كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

جامعة حلوان
Helwan University

HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Information Systems Department

[BCD (Breast Cancer Detection)]

A graduation project dissertation by:

[Radwa Ahmed Abdullah Saleh (20170221)]

[Bassant Hany Omar Omar (20170154)]

[Toka EL-Bazidy Mohamed Ahmed (20170164)]

[Abdelrahman Hazem Wahdan (20170284)]

[Mohamed Raafat Mohamed Abdelwahab (20170438)]

[Mahmoud Mosaad Mohamed Moussa (20170503)]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the Information Systems Department, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

[Dr. Ahmed El-Sayed]

July 2021

Supervised By: Dr. Ahmed El-Sayed

Team Members:

Bassant Hany

Toka El-Bazidy

Radwa Ahmed

Mohamed Raafat

Mahmoud Mosaad

Abdelrahman Hazem

Acknowledgement

We would like to express our great gratitude and immense pleasure at having the opportunity to work with such a dedicated person as **Dr. Ahmed El-Sayed.** It was only possible for this project to see the light thanks to his continuous guidance and on-point advice times and times throughout the year.

We would also like to thank our friends, families, for their encouragement, patience, and assistance over the years. Especially our parents as we are forever indebted to them, who have always done their best for us. Finally, for our faculty for providing the suitable environment that leaded us to represent the best image that computer science graduates of Helwan University are supposed to represent.

Abstract

Now days, breast cancer is the most frequently diagnosed life-threatening cancer in women and the leading cause of cancer death among women. In 2020, there were 2.3 million women diagnosed with breast cancer and 685 000 deaths globally. Unfortunately, it's very common so our project aims to help women by either reminding them to their monthly self-examination for early detection or helping women who already has suspicions about having cancer by detecting the result of their test and making their journey with treatments and dealing with cancer a little bit easier.

Contents

3. Software Design

4.Testing

5. Implementation

6. Results and Conclusion

Chapter One

In this chapter, we are going to discuss and go deeper in the overview of the project and know more about its scope and limitations, and explain some terminologies we will find throughout the document.

Chapter Headlines:

1. Overview
2. Purpose
3. Objectives
4. Product Scope
5. Development Approach & Methodology
6. Constrains

The application *"BCD"* was developed using **flutter**, **Real-time firebase**, and **Python** and the final product is a flutter android application where the features will be showcased.

1.1 Overview:

- This application reminds the user to her self-examination every month.
- It can predict whether the cancer is **benign** or **malignant** based on blood test results.
- It can help the user to set-up a calendar to remind her with her doctor appointments, chemotherapy sessions and her medications.
- It offers community support where other users can share their stories about their journey and how they're dealing or share how their healing stories to inspire others.

1.2 Purpose:

The main target of this Document is to describe our solution in dealing with breast cancer that we will develop. The documentation gives a detailed description of the both functional and non-functional requirements in details and describes the hardware and software interface requirements.

1.3 Objectives:

- Save time: by entering your test results the application can predict if your tumor is benign or malignant
- Raise Awareness: Most women forget to do their monthly check-ups even though they can be lifesaving as detecting if there is a tumor early can help with treatment and increase chances of healing
- Support Patients: Having cancer have its own mental effects as well as physical, knowing that someone is going through the same thing as you can help patients and reading other people's survival stories can inspire them and give them hope

1.4 Product Scope:

– **Planning:**

- Getting the appropriate dataset and studying it
- Determining the functional and non-functional requirements.
- Setting a Gantt chart for the project.
- Determining the resources of the team.
- Training the team.
- Prepare a UI\UX prototype.

- Designing: *Determining the diagrams to be needed like:*

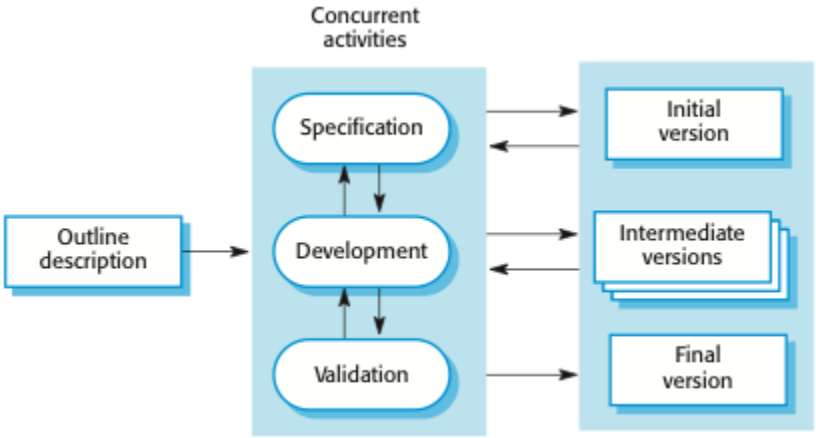
- ✓ Use case diagram
- ✓ Class diagram
- ✓ System Architecture diagram
- ✓ Activity diagram
- ✓ Sequence diagram
- ✓ ERD

- Coding and Testing functions

- Documenting the process

1.5 Development Approach & Methodology:

We followed the iteration model to analyse, design, and implement our project.



1.6 Constrains:

- Scope: project outcome as defined in the contract.
- Risks: uncertainties associated with the project.
- Time: deadline for delivering the output.
- Learning new technologies may take much time.
- Time management.
- Indiscipline “Human factor” like being late in delivering tasks or attending meetings.
- Having a smartphone, tablet, or an iOS device.
- Availability of stable Wi-Fi connection or mobile data.

Chapter TWO

In this chapter, we are going to discuss and go deeper in how we plan the project and show the steps and the instructions that we have followed to plan the website.

Chapter Headlines:

1. Project planning
2. Analysis and Limitation of existing system
3. Need for the new system
4. Analysis of the new system
5. Advantages of the new system
6. Risk and Risk Managements

2.1 Project Planning:

2.1.1 Feasibility Study

Problem Statement:

General Information

In 2020, there were 2.3 million women diagnosed with breast cancer and 685 000 deaths globally. As of the end of 2020, there were 7.8 million women alive who were diagnosed with breast cancer in the past 5 years, making it the world's most prevalent cancer. On average, every 2 minutes a woman is diagnosed with breast cancer and 1 woman will die of breast cancer every 13 minutes. Women needs to be aware of this disease and have support going through the difficult time of having it.

Project Importance:

Accessibility: Users will be able to access the application anytime and there will be no time constraints contrary to what usually happens.
Time Management: Users will be able to plan their appointments and sessions through the phone
Accuracy: The machine learning model used has accuracy of **98.25%** percent
Community support: User will share their stories to inspire and encourage others

Project Idea:

This application is about offering women a way to remind them to self-examine each month and for cancer patients to detect the type of cancer and help them to plan their new journey against this disease. As an application, it requires registration from users. Users can either choose to see a video of the self-examination or go to the prediction tab and enter their test results to determine the type of tumor. Also, users can provide their feedback or ask any questions. Also, they can share stories and see other people's stories.

Reasons for selecting this project:

- An effective way to remind women to do their self-examination as most people use their phones on daily basis.
- Saves time by checking the results directly on the phone
- Acts like a personal assistant to remind patients of their appointments and sessions

Marketing Information

What does the project offer?

The project offers a service downloaded through smart phones:

By downloading the application, you will be able to have access to either self-examine or enter your test results for prediction.

Who are the expected consumers and customers for the application?

● The customers will be women over 18 years old

Why would customers buy this service?

● The prediction is very accurate

● It serves all women not just cancer patients

● It would guarantee that users receive their reminders as most people use their smart phone daily.

Who are the most important competitors in this sector?

- Breast Cancer Healthline
- Brexa - Breast cancer screening
- Outcome4Me

Technical Information

What materials and equipment does the project need?

Hardware: A high functioning smartphone with Wi-Fi is all that is needed.

Software:

● Operating system: Android or IOS

● Programming language: Dart

● Database: Firebase

● Python

Who are the suppliers?

The application will be available on both Play store (Android) and App store (iOS)

Machine Learning:

1-The Dataset

- Using python in data cleaning phase (removing duplicates, filling in missing values, fixing data types) to use it's wide range of data cleaning and data visualization libraries. The dataset used in training the model is The Breast Cancer Wisconsin (Diagnostic) Data Set.
- Data cleaning with pandas library.
- data visualization with matplotlib and seaborn.
- data pre-processing with pandas

2- Implementing the Model:

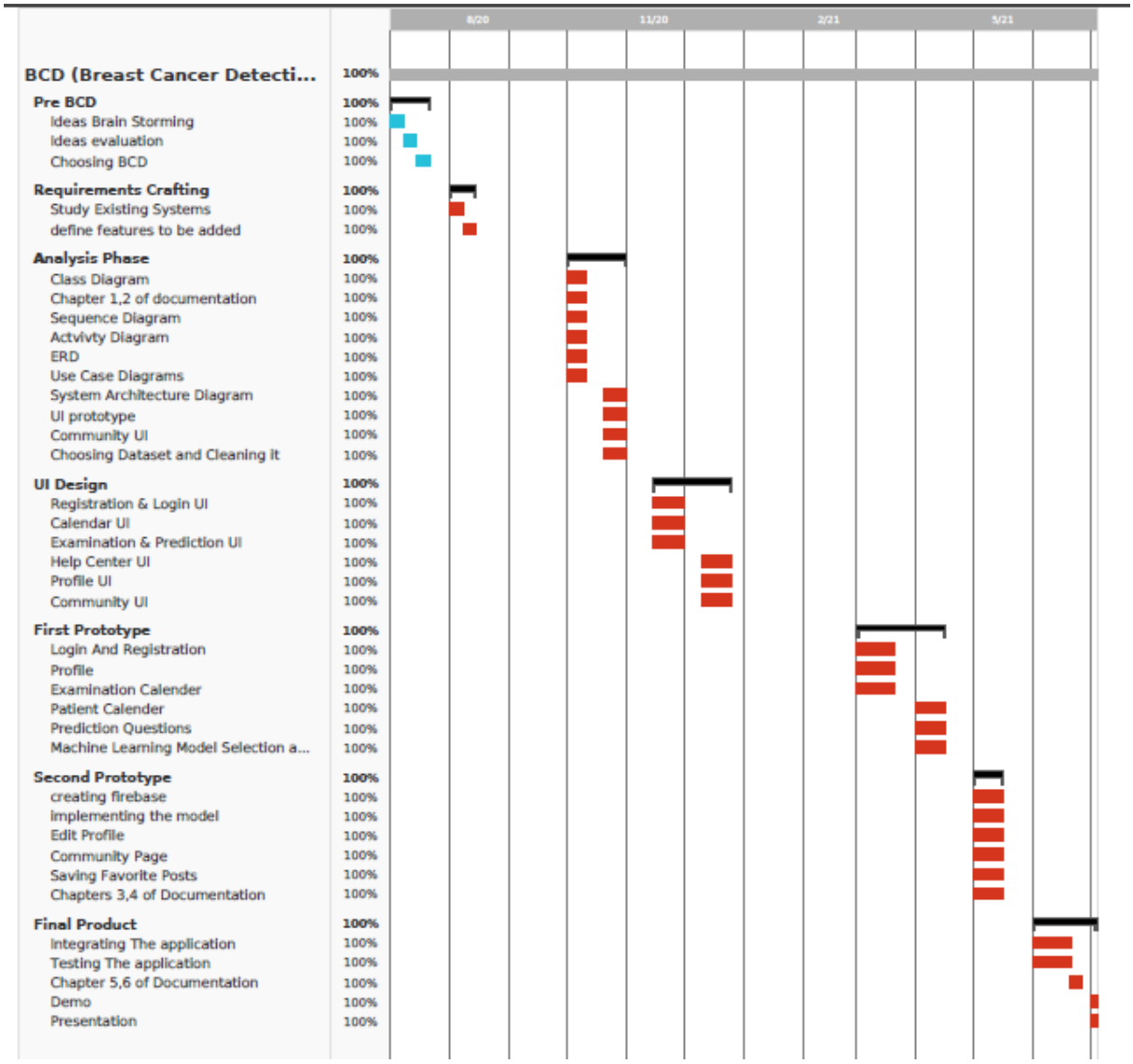
- using machine learning in flutter to enhance performance of the app.
- importing the data set in flutter using ml_dataframe package.
- data splitting in flutter using ml_preprocessing package.
- choosing machine learning instead of deep learning because the relatively small size of the data set so we can prevent overfitting.
- using machine learning in flutter using ml_algo package.
- trying logistic regression , SVM , K Nearest Neighbors , Random Forests classification algorithms.
- choose logistic regression because it achieved the highest accuracy on test data. With **98.25 % accuracy.**

What is logistic regression?

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a *logit*, from *logistic unit*, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a *constant* rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

2.1.2 Estimated Cost: The application is completely for free and needs no specific budget.

2.1.3 Gannt Chart:



2.2 Analysis & Limitations of Existing Systems

- Users would still need to go to doctors to determine the treatment plan if the cancer turns out to be malignant.
- It doesn't support other types of cancer
- It doesn't support communication between users

2.3 Need for New System

As the number of breast cancer patients is increasing some companies released applications to help them in various ways like reminding them to self-examine or plan their journey or predict type of cancer, but no applications have those three main features together there BCD helps the patients in all sort of possible ways.

2.4 Analysis of New System

2.4.1 User Requirements

Mandatory requirements:

1- the user can watch a video of self-examination and set a monthly reminder to remind her to repeat the examination

2- The system should be able to correctly predict type of cancer

3- Patient should be able to save all her appointments and sessions and have a reminder of them

3- System should be understandable by system users who don't have detailed technical knowledge, easy to use and doesn't violate budget

4- The user gets to generate technical reports.

Optional requirements:

1-The user can see a timeline of posts and favorite them.

2- the user can post stories of his own

2-The user can send his feedback or questions

3-The user can edit his profile

Desirable requirements:

1- Users should update their posts

2- The user uses the app with him at any place and any time so our app should be available in any place and any time.

2.4.2 System Requirements:

• Android & iOS

Since our app's code would be written with Flutter\ Dart it'll work on both android & IOS's all versions.

- **Internet connection**

Since we would need to keep the user's data in the cloud, we need an internet connection to store and retrieve the data.

• Storage space

In order to keep the user's data in addition to the application itself, we need a fine storage.

2.4.3 Domain Requirements:

1- Multiple users must be able to use the application simultaneously without corrupting the database (whatever form it may be).

2- Enough memory should be reserved for on the database server to accommodate for any number of users and their data.

3- The necessary software required to run the application.

4- Each user must have a unique username/password combination.

5- The database should be backed up every occasionally in case the original does become corrupt.

6- The application must verify all values before making the change in the database.

7- The application must have update capabilities for future models and accessories.

2.4.4 Functional Requirements:

Register Function

Name	Register
Type	Requirement
Criticality	High
Input	Name / Email/ Password
Output	User gets registered and enters the application
Description	Function allows you to open an account for user on the system
Priority	10/10
Expected Risk	Wrong data entered
Pre-Condition	User downloading the application
Post-Condition	User enters the application

Login Function

Name	Login
Type	Requirement
Criticality	High
Input	Email / password
Output	User enters the application
Description	Function allows you to open an account for user on the system
Priority	10/10
Expected Risk	User login without registering before or using wrong data
Pre-Condition	Having an existing account
Post-Condition	User enters application

Logout Function

Name	Logout
Type	Requirement
Criticality	Medium
Input	Press Logout button
Output	User logs out
Description	Function allows to go to out the system
Priority	8/10
Expected Risk	User doesn’t press the button
Pre-Condition	User already logged in
Post-Condition	User logs out

Edit Profile Function

Name	Edit profile
Type	Requirement
Criticality	Medium
Input	Name, Email, Password
Output	User views his data and if modified it's saved to our database
Description	Function allows you to edit on your profile
Priority	7/10
Expected Risk	User enters invalid data types and submit it
Pre-Condition	User logged in
Post-Condition	User views his data and if modified it's saved to our database.

Examine Function

Name	Examine
Type	Requirement
Criticality	High
Input	User press the examination tab
Output	A video is displayed of how to do self-examination
Description	A function to help women to self-examine themselves at home
Priority	10/10
Expected Risk	Video not playing
Pre-Condition	User logged in
Post-Condition	User can go set monthly reminders

Setting Reminder Function

Name	Set reminder
Type	Requirement
Criticality	High
Input	User enters the examination day's date on the calendar
Output	The app will notify the user after 30 days to repeat the examination
Description	A function that allows the user to self-examination and remind them to repeat it every month
Priority	10/10
Expected Risk	Notification not appearing
Pre-Condition	User logged in
Post-Condition	A notification is set to appear in 30 days

Prediction Function

Name	Prediction
Type	Requirement
Criticality	High
Input	The values of blood test
Output	Type of cancer
Description	A function that allows the user to predict the type of cancer
Priority	10/10
Expected Risk	Error while entering the data
Pre-Condition	User logged in
Post-Condition	User can either plan their journey or set the monthly reminder

Add appointments/sessions date Function

Name	Add appointments/sessions date Function
Type	Requirement
Criticality	High
Input	Dates of doctor appointments Or chemotherapy/Radiation Sessions
Output	A notification to remind user with them
Description	A function to help users to plan their treatment journey
Priority	10/10
Expected Risk	Wrong date entered
Pre-Condition	User Logged in
Post-Condition	User gets a notification of the dates entered

Show Posts Function

Name	Show posts
Type	Requirement
Criticality	Medium
Input	User press the community button on the drawer
Output	Posts of Users is shown
Description	Function that allows users to see the posts/ stories of other patient sharing their experience
Priority	8/10
Expected Risk	Error fetching posts
Pre-Condition	User logged in
Post-Condition	User can favorite some posts

favorite Posts Function

Name	Favorite posts
Type	Requirement
Criticality	Medium
Input	Press the heart button in the post
Output	Post is added to favorite posts tab
Description	A function that allows user to select and filter posts by liking them
Priority	8/10
Expected Risk	Post not added to favorites screen
Pre-Condition	User Logged in
Post-Condition	Post can be shown in separate screen of favorite posts

Show favorite Posts Function

Name	Show favorite posts
Type	Requirement
Criticality	Medium
Input	Press Show Only Favorites button
Output	Only favorited posts are shown
Description	A function that allows user to display all his favorite posts
Priority	8/10
Expected Risk	Posts not loading
Pre-Condition	Posts are being favorited (liked)
Post-Condition	User can display all her favorite posts

Add Posts Function

Name	Add Post
Type	Requirement
Criticality	High
Input	User presses the add post button
Output	A post will be added to the community timeline
Description	A function that allows user to share their own story
Priority	10/10
Expected Risk	Post not added
Pre-Condition	User logged in
Post-Condition	Post is shown to all users

Edit/Delete Posts Function

Name	Edit/Delete Post
Type	Requirement
Criticality	High
Input	User presses the edit/delete button
Output	Post is edited/deleted
Description	A function that allows user to edit or delete her own posts
Priority	10/10
Expected Risk	Post not being edited/deleted
Pre-Condition	User Logged in Post is being posted
Post-Condition	User can continue browsing through posts

Add feedback Function

Name	Add Feedback
Type	Requirement
Criticality	Medium
Input	Name/Email/feedback
Output	Feedback is submitted
Description	A function that allows users to give their feedback or if they have question
Priority	8/10
Expected Risk	Feedback not submitted
Pre-Condition	User logged in
Post-Condition	Feedback

2.4.5 Non-Functional Requirements:

Usability:

Our application built to be easy in use. One of major things that user finds it friendly and easy to deal with.

