

Deliverable #2

Philip Gabardo, Thomas Woudenburg, Shane Noormohamed,
Matthew Paine, Joshua Barkovic, Jonathan Beadle, Jeff Buscarino

November 2, 2013

1 Introduction

1.1 Purpose

The purpose for this document is to establish the domain and requirements of the proposed medical application: "MedConnect". This will be achieved through object oriented analysis using UML diagrams. The intended audience for this document is the developers of the system.

1.2 System Description

MedConnect will be a mobile medical records system to be used by doctors, nurses and patients. The system will be designed to maximize security for the purpose of maintaining the confidentiality, integrity and availability of the information. The system will store data remotely in a digital format, reducing the need for paper usage to keep track of patient records. By maximizing the security and portability of user information, the system will meet the users' needs for privacy while remaining efficient.

The system will be used by doctors, nurses and patients to coordinate appointments and track relevant medical information about patients. MedConnect will allow authorised users to view, create, modify, and remove patients, charts and appointments based on permissions. The system will also include a feature for doctors to plot certain patient data. All data will be stored in an encrypted form. MedConnect will restrict users from accessing certain information based on user permissions.

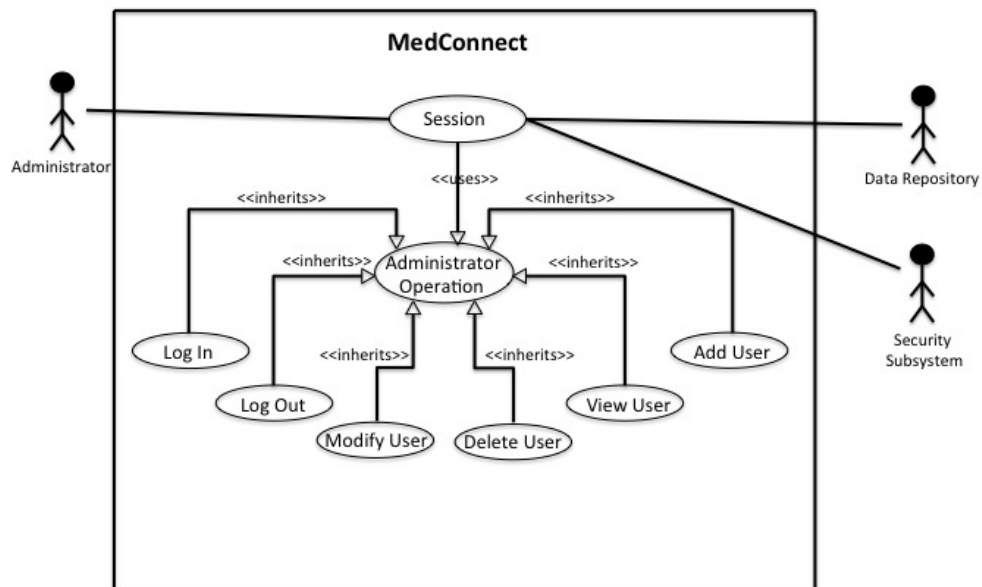
1.3 Overview

The next section of this document includes use case diagrams which illustrate the functional requirements of the system. The following section includes an analysis class diagram that will depict the overall design of the system. Next, the architectural design of the system will be defined explicitly. The final section includes Class Responsibility Collaboration (CRC) Cards for all classes included in the system. The appendix contains a signed document that confirms that contributions of each team member.

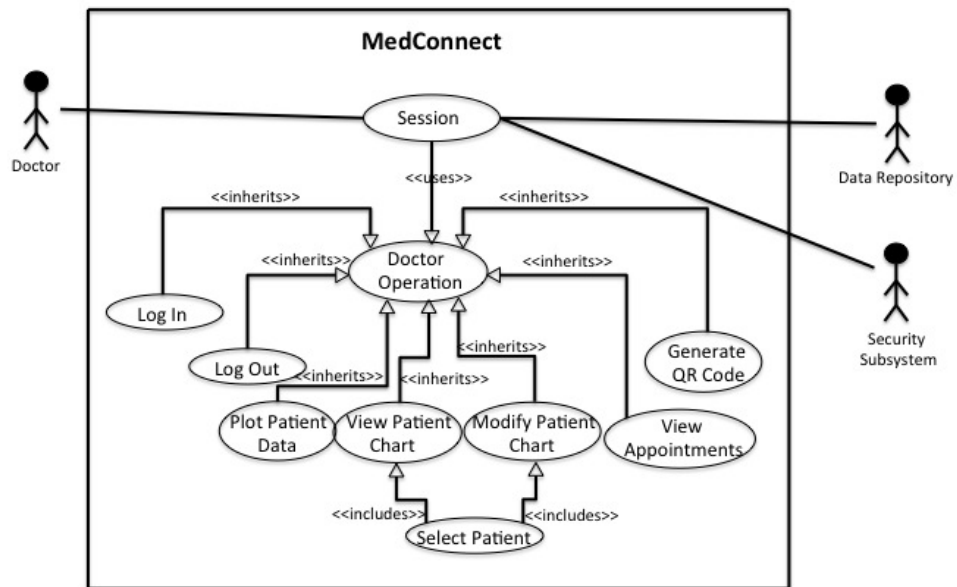
2 Use Case Diagrams

Any use cases with identical names are in fact identical use cases.

