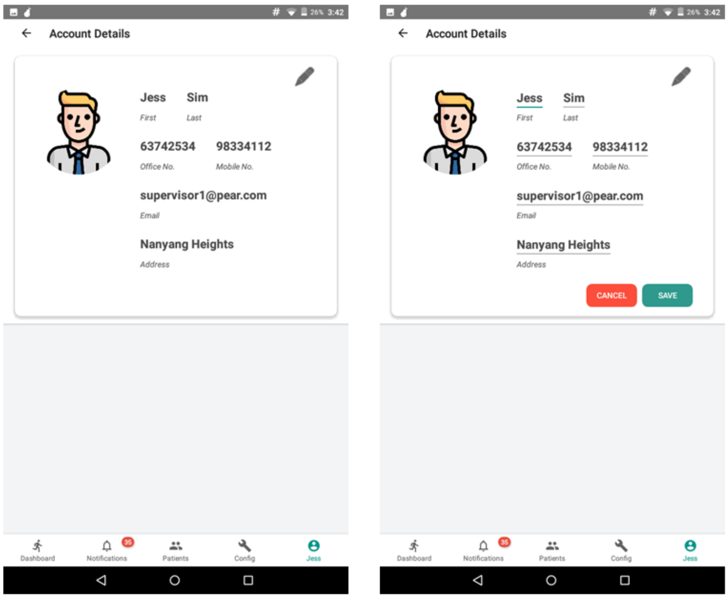
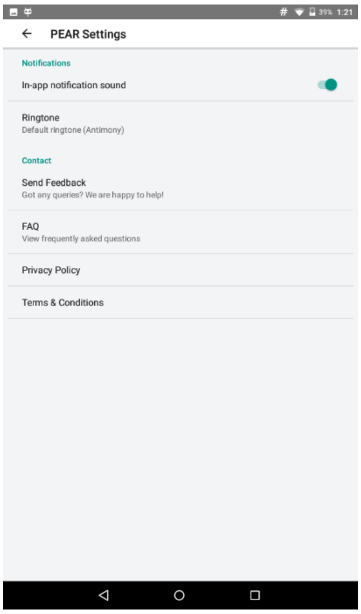


Figure 28: View vs Edit User Information

By selecting the edit button on the top right of the card (Figure 28), the user will be able to edit all his/her personal information. The fields available to edit are first name, last name, office number, mobile number, email, and address. The user will save the changes made by selecting on the save button.

### 6.8.2 Settings

The Settings page (Figure 29) was a new page to allow the user to change settings pertaining to the notification. The user can select a different ringtone when the application receives a new notification. The send feedback button when selected, it will open a menu to allow the user to choose email client so that he/she can write an email to the developer, attached with device information automatically.



PEAR Settings

Notifications

In-app notification sound ●

Ringtone
Default ringtone (Antimony)

Contact

Send Feedback
Got any queries? We are happy to help!

FAQ
View frequently asked questions

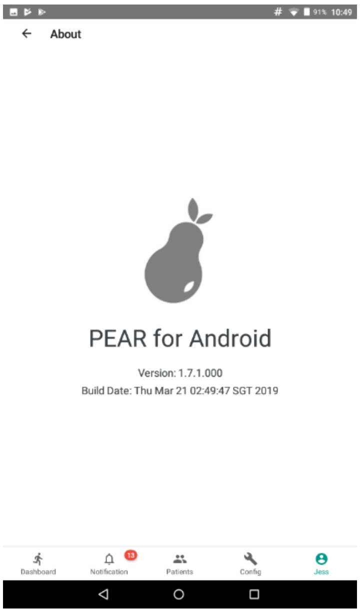
Privacy Policy

Terms & Conditions

*Figure 29: New Settings Page*

### 6.8.3 About

The About page (Figure 30) displays basic information of the application such as version number and build date. The about page had gone through a slight design change compared to the old. It now includes PEAR logo and application name.