Accuracy:

We want to be sure that's the results of predictions are correct.

Effectiveness:

Our application effects on:

Users having to go to a doctor to determine the type of cancer and can tell in the press of a button

Extensibility:

We aim to add more features to our system in the future, expand scope, in addition to more interactive dashboards.

2.5 Advantages of the new system

- It is an integration of more than one system.
- Facilitate communication between patients by allowing them to share their stories
- Doesn’t need to be used in a specific place.
- The system is easy to use by anyone.

2.6 Risk and Risk Management:

Poor password selection: As every new user must register using an e-mail and a password, it is mandatory that the password combination becomes a combination of upper-case letters, lower-case letters, numbers.

Un-experienced end users: Users may face difficulties in using the application. That will be managed be by making a very simple, easy to deal with interface

Merging two technologies as the application was created using flutter and the python, it is likely to face minor performance incidents which will be compensated for by the continuous update for the application in the future.

Bugs affecting the performance Due to the limited timeline and the inexperience of the team members with using the two technologies, bugs might cause the application to slow down or crash, thus the users will be dissatisfied. However, with the continuous updates and the cooperation between the team, bugs will be solved ASAP.

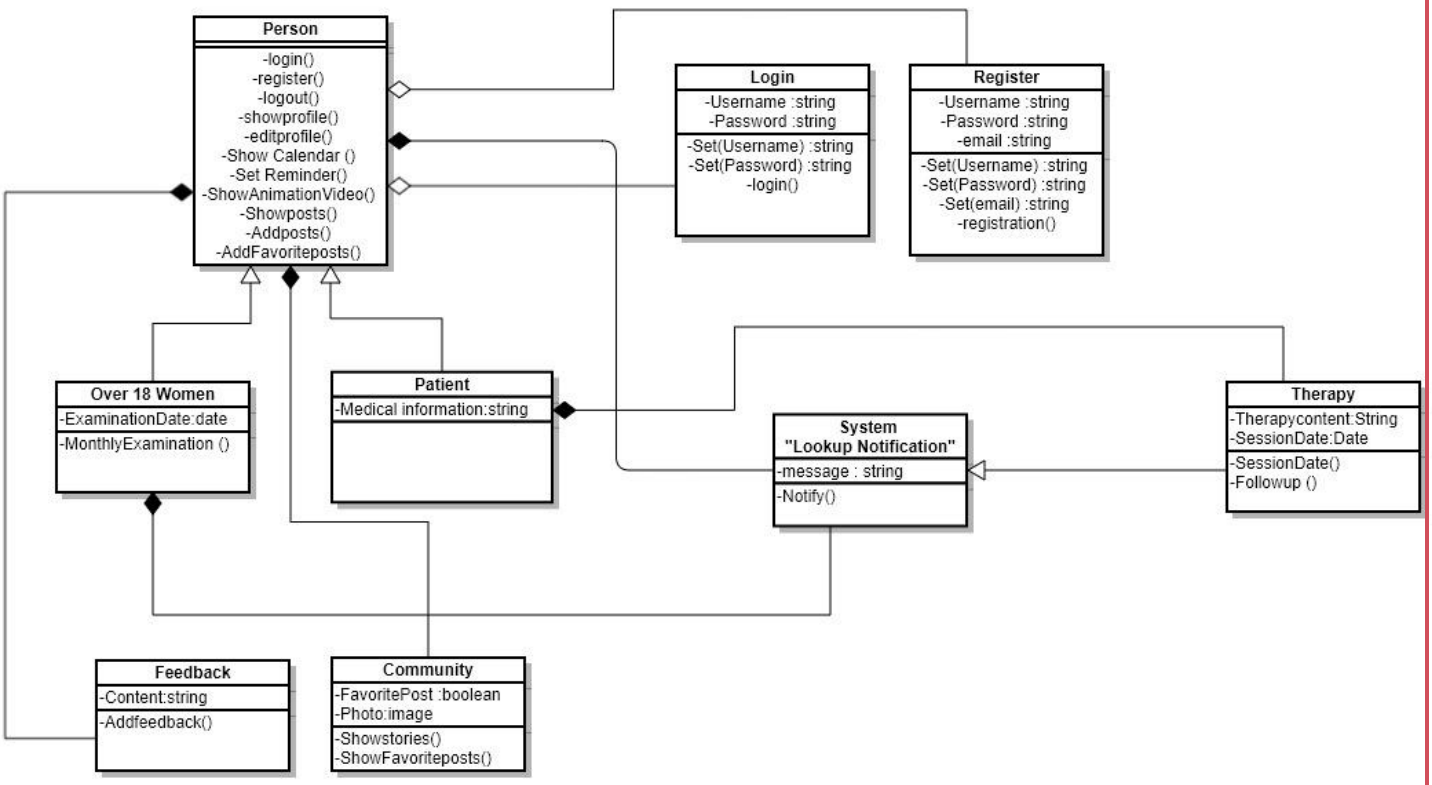
Chapter Three

In this chapter, we are going to discuss and go deeper in our application's design, and present its diagrams and database.

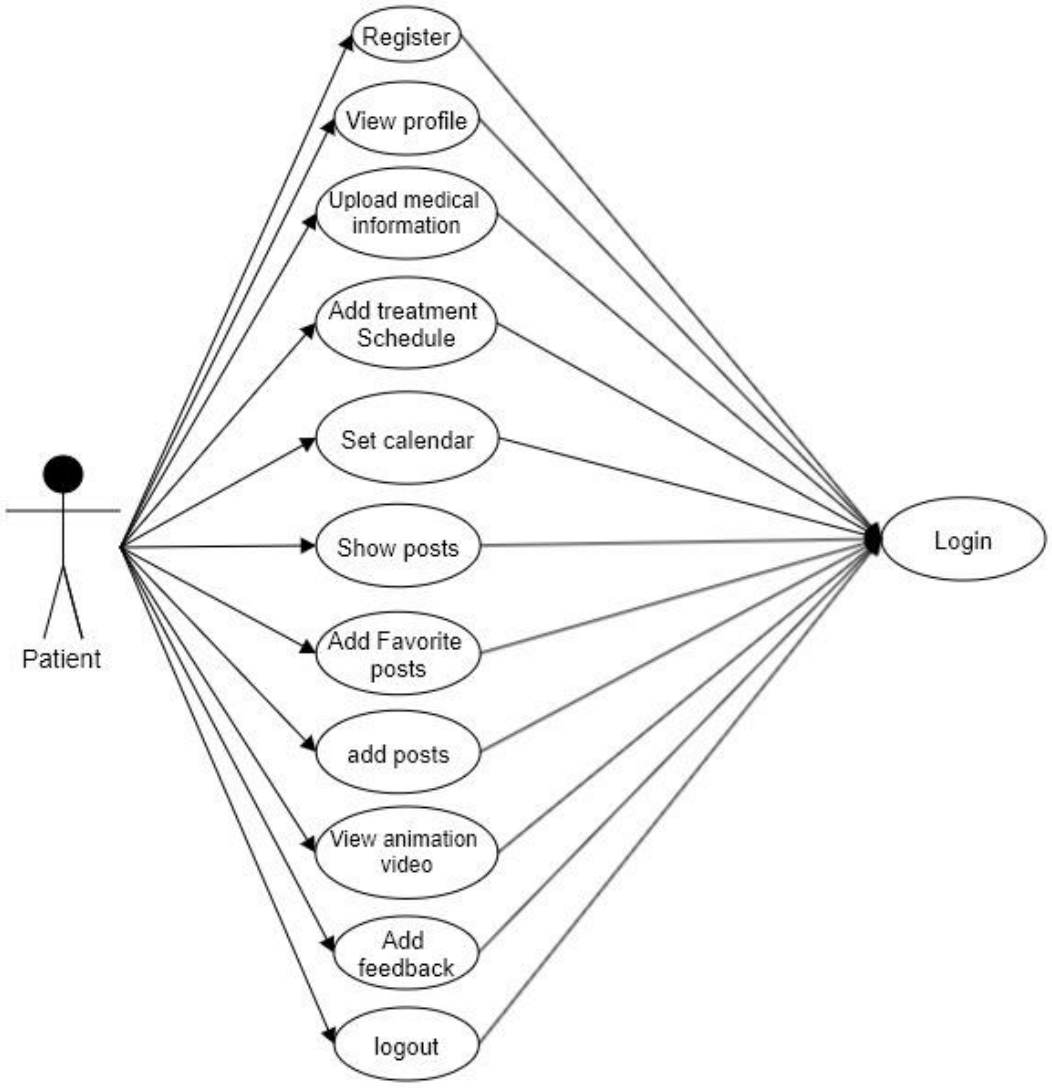
Chapter Headlines:

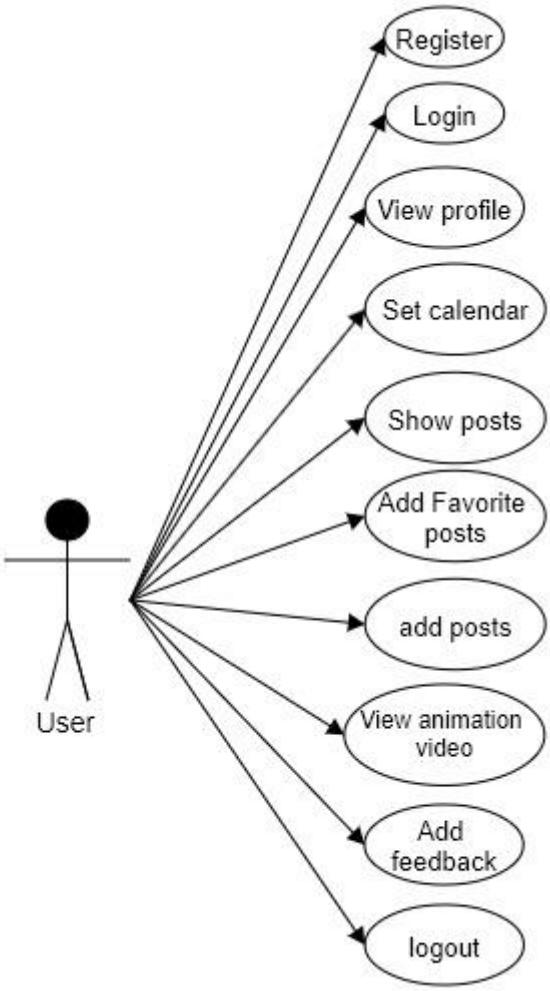
1. Class Diagram
2. Use Case Diagram
3. Activity Diagram
4. Sequence Diagram
5. ERD
6. System Architecture
7. Interface

3.1 Class diagram:

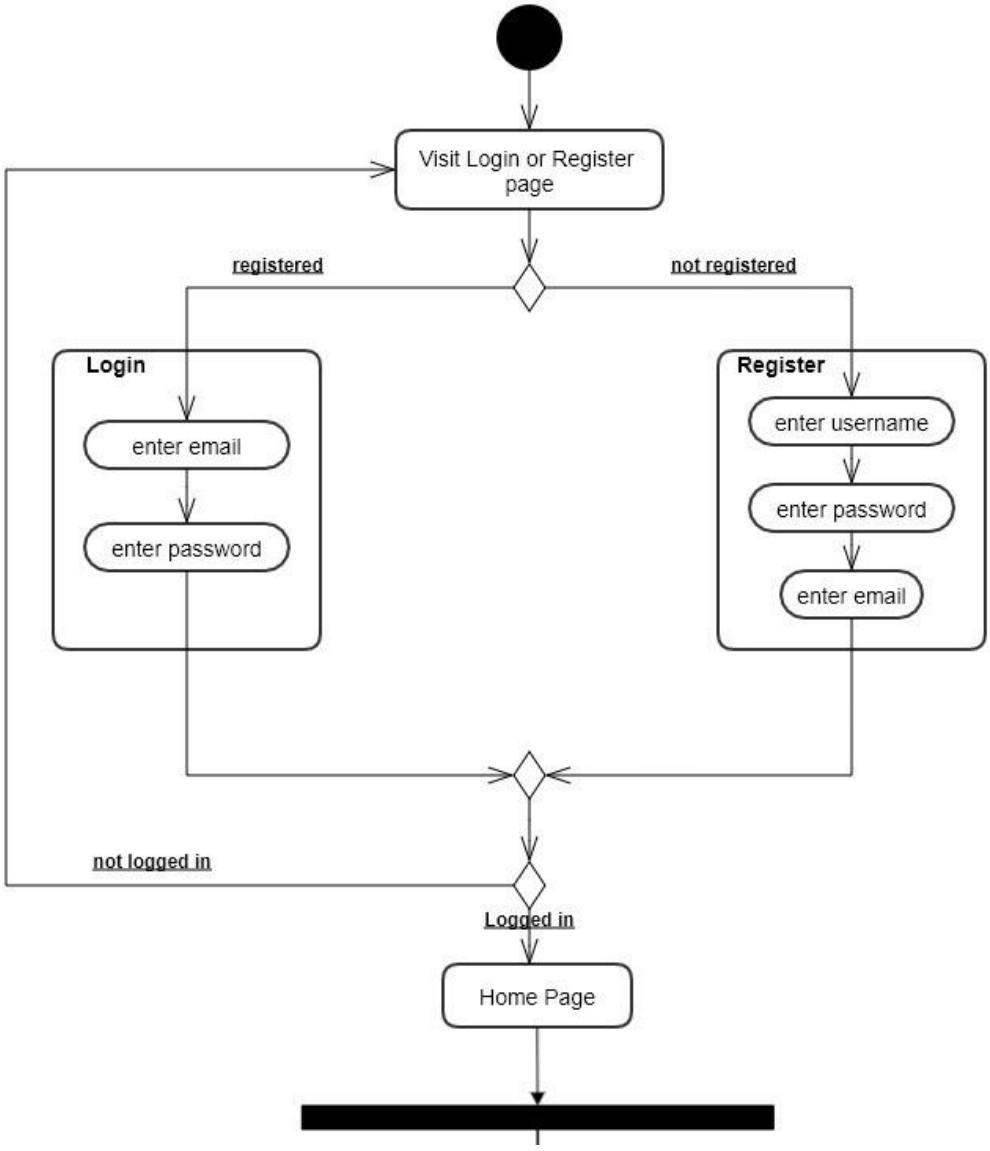


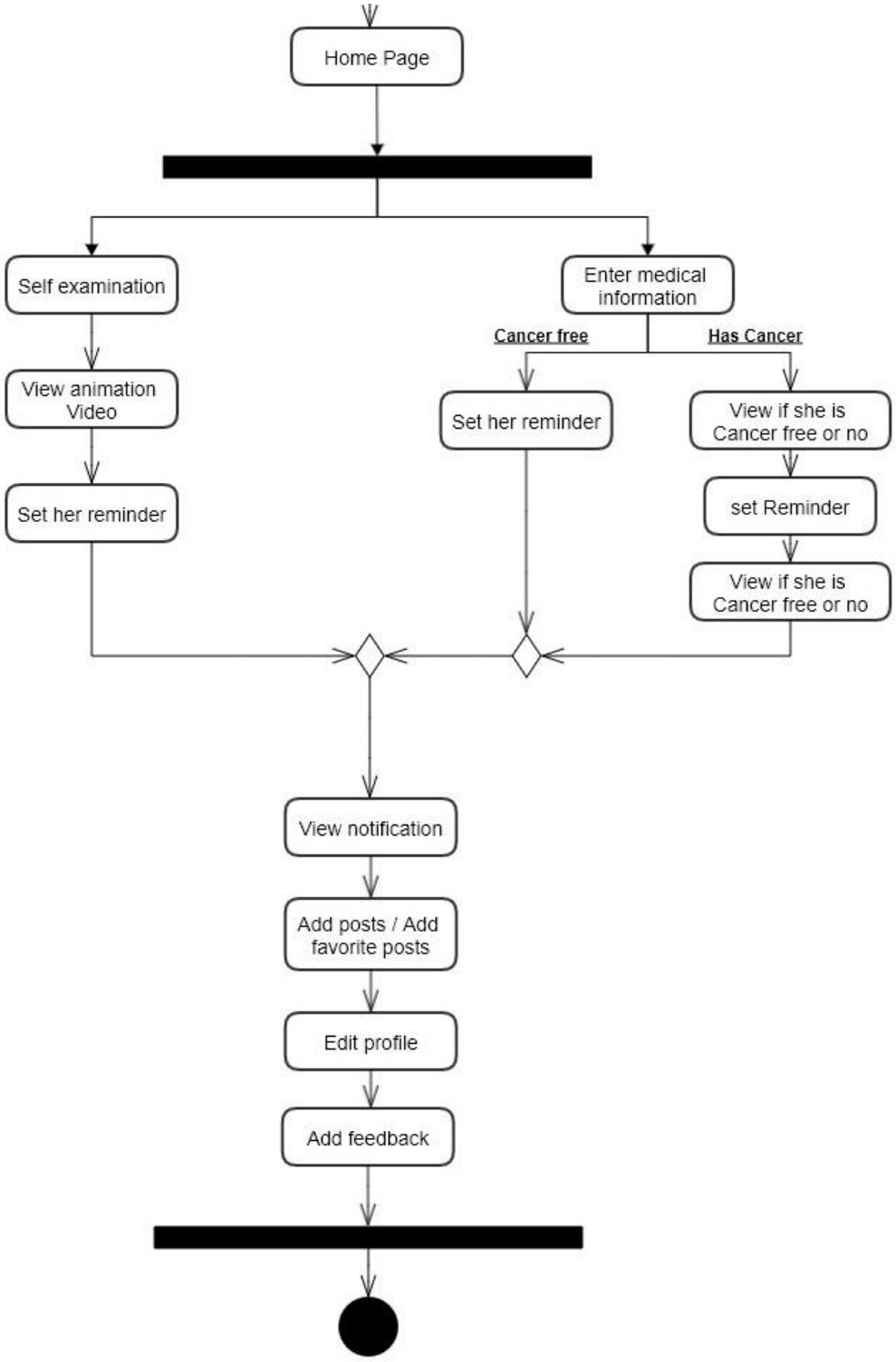
3.2 Use Case Diagrams:



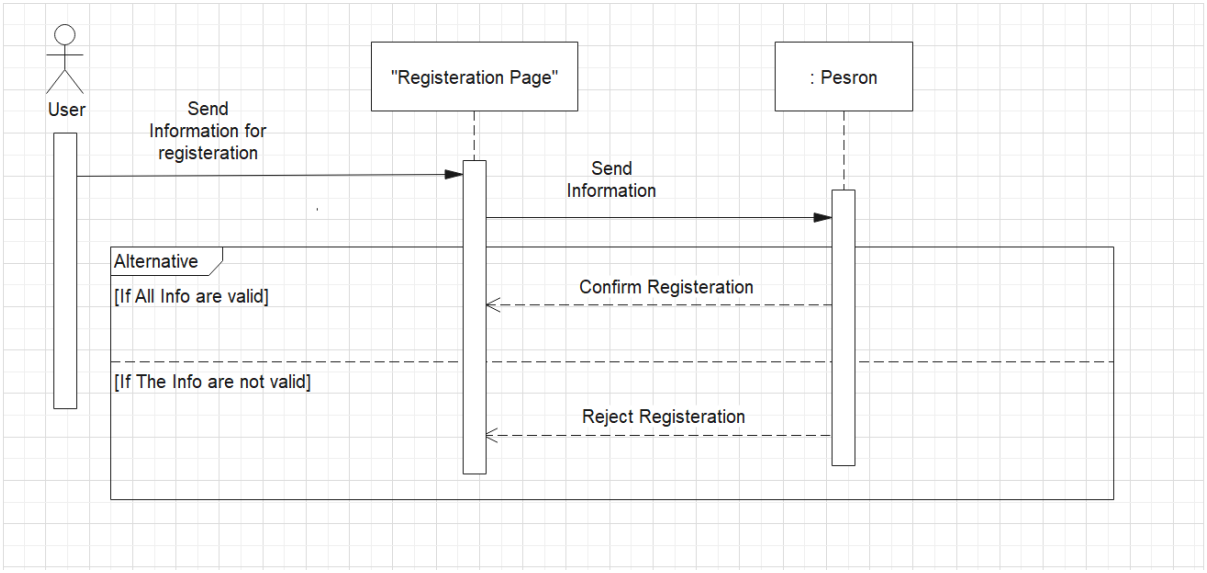

Register
Login
View profile
Set calendar
Show posts
Add Favorite posts
add posts
View animation video
Add feedback
logout
User

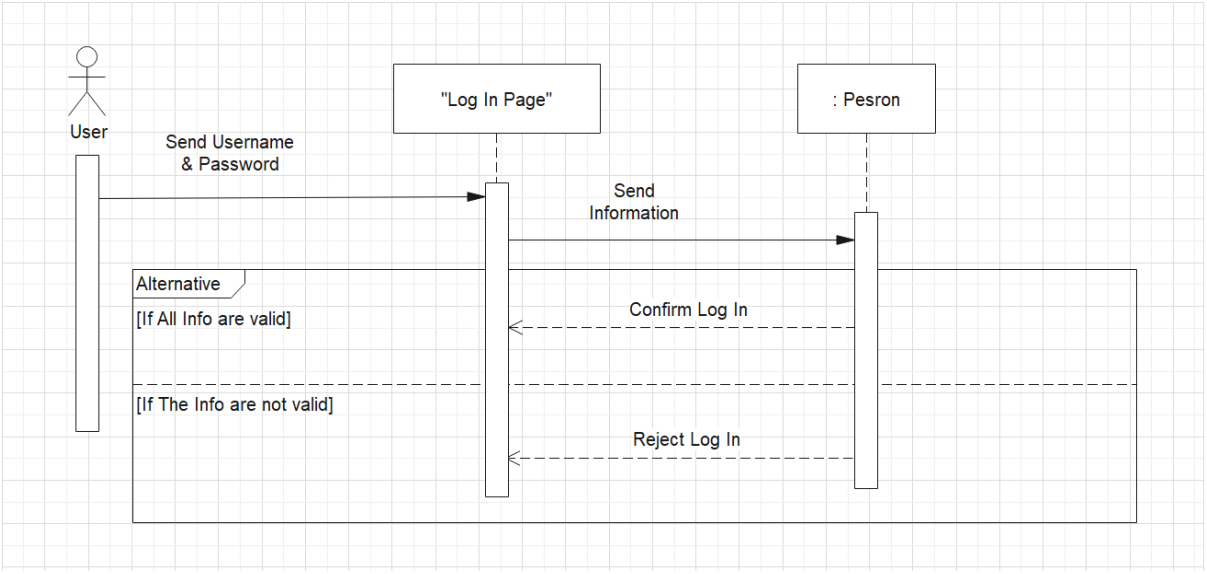
3.3 Activity Diagrams

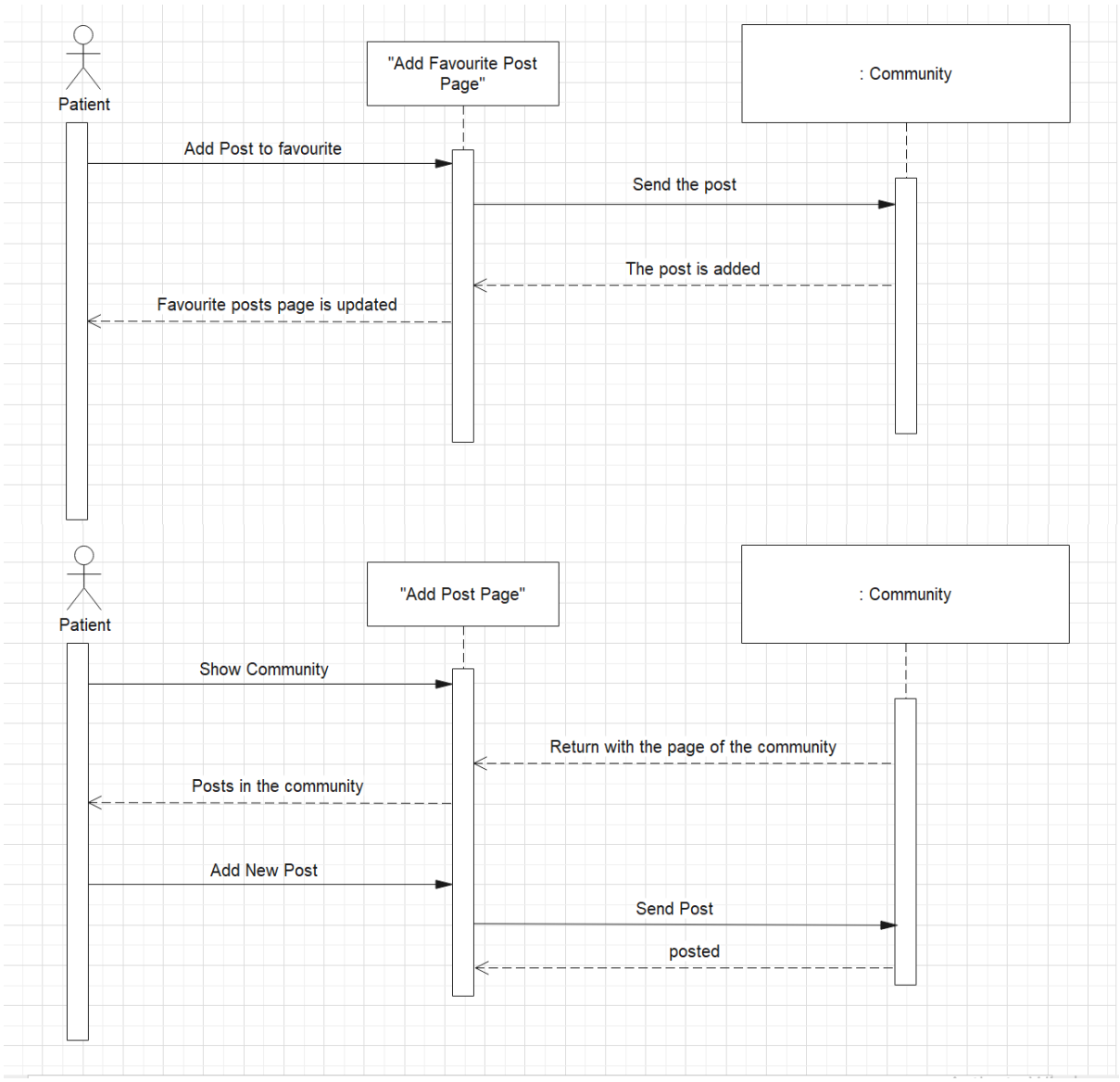


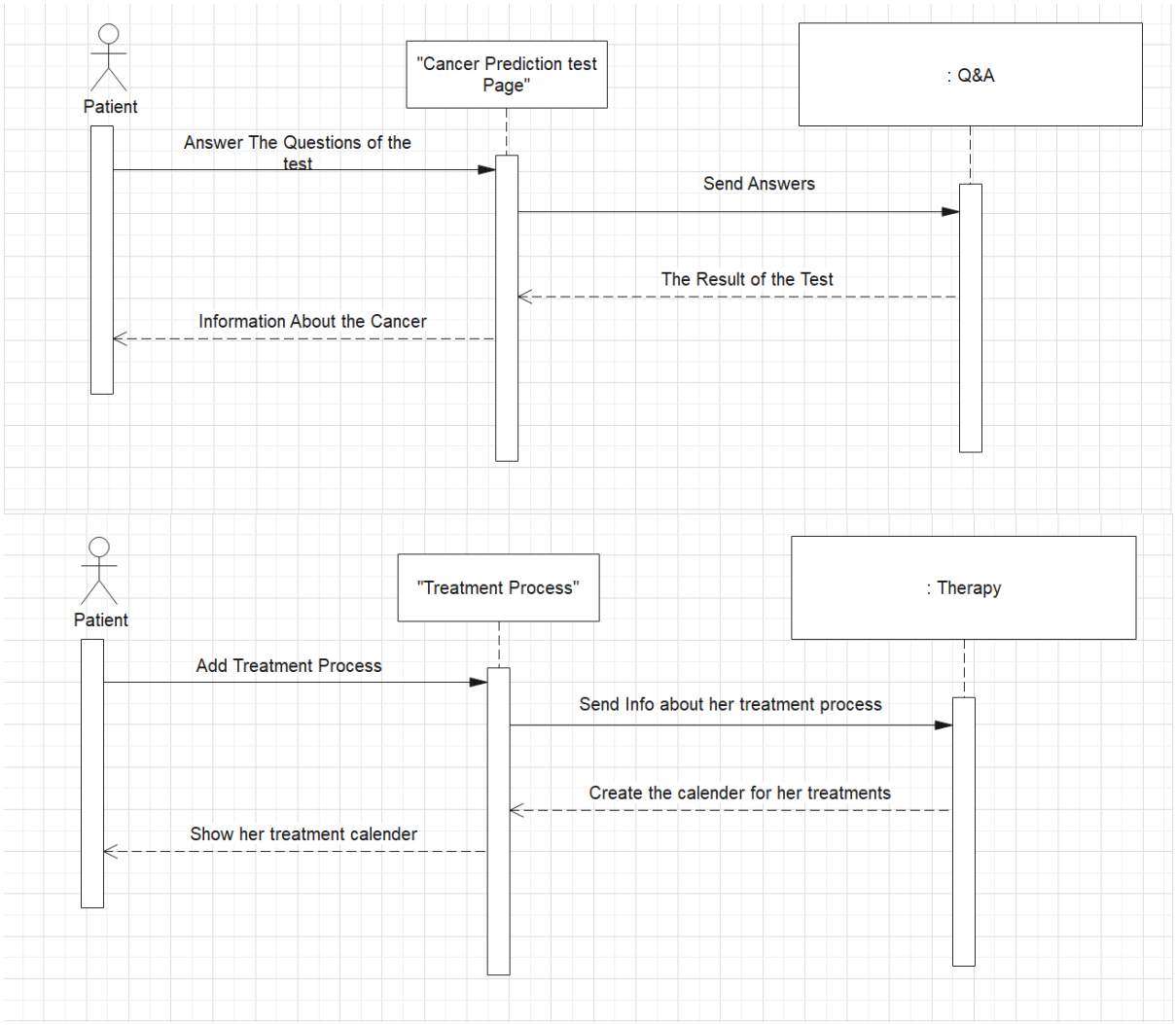

Home Page
Self examination
View animation Video
Set her reminder
Enter medical information
Cancer free
Has Cancer
Set her reminder
View if she is Cancer free or no
set Reminder
View if she is Cancer free or no
View notification
Add posts / Add favorite posts
Edit profile
Add feedback

