**Xinyi Liu - 300307421**

**Shankarenfo Pannagiani Dharma - 300304800**

**Bayron Palacio - 300307807**

PROJECT REPORT DocToGo APP

2020

Table of Contents

[1 INTRODUCTION 2](#_Toc37213520)

[2 USE CASE DIAGRAM AND DATABASE SCHEMA 2](#_Toc37213521)

[3 LOGIN SCREEN -Shankarenfo Pannagiani Dharma 5](#_Toc37213522)

[3.1 Registration -Shankarenfo Pannagiani Dharma 5](#_Toc37213523)

[3.2 Admin Menu -Shankarenfo Pannagiani Dharma 7](#_Toc37213524)

[3.3 Admin Register -Shankarenfo Pannagiani Dharma 7](#_Toc37213525)

[3.4 Admin View all -Shankarenfo Pannagiani Dharma 8](#_Toc37213526)

[3.5 Admin View Specific -Shankarenfo Pannagiani Dharma 8](#_Toc37213527)

[4 PATIENT USER 9](#_Toc37213528)

[4.1 Main Menu - Bayron Palacio 11](#_Toc37213529)

[4.2 Locate Doctor - Bayron Palacio/Xinyi 12](#_Toc37213530)

[4.3 Check Appointment -Xinyi Liu 12](#_Toc37213531)

[4.4 Payments -Xinyi Liu 13](#_Toc37213532)

[4.5 Check History -Xinyi Liu 14](#_Toc37213533)

[4.6 Update Personal Information - Bayron Palacio 15](#_Toc37213534)

[5 DOCTOR USER 16](#_Toc37213535)

[5.1 Appointments -Xinyi Liu 16](#_Toc37213536)

[5.2 Message -Xinyi Liu 17](#_Toc37213537)

[5.3 Report -Xinyi Liu 17](#_Toc37213538)

[6 CASHIER USER 18](#_Toc37213539)

[6.1 New Transaction -Xinyi Liu 19](#_Toc37213540)

[6.2 View Pending Transactions -Xinyi Liu 20](#_Toc37213541)

[6.3 Transaction Archive -Xinyi Liu 22](#_Toc37213542)

# INTRODUCTION

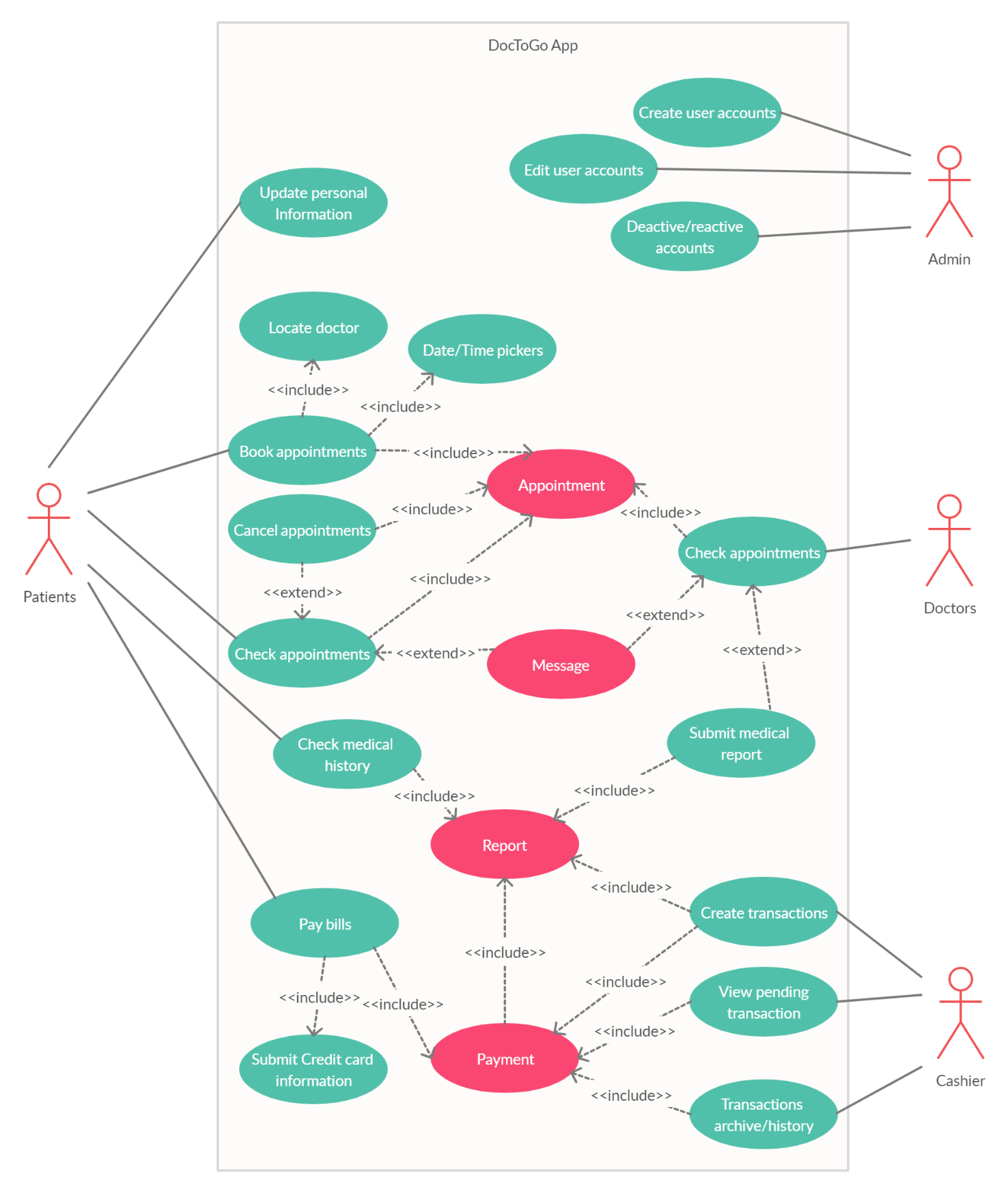
The DocToGo Application was designed and developed on Android Studio and supported for Android OS. The application supports four different types of accounts: Administrator, Patient, Doctor and Cashier that will be explained starting on section two of this document. The user will need to create an account with basic information.

