|  |  |  |
| --- | --- | --- |
|  | **CAN THO UNIVERSITY SOFTWARE CENTER** |  |
| **MEKONG DELTA - APTECH** |

**Aptech Certified Computer Professional (ACCP)**

**SEMESTER II**

***Clinic Management System***

|  |  |
| --- | --- |
| **Instructor:** | **Class: CP2396M02** |
| Mr. Nguyễn Võ Thông Thái | **Team:03** |
| **Students:**  1. A22088 - Thạch Chanh Mô Ni  2. A22084 - Nguyễn Hoàng Khang  3. A22081 - Đỗ Thanh Trọng |

Can Tho, 30/05

**PROBLEM DEFINITION OF PROJECT**

The Clinic Management System (CMS) is a comprehensive application designed to facilitate efficient management of healthcare facilities and operations. It offers a wide range of capabilities and benefits specifically tailored to support administrative tasks and streamline clinic operations.

One of the key capabilities of the Clinic Management System is its ability to centralize and organize patient records, appointments, and medical histories. This allows healthcare providers and administrators to access critical patient information quickly and easily, enabling them to make informed decisions and provide timely care.  
 Another important feature of the CMS is its appointment scheduling functionality. The system automates the appointment booking process, allowing staff to manage appointments more efficiently and reduce scheduling conflicts. Additionally, it can send automated reminders to patients, helping to minimize no-shows and optimize clinic utilization.

Furthermore, the CMS includes robust billing and invoicing capabilities. It can generate invoices for services rendered, track payments, and manage billing cycles, streamlining the financial aspect of clinic management. This helps ensure accurate and timely payments, improving revenue collection and financial stability.

Additionally, the CMS offers inventory management functionality to track and manage clinic supplies, medications, and equipment. It allows administrators to monitor stock levels, reorder supplies as needed, and track usage and expenses. This helps prevent stock-outs and wastage, optimizing inventory control and reducing costs

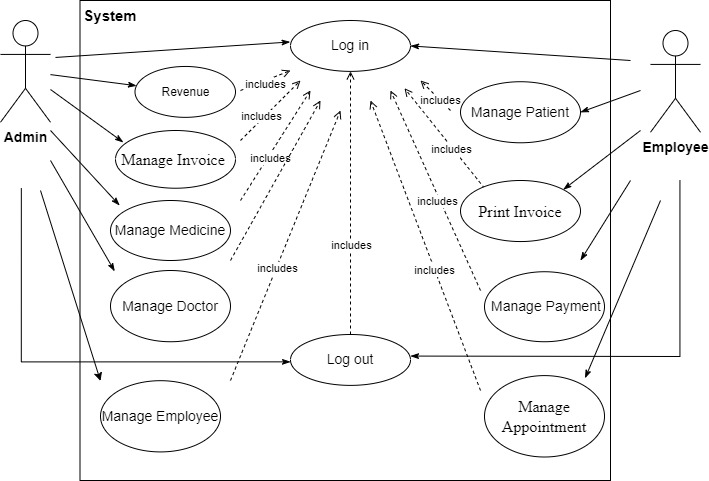
Moreover, the Clinic Management System facilitates communication and collaboration among clinic staff members. It provides a centralized platform for sharing information, exchanging messages, and coordinating tasks, fostering teamwork and improving efficiency.

Overall, the Clinic Management System offers a range of capabilities and benefits for healthcare facility management, including centralized record-keeping, streamlined appointment scheduling, efficient billing and invoicing, effective inventory management, and enhanced communication and collaboration. By leveraging these features, clinics can optimize their operations, improve patient care, and achieve greater organizational efficiency.

**CUSTOMER REQUIREMENTS SPECIFICATION (CRS)**

**Name of Project**: Clinic Management System

1. **Modeling requirements of customer (Use case Diagram)**

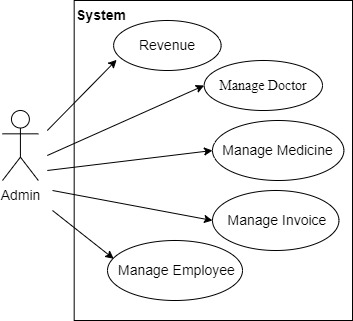


1. **Actor Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Actor** | **Use Case** | **Description** |
|  | Admin | Login | Allow Login in to the system by account |
| Logout | Allow Logout in to the system |
| Manage Medicine | This module allow Admin to manage information of medicine consist of add new medicine , update medicine and delete medicine. |
| Manage Doctor | Allow Admin to manager doctor's information |
| Revenue | Allows Admin to compile medication statistics and medical examination history for online replacement |
| Manage Invoice | This module allow Admin to manage information of invoice consist ofupdate information of prescription. |
|  |  | Manage Employee | Allow Admin to manager employee's information |
|  | Employee | Login | Allow Login in to the system by account |
| Logout | Allow Logout in to the system |
| Manage Payment | This module allows Employee to manage payment history and payment status |
| Manage Appointment | This module allows Employee to manage patient appointment scheduling to save time. |
| Manage Patient | This module allows Employee to manage patient information and disease types. |
| Print Invoice | This module allow Employee to print medical prescription and invoice information. |

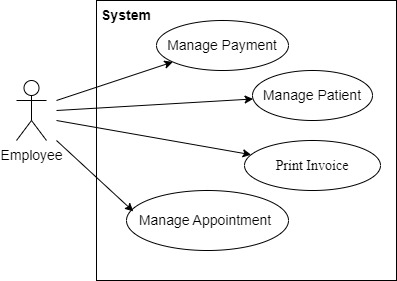
1. **Use case Description**

**\*Admin:** This module allow Admin to manage information of product consist of **Manage Medicine , Manage Doctor ,Mange Employee, Manage Invoice** and **Revenue.**

****

* 1. **Medicine Manager:** This module allow Admin to manage information of product consist of **add new medicine**, **update medicine** and **delete medicine.**
     1. **Add New Medicine**
* Purpose: This function allow Admin to add new information of medicine.
* Inputs:
  + ID of medicine
  + Name of medicine
  + Quantity medicine
  + Category medicine
* Outputs: New information of medicine is added to table of database.
  + 1. **Update Medicine**
* Purpose: this function allow Admin to update information of medicine.
* Inputs:
  + Name of medicine
  + Quantity medicine
  + Category medicine
* Outputs: New information of medicine is updated to table of database.
  + 1. **Delete Medicine**
* Purpose: this function allow Admin to delete information of medicine.
* Inputs: Click “Delete” button.
* Outputs: Information of medicine is deleted in table of database.
  1. **Revenue**
* Purpose: This function allows Admin to compile medication statistics and medical examination history for online replacement.
* Inputs: Total number of patients (day/week/month), number of drugs, type of drugs sold.
* Outputs: Announce results after calculating revenue.
  1. **Manage Doctor:** This module allow Admin to manage information of doctor consist of **add new doctor**, **update doctor** and **delete doctor.**
     1. **Add New Doctor**
* Purpose: This function allow Admin to add new information of doctor.
* Inputs:
  + Doctor ID
  + Doctor Name
  + Phone
  + Age
  + Gender
  + Address
  + Email
* Outputs: New information of doctor is added to table of database.
  + 1. **Update Doctor**
* Purpose: this function allow Admin to update information of doctor.
* Inputs:
  + Doctor Name
  + Phone
  + Age
  + Gender
  + Address
  + Email
* Outputs: New information of doctor is updated to table of database.
  + 1. **Delete Doctor**
* Purpose: this function allow Admin to delete information of doctor.
* Inputs: Click “Delete” button.
* Outputs: Information of doctor is deleted in table of database.
  1. **Manage Employee:**This module allow Admin to manage information of employee consist of **add new employee**, **update employee** and **delete employee.**
     1. **Add New Employee**
* Purpose: This function allow Admin to add new information of employee.
* Inputs:
  + Employee ID
  + EmployeeName
  + Phone
  + Age
  + Gender
  + Address
  + Email
  + Username
  + Password
* Outputs: New information of employee is added to table of database.
  + 1. **Update Employee**
* Purpose: this function allow Admin to update information of employee.
* Inputs:
  + Employee Name
  + Phone
  + Age
  + Gender
  + Address
  + Email
  + Username
  + Password
* Outputs: New information of employee is updated to table of database.
  + 1. **Delete Employee**
* Purpose: this function allow Admin to delete information of employee.
* Inputs: Click “Delete” button.
* Outputs: Information of employee is deleted in table of database.
  1. **Manage Invoice :** This module allow Admin to manage information of invoice consist of **update invoice** and **printing invoice.**
     1. **Update Invoice:**
* Purpose: Allow Admin to update medical invoice to record patient information.
* Inputs: Date of medical examination, quantity of medicine, name of medicine, price.
* Outputs:Medical examination Invoice after update.
  + 1. **Print Invoice:**
* Purpose: Allow Employee to print medical prescription and invoice information.
* Inputs: Date of medical examination,Doctor Name,Patient Name, Name of medicine, Price, Payment Method, Status.
* Outputs: Medical examination prescription and invoice are printed.
  1. **Login**
* Purpose: Log in to your account to manage the apps.
* Inputs: Account's phone number and password.
* Outputs:Notify results after logging in.
  1. **Logout**
* Purpose: Log out to your account to manage the apps.
* Inputs: Log out button.
* Outputs: Notify results after logging out.

**\*Employee:** This module allow Employee to **Manage Patient, Print Invoice, Manage Payment** and **Manage Appointment.**

****

* 1. **Print Invoice** **:** This module allow Employee to manage information of invoice consist of **create new invoice**, **delete patient** and **printing invoice.**
     1. **Create Invoice:**
* Purpose: This function allow Employee to create new information of invoice.
* Inputs:
  + Patient Name
  + Doctor Name
  + Medicine Name
  + Quantity
* Outputs: New information of patient is invoice to table of database.
  + 1. **Delete Invoice:**
* Purpose: this function allow Employee to delete information of invoice.
* Inputs: Click “Delete” button.
* Outputs: Information of invoice is deleted in table of database.
  + 1. **Printing Invoice:**
* Purpose: Allow Employee to print medical prescription and invoice information.
* Inputs: Date of medical examination,Doctor Name,Patient Name, Name of medicine, Price, Payment Method, Status.
* Outputs: Medical examination prescription and invoice are printed to desktop screen.
  1. **Manage Patient :** This module allow Employee to manage information of patient consist of **add new patient**, **update patient** and **delete patient.**
     1. **Add New Patient**
* Purpose: This function allow Employee to add new information of patient .
* Inputs:
  + Patient ID
  + Patient Name
  + Phone
  + Age
  + Gender
  + Address
* Outputs: New information of patient is added to table of database.
  + 1. **Update Patient**
* Purpose: this function allow Employee to update information of patient .
* Inputs:
  + Patient ID
  + Patient Name
  + Phone
  + Age
  + Gender
  + Address
* Outputs: New information of patient is updated to table of database.
  + 1. **Delete Patient**
* Purpose: this function allow Employee to delete information of patient .
* Inputs: Click “Delete” button.
* Outputs: Information of patient is deleted in table of database.
  1. **Manage Appointment :** This module allow Employee to manage information of appointment consist of **add new appointment** , **update appointment** and **delete appointment.**
     1. **Add New Appointment:**
* Purpose: This function allow Employee to add new information of appointment .
* Inputs:
  + Appointment ID
  + Doctor Name
  + Patient Name
  + Date
  + Time
* Outputs: New information of appointment is added to table of database.
  + 1. **Update Appointment:**
* Purpose: this function allow Employee to update information of appointment .
* Inputs:
  + Appointment ID
  + Doctor Name
  + Patient Name
  + Date
  + Time
* Outputs: New information of appointment is updated to table of database.
  + 1. **Delete Appointment:**
* Purpose: this function allow Employee to delete information of appointment .
* Inputs: Click “Delete” button.
* Outputs: Information of appointment is deleted in table of database.
  1. **Manage Payment :**
* Purpose: Allow Employee to manage payment history and payment status.
* Inputs: Name,Age, Gender,Address,Phone of Patient.
* Outputs: Display payment confirmation.
  1. **Login**
* Purpose: Log in to your account to manage the apps.
* Inputs: Account's phone number and password.
* Outputs:Notify results after logging in.
  1. **Logout**
* Purpose: Log out to your account to manage the apps.
* Inputs: Log out button.
* Outputs: Notify results after logging out.

