COEN-6312-2184-UU

Model Driven Software Engineering

Professor: Wahab Hamou-Lhadj, Ph.D., ing.

Deliverable 2

Submitted By:

NAME	ID	EMAIL
Ashish Sharma	40050452	ashish.sharma5293@gmail.com
Harmanpreet Singh	40059358	harmansandhu63@gmail.com
Navjot Kaur Bhamrah	40050459	navjotkaurbhamrah@gmail.com
Amandeep	40046716	amandpsingh03@gmail.com
Raghav Sharda	40053703	raghavsharma2926@outlook.com
Shivya Pant	40068007	shivyapant@gmail.com

Clinic Management System

DOMAIN DESCRIPTION

Problems in healthcare are due to the increase in differentiation and miscellany. Complex methodologies and specialties are coming up each day. Health problems are also increasing with people visiting clinics and hospitals frequently. The load on these health institutions to maintain all their patient's record is also becoming a daunting task.

CMS addresses issues related to the management of a clinic. It's a standalone application where a patient can set up an appointment with a Doctor over the system, record his/her ailments and choose a Specialized Doctor depending upon his/her ailments or health complaints.

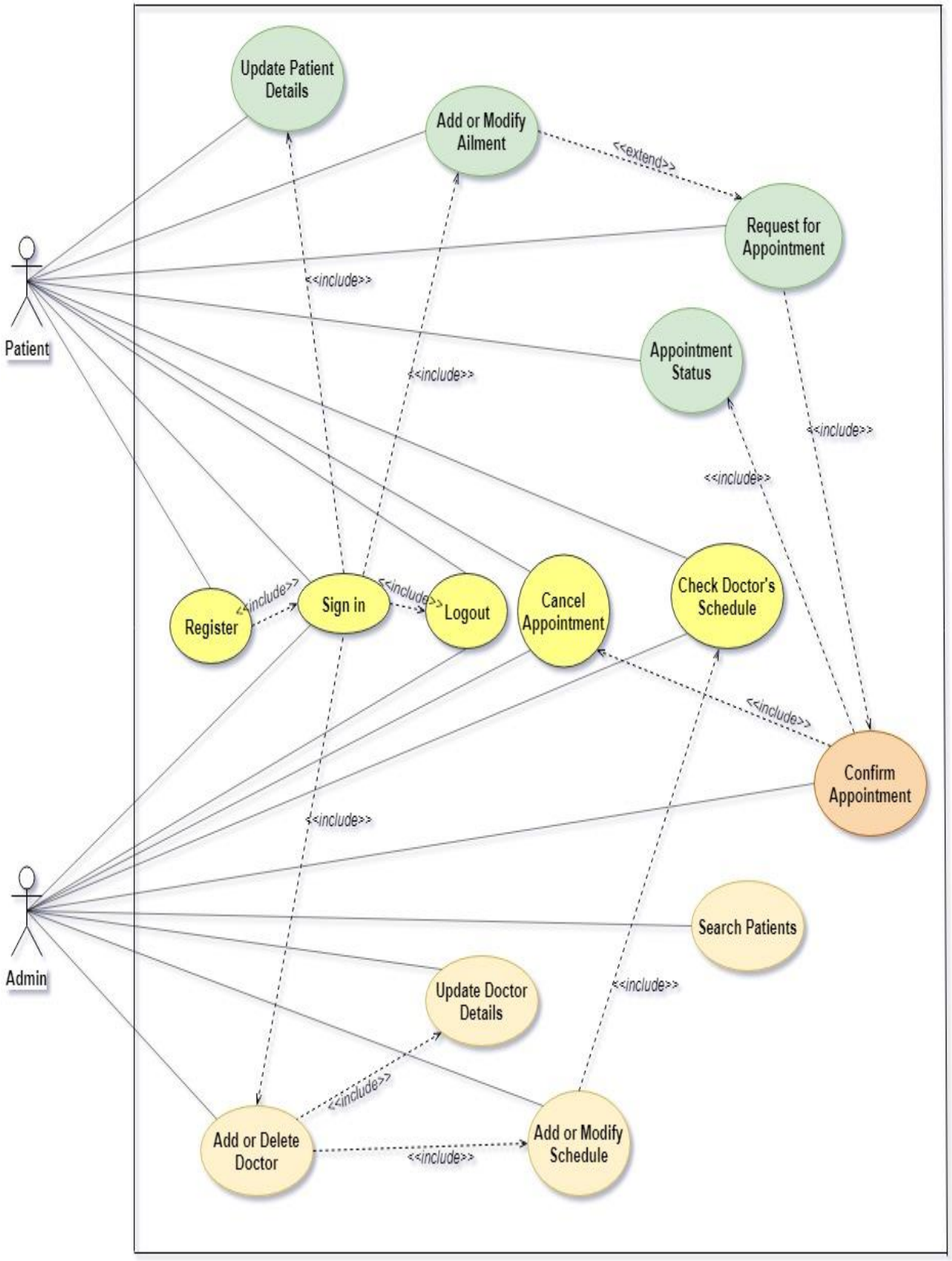
The intent of the system is to digitize patients records so that the retrieval of data can be easy and efficient. It will further help to maintain the record of patients so that it can be retrieved easily and quickly when demanded. The two significant users of the system will be the Administrator and Patients.

