UNIVERSITY of INFORMATION TECHNOLOGY

Hospital Management System

Third Year CS

Requirement Engineering & Database Design

Group(4)

[15.3.2019]

Group Members

☺Aung Khant Myat	3cs-48
☺Aung Min Hein	3cs-52
☺Chu Khin Khin	3cs-81
☺Chue Sint Swe	3cs-50
☺May Myat Mon Kyaw	3cs-73
☺Phyo Ei Ei Thu	3cs-68

Table of Contents

1.0. Introduction

1.1 Purpose

The purpose of this docment is to present a detailed description of the Hospital Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.SRS is the agreement document between the client and the Software developer. Writing this document will also serve the following purposes.

1. Breaking the Requirements Down so that the developers will have the "big picture" view of the development. This gives us an opportunity to plan ahead which saves both project time and cost.

2.Helps in Product Validation, which means that the product we have output is Equal to the standards of the documentation in the SRS which the client satisfied and agreed on.

3.The document being consistent from the beginning to the end helps the reader understand the requirements well.

4. Easy to modify when the development team and user feel the need.

1.2 Scope of the Project

Most of the hospitals in Myanmar are using a manual system to handle things like patients making appointments at the reception, taking patients' records, ward details and other hospital informations in a tons of hard-copied files stored in special locations, calculation of bills, inventory and so on, which is a file-based system and requires lots of effort and manpower. The files can be easily damaged

by fire, insects and natural disasters. Also, could be misplaced by losing data and information. The tendency of making mistakes is high when functioning manually. It is hard to relay on the accuracy of calculations done manually too. It is more obvious for problems to arise. And keeping files takes much time and waste much precious man hours.

1.3 Glossary

Database	Collection of all the information monitored by this system.
User	Any staffs (Doctors,Nurse,etc) and Patients (registered only)
IPD	Inpatient Department(patients who are accommodated in the hospital)
OPD	Outpatient Department(Clinic patients)
Emergency	Emergency Patients
SRS(Software Requirements Specification)	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.

1.4 Reference

[2]Lauesen, S, (2003), Task Descriptions as Functional Requirements, IEEE Computer Society

