NOTTINGHAM
TRENT UNIVERSITY

College of Science and Technology
School of Science and Technology

SOFT30121: Advanced Analysis and Design

Systems Analysis Design and Implementation

By

N0830182: Dayeeta Das
N0803016: Maleesha Dedigama Arachachige
N0796117: Maida Mehonjic
N0834113: Rishi Singh

Project Title: Open-Source Pundits Design, User Manual and Acceptance Testing Documentation

Contents

1. Design Documentation

For the Advanced Analysis Design module, the group was challenged to gather requirements for a pharmacy medication and bloodwork checking system. These requirements have been identified in the first specification document. From the highlighted requirements in the first documentation the group has created a system which allows pharmacists to manage prescriptions of patients and notify patients regarding collections or any bloodwork that the patient requires. This document highlights the key design decisions the group has taken to build this system.

1.1 Architecture Diagram

During the requirements gathering and the planning of the project the group had decided to use Flask as a web development framework due to the range of skillsets the group has. Most of the group had a good understanding of Python, HTML and CSS. By choosing Flask as our programming environment the group adhered to a Model-View-Controller (MVC) architecture benefiting from the advantages of the use of SQL Alchemy.

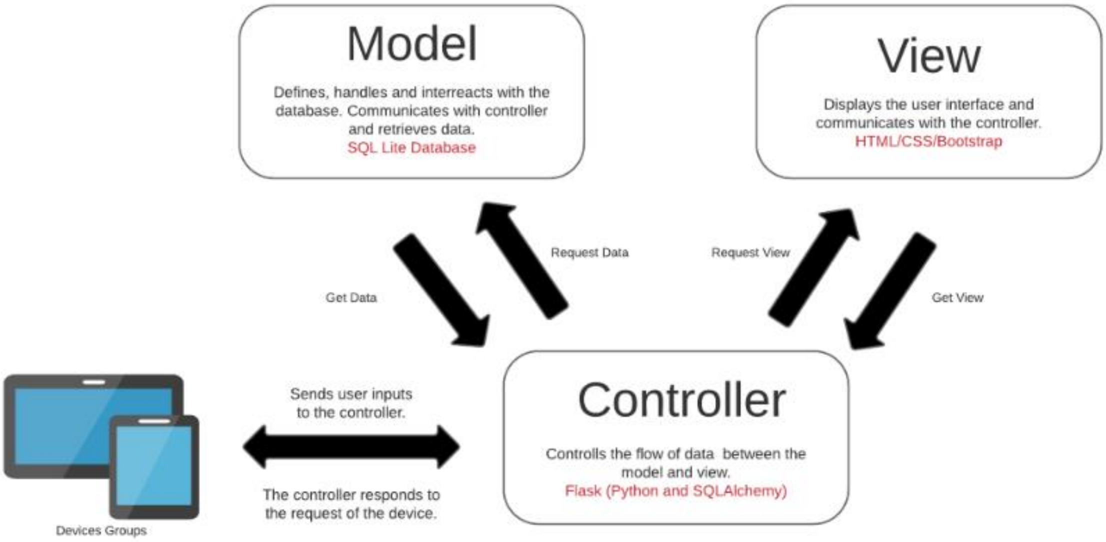

Model
Defines, handles and interreacts with the database. Communicates with controller and retrieves data.
SQL Lite Database
View
Displays the user interface and communicates with the controller.
HTML/CSS/Bootstrap
Request Data
Get Data
Request View
Get View
Sends user inputs to the controller.
The controller responds to the request of the device.
Devices Groups
Controller
Controlls the flow of data between the model and view.
Flask (Python and SQLAlchemy)

Figure 1: MVC Architecture

The MVC architecture displays the separation of an application into three 'logical components: The Model, the View and the Controller' (Unknown, 2021). The diagram in Figure 1 displays the MVC architecture used in this project. A similar architecture has been used by Pop and Alter in their deigning of an MVC Model for a Rapid Web Development project (Dragos-Paul Pop and Nelu, 2014). Due to the use of the MVC architecture Pop and Alter conclude that the development time of a web application has drastically reduced and there is an overall improvement in the speed and the quality in the work conduct using the architecture.

- Model

The Model layer in our project can be identified as the python and SQL Alchemy code which interacts with the SQL Lite database to send and receive data depending on the users request at in the view layer. The Model layer in the current project is be represented by the database.py created which initialises the tables in the SQL Lite database. Communication between this file and the app.py file can be understood as the communication between the Model and controller. The model layer is described as the b business logic of the web program which shows the process of accessing data in a database. In our application the Model is a key layer as we have been able to inject SQL statements and data into the database as dummy data. This has been extremely important to the system to show which data is required from System One when both systems are integrated.

- View

The View layer in our project is represented by the HTML, CSS and Bootstrap which can mainly be found in the templates and static folder. The group has created a templates folder to hold all the HTML files and the static folder contains all the CSS files. The View layer is used for all the interaction between the user and the frontend of the system (UI – user interface).

- Controller

The Controller acts as a middleman between the Model and view as it processes different requests, business logic and the ability to 'manipulate data' (Unknown, 2021). The controller will have the ability to handle user interaction/input from the user interface and perform any logic required, if required the input data can be updated or added to the database in the Model layer. The Controller can request data back from the Model perform any logic required and then pass the final data to the view for the user to view.

Advantages

- The implementation of an MVC architecture 'increases the flexibility and adaptability' of the project due to the loose coupling the Model and the View.

- By using an MVC architecture the frontend and backend developers can work on the implementation simultaneously.

Disadvantages

- Using an MVC architecture can increase the complexity of a project as there are more layers of abstraction.

- Developers in the group are required to have a wide range of programming knowledge specifically in this project python, HTML, CSS, SQL-Alchemy and Flask framework structuring.

MVC Design Pattern

The group decided to use an event driven MVC design pattern for this application due to its benefits of keeping the design of the system simple for developers to understand and implement. The current design pattern has been adapted from the work of Wielemaker who usde mediators to bridge the level abstraction between different components of the architecture (Wielemaker, 2009). As shown in the figure below an event triggers the controller to execute a task depending on the event. The controller then manipulates data accordingly depending on the even itself and the business logic in the controller. After the manipulation of the data, the controller can now modify the view accordingly. The system required a pattern which can take in the user input and manipulate data accordingly to the logic defined in the controller. Accordingly the controller should modify the UI after the data is returned by the Model.

Advantages:

- The MVC design pattern can be seen as an event pattern which responds to the action of an event. Due to this the pattern is scalable and very useful as it allows the appliction to work on different types of devices espically with the use of Flask. The application is targetted towards patients and pharmacists and the ability to use different devices can be seen as a major advantage.
- Due to the pattern the system will multiple users to interact with the system at the same time. Patients and pharmacists will be able to use the applicaition as the system will be to handle both actors.
- This pattern increases the ease of updating the application and debuging the application due to the multiple levels in the application. Due to the intereaction with system one the it is important for the new system to be updated frequently.

Disadvantages:

- The MVC design pattern can be difficult to understand which can cause further issues in the development stages by delaying the progress of the implementation.

- The MVC design pattern is most suitable for small projets as the increase in size of the project can lead to the increase complexity of the implementation and the ability to manage the system.

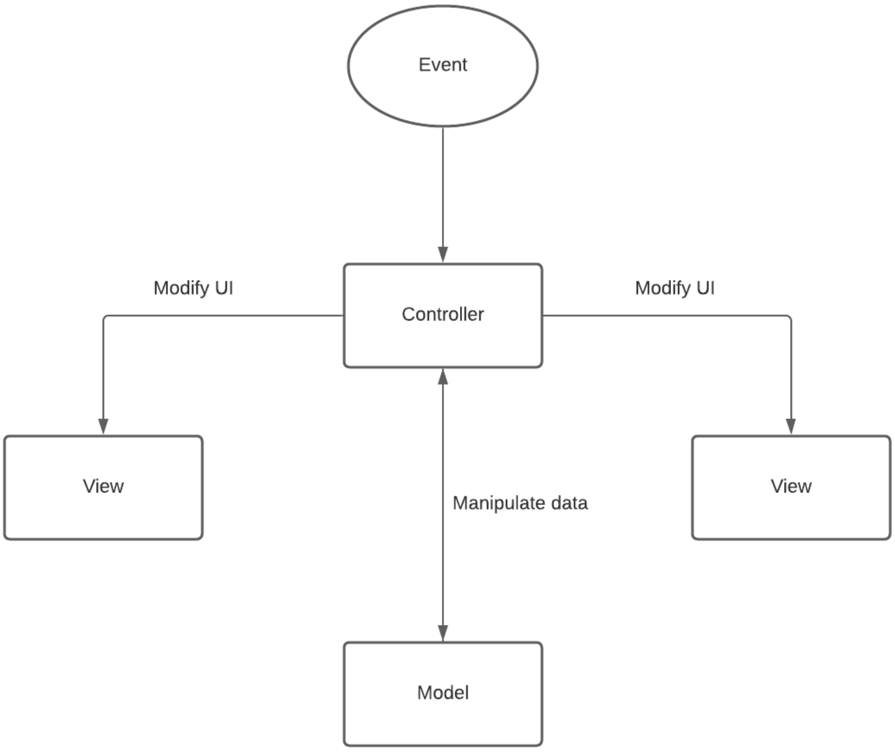


Figure 2: MVC design patterns

1.2 Deployment Diagram

Deployment diagram displays the execution architecture of a system and helps to visualise the physical hardware and software of a system. The diagram demonstrates how the application is deployed and acts upon user input, updates the user's requests, and accesses the database. OCP Pharmacy application uses SQLite for the database, Python Flask was used for the implementation and the application server is displayed using Bootstrap. As shown in the diagram, the system has different layout for the Patient and Pharmacist. Each user is responsible for handling different components of the system.