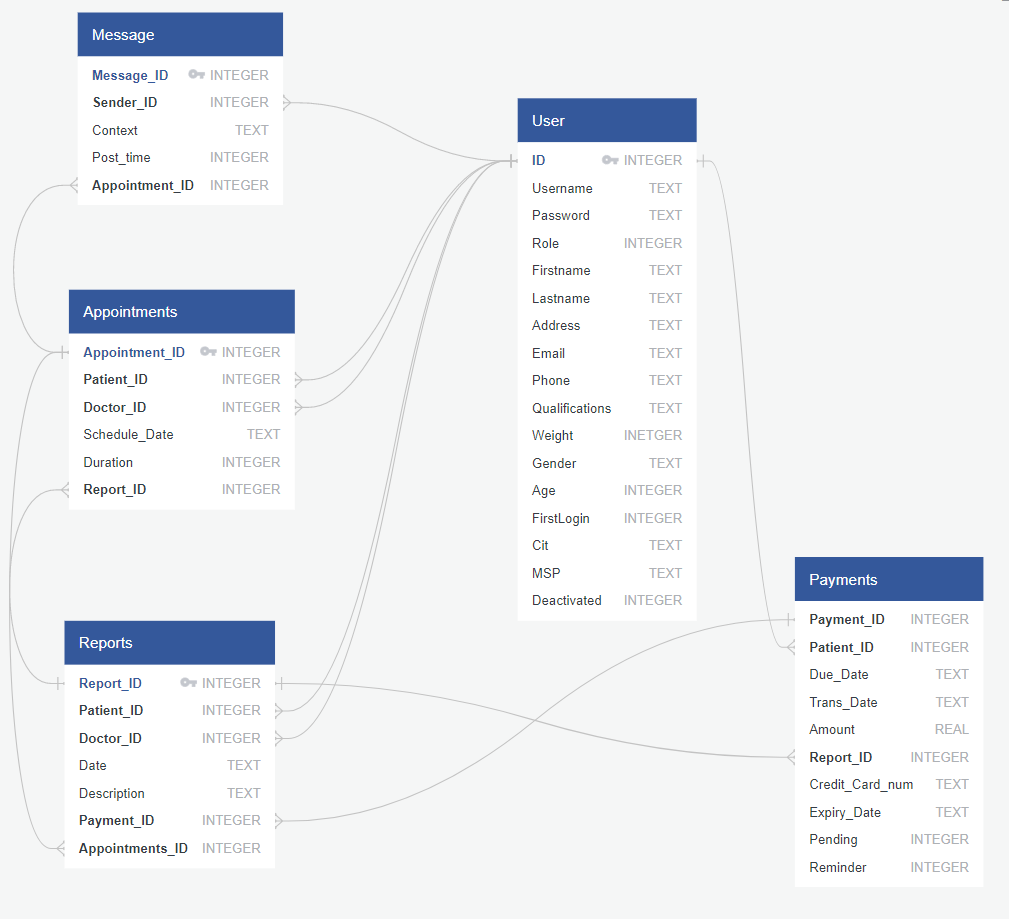
Patients will be able to choose a doctor from a list of the closest doctors according to their city. After choosing the doctor, the patient could book appointments, check doctor information, office details and get online help. Also, patients will be able to view their history, edit their personal information and check the total amount due.

Doctor will see the appointments that been booked by the patients, create report for each appointment, reply messages to the patients and check the report history.

The DocToGo application was developed using Android Studio and the participants of this project used GitHub to create and share their code. GitHub helped us to control, manage and create the final project with the contribution of each participant.

# USE CASE DIAGRAM AND DATABASE SCHEMA



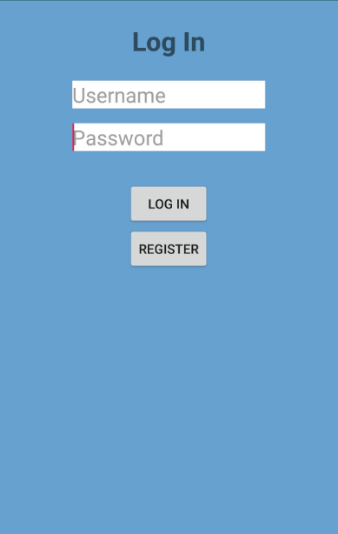


# LOGIN SCREEN -Shankarenfo Pannagiani Dharma

Activity\_main.xml MainActivity.java

The login screen is a simple interface with two input text boxes and two buttons. This screen is named main\_activity since it will be the first screen the user encounters on starting the application.

When the ‘login’ button is pressed, the application will consult the database to check if a matching account is found- the username and password is case-sensitive and must be an exact match. Natively, the database has one hardcoded account: ‘root’ ‘pass’ account, for testing.



Then, it will check the role associated with the account, and send the user to their respective interface. In total, there are four main interfaces available for the user to be sent to. At the same time, the login information is temporarily stored in the cache, so the application can quickly load it from other activities.

The other button present in the interface is the ‘register’ button. This will lead to a new activity where a user can then add themselves into the system database.

## Registration -Shankarenfo Pannagiani Dharma

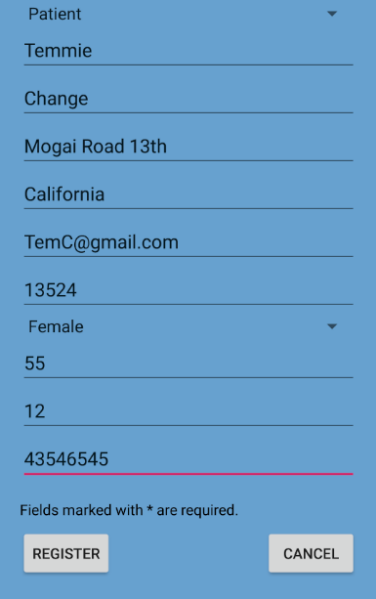
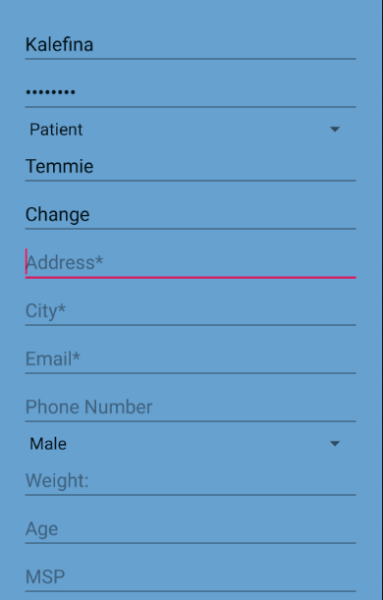
Activity\_register.xml register.java

The register activity allows guest and new users to create their own account in which they can then log in as and use the application. The interface is scrollable, so if any of the components go over the screen, drag the screen to view the rest of the components.

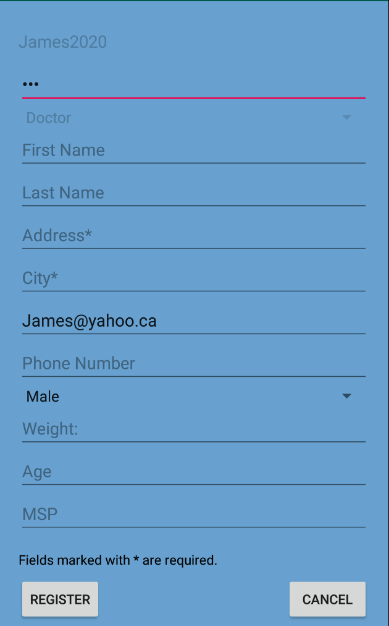
Currently, they can pick only ‘patient’ or ‘doctor’ roles. Admin and Cashier accounts can be created through Admin account functions. Fields marked with ‘\*’ are essential for the database and cannot be left empty. However, some fields can be left empty if it is optional, private or not required, but is useful to have.

If all fields are valid, the application will then insert a new account with the data entered in the fields. The user will then be returned to the login screen, and they can then use their new account credentials to enter the application.

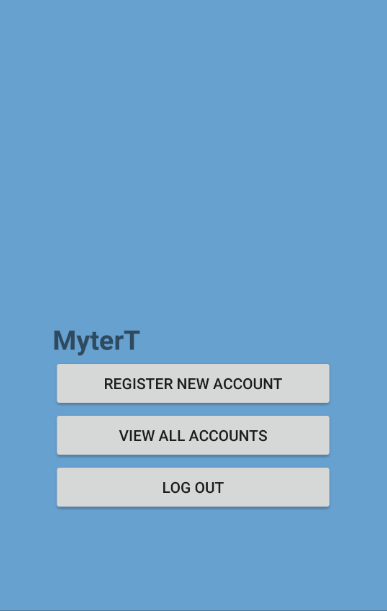
If fields are invalid, a message will show.



If the account was made by admin, and user first time login to this account. The registration page will change to first time login mode. Username edit text and role spinner will be unchangeable.