The application is operated by an admin and a patient. The Admin will be responsible for keeping the information about the doctors up to date for the patient by suggesting the alternative doctors.

The system helps in managing the clinic in a proper and efficient way by keeping track of all the doctors available in the clinic for the day, suggesting doctors according to the ailment, specialization. The system helps to radically reduce the paperwork in keeping the long tedious records for various patients and doctors.

The application aims to solve and helps in making the day to day clinic tasks easy and interactive.

1. USE CASE DIAGRAM



1. Register

Use case name:	UC-01 Register User
Summary:	This use case has Patient as a primary actor who creates an account in Clinic Management system to avail its services.
Basic course of events:	● The patient selects on the New Patient button to register. ● The patient enters his/her valid personal information (email id, phone number, password, etc.) ● Selects the Register button.
Alternative Paths:	NA
Exception points:	1.1 Patient leaves mandatory fields as blank ● The patient enters valid personal information. ● The patient leaves one of the mandatory fields as blank ● The patient selects the Register button 1.2 Patient enters the wrong email address ● The patient enters an invalid email address ● The patient selects the Register button.
Trigger:	Patient selects New Patient button.
Assumptions:	Patient is not registered
Preconditions:	Patient enters valid email address.
Postconditions:	Clinic management system confirms the Patient registered successfully.
Includes:	NA
Extends:	NA
Author:	Harman Sandhu
Date:	6-02-2019

2. Sign in

Use case name:	UC-02 Sign in
Summary:	This use case is being used by the Patient and Admin in order to avail and provide system services. A Patient must authenticate firstly by entering valid credentials.
Normal Flow:	● The user enters valid credentials. ● User selects the login button.
Alternative Paths:	N/A
Exception Points:	1.1 User enters a wrong credential ● System will not allow the user to access its

	services
Triggers:	User enters credentials and select login button.
Assumptions:	● A Patient is already registered ● User Authentication is performed.
Preconditions:	● User account should be activated. ● User enter valid credentials.
Postconditions:	System allows the user to access its services.
Includes:	UC-01
Extends:	NA
Author:	Navjot Kaur Bhamrah
Date:	6-02-2019

3. Logout

Use case name:	UC-03 Logout
Summary:	This use case is being used by the Patient and Doctor to exit system services. A user must log in firstly by entering valid credentials.
Normal Flow:	● User login the system. ● The user selects the Logout button.
Alternative Paths:	● User login the system. ● The user uses the system services. ● User selects the Logout button.
Exception Points:	1.2 User enters a wrong credential ● System will not allow user to access its services
Triggers:	User select logout button.
Assumptions:	● User is already log in
Preconditions:	● User should be already log in
Postconditions:	System allows the user to log out.
Includes:	UC-02
Extends:	NA
Author:	Raghav Sharda
Date:	6-02-2019

4. Update Patient Details

ID and Name	UC-04 Update Patient Details
Summary:	This use case describes the steps to update the details of the patient. Its primary actor is patient

Normal Flow:	● Patient log in the system ● The patient selects the Modify details button. ● Patient updates the personal details.
Alternative Paths:	N/A
Exception Points:	● Patient enters wrong personal details
Triggers:	● Patient selects Modify details button.
Assumptions:	Patient added wrong details while registering.
Preconditions:	Patient is already logged in.
Postconditions:	Patient details are updated
Includes:	UC-02
Extends:	NA
Author:	Amandeep Singh
Date:	6-02-2019

5. Add or Modify Ailment

ID and Name	UC-05 Add or Modify ailment
Summary:	This use case describes the steps to add or update the ailments by Patient.
Normal Flow:	● Patient log in the system ● The patient selects the Add/Modify the ailment button. ● Patient enters the ailment details.
Alternative Paths:	N/A
Exception Points:	● Patient enters wrong ailment
Triggers:	● Patient selects Add/Modify ailment button.
Assumptions:	Patient is registered and logged in.
Preconditions:	Patient is already logged in.
Postconditions:	Ailment details are updated
Includes:	UC-02
Extends:	NA
Author:	Ashish Sharma
Date:	6-02-2019

