CAN THO UNIVERSITY SOFTWARE CENTER

MEKONG DELTA - APTECH

Aptech Certified Computer Professional (ACCP)

SEMESTER II

REMEDY SCHEDULER

Instructor:
Ms. Minh Loan Le Thi

Class: CP2296F04
Team: 2
Students:

1. Student1411880 - Duy Nguyen
2. Student1411876 - Hung Nguyen Duc
3. Student1411882 - Anh Le Huy
4. Student1411874 - Loc Nguyen Phuc

Can Tho, October 2023

PROBLEM DEFINITION OF PROJECT

The current operational status at Remedy Hospital involves manual recording of patient information and medical appointments, as well as some other activities. Operating in this manner poses several challenges, including the risk of errors due to manual data entry, difficulties in managing patient information, appointment scheduling, and limited scalability.

To address these issues, the hospital is planning to develop a desktop application for automated data management, appointment scheduling, and shift organization to improve efficiency and accuracy.

The current operational method at Remedy Hospital faces several limitations:

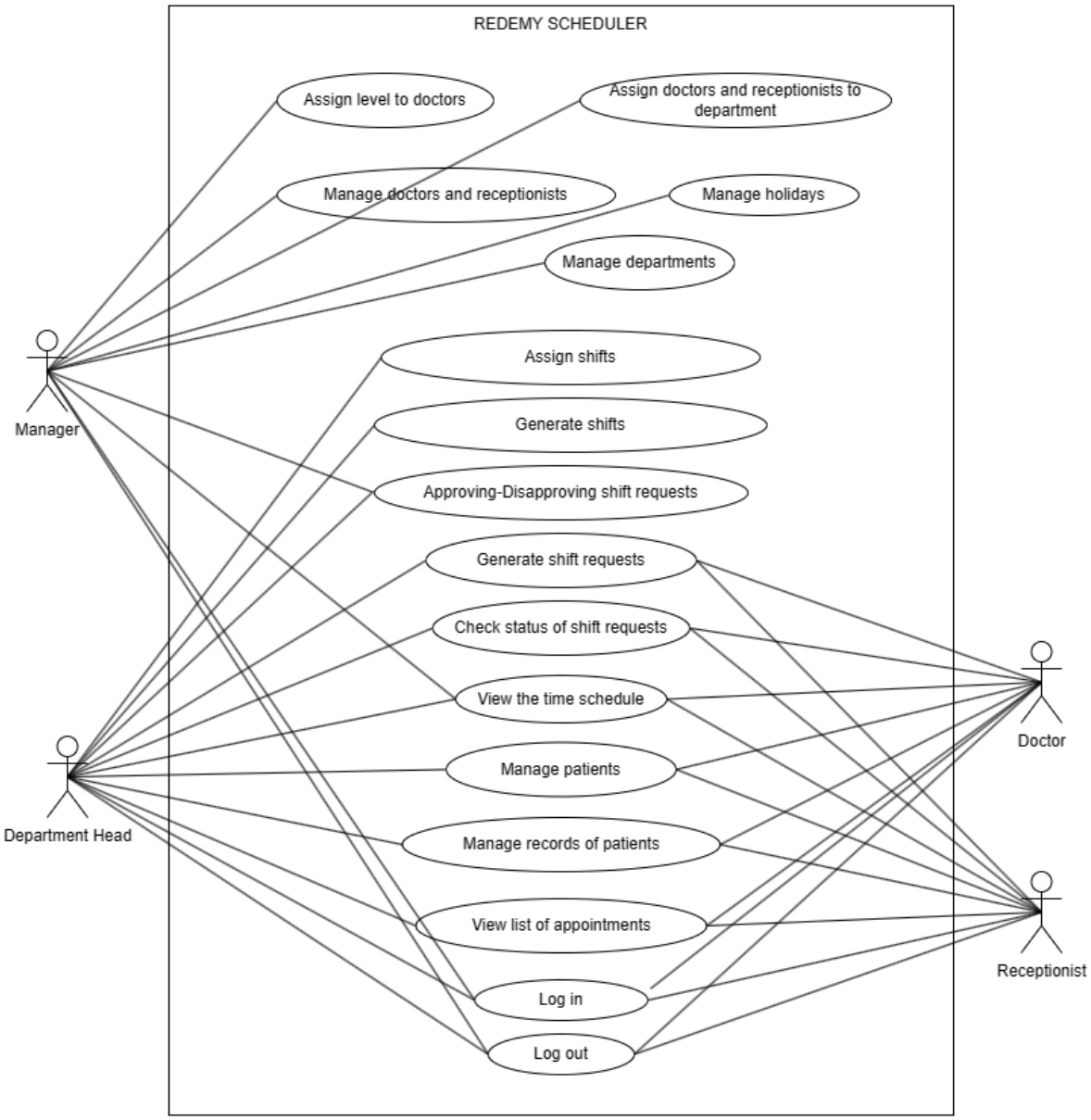
- **Processing and Access Time:** Manual recording in registers slows down the process of accessing information, causing delays and inefficiencies in data management
- **Potential for Errors:** Manual data entry increases the likelihood of errors in patient information, appointments, and doctor names, making it challenging to maintain accurate records
- **Poor Service Performance:** Reliance on manual records sometimes leads to multiple doctors caring for a single patient, affecting the quality of service and patient satisfaction
- **Appointment and Time Management Issues:** Manual scheduling introduces inaccuracies in appointment scheduling and time management, resulting in inefficient scheduling
- **Limited Scalability:** The manual system struggles to expand to meet the increasing demands for healthcare and patient management
- **Inconsistencies in Information:** Managing information across various sources leads to inconsistencies in patient details, appointments, and treatment outcomes

We will develop a system with different user roles: Manager, Department Head, Receptionist and Doctor and our application will have functions related to employee management, patient and medical record management, employee work schedule management, and some additional features.

CUSTOMER REQUIREMENTS SPECIFICATION (CRS)

Name of Project: Remedy Scheduler

1. Modeling requirements of customer (Use case Diagram)



2. Actor Description

No.	Actor	Use Case	Description
1	Manager	Assign level to doctors	This module allows Managers to assign the Doctor change to Department Head
		Assign doctors and receptionists to Department	This module allows Managers to assign the department into which the doctor or receptionist has to serve
		Manage doctors and receptionists	This module allows Managers to insert, update, delete and search for records of doctors and receptionist
		Manage holidays	This module allows Managers to add, alter, delete and view all holidays
		Manage departments	This module allows Managers to insert, update, delete and search for information of departments
		Approving-Disapproving shift requests	This module allows Managers to approve or reject the shift request generated by Department Heads
		View the time schedule	This module allows the Managers to view the time schedule of employees, i.e. shifts assigned by Department Heads
		Log in	This module allows Managers to log in to the application
		Log out	This module allows Managers to log out of the application
2	Department Heads	Generate shifts	This module allows Department Heads to generate shifts
		Assign shifts	This module allows Department Heads to assign shifts to suitable employees
		View the time schedule	This module allows Department Heads to view the time schedule of employees, i.e. shifts assigned by the manager

		Generate shift requests	This module allows Department Heads to generate a request for a shift exchange
		Check status of shift requests	This module allows Department Heads to check the status of the shift request e.g. approved, rejected or pending
		Approving-Disapproving shift requests	This module allows Department Heads to approve or reject the shift request as generated by the subordinates
		Manage patients	This module allows the Department Heads to insert, update, delete and search for information of patients
		Manage records of patients	This module allows the Department Heads to insert, update, delete and search for records of a patient
		View list of appointments	This module allows the Department Heads to view the list of appointments of employees
		Log in	This module allows Department Heads to log in to the application
		Log out	This module allows Department Heads to log out of the application
3	Doctor and Receptionist	View the time schedule	This module allows Doctors and Receptionist to view the time schedule of employees, i.e. shifts assigned by the manager
		Generate shift requests	This module allows Doctors and Receptionist to generate a request for a shift exchange
		Check status of shift requests	This module allows Doctors and Receptionist to check the status of the shift request e.g. approved, rejected or pending
		Manage patients	This module allows Doctors and Receptionist to insert, update, delete and search for information of patients

		Manage records of patients	This module allows Doctors and Receptionist to insert, update, delete and search for records of a patient
		View list of appointments	This module allows Doctors and Receptionist to view the list of appointments of employees
		Log in	This module allows Doctors and Receptionists to log in to the application
		Log out	This module allows Doctors and Receptionists to log out of the application

3. Use case Description

3.1. Assign level to doctors: This module allows Managers to assign the level of employees

- Purpose: This function allows the Manager to change employee's position from Doctor to Department Head
- Inputs: Doctor ID
- Outputs: Position of the Doctor change to Department Head

3.2. Assign doctors and receptionists to department: This module allows Managers to assign the department into which the doctor or receptionist has to serve his service

- Purpose: This function allows Managers to assign a doctor or receptionist to a specific department
- Inputs: Employee ID, department ID
- Outputs: Information about doctors and receptionists is added to the specific department

3.3. Manage holidays: This module allows Managers to add, alter, delete and view all the records of the holiday

- Add record:
  - Purpose: This function allows adding new record
  - Inputs: Holiday ID, holiday name, start date, end date, description
  - Outputs: New record added
- Alter record:
  - Purpose: This function allows alter record

        - Inputs: Holiday ID, new information
        - Outputs: Record updated
    - Delete record:
        - Purpose: This function allows delete record
        - Inputs: Holiday ID
        - Outputs: Record Deleted
    - View all record:
        - Purpose: This function allows viewing all records
        - Inputs: List of record
        - Outputs: Show all records

3.4. Log in: This module allows all to logging in
- Purpose: This function allows logging in
- Inputs: Username, password
- Outputs: Logging in to the application with permission

3.5. Log out: This module allows all to logging out
- Purpose: This function allows logging out
- Inputs: Confirm logging out
- Outputs: Log out of the application

3.6. Assign shifts: This module allows Department Heads to assign shifts to suitable employees
- Purpose: This function allows Department Heads to assign shifts to suitable employees
- Inputs: Employee ID, shift ID
- Outputs: Shift is assigned to the employee

3.7. Generate shifts: This module allows Department Heads to generate shifts
- Purpose: This function allows Department Heads to generate shifts
- Inputs: Department ID, room ID, date of the shift, type
- Outputs: New shift generated

3.8. View the time schedule: This module allows all to view the time schedule of employees
- Purpose: This function allows users to view one’s work schedule
- Inputs: Employee ID
- Outputs: Show the time schedule

3.9. Generate shift requests: This module allows all except Managers to generate a request for a shift exchange

- Purpose: This function allows to generate the request for the shift
- Inputs: Current shift ID, desired shift ID
- Outputs: The request of the shift is created and waiting for confirmation

3.10. Check status of shift requests: This module allows all except Managers to check the status of the shift request e.g. approved, rejected or pending

- Purpose: This function allows checking if shift requests have been approved or not
- Inputs: Request ID
- Outputs: State of shift request e.g. Approved, Pending, Rejected.

3.11. Disapproving-Approving shift requests: This module allows the Managers or Department Heads to approve or reject shift requests generated by subordinates

- Purpose: This function allows Managers and Department Heads to approve or reject shift requests generated by subordinates
- Inputs: Request ID, approve or reject
- Outputs: Update status of the shift request

3.12. View list of appointments: This module allows all except Managers to view list of appointments of employees

- Purpose: This function allows to view list of appointments including information about patient, time, etc.
- Inputs: Employee ID
- Outputs: List of appointments

3.13. Manage doctors and receptionists: This module allows all except Managers to insert, update, delete and search for records of doctors and receptionists

- **Insert record:**
  - Purpose: This function allows adding new records
  - Inputs: Information about doctors and receptionists include name, department, gender, position, address, date of birth, phone, email, specialty, username and password auto generated for account

        - Outputs: New record is added
    - **Update record:**
        - Purpose: This function allows updating records
        - Inputs: Record ID, new information
        - Outputs: Record is updated
    - **Delete record:**
        - Purpose: This function allows deleting records
        - Inputs: Record ID
        - Outputs: Record is deleted
    - **Search record:**
        - Purpose: This function allows searching for records
        - Inputs: Record ID
        - Outputs: Show record

3.14. Manage departments: This module allows Managers to insert, update, delete and search for information of departments

- **Insert department:**
    - Purpose: This function allows adding new departments
    - Inputs: Department ID, department name
    - Outputs: New record is added
- **Update department:**
    - Purpose: This function allows updating departments
    - Inputs: Department ID, new information
    - Outputs: Record is updated
- **Delete record:**
    - Purpose: This function allows deleting departments
    - Inputs: Department ID
    - Outputs: Record is deleted
- **Search department:**
    - Purpose: This function allows searching for departments
    - Inputs: Department ID
    - Outputs: Show record

3.15. Manage patients: This module allows all except Managers to insert, update, delete and search for information of patients

- **Insert patient:**

- Purpose: This function allows adding new information about patient
- Inputs: Information about patients including ID, name, gender, phone, address, date of birth, note
- Outputs: New information about the patient is added

- **Update patient:**
  - Purpose: This function allows updating records
  - Inputs: Patient ID, new information
  - Outputs: Information about the patient is updated
- **Delete patient:**
  - Purpose: This function allows deleting records.
  - Inputs: Patient ID
  - Outputs: Information about the patient is deleted
- **Search patient:**
  - Purpose: This function allows searching records
  - Inputs: Patient ID
  - Outputs: Show information about the patient

3.16. Manage records of patients: This module allows all except Managers to insert, update, delete and search for records of patients

- **Insert record:**
  - Purpose: This function allows adding new records
  - Inputs: Information about the patient including patient ID and the problem
  - Outputs: Record id added
- **Update record:**
  - Purpose: This function allows updating records
  - Inputs: Patient ID, new information
  - Outputs: Record is updated
- **Delete record:**
  - Purpose: This function allows deleting records
  - Inputs: Patient ID
  - Outputs: Record is deleted
- **Search record:**
  - Purpose: This function allows searching for records
  - Inputs: Patient ID

- Outputs: Show record

4. Hardware and Software required for implementing the project

	Server	Client
Hardware	Processor speed: 1.4GHz or faster Processor type: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support RAM: 2GB Hard disk space: 20GB Graphics card: DirectX 9 with WDDM 1.0 driver or later Display: 800 x 600 or higher resolution	Processor speed: 1GHz or faster RAM: 2GB Hard disk space: 20GB Graphics card: DirectX 9 with WDDM 1.0 driver or later Display: 800 x 600 or higher resolution
Software	JDK 20, JRE 20.0.1, Microsoft SQL Server 2022, SQL Server Management Studio 19, SQL Server 2022 Configuration Manager, Windows 10	JDK 20, JRE 20.0.1, Windows 10
Additional Technology	Microsoft JDBC Driver for SQL Server	

CUSTOMER ACCEPTANCE CRITERIA

No.	Customer's Acceptance Criteria
1	The application is executable and can be installed on computer running Windows OS
2	The application has different UI for different users to interact with it and use its features
3	The system has several roles that can interact with the database and have different levels of administration e.g. manager, department head, doctor,…
4	The application only allows users who have appropriate role to use its features
5	The application have features to manage the schedule by adding, deleting, updating,…
6	The application has the feature to assign the level of doctor for managers
7	The application has the feature to assign a doctor to a department for managers
8	The application has the feature to manage doctors and receptionists for managers
9	The application has the for managers to manage holidays
10	The application has the feature to manage departments for managers
11	The application has the feature to assign shifts to doctors for department heads
12	The application has the feature to generate shifts for department heads
13	The application has the feature to approve or reject shift change requests for department heads and managers
14	The application has the feature to generate shift change requests for department heads, doctors and receptionist
15	The application has the feature to check status of shift change requests for department heads, doctors, and receptionists
16	The application has the feature for users to view the time schedule
17	The application has the feature to manage patients for department heads, doctors and receptionists
18	The application has the feature to manage patients' health records for department heads, doctors and receptionists
19	The application has the feature for department heads, doctors and receptionists to view the list of appointments
20	The application has the feature for users to log in and log out

PROJECT PLAN

I. Project Details

1. **Name of the Client:** Remedy Hospital
2. **Date of the Project Plan:** 2023/08/24 to 2023/09/29
3. **Project Vision/Objectives (Define the project vision/objectives as stated by the client):**
   - Project Vision:
     - The project aims to improve Remedy Hospital's operations with systematic patient information management, appointment scheduling, and doctor assignment with the help of computers to enhance efficiency and accuracy in patient service. Simultaneously, the project aims to reduce errors in information management and enhance the patient experience
   - Specific Project Objectives:
     - Employee Assignment: The project will support doctor assignment based on qualifications and experience within the hospital. This facilitates easier doctor management
     - Employee Information Management: The system will assist in managing doctor information, including personal details, department, and specialty
     - Enhanced Service Quality: By systematically managing patient information, medical records, and assigning doctors, the project aims to improve service quality and minimize errors related to patient information management
     - The project will achieve these objectives by developing a desktop application for patient information management and doctor shift scheduling
4. **Scope (Mention the scope of the project giving the location that will be covered, processed, range of services, and so on):**
   - The project includes functions for patient management, doctor management, work shifts management, and appointment scheduling:
     - Patient management function allows the creation, updating, and deletion of patient records, as well as searching for patient information and managing patient medical records

- Employees management function involves creating doctor profiles, categorizing them based on qualifications and specialties, and assigning them to appropriate departments
- Work shifts management function enables the creation and checking of doctor availability in work shifts
- Appointment scheduling and calendar management function includes managing work shifts, handling shift change requests, viewing appointment schedules, tracking vacation schedules, and managing work schedules

5. Our understanding of the client organization (Give the range of services, functions, overview of processes, and so on):

- Manager: Manage department, manage employees, manage shifts, manage holidays, view time schedule
- Department Head: Manage patient and health record, manage shift, view appointment, view time schedule
- Doctor: Manage patient and health record, view appointment, view time schedule, Create shift change request and check status of change request
- Receptionist: Manage patient and health record, view appointment, view time schedule, Create shift change request and check status of change request

6. Project Organization with Responsibilities and Authorities (Give the name of Project team members their roles and responsibilities):

No.	Member	Roles	Responsibilities
1	Duy Nguyen	Leader	Requirements, Design, Development, Testing, Deployment
2	Hung Nguyen Duc	Member	Requirements, Design, Development, Testing, Deployment
3	Anh Le Huy	Member	Requirements, Design, Development, Testing, Deployment

4	Loc Nguyen Phuc	Member	Requirements, Design, Development, Testing, Deployment

II. Project Initiation/Requirement Documents (Information required from the client as inputs regarding his/her system; could be the information about the services, processes, and so on)

While researching the requirements of Remedy Hospital, we also examined the VIETTEL-HIS application to understand the functionalities required for a hospital management system. Additionally, we listened to the client's requirements to make the application more comprehensive and tailored to their specific needs.

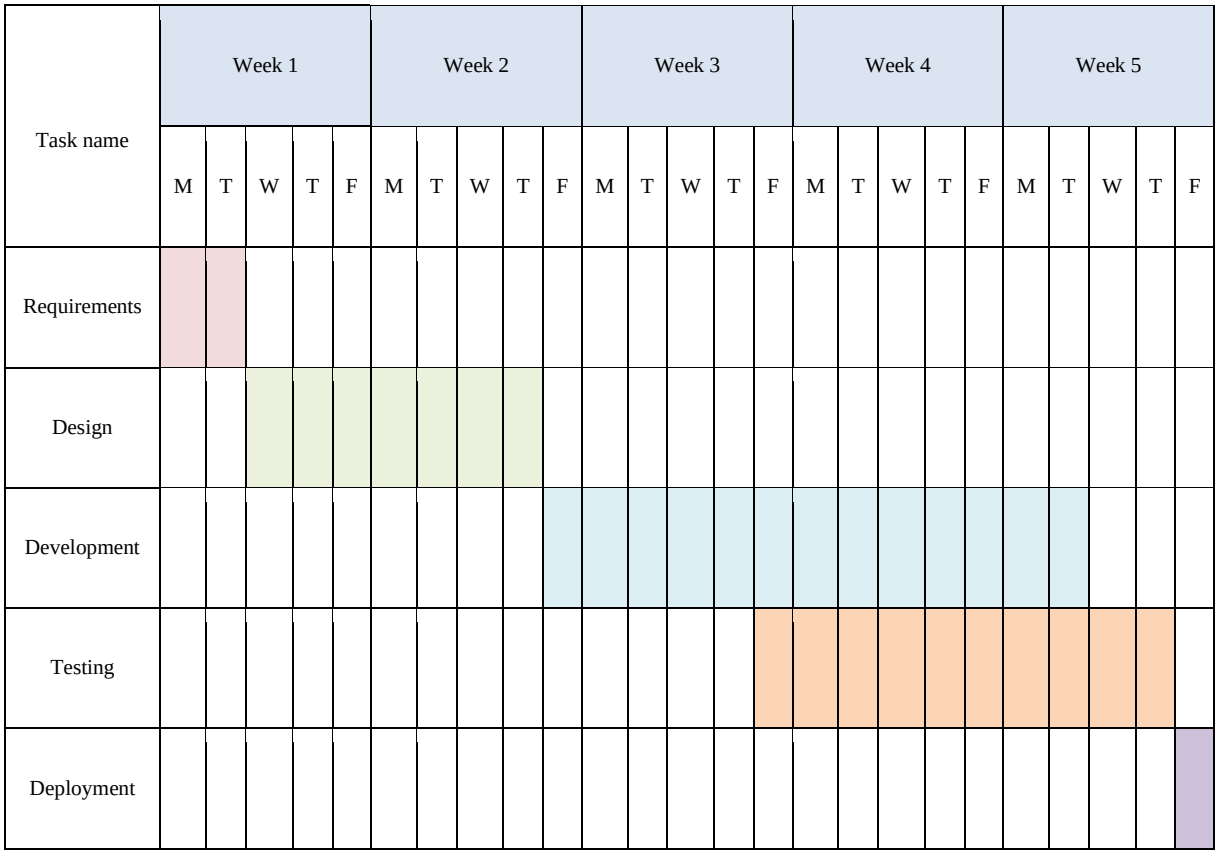
III. Deliverable (The documents to be handed over to the client - such as CRS, Design Document, Installation Manual, User Manual, Maintenance Manual, and Code of documents)

After completing the project, the following will be handed over to the client: User Manual, Installation Guide, Source Code, Documentation.

IV. Project Dependencies (Any event or task outside the scope of the project which will affect the success of the project)

The events encountered during project execution: unexpected computer errors, code consolidation without synchronization leading to non-functionality, varying progress among team members resulting in timeline discrepancies.

V. Major Project Milestones (Generating CRS, Building a Prototype, Gartt - describe the main milestones of the project)



VI. Quality Plan

1. **Review Activities (Review meeting participants, frequency, and so on)**

The project team will meet once a week to consolidate code and provide unified solutions for encountered issues.

2. **Testing Activities (Final Test)**

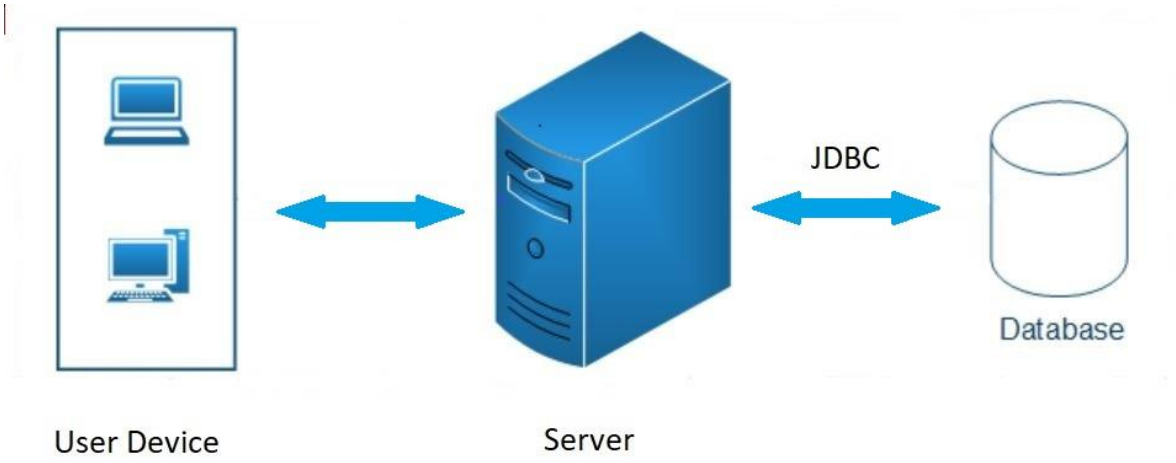
Unit Test.

3. **Backup and Recovery Strategies (In case of disk crash, network failures, and so on)**

Each member stores their code individually, and the team uses a shared GitHub for this project to update their code after each addition, modification, or deletion. GitHub supports data backup, so when needed, we can recover it.

ARCHITECTURE & GUI STANDARD OF APPLICATION

1. Architecture of Application



2. GUI Standard Application

Property	Value
Document Theme and Color Scheme	White-Blue
Form - Background Color	White
Title - Font Size	18
Title - Font Color	Blue
Title - Font Style	SansSerif
Title - Alignment	Center
Background color of Controls on the form	White
Foreground color of Controls on the form	Black
Control Caption - Font Size	14
Control Caption - Font Color	Black
Control Caption - Font Style	Bold
Control caption and controls - Alignment	Left
Command button - Alignment	Left, Right

Interface Design Document

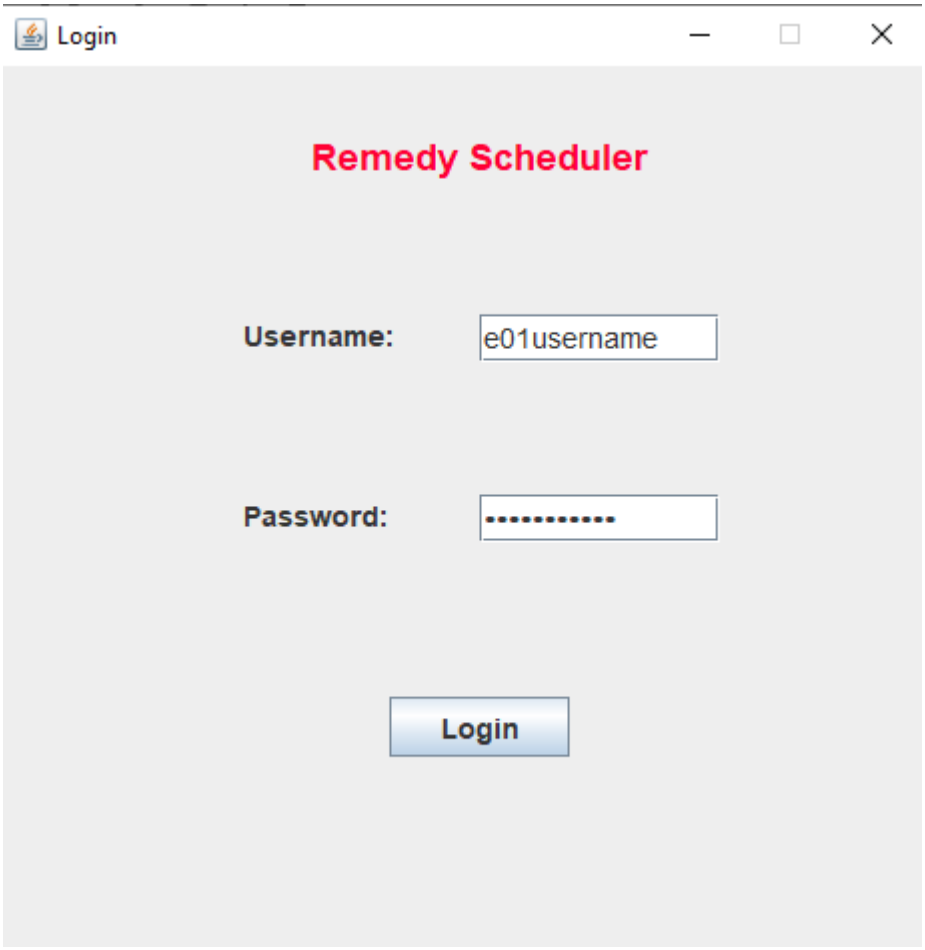
5.1. Document Name: Login form

5.1.1. Description: This form allows user to access to the Remedy Application through using account for login form.

5.1.2. Control on the Document

- Username: Not empty, no only-space
- Password: Not empty

5.1.3. UI Design



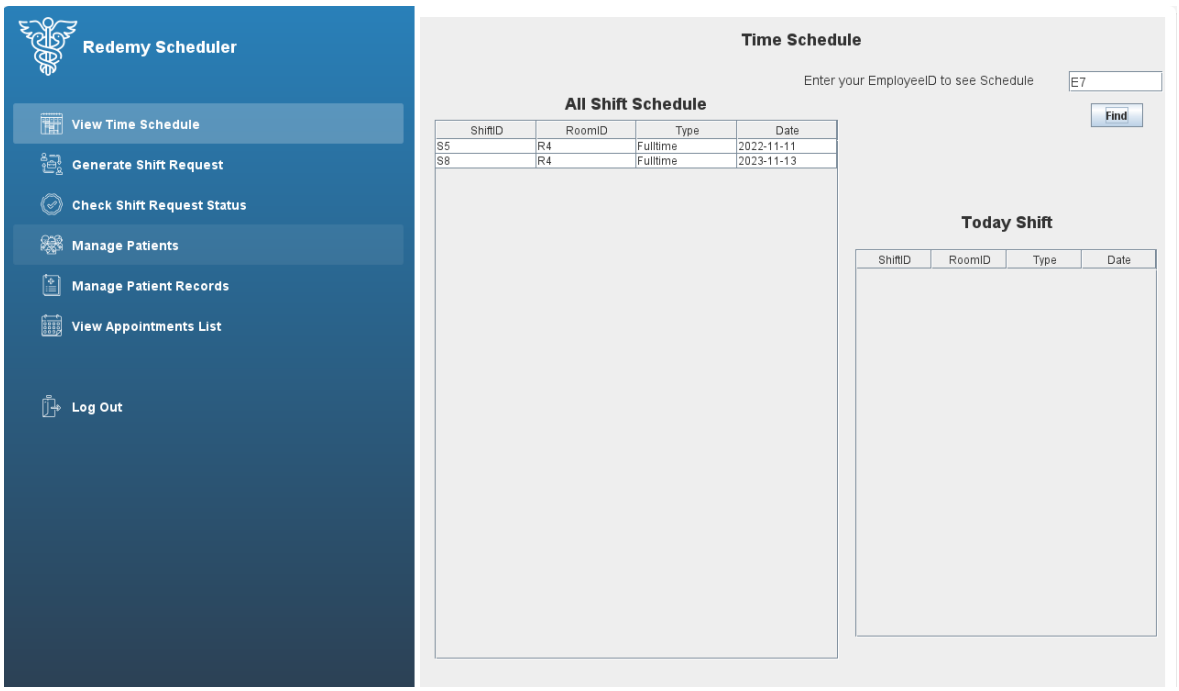
5.2. Document Name: View Time Schedule

5.2.1. Description: This form allows user to view shift time for all shift the employee has and also they can view today shift to know if today the employee has shift to work

