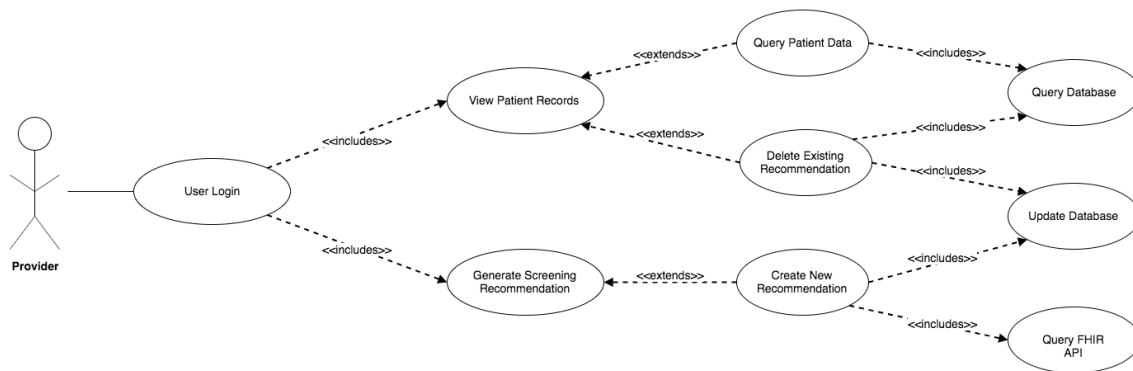


# Use Case Model

**Author:** Cranky Euler

## 1 Use Case Diagram



## 2 Use Case Descriptions

### 2.1 User Login

Requirements: This use case must allow a provider to log in to the application with a unique user ID.

Pre-conditions: The user must have launched the application and already have an existing user ID on the local instance of the application.

Post-conditions: The user will be logged into the application and will be redirected to the main landing page.

Scenarios:

2.1.1: The user has logged and is taken to a main landing screen, where he can choose to view existing screening recommendations for existing patients or generate a new

screening recommendation record.

## 2.2 Generate Screening Recommendation

Requirements: This use case must allow a provider to generate a chlamydia/gonorrhea screening recommendation from existing patient health records and manually-entered data.

Pre-conditions:

The user must have launched the application.

The user must be logged in to the application.

Post-conditions:

A new screening recommendation record will be created for a given patient and written to the application instance database.

The existing recommendation records list will be updated.

Scenarios:

### 2.2.1 Create New Recommendation Records(s)

The user will navigate to the "generate new recommendation screen".

On this screen, the user will need to enter patient details, including but not limited to:

- Full name

- Personal details

- Sexual history

- Other relevant information for chlamydia/gonorrhea screening purposes

Once the user is done editing the inputs listed above, he can submit the information, which will create a new record in the application's database instance.

In addition to using the manually-entered data, the

application will query the external FHIR API to get additional medical history for a given patient, which will be used to generate a patient recommendation.

After the screening recommendation has been successfully saved and updated in the application's database instance, a confirmation message containing the unique ID of the newly created recommendation record will be displayed. Furthermore, the screen will also display the following details:

- Likelihood of a patient to have chlamydia or gonorrhea.

- Recommendation for whether they should get screened for chlamydia/gonorrhea or not.

- Recommendation for frequency of screenings.

## 2.3 View Patient Records

Requirements: This use case must allow a player to choose a cryptogram to choose a cryptogram from a list of available cryptograms, save/submit a solution to a chosen cryptogram, and see previously solved cryptograms.

Pre-conditions:

- The user must have launched the application.

- The user must be logged in to the application.

- At least one patient recommendation record must exist on the application database instance.

Post-conditions:

- The application will display a list of available patient recommendation records.

- The application database instance will be updated when a record is deleted.

Scenarios:

### 2.3.2 Query Patient Data

The application will display a list of existing recommendation records, which the user can use to select a particular record to get additional information for that record.

Once a given record has been selected, it will redirect it to a page that displays the following information for that patient recommendation record:

- Patient details e.g. full name, personal details.

- Recommendation information e.g. frequency of screenings, likeliness to have chlamydia or gonorrhea, screening recommendation (yes/no).