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| **Hardware** | Itel corei7 12500X, Ram 16GB, Rom256GB, Window11 Pro | Itel core i3 2400H, Ram 4GB, Rom 128GB, Window XP |
| **Software** | JDK19 | JDK19 |
| **Additional**  **Technology** |  | |

**CUSTOMER ACCEPTANCE CRITERIA**

|  |  |
| --- | --- |
| **No.** | **CUSTOMER ACCEPTANCE CRITERIA** |
|  | User-Friendly Interface |
|  | Patient Records Management |
|  | Appointment Scheduling |
|  | Secure Data Handling |
|  | Billing and Invoicing |
|  | Electronic Medical Records |
|  | Employee Management |
|  | Doctor Management |
|  | **Inventory and Medical Supplies Management**: |
|  | Reporting and Analytics |
|  | Treatment Process Management |
|  | System Integration |
|  | Customer Support |
|  | **Insurance Records Management**: |
|  | Patient Feedback Management |

**PROJECT PLAN**

1. **Project Details**
2. **Name of the Client:** Clinic Management System

**2. Date of the Project Plan:** 4/22/2024

**3. Project Vision/Objectives:** Clinic Management System, currently difficulties such as revenue management, turnover, number of patients entering and exiting the warehouse have been resolved.

**4. Scope:** The scope of the project includes the entire medicine warehouse, medicine dispensing staff and managers. Positions such as medicine dispenser, patient invoice information printer, medical examination manager, medicine warehouse manager, and administrator will be improved to operate more smoothly.

**5. Our understanding of the client organization:** The warehouse management system includes the functions of adding, editing, deleting drugs entered into the warehouse and storing all drug information as well as patient and data necessary for retrieval and use. Sales staff are responsible for recording patients and drugs sold, printing invoices for customers, being able to search for drugs in stock, and providing patient information to managers; The clinic manager is responsible for managing sales staff, managing revenue, checking invoices; The system administrator is responsible for managing records, managing drugs and prescriptions, and managing suppliers. , general report.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Member** | **Roles** | **Responsibilities** |
| **1** | Thach Chanh Mo Ni | Leader,Code,Designer | - Project leader, ensuring project progress and quality. - Develop source code and design interface for the application. |
| **2** | Nguyen Hoang Khang | Code,Designer | - Develop source code and design interface for applications. |
| **3** | Do Thanh Trong | Code,Tester,Check Doc | - Develop source code for applications.  - Perform functional testing and confirm the correctness of documents.  - Check and ensure the accuracy of project documents. |

1. **Project Initiation/Requirement Documents**

The key objectives of the Clinic Management System (CMS) are:

Efficient Data Management: Enable the clinic to manage large volumes of patient data accurately, securely, and with ease.

Expanded Service Offerings: Support the clinic's expansion plans by providing modules for new services like preventive care and specialized treatments.

Streamlined Administrative Functions: Automate tasks such as appointment scheduling, billing, inventory management, and report generation to reduce manual effort and errors.

Enhanced Patient Experience: Provide a user-friendly interface to enhance the patient experience from appointment booking to post-treatment follow-up.

Scalability and Flexibility: Design the system to accommodate future growth and changes in clinic operations, ensuring scalability and adaptability.

The development of the Clinic Management System (CMS) will involve collaboration between the clinic's management team, healthcare professionals, and IT experts. Together, they will create a robust solution that meets current needs while laying the groundwork for future innovation in healthcare delivery.

1. **Deliverable**

**Products delivered to customers include:**

-A word document file describing the project, working process

-A document printed from a word file

-A system source code file

-A user manual file

-A file of installation instructions

-A database code file (SQL Server)

1. **Project Dependencies**

e success of the project will depend on customer satisfaction, product quality, warehouse management ability, flexibility in operating processes.

1. **Major Project Milestones**

The successful implementation of the **Clinic Management System** will be achieved through the accomplishment of several key milestones. The first milestone is the **Project **Initiation and Planning****, which includes defining the project scope, identifying key stakeholders, and developing a comprehensive project plan. This is followed by the ****Requirements Gathering** and **Analysis**** phase, where detailed consultations with clinic staff and stakeholders are conducted to understand and document the system requirements. The third milestone is the ****System Design****, which involves creating architectural designs, database schema, and user interface prototypes. **The **Development** and **Integration**** phase is next, where the system is coded, integrated with existing systems, and initial testing is performed. Once the system is developed, the ****User Acceptance Testing (UAT)**** milestone ensures the system meets all specified requirements and functions correctly in the real-world environment. The penultimate milestone is the ****Deployment** and **Training****, where the system is rolled out to the clinic, and users are trained to use the new system effectively. Finally, the ****Project Closure** and **Post-Implementation Review**** milestone ensures that the project is formally closed, and a review is conducted to assess project success and gather lessons learned for future projects.

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

* **Review Meeting Participants**: Include project managers, QA team members, key stakeholders, and development team representatives.
* **Frequency**: Conduct review meetings bi-weekly to assess progress, address issues, and plan next steps.
* **Documentation**: Maintain detailed minutes of each review meeting, including action items and decisions made.
* **Review Scope**: Cover requirements review, design review, code review, and test plan review to ensure all aspects meet quality standards.
* **Feedback Mechanism**: Implement a structured feedback mechanism to capture inputs from all participants and incorporate them into project improvements.

1. Testing Activities (Final Test)

* **Unit Testing**: Conduct unit tests to validate individual components and ensure they function correctly in isolation.
* **Integration Testing**: Perform integration tests to verify that different modules and components work together seamlessly.
* **System Testing**: Execute system tests to ensure the complete and integrated system meets the specified requirements.
* **User Acceptance Testing (UAT)**: Engage end-users in UAT to validate the system in a real-world environment and ensure it meets their needs.
* **Regression Testing**: Regularly perform regression tests to ensure that new changes do not negatively impact existing functionality.
* **Performance Testing**: Conduct performance tests to assess the system’s responsiveness, stability, and scalability under load.
* **Final Test**: Before deployment, execute a comprehensive final test to verify the system’s overall quality and readiness for release.

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

· **Regular Backups**: Schedule automatic daily backups of all critical data to secure, off-site locations.

· **Data Redundancy**: Implement data redundancy by storing copies of data across multiple physical locations to mitigate the risk of data loss.

· **Backup Verification**: Regularly verify backups to ensure data integrity and reliability.

· **Disaster Recovery Plan**: Develop and maintain a detailed disaster recovery plan outlining steps to restore operations in the event of a system failure.

· **Recovery Testing**: Periodically test recovery procedures to ensure they are effective and that staff are familiar with the process.

· **Network Redundancy**: Establish network redundancy to prevent disruptions caused by network failures, including backup network connections.

· **Failover Mechanisms**: Implement failover mechanisms to automatically switch to backup systems in case of primary system failures.

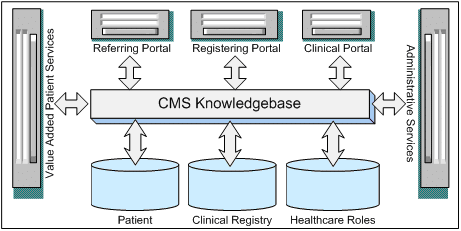
· **Incident Response Team**: Form an incident response team equipped to handle emergencies and execute the recovery plan efficiently.

· **Documentation**: Maintain comprehensive documentation of all backup and recovery processes, including contact information for key personnel and vendors.

· **User Training**: Train users and IT staff on backup procedures and recovery processes to ensure quick and effective response to issues.

**ARCHITECTURE & GUI STANDARD OF APPLICATION**

1. **Architecture of Application**



1. **GUI Standard Application**

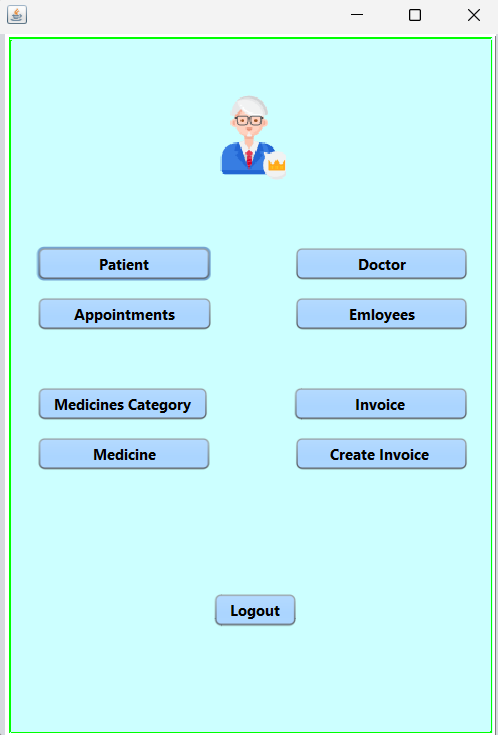
|  |  |
| --- | --- |
| **Property** | **Value** |
| Document Theme and Color Scheme | RGB(51,51,5151) |
| Form - Background Color | RGB(21,21,2121) |
| Title - Font Size | 24px |
| Title - Font Color | #ffffff |
| Title - Font Style | Time New Roman |
| Title - Alignment | Center |
| Background color of Controls on the form | #C7F2A4 |
| Foreground color of Controls on the form | #EEEEEE |
| Control Caption - Font Size | 18px |
| Control Caption - Font Color | #ffffff |
| Control Caption - Font Style | Montserrat |
| Control caption and controls - Alignment | Center |
| Command button - Alienment | Center |

**INTERFACE DESIGN DOCUMENT**

1. **Login:** Login to manage application functions

****

1. **Home Page:** Display application functions on the main screen

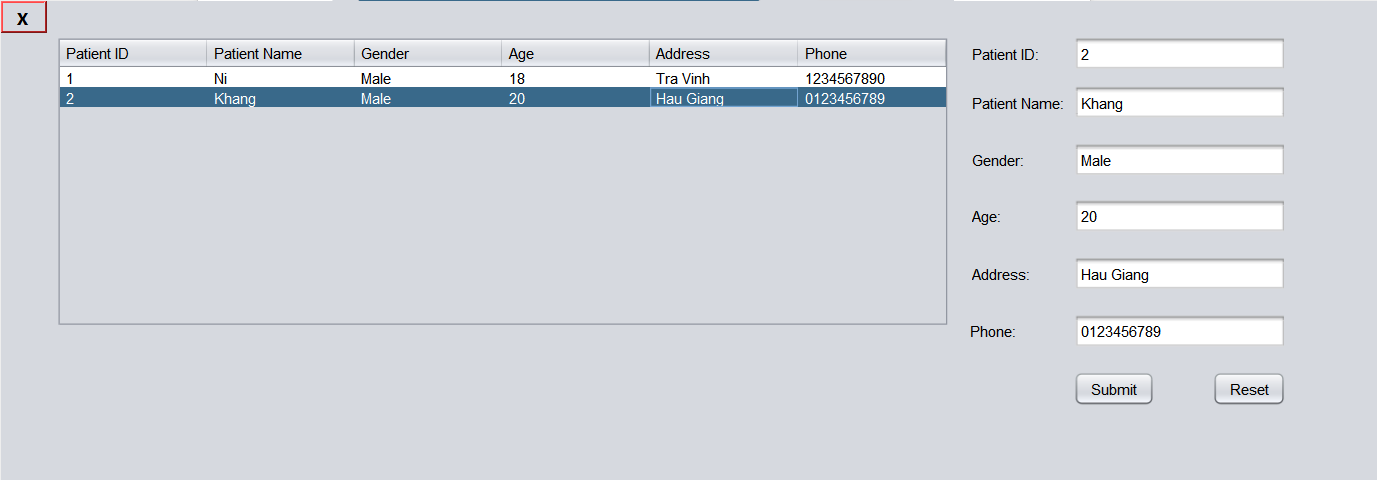


1. **Patient:** Display the form to add new, edit and delete information of Patient