PEAR for Android

Version: 1.7.1.000
Build Date: Thu Mar 21 02:49:47 SGT 2019

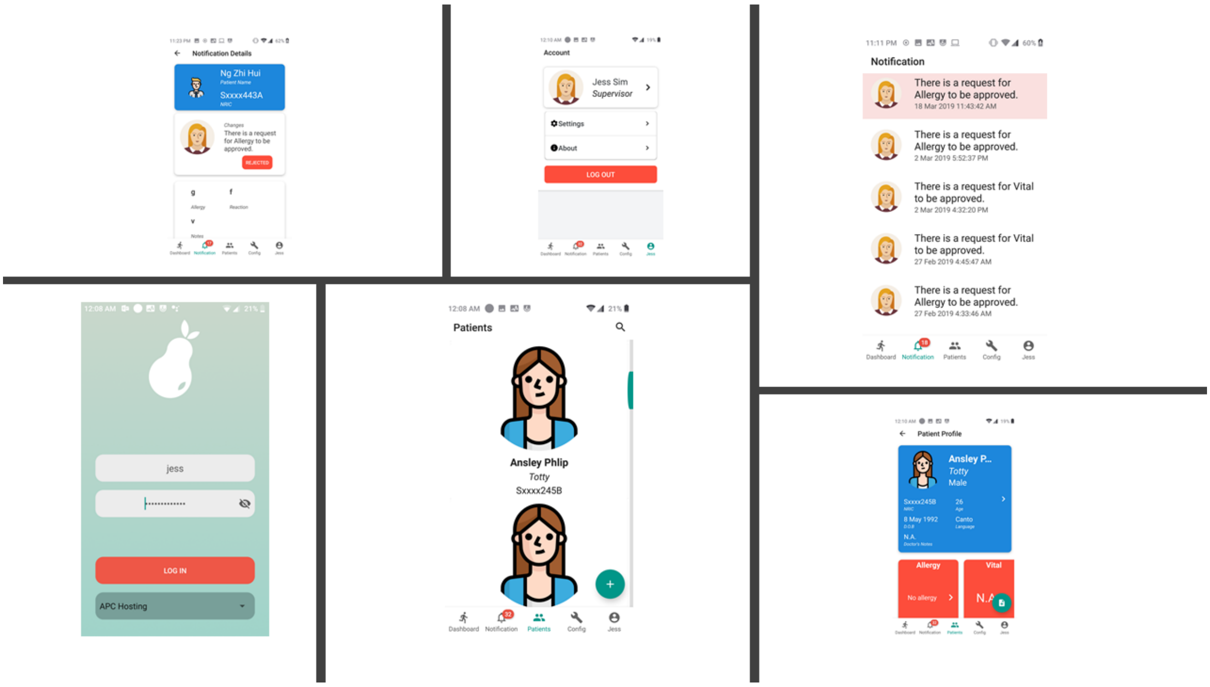
*Figure 30: Updated About Page*

### 6.8.4 Logout

Log out is a new feature to allow the user to log out of the application. After the user has logged out, the user will be brought to the login page. The implementation was to clear the SharedPreferences (created during login) and redirect the user to the login page.

## 6.9 Optimised for Phone User Interface

The application was originally intended to be used on tablet only. The old UI does not work well in a phone. However, considering that the user might want to check out information on the go or tablet is not available. Thus, the new interface was optimised to be used in phone mode (Figure 31). The implementation was done by editing the eXtensible Markup Language (XML) layout files. The changes made was then tested in an Android emulator in phone mode.



Notification Details

Ng Zhi Hui
Patient Name
Sxxxx443A
NRIC

Changes
There is a request for Allergy to be approved.

g
Allergy
f
Reaction
v
Notes

Account

Jess Sim
Supervisor

Notification

There is a request for Allergy to be approved.
18 Mar 2019 11:43:42 AM

There is a request for Allergy to be approved.
2 Mar 2019 5:52:37 PM

There is a request for Vital to be approved.
2 Mar 2019 4:32:20 PM

There is a request for Vital to be approved.
27 Feb 2019 4:45:47 AM

There is a request for Allergy to be approved.
27 Feb 2019 4:33:46 AM

jess

Patients

Ansley Phlip
Totty
Sxxxx245B

Patient Profile

Ansley P...
Totty
Male
Sxxxx245B
NRIC
26
Age
8 May 1992
D.O.B
Canto
Language
N.A.
Doctor's Notes
Allergy
No allergy
Vital
N.A

*Figure 31: Examples of Phone UI optimisation*

## 6.10 AsyncTask

AsyncTask class in Android allows background operations and later update the result in the UI thread. To improve the performance of the mobile application, AsyncTask was implemented in all the author's pages that require fetching of data from the server. The AsyncTask class was used to fetch data in the background while leaving the UI thread free to display loading spinner. Once data was completely retrieved, the loading spinner will then be hidden, and the page will display the required information. The use of AsyncTask had significantly improved the responsiveness of the application as UI thread is no longer held up to wait for completion of data retrieval.

