

Hamzah Wahi

Suhyeok Bang

Jeffery Oh

Brendan O'Connor

Galen Reposh

M02-A01 Project Module and API Design

Use Cases and their related classes

1. A student is submitting vaccination records to university. (Galen)

- Model:
 - Student(PatientModel)
 - Immunization records(RecordListModel)
- View:
 - Login Screen(LoginView)
 - 2FA Screen(TwoFactorView)
 - Menu Screen (MenuView)
 - Immunization Document (RecordView)
 - Download Screen
- Controller:
 - Student Controller(UserCntrl)
 - Login Controller(LoginCntrl)

2. Doctor updating medical records for a 30 year old patient who had the flu. (Brendan)

- Model
 - Doctor(StaffModel)
 - Patient(PatientModel)
 - Record List(RecordListModel)
 - Patient Record(PatientModel)
 -
- View
 - Patient View(PatientView)
 - Record View(RecordView)

- Record Update View (RecordView)
- Controller
 - Doctor Controller(UserCntrl)
 - Patient Controller(PatientCntrl)

3. Nurse accessing medical records for a patient while the patient is in the waiting room.
(Jeff)

- Model:
 - EMT or Hospital Staff(StaffModel)
 - Patient Record(PatientListModel)
 - Patient List
 - Record List
- View:
 - Record View (RecordView)
 - Patient View (PatientView)
 - Login View (LoginView)
 - Profile (ProfileView)
- Controller:
 - Patient Controller (UserCntrl)
 - Record Controller (PatientListCntrl)
 - EMT/Doctor Controller (UserCntrl)
 - Login Controller (LoginCntrl)

4. A patient wanted to switch doctors after moving to a new location, so they needed to transfer medical records. (Suhyeok)

- Model:
 - Patient (PatientModel)
 - Patient List (PatientListModel)
 - Doctor (StaffModel)
 - Record list (RecordListModel)
- View:
 - Login View (LoginView)
 - Record View (RecordView)
 - Patient View (PatientView)
 - Profile View (ProfileView)

- Controller:
 - Patient controller (UserCntrl)
 - Doctor controller (UserCntrl)
 - Login Controller (LoginCntrl)

5. A new patient wants to be registered in the system. He works with his doctor to update his medical records. (Ham)

- Model:
 - Student (PatientModel)
 - Doctor (StaffModel)
- View:
 - Login Screen (LoginView)
 - 2FA Screen (TwoFactorView)
 - Menu Screen (MenuView)
 - Create Account Screen (CreateAccountView)
- Controller:
 - Student Controller (UserCntrl)
 - Doctor Controller (UserCntrl)
 - Login Controller (LoginCntrl)

Total Classes:

- Model:
 - StaffModel
 - PatientModel
 - RecordsListModel
 - PatientListModel
 - UserListModel
 -
- View:
 - LoginView
 - TwoFactorView
 - MenuView
 - CreateAccountView
 - ProfileView
 - PatientListView
 - PatientView
 - PatientRecordsView
- Controller:
 - UserCntrl
 - PatientListCntrl
 - LoginCntrl

LCOM4 Analysis:

User Controller (LCOM4 value = 1): The user controller which is responsible for the backend control of the users variables has 3 primary use cases. It will firstly get the

records of the patient which will return an array or arraylist of attributes. Using those attributes it will be passed to a download method, or an update method. All of these are dependent on the attributes that were fetched by the get method. The

PatientModel

PatientListModel

StaffModel

All the views except LoginView, TwoFactorView, CreateAccountView

Patient List Controller (LCOM4 value = 1): this class is responsible for two different actions: registering a new patient into the system and fetching a pre-existing patient. Both of these methods are dependent on the patient list that this fetched using a get method.

PatientListModel

PatientListView

Login Controller (LCOM4 value = 1): this class allows the person using the application to log into the system. It will take in two different variable string values in the beginning (username and password) but afterwards, will take in one more string value for a 2 Factor Authentication method which is dependent on the authentication method. Once the authentication process is over, it will return a boolean value which will either allow the user in or reject entry.

UserListModel(patient or medical staff)

LoginView

TwoFactorView

CreateAccountView