**\*Add new:** This function allow Admin to add new information of patient



**\*Edit:**This function allow Admin to edit information of patient

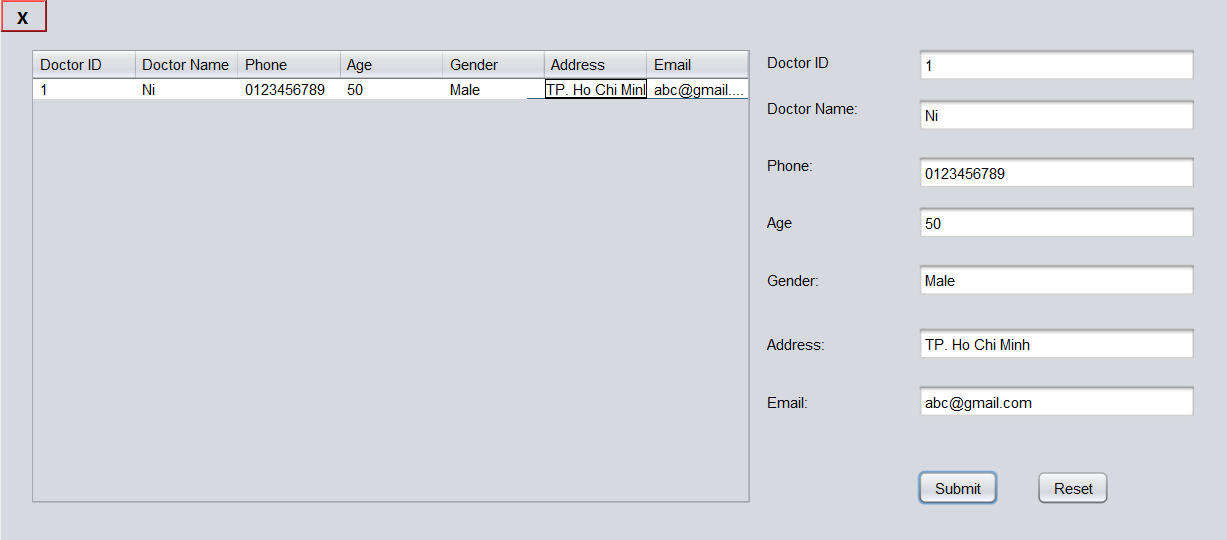


**\*Delete:**This function allow Admin to delete information of patient

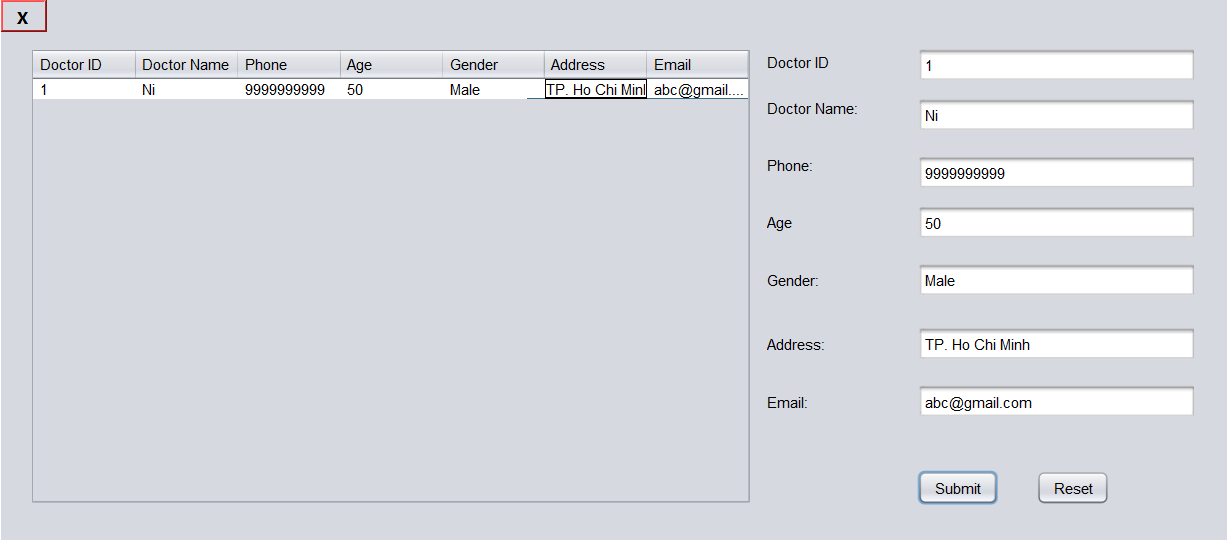


1. **Doctor:** Display the form to add new, edit and delete information of Doctor

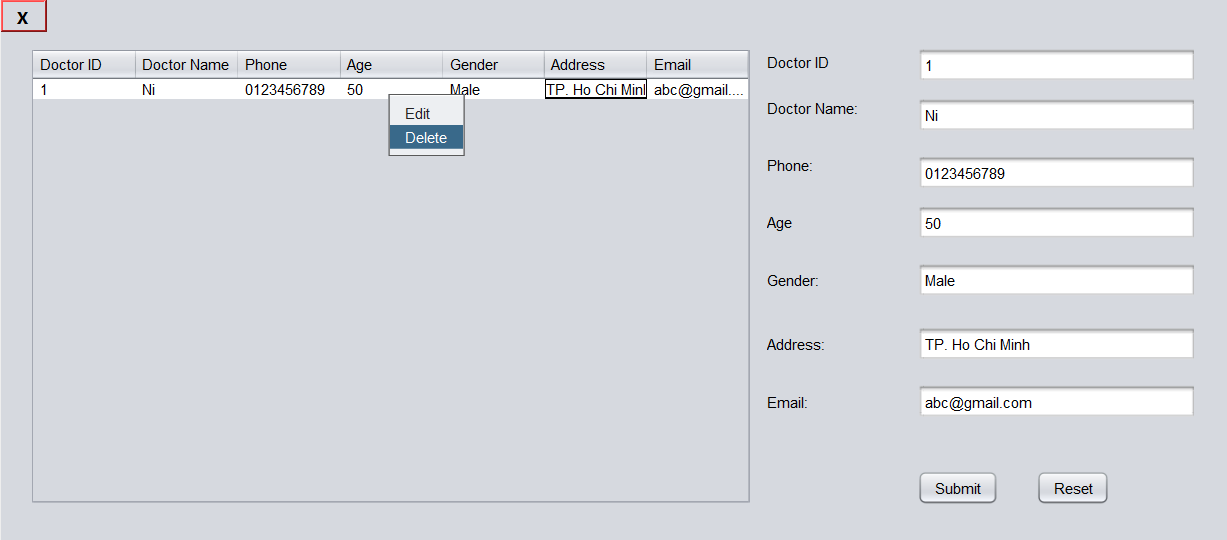
**\*Add new:**This function allow Admin to add new information of doctor



**\*Edit:** This function allow Admin to edit information of doctor



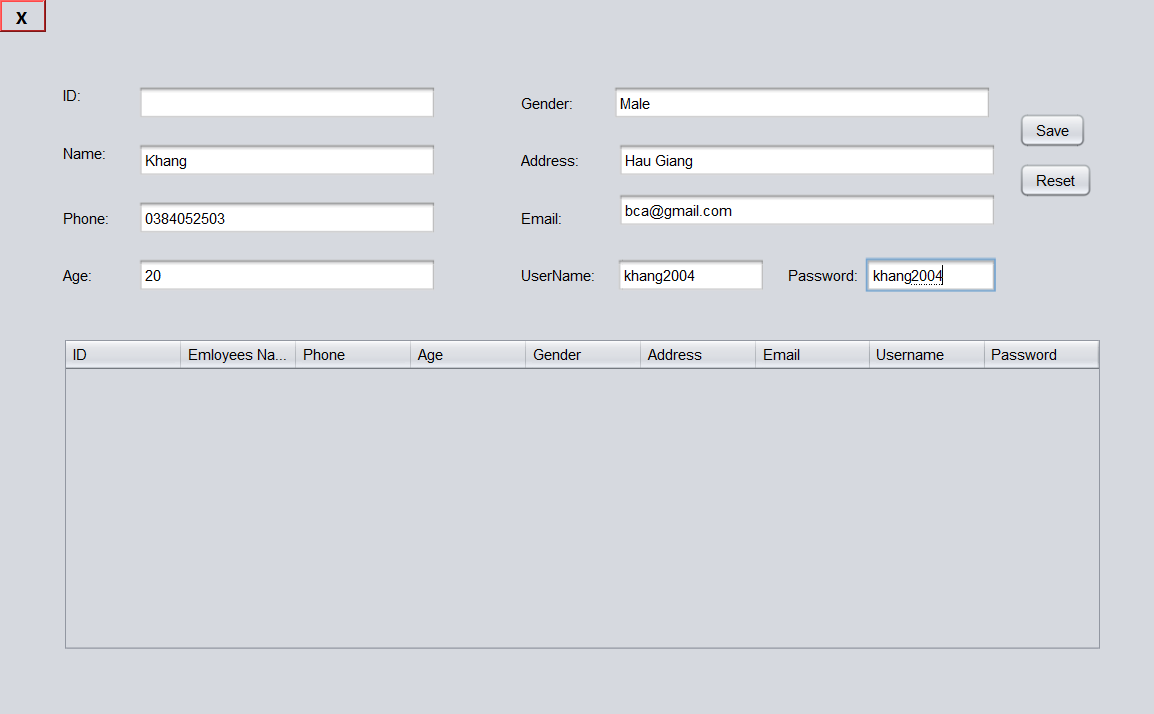
**\*Delete:**This function allow Admin to delete information of doctor

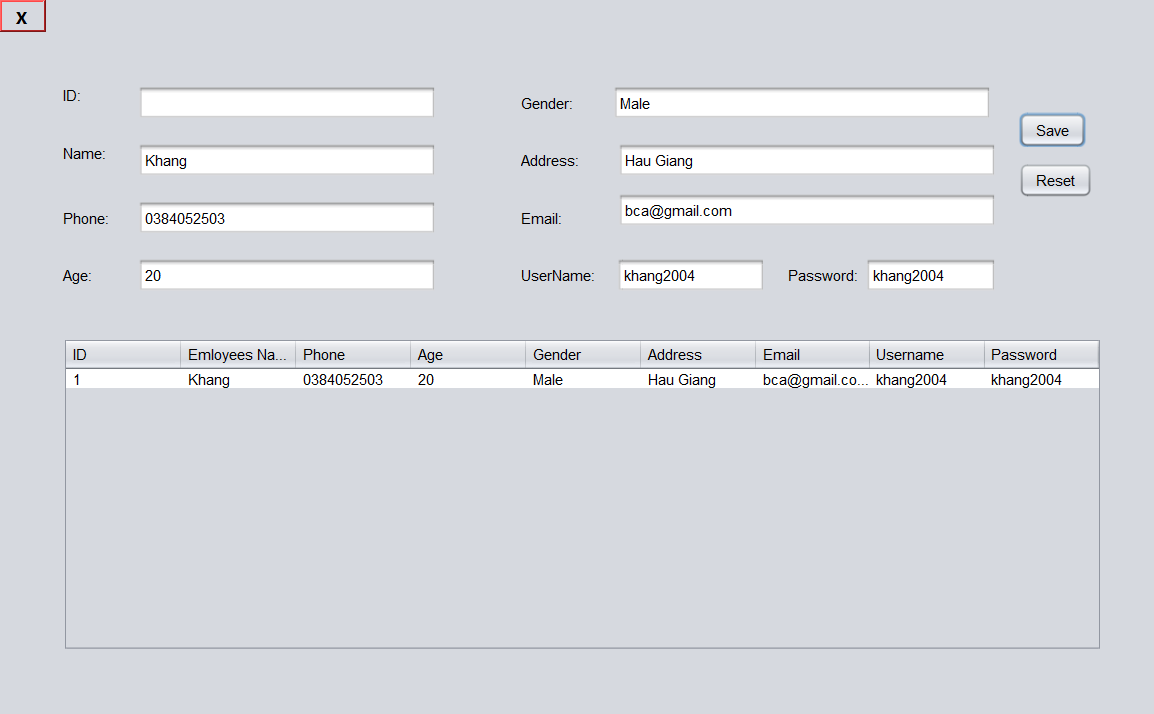


1. **Employee:** Display the form to add new, edit and delete information of Employee.