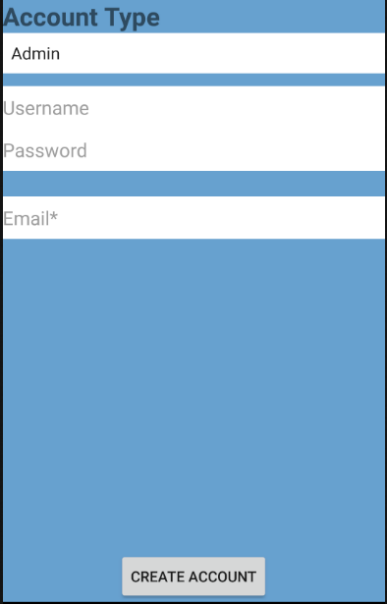


## Admin Menu -Shankarenfo Pannagiani Dharma Activity\_admin\_main.xml admin\_main.java



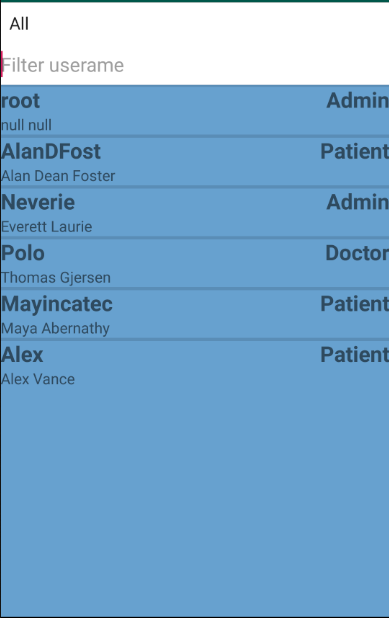
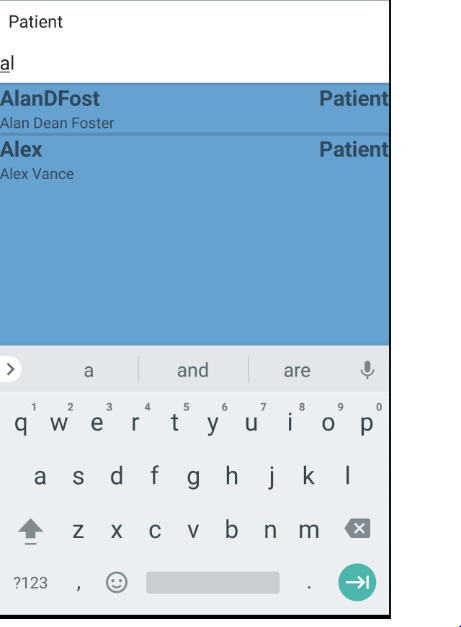
The admin menu is a simple selection menu that will send the admin user either to user registration or view accounts, each with their own specific activity.

## Admin Register -Shankarenfo Pannagiani Dharma activity\_admin\_register.xml admin\_register.java



This activity allows the admin to create an account for somebody else. In this method, the account has a specific flag attached in the database that necessitates a ‘first login’ function- when that user logs in, they will have to input their information and password for themselves.

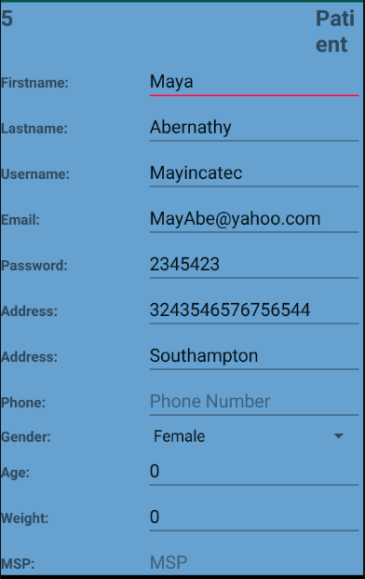
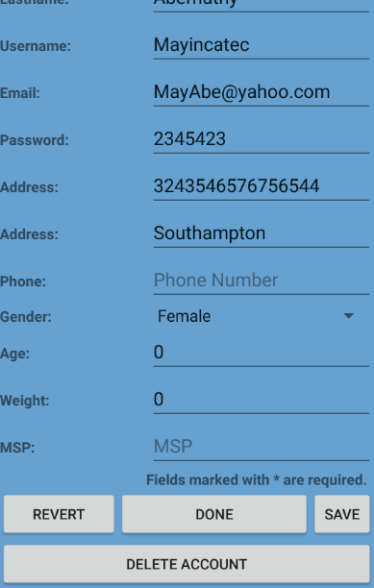
## Admin View all -Shankarenfo Pannagiani Dharma activity\_admin\_\_account\_details.xml admin\_accounts\_view.java

The view all activity displays a list of all users- as well as filters for username and role at the top. The application first gets the filters that are applied in the filters (using IS NOT NULL clause if there are empty filters or ALL filter). List uses a custom adapter and layout.

The admin can then select one of the list items displayed, displaying the details of those accounts.

## Admin View Specific -Shankarenfo Pannagiani Dharma activity\_admin\_\_account\_details.xml admin\_account\_details.java

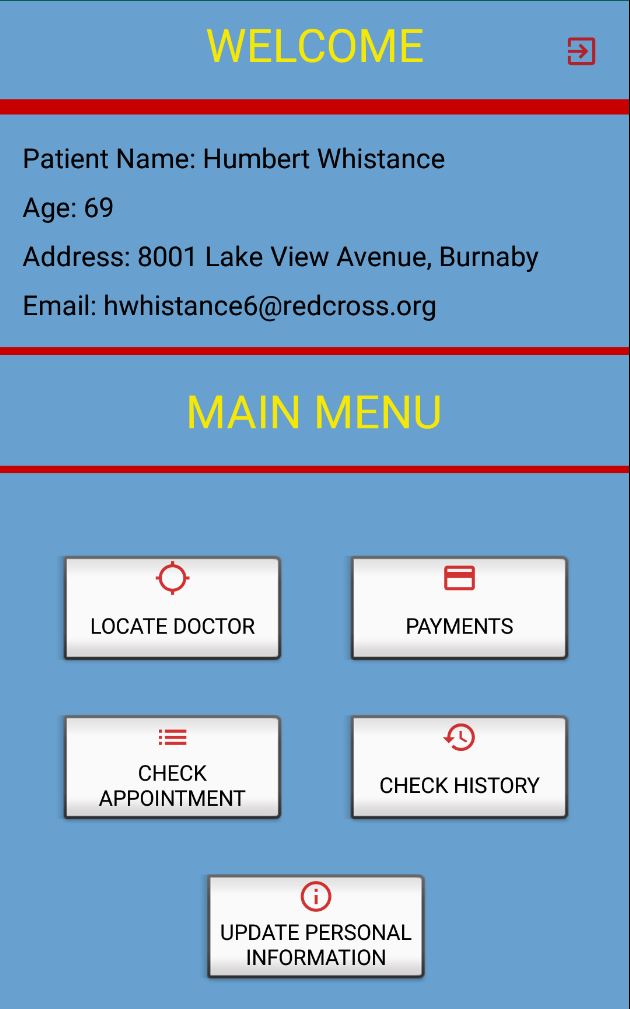
This activity, accessed from the listview above, displays all information of the user in textboxes. The activity gathers info from the database and pastes them into the interface, if any are available. The admin can then make changes to these fields.

However, the changes are not saved until ‘Save’ button is pressed. This button will get the information supplied in the activity and commits the update on the database. If the button is not pressed when the user quits the activity or closes the app, the changes are not saved.

Accompanying these buttons are ‘revert’ and ‘done’ buttons. ‘revert’ button will undo the changes- it will re-retrieve account info from the database and displays them again, overwriting any changes made. The other button simply returns the admin to the listview previous.

