Life Prognosis Management Tool

UML Diagram Description

Use Case Diagram

The use case focuses on the interactions between Patients and Admin.
We have two actors:

1. Patient: represents a user who will interact with the system to manage their health information and view survival rates.
2. Admin: represents a privileged user who manages patient registrations and can export statistics.

Use Cases for Patient:

1. Login: the patient can log into the application/system. The login process includes verifying login credentials
2. Complete registration: the patient completes their registration process which was initialized by an admin. (includes)
3. View Survival rate: the patient can view their calculated survival rate, which was calculated in the system (includes).
4. View My Profile: patient can view their profile information
5. Edit My Profile: patient can update their profile information
6. Download .ics Schedule: patient can download their demise schedule in .ics format

Use cases for Admin:

1. Login: the admin can log into the application/system. The login process includes verifying login credentials
2. Initialize Patient Registration: the admin initiates the registration process for a new patient, which includes generating a UUID for the patient.
3. Download Statistics as Excel: the admin can download statistics and patient data in Excel format

Relationships:

Includes: this indicates that a use case contains the behavior of another use case. For example:

- **login** includes **verify login credentials**, meaning verifying credentials is a part of the login process.

- **Complete registration** includes **initialize patient registration**
- **Initialize patient registration** includes **generate UUID**
- **View survival rate** includes **calculate survival rate**

Class diagram

This class diagram represents the structure of the Life Prognosis Management Tool application.

1. User (abstract class or interface): this contains common attributes for all users: UUID, firstName, lastName, email, and password. This is an abstract class that Patient and Admin inherit from.

2. Patient (extends User): has additional attributes specific to patients: dateOfBirth, isHIVPositive, diagnosisDate, onARTDrugs, ARTStartDate, residentCountry. Methods: estimateSurvivalRate(), updateProfileData(), getProfileData().

3. Admin (extends User): this has no additional attributes. Method: getRole().

4. UserService: this manages user-related operations. Methods:
    a. initializePatientRegistration
    b. createUser
    c. verifyLoginCredentials
    d. handleBashCommands

5. Main: this represents the Application . It serves as the entry point for the application, and it's where the user accesses all the features of the app.
    Methods
    - initiatePatientRegistration(),
    - completePatientRegistration()
    - createUser()
    - verifyLoginCredentials()
    - handleBashCommands()

6. SummaryStat: this handles statistical calculations and data with attributes: totalPatients, averageAge, totalOnART, totalHIVPositive. Methods: createStatistics(), saveStatisticsToExcel().

The diagram shows the relationships between these classes:
- Patient and Admin inherit from User.
- UserManager interacts with User objects.
- SummaryStat is used by Admin for generating reports.

Activity Diagram

Our Activity diagram represent workflows of stepwise activities and actions carried out by the admin and patient

Start