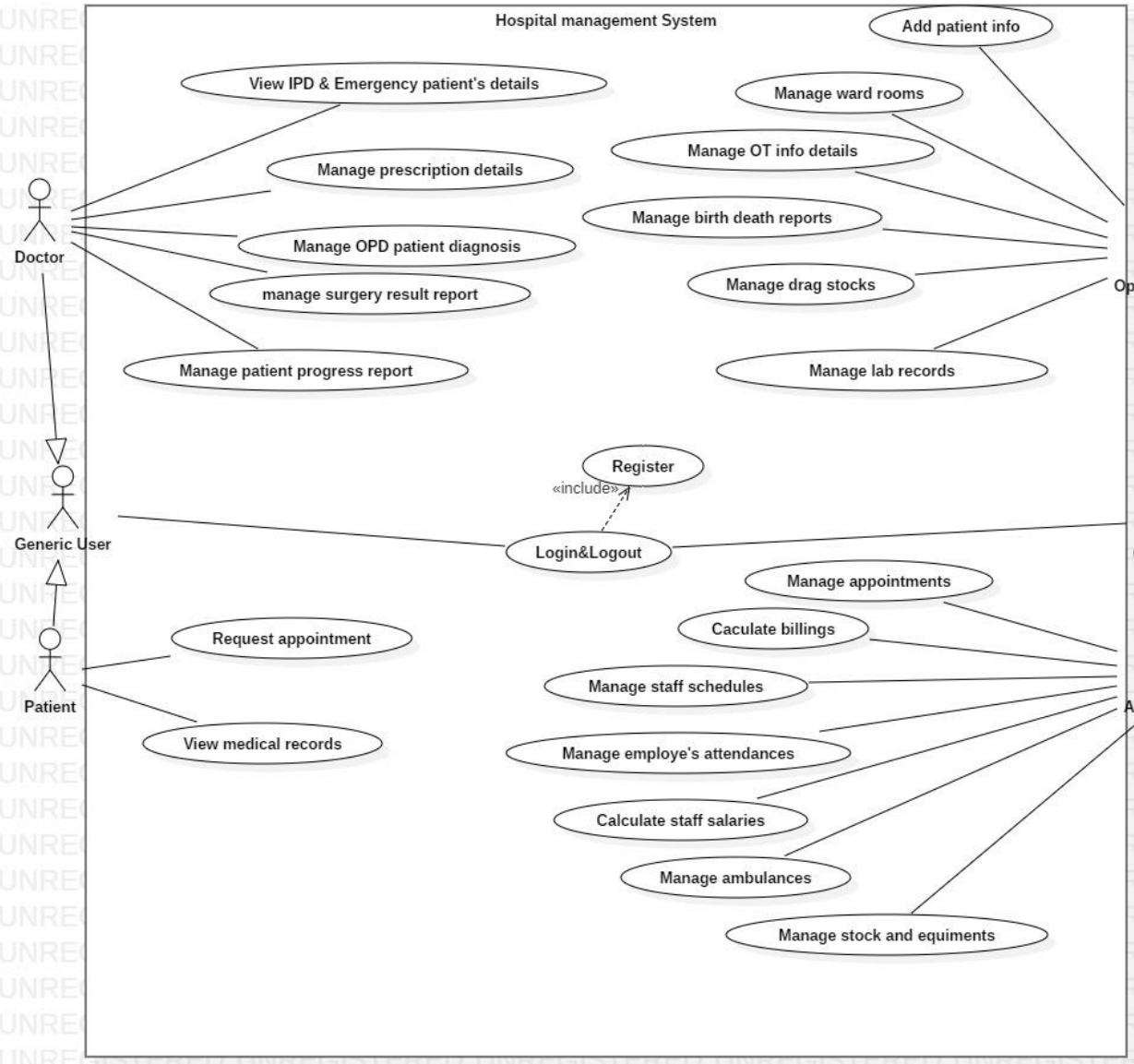
1.5 Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter. The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the

product. Both sections of the document describe the same software product in its entirety but are intended for different audiences and thus use different language.