5.2.2. Control on the document:

- EmployeeID entered: Not empty

5.2.3. UI Design



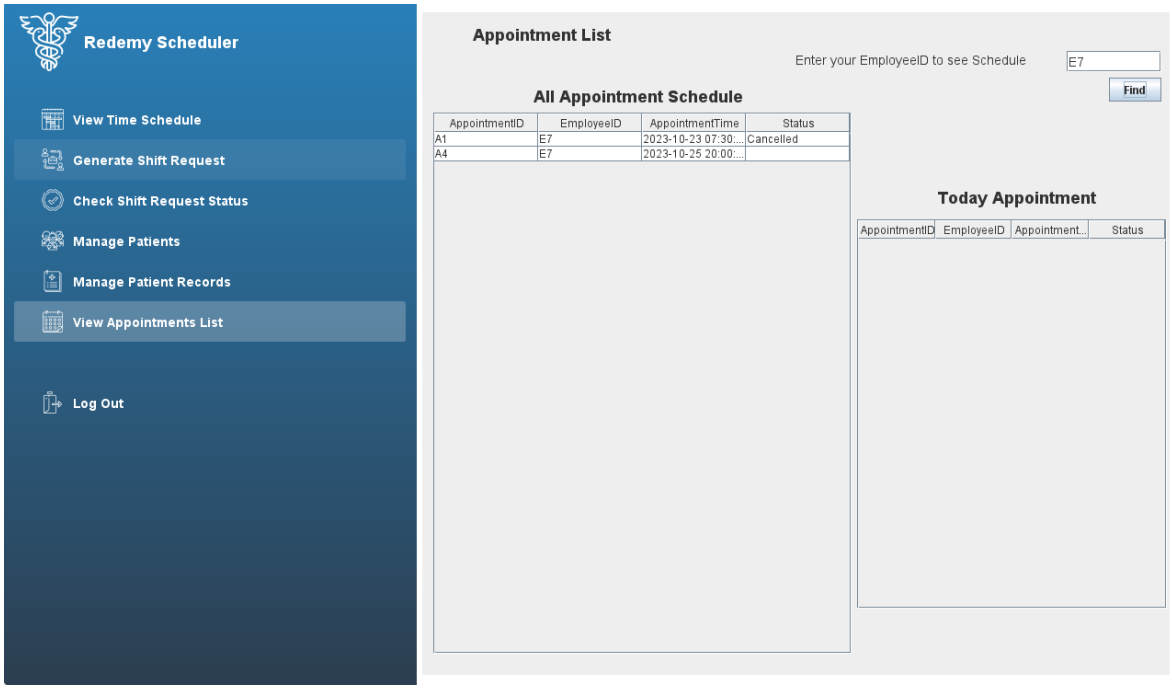
5.3. Document Name: View Appointment List Form

5.3.1. Description: This form allows user to view appointment time for all appointment the employee has been set up to and also they can view today appointment to know if today the employee has appointment to meet

5.3.2. Control on the Document:

- EmployeeID entered: Not empty

5.3.3. UI Design



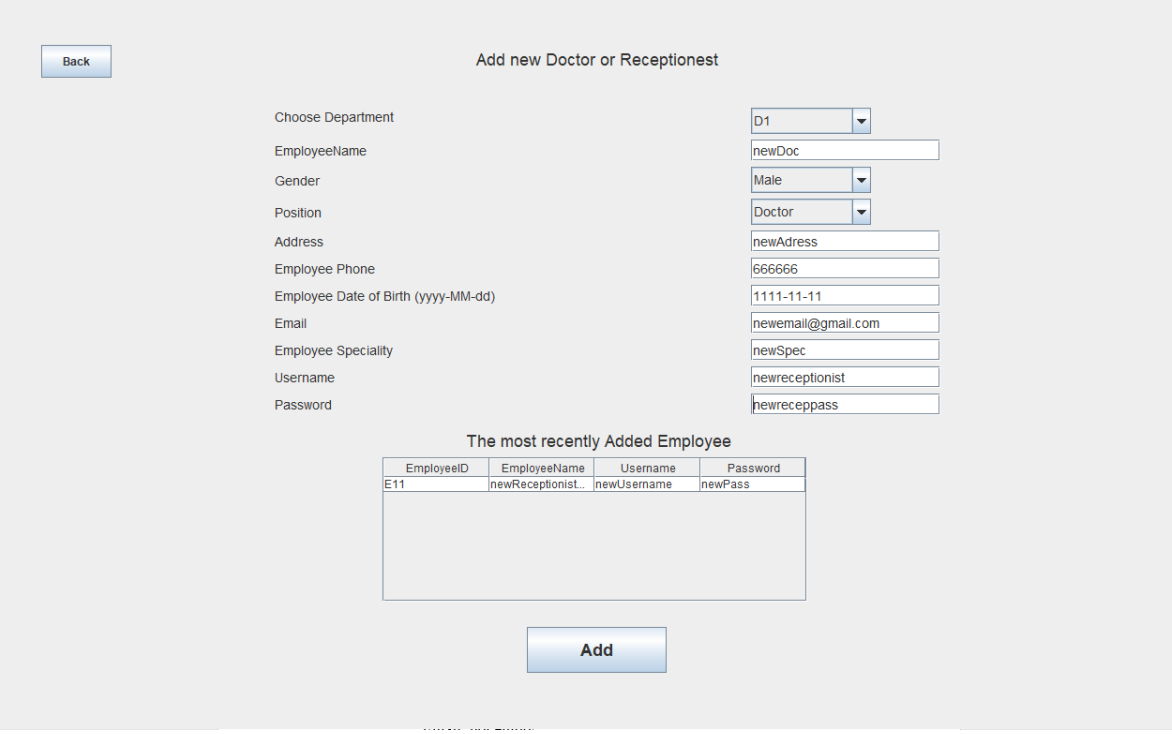
5.4. Document Name: Add Doctor and Receptionist Form

5.4.1. Description: This form allows the user to Add a new Employee which can be a Doctor or a Receptionist

5.4.2. Control on the Document:

- EmployeeName: Not empty
- Address: Not empty
- EmployeePhone: Not empty
- Employee Date of Birth: Not empty, valid date-format (yyyy-MM-dd)
- Email: Not empty
- Employee Specialty: Not empty
- Username: Not empty, Not duplicated
- Password: Not empty

5.4.3. UI Design:



5.5. Document Name: Alter Doctor and Receptionist Form

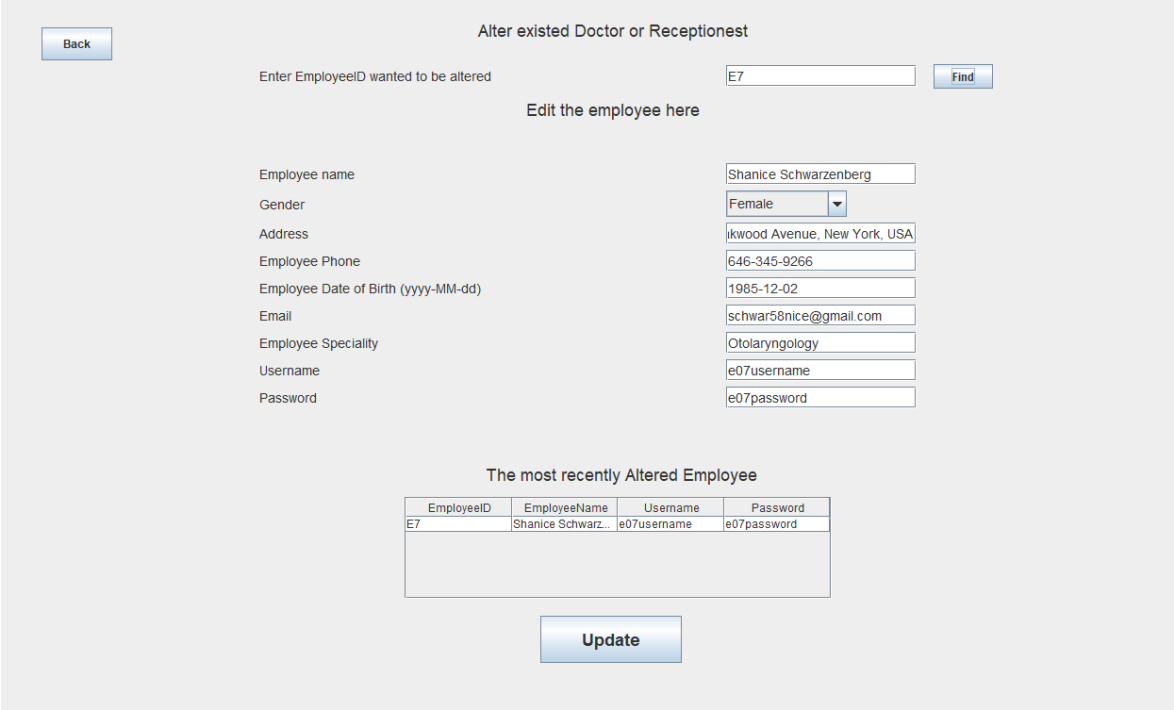
5.5.1. Description: This form allows the user to alter and edit the existing Employee which can be a doctor or a receptionist

5.5.2. Control on the Document

- EmployeeID entered: Not empty
- EmployeeName: Not empty
- Address: Not empty

- EmployeePhone: Not empty
- Employee Date of Birth: Not empty, valid date-format (yyyy-MM-dd)
- Email: Not empty
- Employee Specialty: Not empty
- Username: Not empty, not duplicated
- Password: Not empty

5.5.3. UI Design:



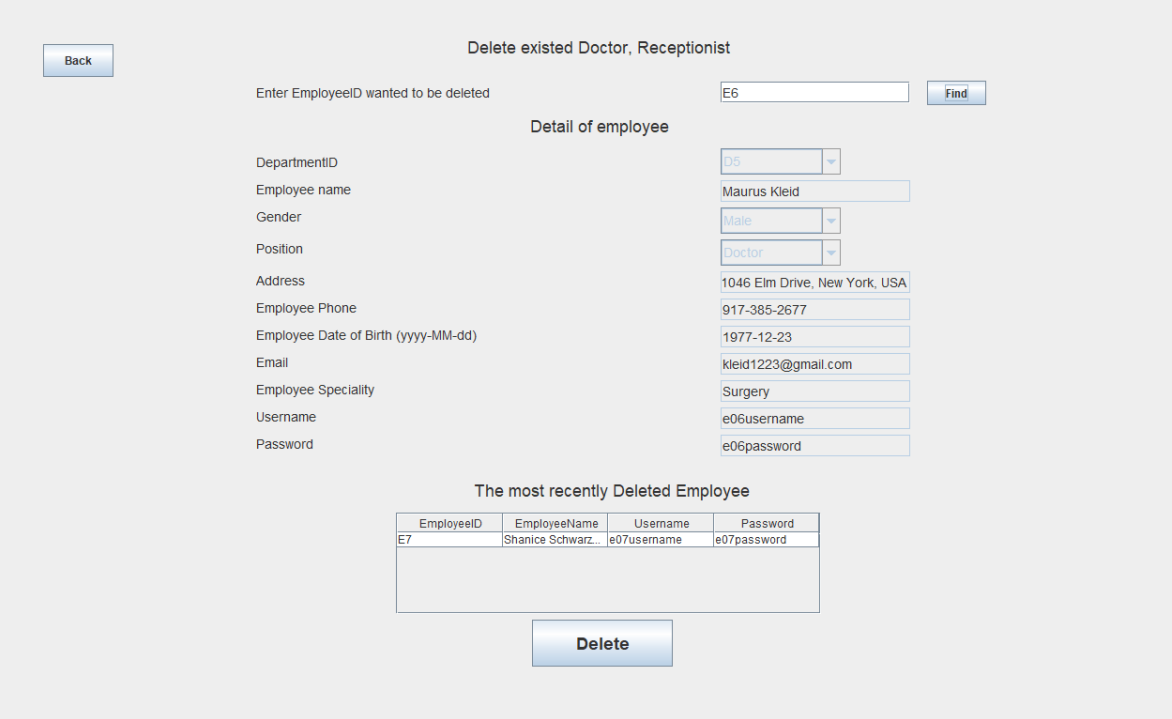
5.6. Document Name: Delete Doctor and Receptionist Form

5.6.1. Description: This form allows the user to delete the existing Employee which can be a doctor or a receptionist

5.6.2. Control on the Document:

- Employee ID entered: Not empty

5.6.3. UI Design:



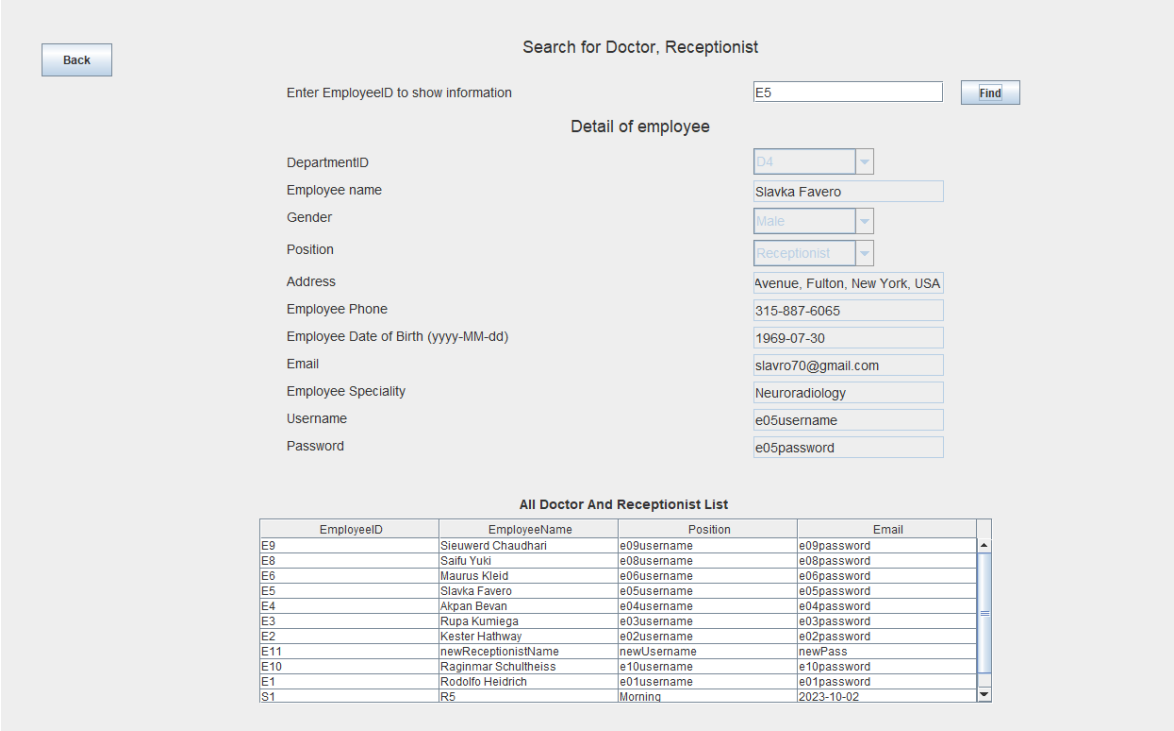
5.7. Document Name: Search Doctor and Receptionist Form

5.7.1. Description: This form allows the user to search the existing Employee which can be a doctor or a receptionist or manager or department head. Also user can see the short information of all employee existing in Employee table

5.7.2. Control on the Document:

- Employee ID entered: Not empty

5.7.3. UI Design:



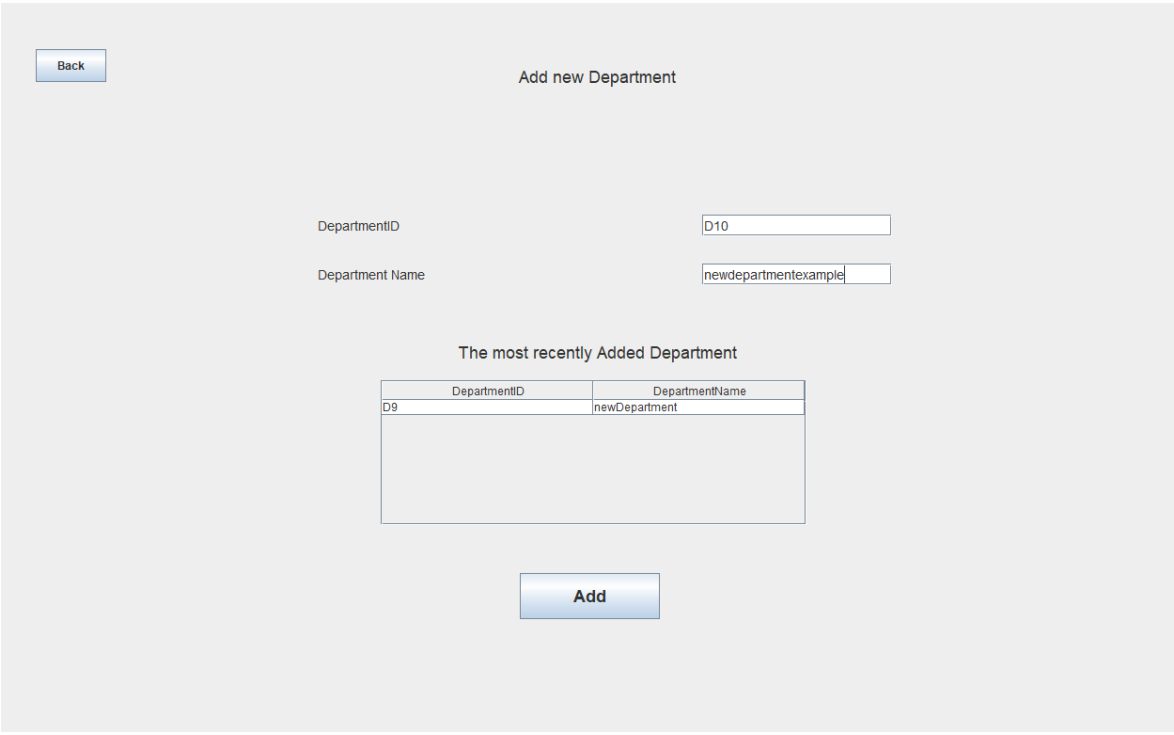
5.8. Document Name: Add Department Form

5.8.1. Description: This form allows the user to Add a new Department

5.8.2. Control on the Document:

- DepartmentID: Not empty, not duplicated
- Department Name: Not empty

5.8.3. UI Design:



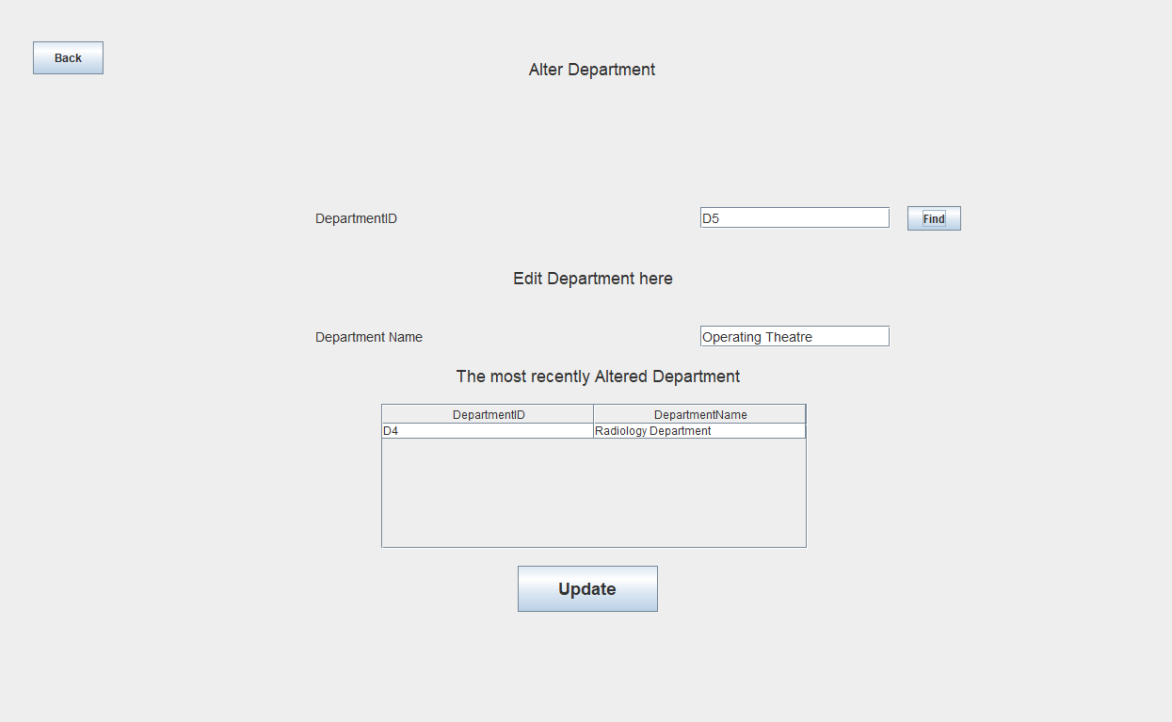
5.9. Document Name: Alter Department Form

5.9.1. Description: This form allows the user to alter an existing Department

5.9.2. Control on the Document:

- DepartmentID: Not empty
- Department Name: Not empty

5.9.3. UI Design:



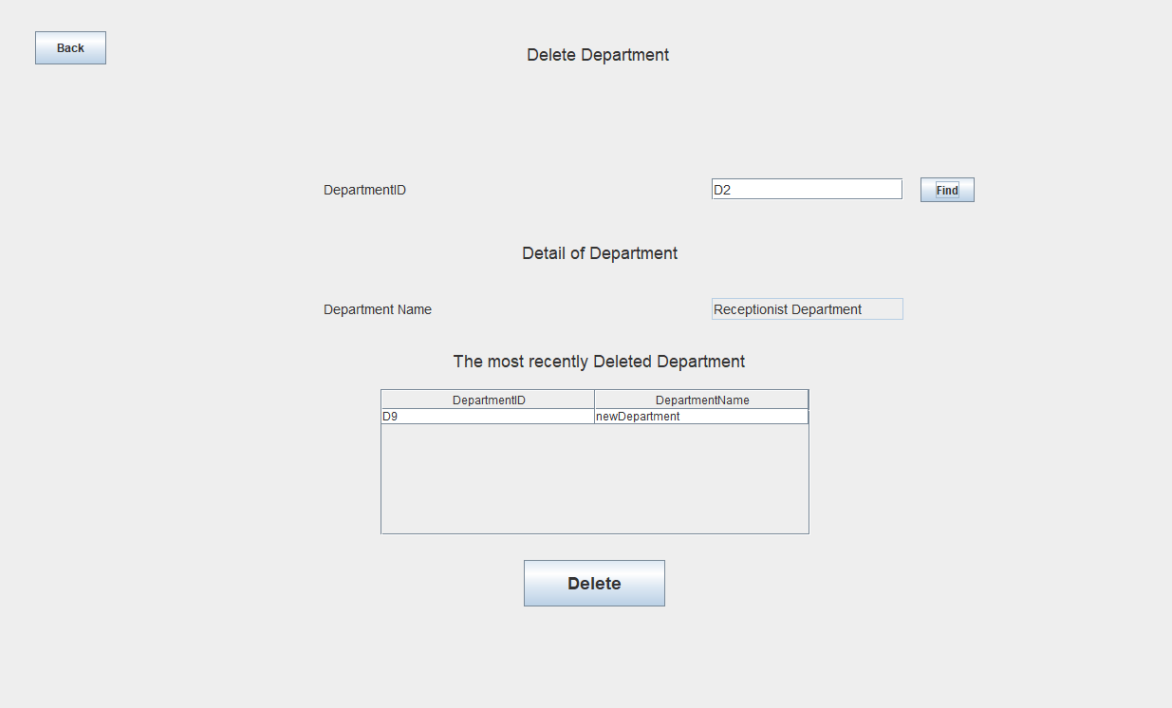
5.10. Document Name: Delete Department Form

5.10.1. Description: This form allows the user to delete the existing Department

5.10.2. Control on the Document:

- DepartmentID: Not empty

5.10.3. UI Design:



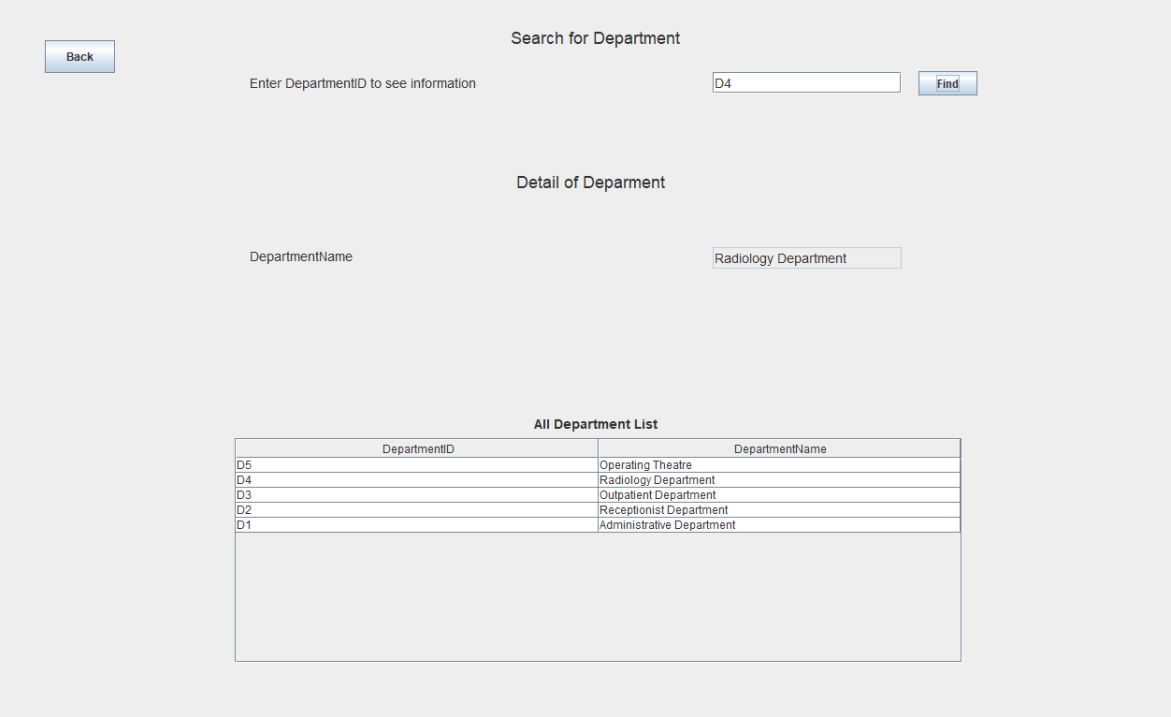
5.11. Document Name: Search Department Form

5.11.1. Description: This form allows the user to search the existing Department. Also user can see the short information of all department existing in Department table