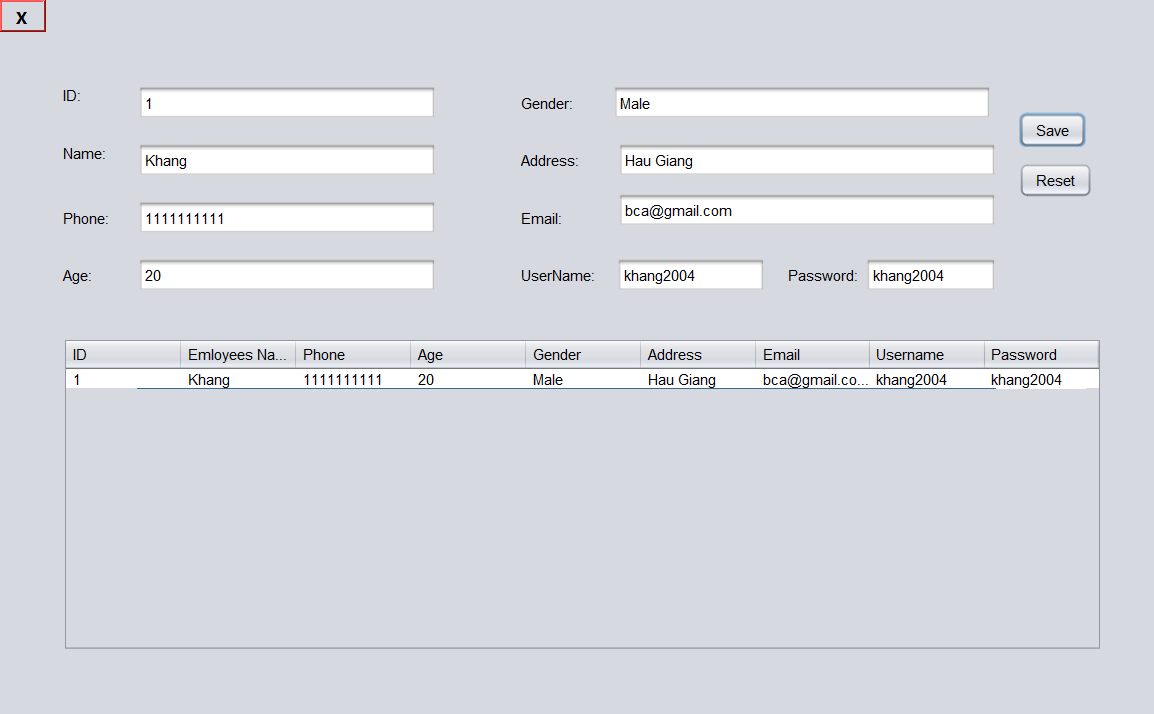
Register username and password to Employee

**\*Add new:** This function allow Admin to add new information of employee

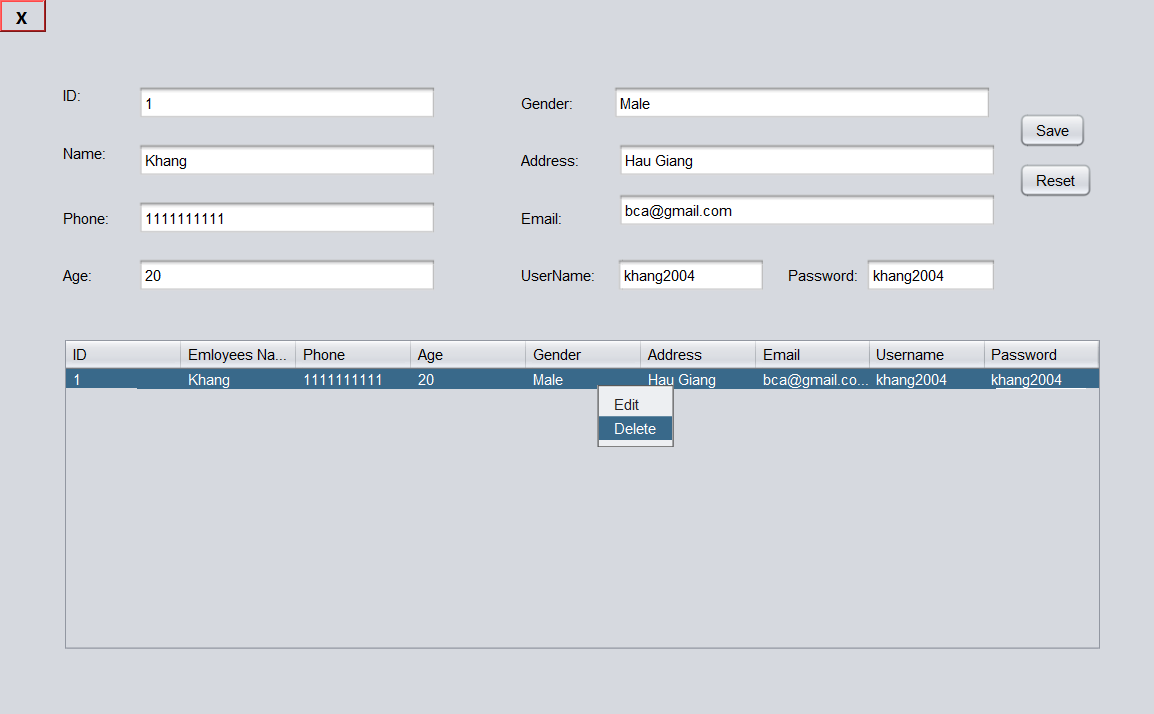




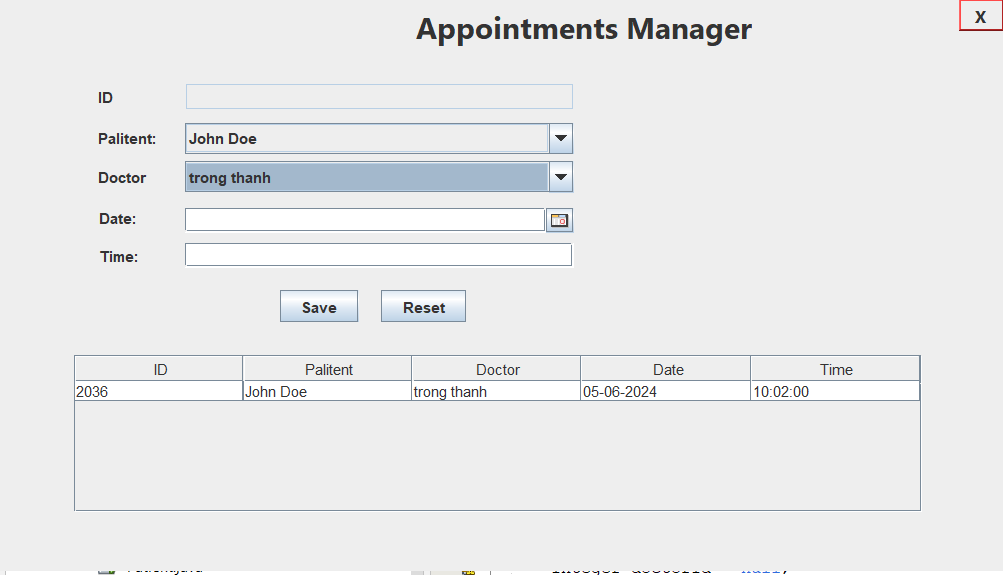
**\*Edit:** This function allow Admin to edit information of employee



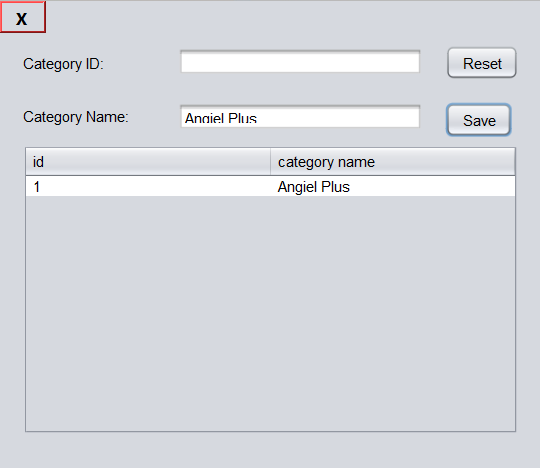
**\*Delete:** This function allow Admin to delete information of employee



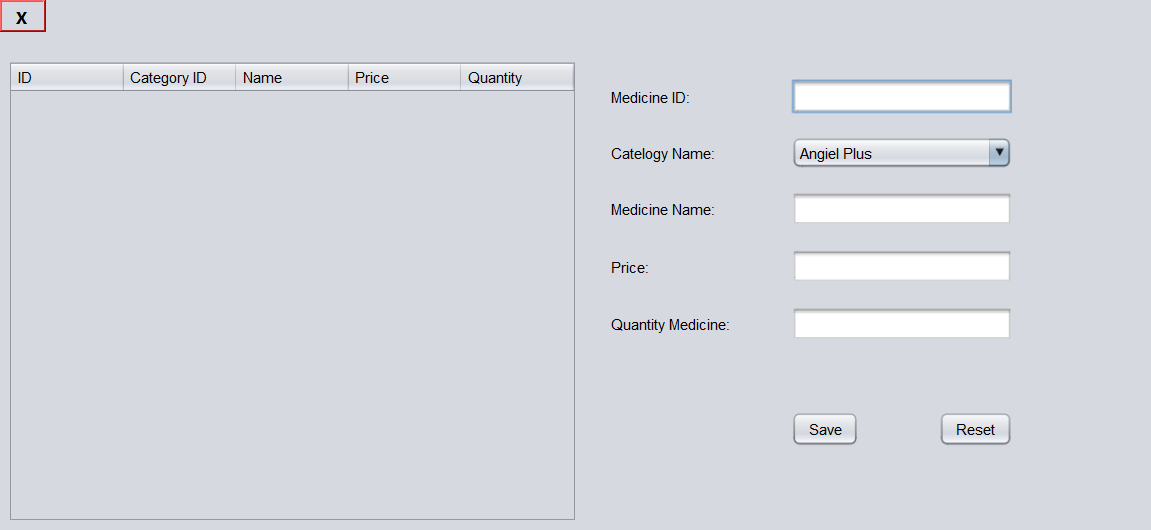
1. **Appoinment:**This function allow Admin to add new and delete information of appoinment.