2.0. Overall Description

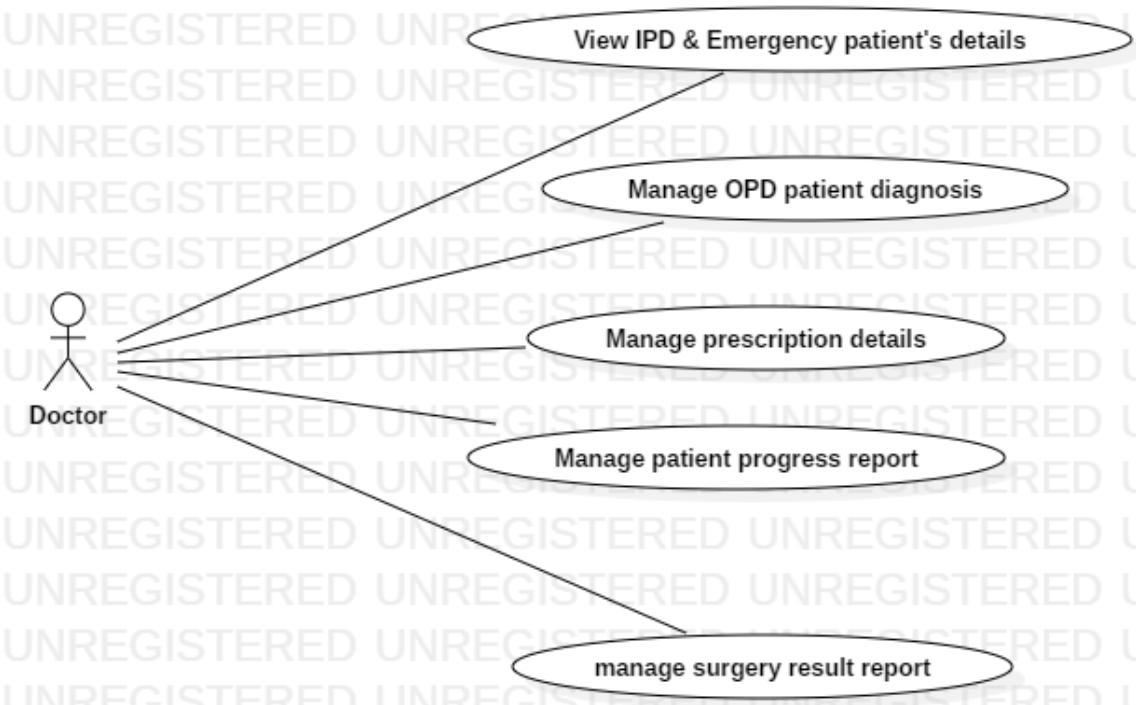
2.1 System Environment



2.2 Functional Requirements Specification

This section outlines the use cases for doctor, patient, operation staff and administrative staff.

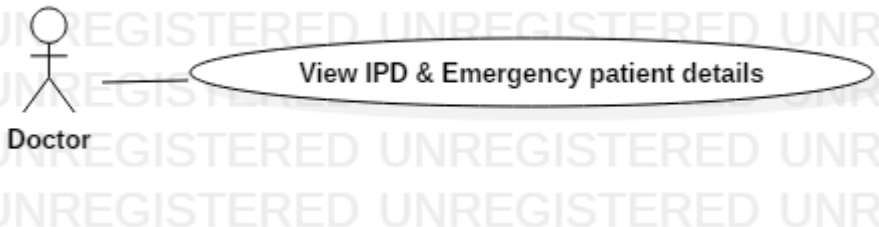
2.2.1 Doctor Use Case



General Use Cases

Use case: View IPD & Emergency patient's details

Diagram:



Brief description

The doctor view IPD and emergency patient′ s record details (especially drug allergies)

Initial step by step

1. **The doctor enter his username and password first.**
2. **Enter the login button to access and control his account.**
3. **He can choose the patient record by patient id/name in search box to view.**
4. **The doctor could continue other patients like the above process.**
5. **Log out his account.**
6. **When the doctor enter his username or password incorrectly, show error message box to try again to log in.**
7. **In searching patient′ s record, he can find by patient′ s id or name. If it does not match, the patient′ s record won′ t be appeared**

View IPD & Emergency patient's details

Name	**View IPD and emergency patient′ s details**
Summary	**In this function, the doctor view IPD and emergency patient′ s record details (especially drug allergies)**
Priority	**3**
Preconditions	**Doctor has login his/her account and the patient has entered the right information in patient′ s record.**
Postconditions	**Doctor has logout his account.**
Primary Actor(s)	**Doctor**
Secondary Actor(s)	**Patient**

Trigger	The doctor has log in to IPD and emergency patient′ s details	
Main Scenario	Step	Action
	1	Doctor enter username and password first.
	2	Enter the login button.
	3	He can choose the patient record by patient id/name in search box
	4	He could continue other patients like the above process.
	5	Log out
Extensions	Step	Branching Action
	1a	Show error message if username/password is wrong
	3a	He won′ t find the patient′ s record details with patient′ s wrong id/password
Open Issues		

Use case: Manage OPD patient diagnosis
Diagram:

Brief description

The doctor manage diagnosis and procedure of his OPD patients

Initial step by step

1. **The doctor choose the appointment session for his patient.**
2. **The system shows the list of patients sorted by the appointment order (ascending).**

3. The doctor chooses each patient from the sorted list .
4. The system shows the medical record book and lab-reports of the patient.
5. The doctor test the patient and write diagnosis and procedure for the patient in the specified position.
6. The doctor clicks the finish button and continues other patients like the above process.
7. .When the doctor search for the patient′ s appointment, the doctor can filter the list of patients by date.
8. The doctor chooses the patient and the display in front of the doctor room showing the next patient′ booking number and name.