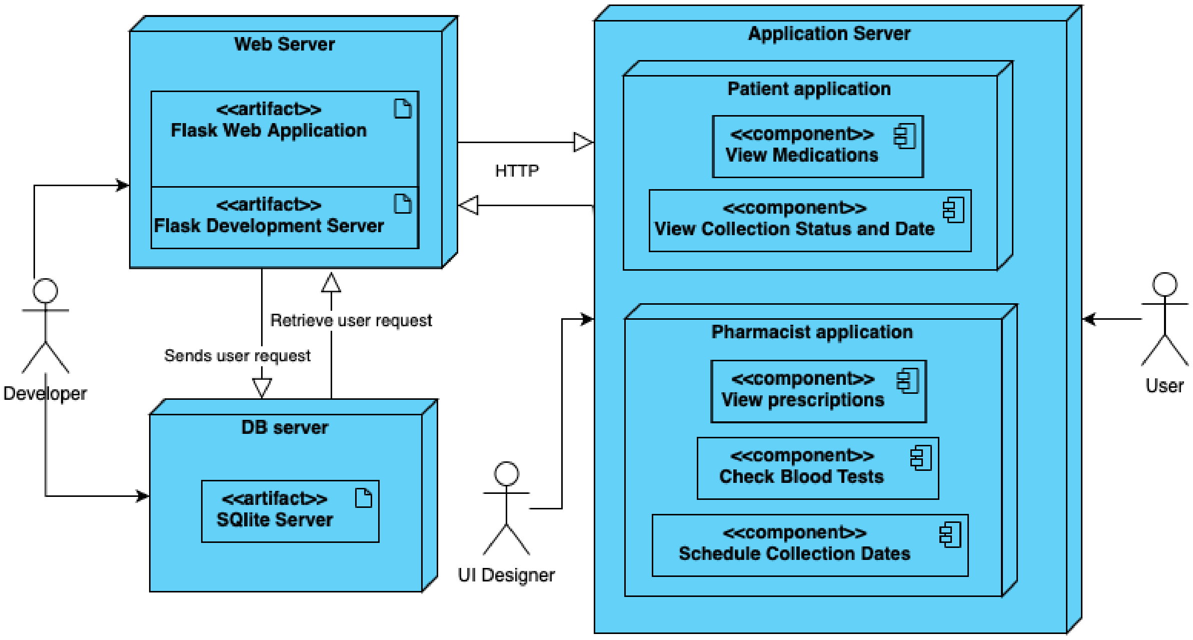


Figure 3: Deployment Diagram

1.3 Process Diagram

A Process Flow Diagram (PFD) is a type of flowchart used for depicting the vital components of a system and the relationship among them. Apart from Computer Science, it is widely used in chemical engineering and process engineering. It is not only used for illustrating a process but also for modifying an existing one and for improving it. Based on the type of process it represents, it may be called a Process Flow Chart, Flowsheet, Block Flow Diagram, Schematic Flow Diagram, Macro Flowchart, Top-down Flowchart, Piping and Instrument Diagram, System Flow Diagram or System Diagram.

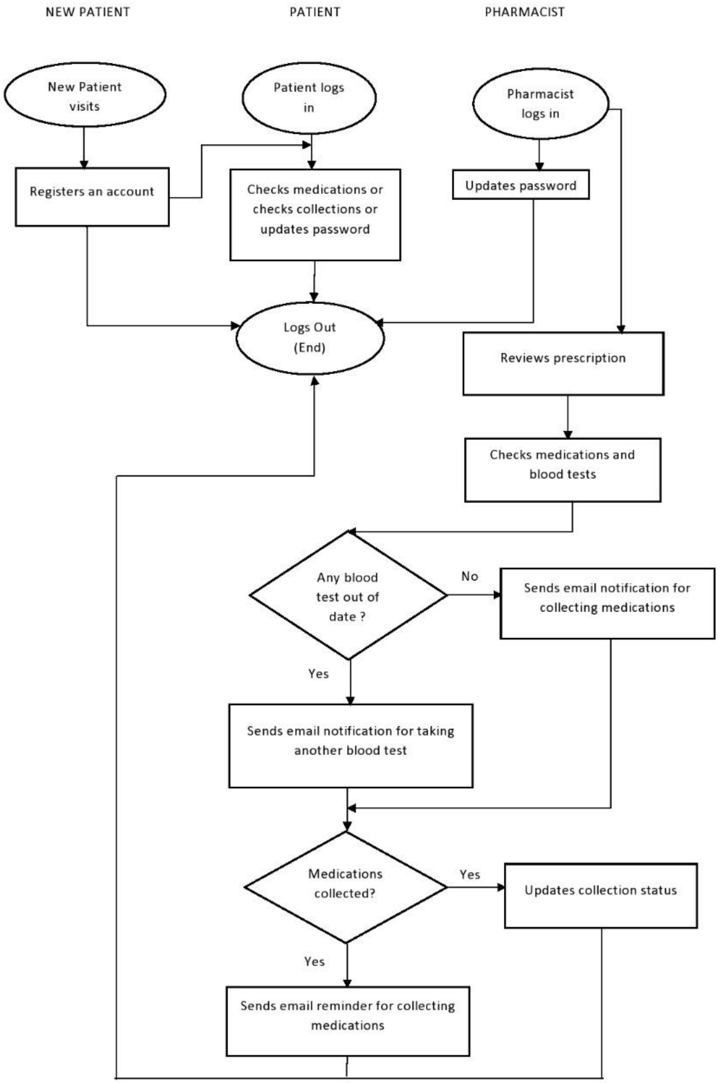

NEW PATIENT
PATIENT
PHARMACIST
New Patient visits
Patient logs in
Pharmacist logs in
Registers an account
Checks medications or checks collections or updates password
Updates password
Logs Out (End)
Reviews prescription
Checks medications and blood tests
Any blood test out of date ?
No
Sends email notification for collecting medications
Yes
Sends email notification for taking another blood test
Medications collected?
Yes
Updates collection status
Yes
Sends email reminder for collecting medications

Figure 4: Process Diagram

Advantages of using the process flow diagram for representing the system.

- Process flow diagram helped in representing a complex procedure in a simple visual form. This enabled everyone in the team to easily understand the processes involved in the system. Moreover, the developers used this diagram for updating and improving the functionality of the system by modifying the processes depicted in the diagram.
- This diagram enabled the developers to identify issues on the floor and accelerated the process of mitigating those issues. This in turn helped in speeding up the development.
- This diagram also helped in documenting the processes involved in the system. Being able to explain why a process is the way it is and have it on hand is a substantial advantage.

Disadvantages involved for using this diagram for representing this system:

- Alterations became a hassle as every time a new modification needed to be added, the entire diagram had to be redrawn from scratch.
- Complicated processes involved in the system might have made the diagram look a little messy.
- Reproducing a Process Flow Diagram can be rather difficult as it cannot be typed, and one must use Word, Excel or any other software for redrawing the shapes and plug words in them.

1.4 Structure Diagram

Structural modelling determines the key information in the problem domain; therefore, analysts can create structural diagrams to represent these identified elements of the system and their interactions with one another.

1.4.1 Class Diagram

The class diagram illustrates the structure of the data required by the system. The diagram shows relationships between classes and operations that can be performed by objects of the class.

The team have identified that the system will comprise of 7 classes. This includes patient, patient blood test, blood test, medication, prescription, collection, and pharmacist. The patient and pharmacist classes also represent the main actors of the system therefore operations relating to actions they can perform have been included.

A single patient may have 0 or many prescriptions. This one patient may also have 0 or many blood tests. One prescription can only have one collection. One prescription may contain one or many medications. One medication may require 0 or many blood tests. One pharmacist can approve one or many prescriptions.

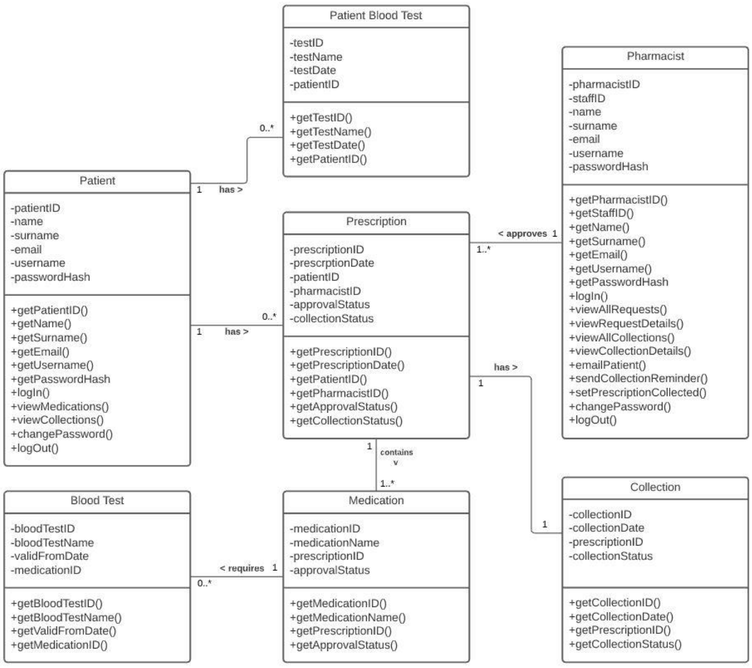


Figure 5: Class Diagram

Advantages of Class Diagrams:

- "Better understand the general overview of the schematics of an application." (lucidchart, n.d.)
- Provides a visual representation of the essential information required by the system.
- Highlights key functionality necessary to the system.
- "Provide an implementation-independent description of types used in a system." (lucidchart, n.d.)

Disadvantages of Class Diagrams:

- A poor understanding of the business requirements and problem domain may cause an inaccurate representation of what the system needs to include.
- More complex systems may be difficult to represent and the diagram may become difficult to understand.
- May require lengthy maintenance which can hinder the progress by the developers whilst they take time to synchronise code with the design.

1.4.2 Object Diagram

The object diagram illustrates instances of classes. The diagram contains example data to provide a better understanding of the information stored in each table and how they relate to each other.

A patient (pat1) has two prescriptions. One of these prescriptions (pres1) only contains one medication (med1). This medication does not require a blood test to be issued therefore the pharmacist (pha1) was able to issue the collection and pat1 was able to collect it.

Pat1 also has a second prescription (pres2) which contains three different medications (med2, med3, and med4). However, med2 and med4 require blood tests. Pat1 has had these two blood tests (adr1 & bp1). The (adr1) test for med2 is up to date and can be approved although the (bp1) test for med4 is out of date so the medication cannot be approved. The whole prescription is then unapproved because all medications requiring blood tests must be valid and up to date to be approved. There is no collection object for pres2 because of an invalid blood test for one of the medications.