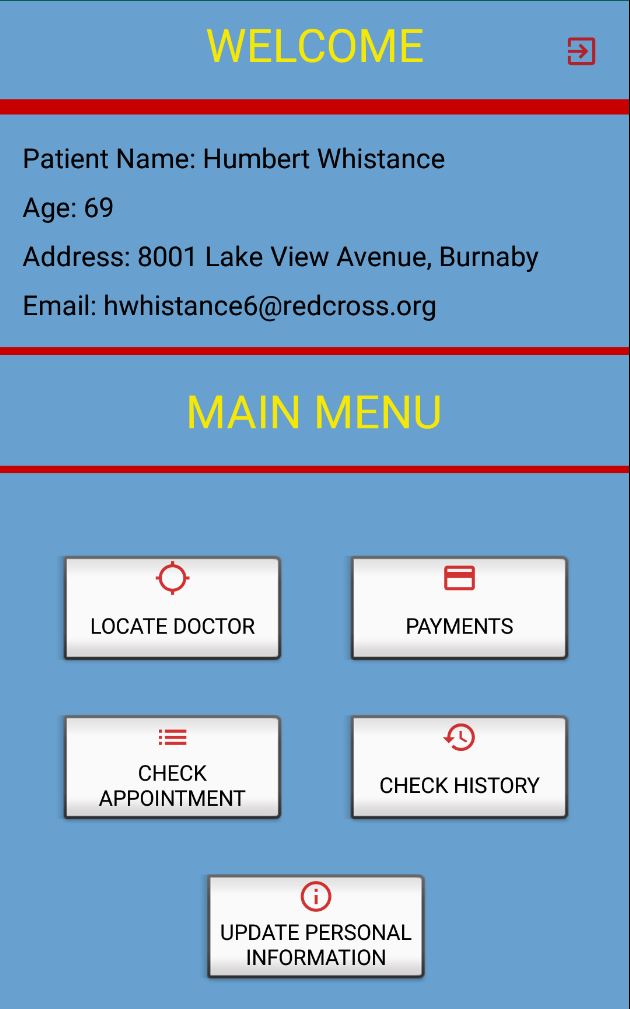
# PATIENT USER

After the patient logs in the application, the Main Patient Activity will be shown. The screen of the patient user will be split in two areas, Information About User and Information about the specific Activity.



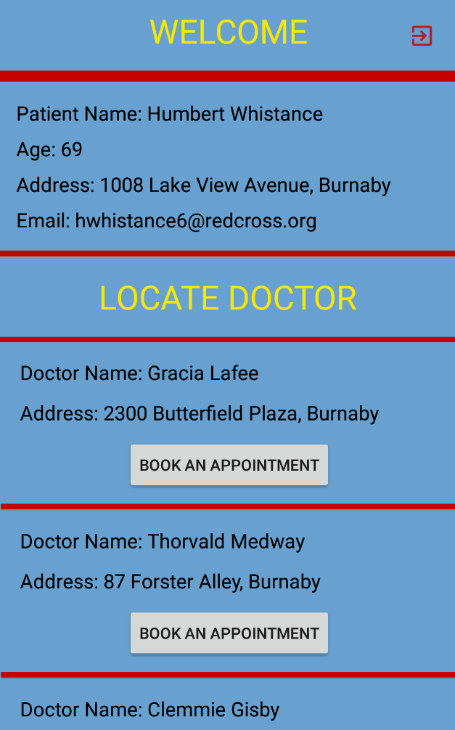
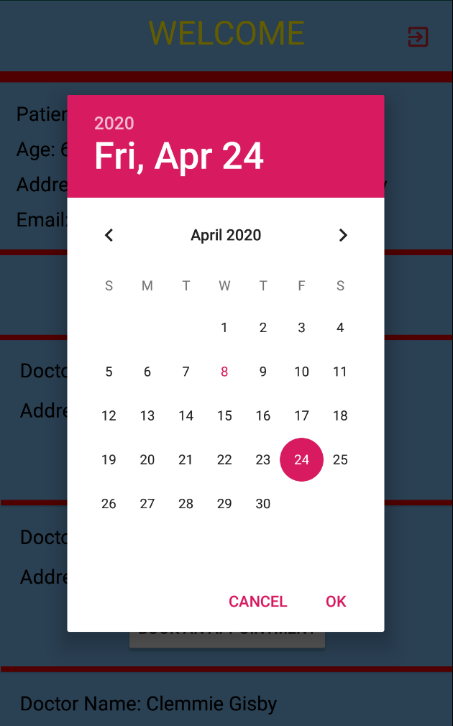
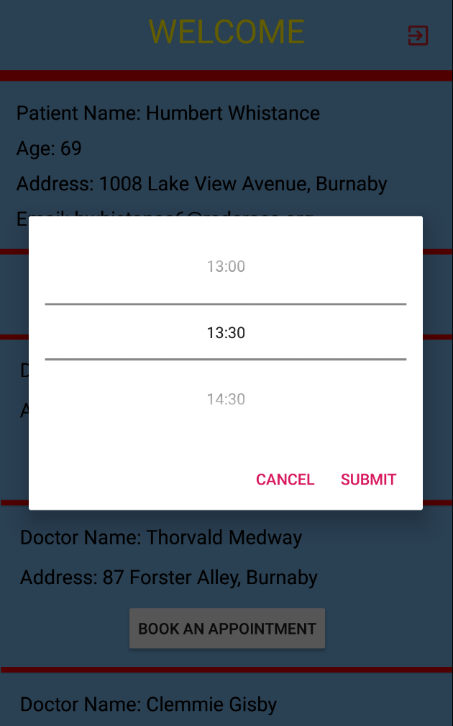
The Information about user will be shown on all Patient activities and it shows the patient Name, Age, Address and Email. Also, the area contains the exit icon . The navigation icon will show all the activities that the patient can enter, and the exit icon will log out the user from the application.

## Main Menu - Bayron Palacio activity\_patient\_main.xml patient\_main.java fragment\_patient\_information.xml patientInformationFragment.java



After the patient logs in into the application, the Main Menu screen will be shown. In this activity, the user will be able to select from five options: Check Appointment, Payments, Locate Doctor and Update Personal Information. Each activity will be described as follows:

## Locate Doctor - Bayron Palacio/Xinyi activity\_patient\_locate\_doctor.xml patient\_locate\_doctor.java LocateDoctorAdapter.java

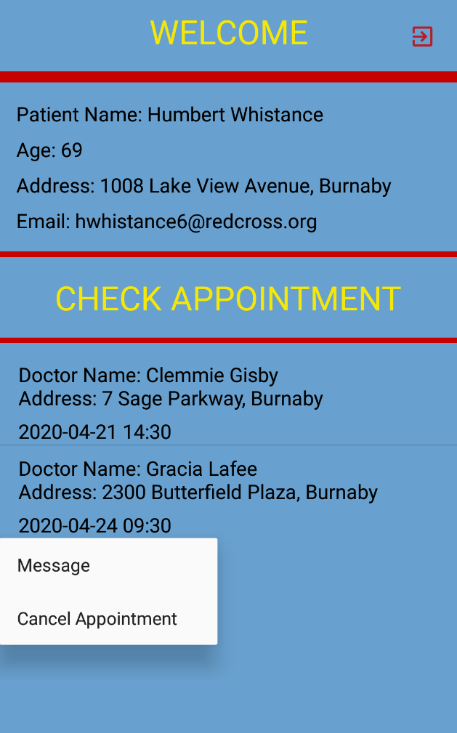
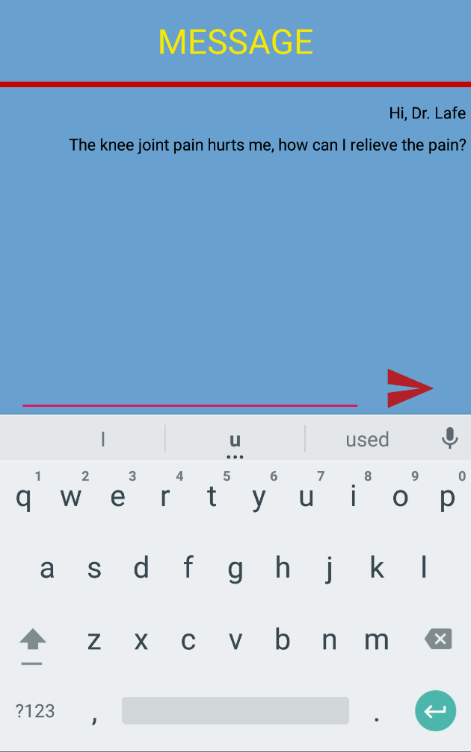
  