6. Request for Appointment

Use case name:	UC-06 Request for appointment
Summary:	This use case has Patient as a primary actor and can request for an appointment for his/her ailments.
Basic course of events:	● Patient log in the system. ● Patient adds his/her ailments in the system. ● The patient selects the request for the appointment

	button.
Alternative Paths:	NA
Exception points:	6.1 Patient has not added any ailment ● The patient enters valid personal information. ● Patient leaves Add/Modify ailment field as blank. 6.2 User enters wrong credentials ● The patient enters wrong password
Trigger:	Patient selects Request for appointment button.
Assumptions:	Patient properly defined his/her health complaints.
Preconditions:	Patient added his/her ailment.
Postconditions:	Admin confirms the appointment by checking doctor schedule.
Includes:	NA
Extends:	UC-05
Author:	Shivya Pant
Date:	6-02-2019

7. Confirm appointment

Use case name:	UC-07 Confirm Appointment
Summary:	This use case has Admin as the primary actor and confirms the appointment.
Basic course of events:	● Admin login the system. ● Admin checks for an appointment request. ● Admin checks the related doctor schedule. ● Admin confirms the appointment.
Alternative Paths:	NA
Exception points:	7.1 Admin confirms two appointments at same schedule ● Admin login the system. ● Admin confirm the appointment that is not in the doctor's schedule. 7.2 Admin missed the appointment request to confirm. ● Admin login the system. ● Admin didn't checked the appointment request.
Trigger:	Admin selects the confirm appointment button.
Assumptions:	Admin finds a fit schedule for the appointment request.
Preconditions:	Doctor's availability for appointment.
Postconditions:	Appointment is confirmed for specific date and time with specific doctor.
Includes:	UC-06

Extends:	NA
Author:	Harman Sandhu
Date:	6-02-2019

8. Appointment Status

Use case name:	UC-08 Appointment Status
Summary:	This use case has Patient as a primary actor in which he/she can check the status of the appointment.
Basic course of events:	● Patient log in the system. ● Patient checks the scheduled appointment.
Alternative Paths:	NA
Exception points:	8.1 There is no active appointment ● Patient log in the system. ● Patient checks for appointment status while there is no active appointment.
Trigger:	Patient selects the Appointment status button.
Assumptions:	Admin has confirmed an appointment.
Preconditions:	Appointment is scheduled by Admin
Postconditions:	Patient can come for appointment as scheduled.
Includes:	UC-07
Extends:	NA
Author:	Amandeep Singh
Date:	6-02-2019

9. Add or Delete Doctor

ID and Name	UC-09 Add or Delete Doctor
Summary:	This use case has Admin as primary actor and can add or delete Doctors of clinic from database.
Normal Flow:	● Admin login the system ● Admin Selects the Add/Delete button. ● Admin adds new doctor or delete old doctor.
Alternative Paths:	NA
Exception Points:	● There are no old doctors to delete and new one to add.
Triggers:	● Admin selects the Add/Delete Doctor button.
Assumptions:	Admin wants to add or delete a doctor.
Preconditions:	Admin is already signed in.
Postconditions:	Admin adds or deletes a doctor in/from database.
Includes:	UC-02
Extends:	NA
Author:	Ashish Sharma

Date:	6-02-2019

10. Update Doctor Details

ID and Name	UC-10 Update Doctor Details
Summary:	This use case describes the steps to update the details of the Doctor. Its primary actor is Admin
Normal Flow:	● Admin login the system ● Admin selects the Modify details button. ● Admin updates the doctor's details and adds specialization.
Alternative Paths:	N/A
Exception Points:	● Admin enter wrong details
Triggers:	● Admin selects Modify details button.
Assumptions:	Admin wants to change doctor's details.
Preconditions:	Admin has already added that doctor whose details he/she wants to update.
Postconditions:	Doctor details are updated
Includes:	UC-09
Extends:	NA
Author:	Raghav Sharda
Date:	6-02-2019

11. Add or Modify Schedule

ID and Name	UC-09 Add or Modify Schedule
Summary:	This use case describes the steps to add or update the schedule by Admin.
Normal Flow:	● Admin login the system ● Admin selects the Add/Modify schedule button. ● Admin enters the Doctor's availability.
Alternative Paths:	N/A
Exception Points:	● Admin enters wrong availability
Triggers:	● Admin selects Add/Modify schedule button.
Assumptions:	Admin has already added that specific doctor.
Preconditions:	Admin has added the doctor whose availability he/she wants to add or update.
Postconditions:	Doctor's schedule is added/updated
Includes:	UC-09
Extends:	NA
Author:	Navjot Kaur Bhamrah
Date:	6-02-2019