## 6.11 Adaptive Icon

Beginning from Android 8.0, Android supports adaptive launcher icons. It allows displaying of application icon in a variety of shapes across different device models. The PEAR application icon was updated to support this feature (Figure 32).

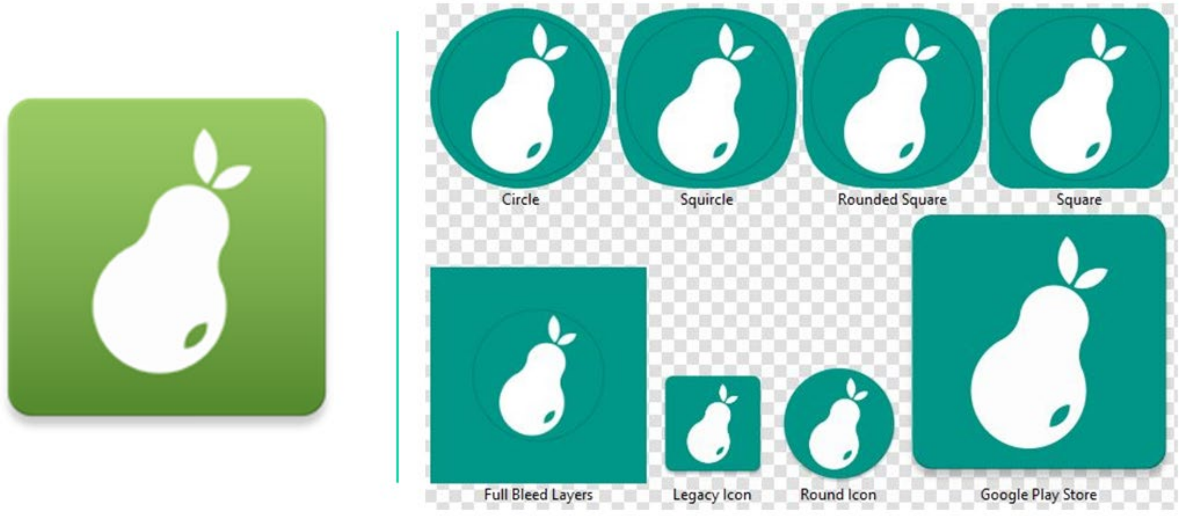
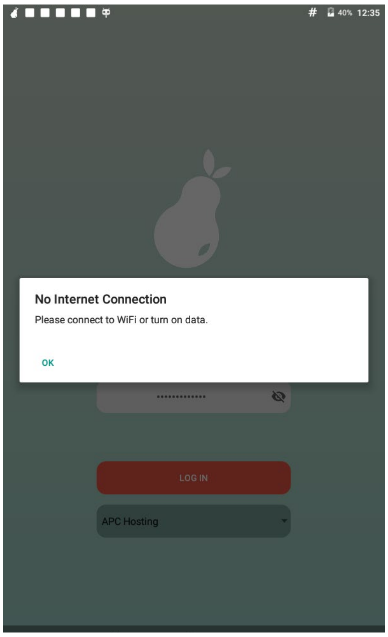


*Figure 32: Old vs New Application Icon*

## 6.12 Network checker

The application requires internet connection to work. Therefore, the network checker was implemented to remind the user to connect their WiFi or turn on their mobile data if they did not have a connection to the internet (Figure 33).



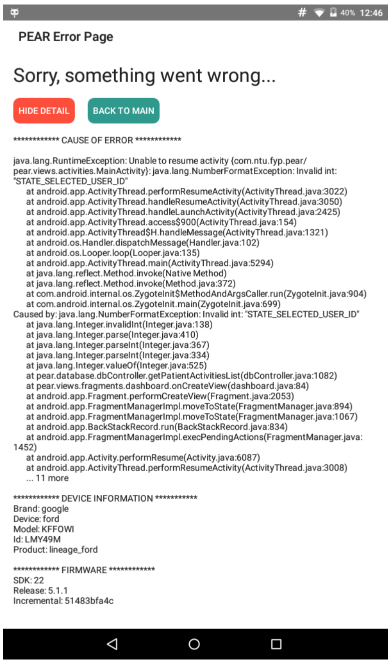
No Internet Connection

Please connect to WiFi or turn on data.