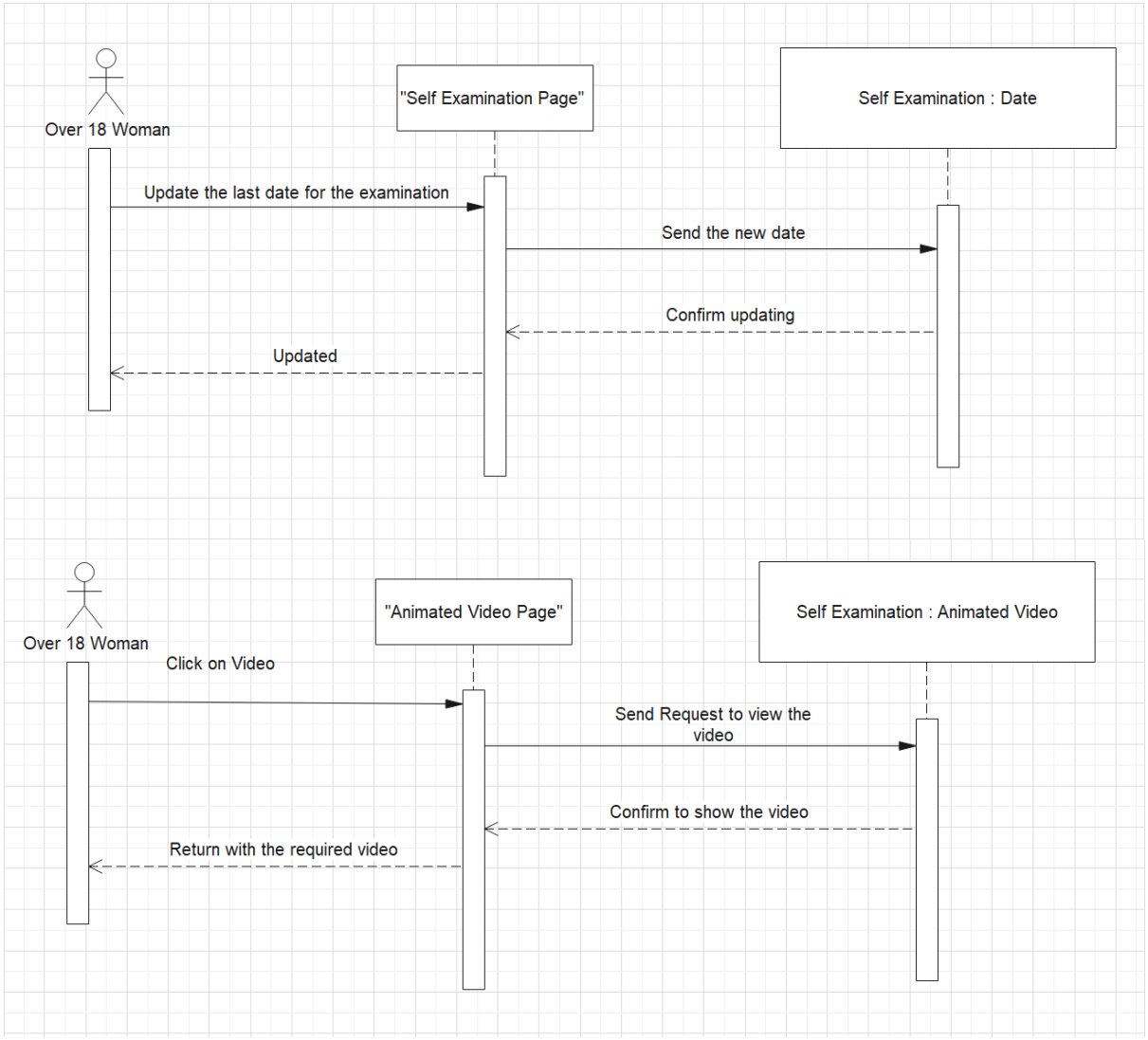
3.4 Sequence Diagrams

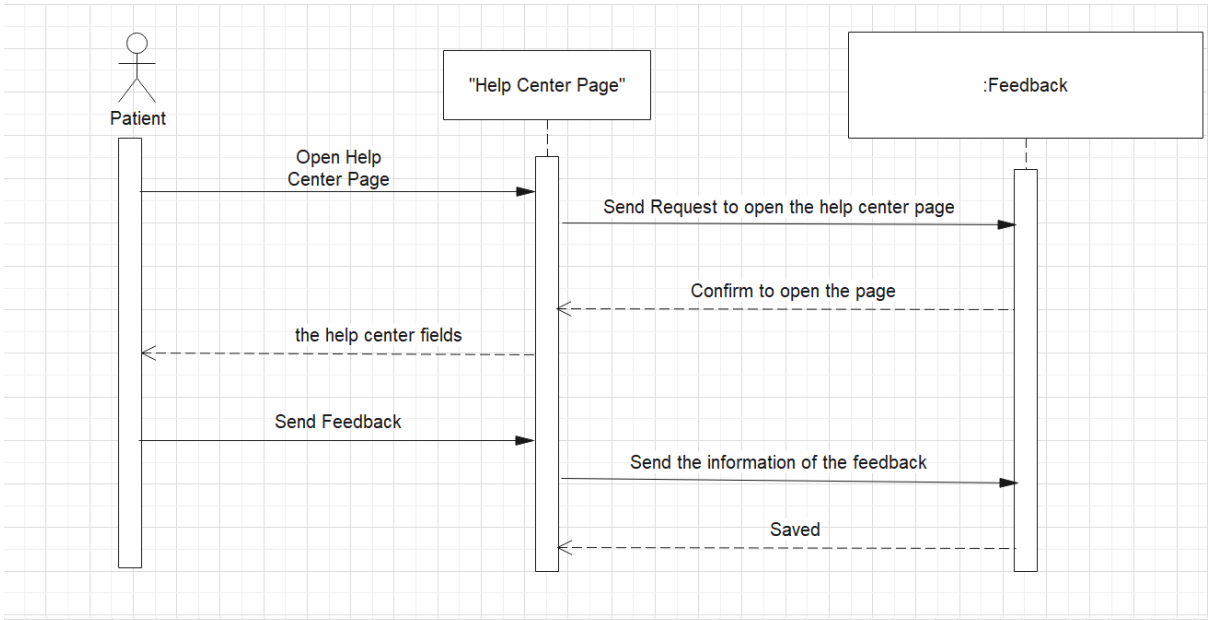


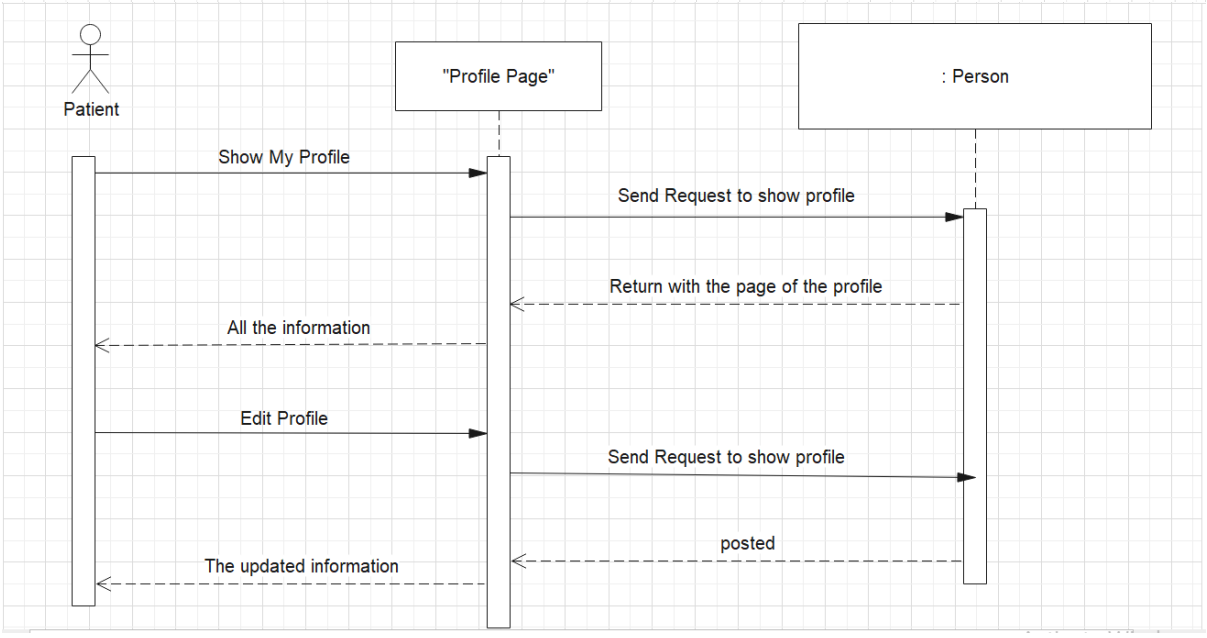


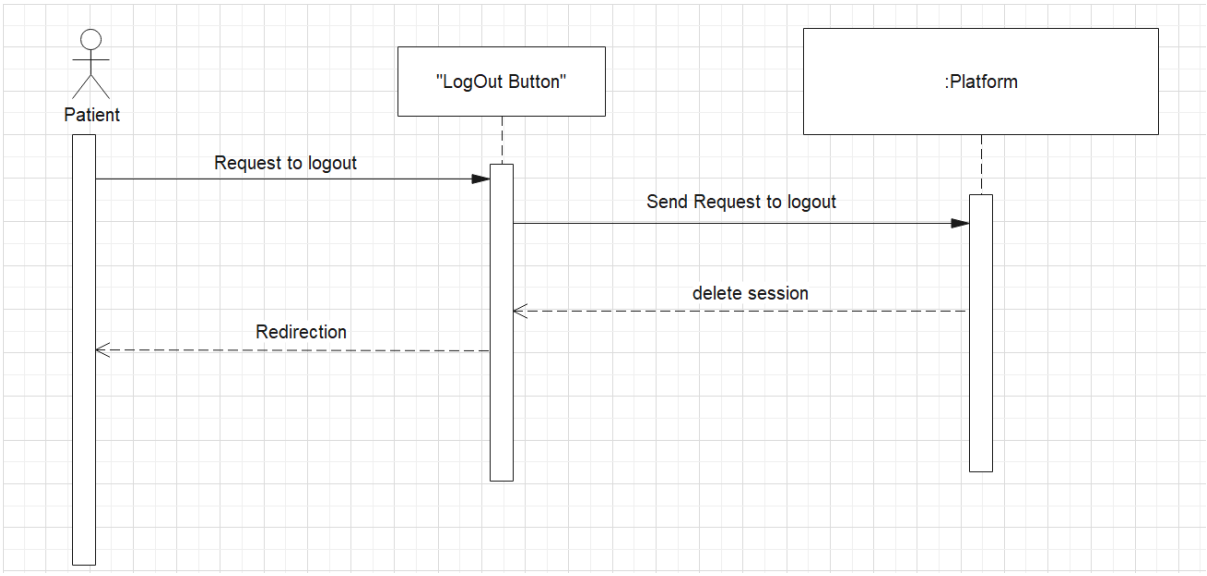

Patient
"Add Favourite Post Page"
: Community
Add Post to favourite
Send the post
The post is added
Favourite posts page is updated
Patient
"Add Post Page"
: Community
Show Community
Return with the page of the community
Posts in the community
Add New Post
Send Post
posted


Patient
"Cancer Prediction test Page"
: Q&A
Answer The Questions of the test
Send Answers
The Result of the Test
Information About the Cancer
Patient
"Treatment Process"
: Therapy
Add Treatment Process
Send Info about her treatment process
Create the calender for her treatments
Show her treatment calender

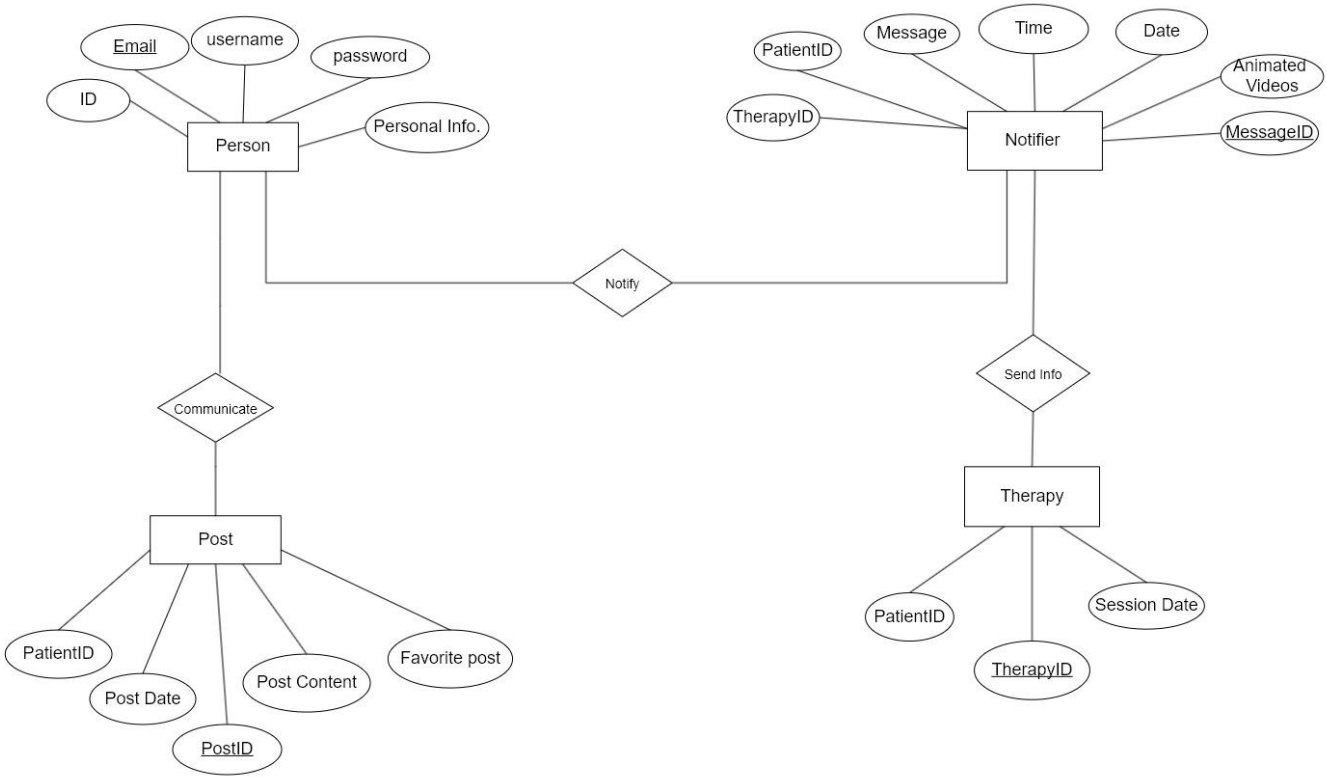

Over 18 Woman
"Self Examination Page"
Self Examination : Date
Update the last date for the examination
Send the new date
Confirm updating
Updated
Over 18 Woman
"Animated Video Page"
Self Examination : Animated Video
Click on Video
Send Request to view the video
Confirm to show the video
Return with the required video


Patient
"Help Center Page"
:Feedback
Open Help Center Page
Send Request to open the help center page
Confirm to open the page
the help center fields
Send Feedback
Send the information of the feedback
Saved

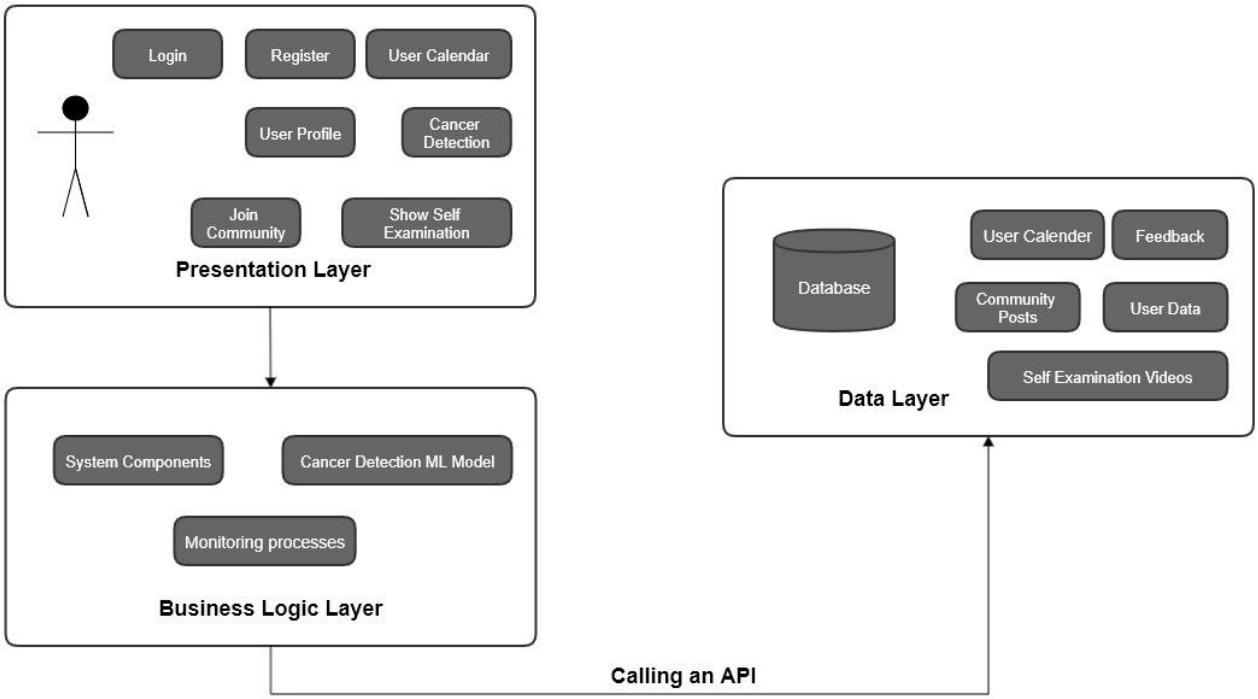

Patient
"Profile Page"
: Person
Show My Profile
Send Request to show profile
Return with the page of the profile
All the information
Edit Profile
Send Request to show profile
posted
The updated information


Patient
"LogOut Button"
:Platform
Request to logout
Send Request to logout
delete session
Redirection

3.5 ERD

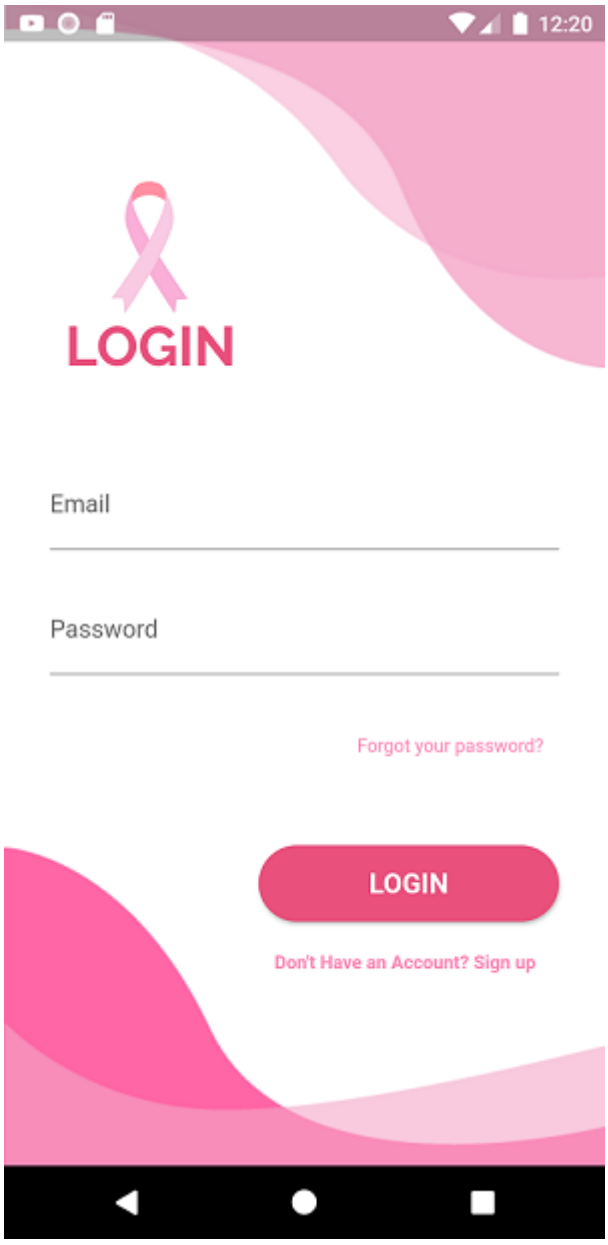

Email
username
password
ID
Person
Personal Info.
PatientID
Message
Time
Date
Animated Videos
TherapyID
Notifier
MessageID
Notify
Send Info
Communicate
Therapy
Post
PatientID
Post Date
Post Content
Favorite post
PostID
PatientID
Session Date
TherapyID