12. Check Doctor's Schedule

ID and Name	UC-12 Check Doctor's Schedule
Summary:	This use case has Admin and Patient as a primary actor in which both can check Doctor's schedule.
Normal Flow:	● User login the system ● The user selects the Check schedule button.
Alternative Paths:	N/A
Exception Points:	● The user enters the wrong credentials.
Triggers:	● The user selects the Check schedule button.
Assumptions:	Admin is logged in and has added a schedule.
Preconditions:	Admin has added Doctor's schedule.
Postconditions:	Doctor's availability is known.
Includes:	UC-11
Extends:	NA
Author:	Shivya Pant
Date:	6-02-2019

13. Cancel Appointment

ID and Name	UC-13 Cancel Appointment
Summary:	This use case has Admin and Patient as a primary actor in which both can cancel the appointment anytime.
Normal Flow:	● User login the system ● The user selects the specific Appointment to cancel. ● The user selects the Cancel Appointment button.
Alternative Paths:	N/A
Exception Points:	● There is no active appointment to cancel.
Triggers:	● User selects the Cancel Appointment button.
Assumptions:	● An appointment is out of the doctor's schedule. ● Patient has some emergency and cannot come to appointment ● Admin has scheduled wrong appointment
Preconditions:	There are active appointments.
Postconditions:	Selected Appointment is cancelled.
Includes:	UC-07
Extends:	NA
Author:	Harman Sandhu
Date:	6-02-2019

14. Search Patients

ID and Name	UC-14 Search Patients

Summary:	This use case can search for patients based on appointment date. Admin is the primary actor.
Normal Flow:	● Admin login the system ● Admin selects the Search patient button and then enters the date to which search is for. ● Admin can see patients for the specific date.
Alternative Paths:	N/A
Exception Points:	● There is no patient for selected date.
Triggers:	● Admin selects the Search patient button
Assumptions:	● Patient is already registered in system and.
Preconditions:	Patient is in the database.
Postconditions:	Admin can view the searched patient details.
Includes:	NA
Extends:	NA
Author:	Harman Sandhu
Date:	6-02-2019

2. FUNCTIONAL REQUIREMENTS

2.1 System Features

2.1.1 log in Account

2.1.1 Description:

To open the account the patient needs to enter login credentials.

2.1.2 Stimulus/response

The patient must enter a valid user id and password to start a user page. If it is confirmed, the user is linked to the user account page. If the user is new to the clinic he/she has to register.

2.1.3 Basic data flow

● Here first the patient enters login id and password.

● After entering the log in information system checks whether entered login id and password are valid or not.

● If it is valid then it is linked to the user account.

● If the user doesn't have a user account then the user needs to register.

2.1.4 Functional requirements

Here administrator and patients are using the same log in pages.

2.2 Admin

2.2.1 Description

Admin is the main user, able to control the whole system. Admin has been given the access to add, delete, update and modify the details in the system.

The system will have the main authentication page where the admin will put in the details to log in for the first time into the system.

2.2.3 Stimulus and response

Admin will be redirected to the page where automatic admin ID will be generated along with the option to set the new password. Admin keeps the details in the system up-to-date.

2.2.4 Basic data flow

- Admin logs into the system.
- Can add/delete/modify doctor's record.
- He/she controls the status of the patient's appointment.

2.2.5 Functional requirements

Admin has access to add/delete/modify the doctor and can view all the details.

2.3 Scheduling an appointment

2.3.1 Description:

Patients can book an appointment by registering into the application. Admin approves the appointment depending on doctors. Patient has to register or log in to book an appointment.

2.3.2 Stimulus/response

Patients should enter their personal information in the sign-up form to get registered and take an appointment. After patient books, an appointment the admin verifies the information and confirms the appointment accordingly.

2.3.3 Basic data flow

- A patient first registers into the application.
- After logging in, the patient enters the appointment information.
- The admin verifies the details from the patient and confirms the appointment.

2.3.4 Functional requirements

- Patients can book an appointment based on ailments and the doctor's specialization.
- A patient can view the appointment details and status.

3. External Interface Requirement

All the interactions of the application with patients, admin, hardware, and software are specified here.

3.1 User Interfaces

The user interface is designed in core java using swing framework. The user interface is very friendly and makes the application easily accessible.

3.2 Software Interfaces

- Operating System: Windows, Linux, Mac OS
- Front End: Core Java
- Back End: Oracle