### 2.3.2 Delete Existing Recommendation Records(s)

On a particular patient recommendation record's information page, the user will have the ability to delete the record by clicking the "Delete" button/

This will prompt the user with a confirmation prompt - if the user selects "Yes", the record will be removed from the application database instance and the user will be redirected back to the main page, which lists all existing recommendation records.

# Design Document

**Team:** *Team Cranky Euler*

## 1 Design Considerations

### 1.1 Assumptions

The purpose is to build a web-based tool that recommend screening for Chlamydia and Gonorrhea in a selection of patients. The app consists then of a doctor/provider facing portal with FHIR integraion that based on medical history for the patient and manual input about the patient can give insights or help decide whether or not they at increased risk of infection.

For this we will use FHIR Api to interact with the database either by querying or modifying, we may add some backend logic to help determine if the patient is at risk of infection or not.

The dependencies should be:

SQL, python, python-flask, javascript, react.js

### 1.2 Constraints

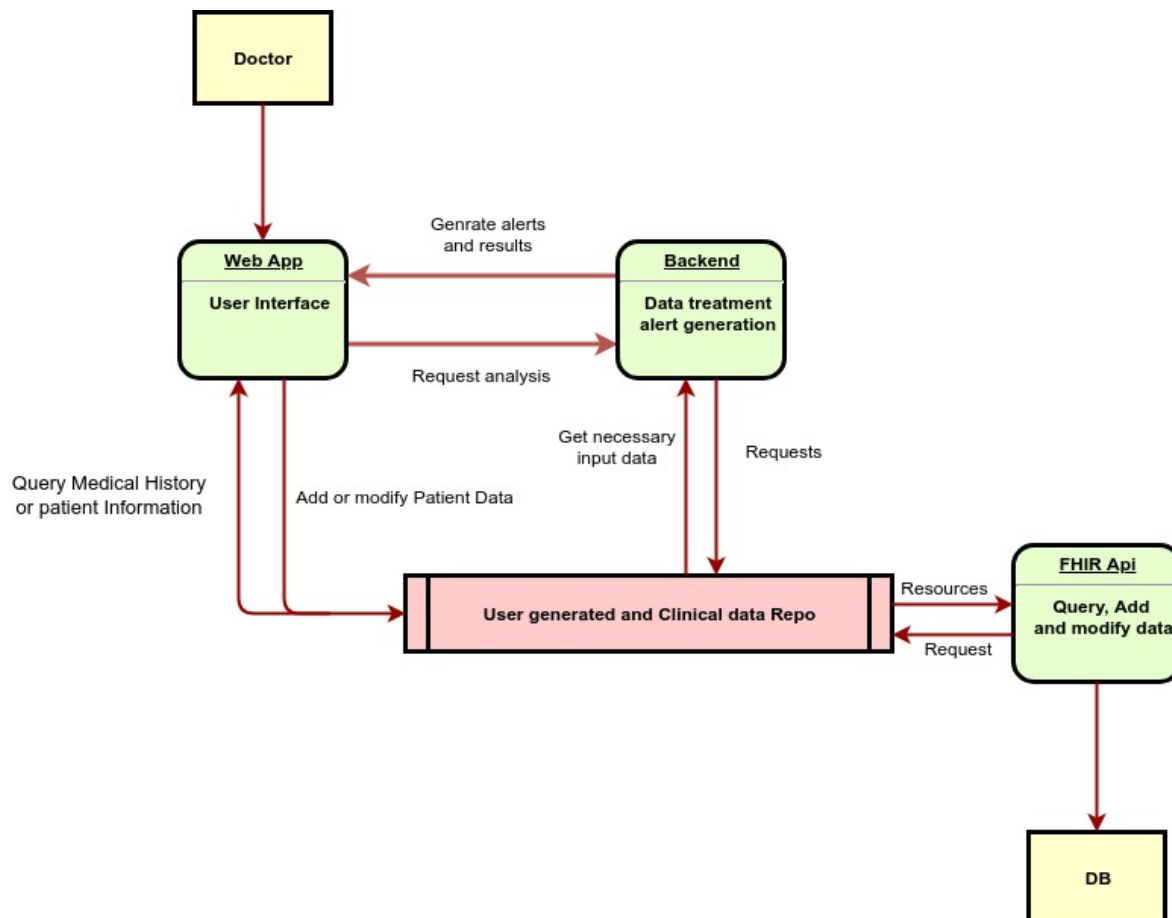
The lack of formal medical experience

### 1.3 System Environment

Software: Chrome/Firefox browser Hardware: Not applicable in our case

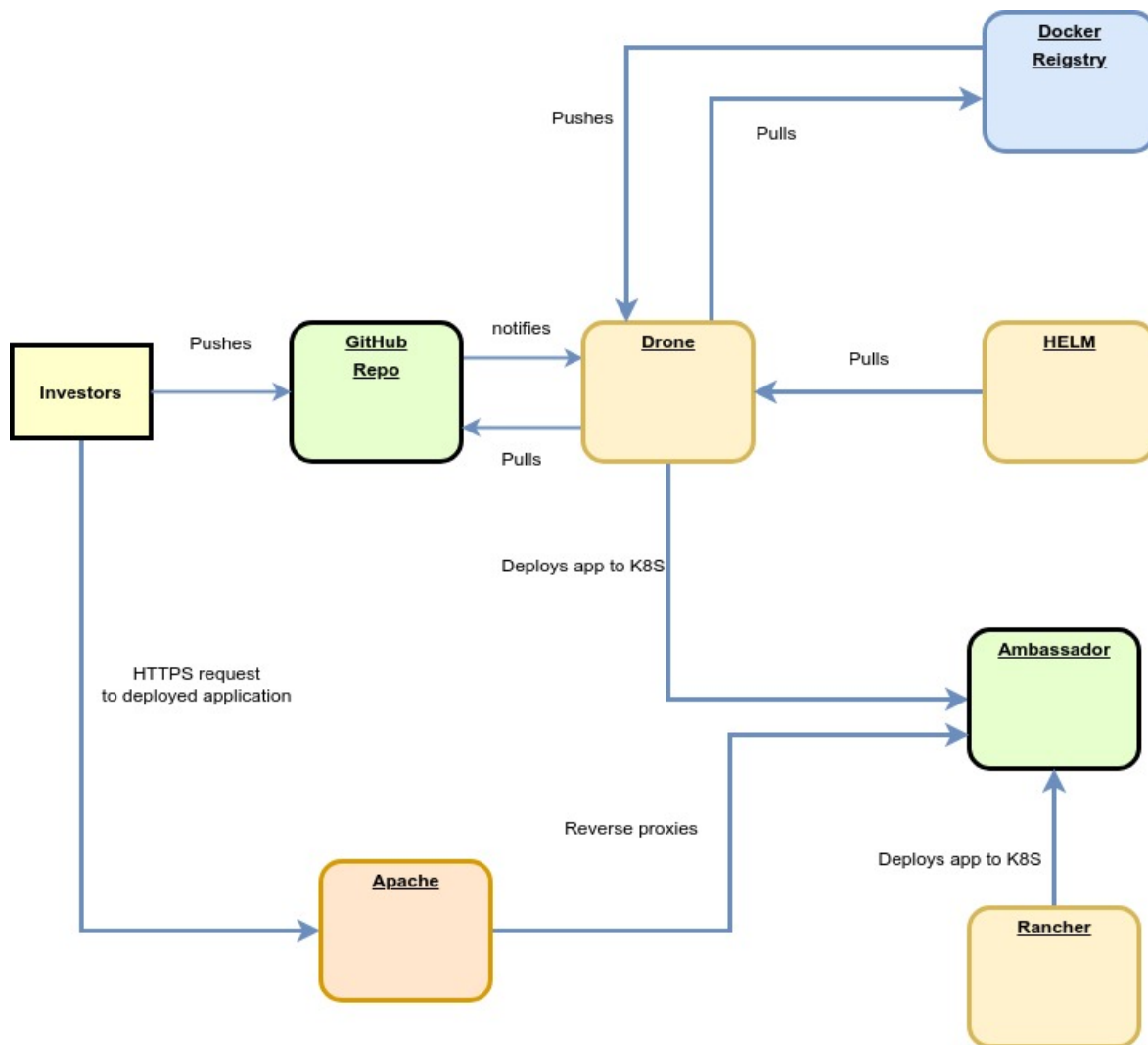
## 2 Architectural Design

### 2.1 Component Diagram

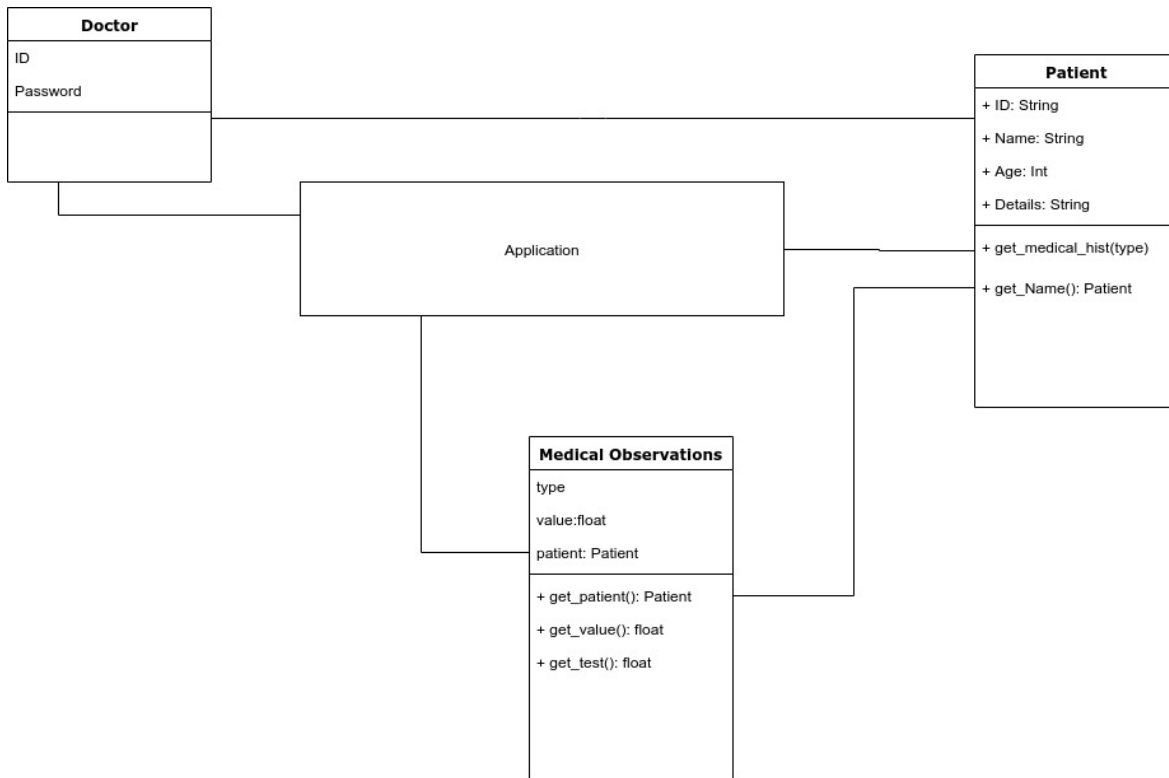


### 2.2 Deployment Diagram

Based on the HDAP lecture:



### 3 Class and Other Diagrams



## 4 User Interface Design

These are very basic UI Design to get an idea on the things we may need to implement

# SafeSex Screening

### Provider Login

ID

Password

Login



[illegible]

## Patients List

Search

- 
- | Device Type          | Percentage of Respondents |
|----------------------|---------------------------|
| Smartphone           | 100%                      |
| Tablet               | 95%                       |
| Smart TV             | 85%                       |
| Smart speaker        | 80%                       |
| Smart home appliance | 75%                       |
| Smart car            | 5%                        |
| Smart watch          | 5%                        |

# SafeSex Screening

Patient Details

Add details

Parameter

Value

Risk Assessment

Value

1	
---	--

Government	Percentage
Current government	85%
Previous government	15%

[illegible]

# Test Plan

**Author:** *Team Cranky Euler*

## 1 Testing Strategy

### 1.1 Overall strategy

Our testing will encompass unit and integration tests. Unit tests will mostly be for the backend functionality to ensure the core functional logic is implemented such as using patient-provided data to determine the risk level of Chlamydia or Gonorrhea. Integration tests will be used to test the whole system as a whole once most of the functionality has been built in. For our unit tests, we will use chai and mocha Javascript libraries. The UI will mostly involve manual testing by the QA lead.