Manage OPD patient diagnosis

Number	2
Name	Manage diagnosis and procedure of patients
Summary	Doctor manages diagnosis and procedures of patients
Priority	4
Preconditions	Doctor view the patient′ s appointment.
Postconditions	Doctor has logout his account.
Primary Actor(s)	Doctor
Secondary Actor(s)	Patient

Trigger	Doctor has login to make diagnosis and procedure of the patient.	
Main Scenario	Step	Action
	1	Doctor choose the appointment session.
	2	System shows the list of patients sorted by the appointment order.

	3	Doctor chooses each patient from the sorted list .
	4	System shows the medical record book and lab-reports of the patient.
	5	Doctor test the patient and write diagnosis and procedure for the patient in the specified position.
	6	Doctor clicks the finish button and continues other patients like the above process.
Extensions	Step	Branching Action
	1a	Doctor can filter the list of patients by date.
	3a	Doctor chooses the patient and the display in front of the doctor room and shows the next patient′ booking number and name.
Open Issues	1	

Use case: Manage prescription details
Diagram:

Brief description

The doctor can manage drug prescription details

Initial step by step

1. **The doctor enter his username and password first.**
2. **Enter the login button to access and control his account.**
3. **The doctor can write and update the prescription for his patient..**
4. **After writing the prescription, the doctor might click the finish button and is shown the alert box including the finish message.**

5. The doctor could continue other patients like the above process.The doctor clicks the finish button and continues other patients like the above process.
6. Log out his account.
7. When the doctor enter his username or password incorrectly, show error message box to try again to log in.
8. While uploading the patient' s prescription details, if the connection is not strong enough, the system should show the warning alert box.

Manage prescription details

Name	Manage prescription details
Summary	The doctor can manage drug prescription details
Priority	3
Preconditions	The doctor has login to his account.
Postconditions	The doctor has log out to his account.
Primary Actor(s)	Doctor
Secondary Actor(s)	Patient

Trigger	Manage prescription details	
Main Scenario	Step	Action
	1	Doctor enter username and password first.
	2	Enter the login button.
	3	The doctor can write and update the prescription for his patient.
	4	Doctor clicks the finish button and show the alert box including the finish message.

	5	He could continue other patients like the above process.
	6	Log out
Extensions	Step	Branching Action
	1a	Show error message if username/password is wrong
	4a	If the connection is not strong enough, the system should show the alert box.
Open Issues	1	

Use case: Manage patient progress report
Diagram:

Brief description

The doctor manage his patients' progress report

Initial step by step

1. **The doctor has login to his account.**
2. **The doctor can choose his patient progress report by name or id**
3. **The doctor write his patient' s report with exact date and time.**
4. **The system shows the medical record book and lab-reports of the patient.**
5. **The system shows the medical record book and lab-reports of the patient.**
6. **The doctor can log out his account.**
7. **If the connection is not strong enough, the system should show the alert box.**

Manage patient progress report

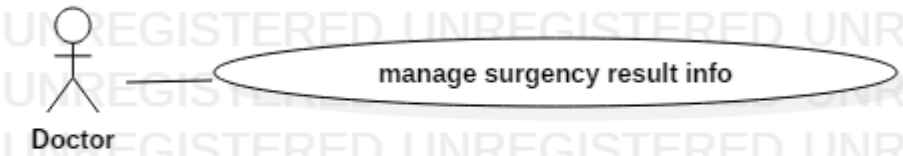
Number	5

Name	Manage patient progress report
Summary	The doctor manage his patients' progress report
Priority	3
Preconditions	The doctor has login to his account.
Postconditions	Doctor has logout his account.
Primary Actor(s)	Doctor
Secondary Actor(s)	Patient