5.11.2. Control on the Document:

- DepartmentID: Not empty

5.11.3. UI Design:



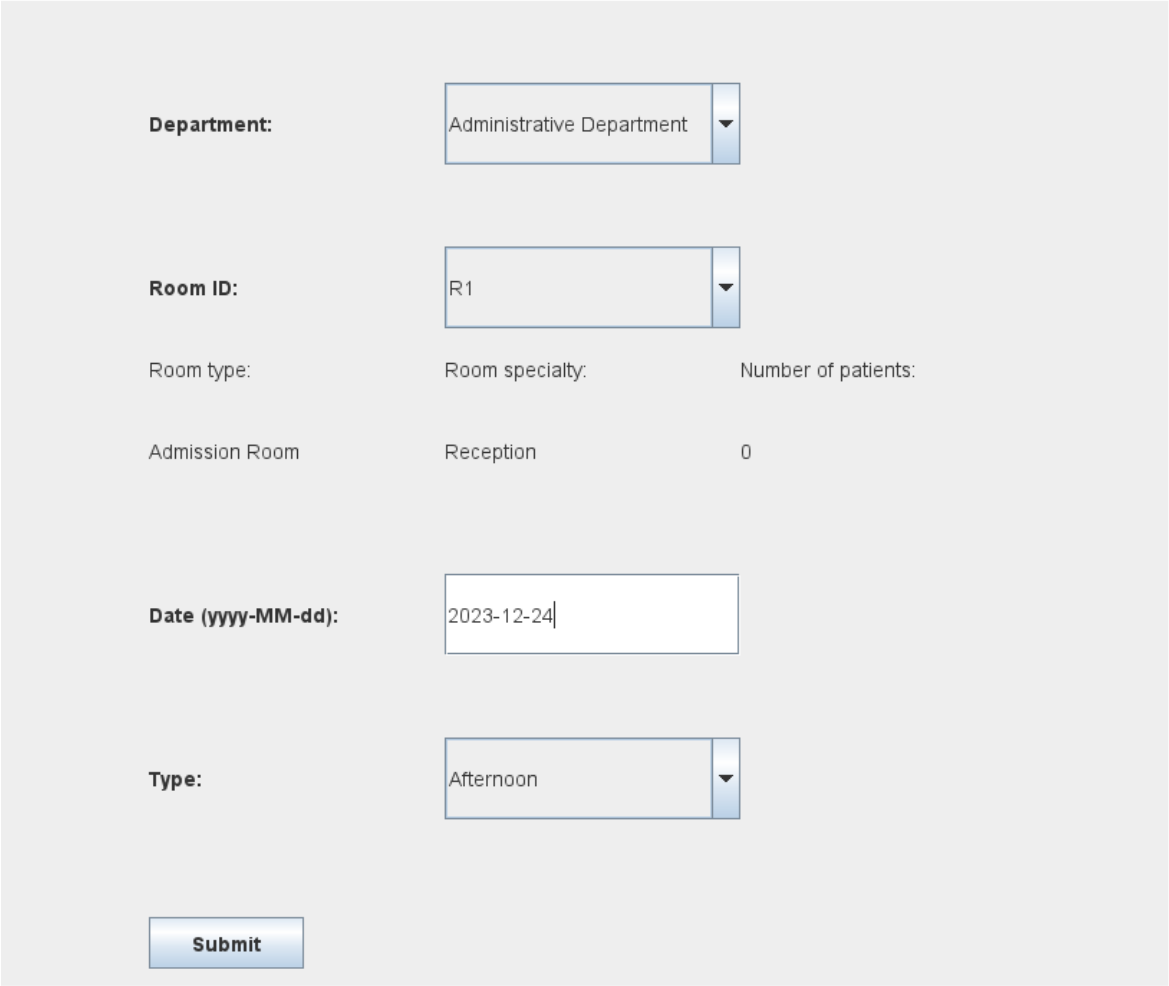
5.12. Document Name: Generate Shift Form

5.12.1. Description: This form allows user to generate a shift on a specific date

5.12.2. Control on the Document:

- Date: Not empty, valid date-format (must be yyyy-MM-dd)

5.12.3. UI Design:



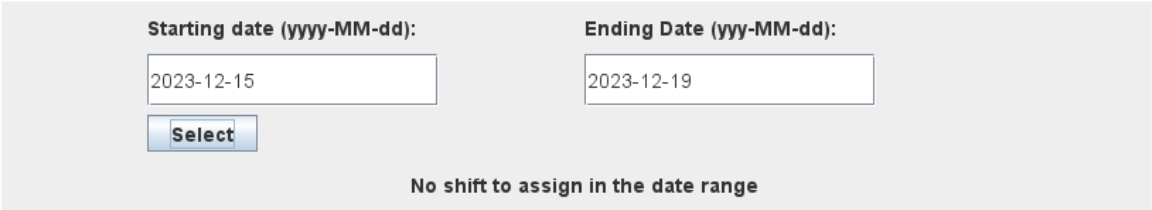
5.13. Document Name: Assign Shift Form

5.13.1. Description: This form allows user to assign a shift if there is any shift existed relevant with the start and end date

5.13.2. Control on the Document:

- Starting date: Not empty, valid date-format (must be yyyy-MM-dd)
- Ending date: Not empty, valid date-format (must be yyyy-MM-dd)

5.13.3. UI Design:



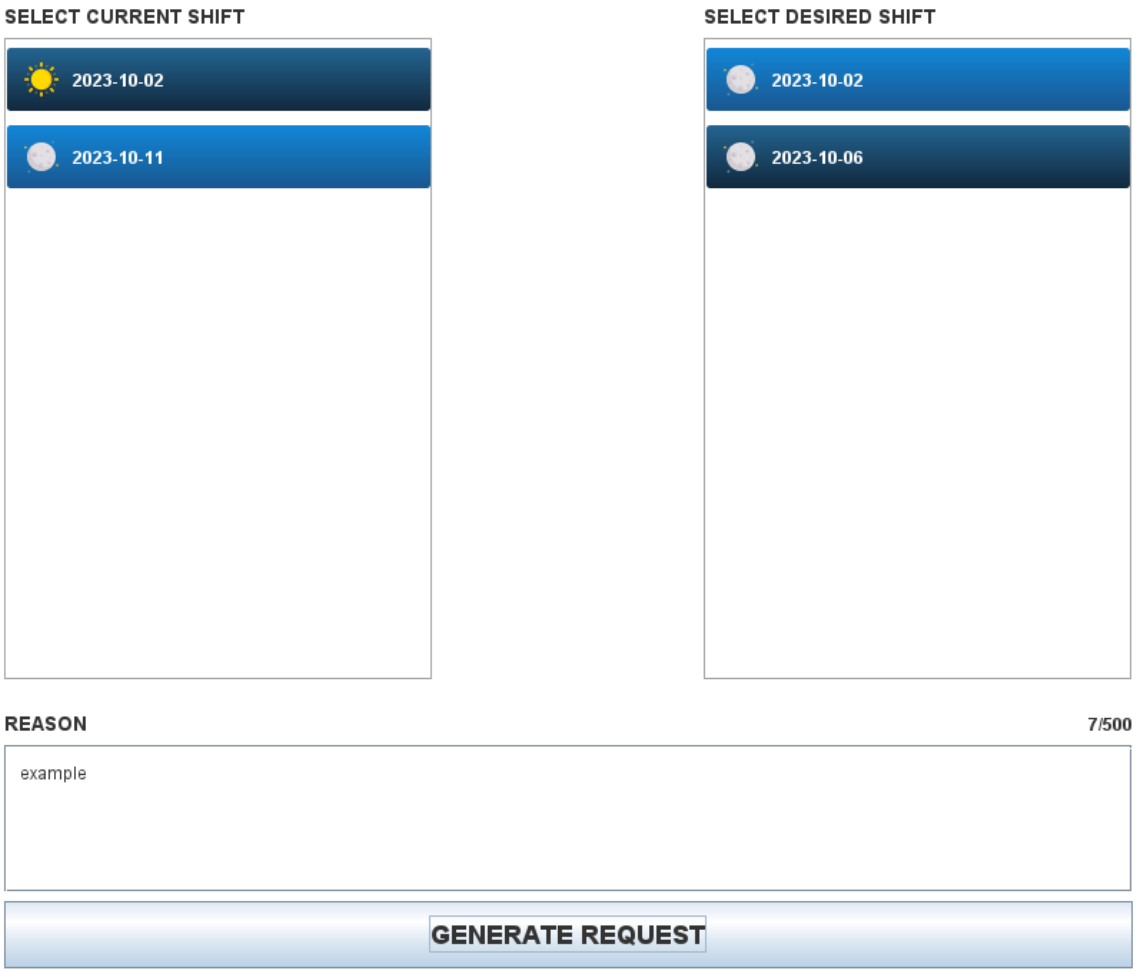
5.14. Document Name: Generate Request for Shift Form

5.14.1. Description: This form allows user to generate request on a shift with the reason of that shift request

5.14.2. Control on the Document:

- Select Current Shift: Must select a timeline
- Select Desired Shift: Must select a timeline
- Reason: Not empty

5.14.3. UI Design:



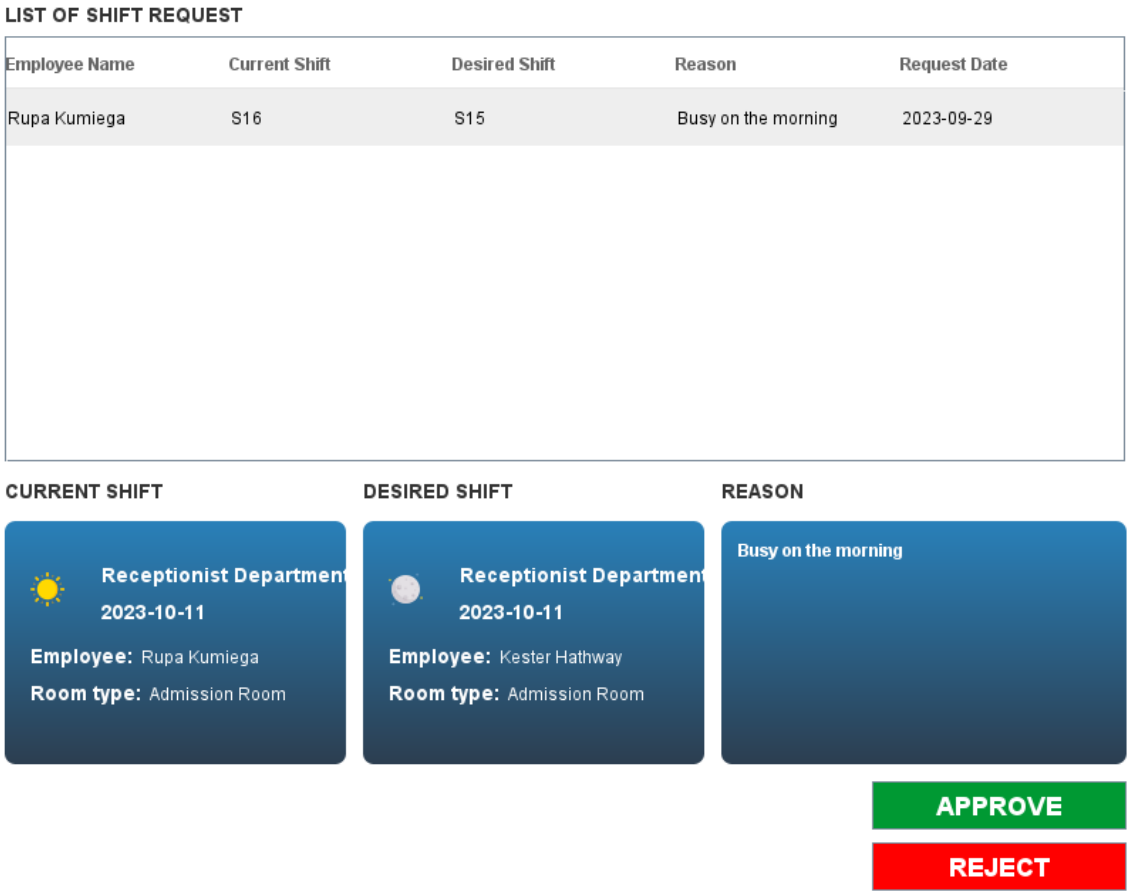
5.15. Document Name: Approving or Disapproving Shift Request Form

5.15.1. Description: This form allows users to select a shift and decide to approve it or decline it

5.15.2. Control on The Document:

- List of Shift Request: Must select a request

5.15.3. UI Design:

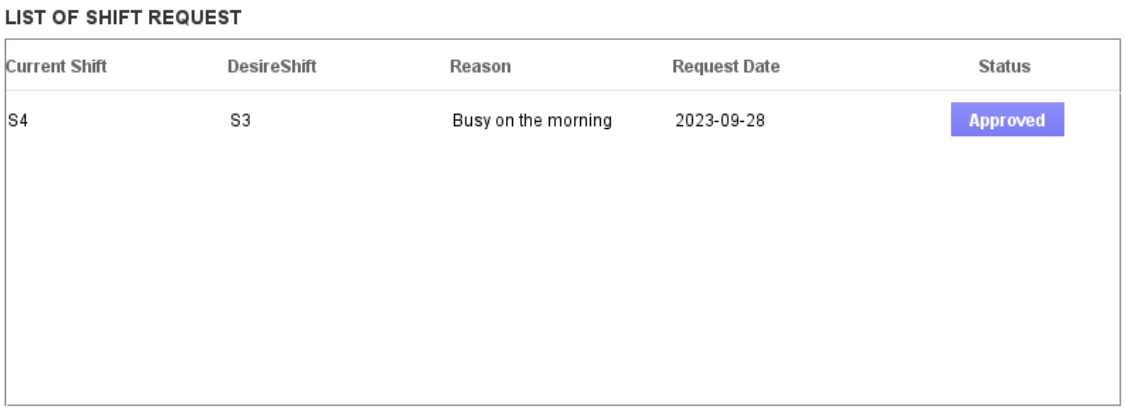


5.16. Document Name: Check status of Shift Request Form

5.16.1. Description: This form allows users to view the status of all shift requests which have been generated so far

5.16.2. Control on the Document:

5.16.3. UI Design:



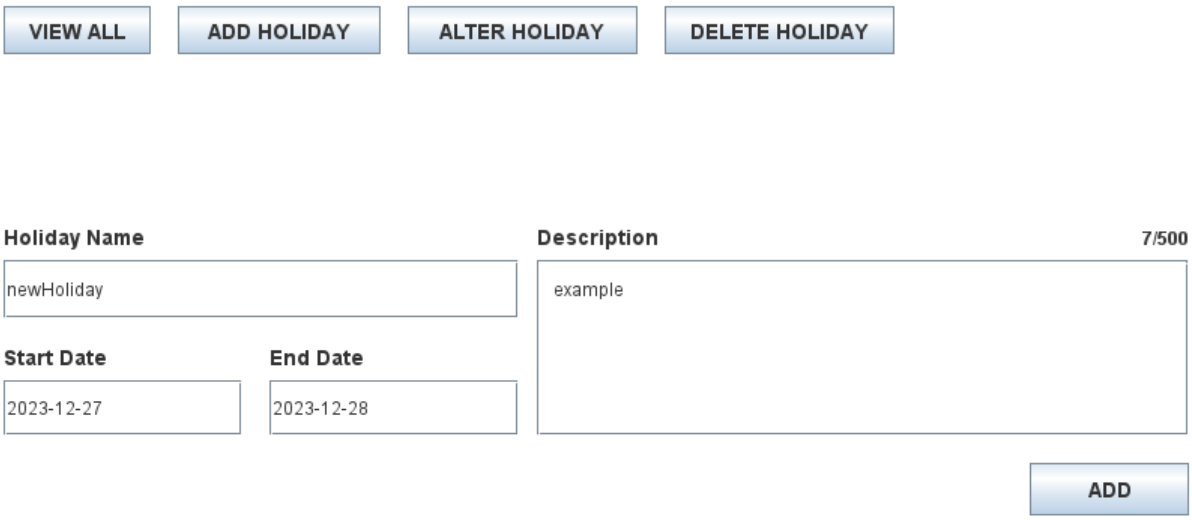
5.17. Document Name: Add Holiday Form

5.17.1. Description: This form allows users to add a Holiday with specifical information

5.17.2. Control on the Document:

- Holiday Name: not empty
- Star Date: Not empty, valid date-format (must be yyyy-MM-dd)
- End Date: Not empty, valid date-format (must be yyyy-MM-dd)
- Description: Not empty

5.17.3. UI Design:



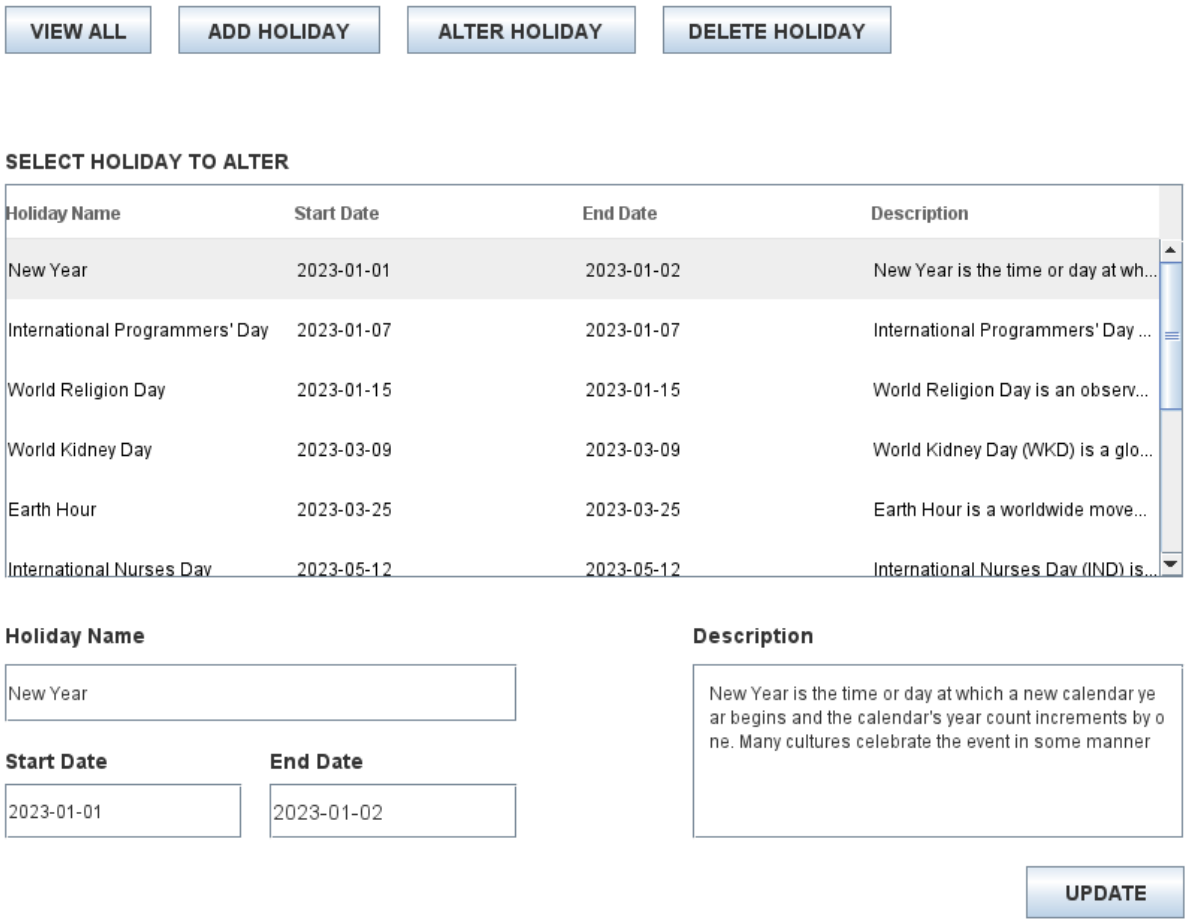
5.18. Document Name: Alter Holiday Form

5.18.1. Description: This form allows user to alter Holidays which are currently existed in a year with specifical information

5.18.2. Control on the Document:

- Select Holiday to Alter: Must select a Holidayline
- Holiday Name: not empty
- Star Date: not empty, valid date-format (must be yyyy-MM-dd)
- End Date: not empty, valid date-format (must be yyyy-MM-dd)
- Description: not empty

5.18.3. UI Design:



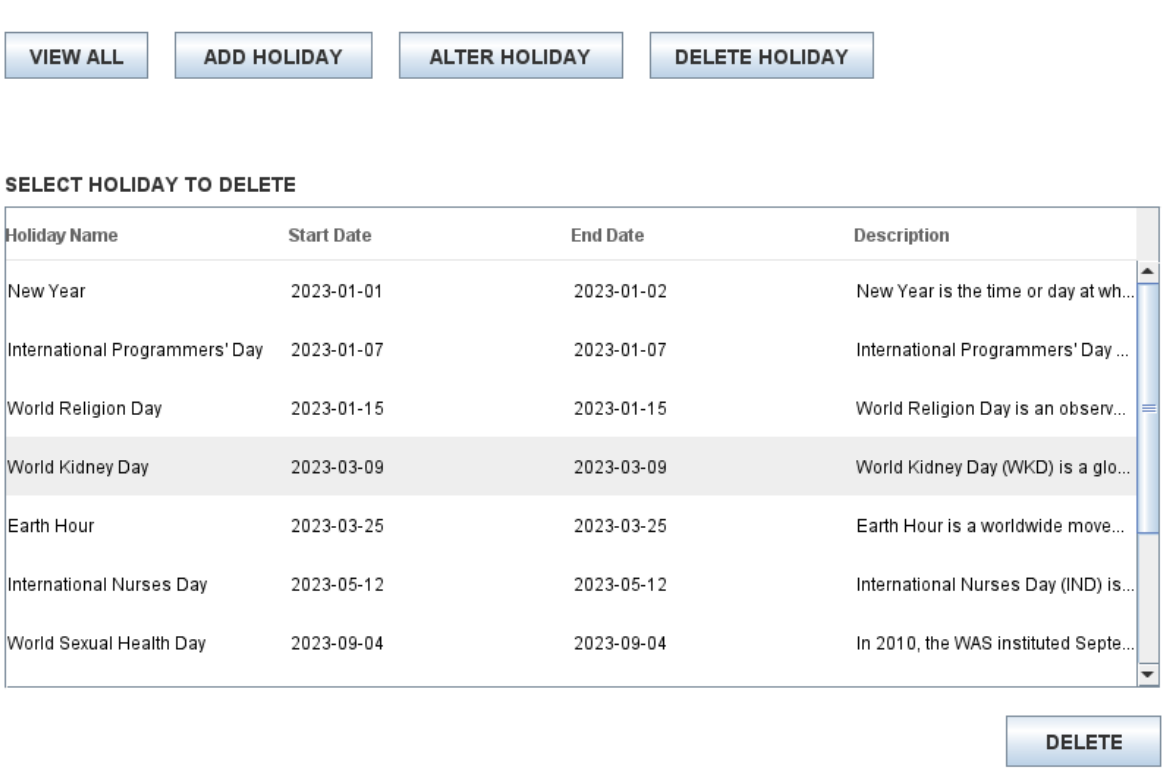
5.19. Document Name: Delete Holiday Form

5.19.1. Description: This form allows user to delete Holidays which currently with specifical information

5.19.2. Control on the Document:

- Select Holiday to Delete: Must select a Holidayline

5.19.3. UI Design:

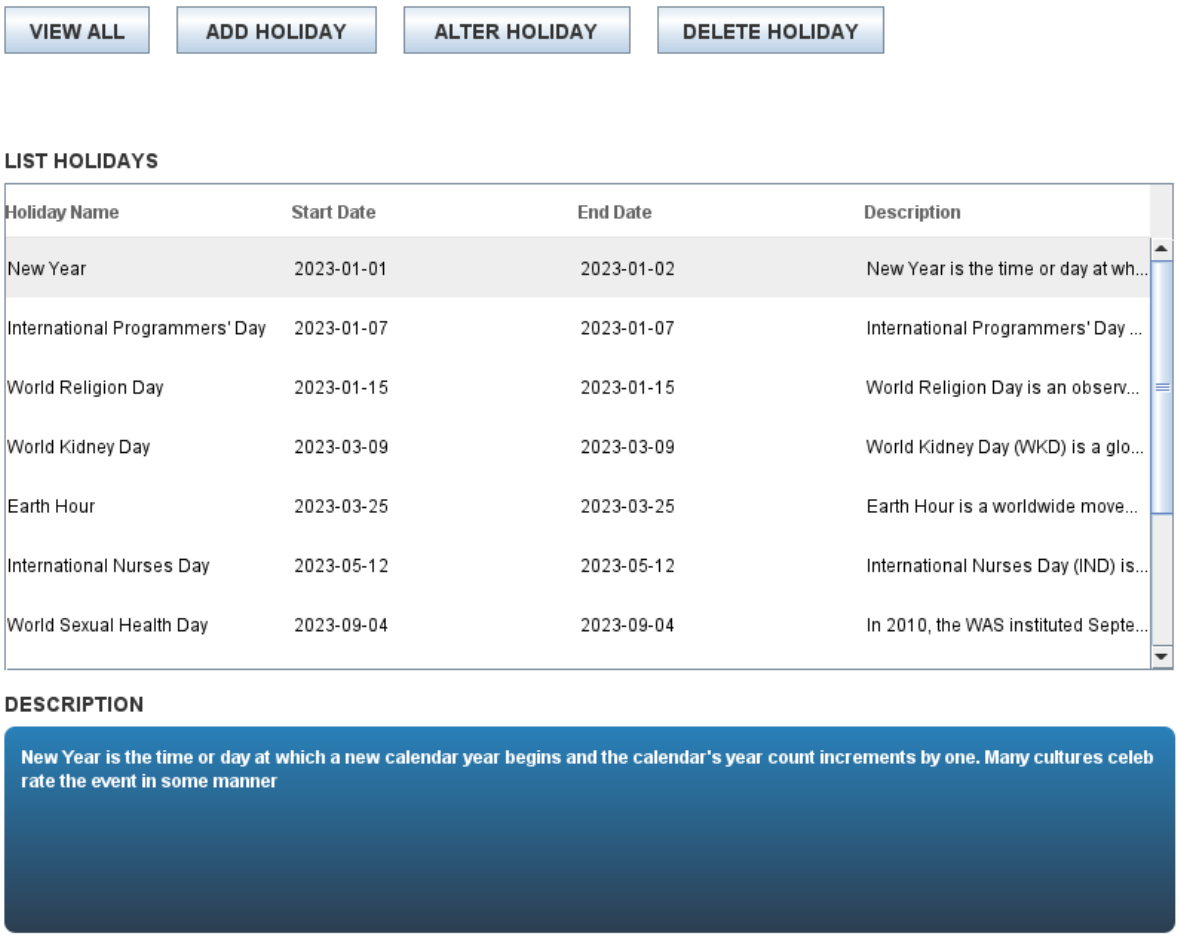


5.20. Document Name: View List Holiday Form

5.20.1. Description: This form allows user to view all Holiday with specifical information

5.20.2. Control on the Document:

5.20.3. UI Design:



5.21. Document Name: Add Patient Form

5.21.1. Description:
This form allows user to add new Patient with detail information

5.21.2. Control on the Document:

- PatientID: Not empty, not duplicated
- Patient Name: Not empty
- Gender: Not empty, must be Male or Female
- Phone: Not empty
- Address: Not empty
- Date of Birth: Not empty, valid date-format (must be yyyy-MM-dd)
- Note: Can be empty

5.21.3. UI Design:

5.22. Document Name: Update Patient Form

5.22.1. Description: This form allows user to alter existed Patient with detail information

5.22.2. Control on the Document:

- PatientID: Not empty
- Patient Name: Not empty
- Gender: Not empty, must be Male or Female
- Phone: Not empty
- Address: Not empty
- Date of Birth: Not empty, valid date-format (must be yyyy-MM-dd)
- Note: Can be empty

5.22.3. UI Design:

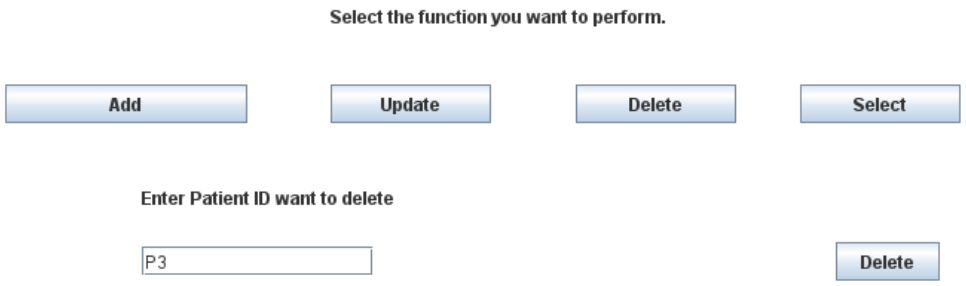
5.23. Document Name: Delete Patient Form

5.23.1. Description: This form allows user to delete existed Patient with detail information

5.23.2. Control on the Document:

- PatientID: Not empty

5.23.3. UI Design:



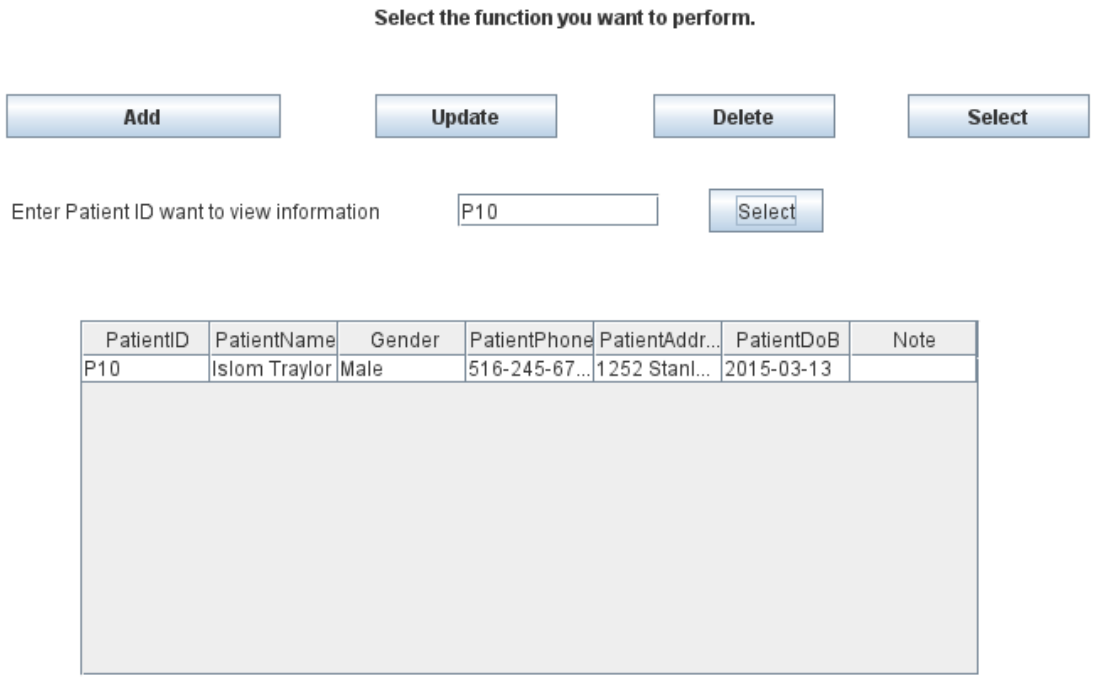
5.24. Document Name: Select Patient Form