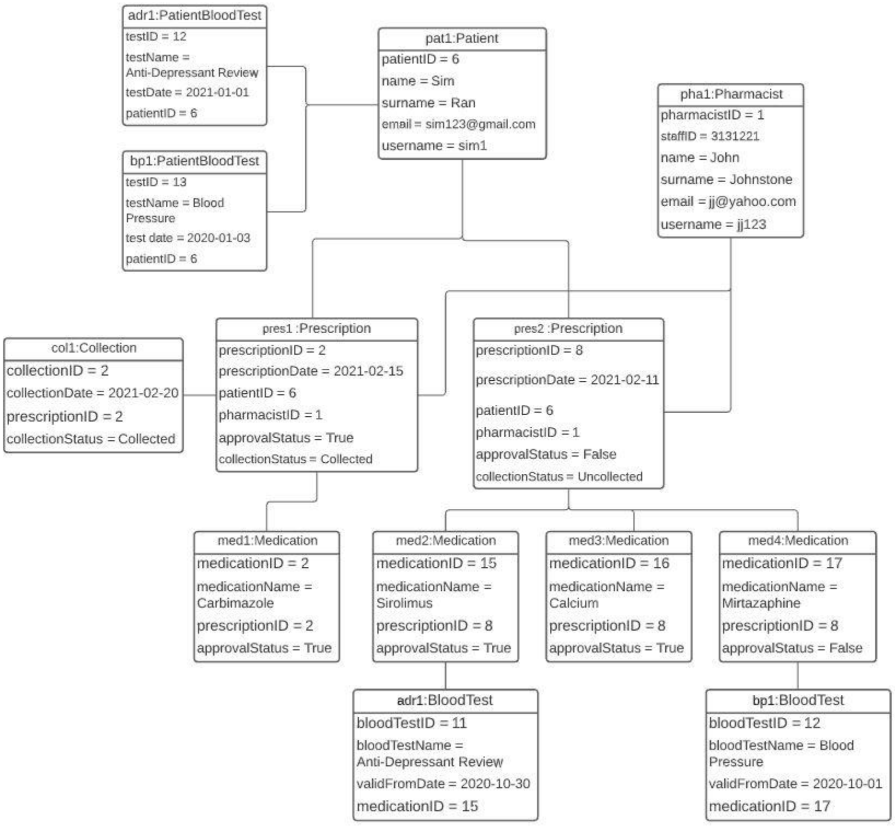


Figure 6: Object Diagram

Advantages of Object Diagrams:

- Illustrates a specific example of the structure described by the class diagram.
- Shows what type of data is expected to be held by an object.
- Visualises the multiplicities from the relationships labelled in the class diagram.
- Develops a better understanding of how objects relate and interact with each other.

Disadvantages of object diagrams:

- It depends on the class diagram.
    - If the class diagram is a poor representation of the information and processes required by the system, the object diagram will also be inaccurate.
    - Any changes made to the class diagram needs to be reflected in the object diagram.
    - Object diagram may require long maintenance if changes are made to the class diagram.

2. User Help documentation

Types of Users: Pharmacists and the Patients.
Views: Some web pages are common for both types of user. Other pages vary depending on the information that a particular user can access.
Login: Both types of users need to login when they visit the *Open-Source Pundits (OCP)* website.
They will need to enter their email and the password that was used for the registration of their accounts. The *Login pages* have the same design for both users.

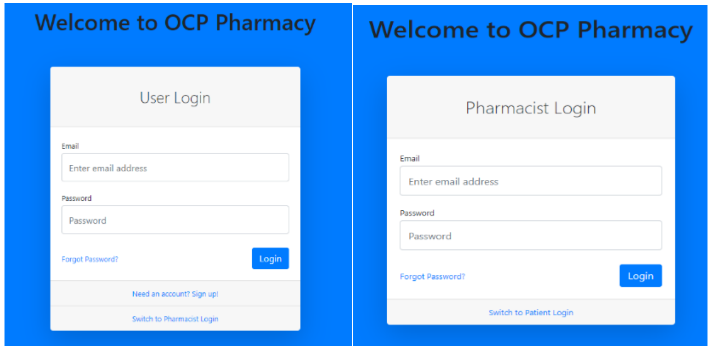


Fig 7. Login pages for the Pharmacist and the User.

Change Password: The *Change Password* section under the *Account* section of the *Settings* Page has a common template for both pharmacists and users. Here, a user needs to enter their existing password and the new password with which they want to replace the existing one.
Once the user hits the *'Change Password'* button the password gets updated and stored in the database.

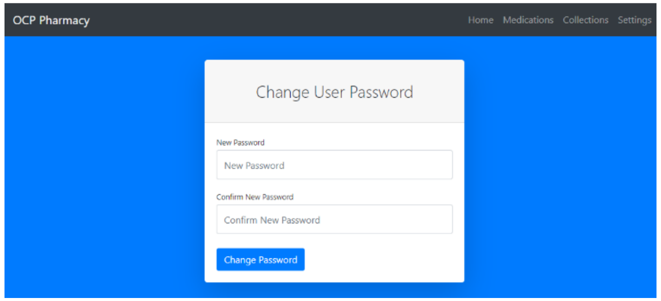
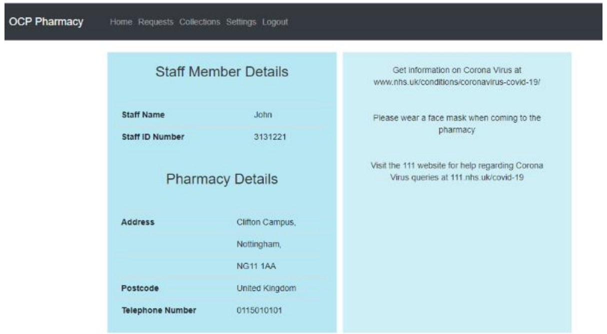


Fig 8. The picture above shows the Change Password page.

Pages Specific to the Pharmacists: Once, a Pharmacist logs in there are certain pages that only the Pharmacist can view. This includes the following pages: *Requests, Collections,* and the *Pharmacist's Home Page.*

Home Page: The Home Page for a Pharmacist displays the details for that particular Pharmacist.



Staff Member Details

Staff Name	John
Staff ID Number	3131221

Pharmacy Details

Address	Clifton Campus, Nottingham, NG11 1AA
Postcode	United Kingdom
Telephone Number	0115010101

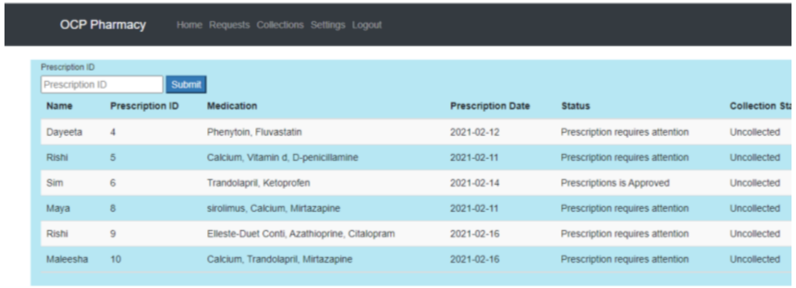
Get information on Corona Virus at www.nhs.uk/conditions/coronavirus-covid-19/

Please wear a face mask when coming to the pharmacy

Visit the 111 website for help regarding Corona Virus queries at 111.nhs.uk/covid-19

Fig 9. Home page and details for the Pharmacist named John.

Requests Page: The Pharmacist is the only user who would be able to access this page. This Page displays all the current prescription details along with the Patient's names which are either waiting to be upheld or approved.

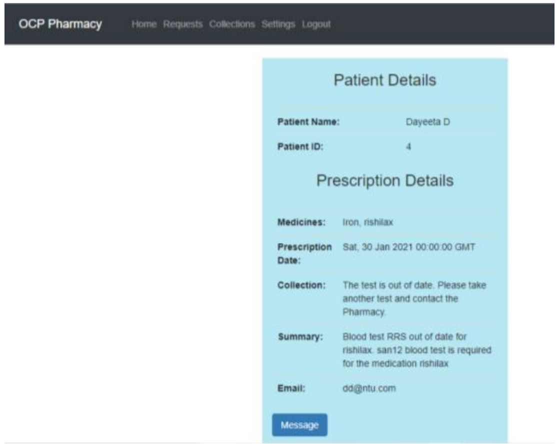


Name	Prescription ID	Medication	Prescription Date	Status	Collection St
Dayeeta	4	Phenytoin, Fluvastatin	2021-02-12	Prescription requires attention	Uncollected
Rishi	5	Calcium, Vitamin d, D-penicillamine	2021-02-11	Prescription requires attention	Uncollected
Sim	6	Trandolapril, Ketoprofen	2021-02-14	Prescriptions is Approved	Uncollected
Maya	8	sirolimus, Calcium, Mirtazapine	2021-02-11	Prescription requires attention	Uncollected
Rishi	9	Elleste-Duet Conti, Azathioprine, Citalopram	2021-02-16	Prescription requires attention	Uncollected
Maleesha	10	Calcium, Trandolapril, Mirtazapine	2021-02-16	Prescription requires attention	Uncollected

Fig 10. Requests page containing prescription details table.

The Pharmacist can approve a prescription or send an email to a patient depending on the status of the prescription. The Pharmacist needs to enter the Prescription ID for the Patient in the *"Prescription ID"* box and click the *"Submit"* button.

Once, the Pharmacist clicks the *"Submit"* button, it takes them to another page where an email summary has been created for sending an email to the patient whose ID was previously entered.



Patient Details

Patient Name: Dayeeta D

Patient ID: 4

Prescription Details

Medicines: Iron, rishilax

Prescription Date: Sat, 30 Jan 2021 00:00:00 GMT

Collection: The test is out of date. Please take another test and contact the Pharmacy.

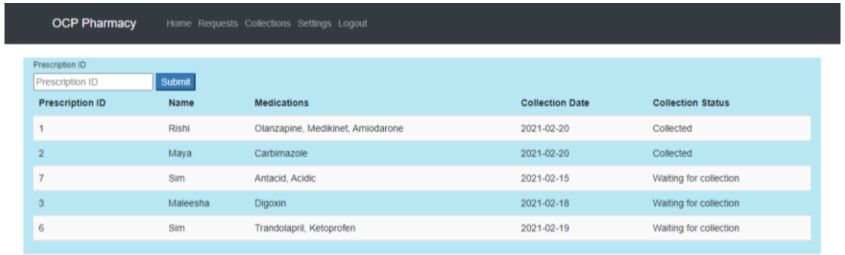
Summary: Blood test RRS out of date for rishilax. san12 blood test is required for the medication rishilax

Email: dd@ntu.com

Fig 11. Template for the email summary.