Trigger	The doctor manage his patient' s progress reports.	
Main Scenario	Step	Action
	1	Doctor has login to his account.
	2	He can choose his patient progress report
	3	He write his patient' s report with date and time.
	4	System shows the medical record book and lab-reports of the patient.
	5	Doctor clicks the finish button and continues other patients like the above process
	6	Log out
Extensions	Step	Branching Action
	5a	If the connection is not strong enough, the system should show the alert box.
Open Issues	1	

Use case: manage surgery result report

Diagram:



Brief description

Manage surgery request report

Initial step by step

1. **The doctor choose his patient' s surgery request report.**
2. **The system shows the list of patients with requested reports sorted by requested date..**
3. **The system shows to the doctor the medical record book and lab-reports of the patient.**
4. **The doctor choose the surgery lab and arrange the assist doctors, nurses and lab equipments**
5. **The doctor clicks the finish button and continues other patients like the above process.**
6. **The doctor can filter the list of patients' request reports by date. (today-next 3 days)**

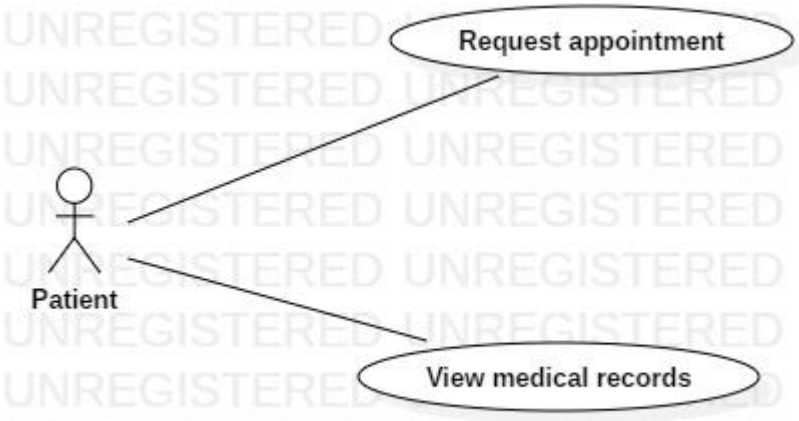
Manage surgery result report

Number	**6**
Name	**Manage surgery request report**
Summary	**The doctor manage the surgery request report**
Priority	**3**
Preconditions	**Doctor view the surgery request report .**
Postconditions	**Show finish message that the report has managed.**

Primary Actor(s)	Doctor
Secondary Actor(s)	Patient

Trigger	the doctor manage his patients′ surgery request reports	
Main Scenario	Step	Action
	1	Doctor choose the patient′ s surgery request report.
	2	System shows the list of patients sorted by requested date..
	3	System shows the medical record book and lab-reports of the patient.
	4	He choose the surgery lab and arrange the assist doctors, nurses and lab equipments
	5	Doctor clicks the finish button and continues other patients like the above process.
Extensions	Step	Branching Action
	2a	Doctor can filter the list of patients′ request reports by date.(today-next 3 days)
Open Issues	1	

2.2.2 Patient Use Case



General Use Cases

Use case: Request appointment
Diagram:

Brief Description

Patients request appointments

Initial step by step

1. **Patient log into his/her account**
2. **System will show form**
3. **Patient must fill the form and can choose doctors**
4. **Enter confirm button**
5. **Log out**
6. **Show error message if username/password is wrong**
7. **Appointment cannot be submitted if patient does not complete form**

Request appointment

Name	**Request Appointment**
Summary	**Patients request appointments**
Priority	**3**
Preconditions	**Patient has logged into his/her account**
Postconditions	**Patient has been appointment completely**
Primary Actor(s)	**Patient**
Secondary Actor(s)	**Patient Database**