3.6 System Architecture

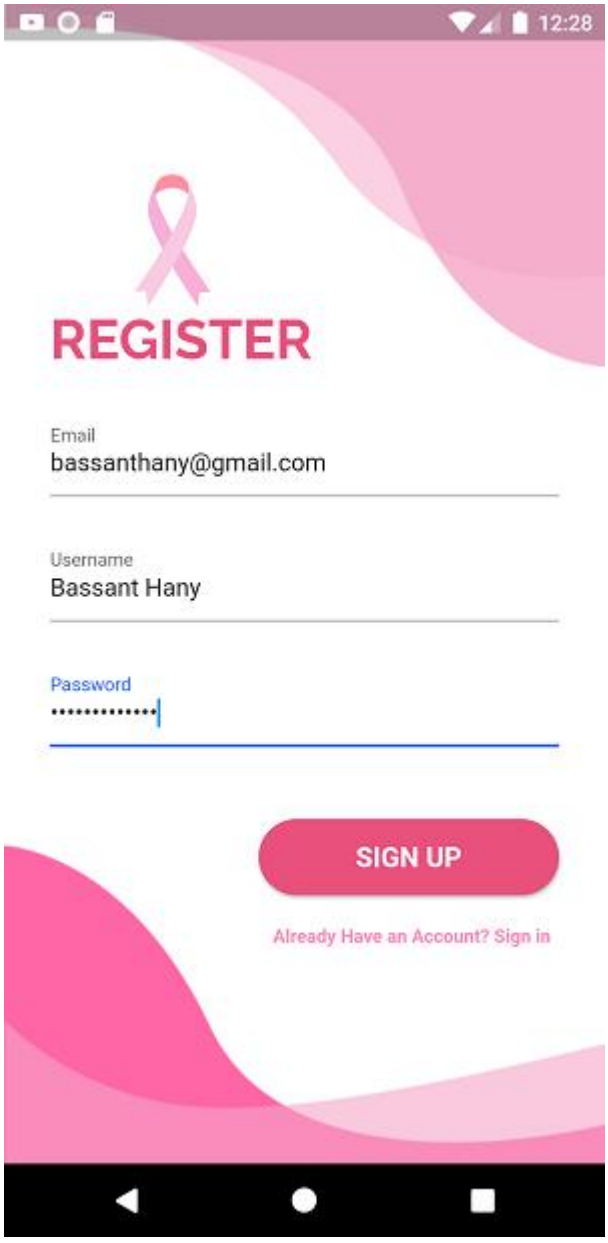


3.7 Interface

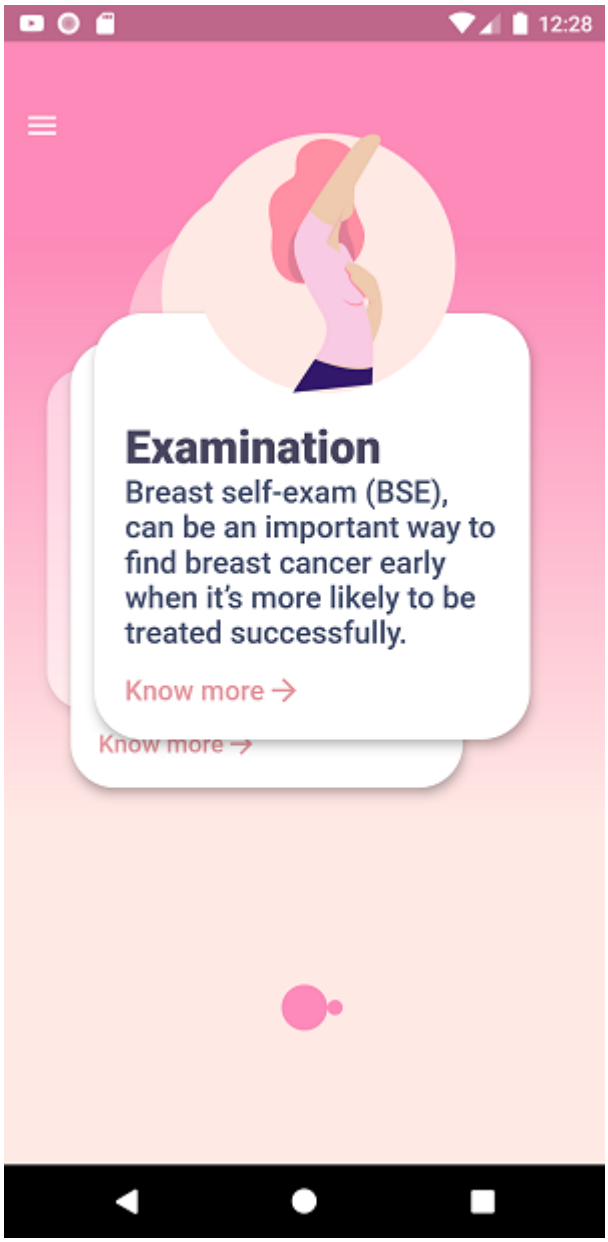
Login Page



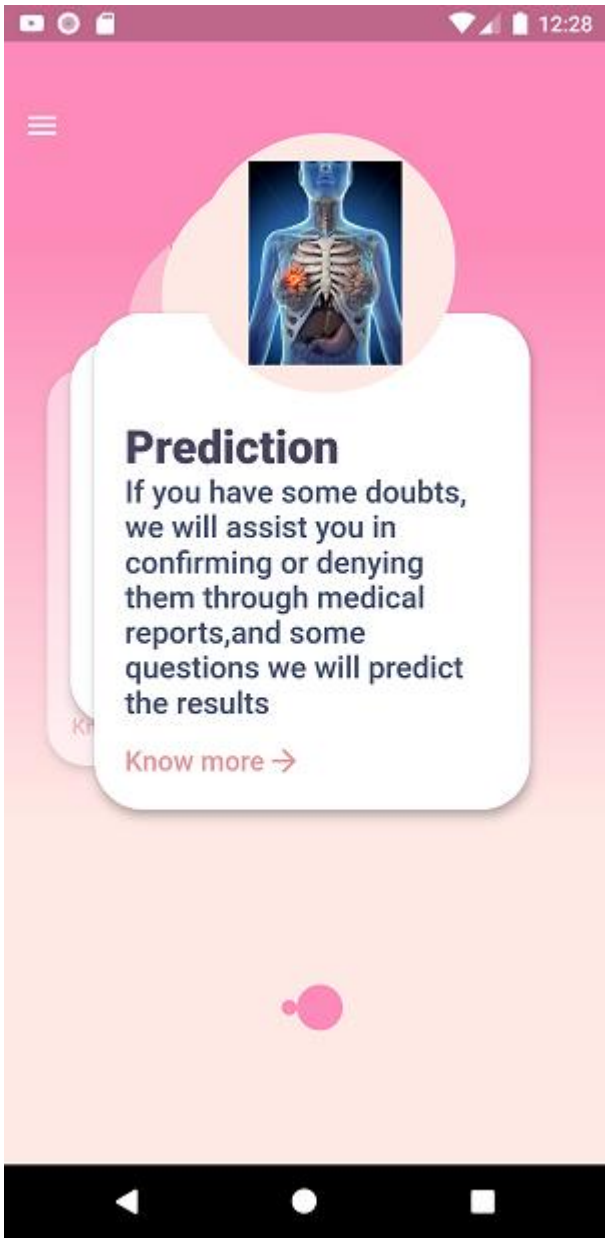
Registration Page



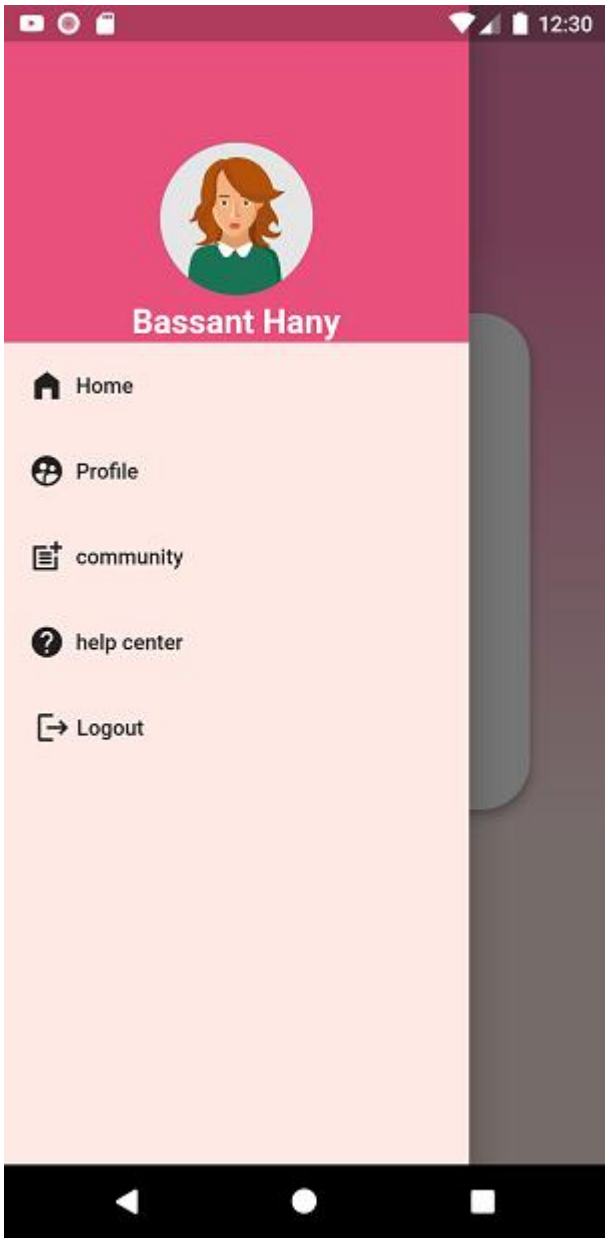
Home Page 1


12:28
Examination
Breast self-exam (BSE), can be an important way to find breast cancer early when it's more likely to be treated successfully.
Know more
Know more

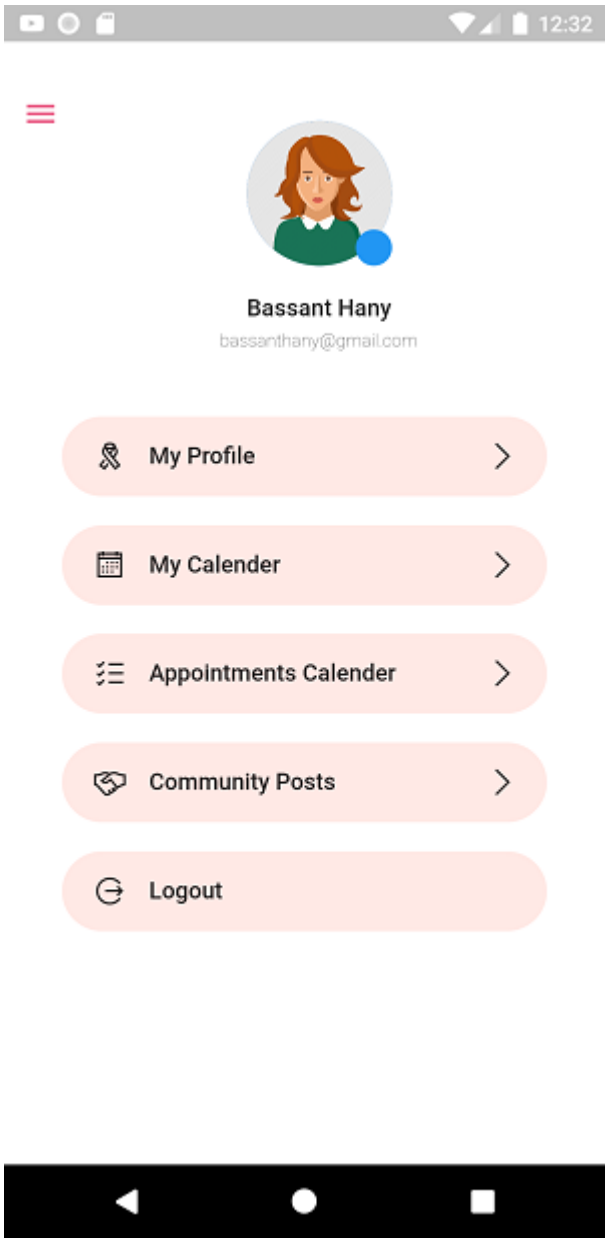
Home Page 2


12:28
Prediction
If you have some doubts, we will assist you in confirming or denying them through medical reports,and some questions we will predict the results
Know more