The Pharmacist should click the *"Message"* button for sending the mail to the patient. Once, the mail has been successfully sent to the patient an email confirmation page will appear.

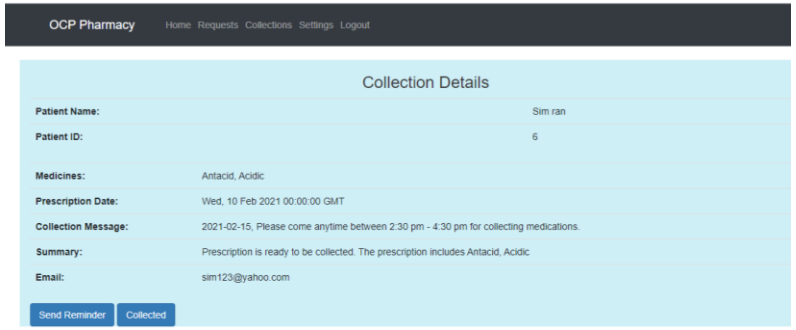
Collections Page: The Collections Page displays the prescription details for those patients whose prescriptions has been approved and is awaiting collection.



Prescription ID	Name	Medications	Collection Date	Collection Status
1	Rishi	Olanzapine, Medikinet, Amiodarone	2021-02-20	Collected
2	Maya	Carbimazole	2021-02-20	Collected
7	Sim	Antacid, Acidic	2021-02-15	Waiting for collection
3	Maleesha	Digoxin	2021-02-18	Waiting for collection
6	Sim	Trandolapril, Ketoprofen	2021-02-19	Waiting for collection

Fig 12. Collections Page on the Pharmacist's side.

The Pharmacist can enter the Prescription ID of a patient in the *"Prescription ID"* box and click the *"Submit"* button that will redirect them to another page.



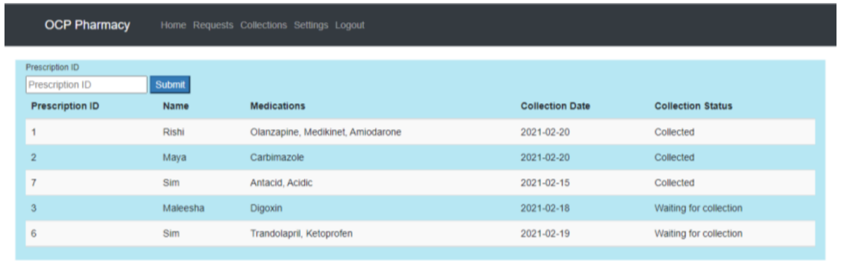
Collection Details

Patient Name:	Sim ran
Patient ID:	6
Medicines:	Antacid, Acidic
Prescription Date:	Wed, 10 Feb 2021 00:00:00 GMT
Collection Message:	2021-02-15, Please come anytime between 2:30 pm - 4:30 pm for collecting medications.
Summary:	Prescription is ready to be collected. The prescription includes Antacid, Acidic
Email:	sim123@yahoo.com

Fig 13. Template for collection summary.

The Pharmacist can click the *"Send Reminder"* button to email this particular patient to remind them of their collection. The message contains all the information that is displayed in the summary. The *"Collected"* button is clicked if the patient had already collected their prescription. This updates the status of this patient's collection.

If the Pharmacist clicks the *"Send Reminder"* button, they are shown a confirmation message stating that the email has been sent successfully. However, if the Pharmacist clicks on the *"Collected"* button, it will redirect them to the Collections Page where the prescription status for this particular patient would be updated to "Collected".

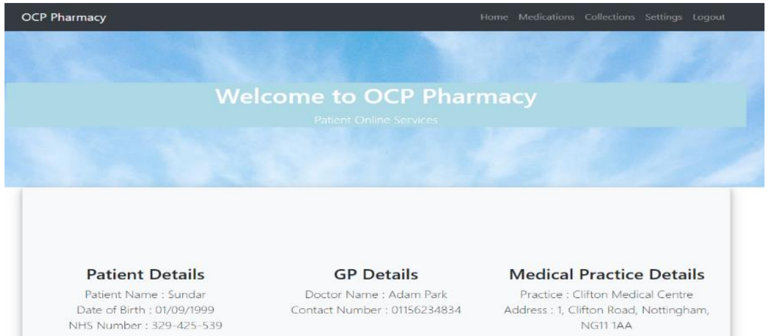


Prescription ID	Name	Medications	Collection Date	Collection Status
1	Rishi	Olanzapine, Medikinet, Amiodarone	2021-02-20	Collected
2	Maya	Carbimazole	2021-02-20	Collected
7	Sim	Antacid, Acidic	2021-02-15	Collected
3	Maleesha	Digoxin	2021-02-18	Waiting for collection
6	Sim	Trandolapril, Ketoprofen	2021-02-19	Waiting for collection

Fig 14. Collection Status for 'Sim' has been updated to "Collected".

Pages Specific to the Patients

Home Page: The Patient side Home Page has a different view and design as compared to the Pharmacist's side.
Most of this page is static except for the Patient Details section. This contains information about the particular patient who has logged in.



Patient Details
Patient Name : Sundar
Date of Birth : 01/09/1999
NHS Number : 329-425-539

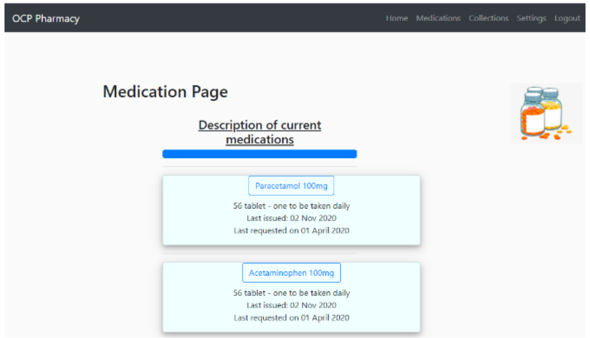
GP Details
Doctor Name : Adam Park
Contact Number : 01156234834

Medical Practice Details
Practice : Clifton Medical Centre
Address : 1, Clifton Road, Nottingham, NG11 1AA

Fig 15. Design for the patient home page.

Medications Page

This page is specific to the patients. It is a static page that displays the commonly prescribed medicines by the pharmacy.



Medication Page

Description of current medications

Paracetamol 100mg

56 tablet - one to be taken daily
Last issued: 02 Nov 2020
Last requested on 01 April 2020

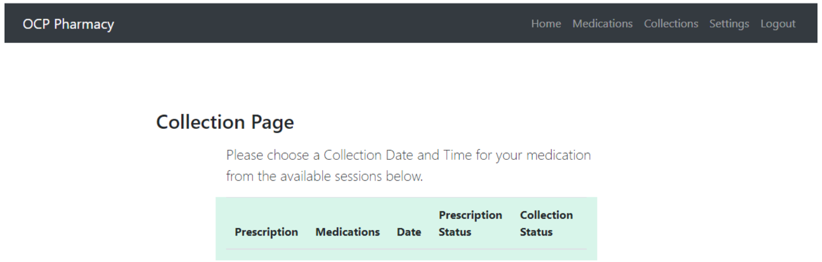
Acetaminophen 100mg

56 tablet - one to be taken daily
Last issued: 02 Nov 2020
Last requested on 01 April 2020

Fig 16. Patient medications page.

Collections Page

The Collections Page on the Patient's side has a different view to the Pharmacist's side. It displays the collection details only for this particular patient and is mostly a dynamic page.



Collection Page

Please choose a Collection Date and Time for your medication from the available sessions below.

Prescription	Medications	Date	Prescription Status	Collection Status

Fig 17. Patient collection Page for a particular user who does not have any prescription that needs to be collected at the moment.

3. Acceptance test plan

3.1 Functional Testing

Requirements	Test	Expected Result	Actual Result	Pass/Fail
System must verify the pharmacist and patient details and authorise access to the system.	Login page loads content	The expected result from this test was the website to load the Login form.	The actual result from this test was the Login page loading the Login form	Pass
System must create a log containing the email addresses and passwords and update it.	*Login* button on first page works	The expected result from this is the user's credentials (email address and password) being taken in and the *Login* button takes them to the landing Dashboard page.	The actual result from this test was that the user was taken to their account if the details were correct, and the landing Dashboard was displayed.	Pass
System must allow user to enter email address.	Email address box on the Login form accepts user input	The expected result from this test was for the user to be able to enter text into the input box.	The actual result from this test was the email address box allowing user input.	Pass
System must allow user to enter a password.	Password box on the Login form accepts user input.	The expected result from this test was for the user to be able to enter text into the input box.	The actual result from this test was the password box allowing user input.	Pass
System should provide security for the user.	Password box on the Login form hides the user's input.	The expected result from this test was the user input to be replaced by dots to hide their password.	The actual result from this test was the password box hiding the input by replacing characters with dots.	Pass
System must be able to authorise users of the system.	The website attempts to log the user in when the *Login* button is clicked.	The expected result from this test was for the website to run the login script and attempt to log the user in.	The actual result from this test was the website running the login script when the *Login* button was clicked.	Pass
System must be able to authenticate users.	The Login form rejects the login if the email address or password field is empty.	The expected result from this test was for the site to reject the login attempt if either the email or password fields had been left empty.	The actual result from this test was the website rejecting the login if any of the fields were left empty.	Pass
System must be able to authenticate users.	The login form rejects the login if the email address is invalid.	The expected result from this test was for the site to reject the login attempt if the email was not found in the database.	The actual result from this test was the website rejecting the input if the email was not found.	Pass
System must be able to authenticate users.	The login form rejects the login if the password is invalid.	The expected result from this test was for the site to reject the login attempt if the password is incorrect.	The actual result from this test was the website rejecting the input if email address was not found.	Pass