Trigger	Patient has logged into account and request appointment	
Main Scenario	Step	Action
	1	Patient log in to his/her account
	2	System will show form
	3	Patient must fill the form and can choose doctors
	4	Enter confirm button
	5	Log out
Extensions	Step	Branching Action
	5a	Show error message if username/password is wrong
	5b	Appointment cannot be submitted if patient does not complete form.
Open Issues		

Use case: View medical records
Diagram:

Brief Description

Patient view his/her treatment records

Initial step by step

1. **Patient log into his/her account**
2. **System will show categories**
3. **System will choose record category**

4. System will show patient′ s arrival date to hospital, all sorts of cures, conditions, and medicines that he/she takes
5. Log out
6. Show error message if username/password is wrong
7. Should the system ask if the patient want to see the condition?

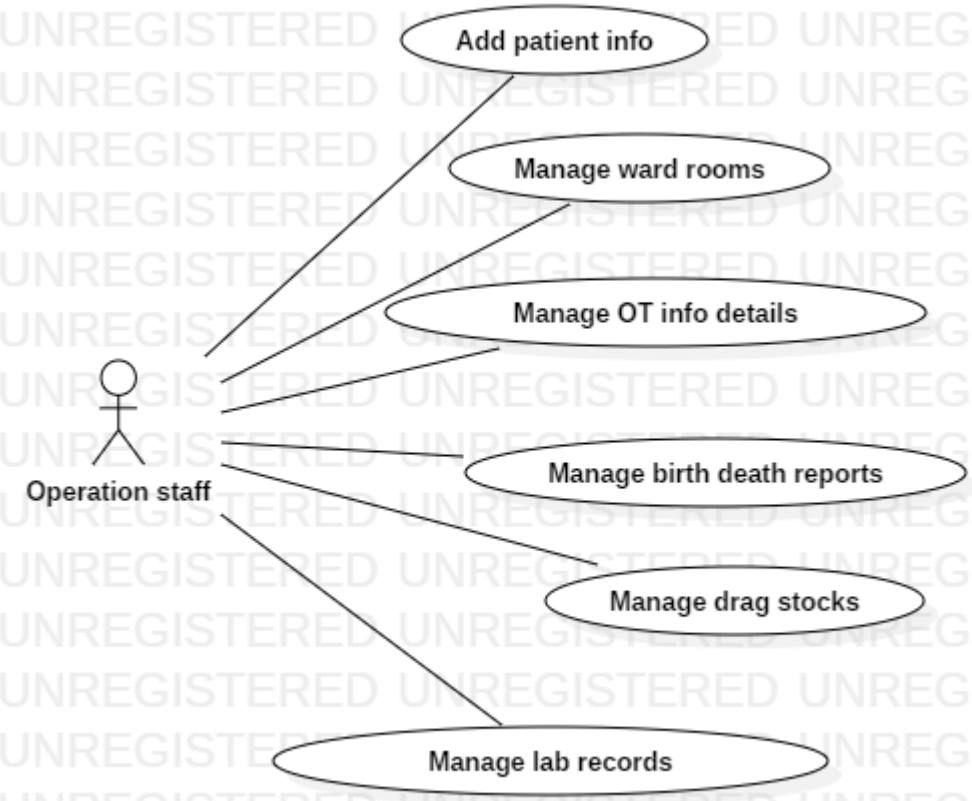
View medical records

Name	View Medical Records
Summary	Patient view his/her treatment records
Priority	3
Preconditions	Patient has logged into his/her account
Postconditions	Patient has known his/her medical records and logged out
Primary Actor(s)	Patient
Secondary Actor(s)	Record Database