### 1.2 Test Selection

We will employ white box testing techniques in the project which will involve working with the source code. We will focus on independent testable features such as authentication, addition of patient information and fetching recommendation details. We anticipate that most of our tests will be unit tests.

### 1.3 Adequacy Criterion

We will strive to achieve 70% test coverage as measured by tools such as Istanbul to test this coverage.

### 1.4 Bug Tracking

We will track bugs by creating Github issues with label *bug* in our assigned Github repository. This will ensure visibility to the whole team as well as our TA.

## 1.5 Technology

We will use chai and mocha testing libraries for our unit tests. We hope to use Selenium for the test of our web application as well if time allows.

## 2 Test Cases

Preliminary test cases:

Purpose	Steps to perform the test	Expected result	Actual Result	Pass/fail
Test that an entering correct username results in a successful login	On the login screen of the app, type in the username of the provider and press the log in button	Success		
Test that entering incorrect username results in unsuccessful login	On the login screen of the app, type in the username of the provider and press the log in button	Error. Incorrect username.		
Test adding patient data into the system	On the home page, upon a successful login, the provider can enter patient information.	Patient data is successfully saved.		
Test that a correct recommendation is given for screening of Chlamydia	Add a new patient or search for a new patient whose sexual history that matches a symptom of Chlamydia.	Success. Go for screening for Chlamydia once every 6 months.		
Test that a correct recommendation is given for screening of Gonorrhea	Add a new patient or search for a new patient whose sexual history that matches a symptom of Gonorrhea.	Success. Go for screening for Gonorrhea once every 6 months.		
Test that a correct recommendation is given if no symptoms or Chlamydia and Gonorrhea are found	On the recommendation screen, the patient should enter their bio information and sexual history that is not at risk,	Success. It looks like you're safe. No need for screening. Check back in 6 months		
Test that we can fetch a list of patients from the FHIR <a href="#">api</a> .	On the home page, click patient data.	Patients data is returned if available.		

CRANKY EULER GROUP PROJ.

TITLE	Name	Email	
PROJECT MANAGER	David Awad	me@davidawad.com	
QUALITY ASSURANCE	Anne Chepkaitany	rerimolanne.4@gmail.com	
SOFTWARE DEVELOPER	Wassim Fourati	wassim9429@gmail.com	
SOFTWARE DEVELOPER	Federico Ciner	federico.ciner@gmail.com	

PHASE	DETAILS	SEPTEMBER				OCTOBER				NOVEMBER				DECEMBER	
PROJECT WEEK:		Enter the date of the first Monday of each month -->													
			9	16	21	30	7	14	21	28	4	11	18	25	2
1	Deliverable 0:Team confirmation	- Team Confirmation Part A	Team Confirmation Part A				TEAM CONFIRMATION PART B								
		- Team Confirmation Part B													
		- Hello World Deployment					Hello World Deployment!								
	Deliverables 1 & 2	- Deliverable 1: Planning & Requirements					Deliverable 1: Planning & Requirements								
		- Deliverable 2: Product Design					Deliverable 2: Product Design								
3	Deliverable 3	- Deliverable 3: Develop & Test I					Deliverable 3: Develop & Test I								
		- Deliverable 3: Develop & Test II						Deliverable 3: Develop & Test II							
		- Deliverable 3: Develop & Test III							Deliverable 3: Develop & Test III						
		- Deliverable 3: Develop & Test IV								Deliverable 3: Develop & Test IV					
4	Deliverable 4	- Deliverable 4: Deployment									Deliverable 4: Deployment				
		- Deliverable 5: Final Submission										Deliverable 5: Final Submission			
5	Submission and Evaluation	- Team Member Evaluation												Team Member Evaluation	
		- Mentor Evaluation												Mentor Evaluation	

P  
R  
O  
J  
E  
C  
T  
  
E  
N  
D