System must have a pharmacist's page.	The Pharmacist dashboard page is loaded if the pharmacist is logged in.	The expected result from this test was for the site to take the user to the Pharmacist Dashboard page when the *Login* button was clicked. The user had to have a verified Pharmacist login details.	The actual result from this test was the website taking the user to the Pharmacist Dashboard page when the *Login* button was clicked.	Pass
System should have a patient's side view.	The Patients page is loaded if a patient is logged in.	The expected result from this test was for the site to take the user to the Patient Dashboard page when the *Login* button was clicked. The user had to have a verified Patient login details.	The actual result from this test was the website taking the user to the Patient Dashboard page when the *Login* button was clicked, and the patient logins are verified.	Pass
System should have a patient's side view.	Pharmacist Page loads the content when logged in.	The expected result from this test was the website to load the Pharmacists Page content correctly.	The actual result from this test was the website to load and display the Pharmacist Dashboard page content.	Pass
Pharmacist must be able to view prescriptions.	Request page loads when the user clicks on the *Request* button from the navigation bar.	The expected result from this test was the website to load the Request page.	The actual result from this test was the website to load and display the Request page.	Pass
Pharmacist must be able to see collections.	Collections page loads when the user clicks on the *Collection* button from the navigation bar.	The expected result from this test was the website to load the Collection page.	The actual result from this test was the website to load and display the Collection page.	Pass
System should have a settings page.	Settings page loads when the user clicks on the *Settings* button from the navigation bar.	The expected result from this test was the website to load the Settings page.	The actual result from this test was the website to load and display the Settings page.	Pass
System must be able to logout users.	Logout page loads when the user clicks on the *Logout* button from the navigation bar.	The expected result from this test was the website Logout the user.	The actual result from this test was the website to logout the user from the system.	Pass
System must notify the pharmacist about a medical request.	The pharmacist enters a prescription ID in the input box.	The expected result from this test was for the Pharmacist to be able to enter text into the input box.	The actual result from this test was the Pharmacist being able to enter a prescription ID to the input box.	Pass

The system must maintain a log containing the medical ID and the dates whenever a medical request is made and save it automatically.	The Pharmacist submits the prescription ID when the *Submit* button is clicked.	The expected result from this test was for the Pharmacist to enter the ID, e.g., 5, and click submit to be displayed with the prescription details for the specific patient.	The actual result from this test was the Pharmacist entered the ID, e.g., 5, submitted and was displayed with the prescription details for the specific patient.	Pass
System must allow the pharmacist to check whether the medication requires a blood test.	Prescription status page loads when the Pharmacists clicks on the *Message* button.	The expected result from this test was to take the Pharmacists to the prescription status page and be shown the "Email successfully sent" message.	The actual result from this test was the Pharmacists to be directed to the prescription status page and be displayed with successful message.	Pass
The system must maintain a log containing the medical ID's and collections ID's whenever a collection is scheduled and automatically save it.	The pharmacist enters a prescription ID in the input box.	The expected result from this test was for the Pharmacist to be able to enter text into the input box.	The actual result from this test was the Pharmacist being able to enter a prescription ID to the input box.	Pass
The system must be able to keep track of the orders that has been approved by the pharmacist and yet to be approved.	The Pharmacist submits the prescription ID when the *Submit* button is clicked.	The expected result from this test was for the Pharmacist to enter the ID, e.g., 1, and click submit to be displayed with the Collection details for the specific patient.	The actual result from this test was the Pharmacist to enter the ID, e.g., 1, submitted and was displayed with the Collection details for the specific patient.	Pass
The pharmacist could be able to notify the patient.	The Pharmacist clicks on the *Message* button.	The expected result from this test was for the Pharmacist to be redirected to the Collection page when *Message* button is clicked.	The actual result from this test was the Pharmacist was redirected to the Collection page when *Collected* button was clicked.	Pass
The pharmacist could be able to notify the patient.	The Pharmacist clicks on the *Send Reminder* button.	The expected result from this test was for the Pharmacist to be able to send another email to the patient to be reminded of their collection status.	The actual result from this test was for the Pharmacist to be able to send another email to the patient to be reminded of their collection status	Pass
System must allow patient to log in.	Patient Page loads the content when logged in.	The expected result from this test was the website to load the Patient Page content correctly.	The actual result from this test was the website to load and display the Patient Dashboard page content.	Pass
Patient could see a list of medications.	Medications page loads when the user clicks on the *Medications* button from the nav bar.	The expected result from this test was the website to load the Medication page.	The actual result from this test was the website to load and display the Medication page.	Pass

Patient could see list of collections.	Collections page loads when the user clicks on the *Collections* button from the nav bar.	The expected result from this test was the website to load the Collection page.	The actual result from this test was the website to load and display the Collection page.	Pass
Settings page could be available to the patient.	Settings page loads when the user clicks on the *Settings* button from the nav bar.	The expected result from this test was the website to load the Settings page.	The actual result from this test was the website to load and display the Settings page.	Pass
System must allow user to logout.	Logout page loads when the user clicks on the *Logout* button from the nav bar.	The expected result from this test was the website Logout the user.	The actual result from this test was the website to logout the user from the system.	Pass
System could show medications of patient.	Medication page is displayed with a description of Patient’s medications.	The expected result from this test was to display the medications and its descriptions the patient is currently taking.	The actual result from this test was to display the medications the patient currently taking.	Fail
System must display the selected collection date and timeslot to the Patient.	Patients are displayed with the allocated timeslot for the collection.	The expected result from this was to be shown with allocated time slot for the collection.	The actual result from this test was timeslot and date were presented to the Patient.	Pass
System must allow the user to reset the password in case it is wrong.	Change Password page is displayed when the Patient clicks on *Change Password* button.	The expected result from this was to direct the user to change password page.	The actual result from this page was patient was directed to the right page for them to change the password.	Pass

3.2 Performance Positive and Negative Test Plan

Login

Test Case Description	Test Type (Positive/ Negative)	Result (Pass/ Fail)	Result Description
Pharmacist log into system with correct email and password.	Positive	Pass	Correct credentials for pharmacist were entered. Pharmacist successfully logged in and was re-directed to the pharmacist home page.
Patient log into system with correct email and password.	Positive	Pass	Correct credentials for pharmacist were entered. Patient successfully logged in and was re-directed to the patient home page.
Pharmacist cannot log into system with misspelt email but correct password.	Negative	Pass	A misspelt pharmacist email was entered. The user could not log into the system. "Login Unsuccessful" displayed.
Pharmacist cannot log into system with correct email but misspelt password.	Negative	Pass	A misspelt pharmacist password was entered. The user could not log into the system. "Login Unsuccessful" displayed.
Pharmacist cannot log into system with email and password that do not match. (existing username and password)	Negative	Pass	Unmatched pharmacist credentials were entered. The user could not log into the system. "Login Unsuccessful" displayed.
Patient cannot log into system with misspelt email but correct password.	Negative	Pass	A misspelt patient email was entered. The user could not log into the system. "Login Unsuccessful" displayed.
Patient cannot log into system with correct email but misspelt password.	Negative	Pass	A misspelt patient password was entered. The user could not log into the system. "Login Unsuccessful" displayed.
Patient cannot log into system with email and password that do not match. (existing username and password)	Negative	Pass	Unmatched patient credentials were entered. The user could not log into the system. "Login Unsuccessful" displayed.

Navigation

Test Case Description	Test Type (Positive/ Negative)	Result (Pass/ Fail)	Result Description
Patient nav bar re-directing correctly to the home page.	Positive	Pass	"Home" was clicked, and the user was successfully re-directed to the patient home page.
Patient nav bar re-directing correctly to the medications page.	Positive	Pass	"Medications" was clicked, and the user was successfully re-directed to the patient medications page.
Patient nav bar re-directing correctly to the collections	Positive	Pass	"Collections" was clicked, and the user was successfully re-directed to the

page.			patient collections page.
Patient nav bar re-directing correctly to the settings page.	Positive	Pass	“Settings” was clicked, and the user was successfully re-directed to the patient settings page.
Patient nav bar re-directs correctly to patient login page when “logout” is clicked.	Positive	Pass	“Logout” was clicked, and the user was successfully logged out and re-directed to the patient login page.
Pharmacist nav bar re-directing correctly to the home page.	Positive	Pass	“Home” was clicked, and the user was successfully re-directed to the pharmacist home page.
Pharmacist nav bar re-directing correctly to the requests page.	Positive	Pass	“Requests” was clicked, and the user was successfully re-directed to the pharmacist requests page.
Pharmacist nav bar re-directing correctly to the collections page.	Positive	Pass	“Collections” was clicked, and the user was successfully re-directed to the pharmacist collections page.
Pharmacist nav bar re-directing correctly to the settings page.	Positive	Pass	“Settings” was clicked, and the user was successfully re-directed to the pharmacist settings page.
Pharmacist nav bar re-directs correctly to pharmacist login page when “logout” is clicked.	Positive	Pass	“Logout” was clicked, and the user was successfully logged out and re-directed to the pharmacist login page.
Pharmacist pages are inaccessible for patient users.	Negative	Pass	Patient navigation re-directs patient to patient pages.
Patient pages are inaccessible for pharmacist users.	Negative	Pass	Pharmacist navigation re-directs pharmacist to pharmacist pages.
Pages are inaccessible for non-logged in users.	Negative	Pass	Login screen with error message displayed. “Please log in to access this page.”

Reading and output of data from database

Test Case Description	Test Type (Positive/ Negative)	Result (Pass/ Fail)	Result Description
Home page displays correct details about the logged in user.	Positive	Pass	The details of the logged in user is displayed and are correct.
Correct collection details are displayed for the patient.	Positive	Pass	The details of the logged in user is displayed and are correct.
Requests table holds correct data.	Positive	Pass	The requests table is populated with the correct information.
Collections table holds correct data.	Positive	Pass	The collections table is populated with the correct information.