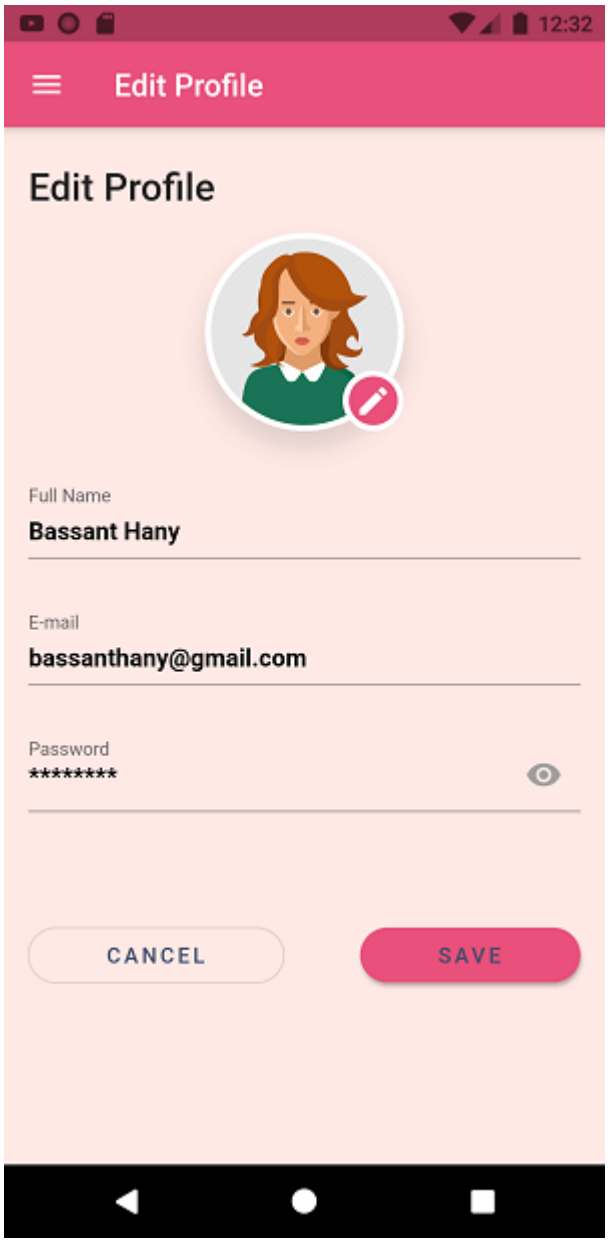
App Drawer


12:30
Bassant Hany
Home
Profile
community
help center
Logout

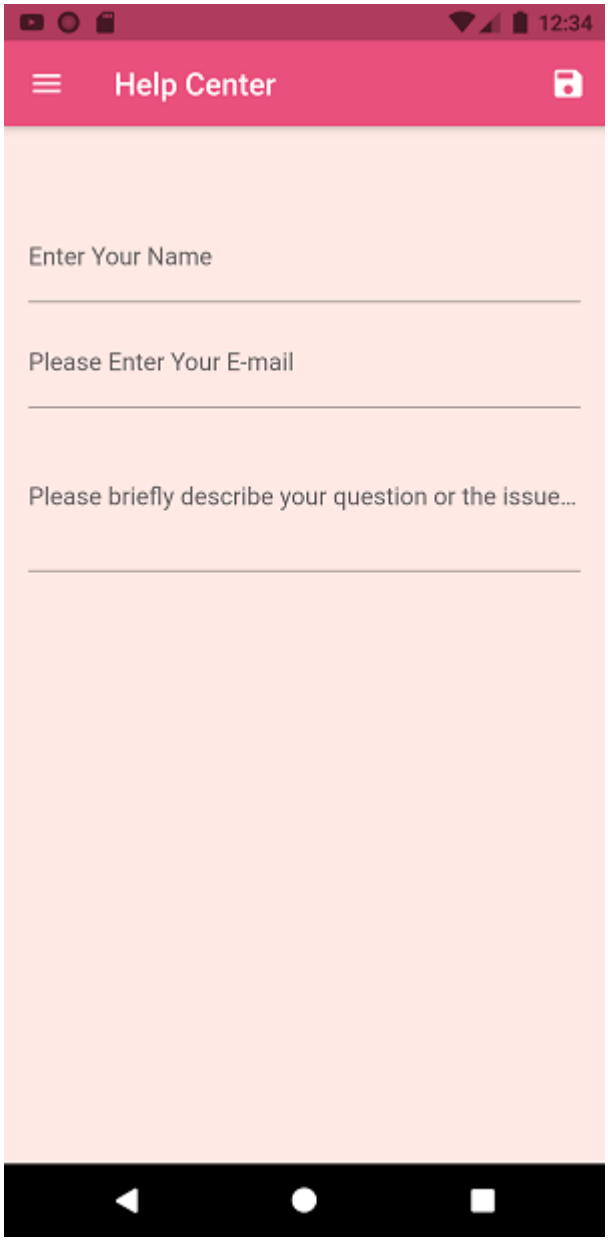
Profile Page



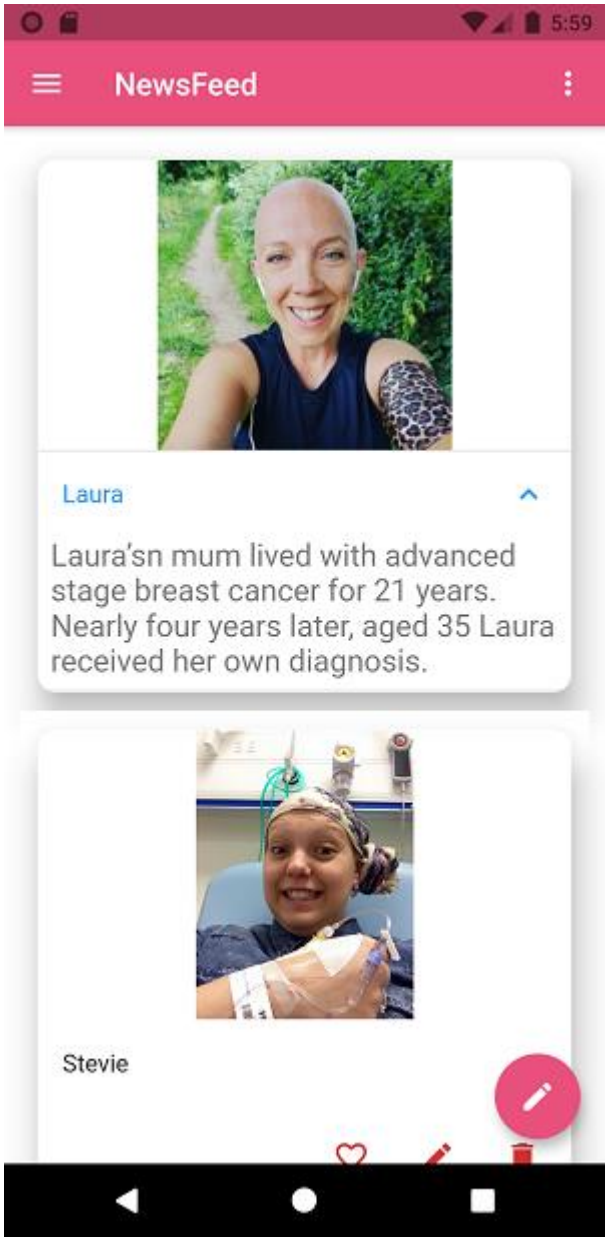
Edit Profile Page



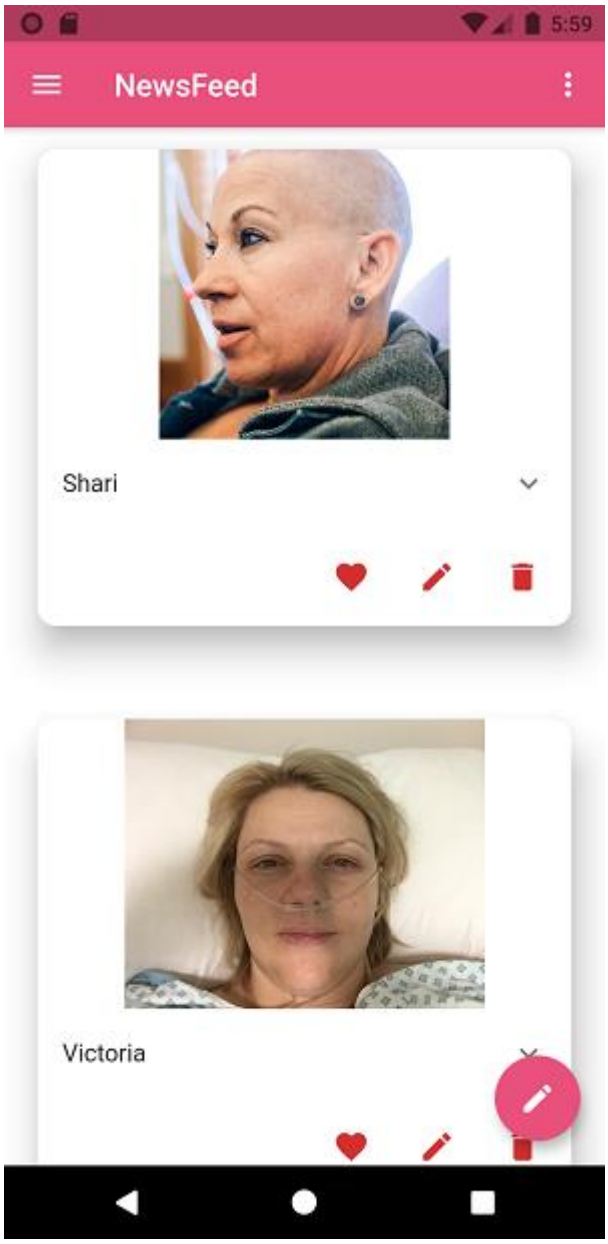
Help Center Page



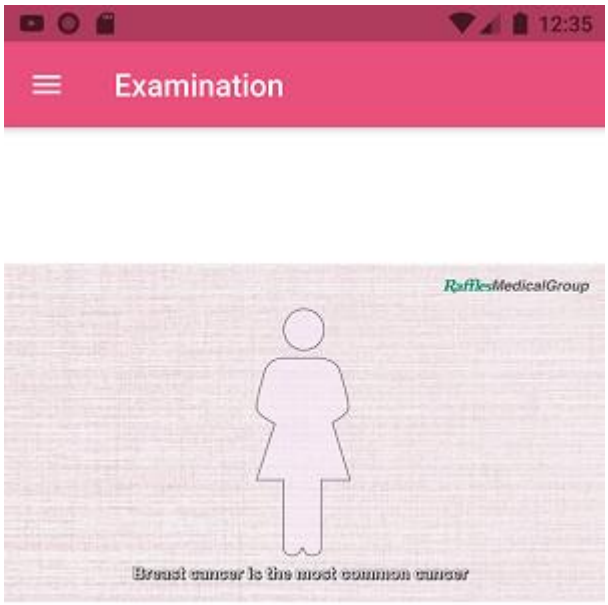
Community Posts (News Feed) Page



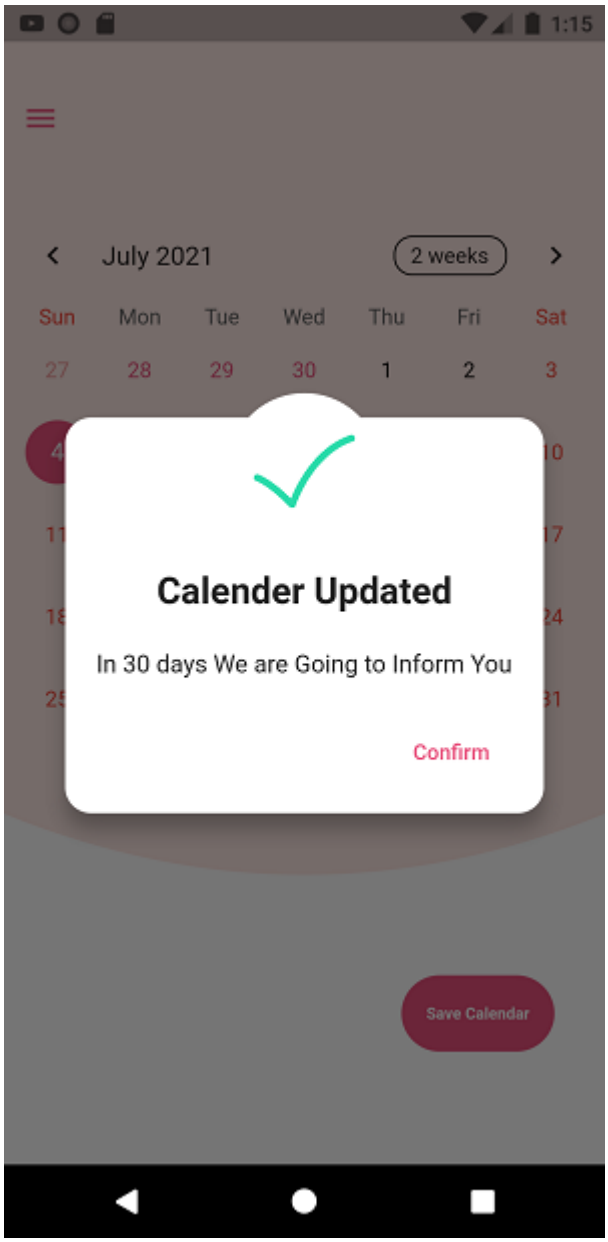
Favorite Posts



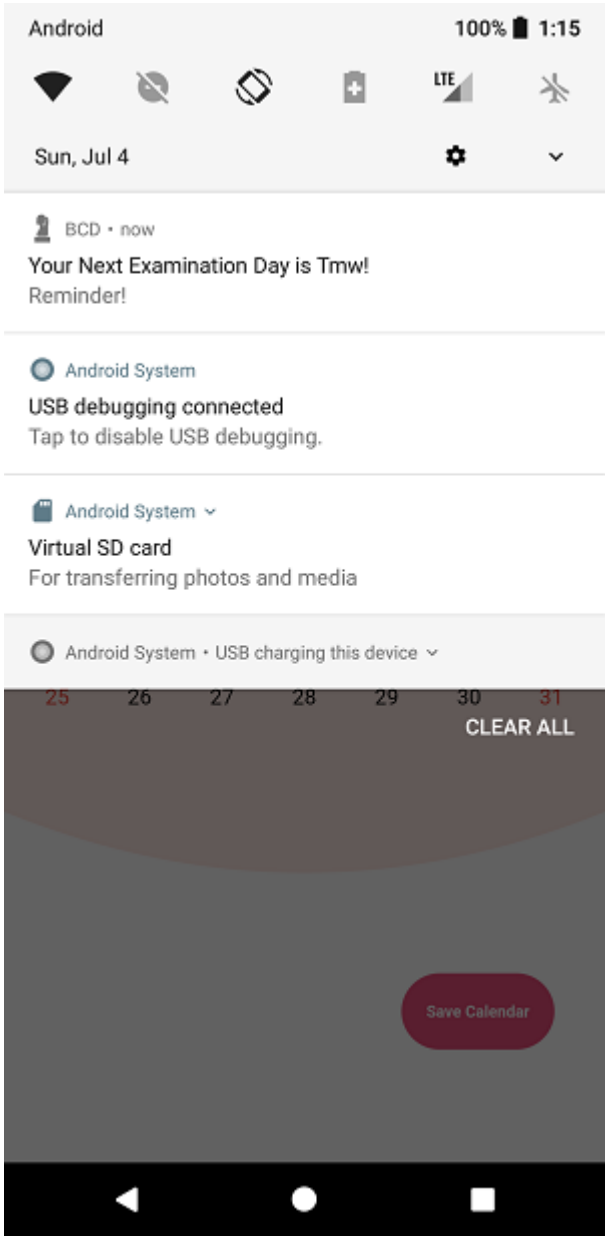
Self-Examination Video page



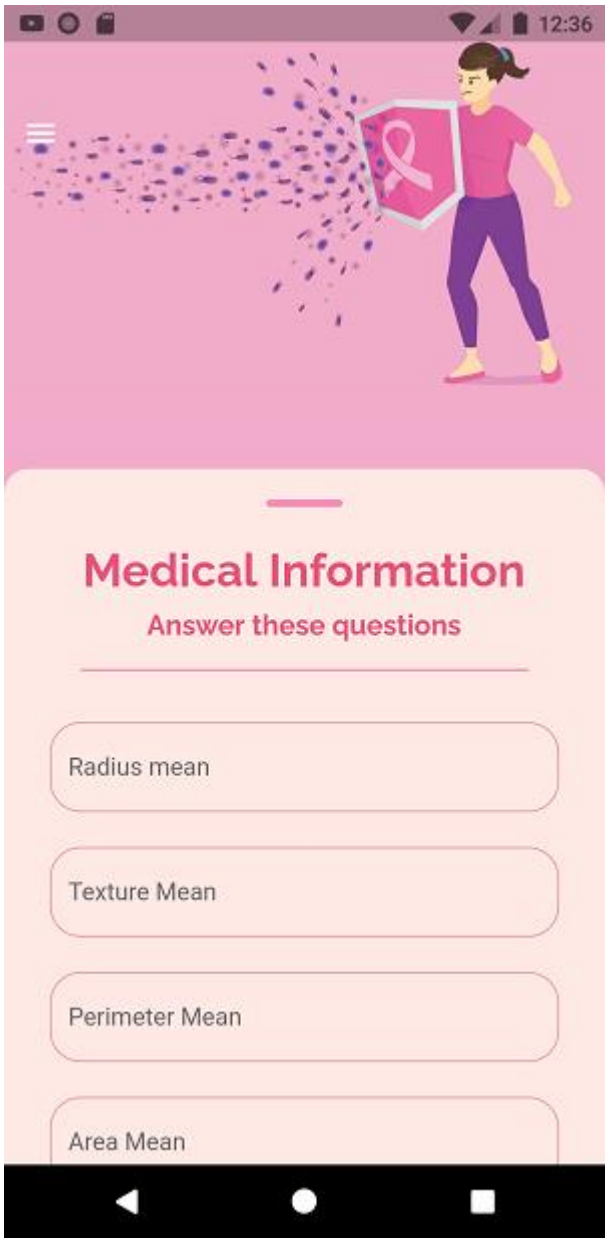
Self-Examination Monthly reminder Pop-up



Self-Examination Notifications

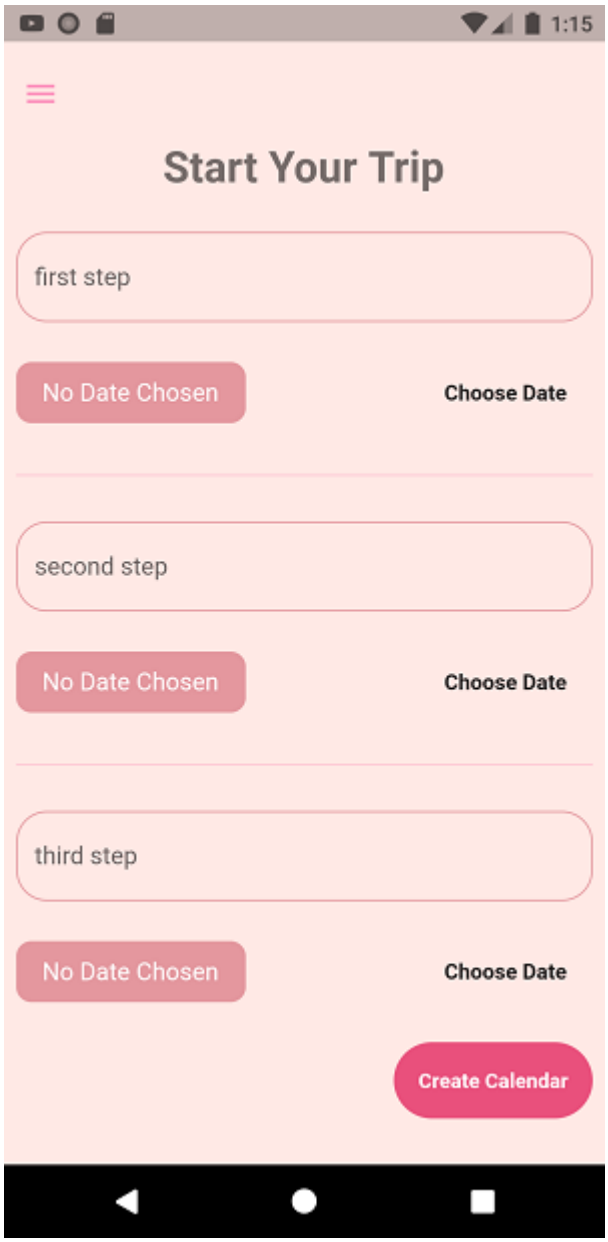


Prediction Questions Page

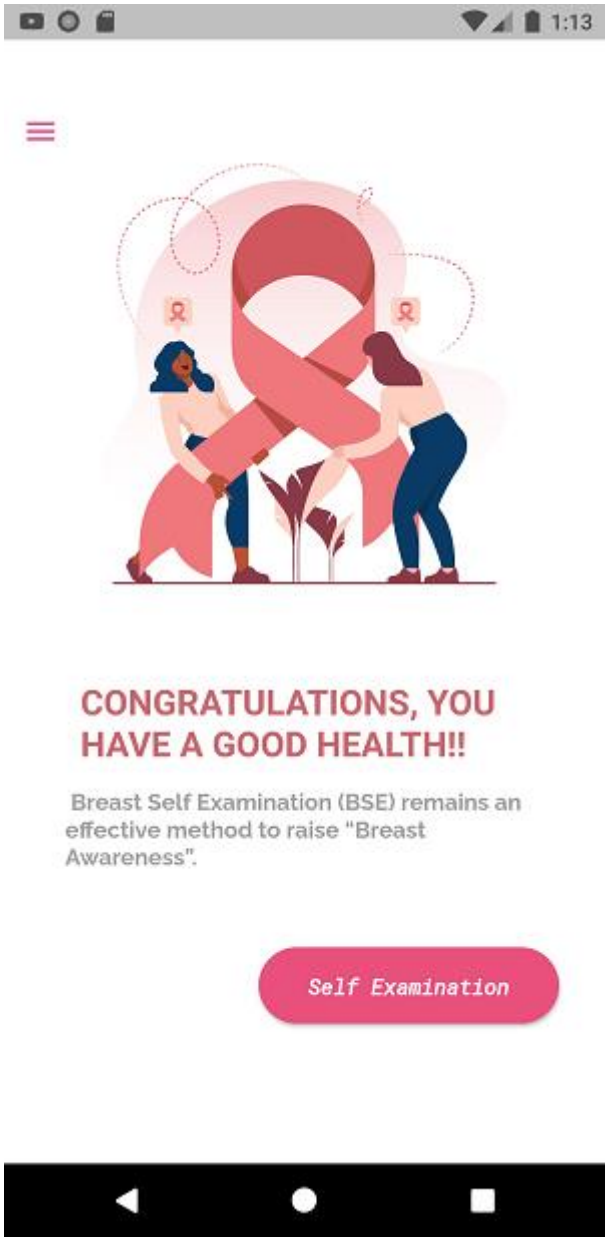


Result Page 1

Patient Calendar page



Result Page 2



Chapter Four

In this chapter we're going to discuss and go deeper in BCD testing, and present the types of testing to be used and test cases we examined our application through.

Chapter Headlines:

1. Functional Testing
2. Non-Functional Testing
3. Testing Inputs

4.1 Functional Testing

1. Unit Testing

Testing of individual items (e.g. modules, programs, objects, classes, etc.)Usually as part of the coding phase, in isolation from other development item sand the system as a whole.

2. Integration testing

Testing the interfaces between major (e.g. systems level application modules) and minor (e.g. individual programs or components) items within an application or system which must interact with each other.

3. System testing

Testing a system behavior as a whole when development is finished and the system can be tested as complete entity.

4. Regression Testing

To check older functionality after integrating new functionality.

5. Acceptance testing

Testing to ensure that a development is ready to be deployed into the business, operational or production environment.

4.2 Non-Functional Testing

1. Performance Testing

Accomplished a designated function regarding processing time and through put rate.

2. Load Testing

Measuring the behavior of within creasing load which can be handled by the component or system.

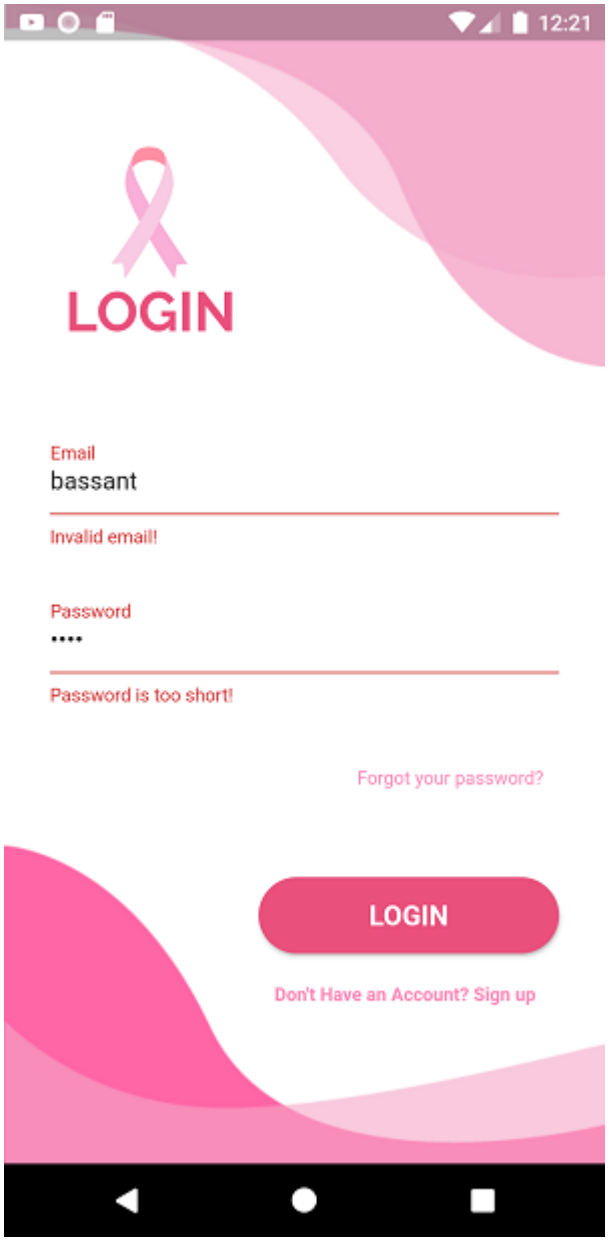
3. Stress Testing

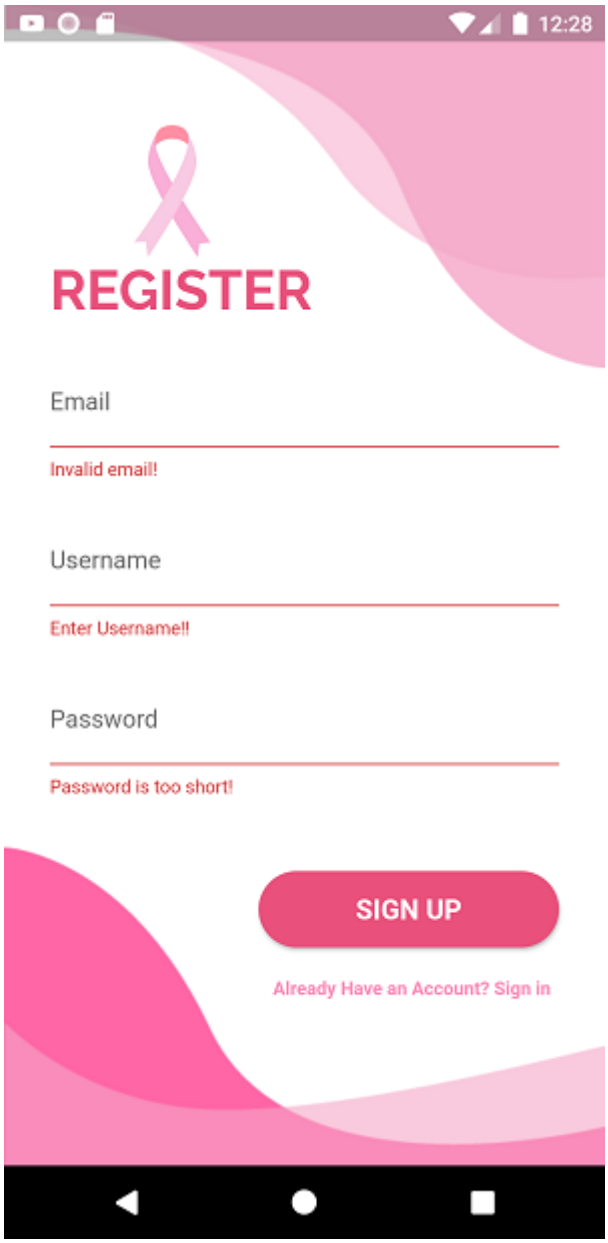
Evaluate a system or component at or beyond the limits of its specified requirements.