2.1 Administrator

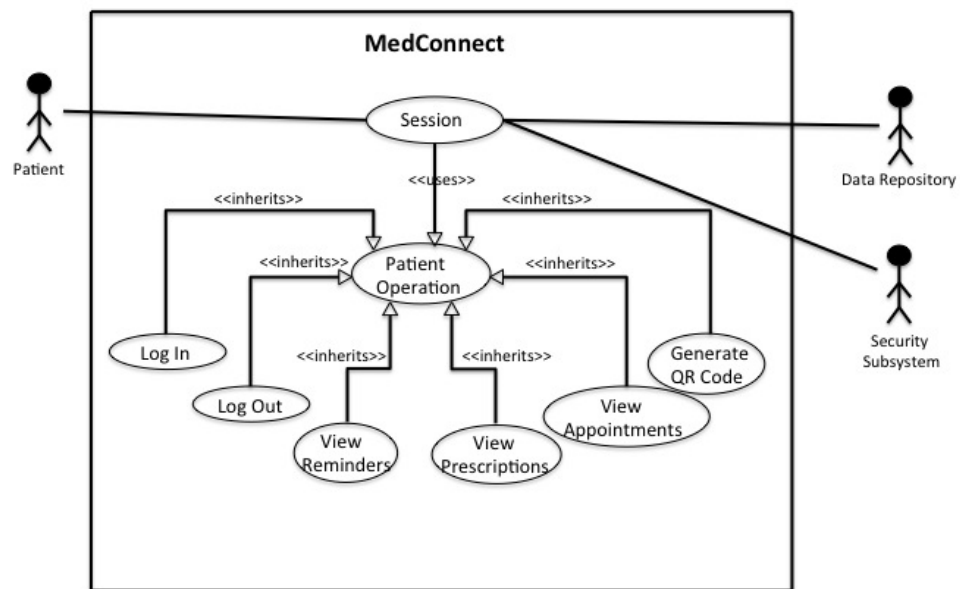


2.2 Doctor

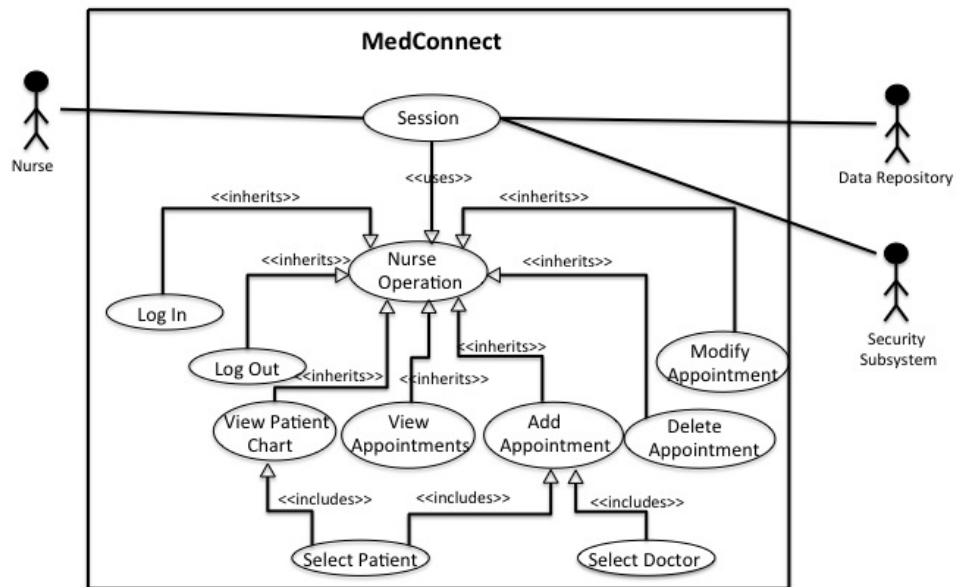


Note: In the "Select Patient" use case, the doctor either manually enters the patient's name or scans the patient's QR code.

2.3 Patient



2.4 Nurse



Note: In the "Select Patient" use case, the nurse either manually enters the patient's name or scans the patient's QR code. In the "Select Doctor" use case, the nurse either manually enters the doctor's name or scans the doctor's QR code.