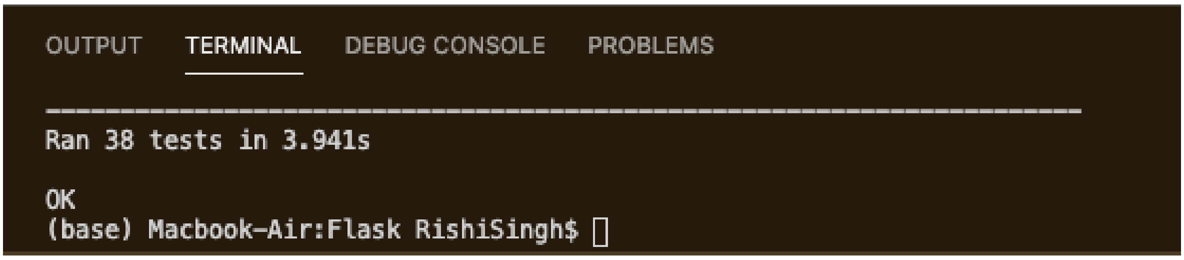
Checking medication logic

Test Case Description	Test Type (Positive/ Negative)	Result (Pass/ Fail)	Result Description
A prescription with one medication that does not require bloodwork can be collected.	Positive	Pass	Prescription is approved ("does not require attention") therefore is moved to the collections table once patient is emailed.
A prescription with one medication that requires bloodwork with valid test can be collected.	Positive	Pass	Prescription is approved ("does not require attention") therefore is moved to the collections table once patient is emailed.
A prescription with one medication that requires bloodwork with invalid test cannot be collected.	Negative	Pass	Prescription is not approved ("requires attention") therefore is not moved to the collections table.
A prescription with medication that does not require bloodwork and medication that requires bloodwork with valid test can be collected.	Positive	Pass	Prescription is approved ("does not require attention") therefore is moved to the collections table once patient is emailed.
A prescription with medication that does not require bloodwork and medication that requires bloodwork with invalid test cannot be collected.	Negative	Pass	Prescription is not approved ("requires attention") therefore is not moved to the collections table.
A prescription with medication that requires bloodwork with valid test and medication that requires bloodwork with invalid test cannot be collected.	Negative	Pass	Prescription is not approved ("requires attention") therefore is not moved to the collections table.
An unapproved prescription displays details of which medication requires bloodwork and which test/s are needed.	Positive	Pass	Unapproved prescription details display the medications with invalid test and which test/s are required.
A collection email is sent to the patient with an approved prescription when the pharmacist presses "Message" button.	Positive	Pass	An email with the prescription's contents, collection date and time is successfully sent to the patient.
An email is sent to the patient with an unapproved prescription when the pharmacist presses "Message" button.	Positive	Pass	An email with the unapproved medications and the tests required is successfully sent to the patient.
A reminder email is sent to the patient with an approved prescription is sent when the pharmacist presses the "Send Reminder" button.	Positive	Pass	A reminder email with collection details is successfully sent to the patient.

3.3 Unit tests

3.3.1 Unittest Framework

The functional testing is supported by the unit tests created in the unitTesting.py file which can be found with the source code. The unit tests created to for each HTML page located in the templates folder. A total of 38-unit tests have been created which have all passed and can be seen in the figure below. Unit tests are particularly important to the testing phase of this project as the manual visual testing in the functional tests can be automated by unit testing using the unittest framework which is compatible with flask. This ensures that each individual section of the program is thoroughly checked, and errors can be identified.



```
----------------------------------------------------------------------
Ran 38 tests in 3.941s

OK
(base) Macbook-Air:Flask RishiSingh$ ▯
```

Figure 18: Unit Test Result

Unit tests can be described as the being the 'most effective means to test' small components of the program checking the behaviour of the user's inputs with the behaviour of what is expected from the system (Ellims, Bridges and Ince, 2021). The advantage of using unit tests to check the implementation of the system is by automating small tests so errors can be easily identified as time is saved due to the automation of the tasks and there are less chances making errors during the testing.

The **unitTesting.py** file has been split into 8 different sections which represent the main html pages that interreact with the systems business logic. An example of the code can be seen in the **unitTesting.py** file and the result of the execution of the unit tests can be seen the figure above. The unit tests focus on the loading of specific HTML pages along with the content being produced on the pages. Along with this the interaction between the user and the system has also been checked by inputting parameters such as the prescription id in specific unit tests. An example of this can be seen by using the value 2 as a prescription ID and the expected behaviour from the system is to return data regarding the prescription with an ID 2. In this case the name of the patient is check which is 'Maya.' This test, successfully passes, and similar tests have been executed for other pages in the system as well.

References

Ellims, M., Bridges, J. and Ince, D.C. (2021). Unit testing in practice - Open Research Online. *Open.ac.uk*. [online] Available at: http://oro.open.ac.uk/3667/ [Accessed 19 Feb. 2021].

lucidchart.com, n.d., *What is Class Diagram?* [Online] Available at: https://www.lucidchart.com/pages/uml-class-diagram/ [Accessed January 2021].

educba.com, n.d., *Class Diagram*. [Online] Available at: https://www.educba.com/class-diagram/ [Accessed January 2021].

Visual-paradigm.com. 2021. *What is Deployment Diagram?*. [online] Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/ [Accessed 17 February 2021].

Lucidchart. (2019). What is a Process Flow Diagram. [online] Available at: https://www.lucidchart.com/pages/process-flow-diagrams.

PlanetTogether (n.d.). Advantages and Disadvantages of Flowcharts. [online] Available at: https://www.planettogether.com/blog/advantages-and-disadvantages-of-flowcharts.

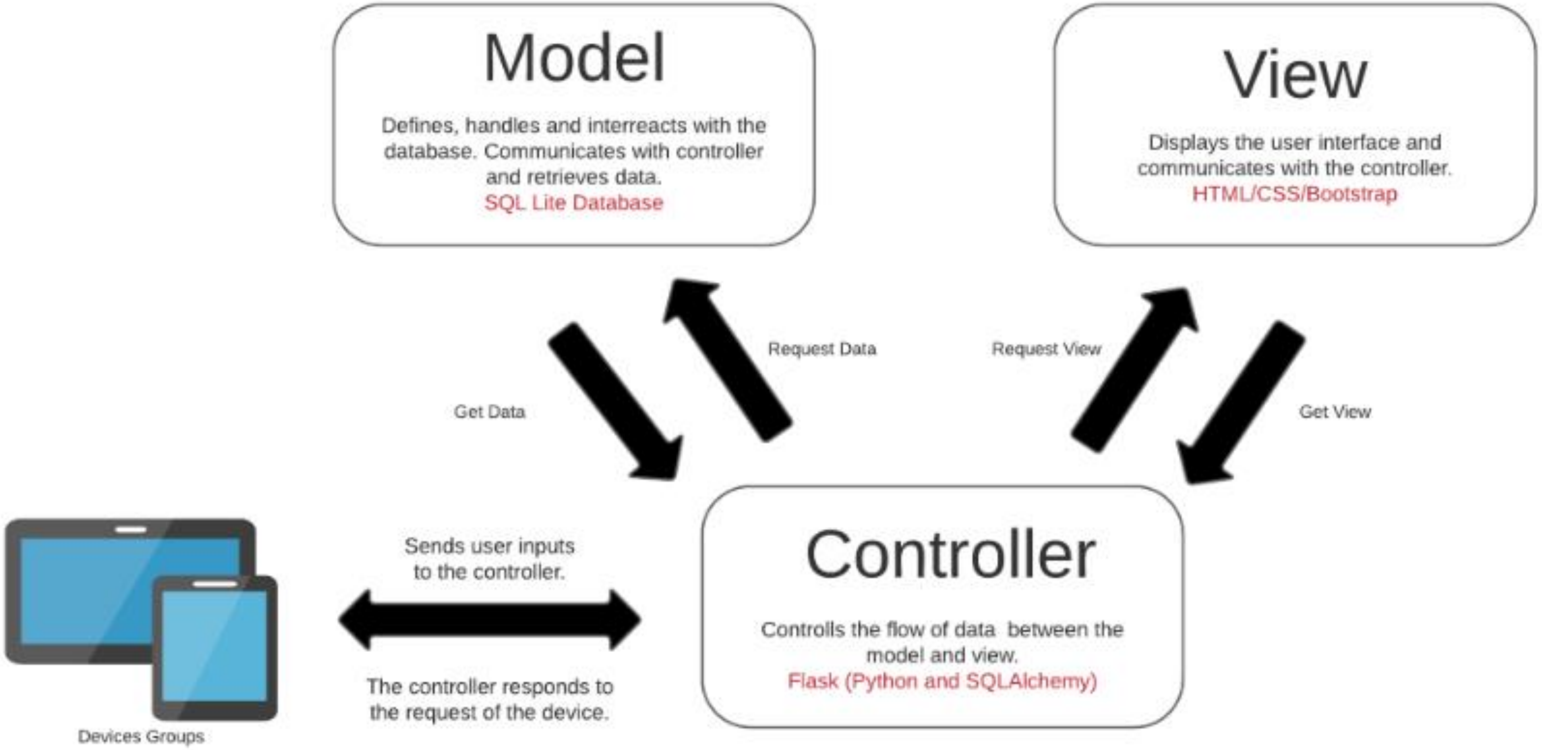
Wielemaker, J. (2021). *Logic programming for knowledge-intensive interactive applications*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/254852917_Logic_programming_for_knowledge-intensive_interactive_applications [Accessed 19 Feb. 2021].

Unkown. (2021). *MVC Framework - Introduction - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm [Accessed 19 Feb. 2021].