In the Locate Doctor activity, the patient will be able to see a scrollable list of doctors that are in the same area. City was taken as search key in order to find all the doctors that are in the same city as the patient. The patient can select one of the doctors and book an appointment. When the Book an Appointment button is clicked, a calendar will pop – up to choose a specific date for the appointment. After the date is chosen, a clock will pop – up with the hours that the doctor is available. The time is from 8 am to 4:30 pm. Also, if one of the hours has been chosen by another patient, this time will not be shown to the other users. The picture above shows that the a 14:00 - 14:29 appointment has been made by another patient.

## Check Appointment -Xinyi Liu

activity\_patient\_check\_appointment.xml patient\_check\_appointment.java   
activity\_message.xml single\_message.xml message.java singleMessage.java MessageAdapter.java

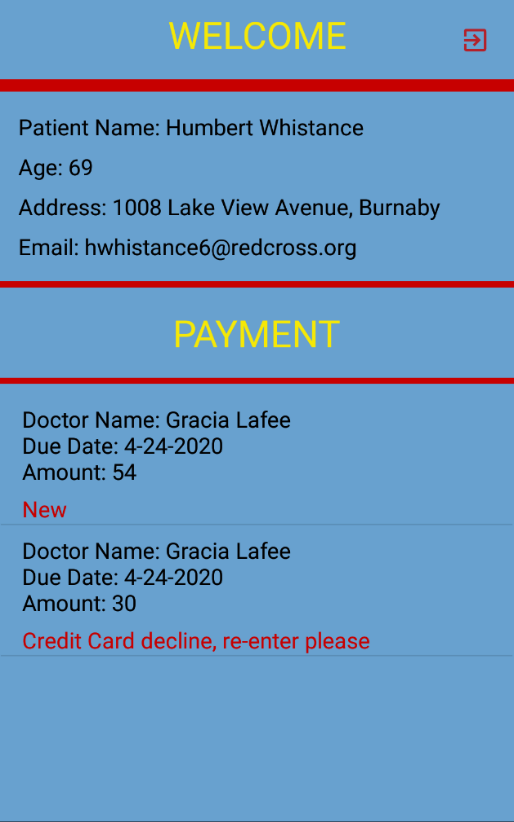
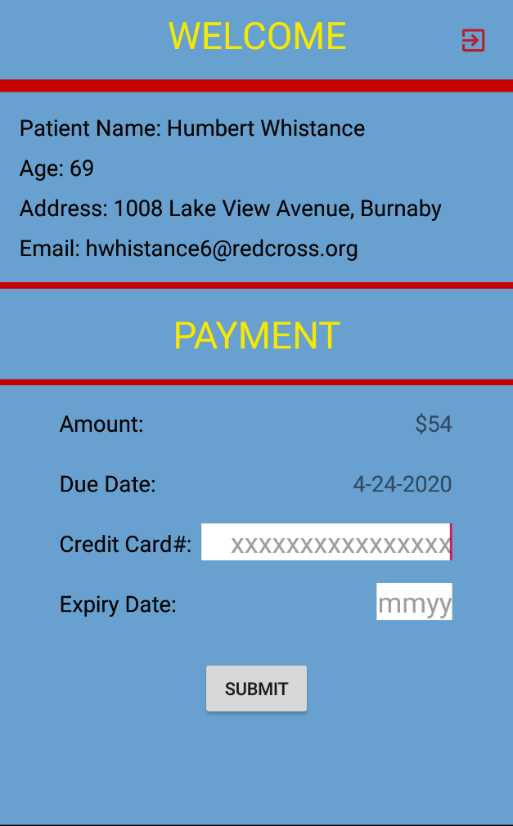
In the Check Appointment activity, the patient will be able to see the appointments that have been booked in a scrollable list. There, the patient can click on each appointment and a pop-menu will be displayed with two options Cancel Appointment and Message.

When the patient clicks on Cancel Appointment, the appointment will be canceled and removed from the scrollable list on the Check Appointment Activity. Also, the time that was scheduled this appointment, it will be available again for all the patients.

When the patient clicks on the Message option, another screen will be show. The patient can send a message to the doctor. The messages that were sent by the patient will be shown in the doctor activities which will be explained in the doctor session.

## Payments -Xinyi Liu activity\_patient\_payment.xml patient\_payment.java activity\_patient\_credit\_card.xml patient\_creditCard.java

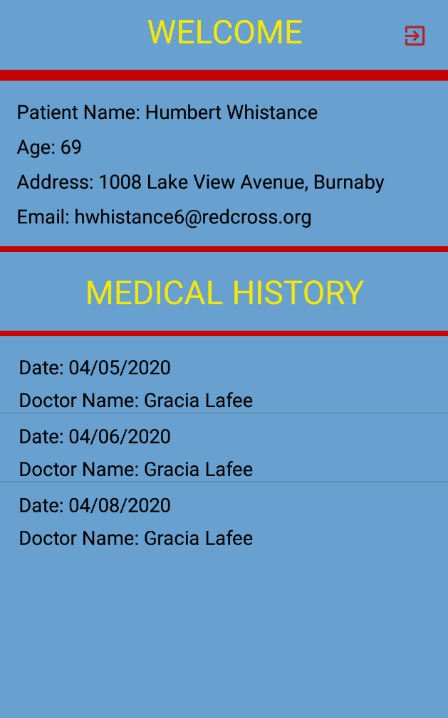
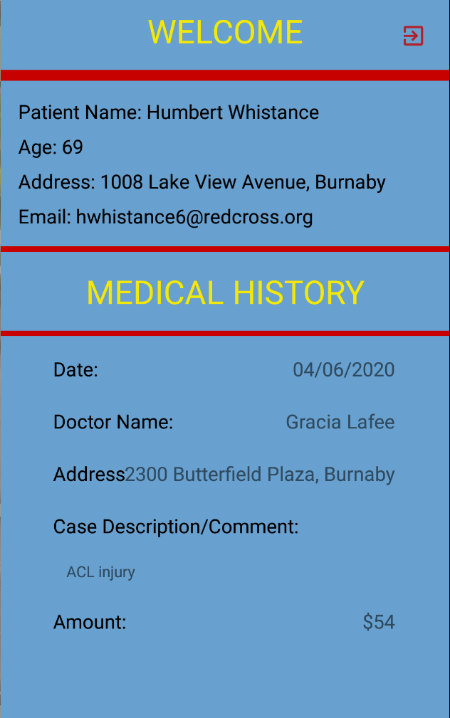
 

Patients can read the bills on the payment page and click to enter their credit card info to submit the payment. After submitting credit card info, the tab will show text “processing” and become unclickable.