5.24.1. Description: This form allows user to view a Patient with detail information or view all the currently existed Patient

5.24.2. Control on the Document:

- PatientID: Not empty

5.24.3. UI Design:



5.25. Document Name: Add Health Record Form

5.25.1. Description: This form allows user to add new Record with detail information

5.25.2. Control on the Document:

- PatientID: Not empty, not duplicated
- Old Record: Not empty
- Name: Not empty
- Day of Birth: Not empty, valid date-format (must be yyyy-MM-dd)
- RecordID: Not empty
- Enter Health Record: Not empty

5.25.3. UI Design:

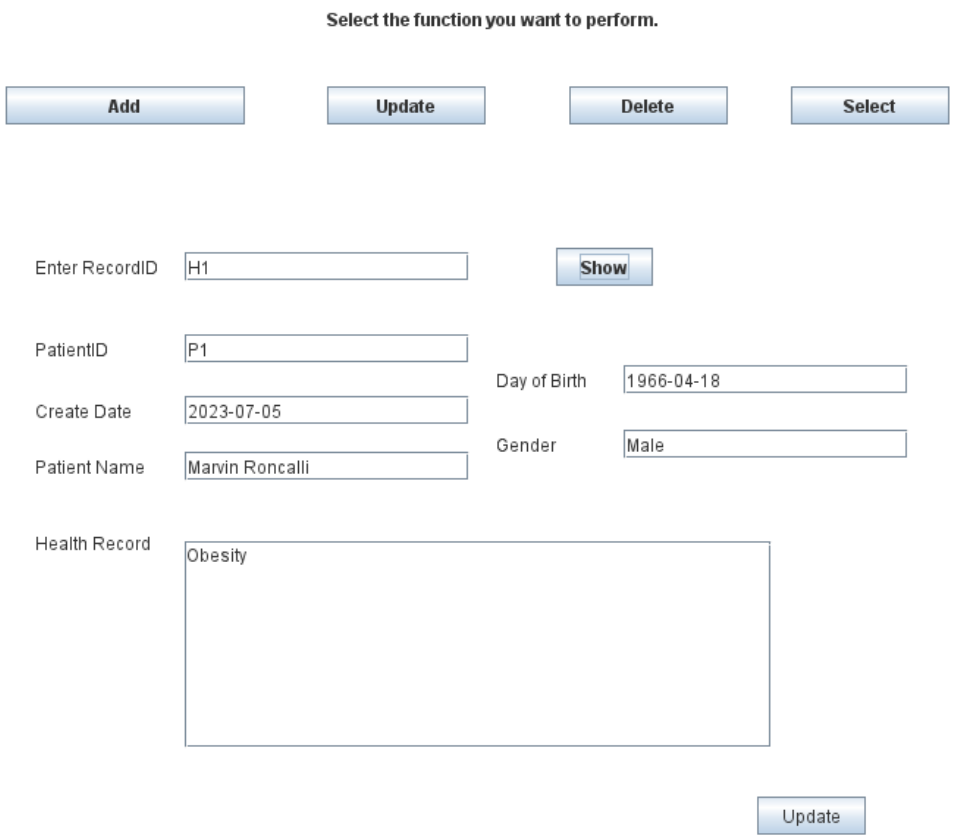
5.26. Document Name: Update Health Record Form

5.26.1. Description: This form allows user to alter existed Record with detail information

5.26.2. Control on the Document:

- RecordID: Not empty
- PatientID: Not empty
- Create Date: Not empty, valid date-format
- Patient Name: Not empty
- Day of Birth: Not empty, valid date-format
- Gender: Not empty
- Health Record: Not empty

5.26.3. UI Design:



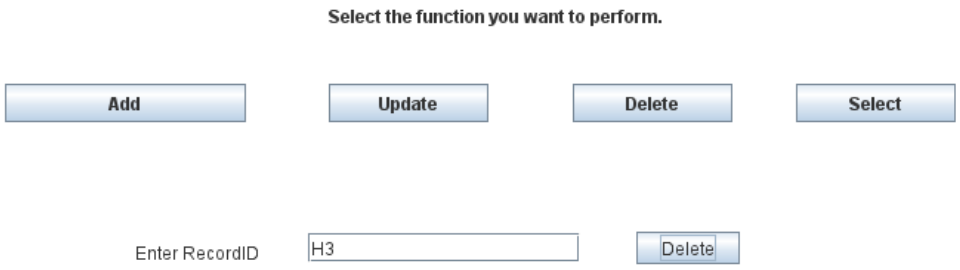
5.27. Document Name: Delete Health Record Form

5.27.1. Description: This form allows user to delete existed Record with detail information

5.27.2. Control on the Document:

- RecordID: Not empty

5.27.3. UI Design:



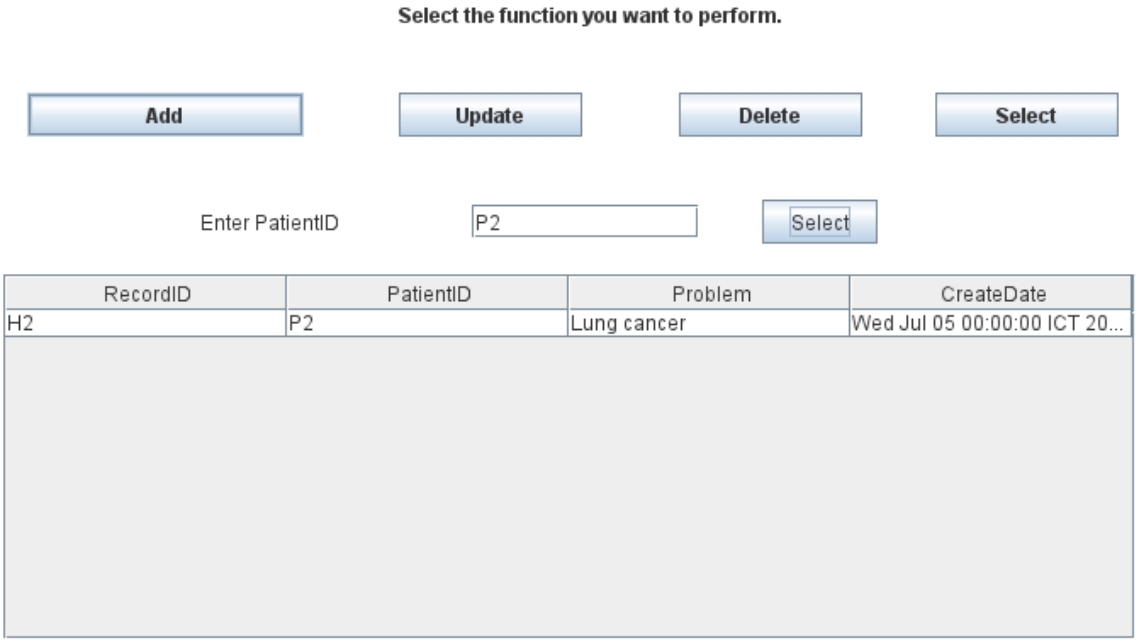
5.28. Document Name: Select Health Record Form

5.28.1. Description: This form allows user to view a record of an existed patient with detail information or view the detail information of all patient currently existed

5.28.2. Control on the Document:

- PatientID: Not empty

5.28.3. UI Design:



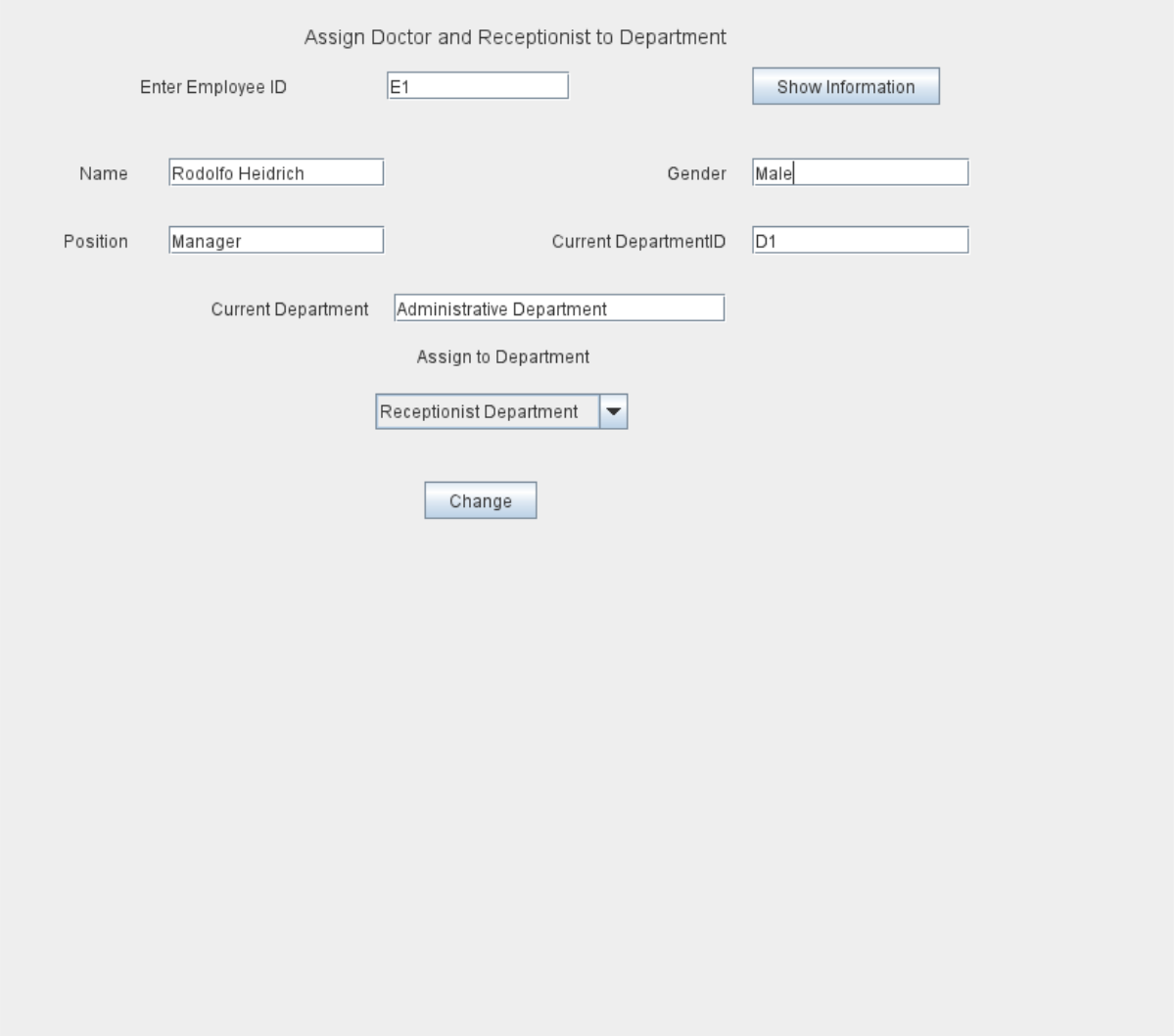
5.29. Document Name: Assign Doctor, Receptionist to Department Form

5.29.1. Description:
This form allows user to assign existed Doctors or Receptionists with new Departments information and change it from the old one

5.29.2. Control on the Document:

- EmployeeID: Not empty

5.29.3. UI Design:



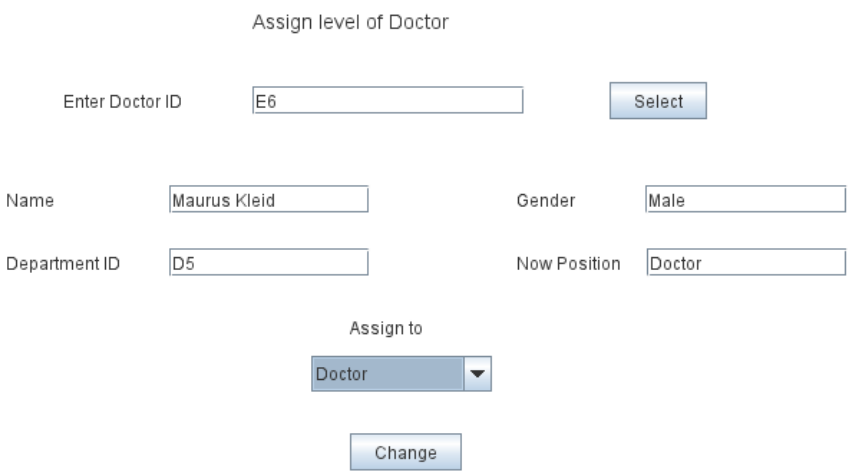
5.30. Document Name: Set Doctor Level Form

5.30.1. Description: This form allows user to set level of existed Doctors with new Position Level

5.30.2. Control on the Document:

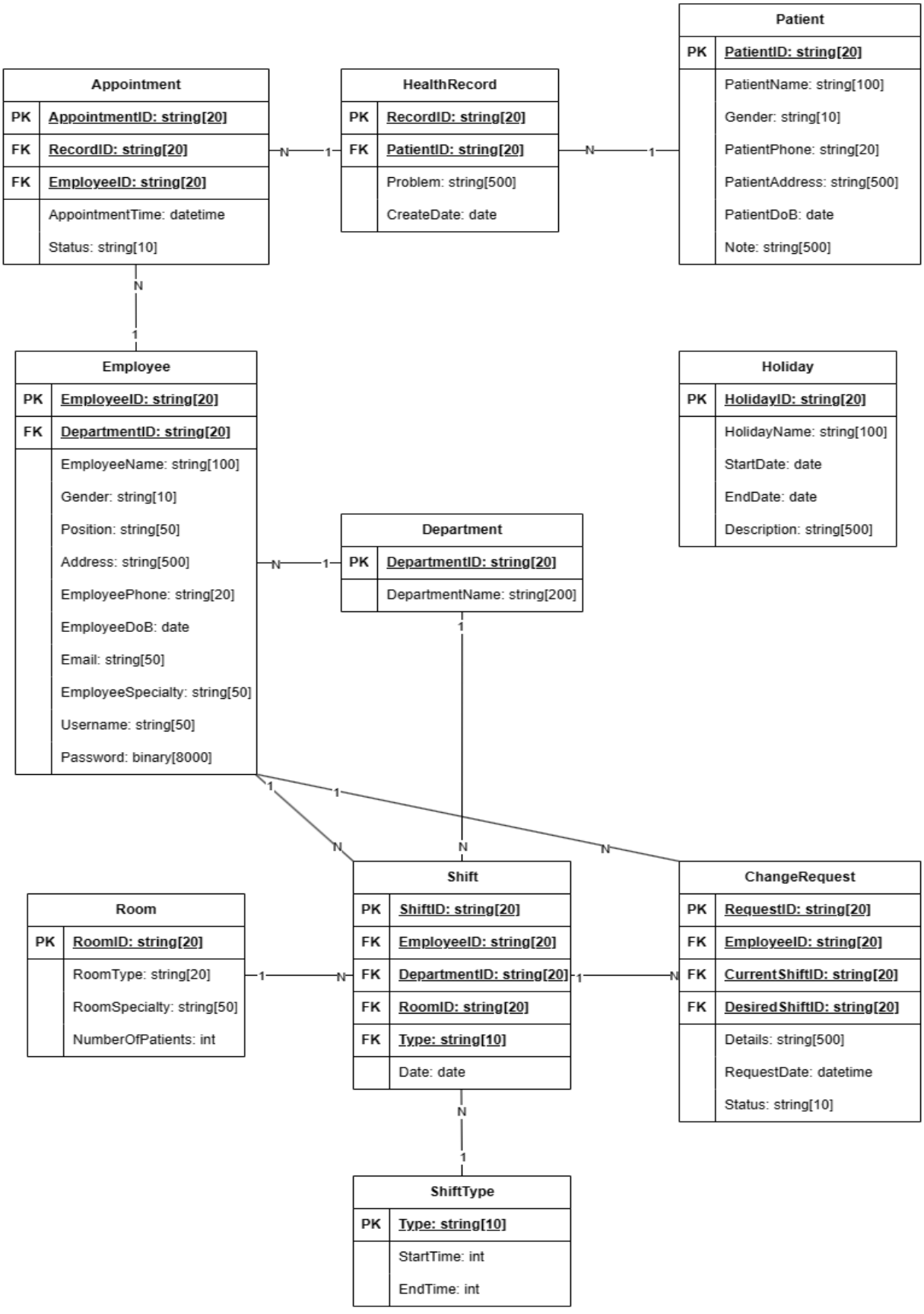
- DoctorID: Not empty

5.30.3. UI Design:


Assign level of Doctor
Enter Doctor ID
E6
Select
Name
Maurus Kleid
Gender
Male
Department ID
D5
Now Position
Doctor
Assign to
Doctor
Change

DATABASE DESIGN DOCUMENT

I. ER/Class Diagram



II. Table Description

Employee			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
EmployeeID	string[20]	Primary key	Employee's ID
DepartmentID	string[20]	Foreign key and not null	Department's ID
EmployeeName	string[100]	Not null	Employee's name
Gender	string[10]	Female or Male and not null	Employee's gender
Position	string[50]	Not null	Employee's position
Address	string[500]	Not null	Employee's address
EmployeePhone	string[20]	Not null	Employee's phone number
EmployeeDoB	date	Not null	Employee's date of birth
Email	string[50]	Not null	Employee's email
EmployeeSpecialty	string[50]	Not null	Employee's specialty
Username	string[50]	Not null and unique	Employee's username
Password	binary[8000]	Not null	Employee's encrypted password

Department			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
DepartmentID	string[20]	Primary key	Department's ID
DepartmentName	string[200]	Not null and unique	Department's name

Patient			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
PatientID	string[20]	Primary key	Patient's ID
PatientName	string[100]	Not null	Patient's name
Gender	string[10]	Female or Male and not null	Patient's gender
PatientPhone	string[20]		Patient's ID
PatientAddress	string[500]		Patient's health problem
PatientDoB	date	Not null	Patient's date of birth
Note	string[500]		Notes about the patient

Room			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
RoomID	string[20]	Primary key	Room's ID
RoomType	string[20]	Not null	Room's type
RoomSpecialty	string[50]	Not null	Room's specialty
NumberOfPatients	Int	>= 0 and not null and has the default value 0	Number of patients stay in the room

HealthRecord			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
RecordID	string[20]	Primary key	Record's ID
PatientID	string[20]	Foreign key and not null	Patient's ID
Problem	string[500]	Not null	Patient's health problem

Holiday			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
HolidayID	string[20]	Primary key	Holiday's ID
HolidayName	String[100]	Not null	Holiday's name
StartDate	date	Not null	Holiday's starting time
EndDate	date	Not null and >= StartDate	Holiday's ending time
Description	string[500]		Holiday's descriptions and notes

Appointment			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
AppointmentID	string[20]	Primary key	Appoinment's ID
RecordID	string[20]	Foreign key and not null	Record's ID
EmployeeID	string[20]	Foreign key and not null	ID of employee who is responsible for the patient
AppointmentTime	datetime	Not null	Appointment time
Status	string[10]		Status of the appointment

ShiftType			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
Type	string[10]	Primary key	Shift's type
StartTime	int	Not null, >= 0 and <= 24	Shift's starting time
EndTime	int	Not null, >= 0, <= 24 and > StartTime	Shift's ending time

Shift			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
ShiftID	string[20]	Primary key	Shift's ID
EmployeeID	string[20]	Foreign key and the employee's specialty is as the same as the room's specialty	ID of the employee who is responsible for the shift
DepartmentID	string[20]	Foreign key and not null	ID of the department that is responsible for the shift
RoomID	string[20]	Foreign key, not null, the room's specialty is as the same as the employee's specialty	Room's ID of the shift
Type	string[10]	Foreign key, not null	Shift's type
Date	date	Not null	Shift's date
StartTime	int	Not null, >= 0 and <= 24	Shift's starting time
EndTime	int	Not null, >= 0, <= 24 and > StartTime	Shift's ending time

ChangeRequest			
Field Name	**Data Type**	**Constraint (Primary Key, Foreign Key, Identity, Not Null,...)**	**Description**
RequestID	string[20]	Primary key	Request's ID
EmployeeID	string[20]	Foreign key and not null	ID of the employee who made the change request
CurrentShiftID	string[20]	Foreign key, not null, the shift's employee ID differs from desired shift;s employeeID but have the same employee's specialty	Shift's ID to change
DesiredShiftID	string[20]	Foreign key and not null, the shift's employee ID differs from current shift;s employeeID but have the same employee's specialty	Desired shift's ID
Details	string[500]	Not null	Request's details
RequestDate	date	Not null	Request's generation date
Status	string[10]	Not null and has the default value Pending	Request's status

PROCESS DESIGN DOCUMENT

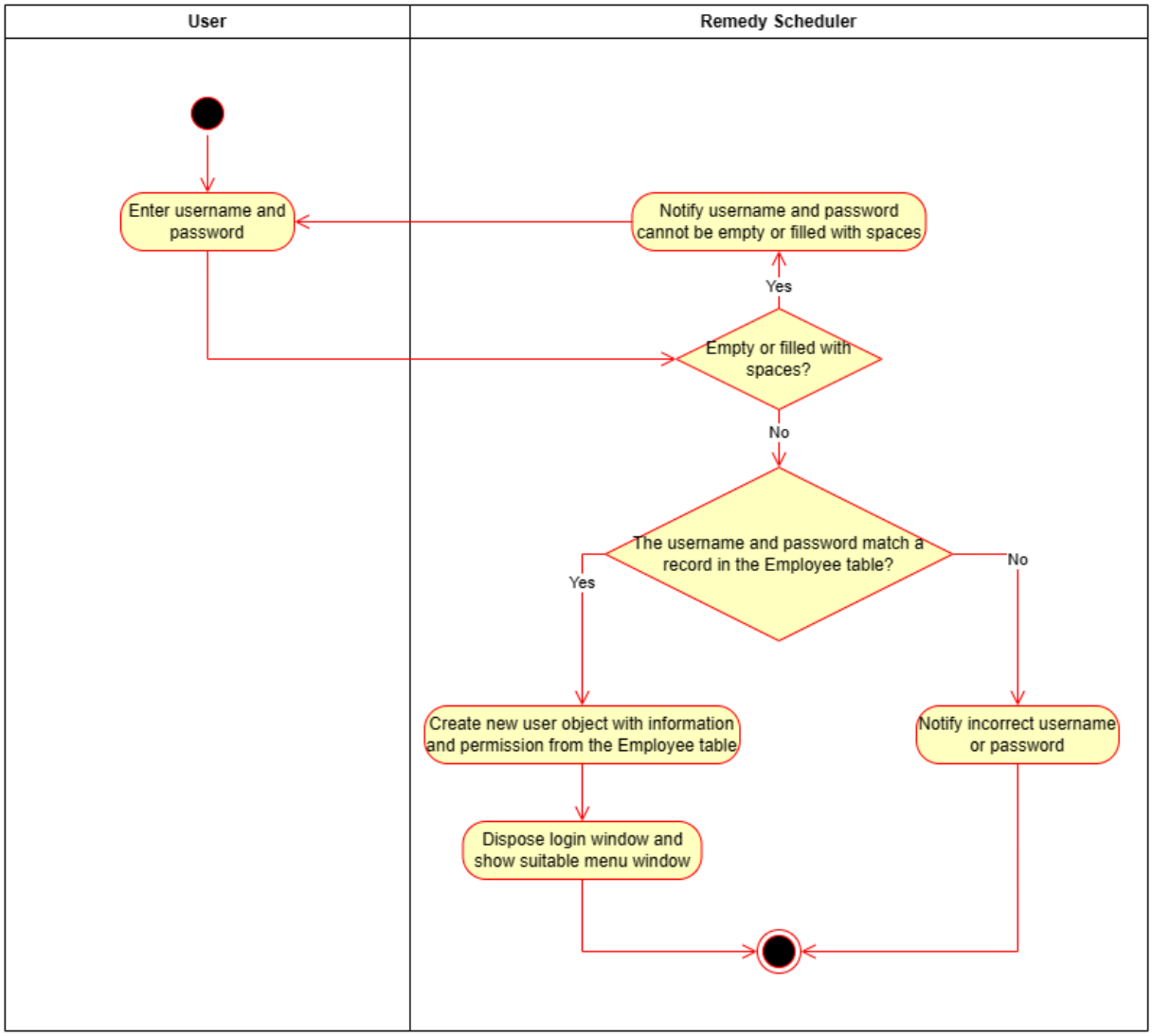
(Activity diagram, sequence diagram)

7.1. Process Name

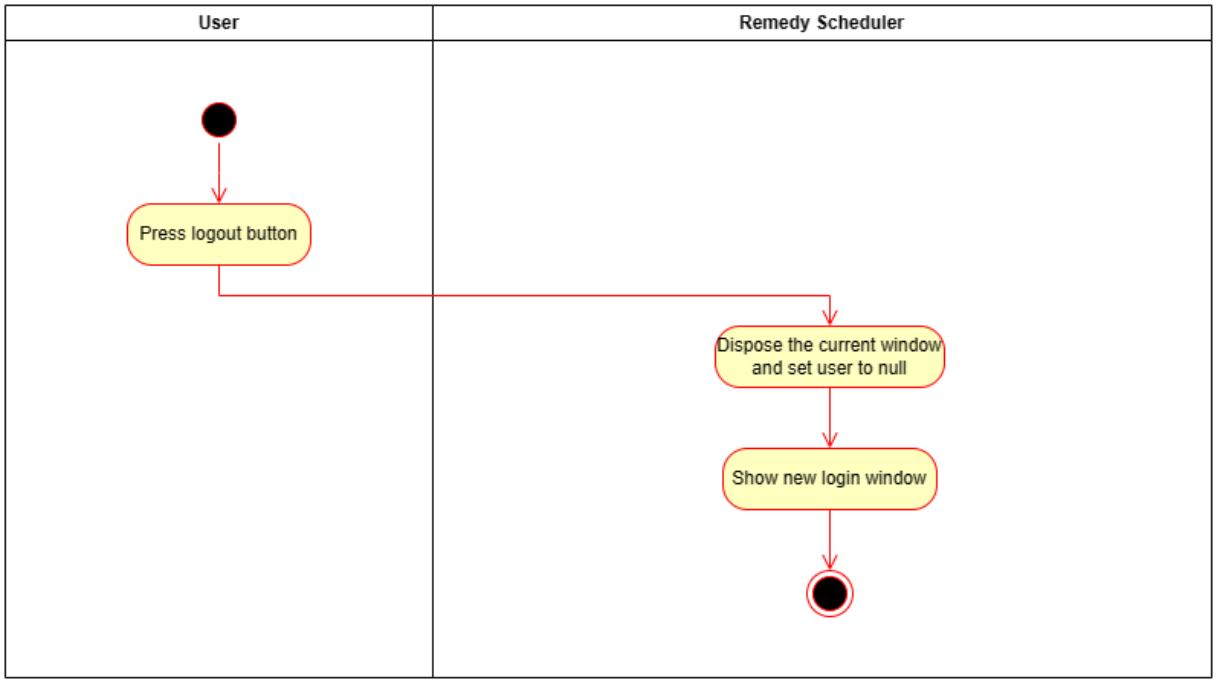
7.1.1. Log in Process
7.1.2. Log out Process
7.1.3. Generate shift Process
7.1.4. Assign shift Process
7.1.5. Generate shift request Process
7.1.6. Approving-Disapproving shift request Process
7.1.7. Check status of shift request Process
7.1.8. Add holiday Process
7.1.9. Alter holiday Process
7.1.10. Delete holiday Process
7.1.11. View list of holidays Process
7.1.12. Add patient Process
7.1.13. Update patient Process
7.1.14. Delete patient Process
7.1.15. Select patient Process
7.1.16. Add health record Process
7.1.17. Update health record Process
7.1.18. Delete health record Process
7.1.19. Select health record Process
7.1.20. Assign level to doctors Process
7.1.21. Assign doctors and receptionists to department Process
7.1.22. Add doctor and receptionist Process
7.1.23. Alter doctor and receptionist Process
7.1.24. Delete doctor and receptionist Process
7.1.25. Search doctor and receptionist Process
7.1.26. Add department Process
7.1.27. Alter department Process
7.1.28. Delete department Process
7.1.29. Search department Process
7.1.30. View time schedule Process
7.1.31. View appointment list Process