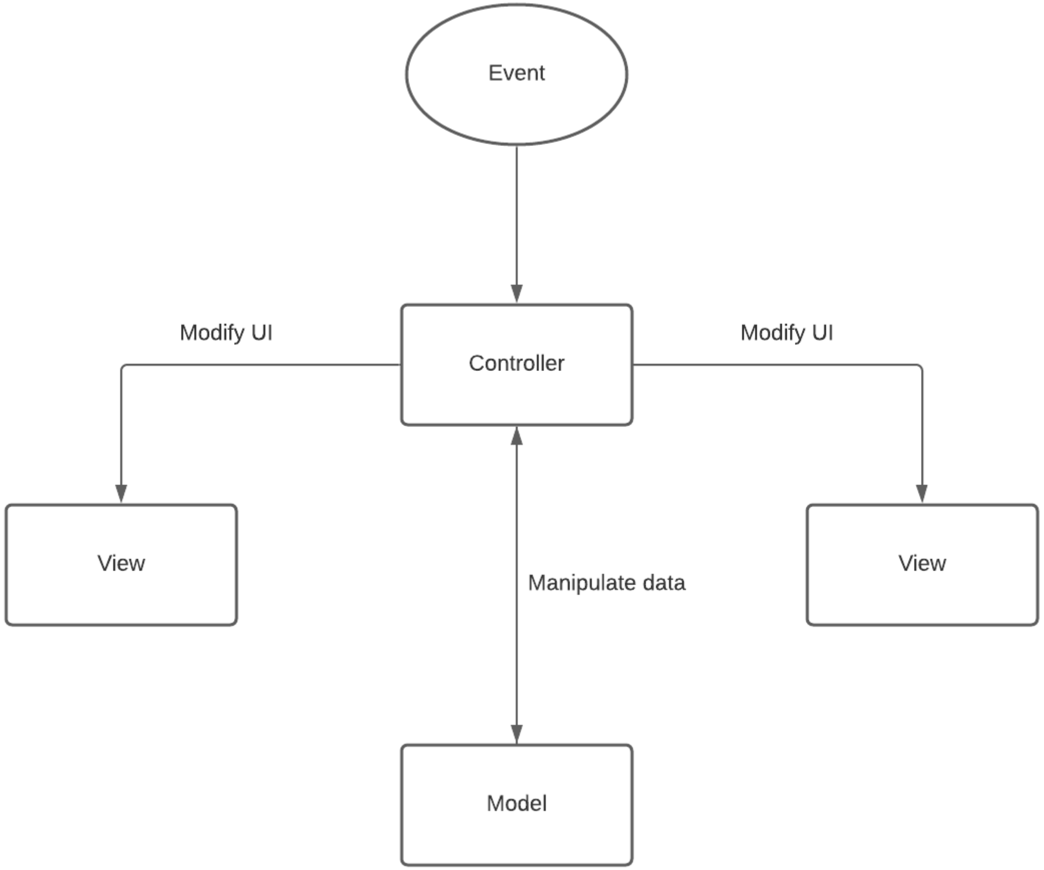
Dragos-Paul Pop and Nelu, A. (2014). *Designing an MVC Model for Rapid Web Application Development*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/275540078_Designing_an_MVC_Model_for_Rapid_Web_Application_Development [Accessed 19 Feb. 2021].

Appendices

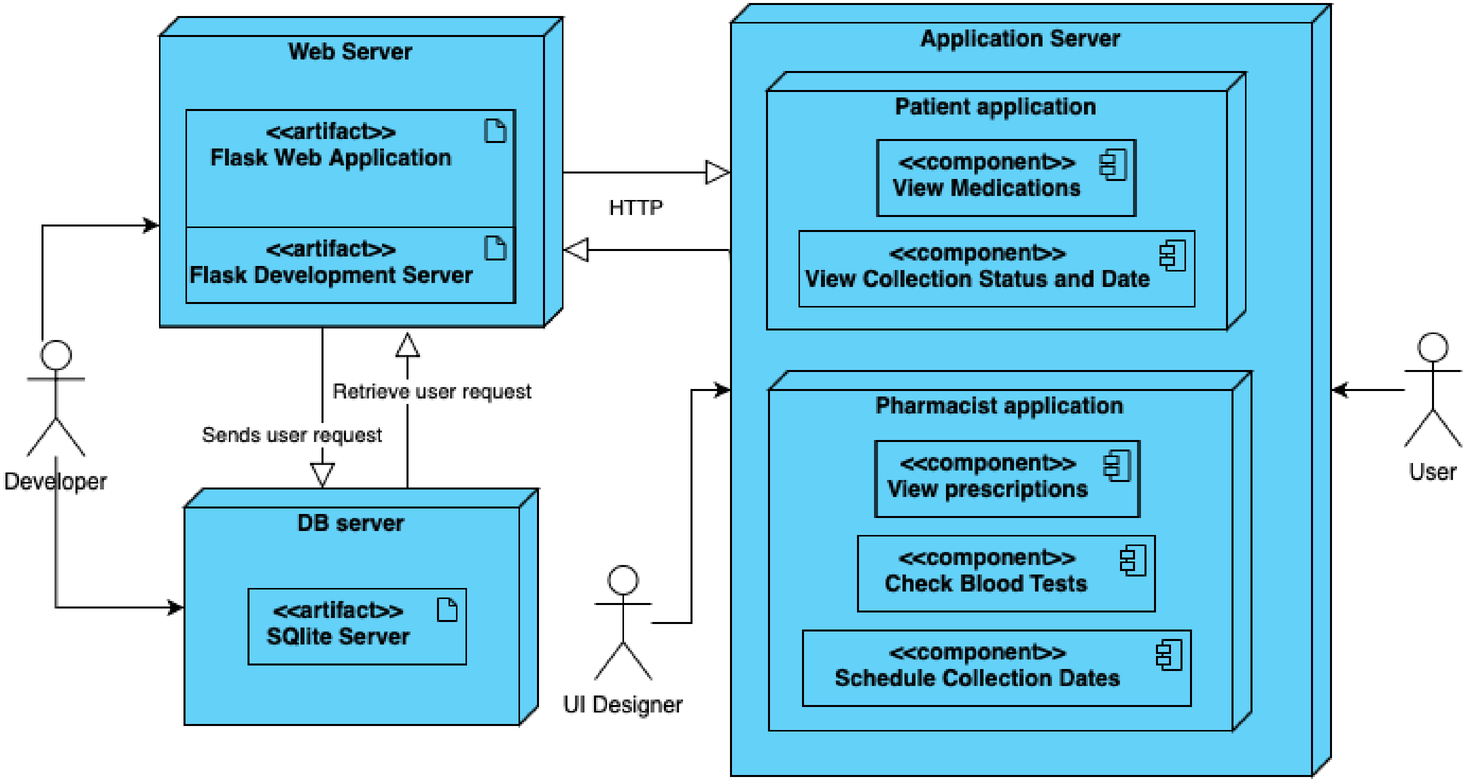
Appendix A:


Model
Defines, handles and interreacts with the database. Communicates with controller and retrieves data.
SQL Lite Database
View
Displays the user interface and communicates with the controller.
HTML/CSS/Bootstrap
Request Data
Request View
Get Data
Get View
Sends user inputs to the controller.
Controller
Controlls the flow of data between the model and view.
Flask (Python and SQLAlchemy)
The controller responds to the request of the device.
Devices Groups