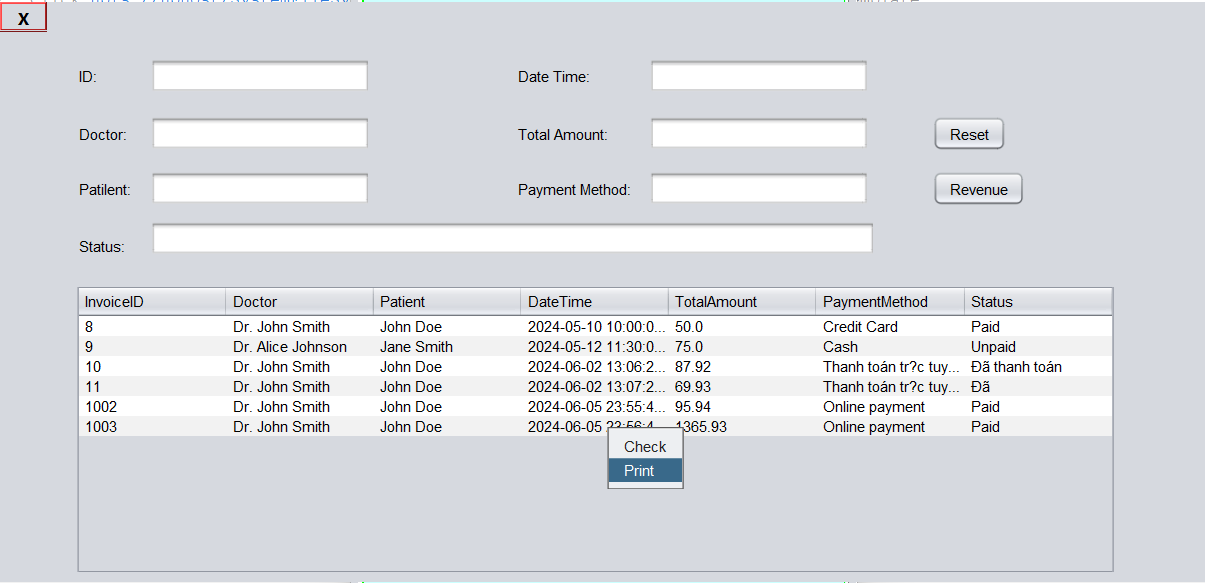
1. **Medicine Category:** This function allow Admin to manage medicine category.



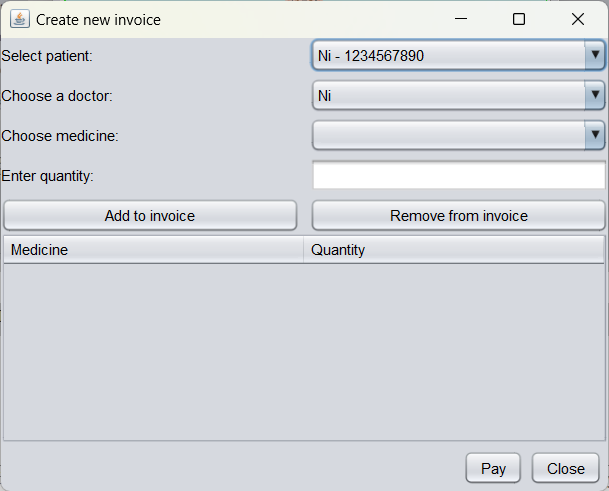
1. **Medicine :** Display the form to add new, edit and delete information of Medicine.



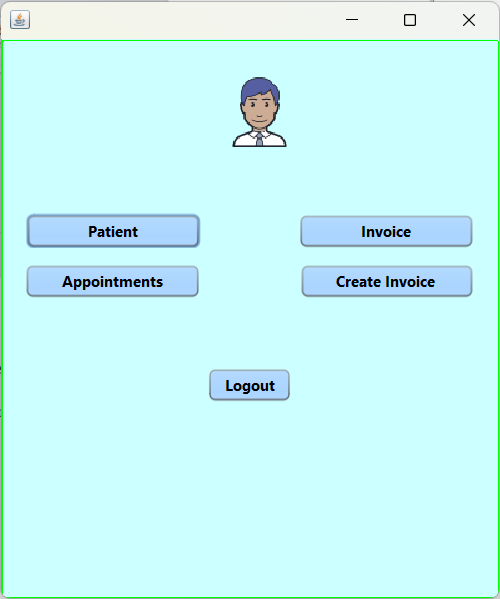
1. **Print Invoice:** Print the invoice and include the prescription



1. **Create Invoice:** This function allowEmployee invoices for patients to pay

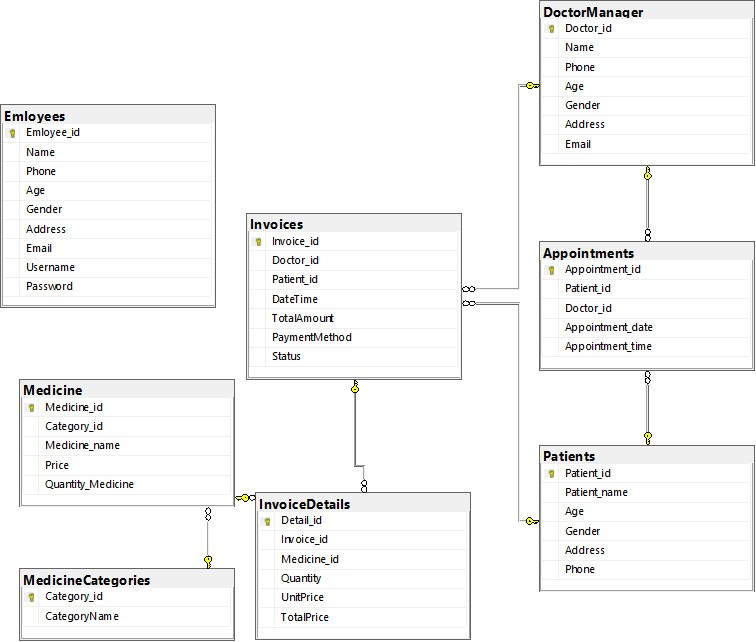


1. **Employee Home Page:** Display funtion Patient, Invoice, Create Invoce, Appointment Create Invoice and This function is the same as the Admin function



**DATABASE DESIGN DOCUMENT**

1. **ER/Class Diagram.**



**II. Table Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Manage Doctor** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Doctor\_ID | Int | Primary Key | Unique doctor ID |
| Name | Varchar(100) | Not null | Doctor's name |
| Phone | Varchar(100) | Not null | Doctor's phone |
| Age | Int | Not null | Doctor's age |
| Gender | Varchar(25) | Not null | Doctor's gender |
| Address | Varchar(255) | Not null | Doctor's address |

|  |  |  |  |
| --- | --- | --- | --- |
| **Manage Patient** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Patient\_ID | Int | Primary Key | Unique patient ID |
| Patient\_name | Varchar(100) | Not null | Patient's name |
| Gender | Varchar(25) | Not null | Patient's gender |
| Address | Varchar(100) | Not null | Patient's address |
| Phone | Varchar(25) | Not null | Patient's phone number |

|  |  |  |  |
| --- | --- | --- | --- |
| **Medicine** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Medicine\_ID | Int | Primary Key | Unique medicine ID |
| Category\_ID | Int | Not null | REFERENCES MedicineCategories(Category\_id) |
| Medicine\_name | Varchar(100) | Not null | Name of the medicine |
| Price | Decimal(10,2) | Not null | Price of the medicine |
| Quantity\_Medicine | Int | Not null | Quantity of the medicine |

|  |  |  |  |
| --- | --- | --- | --- |
| **Appointments** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Appointment \_ID | Int | Primary Key | Unique appointment ID |
| Patient\_ID | Int | Not null | REFERENCES Patients(Patient\_id) |
| Doctor\_ID | Int | Not null | REFERENCES ManageDoctor(Doctor\_id) |
| Appointment\_date | Date | Not null | Date of the appointment |
| Appointment\_time | Time | Not null | Time of the appointment |

|  |  |  |  |
| --- | --- | --- | --- |
| **Manage Invoices** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Invoice\_ID | Int | Primary Key | ID Invoice of Medicine |
| Doctor\_ID | Int | Not null | REFERENCES ManageDoctor(Doctor\_id) |
| Patient\_ID | Int | Not null | REFERENCES Patients(Patient\_id) |
| DateTime | DATEtime | Not null | Date and time |
| TotalAmount | Decimal(10,2) | Not null | Total |
| PaymentMethod | Varchar(100) | Not null | Payment of· methods |
| Status | Text | Not null | Status |

|  |  |  |  |
| --- | --- | --- | --- |
| **InvoiceDetails** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Detail\_ID | Int | Primary Key | ID Detail of Medicine |
| Invoice\_ID | Int | Not null | REFERENCES Invoices(Invoice\_id) |
| Medicine\_ID | Int | Not null | REFERENCES Medicine(Medicine\_id) |
| Quantity | Int | Not null | Quantity of medicine |
| UnitPrice | Decimal(10,2) | Not null | Unit of price |
| TotalPrice | Decimal(10,2) | Not null | Total of price |

|  |  |  |  |
| --- | --- | --- | --- |
| **MedicineCategories** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Category\_ID | Int | Primary Key | Unique category ID |
| CategoryName | Varchar(100) | Not null | Name of the medicine category |

|  |  |  |  |
| --- | --- | --- | --- |
| **Manage Employee** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Employee\_ID | Int | Primary Key | Unique doctor ID |
| Employee Name | Varchar(100) | Not null | Employee's name |
| Phone | Varchar(100) | Not null | Employee's phone |
| Email | Varchar(100) | Not null | Employee's phone |
| Age | Int | Not null | Employee's age |
| Gender | Varchar(25) | Not null | Employee's gender |
| Address | Varchar(255) | Not null | Employee's address |
| Username | Varchar(100) | Not null | Employee's username |
| Password | Varchar(100) | Not null | Employee's password |

**PROCESS DESIGN DOCUMENT**

1. **Login and Logout:**

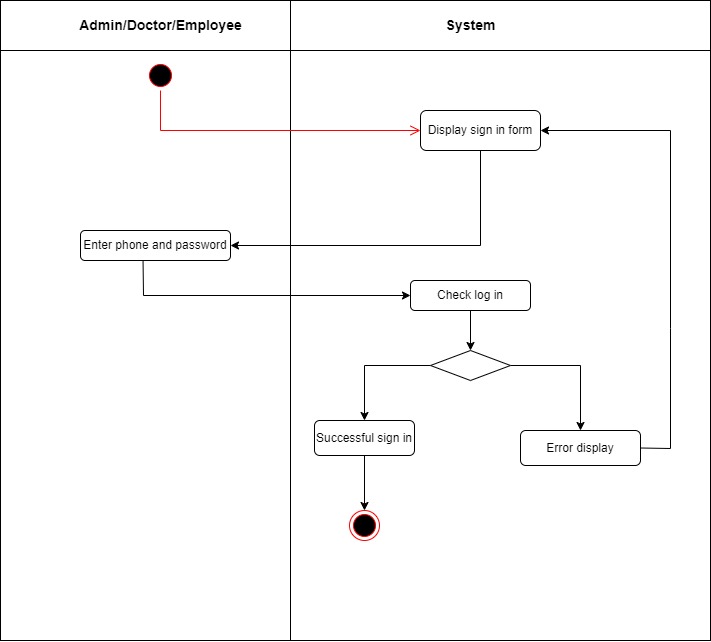


Figure 1.1 - Activity diagram login a connection.

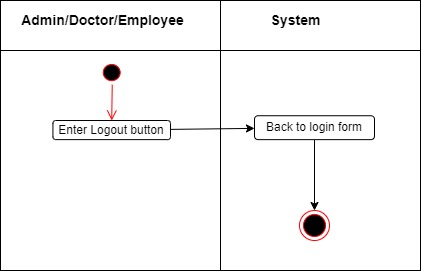


Figure 1.2 - Activity diagram logout a connection.