7.2. Process Description and Details

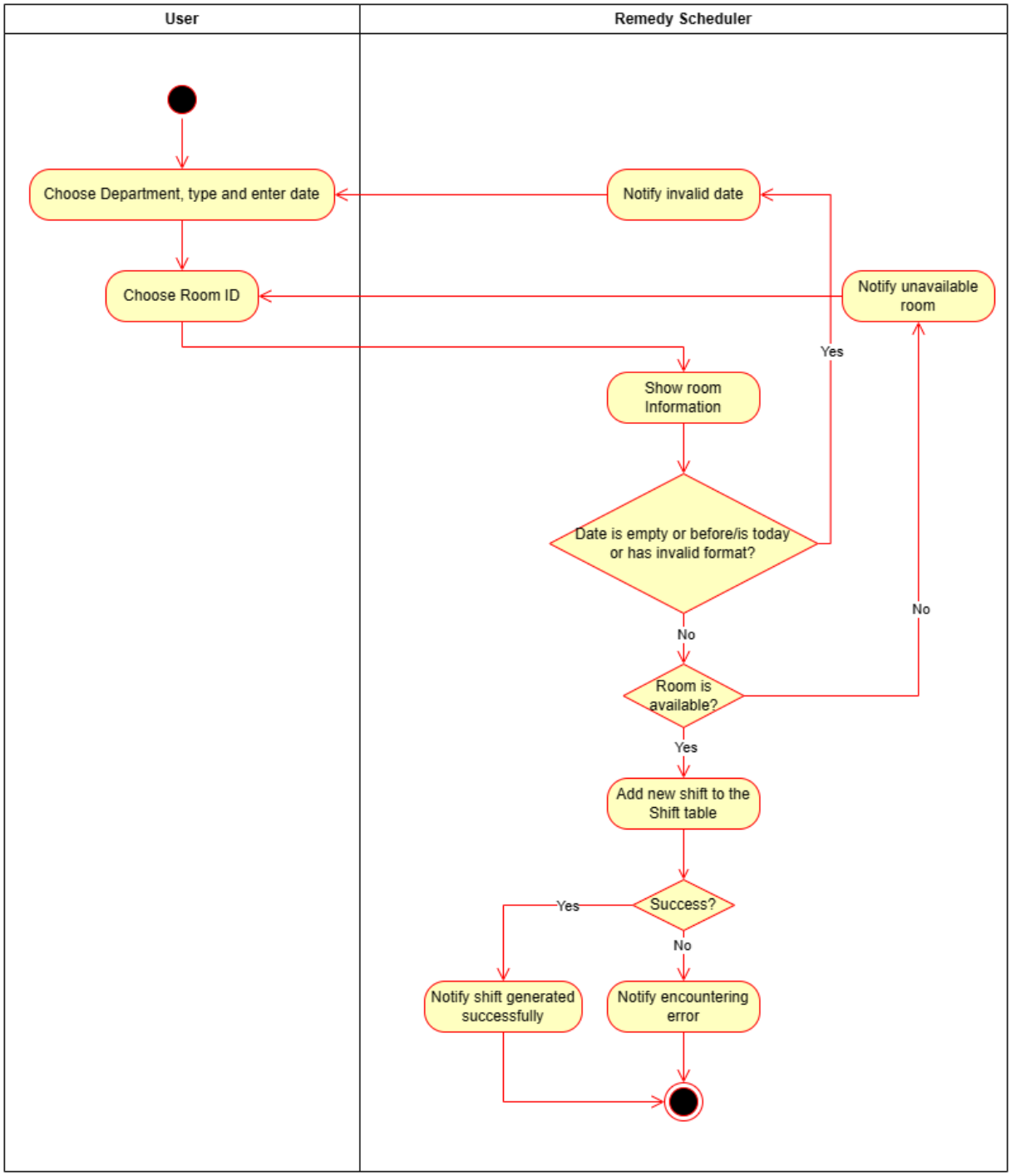
7.2.1. Log in


User
Remedy Scheduler
Enter username and password
Notify username and password cannot be empty or filled with spaces
Yes
Empty or filled with spaces?
No
The username and password match a record in the Employee table?
Yes
No
Create new user object with information and permission from the Employee table
Notify incorrect username or password
Dispose login window and show suitable menu window

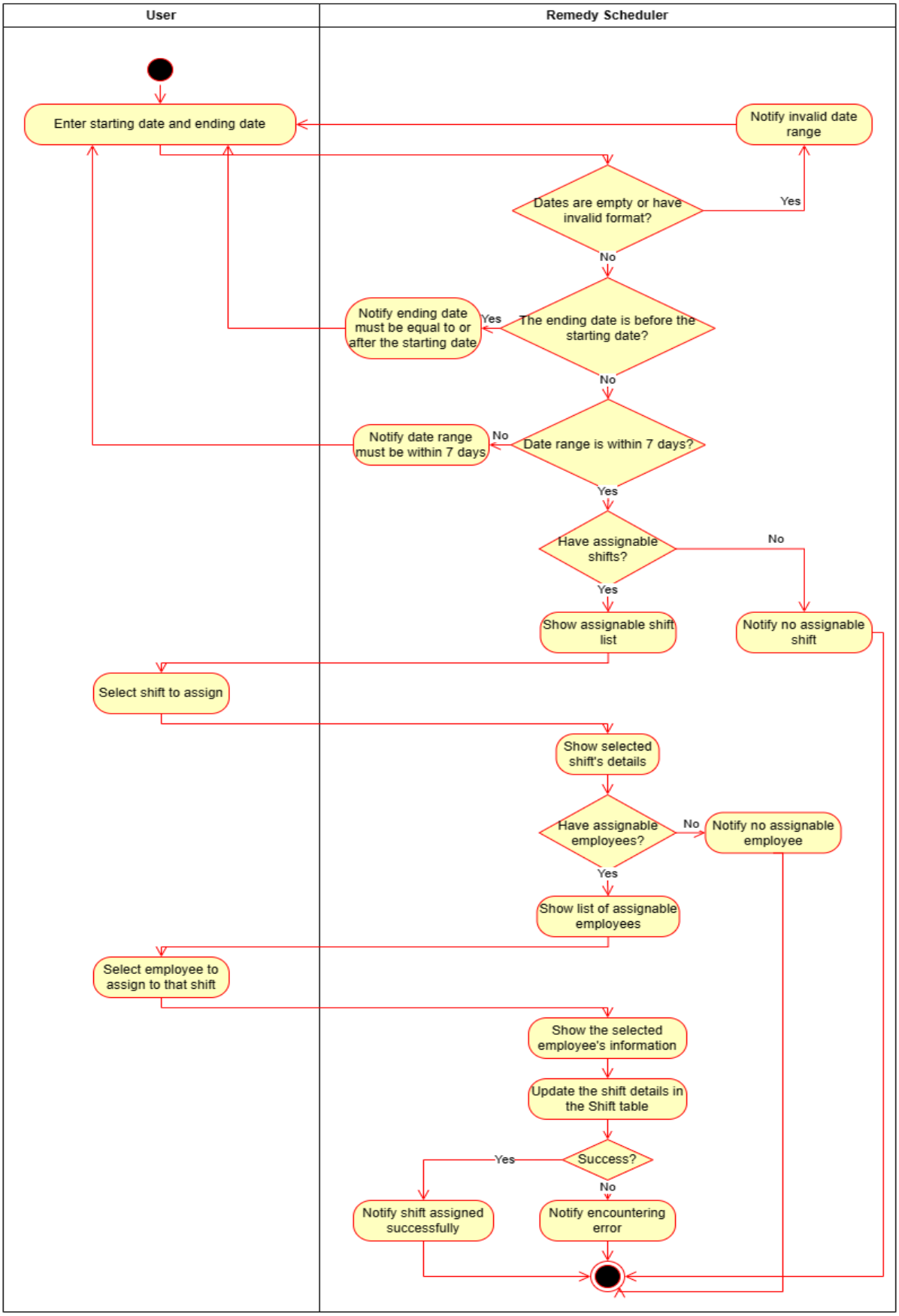
7.2.2. Log out


User
Remedy Scheduler
Press logout button
Dispose the current window and set user to null
Show new login window

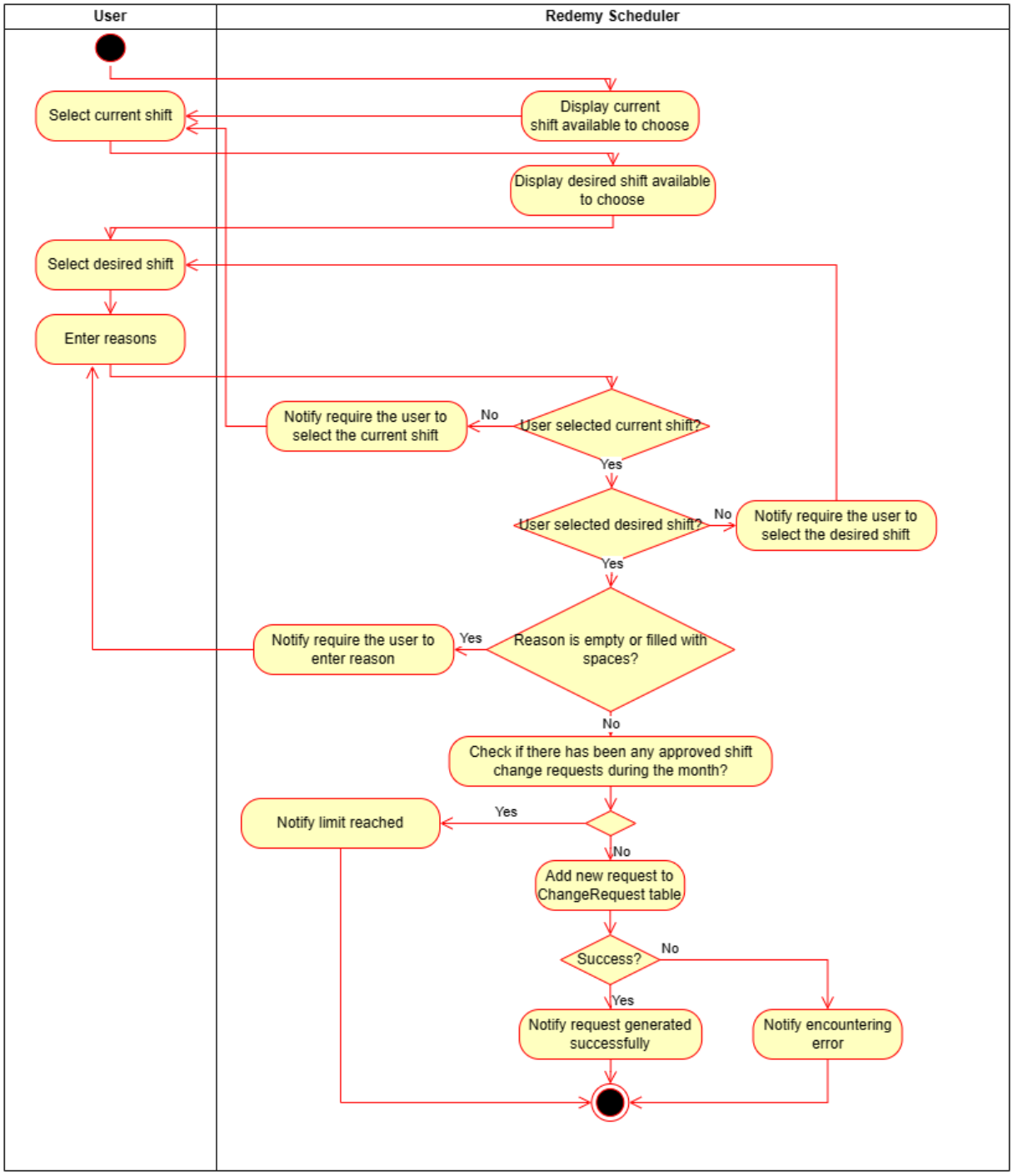
7.2.3. Generate shift



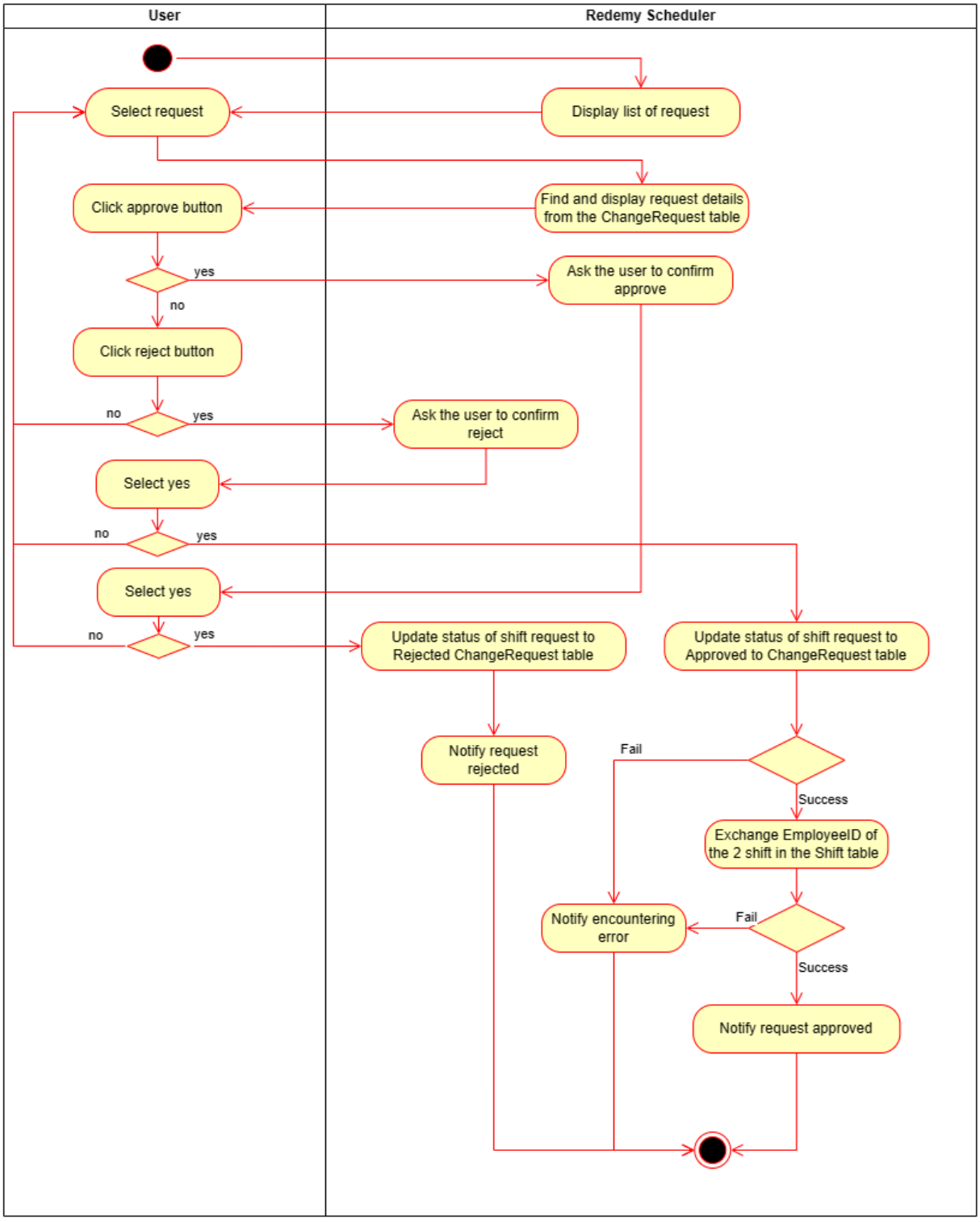
7.2.4. Assign shift



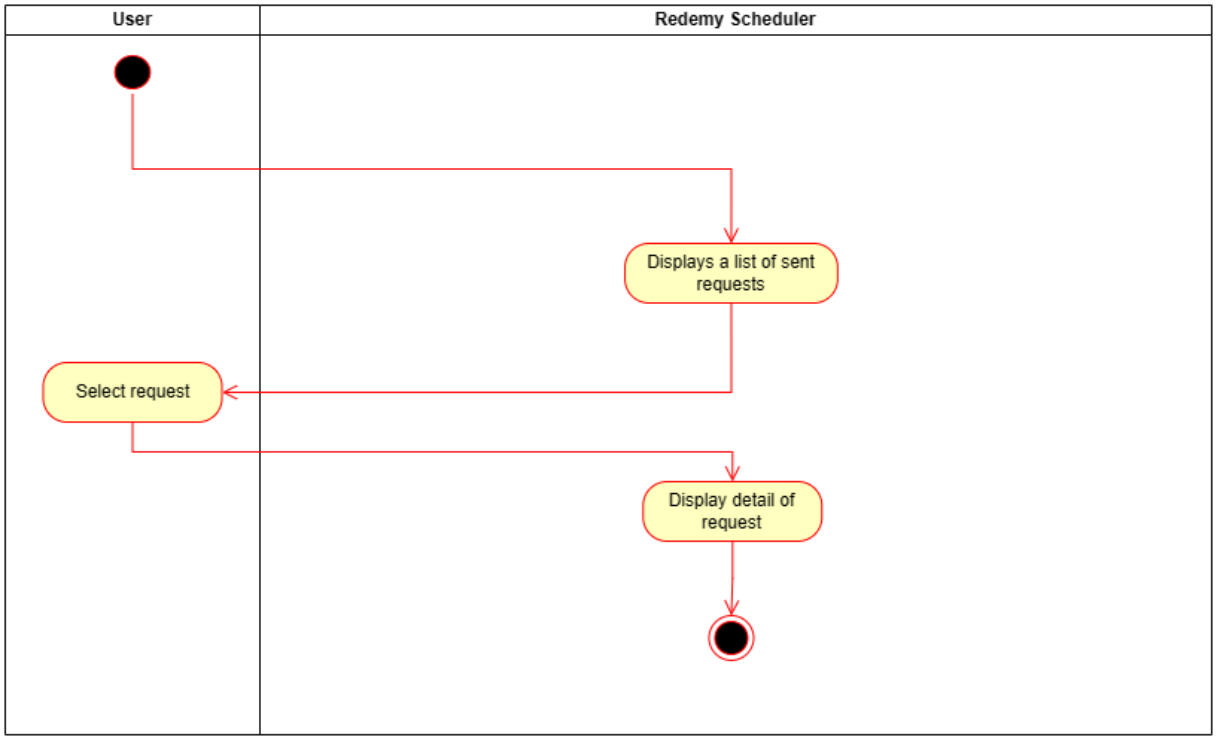
7.2.5. Generate shift request



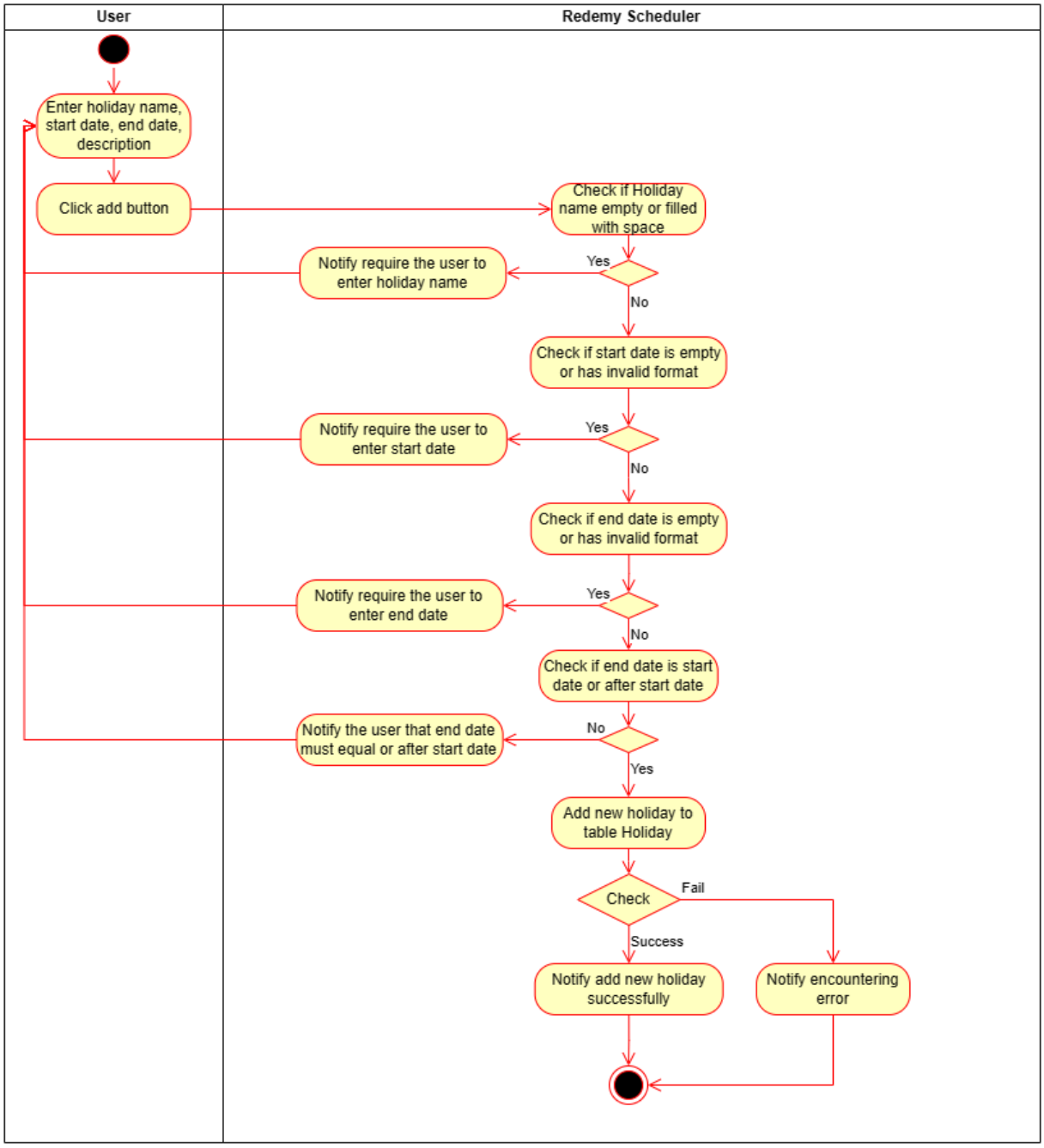
7.2.6. Approving-Disapproving shift request