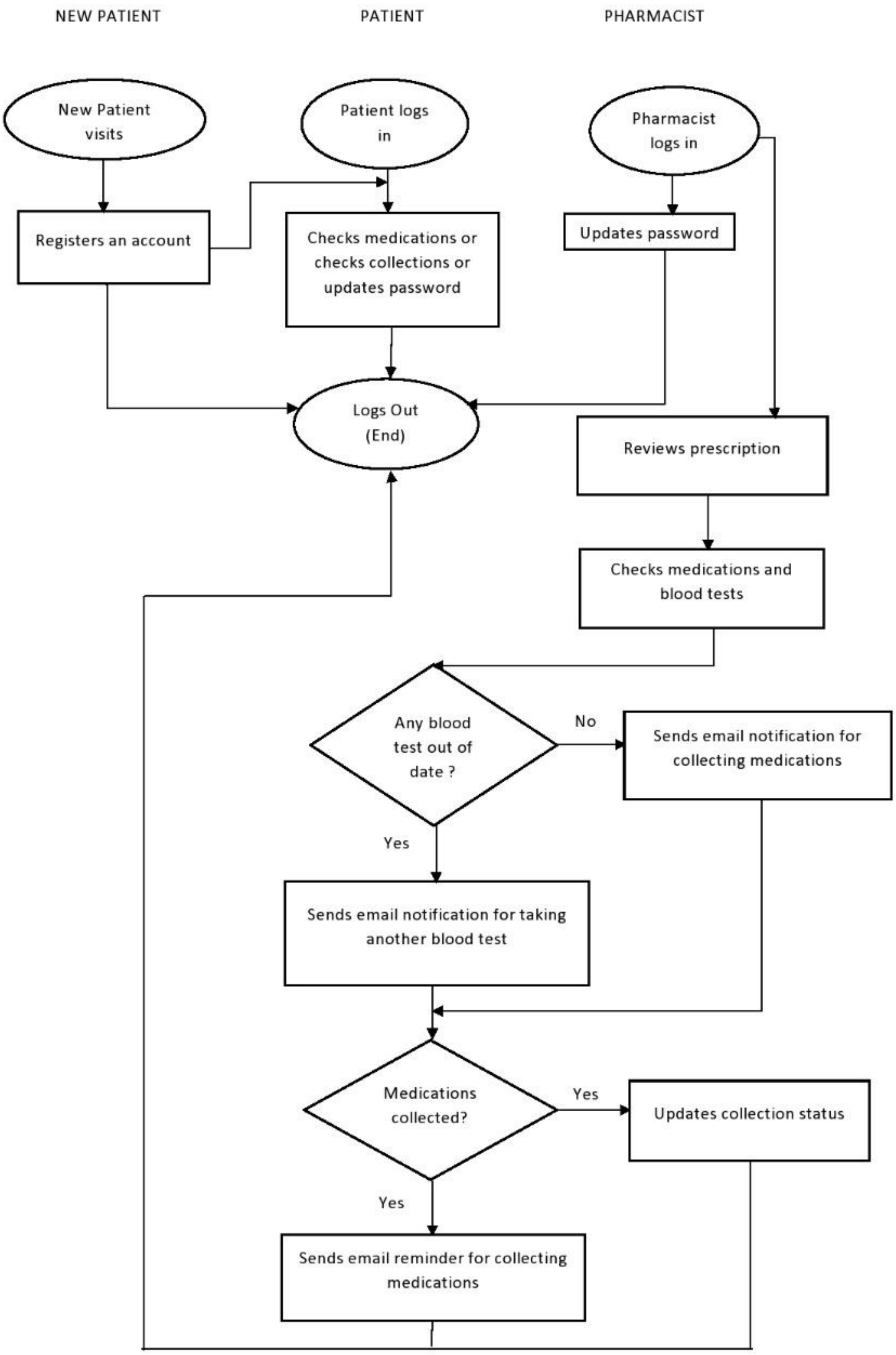
Appendix B


Event
Modify UI
Controller
Modify UI
View
Manipulate data
View
Model

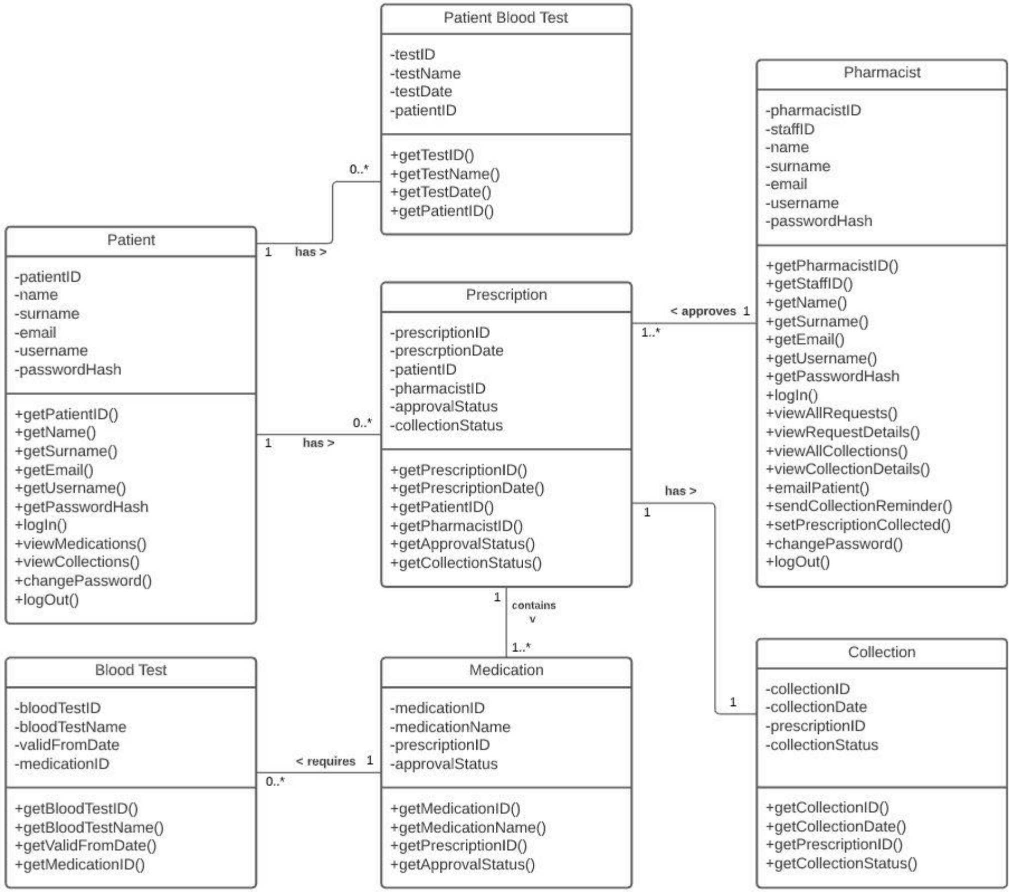
Appendix C


Web Server
<<artifact>>
Flask Web Application
<<artifact>>
Flask Development Server
HTTP
Application Server
Patient application
<<component>>
View Medications
<<component>>
View Collection Status and Date
Pharmacist application
<<component>>
View prescriptions
<<component>>
Check Blood Tests
<<component>>
Schedule Collection Dates
Retrieve user request
Sends user request
Developer
DB server
<<artifact>>
SQlite Server
UI Designer
User

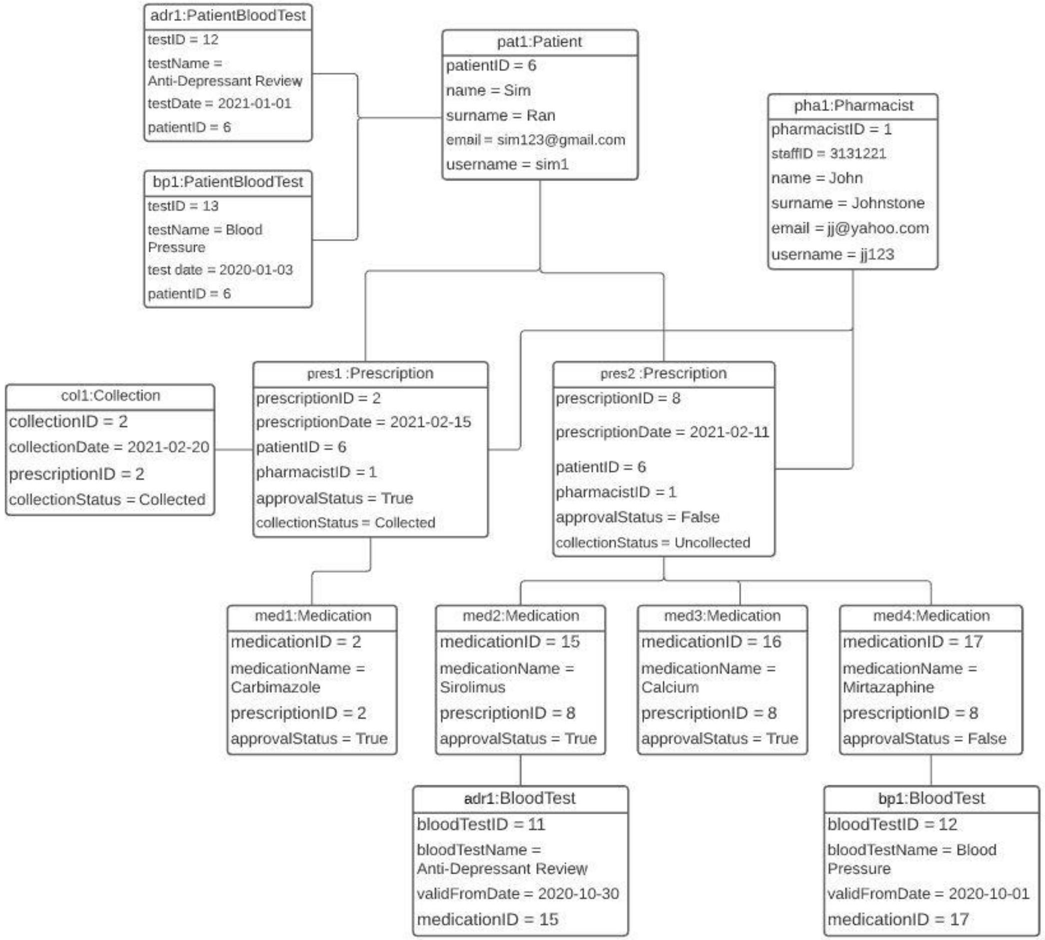
Appendix D


NEW PATIENT
PATIENT
PHARMACIST
New Patient visits
Patient logs in
Pharmacist logs in
Registers an account
Checks medications or checks collections or updates password
Updates password
Logs Out (End)
Reviews prescription
Checks medications and blood tests
Any blood test out of date ?
No
Sends email notification for collecting medications
Yes
Sends email notification for taking another blood test
Medications collected?
Yes
Updates collection status
Yes
Sends email reminder for collecting medications

Appendix E


Patient Blood Test
-testID
-testName
-testDate
-patientID
+getTestID()
+getTestName()
+getTestDate()
+getPatientID()
Pharmacist
-pharmacistID
-staffID
-name
-surname
-email
-username
-passwordHash
+getPharmacistID()
+getStaffID()
+getName()
+getSurname()
+getEmail()
+getUsername()
+getPasswordHash
+logIn()
+viewAllRequests()
+viewRequestDetails()
+viewAllCollections()
+viewCollectionDetails()
+emailPatient()
+sendCollectionReminder()
+setPrescriptionCollected()
+changePassword()
+logOut()
0..*
1
has >
Patient
-patientID
-name
-surname
-email
-username
-passwordHash
+getPatientID()
+getName()
+getSurname()
+getEmail()
+getUsername()
+getPasswordHash
+logIn()
+viewMedications()
+viewCollections()
+changePassword()
+logOut()
Prescription
-prescriptionID
-prescrptionDate
-patientID
-pharmacistID
-approvalStatus
-collectionStatus
+getPrescriptionID()
+getPrescriptionDate()
+getPatientID()
+getPharmacistID()
+getApprovalStatus()
+getCollectionStatus()
< approves 1
1..*
0..*
1
has >
has >
1
1
contains
v
1..*
Blood Test
-bloodTestID
-bloodTestName
-validFromDate
-medicationID
+getBloodTestID()
+getBloodTestName()
+getValidFromDate()
+getMedicationID()
Medication
-medicationID
-medicationName
-prescriptionID
-approvalStatus
+getMedicationID()
+getMedicationName()
+getPrescriptionID()
+getApprovalStatus()
< requires 1
0..*
Collection
-collectionID
-collectionDate
-prescriptionID
-collectionStatus
+getCollectionID()
+getCollectionDate()
+getPrescriptionID()
+getCollectionStatus()
1

Appendix F


adr1:PatientBloodTest
testID = 12
testName = Anti-Depressant Review
testDate = 2021-01-01
patientID = 6
bp1:PatientBloodTest
testID = 13
testName = Blood Pressure
test date = 2020-01-03
patientID = 6
pat1:Patient
patientID = 6
name = Sim
surname = Ran
email = sim123@gmail.com
username = sim1
pha1:Pharmacist
pharmacistID = 1
staffID = 3131221
name = John
surname = Johnstone
email = jj@yahoo.com
username = jj123
col1:Collection
collectionID = 2
collectionDate = 2021-02-20
prescriptionID = 2
collectionStatus = Collected
pres1 :Prescription
prescriptionID = 2
prescriptionDate = 2021-02-15
patientID = 6
pharmacistID = 1
approvalStatus = True
collectionStatus = Collected
pres2 :Prescription
prescriptionID = 8
prescriptionDate = 2021-02-11
patientID = 6
pharmacistID = 1
approvalStatus = False
collectionStatus = Uncollected
med1:Medication
medicationID = 2
medicationName = Carbimazole
prescriptionID = 2
approvalStatus = True
med2:Medication
medicationID = 15
medicationName = Sirolimus
prescriptionID = 8
approvalStatus = True
med3:Medication
medicationID = 16
medicationName = Calcium
prescriptionID = 8
approvalStatus = True
med4:Medication
medicationID = 17
medicationName = Mirtazaphine
prescriptionID = 8
approvalStatus = False
adr1:BloodTest
bloodTestID = 11
bloodTestName = Anti-Depressant Review
validFromDate = 2020-10-30
medicationID = 15
bp1:BloodTest
bloodTestID = 12
bloodTestName = Blood Pressure
validFromDate = 2020-10-01
medicationID = 17