1. **Admin:**

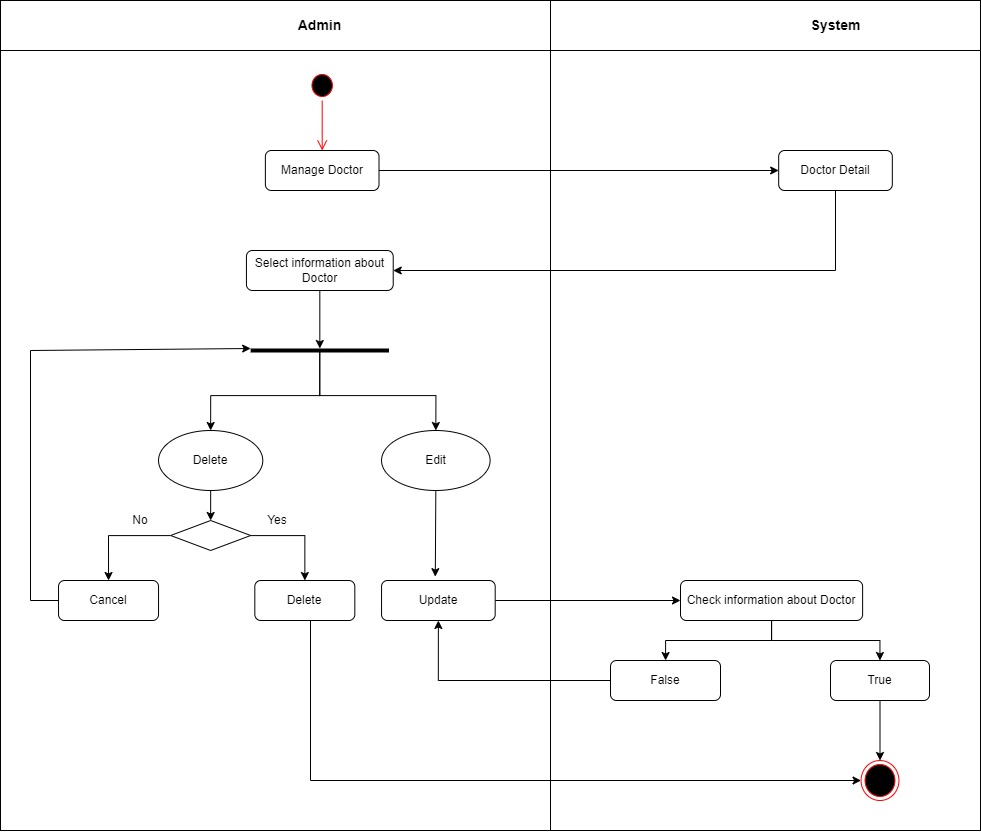
****

Figure 2.1 - Activity diagram Manage Doctor a connection.

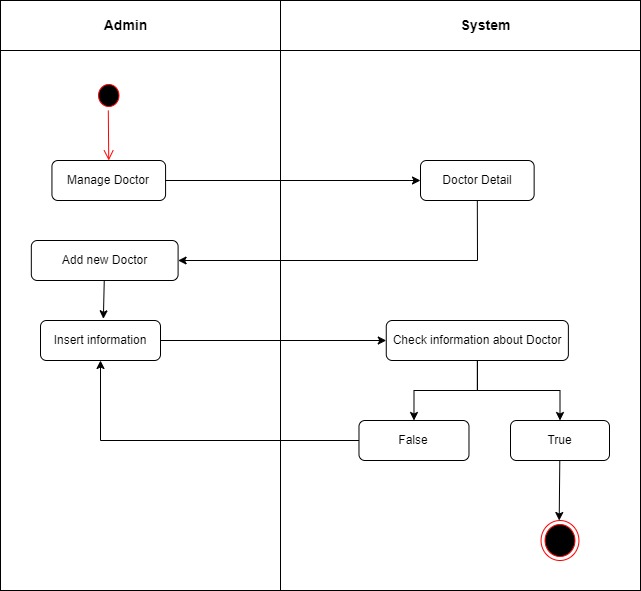


Figure 2.2 - Activity diagram Add new Doctor a connection.

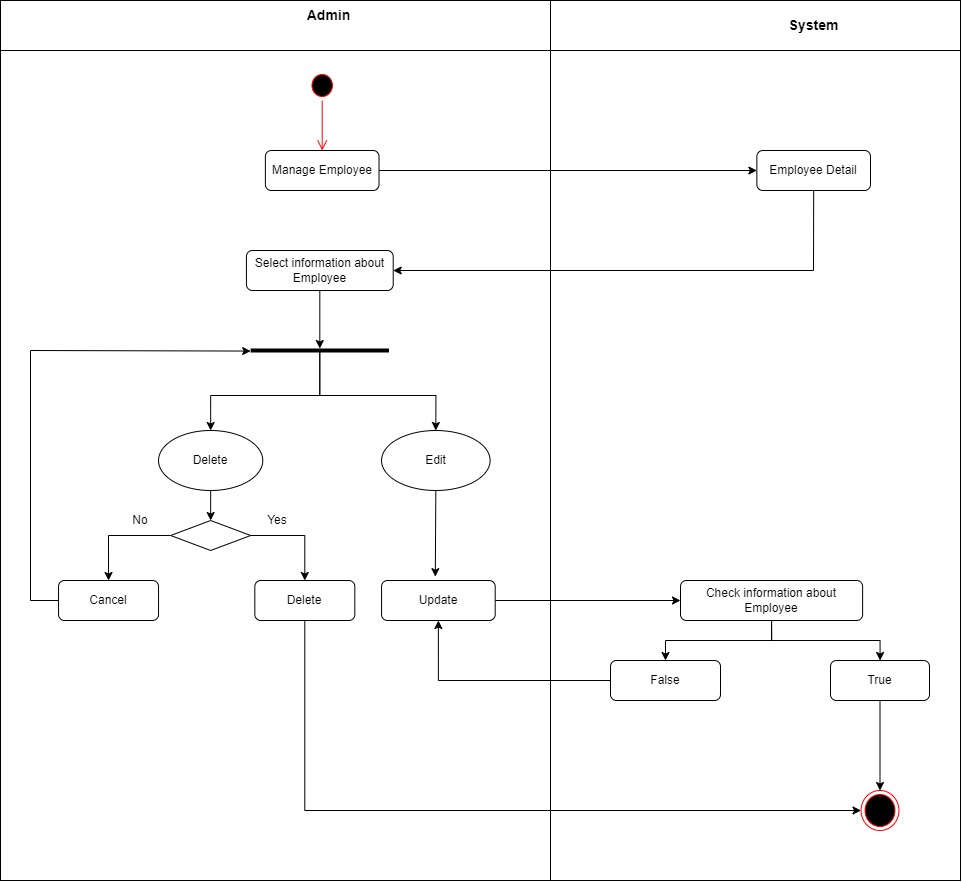


Figure 2.3 - Activity diagram Manage Employee a connection.

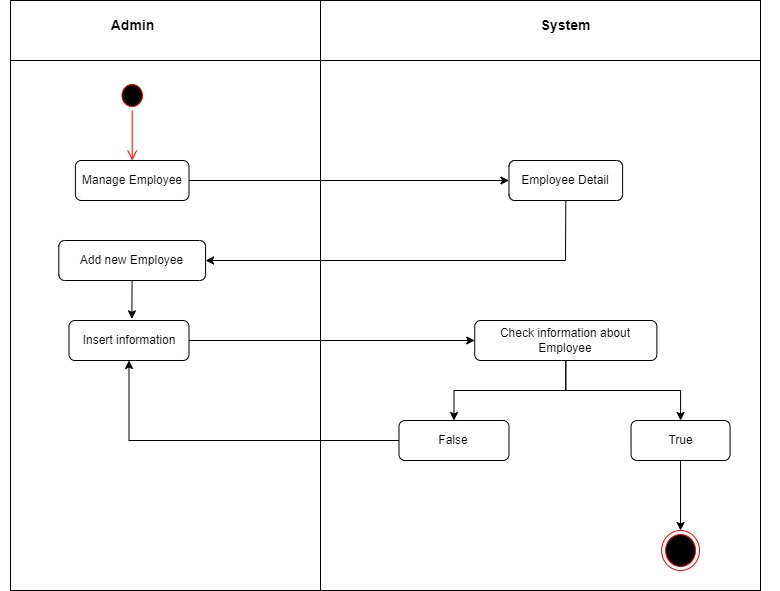


Figure 2.4 - Activity diagram Add new Employee a connection.

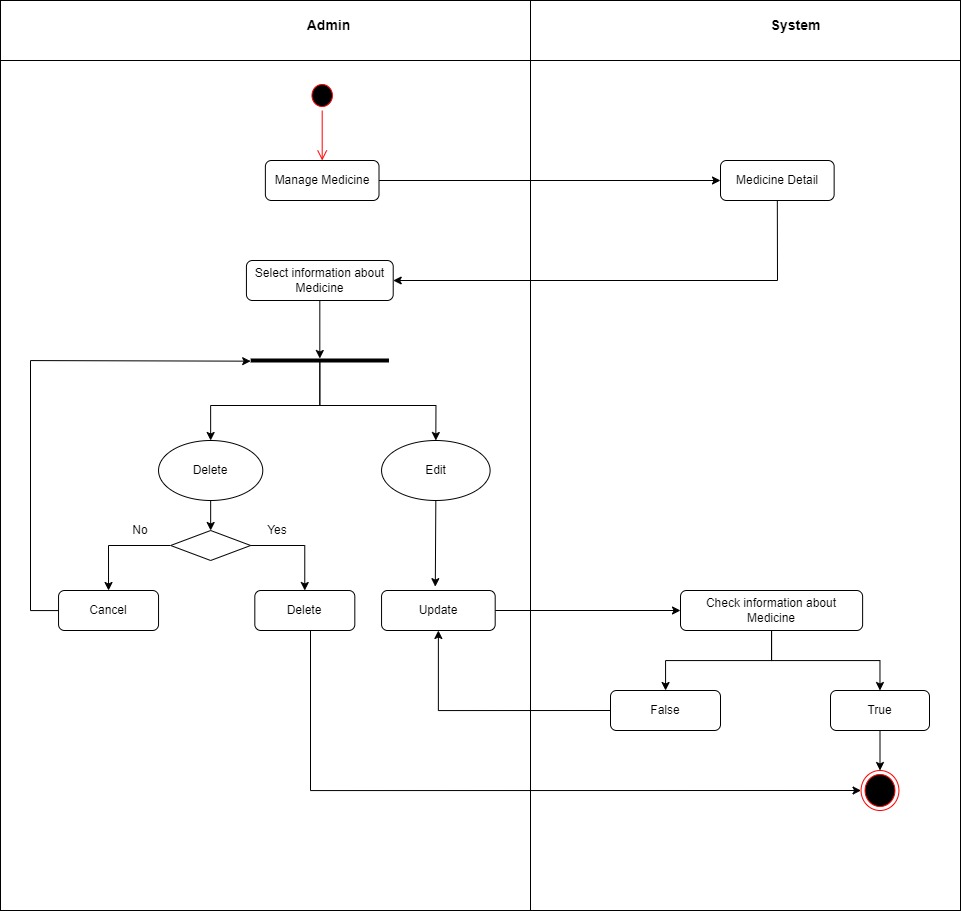


Figure 2.5 - Activity diagram Manage Medicine a connection.

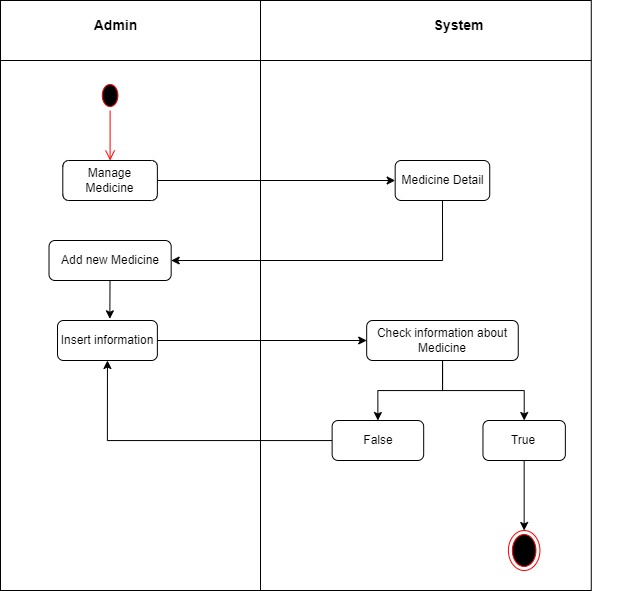


Figure 2.6 - Activity diagram Add new Medicine a connection.