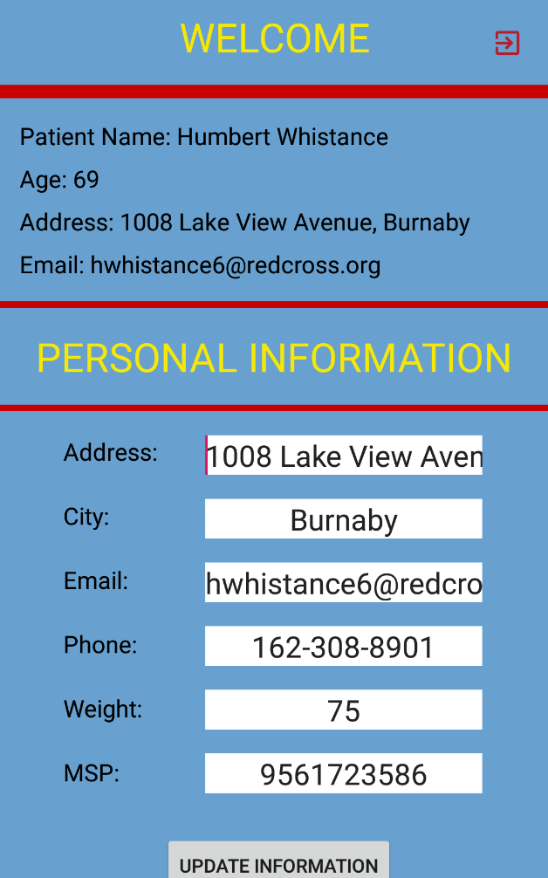
## Check History -Xinyi Liu

activity\_patient\_check\_history.xml patient\_check\_history.java   
activity\_patient\_check\_history\_details.xml patient\_check\_history\_details.java

When the patient goes to the Medical History Activity, a list of reports will be show for the appointments that the patients has attended. For each item of the list, the user can click on it to see specific information such as date, doctor name, Description and amount of the appointment.

## Update Personal Information - Bayron Palacio activity\_patient\_update\_information.xml patientUpdateInformation.java



In this activity, the patient will be available to update his/her personal information such as Address, City, Email, Phone, Weight or MSP. After the information has been entered, the patient needs to click on Update Information to execute the task. The new information will be updated right away in the User Information Area.

# DOCTOR USER

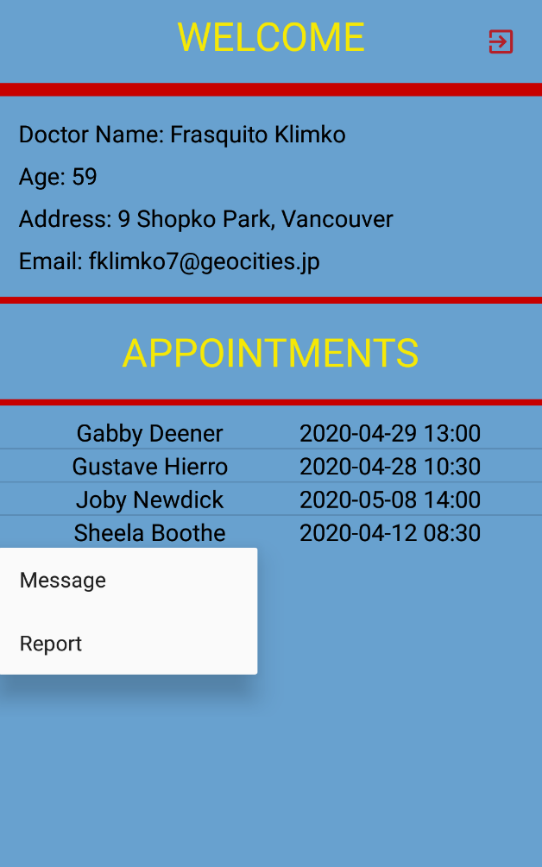
When a doctor logs in the application, the main activity will be shown. The screen is divided in two areas: information about user and specific information about activity, same as patient user.

In the information doctor area, it will be displayed the doctor name, age, address and email.

## Appointments -Xinyi Liu

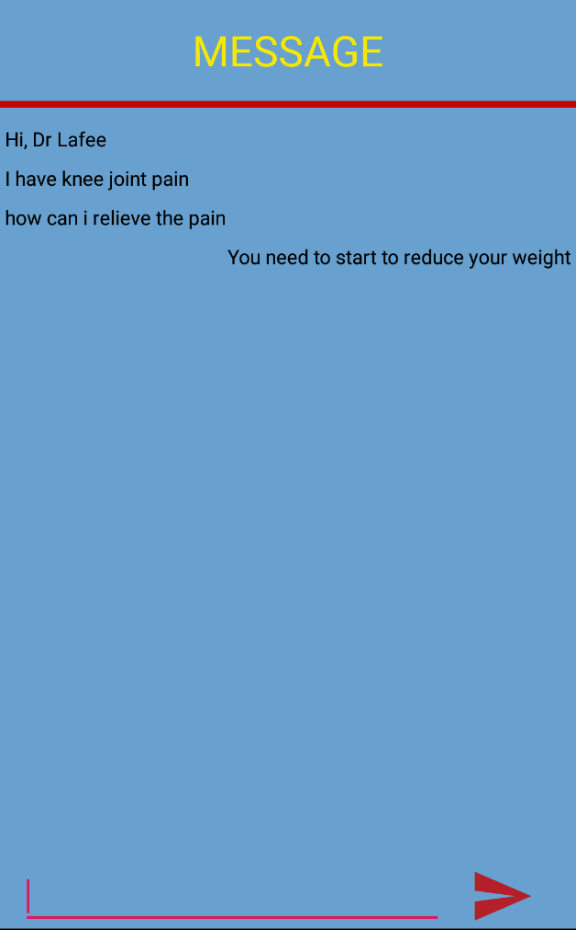
activity\_doctor\_main.xml doctor\_main.java   
fragment\_doctor\_appoint.xml DoctorAppointFragment.java

Appointments is the main activity of the doctor user. This activity will be shown the appointments that have been booked for the doctor that has login. It will be displayed in a list form with the patient name and appointment date.

The doctor can click on each patient to reply a message that was sent by the patient or create the patient report.

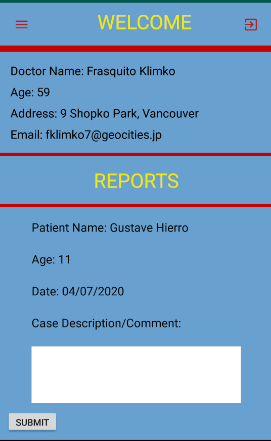
## Message -Xinyi Liu activity\_message.xml single\_message.xml message.java singleMessage.java MessageAdapter.java



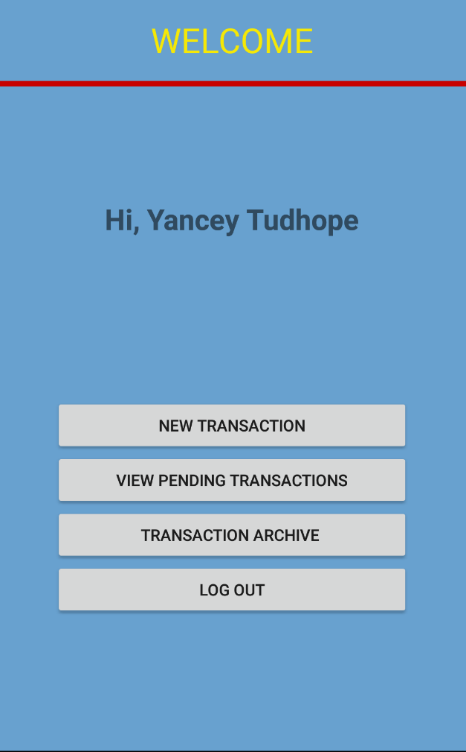
When the doctor clicks on Message, another screen will be shown with the message that has been sent by the patient and the doctor can reply to the patient. The message can be sent back and forward until the report is created. After the report is created, the conversation is closed and both parties, patient and doctor, will not have access to this conversation anymore.