*Figure 33: Network Checker Reminder*

## 6.13 Error page / Uncaught Exception Handler

A global error page (Figure 34) was implemented in the application to handle any uncaught exception. Without this error page implementation, the application will only show the standard Android operating system error message, the application has stopped. From the error page, the user can view details of the error and after that maybe send the screenshot to the developer. The back to main button when selected allows the user to go back to the main page.



PEAR Error Page

Sorry, something went wrong...

```
************ CAUSE OF ERROR ************

java.lang.RuntimeException: Unable to resume activity {com.ntu.fyp.pear/
pear.views.activities.MainActivity}: java.lang.NumberFormatException: Invalid int:
"STATE_SELECTED_USER_ID"
    at android.app.ActivityThread.performResumeActivity(ActivityThread.java:3022)
    at android.app.ActivityThread.handleResumeActivity(ActivityThread.java:3050)
    at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2425)
    at android.app.ActivityThread.access$900(ActivityThread.java:154)
    at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1321)
    at android.os.Handler.dispatchMessage(Handler.java:102)
    at android.os.Looper.loop(Looper.java:135)
    at android.app.ActivityThread.main(ActivityThread.java:5294)
    at java.lang.reflect.Method.invoke(Native Method)
    at java.lang.reflect.Method.invoke(Method.java:372)
    at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:904)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:699)
Caused by: java.lang.NumberFormatException: Invalid int: "STATE_SELECTED_USER_ID"
    at java.lang.Integer.invalidInt(Integer.java:138)
    at java.lang.Integer.parse(Integer.java:410)
    at java.lang.Integer.parseInt(Integer.java:367)
    at java.lang.Integer.parseInt(Integer.java:334)
    at java.lang.Integer.valueOf(Integer.java:525)
    at pear.database.dbController.getPatientActivitiesList(dbController.java:1082)
    at pear.views.fragments.dashboard.onCreateView(dashboard.java:84)
    at android.app.Fragment.performCreateView(Fragment.java:2053)
    at android.app.FragmentManagerImpl.moveToState(FragmentManager.java:894)
    at android.app.FragmentManagerImpl.moveToState(FragmentManager.java:1067)
    at android.app.BackStackRecord.run(BackStackRecord.java:834)
    at android.app.FragmentManagerImpl.execPendingActions(FragmentManager.java:
1452)
    at android.app.Activity.performResume(Activity.java:6087)
    at android.app.ActivityThread.performResumeActivity(ActivityThread.java:3008)
    ... 11 more

************ DEVICE INFORMATION ***********
Brand: google
Device: ford
Model: KFFOWI
Id: LMY49M
Product: lineage_ford

************ FIRMWARE ************
SDK: 22
Release: 5.1.1
Incremental: 51483bfa4c
```

*Figure 34: New Error Page*

# 7. System Testing

It is important to ensure functionality works as intended after implementation thus the following tests were conducted for the respective pages:

## 7.1 Notifications

### 7.1.1 In-app Notifications

*Table 3: Test case and result of In-app Notifications page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the Notification page** | The application should show the notification page when notification destination is selected in the navigation bar | Pass |
| **Display user notifications** | The application should show a list of all user notifications | Pass |
| **Include message** | The list of notifications should include notification's message | Pass |
| **Include time** | The list of notifications should include notification's receiving time | Pass |
| **The list of notifications should be sorted** | The list of notifications should be sorted in order of date and time, latest at the top | Pass |
| **Unread Notification** | The unread notification should be highlighted in light orange. | Pass |

### 7.1.2 Push Notifications

*Table 4: Test case and result of Push Notification*

| Test case | Expected Result | Result |
|---|---|---|
| **Supervisor receive push notification** | The supervisor should receive a push notification when caregiver made a change on a Patient's Information | Pass |
| **Caregiver receive push notification on a change made** | The caregiver should receive a push notification when caregiver made a change on a Patient's Information | Pass |
| **Caregiver receive push notification on the supervisor's accept** | The caregiver should receive a notification when supervisor accept the request | Pass |
| **Caregiver receive push notification on the supervisor's reject** | The caregiver should receive a notification when the supervisor rejects the request | Pass |