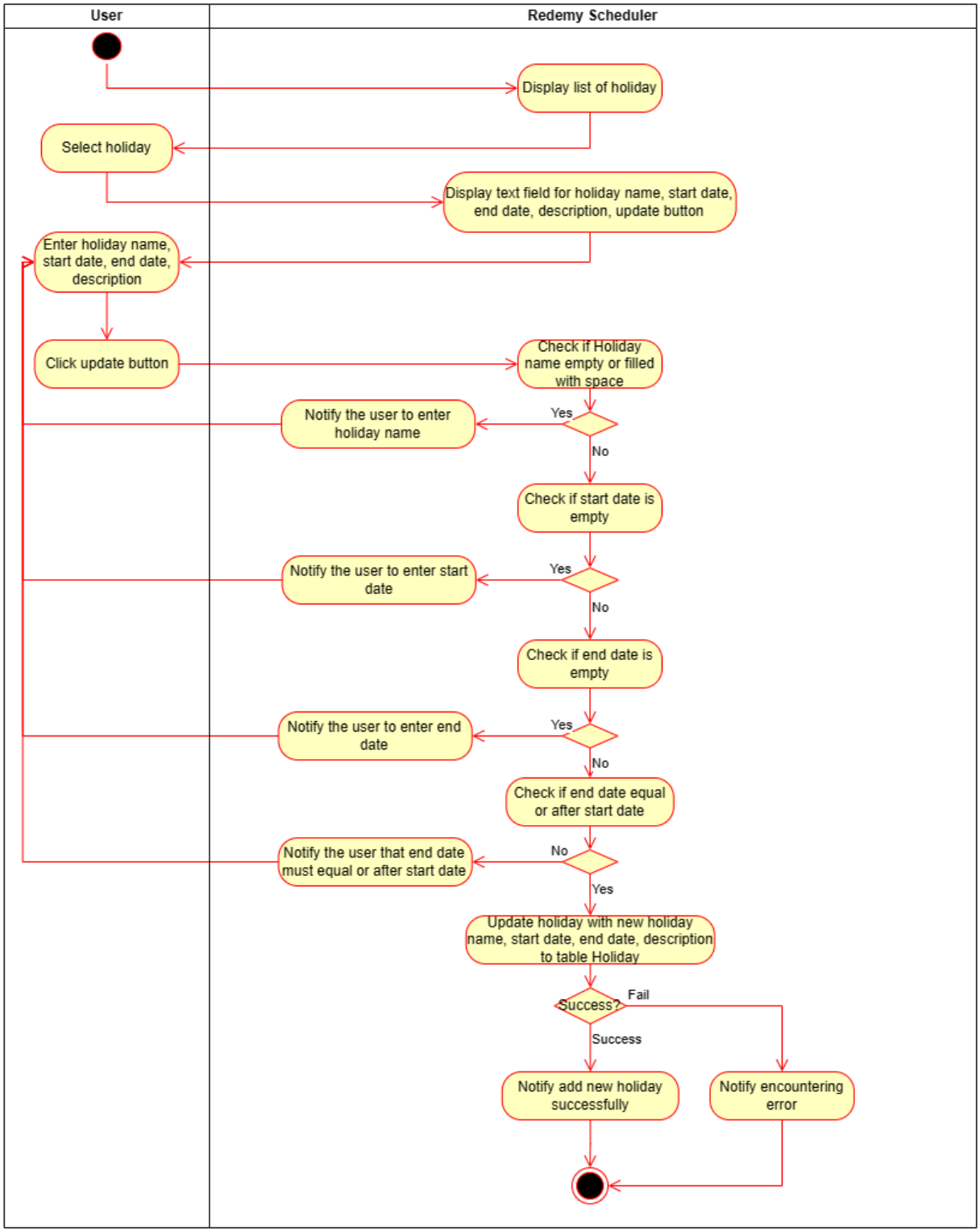
7.2.7. Check status of shift request



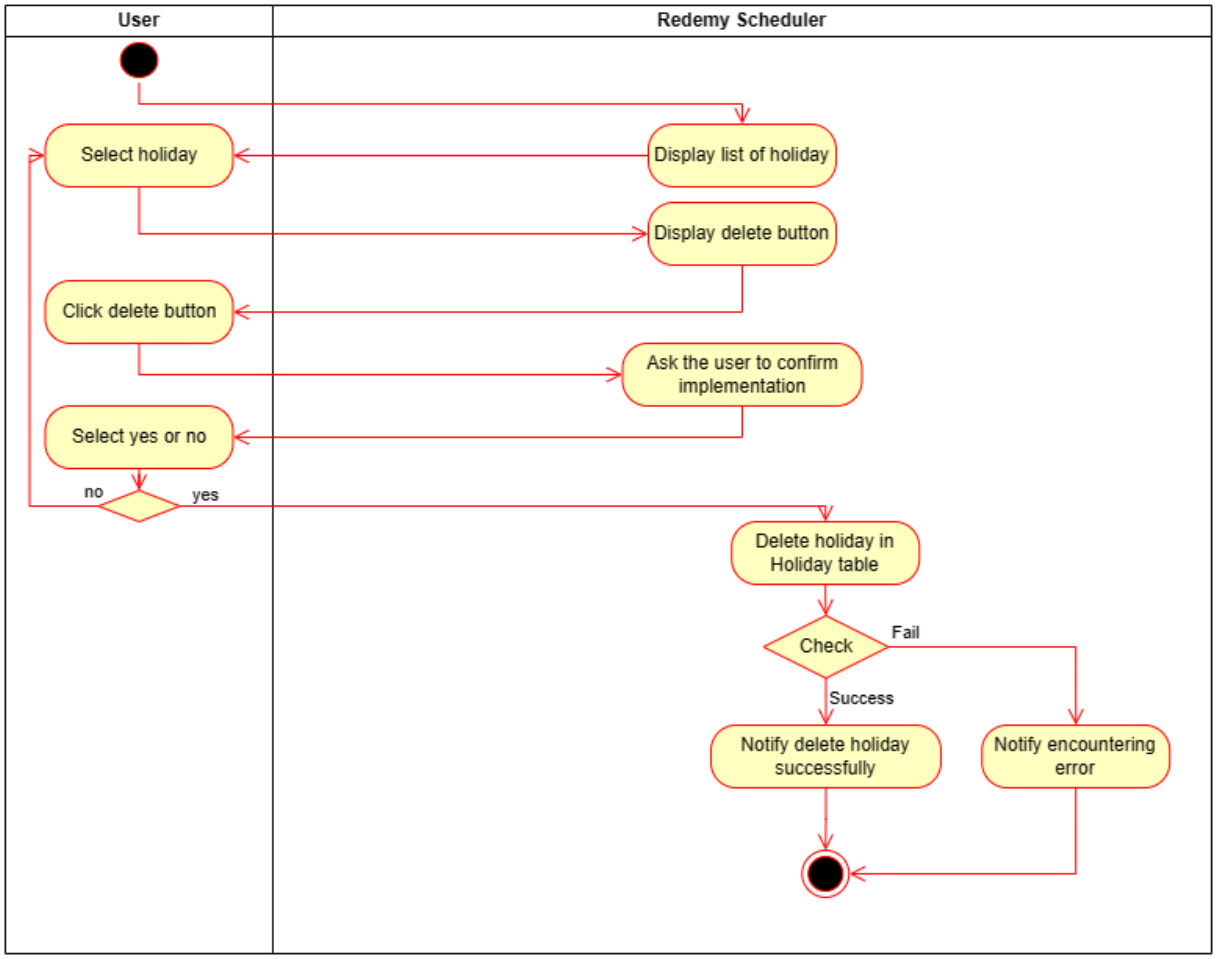
7.2.8. Add holiday



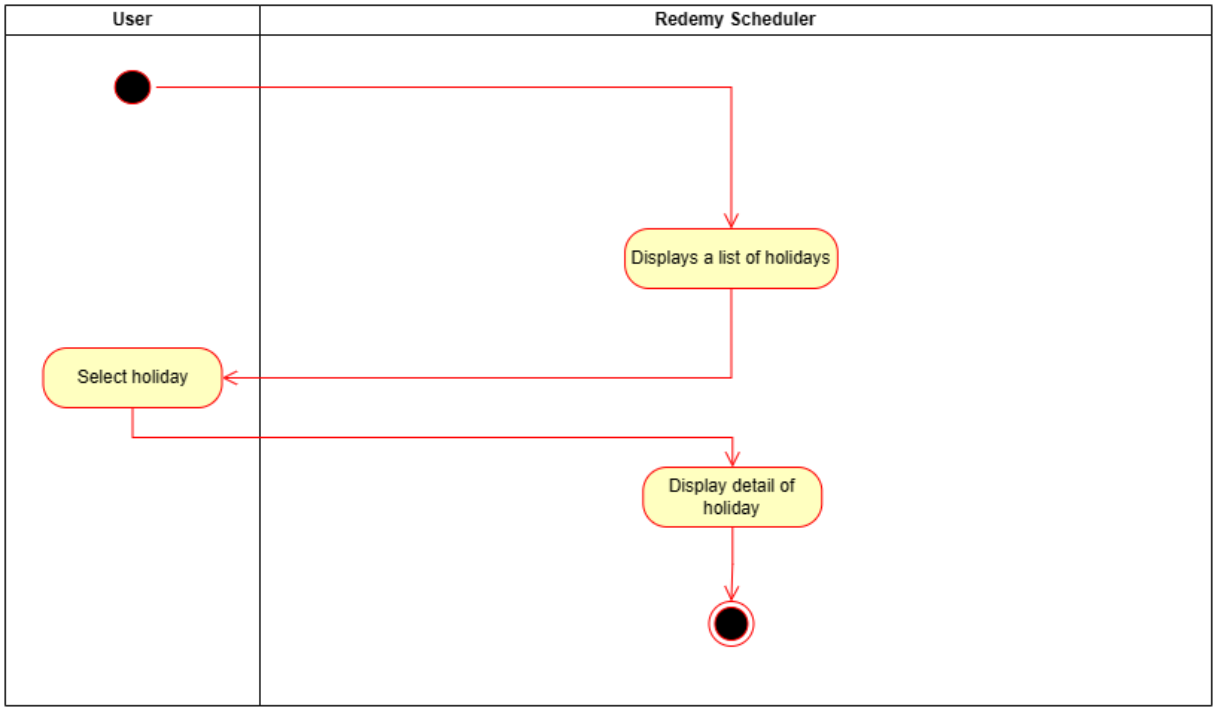
7.2.9. Alter holiday



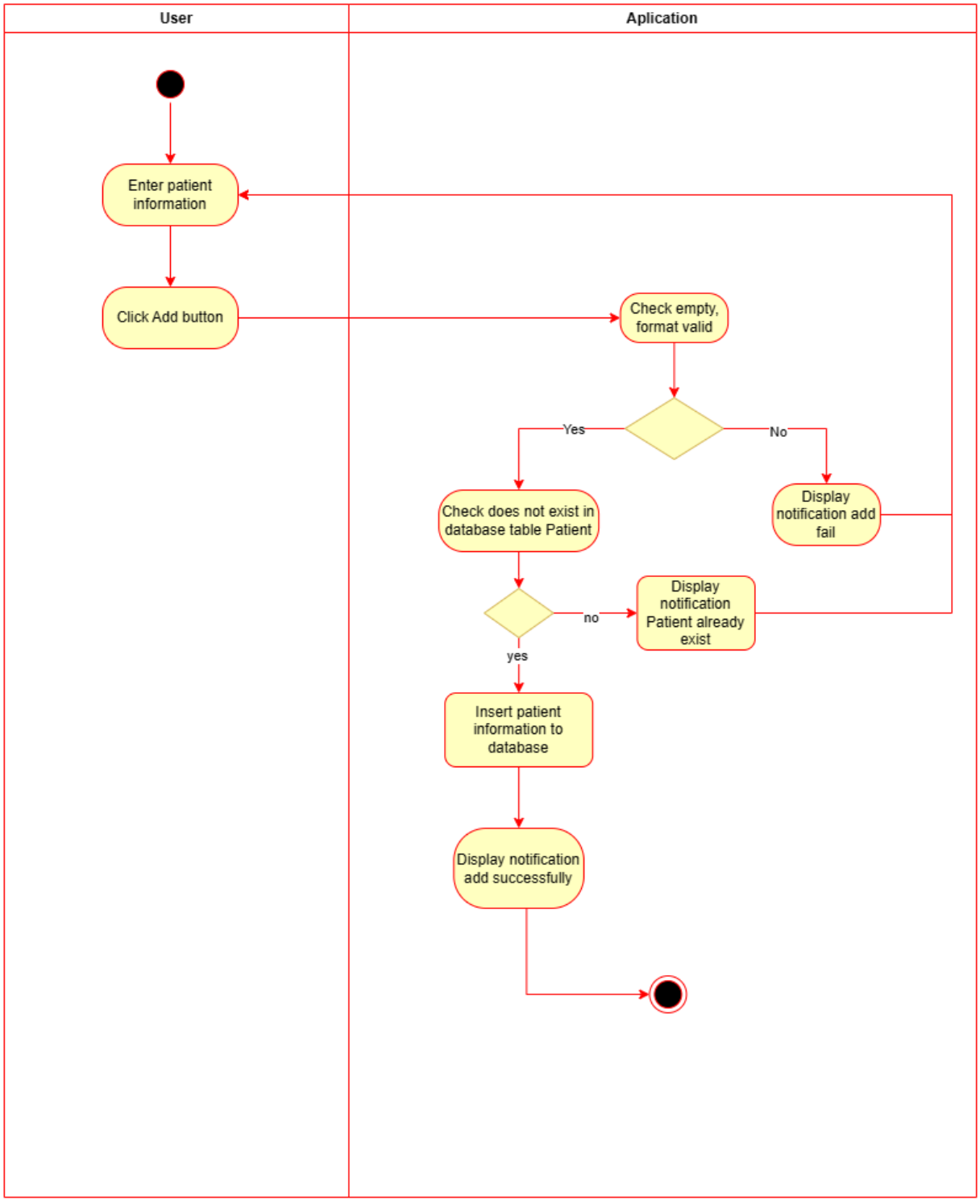
7.2.10. Delete holiday



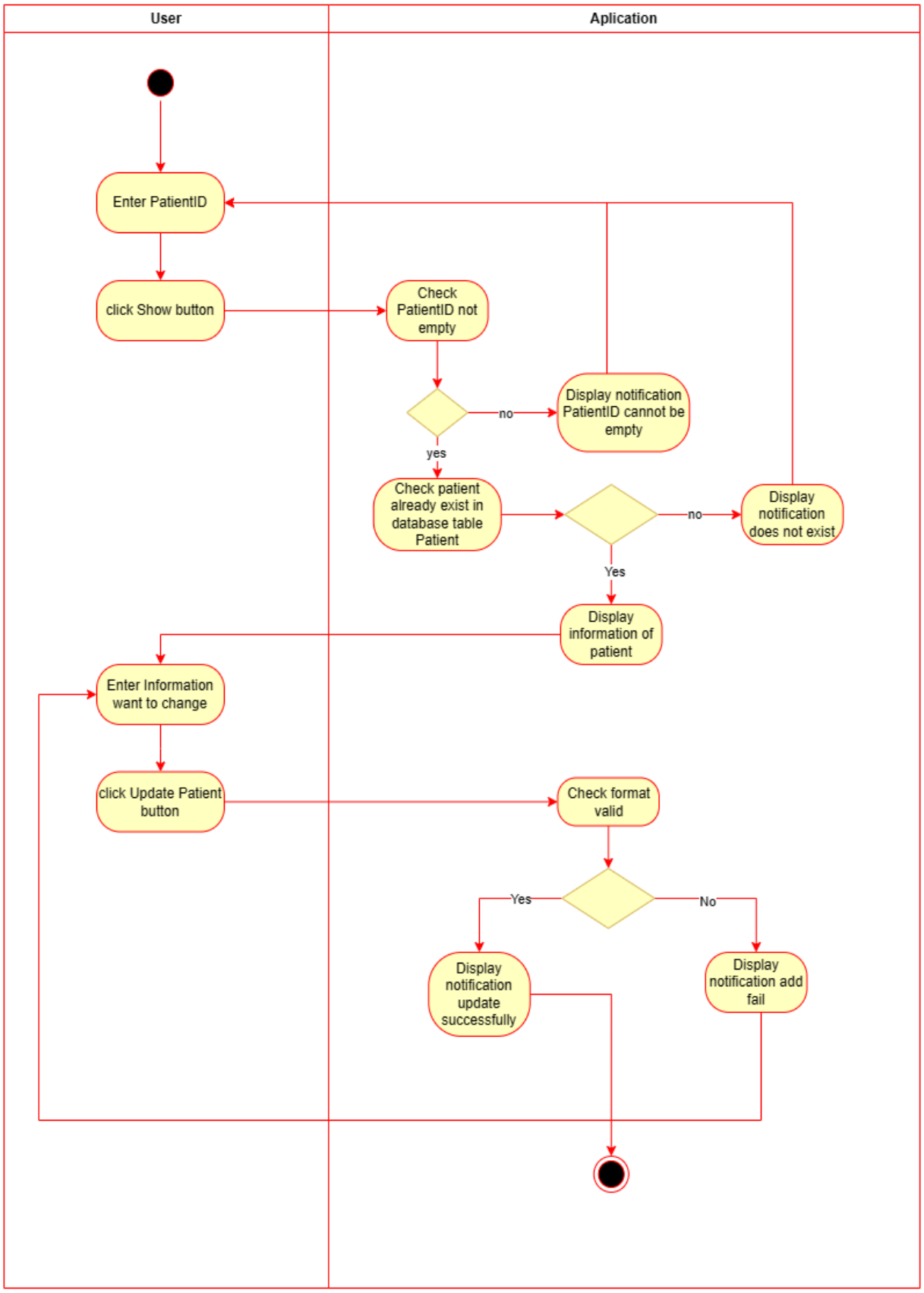
7.2.11. View list of holidays



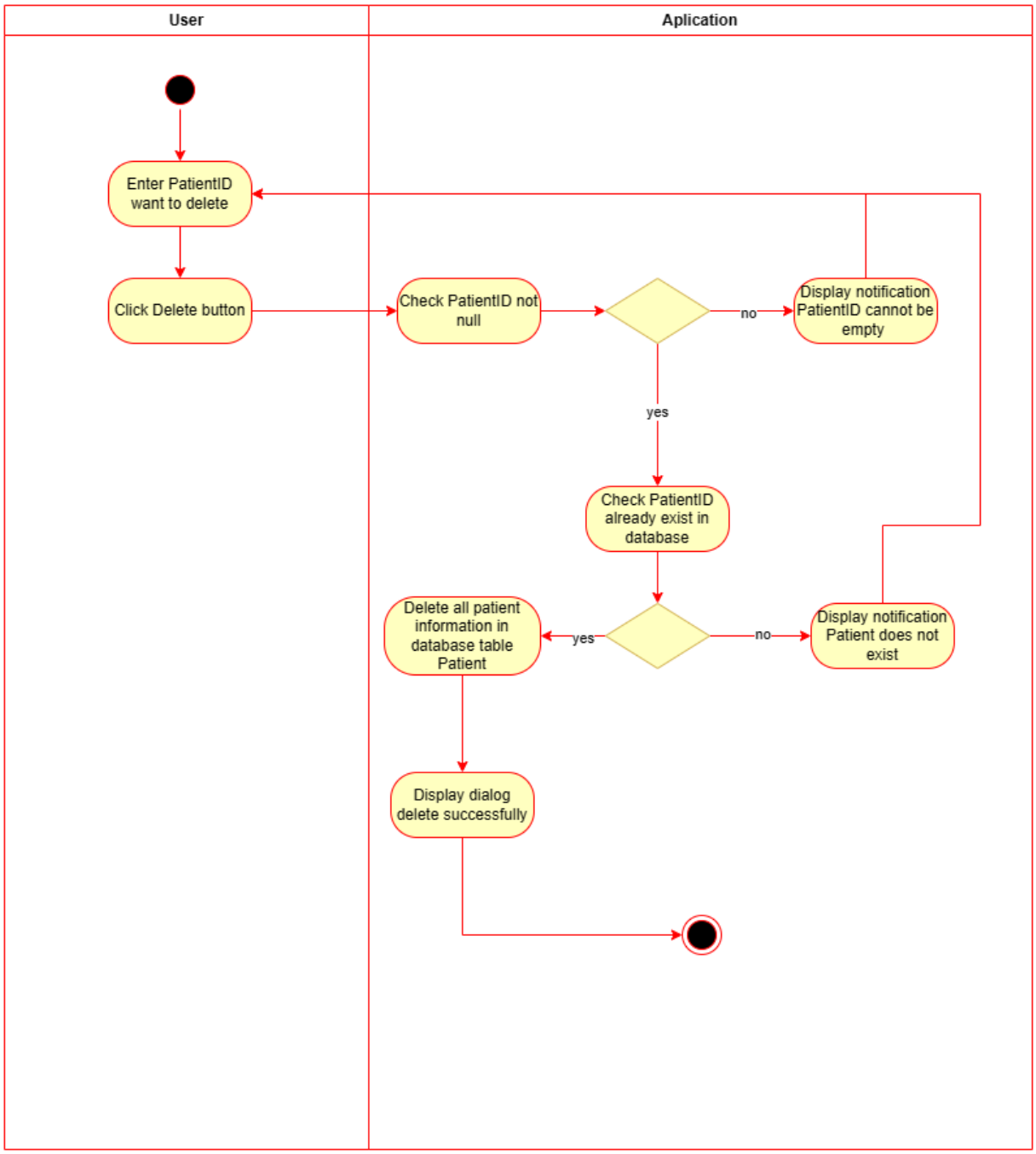
7.2.12. Add patient



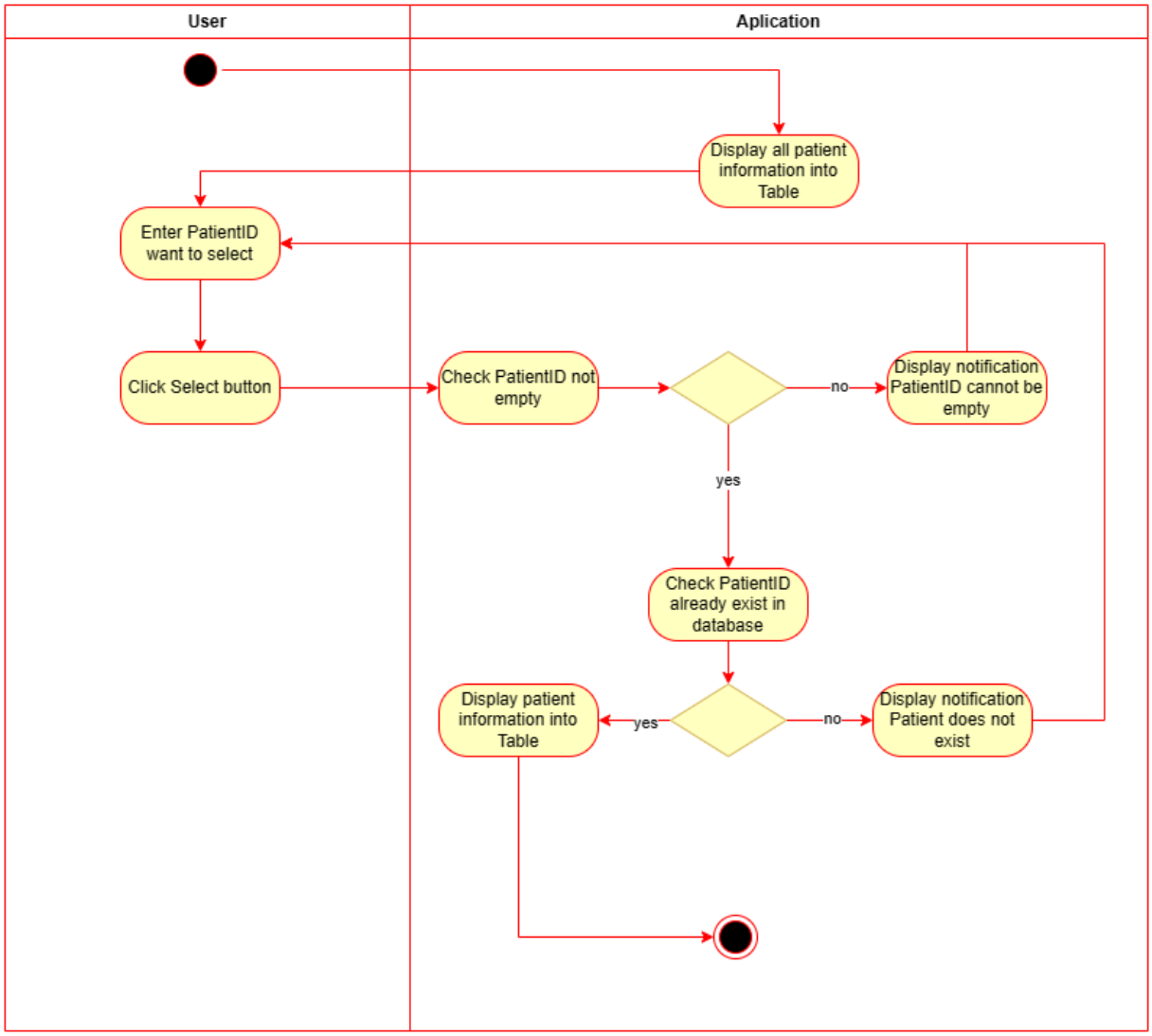
7.2.13. Update patient



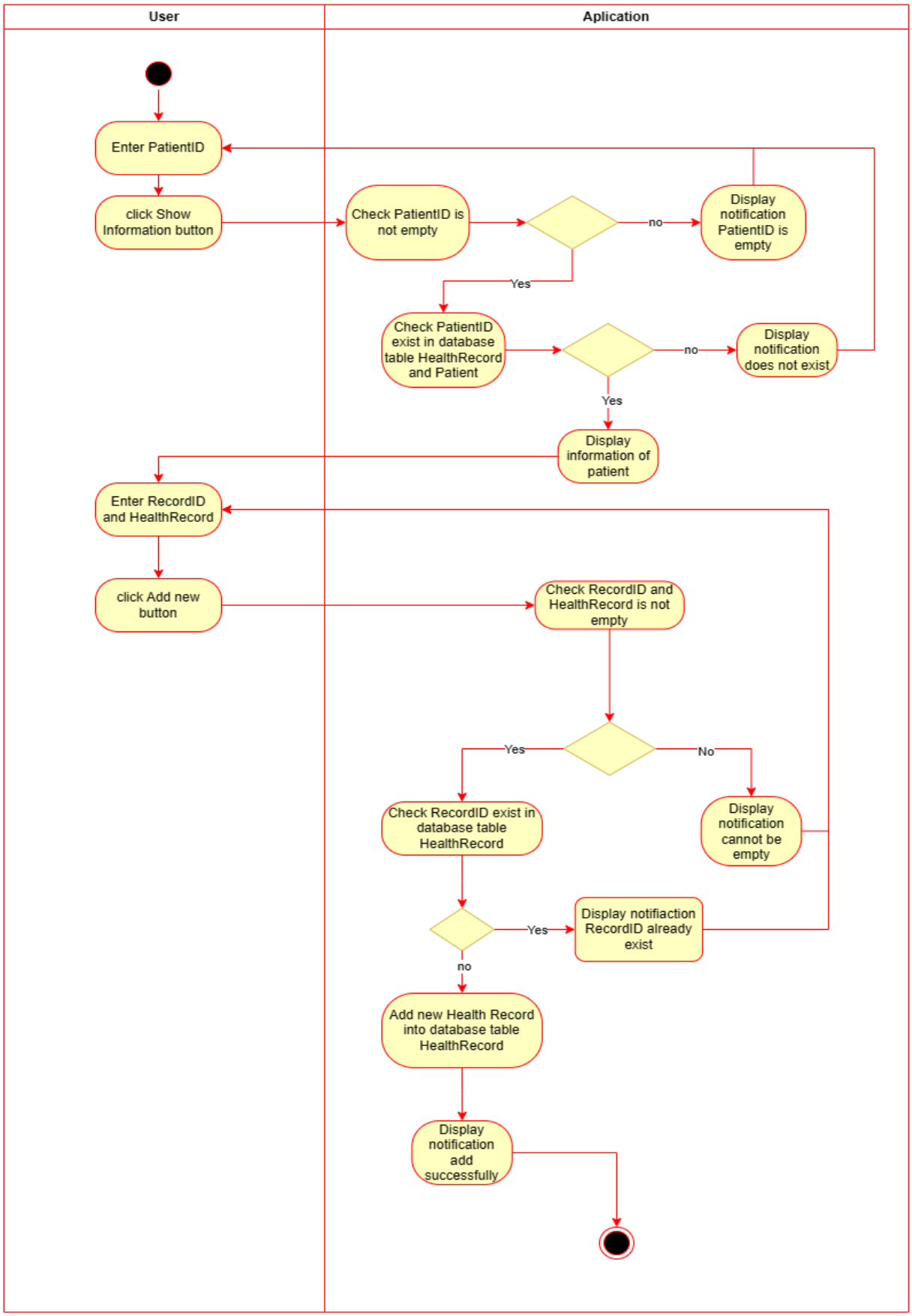
7.2.14. Delete patient