## Report -Xinyi Liu activity\_doctor\_report.xml doctor\_report.java

Doctor writes down the comments and sends the report to the cashier.



# CASHIER USER



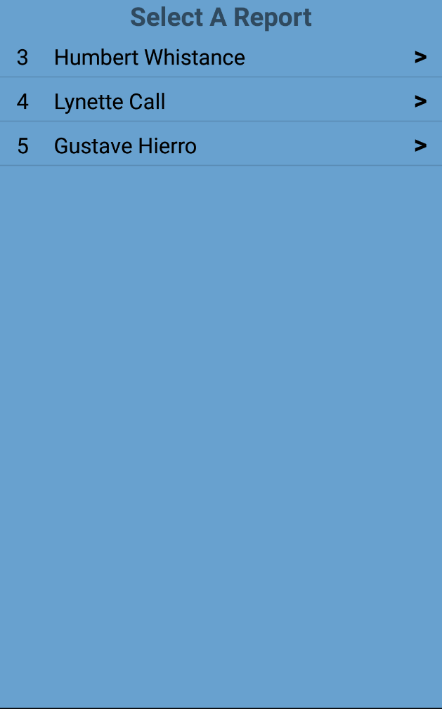
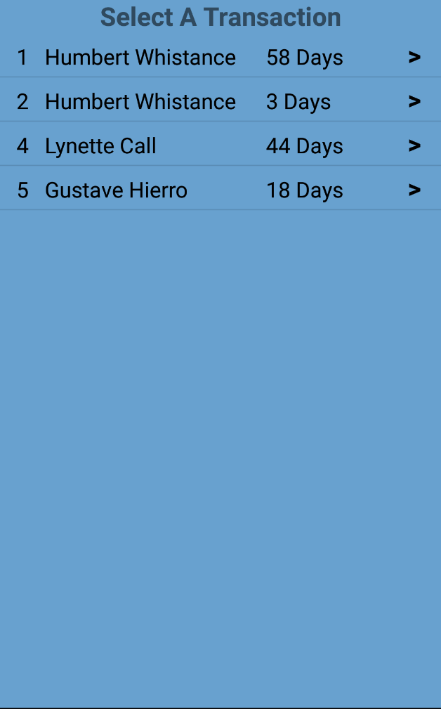
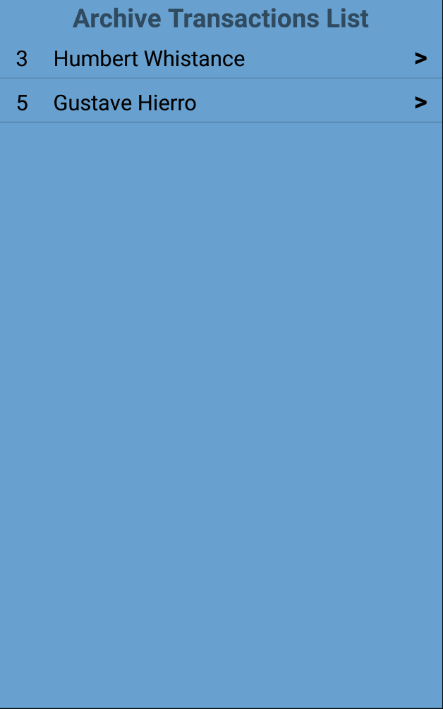
activity\_cashier\_main.xml cashier\_main.java

After cashiers log in to their accounts, they will redirect to the welcome page.

The “New Transaction” button will guide cashiers to a list of new report from doctors, require cashiers to set amount and due date, then set the bill to patient. (Select A report)

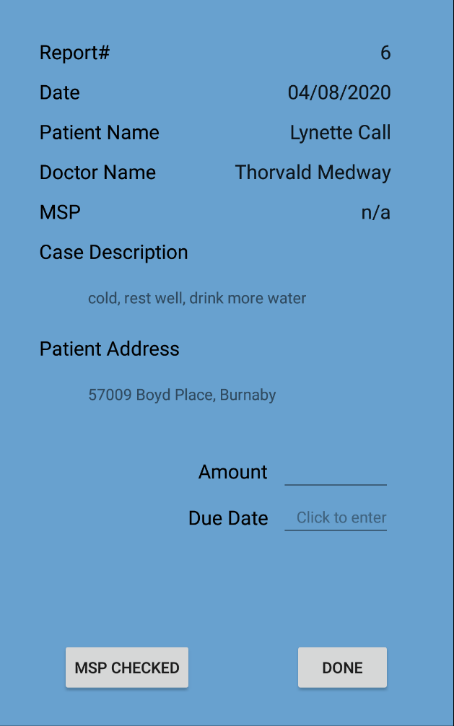
The “View Pending Transaction” button will guide cashiers to a list of the bills already send to patient with the due. They can edit those pending transaction through here. (Select A Transaction)

The “Transaction Archive” button will guide cashier to a list of closed transactions. (Archive Transactions List)

activity\_cashier\_list\_new.xml cashier\_list\_new.java  
activity\_cashier\_list\_view.xml cashier\_list\_view.java  
activity\_cashier\_list\_archive.xml cashier\_list\_archive.java

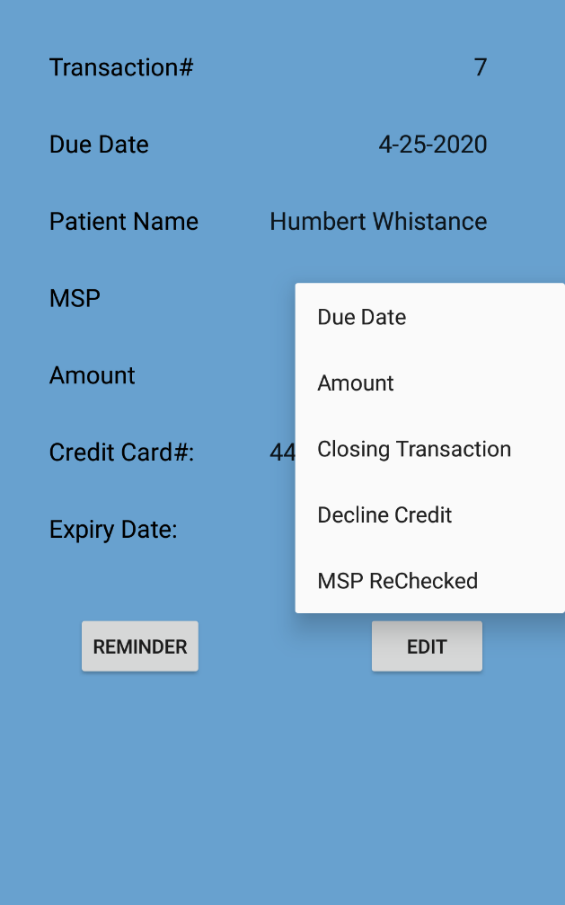
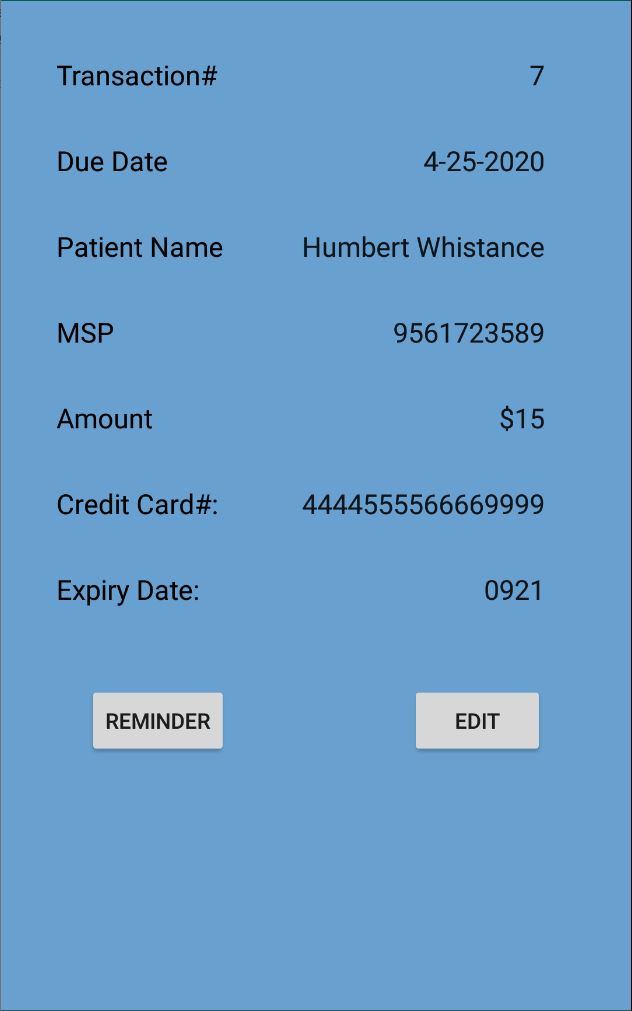
## New Transaction -Xinyi Liu activity\_cashier\_transaction\_new.xml cashier\_transaction\_new.java