## 7.2 Notification Details

*Table 5: Test case and result of Notification Details page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the Notification Details page** | The application should show the notification details page when the user selects on a notification in the notification page | Pass |
| **Patient brief information card** | The page should show Name and masked NRIC of the patient at the top | Pass |
| **Notification Message** | The page should show the notification message | Pass |
| **Show Accept and Reject Button** | The page should show the accept and reject button if the user is a supervisor, and the request is still pending | Pass |
| **Show disabled accept / reject button** | The page should show the disabled accept / reject button if the request had been accepted or rejected | Pass |
| **Display Status** | The page should show the request's status if the user is a caregiver | Pass |
| **Accept Button** | The supervisor should be able to select the accept button | Pass |
| **Reject Button** | The supervisor should be able to select the reject button | Pass |
| **Reject Reason** | The supervisor should be able to enter the reject reason if it's a reject | Pass |
| **Empty Reason** | The supervisor should not be able to reject without entering a reason | Pass |
| **Display reject reason** | The page should show the reject reason if the supervisor had rejected the request | Pass |
| **Changes made / new information (Allergy)** | The page should show the difference of the original information and changes made or new information which was just added | Pass |

## 7.3 Patients

*Table 6: Test case and result of Patients page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the Patients page** | The application should show the patients page when Patients destination is selected in the navigation bar | Pass |
| **The list of all patients** | The page should show a list of all patients in the care centre | Pass |
| **Order of the patients** | The patients should be sorted in alphabetical order | Pass |
| **The Information of patients** | The page should include information of the patients, name, preferred name and masked NRIC | Pass |

| | | |
|---|---|---|
| **Search** | The page should allow the user to do a search on the patients | Pass |
| **Alphabetical fast scroll** | The page should allow the user to do an alphabetical fast scroll by drugging on the scroll bar | Pass |
| **Selectable Patient** | The page should allow the user to select any of the patient | Pass |

# 7.4 Patient Profile

*Table 7: Test case and result of Patient Profile page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the Patients Profile page** | The application should show the patients profile page when the user selects a patient in the Patients page | Pass |
| **Patient Information Card** | The page should show the patient information card with full name, preferred name, gender, masked NRIC, age, date of birth, preferred language, and the doctor's note | Pass |
| **Allergy Card** | The page should show the allergy card with allergies of the patient | Pass |
| **Vital Card** | The page should show the Vital card with the most recent vital the patient took | Pass |
| **Preference Card** | The page should show the preference card with likes and dislikes of the patient | Pass |
| **Icon cards** | The page should show the six cards, problem log, activity preference, routine, prescription, holiday experience, and photo album with their icon | Pass |
| **Patient's schedule** | The page should show the patient's schedule of the day under all the cards | Pass |
| **Add problem log** | The page should allow the user to add problem log from the same page | Pass |
| **Selectable Cards** | All the cards on the page should be selectable | Pass |

# 7.5 Account

*Table 8: Test case and result of Account page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the Account page** | The application should show the account page when the user selects the Account destination in the navigation bar | Pass |
| **Brief User Information Card** | The page should show the brief user information card with the first name, last name and role of the user logged in | Pass |

| | | |
|---|---|---|
| **Selectable User Information, Settings and About card** | The brief user information, settings and about cards should be selectable | Pass |
| **Log out** | The page should allow the user to log out of the application when the user select on the log out button | Pass |

## 7.6 User Information

*Table 9: Test case and result of user information page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the User Information page** | The application should show the user information page when the user selects the brief user information card in the Account page | Pass |
| **Viewing user information** | The page should show first name, last name, office number, mobile number, email, and address of the user | Pass |
| **Edit user information** | The page should allow the user to edit and save their information | Pass |
| **Empty input validation** | The page should not allow the user to save empty input, an error message should show if it is empty | Pass |
| **Phone number validation** | The page should not allow the user to save phone number with incorrect format, an error message should show if it is incorrect | Pass |
| **Email validation** | The page should not allow the user to save email with incorrect format, an error message should show if it is incorrect | Pass |