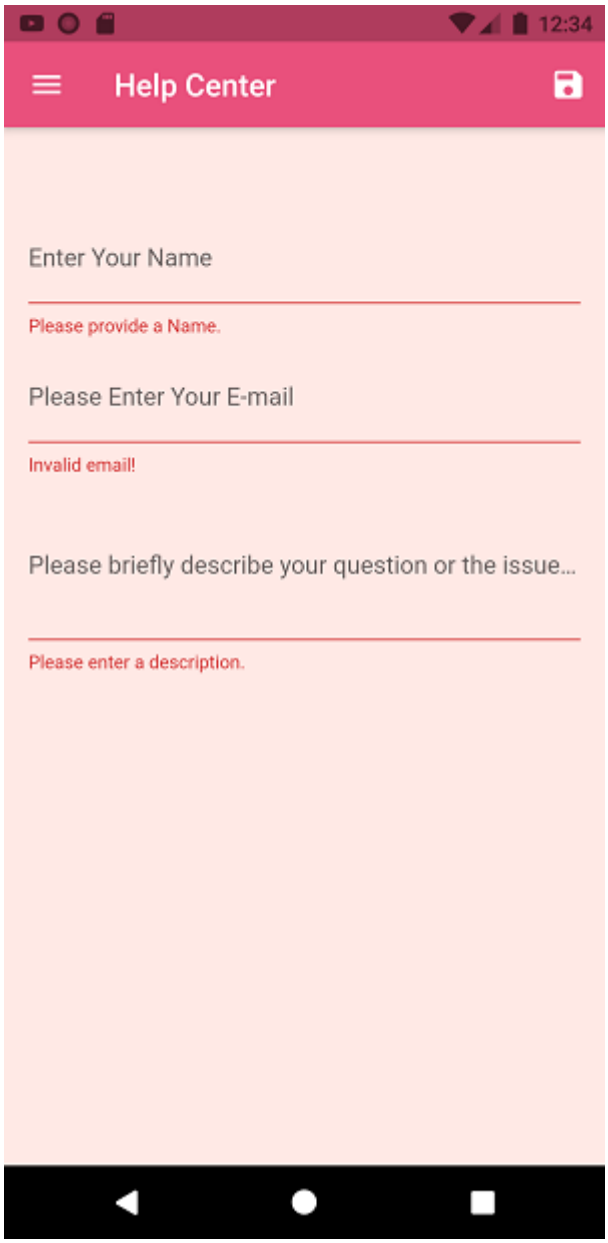
4. Security Testing

Testing how well the system protects against unauthorized internal or external access.

4.3 Testing Inputs




12:28
REGISTER
Email
Invalid email!
Username
Enter Username!!
Password
Password is too short!
SIGN UP
Already Have an Account? Sign in


12:34
Help Center
Enter Your Name
Please provide a Name.
Please Enter Your E-mail
Invalid email!
Please briefly describe your question or the issue...
Please enter a description.

Chapter Five

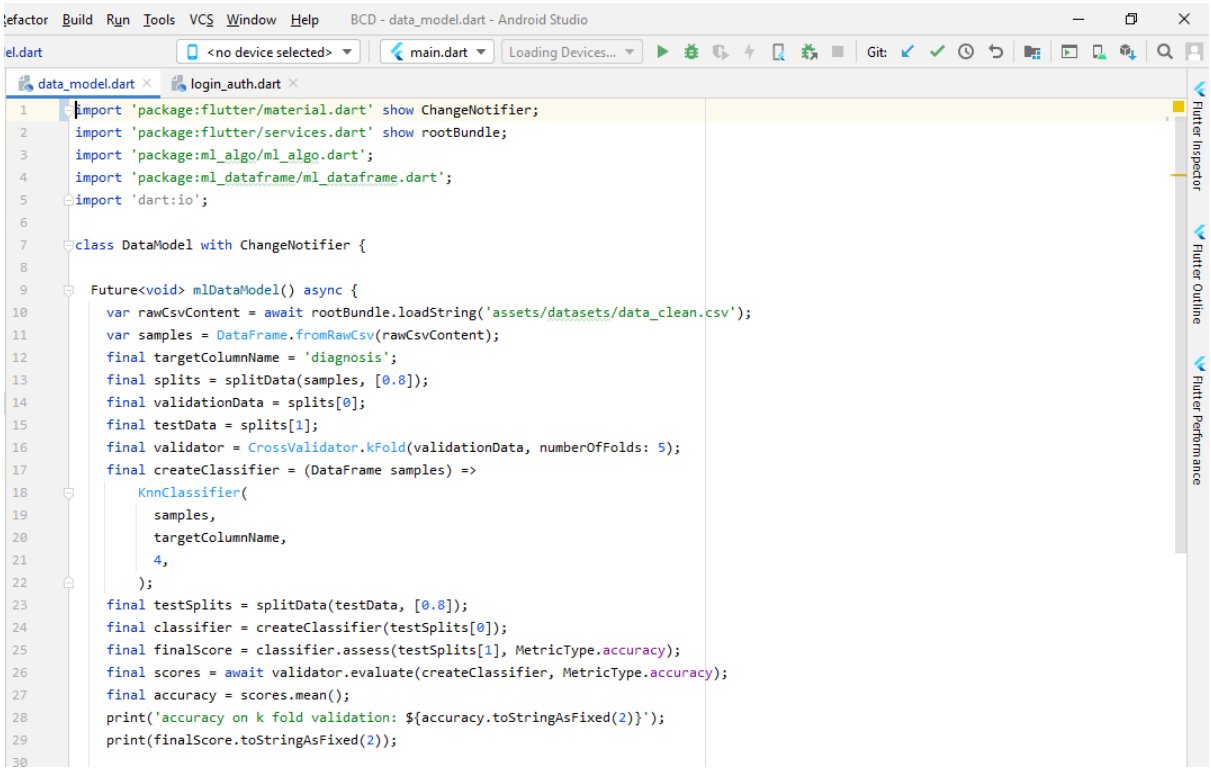
In this chapter we're going to discuss and go deeper in I-Learn system's implementation, and present its code and the algorithms used to build it.

Chapter Headlines:

1. Implementation
2. Flowchart

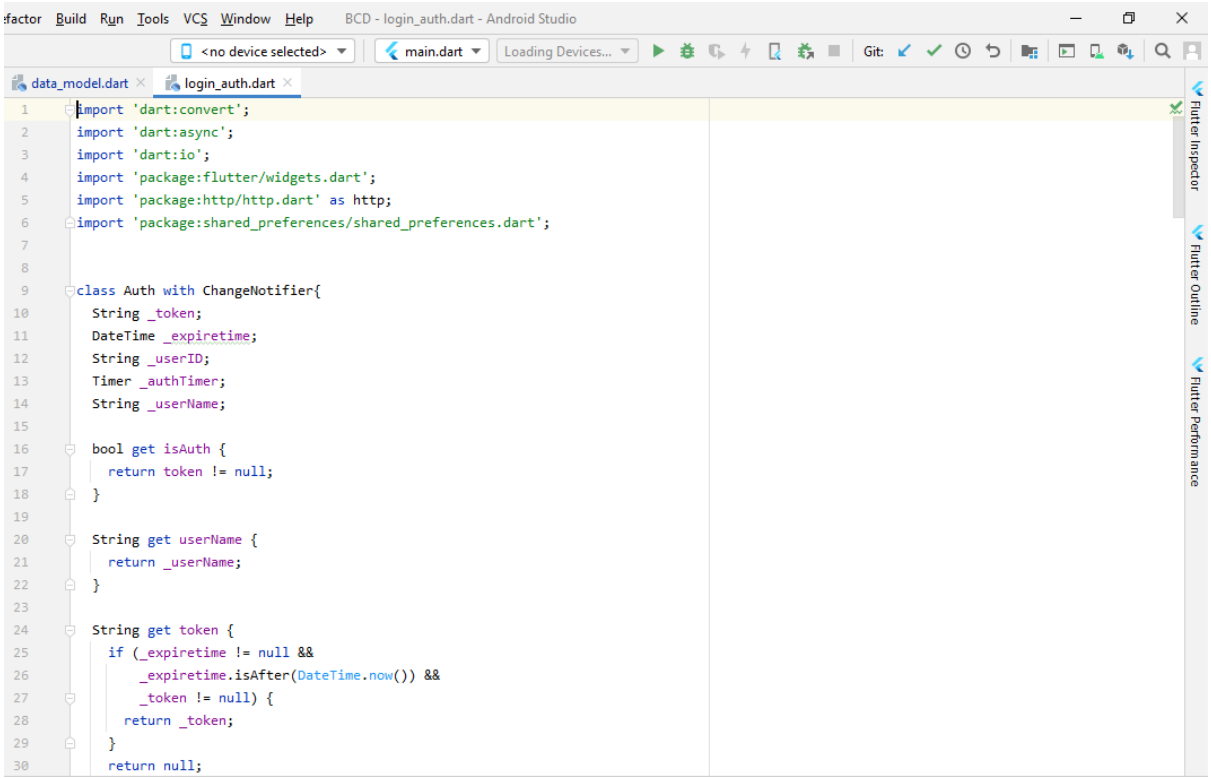
5.1 Implementation

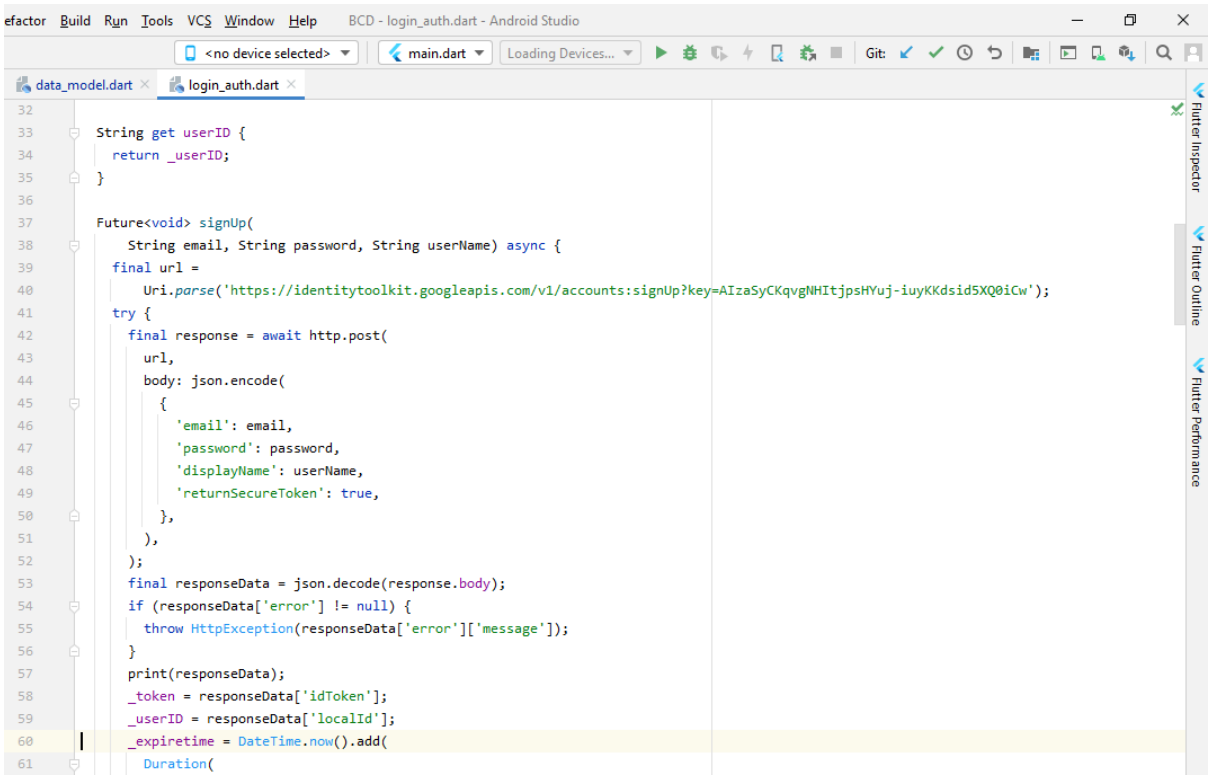
Machine Learning Model

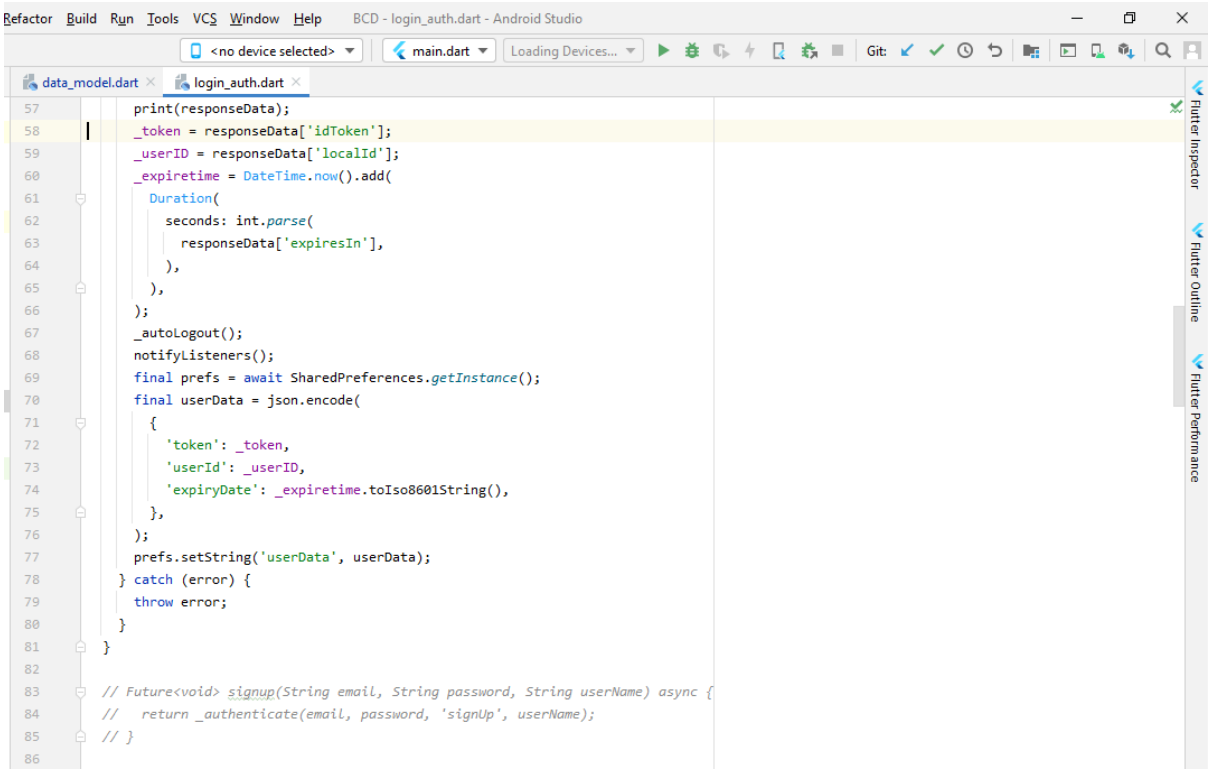



```dart
import 'package:flutter/material.dart' show ChangeNotifier;
import 'package:flutter/services.dart' show rootBundle;
import 'package:ml_algo/ml_algo.dart';
import 'package:ml_dataframe/ml_dataframe.dart';
import 'dart:io';

class DataModel with ChangeNotifier {

  Future<void> mlDataModel() async {
    var rawCsvContent = await rootBundle.loadString('assets/datasets/data_clean.csv');
    var samples = DataFrame.fromRawCsv(rawCsvContent);
    final targetColumnName = 'diagnosis';
    final splits = splitData(samples, [0.8]);
    final validationData = splits[0];
    final testData = splits[1];
    final validator = CrossValidator.kFold(validationData, numberOfFolds: 5);
    final createClassifier = (DataFrame samples) =>
        KnnClassifier(
          samples,
          targetColumnName,
          4,
        );
    final testSplits = splitData(testData, [0.8]);
    final classifier = createClassifier(testSplits[0]);
    final finalScore = classifier.assess(testSplits[1], MetricType.accuracy);
    final scores = await validator.evaluate(createClassifier, MetricType.accuracy);
    final accuracy = scores.mean();
    print('accuracy on k fold validation: ${accuracy.toStringAsFixed(2)}');
    print(finalScore.toStringAsFixed(2));
```