The cashier receives the report from the doctor. Then cashier can enter the amount and due date for this bill. Click done button will send the bill to the customer.

If cashier verify the MSP number, he/she can click the MSP check button that will close the billing and set the amount to 0, the patient will not be required to pay the bill. The transaction will go to the archive container.

## View Pending Transactions -Xinyi Liu activity\_cashier\_transactions\_view.xml cashier\_transactions\_view.java



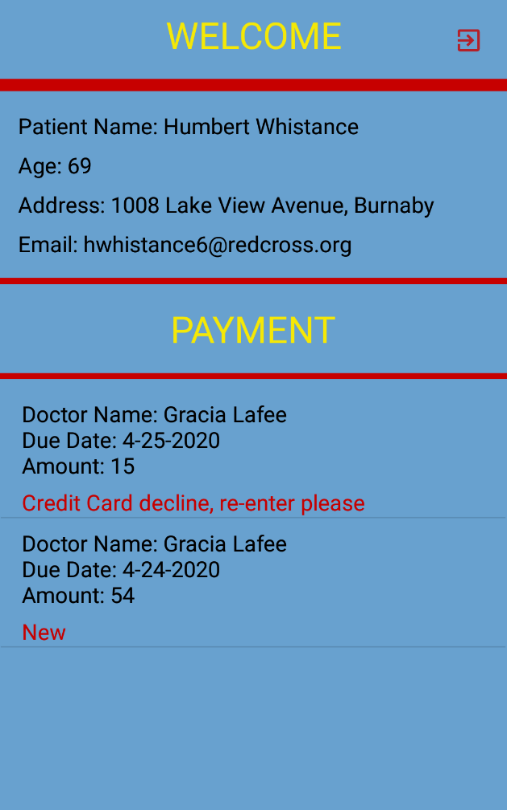
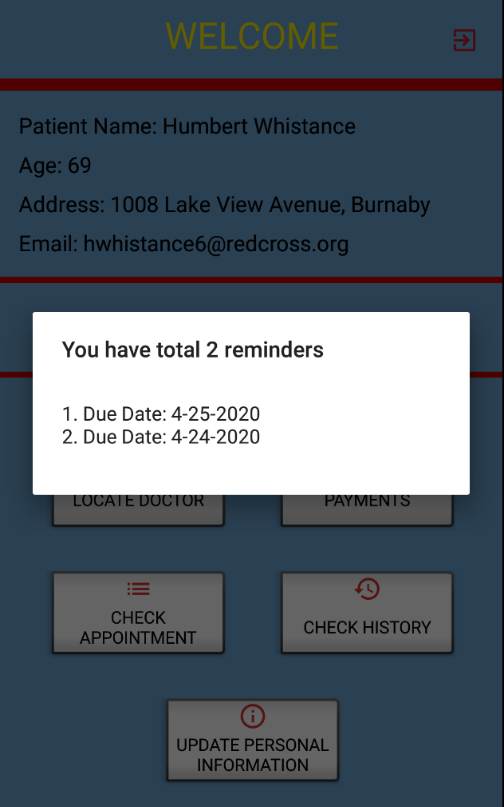
Cashiers can view the credit card info and change the due date or amount.

After cashiers check the credit card information, if the credit card information is accepted, cashiers will click closes Transaction, it means payment is accepted, close the billing.

If the credit card information should be declined, cashier can click Declining Credit. System will ask patients to update their credit card info.

MSP rechecked also closed the billing. Base on the MSP manually checking.

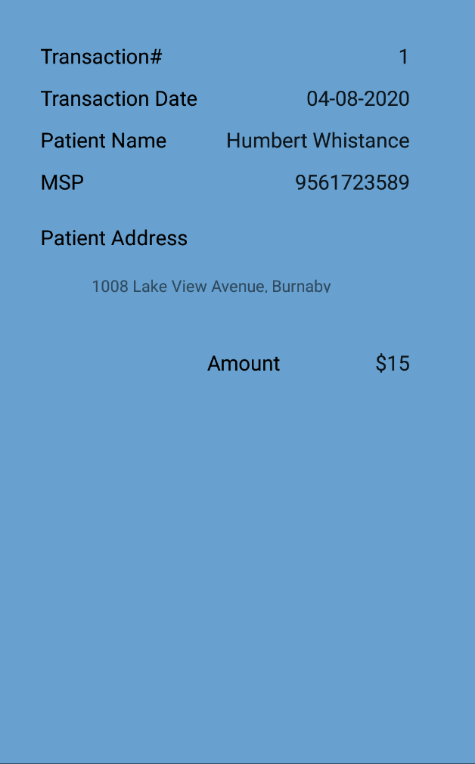
Reminder will send a one-time reminder to the patient.



Reminder will show one time when the patient login.

## Transaction Archive -Xinyi Liu

activity\_cashier\_transactions\_archive.xml cashier\_transactions\_archive.java



A collection of the history transaction.

# EXTRAS This application has 24 screen xml files, and 8 list view xml files, 3 fragment xml files, 1 number picker xml, 1 single message xml and 3 popup menu xml files. ( total 40 xml files)

This application total has 31 java files, include 3 fragments, 1 database helper and 2 adapters.  
  
popup menu and customer number picker are the two features that don’t covered in class.

In the booking appointment feature, we used time picker at beginning, but we figure out it can not tell patient which time is available or not. So, we choose to learn some other way on the Internet. So found the customer number picker can fit our requirement.

How does the customer number picker work in our project?

First, the doctor only accepts appointments from 8:00-16:30, 30 mins interval . So, we convert hours and minutes to an integer array A. Hours multiply by 2, if the minutes is 30, plus one. For example, 9:00 9\*2=18, 13:30 13\*2+1=27. So, we set a integer array from 16 to 33.

Second, system will store all booked times for the doctor at selected day and convert the times to array B by same method as first step.

Third, delete the same element between array A and array B. So, system can get an available time array C.

Fourth, reverse the integer array C to a String array D which has the time String.

Fifth, Set the array D to number picker. Then the patient only can select available time for the appointment.