3 Analysis Class Diagram

4 Architectural Design

4.1 System Architecture

4.2 Subsystems

5 Class Responsibility Collaboration (CRC) Cards

Class: AdminStartPage	
Responsibility	Collaborator
Know AdminController	
Handle button-click event of <i>Add New User</i> button	AdminController
Handle button-click event of <i>Modify User</i> button	AdminController
Handle button-click event of <i>Log Out</i> button	AdminController

Class: AddUserPage	
Responsibility	Collaborator
Know username of new user	
Know password of new user	
Know group of new user	
Know personal information (ie. first name, last name, phone number, email) of new user	
Know AdminController	
Handle button-click event of <i>Submit</i> button	AdminController
Display an error message if the username already exists in the system	

Class: ModifyUserPage	
Responsibility	Collaborator
Know username of account to modify	
Know new information of account to modify	
Know AdminController	
Handle button-click event of <i>Update</i> button	AdminController
Handle button-click event of <i>Delete</i> button	AdminController
Display an error message if the username doesn't exist in the system	

Class: AdminController	
Responsibility	Collaborator
Know Security	
Know AdminStartPage	
Know AddUser	
Know ModifyUser	
Know LoginPage	
Handle logout request	DBInterface
Add a user to the system	Security, DBInterface
Delete a user from the system	Security, DBInterface
Modify an existing user in the system	Security, DBInterface

Class: NurseStartPage	
Responsibility	Collaborator
Know NurseController	
Handle button-click event of <i>Add New Appointment</i> button	NurseController
Handle button-click event of <i>Modify Appointment</i> button	NurseController
Handle button-click event of <i>Log Out</i> button	NurseController

Class: CreateAppointmentPage	
Responsibility	Collaborator
Know NurseController	
Know information of the appointment to be scheduled	ScanQRCode
Handle button-click event of <i>Add Appointment</i> button	NurseController
Display an error message if the time of the appointment to be scheduled is in the past, or if the appointment to be scheduled conflicts with another appointment	

Class: ModifyAppointmentPage	
Responsibility	Collaborator
Display information of appointment to be modified	
Know new information of appointment to be modified	
Handle button-click event of <i>Modify</i> button	NurseController
Handle button-click event of <i>Delete</i> button	NurseController
Display an error message if the new time of the appointment to be modified is in the past, or if the new appointment information conflicts with another appointment	

Class: ScanQRCode	
Responsibility	Collaborator
Decode a QR code	

Class: NurseController	
Responsibility	Collaborator
Know Security	
Know NurseStartPage	
Know AddAppointmentPage	
Know ModifyAppointmentPage	
Know ViewPCPage	
Handle logout request	DBInterface
Add an appointment to the system	Security, DBInterface
Delete an appointment from the system	Security, DBInterface
Modify an appointment in the system	Security, DBInterface
Retrieve an appointment from the system	Security, DBInterface

Class: LoginPage	
Responsibility	Collaborator
Know username of user	
Know password of user	
Know LoginController	
Handle button-click event of <i>Submit</i> button	LoginController
Display an error message if the user's username-password pair does not exist in the system	

Class: LoginController	
Responsibility	Collaborator
Handle login request	
Know LoginPage	
Know Security	
Create AdminController	
Create PatientController	
Create NurseController	
Create DoctorController	

Class: Security	
Responsibility	Collaborator
Decrypt data	
Encrypt data	

Class: DBInterface	
Responsibility	Collaborator
Verify the existence of a username	
Add a user's username, password, group and personal information to the database	
Update a user's password, group or personal information in the database	
Delete a user's username, password, group and personal information in the database	
Add a patient's chart to the database	
Update a patient's chart in the database	
Delete a patient's chart from the database	
Add an appointment to the database	
Update an appointment in the database	
Delete an appointment from the database	

Class: DoctorStartPage	
Responsibility	Collaborator
Know DoctorController	
Handle button-click event of <i>View Patient Chart</i> button	DoctorController
Handle button-click event of <i>Modify Patient Chart</i> button	DoctorController
Handle button-click event of <i>View Appointments</i> button	DoctorController
Handle button-click event of <i>Log Out</i> button	NurseController

Class: DoctorController	
Responsibility	Collaborator
Know Security	
Know DoctorStartPage	
Know ViewPCPage	
Know ModifyPCPage	
Know ViewAppointmentsPage	
Handle logout request	DBInterface
Update a patient chart in the system	Security, DBInterface
Retrieve a patient chart from the system	Security, DBInterface
Retrieve a doctor's appointments from the system	Security, DBInterface

Class: ViewPCPage	
Responsibility	Collaborator
Know patient	ScanQRCode
Display patient's chart	

Class: PatientStartPage	
Responsibility	Collaborator
Know PatientController	
Handle button-click event of <i>View Prescriptions</i> button	PatientController
Handle button-click event of <i>View Appointments</i> button	PatientController
Handle button-click event of <i>View Reminders</i> button	PatientController
Handle button-click event of <i>Logout</i> button	PatientController

Class: PatientController	
Responsibility	Collaborator
Know Security	
Know PatientStartPage	
Know ViewPrescriptionsPage	
Know ViewAppointmentPage	
Know ViewRemindersPage	
Handle logout request	DBInterface
Retrieve a patient's prescriptions from the system	Security, DBInterface
Retrieve a patient's appointments from the system	Security, DBInterface
Retrieve a patient's reminders from the system	Security, DBInterface

Class: ViewPrescriptionsPage	
Responsibility	Collaborator
Display patient's prescriptions	

Class: ViewRemindersPage	
Responsibility	Collaborator
Display patient's reminders (appointments scheduled within the next 24 hours and expired prescriptions)	

Class: ViewAppointmentsPage	
Display patient's appointments	
Responsibility	Collaborator