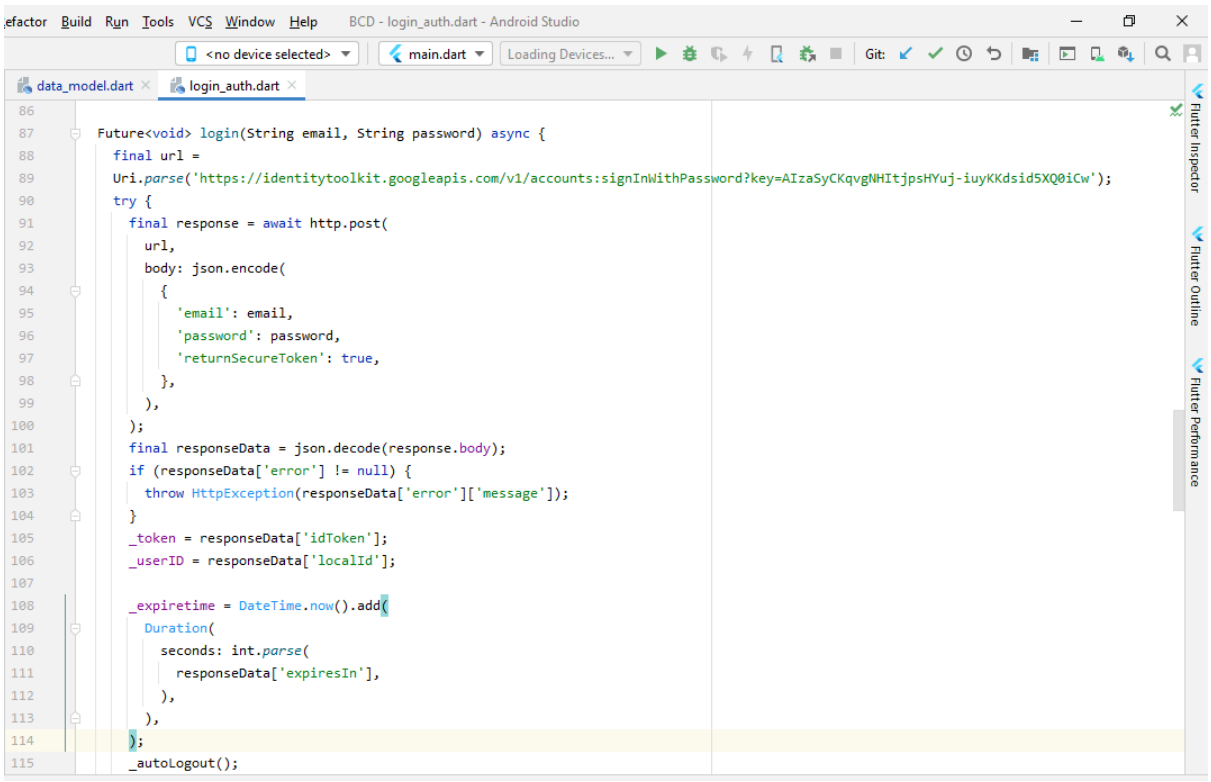
Authorization, Login and logout



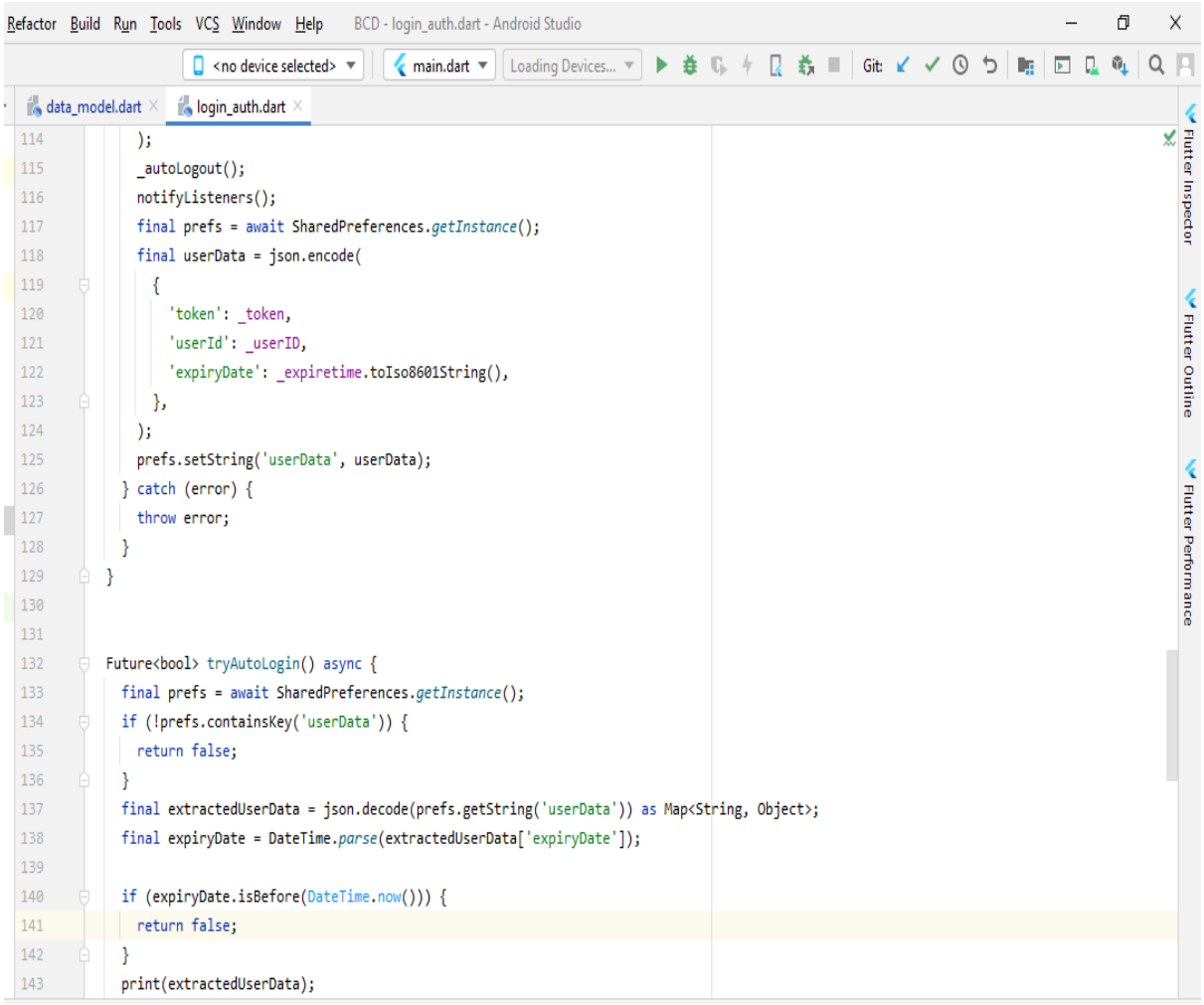


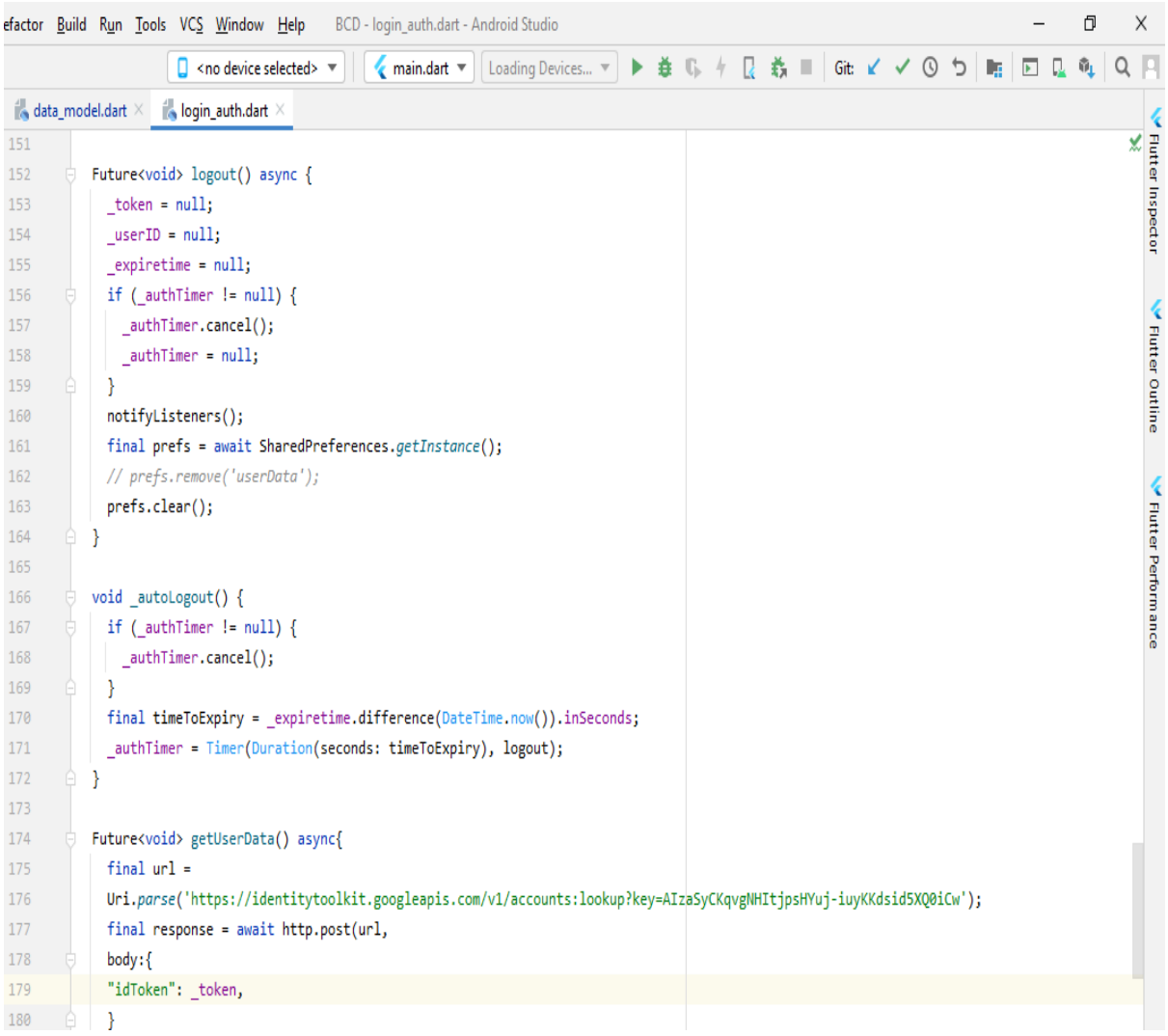



```dart
      print(responseData);
      _token = responseData['idToken'];
      _userID = responseData['localId'];
      _expiretime = DateTime.now().add(
        Duration(
          seconds: int.parse(
            responseData['expiresIn'],
          ),
        ),
      );
      _autoLogout();
      notifyListeners();
      final prefs = await SharedPreferences.getInstance();
      final userData = json.encode(
        {
          'token': _token,
          'userId': _userID,
          'expiryDate': _expiretime.toIso8601String(),
        },
      );
      prefs.setString('userData', userData);
    } catch (error) {
      throw error;
    }
  }

  // Future<void> signup(String email, String password, String userName) async {
  //   return _authenticate(email, password, 'signUp', userName);
  // }
```



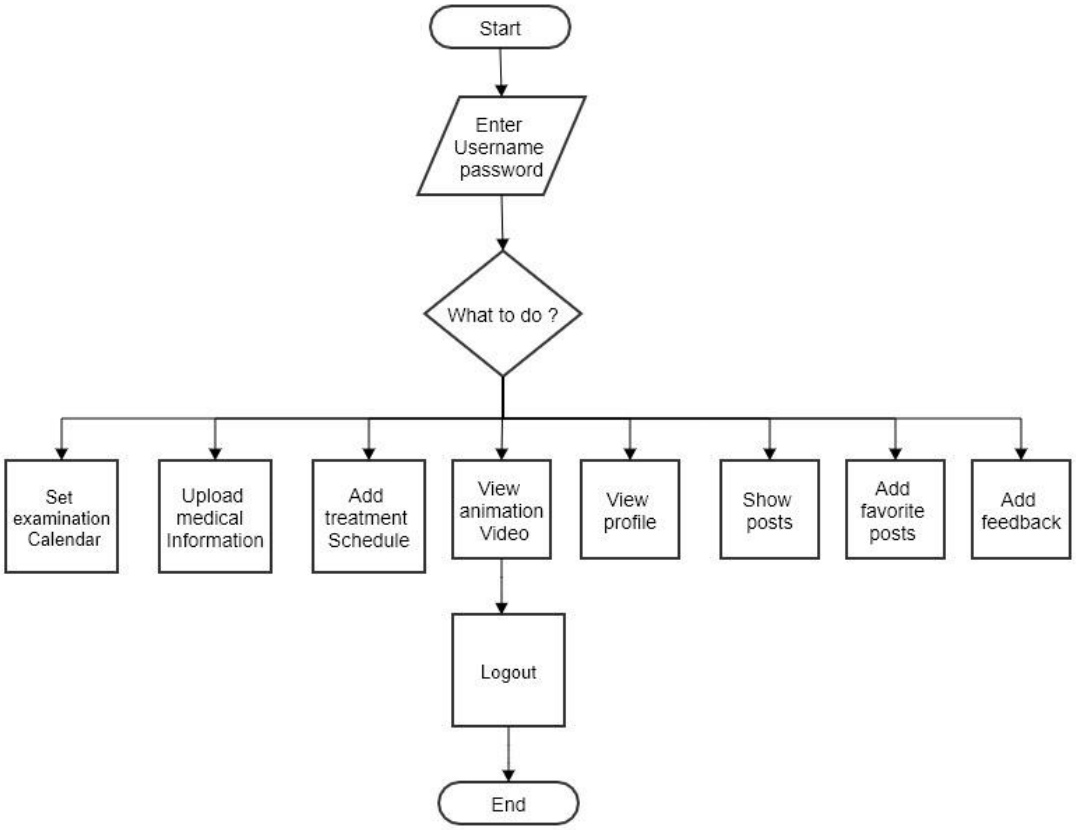

```dart

  Future<void> login(String email, String password) async {
    final url =
    Uri.parse('https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=AIzaSyCKqvgNHItjpsHYuj-iuyKKdsid5XQ0iCw');
    try {
      final response = await http.post(
        url,
        body: json.encode(
          {
            'email': email,
            'password': password,
            'returnSecureToken': true,
          },
        ),
      );
      final responseData = json.decode(response.body);
      if (responseData['error'] != null) {
        throw HttpException(responseData['error']['message']);
      }
      _token = responseData['idToken'];
      _userID = responseData['localId'];

      _expiretime = DateTime.now().add(
        Duration(
          seconds: int.parse(
            responseData['expiresIn'],
          ),
        ),
      );
      _autoLogout();
```


Refactor Build Run Tools VCS Window Help
BCD - login_auth.dart - Android Studio
<no device selected>
main.dart
Loading Devices...
Git:
data_model.dart
login_auth.dart
);
_autoLogout();
notifyListeners();
final prefs = await SharedPreferences.getInstance();
final userData = json.encode(
{
'token': _token,
'userId': _userID,
'expiryDate': _expiretime.toIso8601String(),
},
);
prefs.setString('userData', userData);
} catch (error) {
throw error;
}
}
Future<bool> tryAutoLogin() async {
final prefs = await SharedPreferences.getInstance();
if (!prefs.containsKey('userData')) {
return false;
}
final extractedUserData = json.decode(prefs.getString('userData')) as Map<String, Object>;
final expiryDate = DateTime.parse(extractedUserData['expiryDate']);
if (expiryDate.isBefore(DateTime.now())) {
return false;
}
print(extractedUserData);
Flutter Inspector
Flutter Outline
Flutter Performance




```dart
151
152  Future<void> logout() async {
153    _token = null;
154    _userID = null;
155    _expiretime = null;
156    if (_authTimer != null) {
157      _authTimer.cancel();
158      _authTimer = null;
159    }
160    notifyListeners();
161    final prefs = await SharedPreferences.getInstance();
162    // prefs.remove('userData');
163    prefs.clear();
164  }
165
166  void _autoLogout() {
167    if (_authTimer != null) {
168      _authTimer.cancel();
169    }
170    final timeToExpiry = _expiretime.difference(DateTime.now()).inSeconds;
171    _authTimer = Timer(Duration(seconds: timeToExpiry), logout);
172  }
173
174  Future<void> getUserData() async{
175    final url =
176    Uri.parse('https://identitytoolkit.googleapis.com/v1/accounts:lookup?key=AIzaSyCKqvgNHItjpsHYuj-iuyKKdsid5XQ0iCw');
177    final response = await http.post(url,
178    body:{
179    "idToken": _token,
180    }
```

5.2 Flowchart



Chapter Six

In this chapter we're going to find out the results of the project,our recommendations that will be added to the system in the future, also will write the project conclusion clearly.

Chapter Headlines:

1. Results
2. Future Plan
3. Conclusion

6.1 Results:

We developed a system that allows for women users to be a kind of support to the patients and a reminder to the rest. The application has successfully predicted some test results.

Users can set notifications for various reasons like self-examination or sessions and appointments.

She can also post her story, browse others and add the best to her favorites

6.2 Future Plan:

Patients can communicate with real life doctors

Medical advices and tip

Breast Histopathology Images

Symptom Management & Tracking

Add more personalized features to the user profile like uploading a picture

6.1 Conclusion:

Eventually, in this project we tried to make the best give breast cancer patients or those who are at risk all the help we can offer, it's just a way of expressing our duty towards our society as we felt responsible enough to participate in making this journey a little bit easier with one of the most powerful tools we can use and we chose as our studying field years ago which is TECHNOLOGY. In summary, our society would be more convenient if we use what we have learned to help our women to be more aware of breast cancer as it's very common nowadays. We should teach everyone to use all his knowledge to help build his community. Finally, realizing the fact that "a few clicks" can do a lot to a society, will definitely make us more aware of the technology role and become better versions of ourselves.