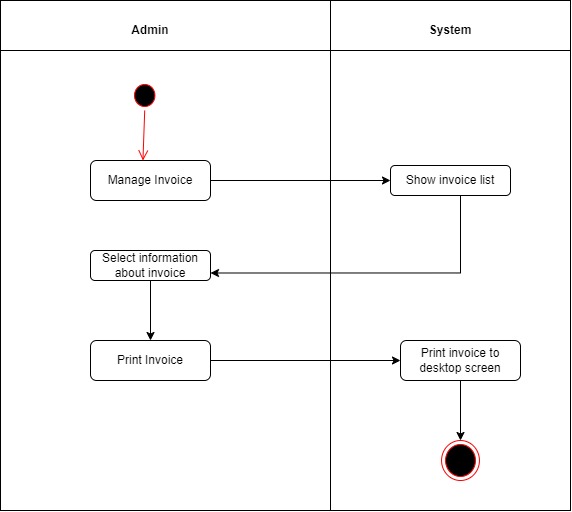


Figure 2.7 - Activity diagram Manage Invoice a connection.

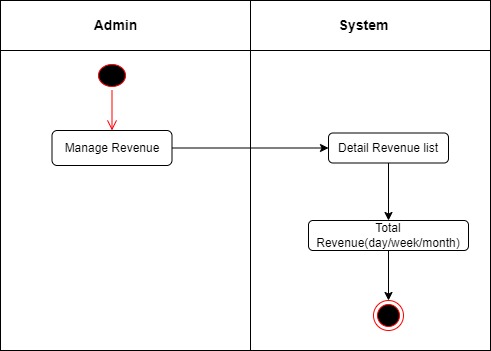


Figure 2.8 - Activity diagram Revenue a connection.

1. **Employee:**

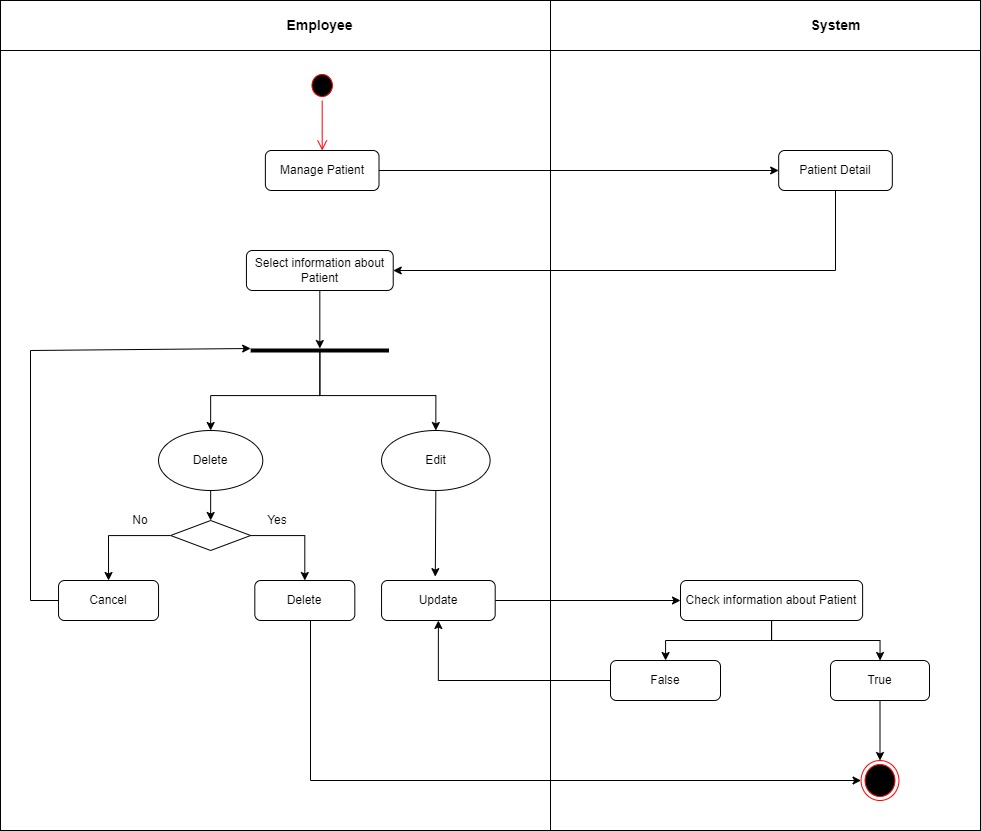


Figure 3.1 - Activity diagram Manage Patient a connection.

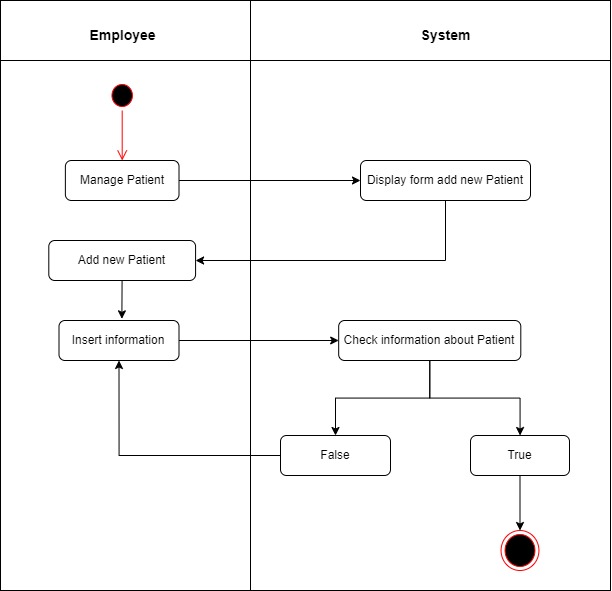


Figure 3.2 - Activity diagram add new Patient a connection.

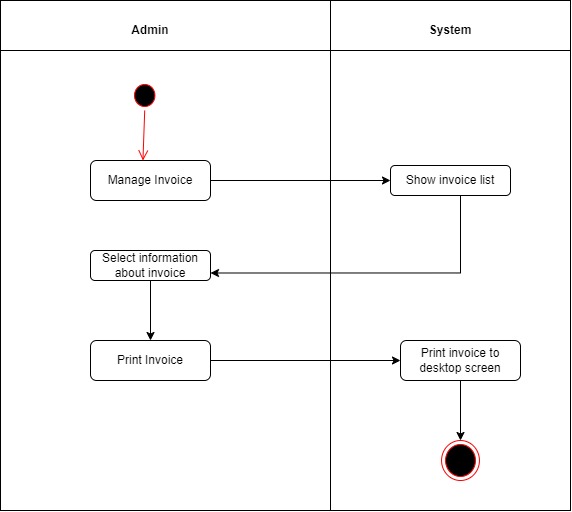


Figure 3.3 - Activity diagram Print Invoice a connection.

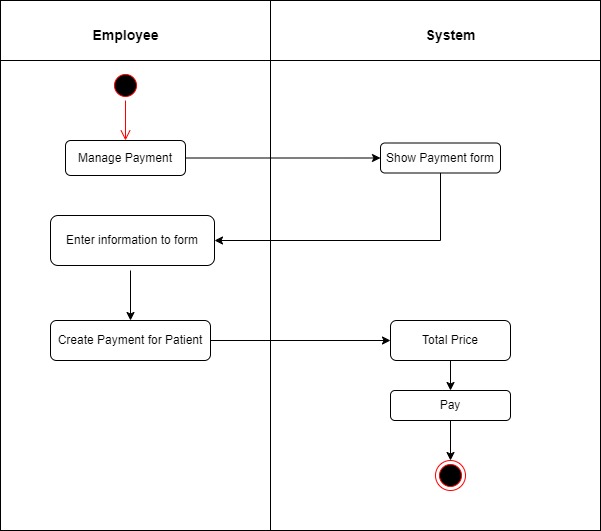


Figure 3.4 - Activity diagram Manage Payment a connection.

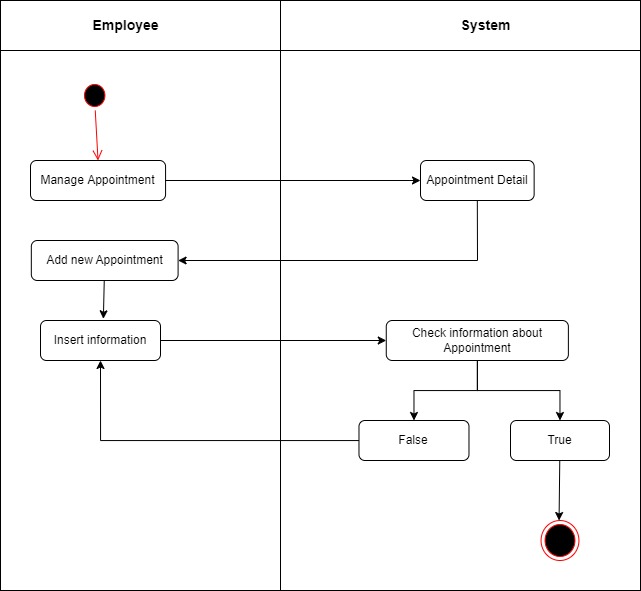


Figure 3.5 - Activity diagram add new Appointment a connection.

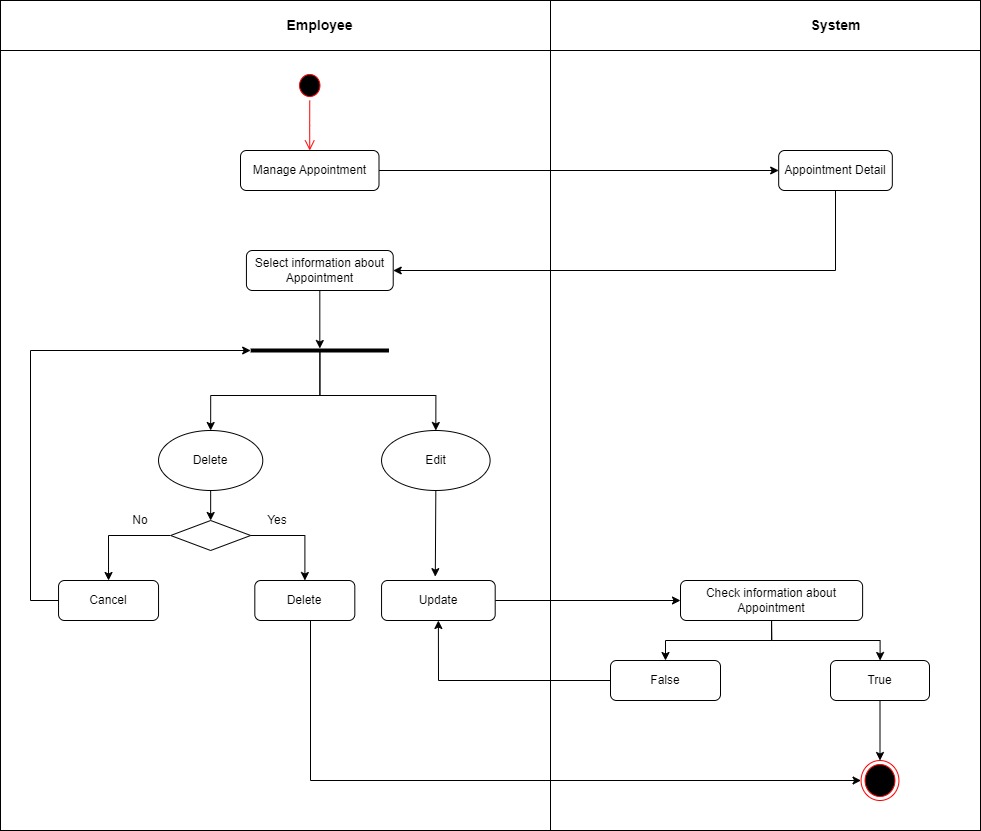


Figure 3.6 - Activity diagram Manage Appointment a connection.