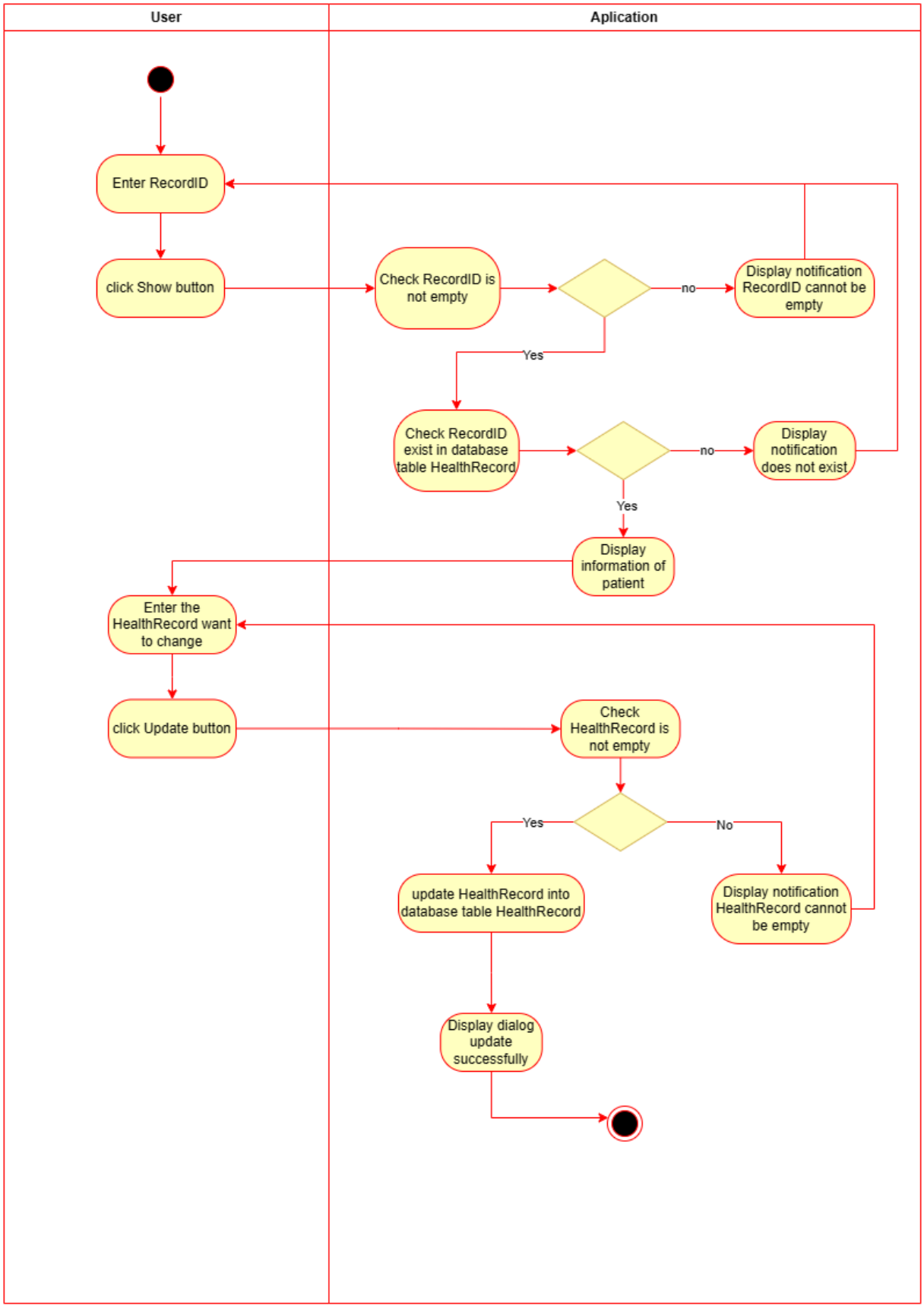
7.2.15. Select patient



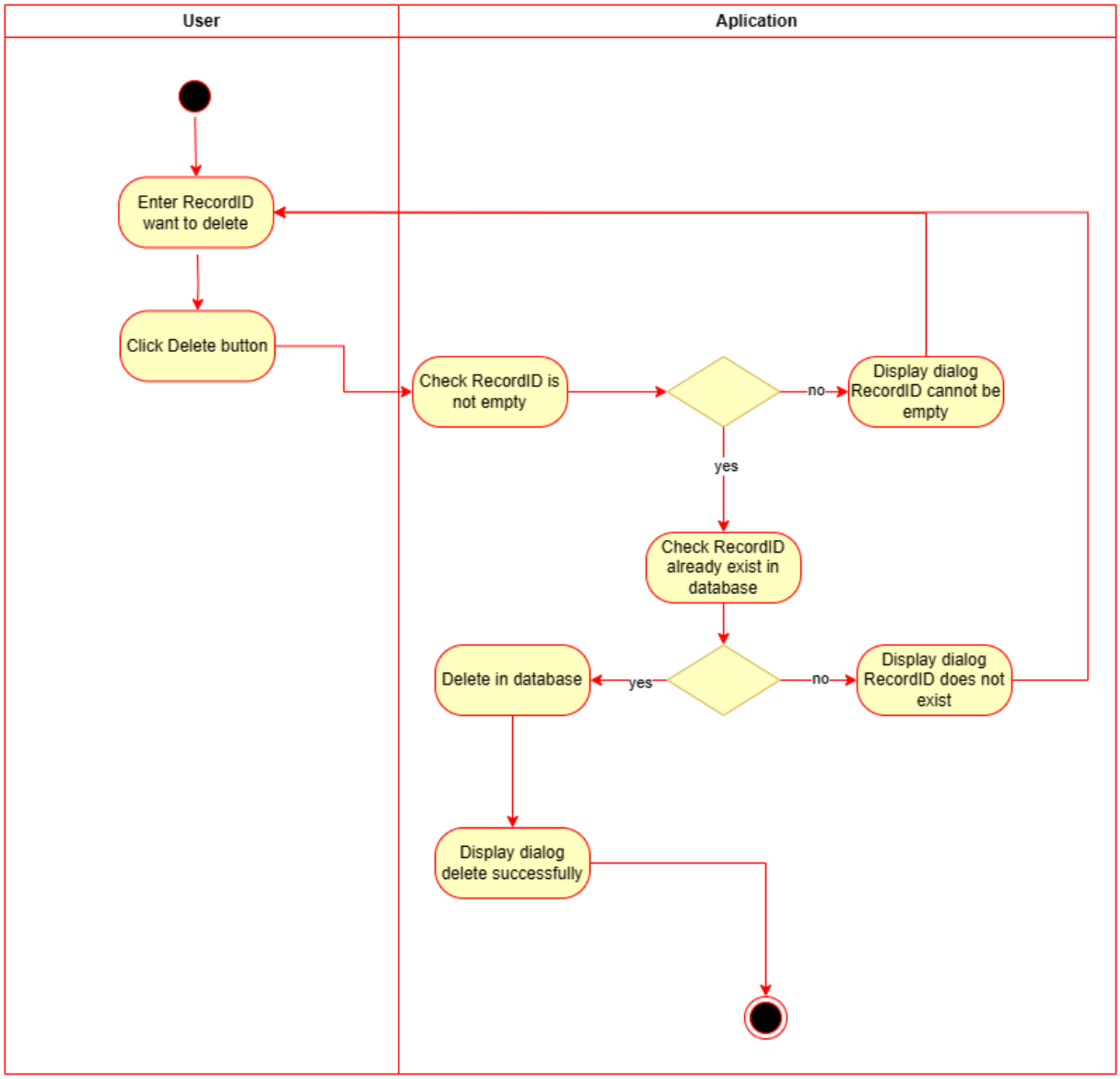
7.2.16. Add health record



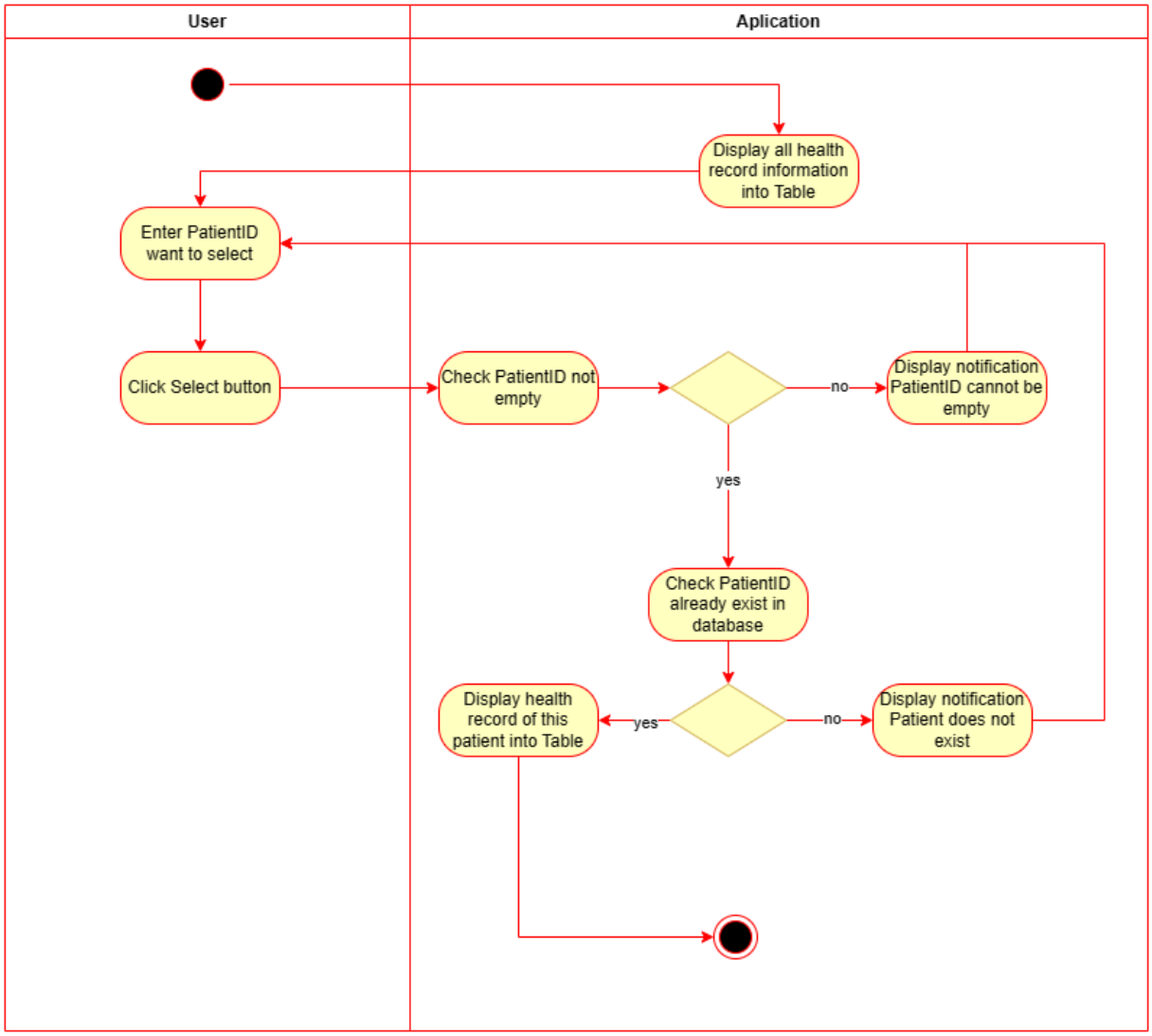
7.2.17. Update health record



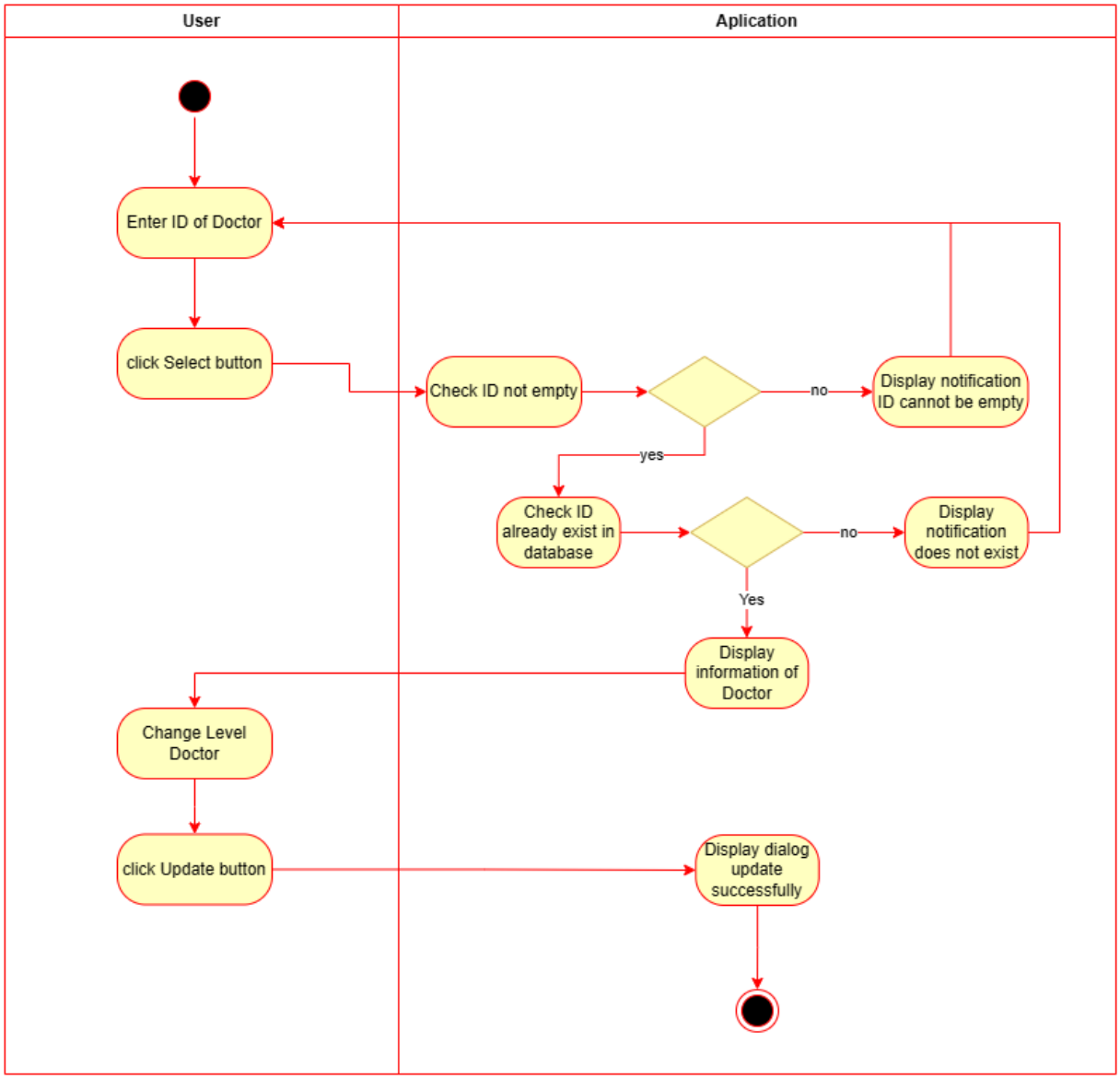
7.2.18. Delete health record



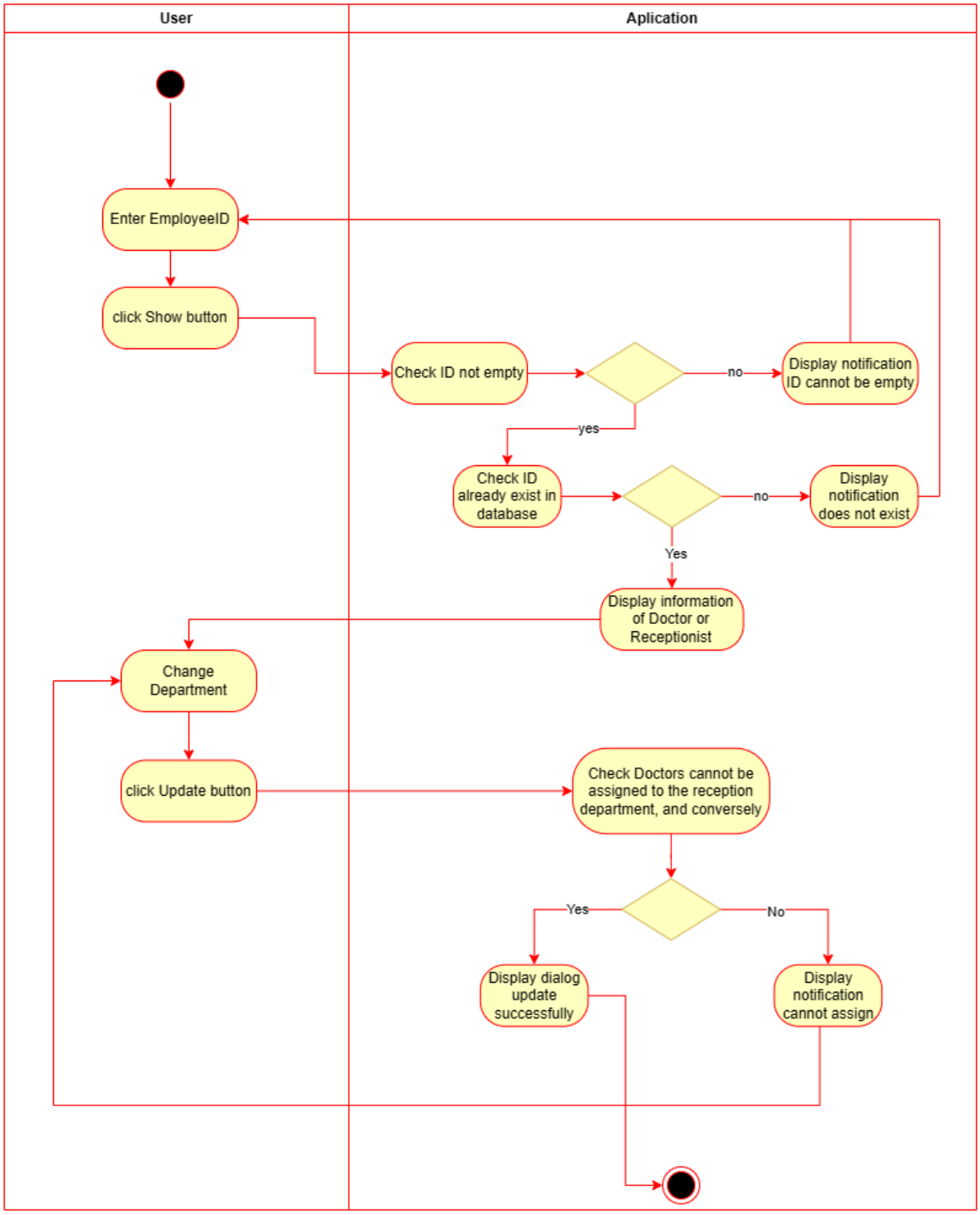
7.2.19. Select health record



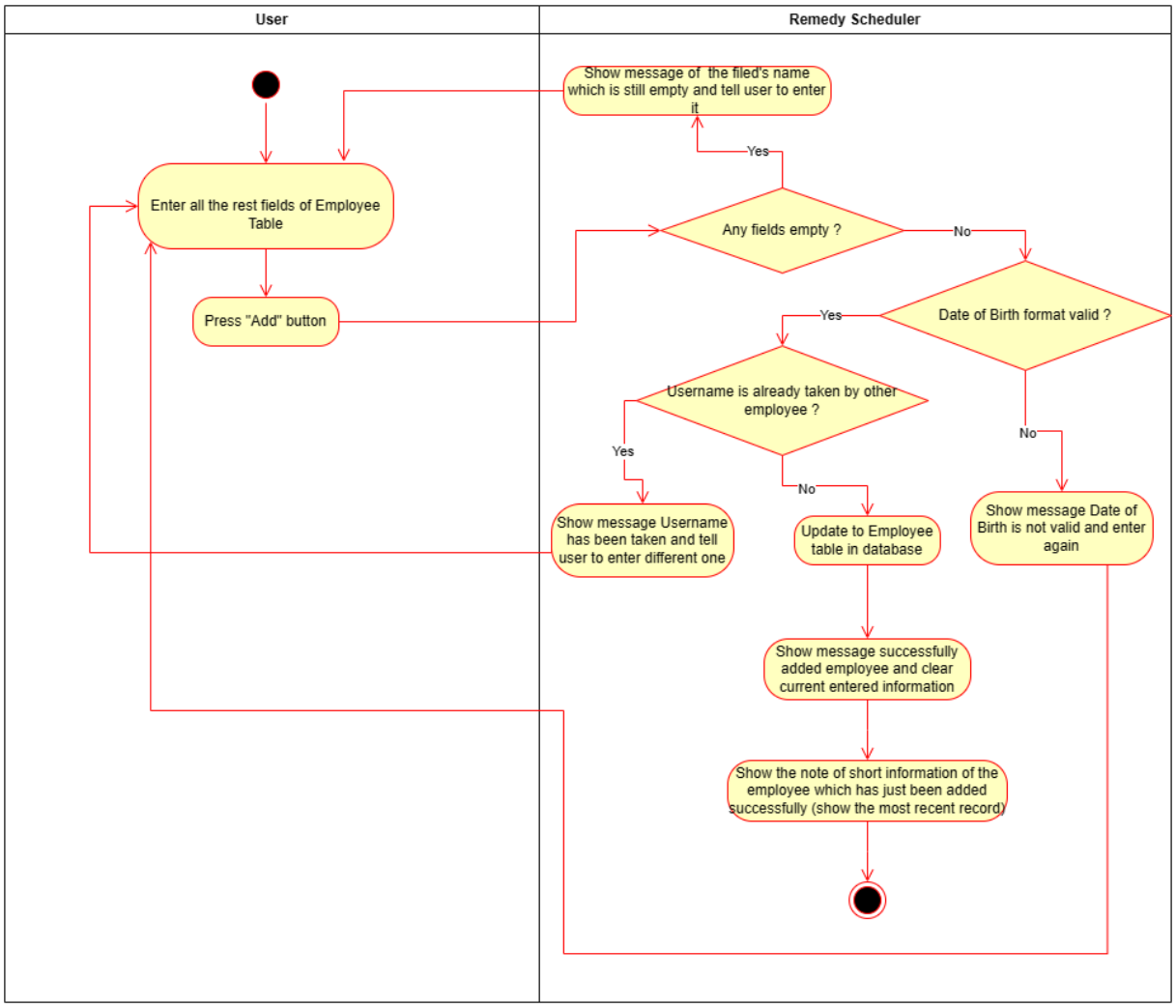
7.2.20. Assign level to doctors



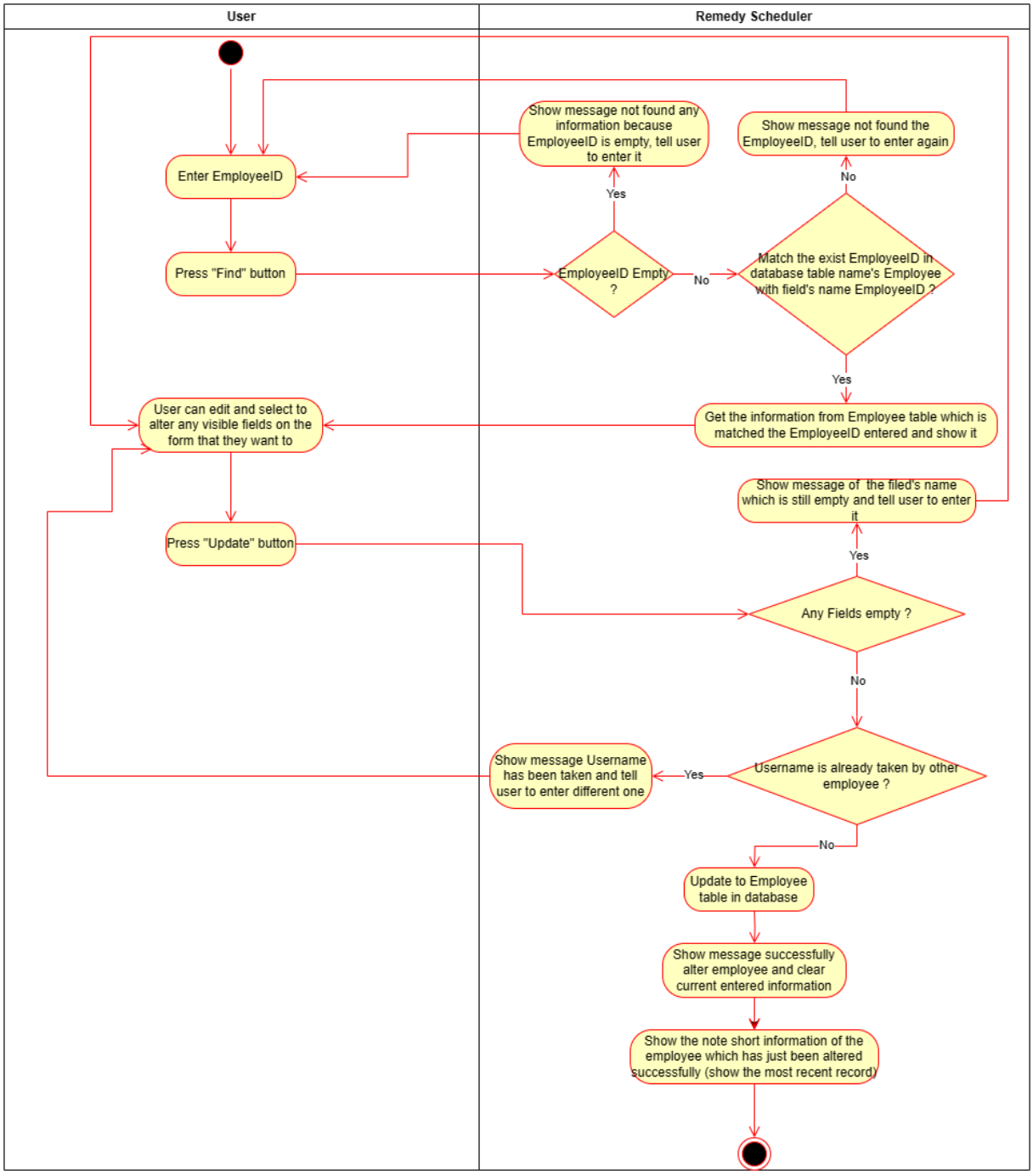
7.2.21. Assign doctors and receptionists to department