## 7.7 Settings

*Table 10: Test case and result of Settings page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the Settings page** | The application should show the settings page when the user selects the settings card in the Account page | Pass |
| **Turn off in-app notification ringtone** | The page should allow the user to turn off in-app notification ringtone | Pass |
| **Change in-app notification ringtone** | The page should allow the user to changed in-app notification ringtone | Pass |
| **Send feedback** | The page should open the email client chooser menu when the user selects send feedback | Pass |

# 7.8 About

*Table 11: Test case and result of About page*

| Test case | Expected Result | Result |
|---|---|---|
| **Launching of the About page** | The application should show the about page when the user selects the about card in the Account page | Pass |
| **PEAR Logo** | The page should show the PEAR Logo | Pass |
| **Version number** | The page should show the application version number | Pass |
| **Build date** | The page should show the application build date | Pass |

# 8. Conclusion

## 8.1 Summary

In the project, many changes were made to improve the android application. The User Interface (UI) of the application was revamped. The new UI aims to provide an updated and modern UI that is easy to use. The author had implemented a total of 8 pages based on the new User Interface design. The performance was improved by only fetching the necessary information for each page via the RESTful API. The application's responsiveness was also improved with the use of AsyncTask class. With all the improvements made, the author fulfilled the objective of the project to improve the user experience of the user of the application.

## 8.2 Future Improvements

There is always room for improvement for the Android application. Following are ideas suggested by the author:

### 8.2.1 Continuous optimisation for phone UI

The phone UI was only implemented for a few pages, but there are many more pages did not go through the phone UI optimisation treatment. To be fully functional in phone, layout files of many more pages are required to tweak to display them correctly in phone.

### 8.2.2 Advanced Settings

The application currently only supports very basic settings. Thus, it recommended that the application can have more advanced settings such as turning off push notification completely.

# References

[1] "Dementia Care Practice Recommendations | Alzheimer's Association," Alzheimer's Association, 2018. [Online]. Available: https://www.alz.org/professionals/professional-providers/dementia_care_practice_recommendations. [Accessed 6 September 2018].

[2] J. Tai, "One in 10 people over 60 have dementia, new Singapore study claims," The Straits Times, 25 March 2015. [Online]. Available: https://www.straitstimes.com/singapore/health/one-in-10-people-over-60-have-dementia-new-singapore-study-claims. [Accessed 6 September 2018].

[3] "Dementia on the Rise in Singapore," Temasek Trust, 2015. [Online]. Available: http://www.temasektrust.org.sg/newsletter-item/dementia-on-the-rise-in-singapore/. [Accessed 6 September 2018].

[4] Sam Fazio, Douglas Pace, Janice Flinner, Beth Kallmyer, "The Fundamentals of Person-Centered Care for Individuals With Dementia," The Gerontological Society of America, 18 January 2018. [Online]. Available: https://academic.oup.com/gerontologist/article/58/suppl_1/S10/4816735. [Accessed 6 September 2018].

[5] "Dementia - Symptoms and causes - Mayo Clinic," Mayo Foundation for Medical Education and Research (MFMER), [Online]. Available: https://www.mayoclinic.org/diseases-conditions/dementia/symptoms-causes/syc-20352013. [Accessed 23 March 2019].

[6] "Dementia | Symptoms, Diagnosis, Causes, Treatments," Alzheimer's Association, [Online]. Available: https://www.alz.org/alzheimers-dementia/what-is-dementia. [Accessed 23 March 2019].

[7] "Person-centred care | Alzheimer's Society," Alzheimer's Society, [Online]. Available: https://www.alzheimers.org.uk/about-dementia/treatments/person-centred-care. [Accessed 23 March 2019].

[8] T. Z. R. Ashley, "Final Year Project Report - Person-centred Care for Dementia Patients (Front End Android App)," Singapore, 2018.

[9] "Android 9 Pie," Google LLC, [Online]. Available: https://www.android.com/versions/pie-9-0/. [Accessed 23 March 2019].

[10] Anthony, "Why Rounded Corners Are Easier on the Eyes," 17 August 2011. [Online]. Available: https://uxmovement.com/thinking/why-rounded-corners-are-easier-on-the-eyes/. [Accessed 23 March 2019].