**CODING STANDARDS DOCUMENT**

1. **Programming Standards**

The purpose of a coding standards document is to make sure that all code is designed, written and laid out the same way to make it easier for a developer to switch from one person’s work to another without the needed change of mentality to read someone else’s style

It’s all about uniformity, and nothing about “Right and wrong”

1. **Standards for code writing style**

Code should be written in a consistent style throughout the project

Use descriptive variable names

Use comments to explain complex code

Use whitespace to make code more readable

1. **Standards for declaring Variables**

Declare variables at the beginning of the function or block

Use meaningful variable names

1. **Standards for Function Declarations**

Functions should be declared before they are used

Functions should have a clear purpose and should be named accordingly

1. **Other Standards**

**TASK SHEET**

**1. Project Title:** Clinic Management System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Task** | **Actual Start Date** | **Actual Days** | **Teammate Name** | **Status** |
|  | Document: 1 | 28/4/2024 | 1 day | Thach Chanh Mo Ni | Complete |
|  | Document: 2 | 30/4/2024 | 5 days | Thach Chanh Mo Ni | Complete |
|  | Design Database | 30/4/2024 | 3 days | Do Thanh Trong | Complete |
|  | Design Layout UI | 01/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Document: 3 | 01/5/2024 | 1 day | Nguyen Hoang Khang | Complete |
|  | Use Case Diagram | 03/5/2024 | 2 days | Thach Chanh Mo Ni | Complete |
|  | Document: 4 | 05/5/2024 | 2 days | Nguyen Hoang Khang | Complete |
|  | Use Case Diagram Remake | 06/5/2024 | 1 day | Thach Chanh Mo Ni,Do Thanh Trong | Complete |
|  | Document: 5 | 07/5/2024 | 3 days | Thach Chanh Mo Ni | Complete |
|  | Design Database Remake | 08/5/2024 | 1 day | Do Thanh Trong, Nguyen Hoang Khang | Complete |
|  | Class Diagram | 09/5/2024 | 2 days | Thach Chanh Mo Ni | Complete |
|  | Code Home Page | 10/5/2024 | 3 days | Nguyen Hoang Khang | Complete |
|  | Code css Home Page | 10/5/2024 | 3 days | Nguyen Hoang Khang | Complete |
|  | Code Employee Page | 12/5/2024 | 3 days | Nguyen Hoang Khang | Complete |
|  | Code css Employee Page | 12/5/2024 | 3 days | Nguyen Hoang Khang | Complete |
|  | Code Log in | 13/5/2024 | 2 days | Nguyen Hoang Khang | Complete |
|  | Code Log out | 13/5/2024 | 2 days | Nguyen Hoang Khang | Complete |
|  | Document: 6 | 13/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Code Manage Doctor | 13/5/2024 | 3 days | Thach Chanh Mo Ni | Complete |
|  | Code add new Manage Doctor | 14/5/2024 | 3 days | Thach Chanh Mo Ni | Complete |
|  | Code edit Manage Doctor | 15/5/2024 | 2 days | Thach Chanh Mo Ni | Complete |
|  | Code delete Manage Doctor | 15/5/2024 | 2 days | Thach Chanh Mo Ni | Complete |
|  | Activity Diagram | 16/5/2024 | 2 days | Thạch Chanh Mo Ni,Do Thanh Trong, Nguyen Hoang Khang | Complete |
|  | Document: 7 | 17/5/2024 | 10 days | Thạch Chanh Mo Ni,Do Thanh Trong, Nguyen Hoang Khang | Complete |
|  | Document: 8 | 17/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Code add new Manage Employee | 18/5/2024 | 3 days | Thach Chanh Mo Ni | Complete |
|  | Code edit Manage Employee | 18/5/2024 | 2 days | Thach Chanh Mo Ni | Complete |
|  | Code delete Manage Employee | 18/5/2024 | 2 days | Thach Chanh Mo Ni | Complete |
|  | Code Create Invoice | 18/5/2024 | 3 days | Do Thanh Trong | Complete |
|  | Code Print Invoice | 19/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Code css Invoice | 20/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Document: 9 | 20/5/2024 | 2 days | Thach Chanh Mo Ni | Complete |
|  | Code add new Manage Appoinernt | 20/5/2024 | 3 days | Do Thanh Trong | Complete |
|  | Code edit Manage Appoinment | 20/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Code delete Manage Appoinment | 20/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Code css Appoinment | 21/5/2024 | 1 day | Do Thanh Trong | Complete |
|  | Document: 10 | 22/5/2024 | 3 days | Do Thanh Trong | Complete |
|  | Code Revenue | 23/5/2024 | 3 days | Thach Chanh Mo Ni | Complete |
|  | Code css Revenue | 23/5/2024 | 3 days | Thach Chanh Mo Ni | Complete |
|  | Document: 11 | 24/5/2024 | 2 days | Nguyen Hoang Khang | Complete |
|  | Check code | 25/5/2024 | 1 day | Do Thanh Trong | Complete |
|  | Document: 12 | 26/5/2024 | 2 days | Do Thanh Trong | Complete |
|  | Check Document | 26/5/2024 | 1 day | Thach Chanh Mo Ni | Complete |
|  | Test full app the first | 28/5/2024 | 1 day | Thach Chanh Mo Ni, Do Thanh Trong, Ngutyen Hoang Khang | Complete |
|  | Use guide | 28/5/2024 | 1 day | Nguyen Hoang Khang | Complete |
|  | Installation | 29/5/2024 | 1 day | Do Thanh Trong | Complete |
|  | Test full app the second | 31/5/2024 | 1 day | Do Thanh Trong,Thach Chanh Mo Ni | Complete |

**TESTING DOCUMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Name** | **Task** | **Result** | |
| **Expected** | **Actual Result**  Pass/Fail-Comments |
|  | Add new employee | Enter employee details:  Employee ID: 1  Email: abc@gmail.com Phone:0123456789  Address:Tra Vinh FIRST Name: nguyen  Second name: khang | Record is added to the database | Pass |
| - Second scenario:  Same data except for the following part: Phone: abc,Email: abc | System gives a warning that the Phone,Email format not correct. | Fail - System crash - Warning |
|  | Add New Medicine | - First scenario:  Type: Medicine  Medicine ID: 1 Name Medicine: cough drops  Quantity: 100 Category: ear, nose and throat | Record is added to the database | Pass |
| - Second scenario:  Same data except for the following part: Quantity:  -100 | System gives a warning that the Quantity format not correct. | Fail - System crash - Warning |
|  | Add new Doctor | Enter Doctor details:  Doctor ID: 1 Email: abc@gmail.com Phone:0123456789  Address:Tra Vinh  Phone:123456789 Name: nguyen khang | Record is added to the database | Pass |
| - Second scenario:  Same data except for the following part: Phone: abc,Email: abc | System gives a warning that the Phone,Email format not correct. | Fail - System crash - Warning |
|  | Edit Medicine | Enter data Name Medicine: cough drops  Quantity: 100 Category: ear, nose and throat | Record is added to the database,Show notification success | Pass |
|  | Edit Employe | Enter data Email: abc@gmail.com Phone:123456789  Address:Tra Vinh | Record is added to the database,Show notification success | Pass |
|  | Edit  Doctor | Enter data Email: abc@gmail.com Phone:123456789  Address:Tra Vinh | Record is added to the database,Show notification success | Pass |
|  | Delete Medicine | Press “Delete” button in Manage Medicine | Delete Data and Show notification success | Pass |
|  | Delete Employee | Press “Delete” button in Manage Employee | Delete Data and Show notification success | Pass |
|  | Delete Doctor | Press “Delete” button in Manage Doctor | Delete Data and Show notification success | Pass |
|  | Add new Patient | Enter patient details:  Patient ID: 1 Phone:0123456789  Age: 18  Address:Tra Vinh  FIRST Name: nguyen  Second name: khang | Record is added to the database | Pass |
| - Second scenario:  Same data except for the following part:  Phone: abc  Age: abc | System gives a warning that the Phone,Age format not correct. | Fail - System crash - Warning |
|  | Edit  Patient | Enter data Phone:0123456789  Address:Tra Vinh  Age: 18 | Record is added to the database,Show notification success | Pass |
|  | Delete Patient | Press “Delete” button in Manage Patient | Delete Data and Show notification success | Pass |
|  | Add new Appoiment | Enter appoiment details:  Appoiment ID: 1 Date: 10-6-2024  Time: 8:00 Patient Name: nguyen hoang khang  Doctor Name: Dr Smith | Record is added to the database | Pass |
| - Second scenario:  Same data except for the following part:  Time: -8:00  Date: abc | System gives a warning that the Date, Time format not correct. | Fail - System crash - Warning |
|  | Edit  Appoiment | Date: 10-6-2024  Time: 8:00 Doctor Name: Dr Smith | Record is added to the database,Show notification success | Pass |
|  | Delete Patient | Press “Delete” button in Manage Appoiment | Delete Data and Show notification success | Pass |
|  | Create Invoice | Enter invoice details:  Doctor Name:Dr Smith  Patient Name: nguyen hoang khang  Medicine:Cold cures  Medicine Category: Vitamin | Record is created to the database | Pass |
|  | Print Invoice | Get data from Invoice then make print invoice | After printing, the invoice will print to the desktop screen | Pass |
|  | Log in | Enter account:  Username: admin  Password: admin | Log in to the system | Pass |
| Second scenario:  Username: abc Password: abc | System gives a warning that the username or password format not correct. | Fail - System crash - Warning |
|  | Log out | Press “Delete” button in Home Page | Log out to the Sysrem | Pass |

**PROJECT REVIEW**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Project Plan Activity/Milestone** | **Work Specification** | **Status of the Activity** | **Remarks** | **Responsibility** |
| 28/4/2024 | Define project theme | Define project | Project is identified |  | whole eam |
| 12/5/2024 | Start building the database | Build database and add data | finished |  | whole team |
| 13/5/2024 | Complete Home Admin | Complete interface, features and integrations | finished |  | Do Thanh Trong |
| 13/5/2024 | Complete Home Employee | Complete interface, features and integrations | finished |  | Thach Chanh Mo Ni |
| 15/5/2024 | Complete Log in | Complete interface, features and integrations | finished |  | Nguyen Hoang Khang |
| 15/5/2024 | Complete Log out | Complete interface, features and integrations | finished |  | Nguyen Hoang Khang |
| 17/5/2024 | Complete the Manage Doctor | Complete interface, features and integrations | finished |  | Thach Chanh Mo Ni |
| 18/5/2024 | Complete the Manage Patient | Complete interface, features and integrations | finished |  | Thach Chanh Mo Ni |
| 23/5/2024 | Complete the Manage Employee | Complete interface, features and integrations | finished |  | Thach Chanh Mo Ni |
| 25/5/2024 | Complete Invoice | Complete interfac and integrations | finished |  | Do Thanh Trong |
| 26/5/2024 | Complete Print Invoice | Complete interfac and integrations | finished |  | Do Thanh Trong |
| 28/5/2024 | Complete Appionment | Complete interfac and integrations | finished |  | Do Thanh Trong |
| 30/5/2024 | Complete Revenue | Complete interfac and integrations | finished |  | Thach Chanh Mo Ni |
| 1/6/2024 | Complete Medicine | Complete interfac and integrations | finished |  | Nguyen Hoang Khang |
| 2/6/2024 | Complete Medicine Category | Complete interfac and integrations | finished |  | Nguyen Hoang Khang |
| 3/6/2024 | Complete Payment | Complete interfac and integrations | finished |  | Do Thanh Trong |

**FINAL CHECK LIST**

|  |  |  |
| --- | --- | --- |
| **No.** | **Aspect tested** | **Suggestion/Remarks** |
|  | Have all the modules been property integrated and are they completely functional? | YES |
|  | Are all the validations happening as specified in Process Design? | YES |
|  | Have all Design and Coding standards been followed and implemented? | YES |
|  | Have all the modules been property integrated and are they completely functional? | YES |
|  | Are all the validations happening as specified in Process Design? | YES |
|  | Have all Design and Coding standards been followed and implemented? | YES |
|  | Have all the modules been property integrated and are they completely functional? | YES |
|  | Are all the validations happening as specified in Process Design? | YES |