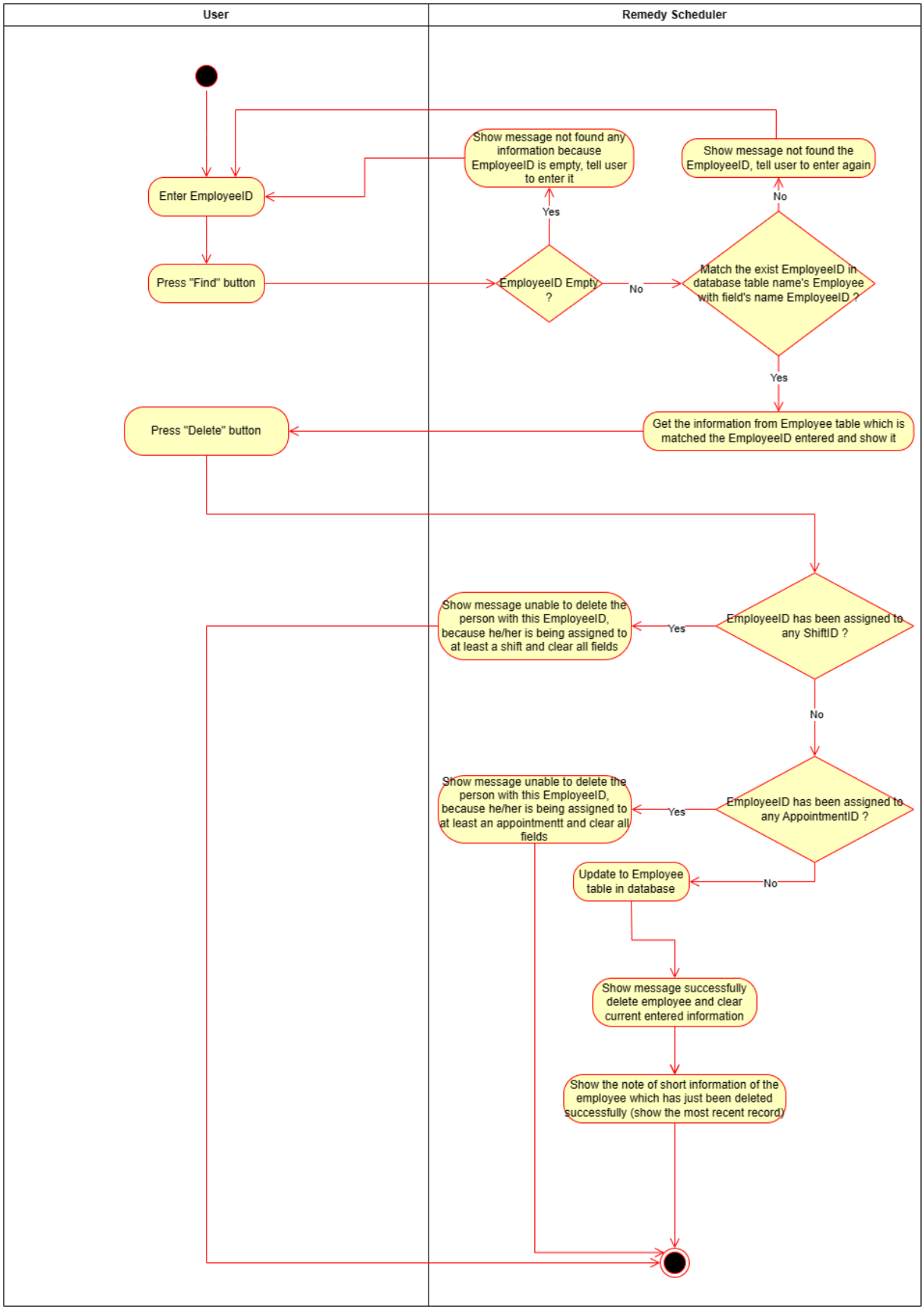
7.2.22. Add doctor and receptionist



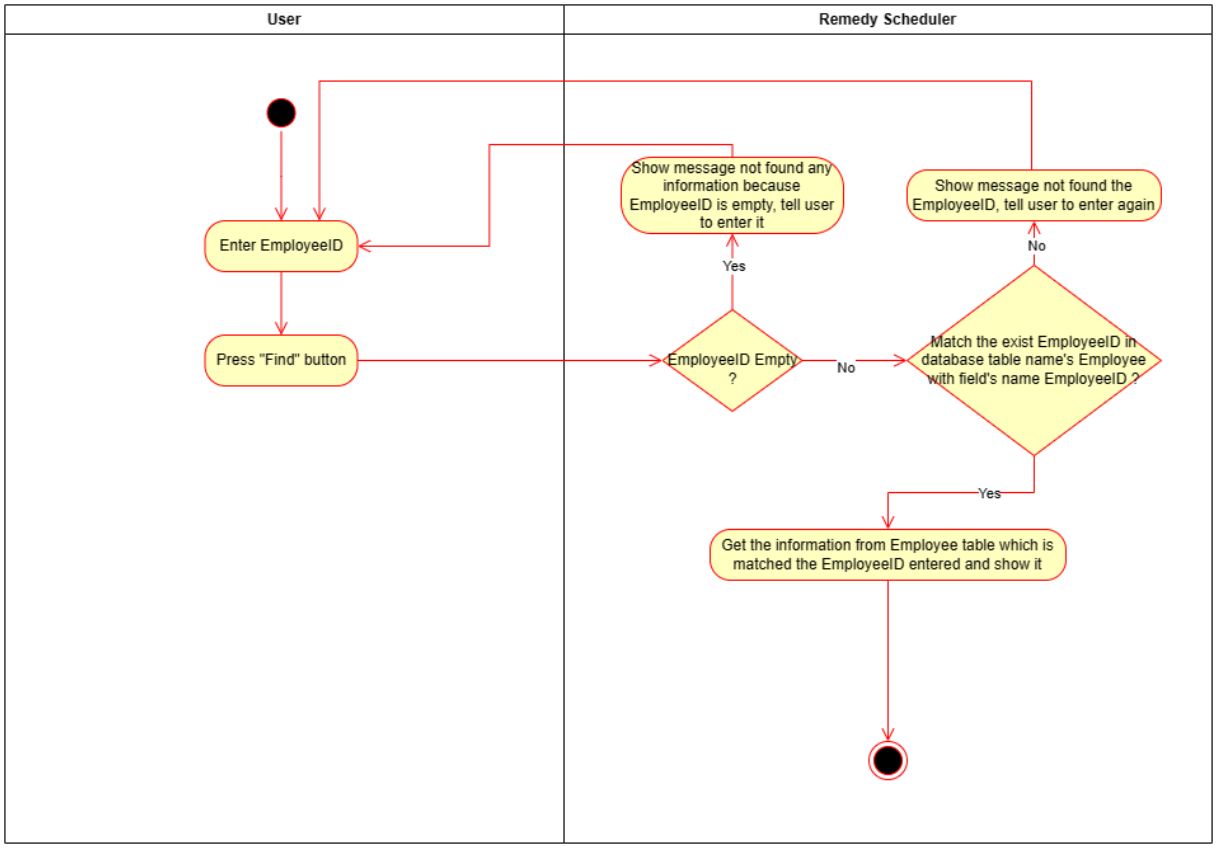
7.2.23. Alter doctor and receptionist



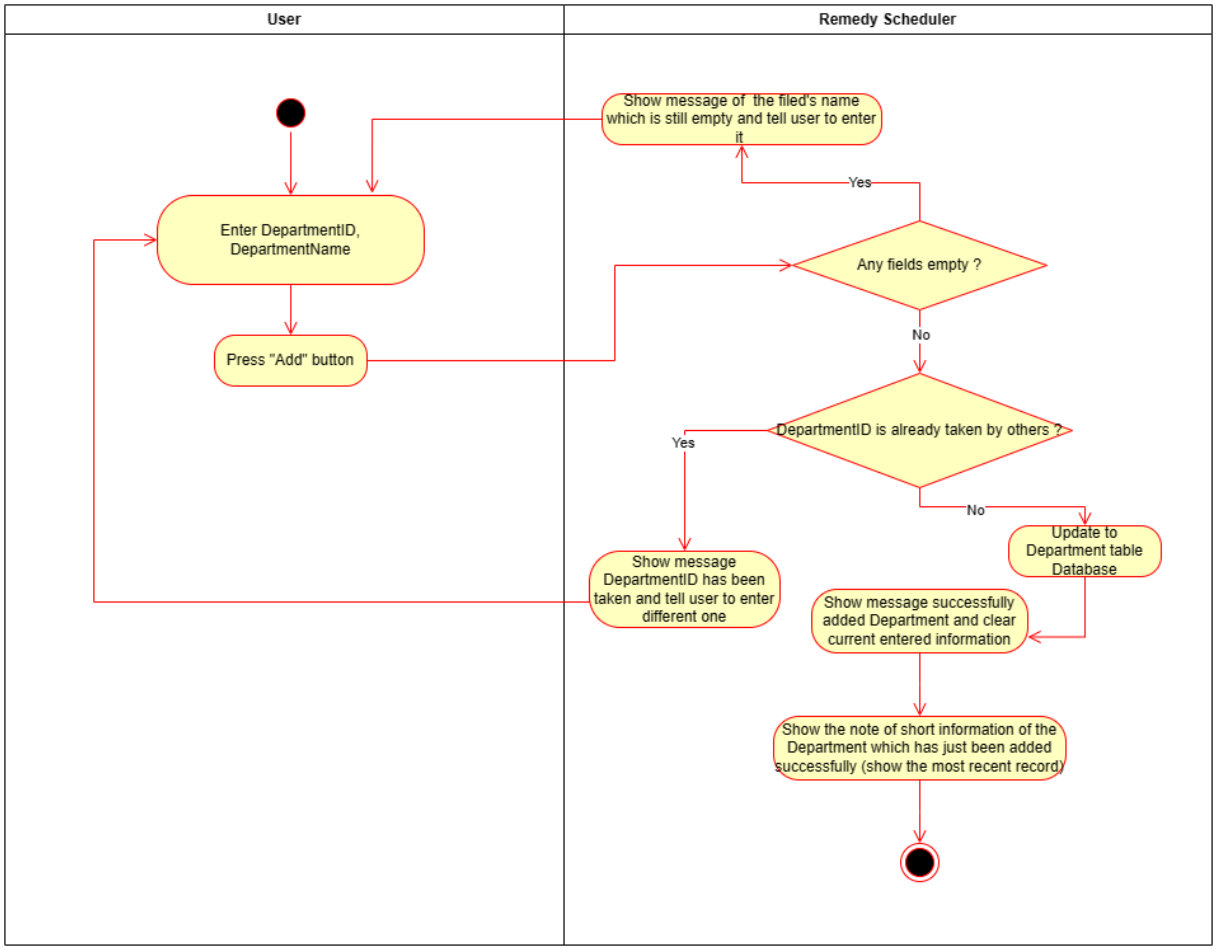
7.2.24. Delete doctor and receptionist



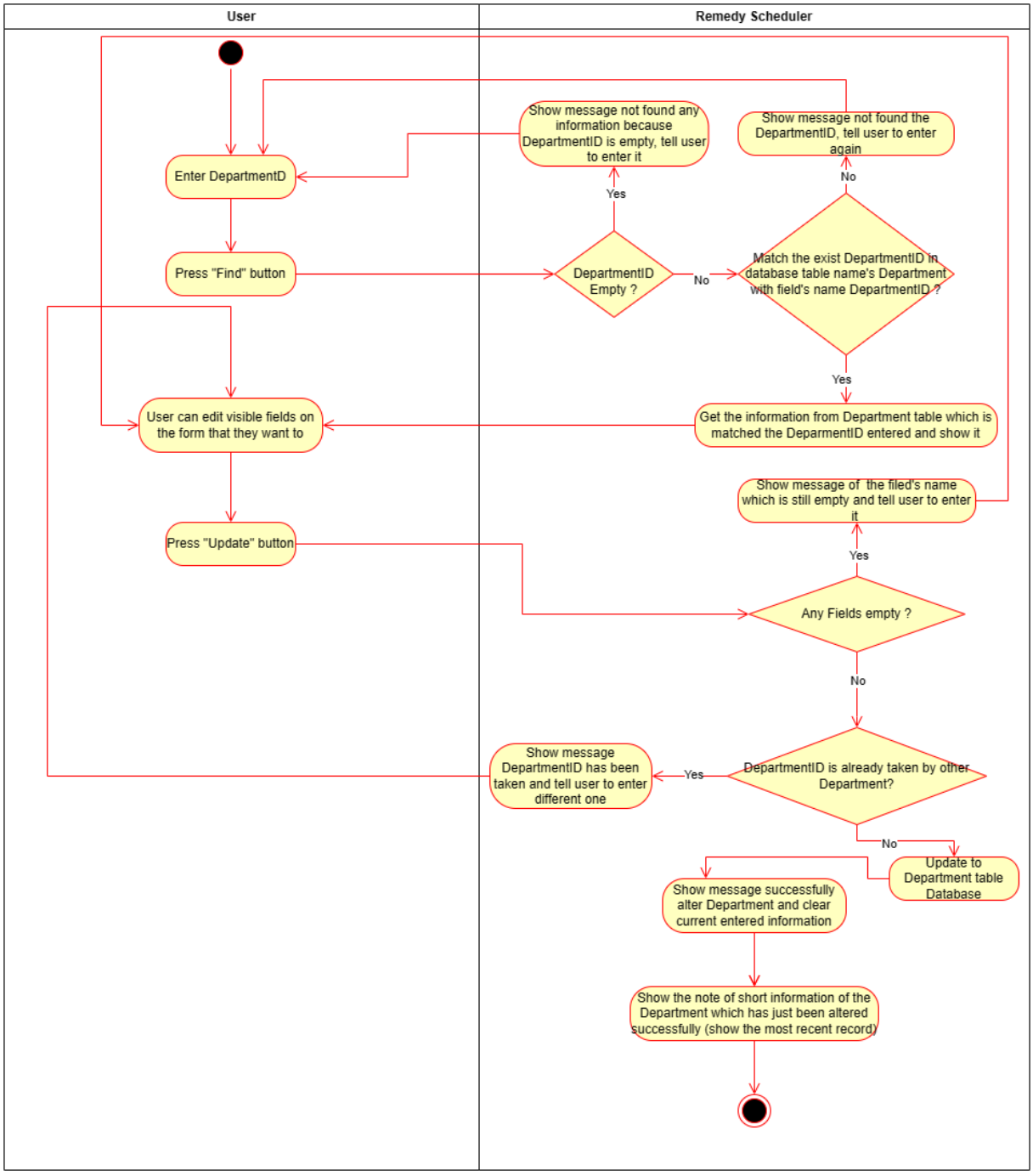
7.2.25. Search doctor and receptionist



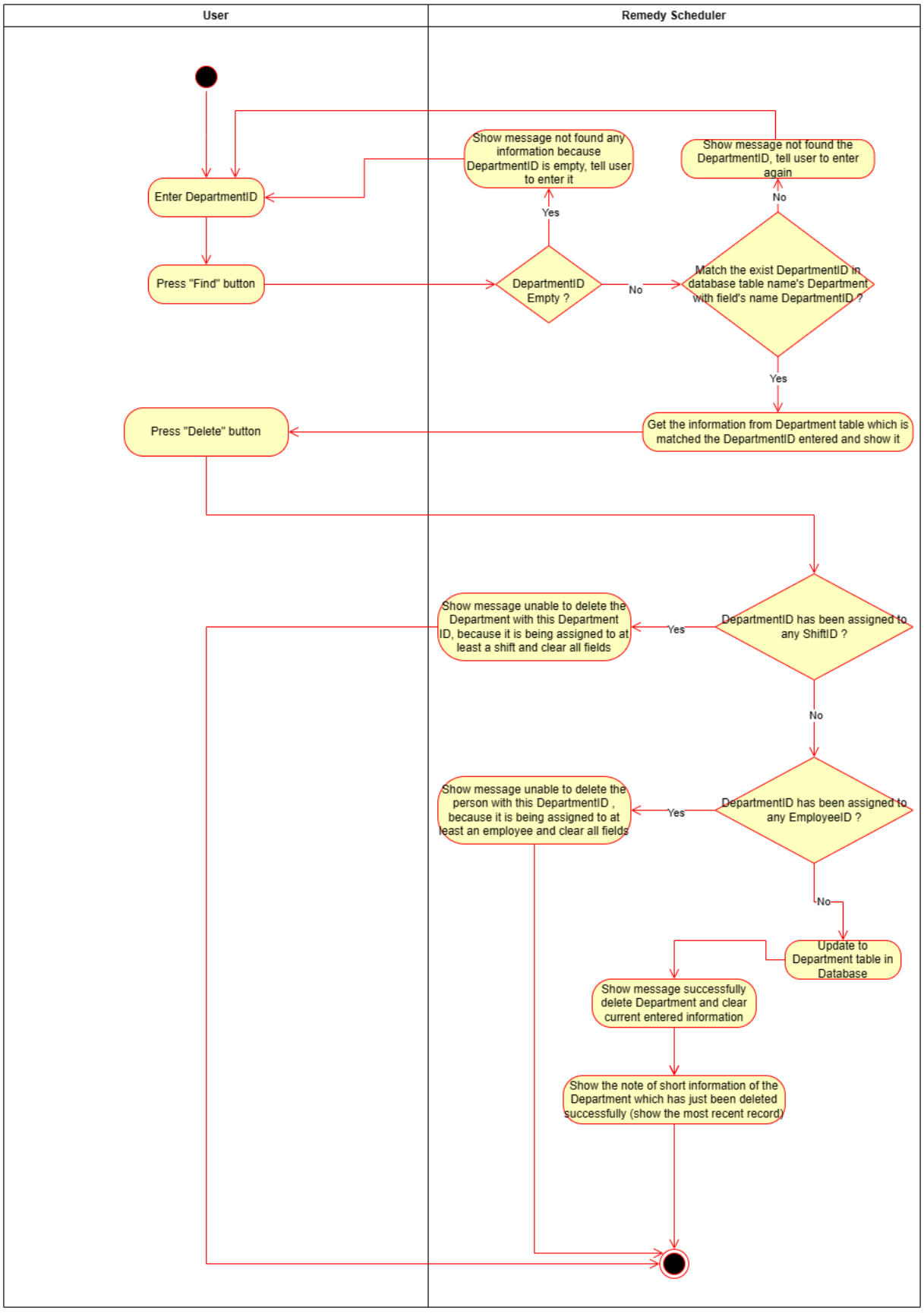
7.2.26. Add department



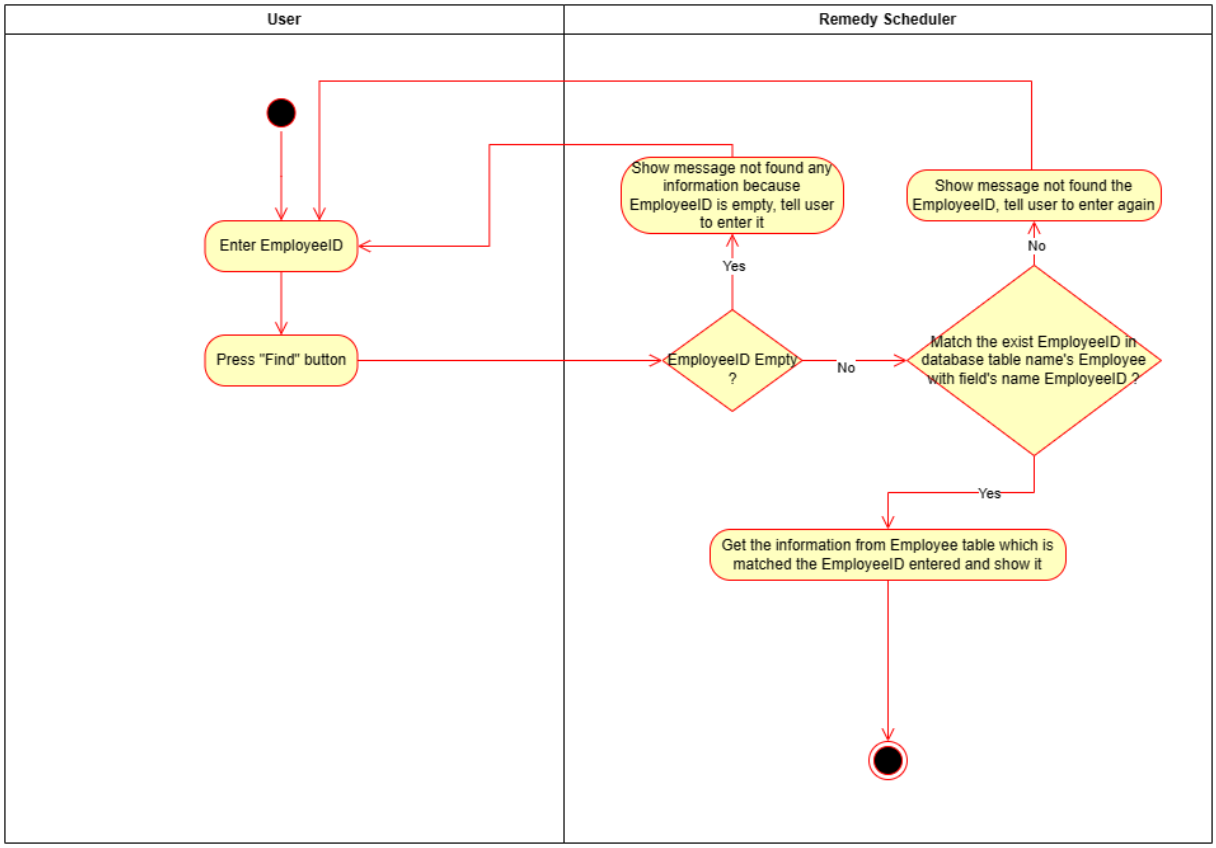
7.2.27. Alter department



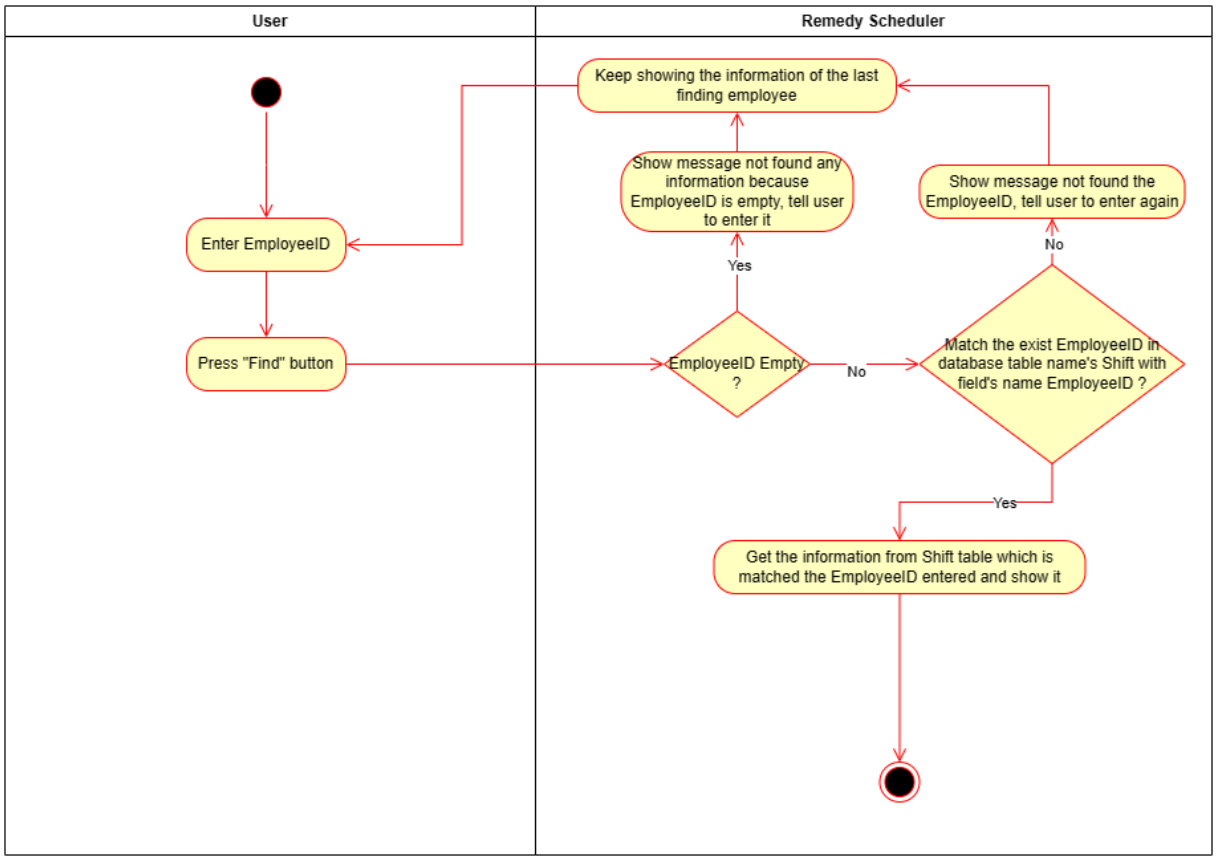
7.2.28. Delete department



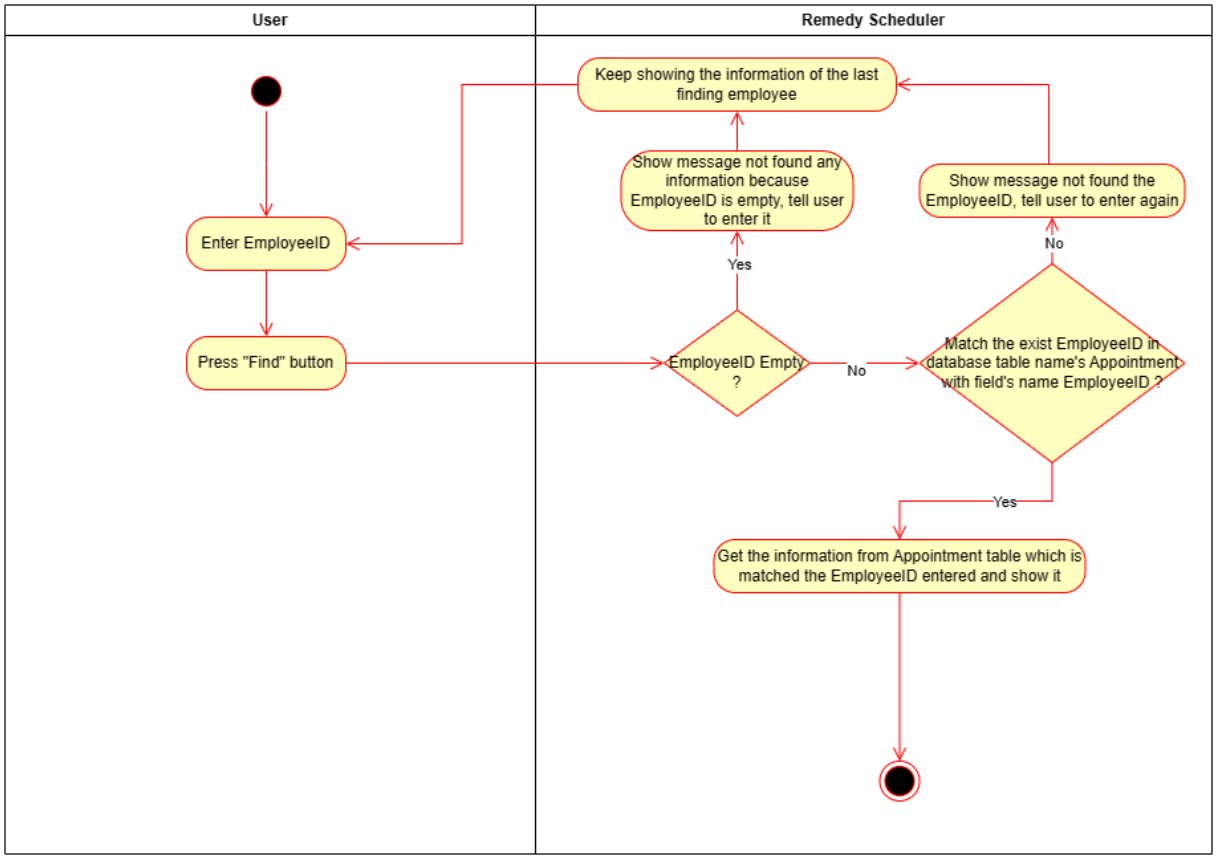
7.2.29. Search department



7.2.30. View time schedule



7.2.31. View appointment list



CODING STANDARDS DOCUMENT

I. Programming Standards

One of the fundamental aspects of maintaining a consistent and readable codebase is adhering to well-defined naming conventions. Properly chosen names for variables, functions, and classes can significantly enhance code clarity and promote effective collaboration among developers. In this project, we follow the below naming conventions:

1. **Classes:**
   - Use PascalCase for class names
   - Begin class names with an uppercase letter (e.g., CustomerOrder, FileManager)
   - Opt for meaningful and intuitive class names that accurately represent the entity or concept they model.
2. **Modules and Packages:**
   - Use lowercase letters for module and package names
   - Separate words with underscores (e.g., utils, data_processing)
   - Aim for concise yet meaningful names that reflect the module's purpose.
3. **Avoid Abbreviations and Acronyms:**
   - While brevity is important, avoid excessive use of abbreviations or acronyms
   - Prioritize clarity over brevity to ensure future developers can understand the codebase without ambiguity
4. **Comments and Documentation:**
   - Complement the chosen names with clear and concise comments to explain complex logic or intentions behind certain code sections

II. Standards for code writing style

- Use consistent indentation with either 4 spaces or tabs
- Keep each line of code within 80-100 characters
- Utilize comments to explain important, complex, or design-related parts of the code
- Maintain one statement per line
- Place opening braces at the end of the declaration line, not on a new line

III. Standards for declaring Variables

- Use camelCase for variable names

- Begin variable names with a lowercase letter (e.g., itemCount, calculateTotal)
- Choose descriptive names that convey the purpose or content of the variable
- Avoid using single-letter variable names unless they represent loop counters
- For constants, use uppercase letters with underscores (e.g., MAX_ITEMS, DEFAULT_TIMEOUT)

IV. Standards for Function Declarations

- Use camelCase for functions names
- Begin variable names with a lowercase letter (e.g., run, getName())
- Choose descriptive names that convey the purpose or content of the function
- Specify data types for each parameter

V. Other Standards

TASK SHEET

I. Project Title: Remedy Schedule

No.	Task	Actual Start Date	Actual Days	Teammate Name	Status
1	Form 1	2023/08/25	4	Loc Nguyen Phuc	Done
2	Form 2A	2023/08/25	4	Duy Nguyen	Done
3	Form 2B	2023/08/25	4	Hung Nguyen Duc	Done
4	Form 3	2023/08/25	4	Phuc Loc Nguyen	Done
5	Form 4	2023/08/25	4	Loc Nguyen Phuc	Done
6	Form 5	2023/08/30	7	Anh Le Huy	Done
7	Form 6	2023/08/30	7	Hung Nguyen Duc	Done
8	Form 7	2023/08/30	7	Anh Le Huy	Done
9	Form 8	2023/08/30	7	Duy Nguyen	Done
10	Form 9	2023/09/26	1	Duy Nguyen	Done
11	Form 10	2023/09/24	2	Hung Nguyen Duc	Done
12	Form 11	2023/08/25	3	Duy Nguyen	Done
13	Form 12	2023/09/20	5	Anh Le Huy	Done
14	Synthetic code	2023/09/15	11	Duy Nguyen	Done
15	Synthetic document	2023/09/20	6	Hung Nguyen Duc	Done
16	Database	2023/08/30	7	Hung Nguyen Duc	Done

17	Assign shifts	2023/09/08	14	Hung Nguyen Duc	Done
18	Generate shifts	2023/09/08	14	Hung Nguyen Duc	Done
19	Log in	2023/09/08	14	Hung Nguyen Duc	Done
20	Log out	2023/09/08	14	Hung Nguyen Duc	Done
21	Generate shift requests	2023/09/08	14	Duy Nguyen	Done
22	Approving-Disapproving the shift request	2023/09/08	14	Duy Nguyen	Done
23	Check status of shift requests	2023/09/08	14	Duy Nguyen	Done
24	Manage holidays	2023/09/08	14	Duy Nguyen	Done
25	Manage patients	2023/09/08	14	Loc Nguyen Phuc	Done
26	Manage records of patients	2023/09/08	14	Loc Nguyen Phuc	Done
27	Assign level to doctors	2023/09/08	14	Loc Nguyen Phuc	Done
28	Assign doctors and receptionist to Department	2023/09/08	14	Loc Nguyen Phuc	Done
29	View time schedule	2023/09/08	14	Anh Le Huy	Done
30	View list of appointments	2023/09/08	14	Anh Le Huy	Done
31	Manage doctors and receptionists	2023/09/08	14	Anh Le Huy	Done
32	Manage departments	2023/09/08	14	Anh Le Huy	Done
33	Installation guide	2023/09/22	4	Hung Nguyen Duc	Done
34	User manual	2023/09/22	4	Hung Nguyen Duc	Done

35	Main menu	2023/09/08	7	Duy Nguyen	Done

TESTING DOCUMENT

ID	Test Case Name	Task	Result	
			Expected	Actual Result Pass/Fail-Comments
1	Login	Enter login details • Username: e04username • Password: e04password	Show menu window for department heads	Pass
		Enter login details • Username: [multiple spaces] • Password: e01password	Notify username and password cannot be empty or filled with spaces	Pass
2	Logout	Press logout button	Show login window	Pass
3	Generate new shift	Enter/choose shift details: • Department: Operating Theatre • Room ID: R6 • Date: 2023-09-10 • Type: Afternoon Press submit button	Notify please check shift date	Pass
		Same as above but with different date 2023-10-30	Add new shift to the Shift table with ShiftID S17, Employee ID null, Department ID D5, RoomID R6, Type Afternoon, Date 2023-10-30 and notify new shift generated successfully	Pass
4	Assign Shift	Enter details: • Starting date: 2023-09-10	Notify no shift to assign in the date range	Pass

		• Ending date: 2023-09-15 Press select		
		Enter details: • Starting date: 2023-10-30 • Ending date: 2023-11-05 Press select Select Shift ID S17 Select Sieuwerd Chaudhari Press Submit	Update EmployeeID E9 to the record having ShiftID S17 in the Shift table and notify shift assigned successfully	Pass
5	Generate a request for shift change	Select a current shift and a desired then click generate request	Notify enter reason	Pass
		Same as above but enter reason	Add new request to the ChangeRequest table with provided details and have status Pending and notify request generated successfully	Pass
6	Holiday management	Select view all	Show table with all holidays and their details	Pass
		Select add holiday Enter details: • Holiday name: Test holiday • Start Date: 2023-09-26 • End Date: 2023-09-27 Reason: Test description	Add new holiday with provided details to the Holiday table and notify holiday added successfully	Pass
		Select alter holiday	Update the description of Test	Pass

		Select Test Holiday Edit description to Edited description Click update Click Yes	Holiday to Edited description and notify holiday updated successfully	
		Select delete holiday Select Test holiday Click delete Click yes	Delete the Test holiday from the Holiday table and notify holiday deleted successfully	Pass
6	Shift request approval	Select shift request of Rupa Kumiega Click approve Click Yes	Update the status of the change request to Approved and exchange the Employee ID of 2 record having ShiftID S15 and S16 in the Shift table	Pass
7	Check status of shift request	Login as Rupa Kumiega (E3) Select Check Shift Request Status	Show the shift request details	Pass
8	Manage patient	Select Add Enter details: Patient ID: P30 Patient Name: Test name Gender: Male Phone: 012346789 Address: Test address Day of birth: 2000-01-01 Click Add new patient	Add new patient with provided details to the Patient table	Pass

		Select Update Enter Patient ID P30 Click Show Change: • Patient Name to Edited Name • Gender to Female Note to Added note Click Update patient	Update the PatientName, Gender and Note of the patient having PatientID P30 in the Patient table to provided information	Pass
		Click Select Enter P30 Click Select	Show details of the patient having PatientID P30 in the Patient table	Pass
		Click Delete Enter P30 Click Delete	Delete the record having PatientID P30 from the Patient table	Pass
9	Manage patient records	Click Add Enter Patient ID P10 Click Show Information Enter RecordID H11 Enter Health Record Test Click Add new	Notify the new health record added successfully and add the new health record with provided details and today as CreateDate to the HealthRecord table	Pass
		Click Select Enter P10 Click Select	Show all health records of the patient having PatientID P10 in the Patient table	Pass
		Click Update Enter RecordID H11	Update the Problem of the health record having RecordID H11 in the HealthRecord	Pass

		Click Show Change: Health Record to Record Click Update	table to Record and notify health record updated successfully	
		Click Delete Enter Health RecordID H11 Click Delete	Delete the health record having RecordID H11 in the HealthRecord table and notify health record deleted successfully	Pass
10	Assign doctor level	Enter Doctor ID E10 Click Select Select Assign to Department Head Click Change	Update the Position of the employee having EmployeeID E10 in the Employee table to Department Head and notify position updated successfully	Pass
11	Assign doctors and receptionists to department	Enter Employee ID E10 Click Show Information Select Assign to Department Operating Theatre Click Change	Update the DepartmentID of the employee having EmployeeID E10 in the Employee table to D5 and notify department ID updated successfully	Pass
12	Manage doctors and receptionists	Click Add Enter or select details: • Department: D3 • EmployeeName: Test Name • Gender: Female • Position: Doctor • Address: Test address • Employee phone: 0123456789	Notify created successfully, add the new employee with provided details to the Employee table and show recently added employee's information in the The most recently Added Employee table	Pass

		• Employee Date of Birth (yyyy-MM-dd): 2000-01-01 • Email: testEmail@gmail.com • Employee Specialty: Test specialty • Username: e??username • Password: e??password Click Add		
		Click Alter Enter Employee ID E11 Click Find Change: Employee name to name Gender to Male Address to address Password to password Click Update	Notify altered successfully and update the EmployeeName, Gender, Address, Password the employee having EmployeeID E11 in the Employee table to provided information and show recently updated employee's information in the The most recently Altered Employee table	Pass
		Click Search Enter EmployeeID E11 Click Find	Notify employee successfully found and show information of the employee having EmployeeID E11 in the Employee table	Pass
		Click Delete Enter EmployeeID E11 Click OK	Notify deleted successfully, delete the employee having EmployeeID E11 in the Employee table	Pass

		Click Delete Click Yes	and show recently deleted employee's information in the The most recently Deleted Employee table	
13	Manage departments	Click Add Enter details: • DeaprtmentID: D6 • Department Name: Test name Click Add	Notify created successfully, add the new department with provided details to the Department table and show recently added department's information in the The most recently Added Department table	
		Click Alter Enter Department ID D6 Change Department Name to Test department Click Update	Notify altered successfully and update the DepartmentName of the department having DepartmentID D6 in the Department table to provided information and show recently updated department's information in the The most recently Altered Department table	Pass
		Click Search Enter EmployeeID D6 Click Find	Notify department successfully found and show information of the department having DepartmentID D6 in the Department table	Pass
		Click Delete Enter DepartmentID D6	Notify deleted successfully, delete the department having	Pass

		Click Find Click Delete Click Yes	DepartmentID D6 in the Department table and show recently deleted employee's information in the The most recently Deleted Department table	
14	View time schedule	Enter EmployeeID E10 Click Find	Notify employee found and show shift schedule of the employee having EmployeeID E10 from the Shift table	Pass
15	View appointment list	Enter EmployeeID E4 Click Find	Notify employee found and show appointments of the employee having EmployeeID E4 from the Appointment table	Pass

PROJECT REVIEW

Date	Project Plan Activity/Milestone	Work Specification	Status of the Activity	Remarks	Responsibility
2023/08/27	Requirements	Collect documents and references Determine what needs to be done Assign expected work to the team (document, demo) Discuss form 1, 2, 6, 9	Done		Duy Nguyen
2023/09/04	Design, development	Discuss form 1, 2, 3, 4, 5, 6, 7, 9 Discuss database	Done		Duy Nguyen
2023/09/11	Design, development	Build general application interface (temporarily unfunctional)	Done		Duy Nguyen
2023/09/18	Development	Complete 90% of the demo and documents	Done		Duy Nguyen
2023/09/24	Testing	Test the application Discuss bugs appearing during test	Done		Duy Nguyen
2023/10/02	Deployment	Install the application,	Done		Duy Nguyen

		check the final demo			

FINAL CHECK LIST

No.	Aspect tested	Suggestion/Remarks
1	Have all the modules been property integrated and are they completely functional?	Yes
2	Are all the validations happening as specified in Process Design?	Yes
3	Have all Design and Coding standards been followed and implemented?	Yes
4	Have all the GUI forms performed the functions smoothly and exactly to the purpose of the initiation?	Yes
5	Have all the GUIs forms contained enough functional processes to run from the beginning to the end on a run?	Yes
6	Are all the GUIs easy-looking to use, the instruction detail is good enough for the user to follow?	Yes
7	Are there any bonus functional processes added to help users get to use the program easier?	No