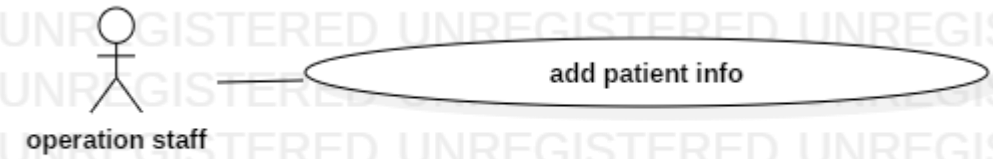
Trigger	Patient has viewed medical records	
Main Scenario	Step	Action
	1	Patient log in to his/her account
	2	System will show categories
	3	Patient will choose record category
	4	System will show patient′ s arrival date to hospital, all sorts of cures, conditions, and medicines that he/she takes
	5	Log out

Extensions	Step	Branching Action
	5a	Show error message if username/password is wrong
Open Issues	1	Should the system ask if the patients want to see the condition?

2.2.3 Operation staff Use Case



General Use Cases
Use case: Add patient info
Diagram:



Brief Description

Staffs will add information about patients

Initial step by step

1. **Staff log into his/her account**
2. **System will show categories**
3. **Staff will choose add patient category**
4. **Add patient' s information and validate it**
5. **If information is in format, staff will click submit button**
6. **Log out**
7. **Show error message if username/password is wrong**

Add patient info

Name	**Add Patient Info**
Summary	**Staffs will add information about patients**
Priority	**3**
Preconditions	**Staff has logged into his/her account**
Postconditions	**Staff has been added information completely**
Primary Actor(s)	**Staff**
Secondary Actor(s)	**Patient**

Trigger	**Staff will log into account and add patient' s information**	
Main Scenario	**Step**	**Action**
	1	**Staff log in his/her account**
	2	**System will show categories**
	3	**Staff will choose add patient category**

	4	Add patient′ s information and validate it
	5	If information is in format. staff will click submit button
	6	Log out
Extensions	Step	Branching Action
	5a	Show error message if username/ password is wrong
Open Issues		

Use case: Manage ward rooms
Diagram:

Brief Description

Staff will manage ward rooms for patients

Initial step by step

1. **Staff log into his/her account**
2. **Staff will show categories**
3. **Staff will choose ward room category**
4. **Staff categorize ward room**
5. **Assign/Remove patients from ward rooms**
6. **Staff checks ward rooms/beds availability**
7. **Staff track information when shifted from one ward to another ward**

Manage ward rooms

Name	Manage ward rooms
Summary	Staff will manage ward rooms for patients
Priority	3
Preconditions	Staff has logged into his/her account
Postconditions	Staff manages ward rooms
Primary Actor(s)	Staff
Secondary Actor(s)	Patient

Trigger	Staff will log into account and manage ward rooms	
Main Scenario	Step	Action
	1	Staff log in his/her account
	2	System will show categories
	3	Staff will choose ward room category
	4	Staff categorize ward room
	5	Assign/Remove patients from ward rooms
	6	Staff checks ward rooms/beds availability
	7	Staff track information when shifted from one ward to another ward
	8	Enter submit button if all stuffs are done
	9	Log out

Extensions	Step	Branching Action
	5a	Show error message if username/password is wrong
	5b	Patient' s ward room numbers, name of doctors who patients see are shown on board near reception
Open Issues		

Use case: Manage OT info details
Diagram:

Brief Description

Staff will manage OT information of patients

Initial step by step

1. **Staff log into his/her account**
2. **System will show categories**
3. **Staff will choose OT information category**
4. **Staff chooses patient to see information and records**
5. **Staff can insert/update/delete patient' s OT information, team name and surgeon team**
6. **Enter submit button**
7. **Log out .**
8. **Show error message if username/password is wrong**
9. **Should the system ask if the staff wants to see starting time and finishing time of patients meeting with doctors?**

Manage OT info details

Name	**Manage OT info details**
Summary	**Staff will manage OT information of patients**
Priority	**3**
Preconditions	**Staff has logged into his/her account**
Postconditions	**Staff manages OT information details**
Primary Actor(s)	**Staff**
Secondary Actor(s)	**Patient**

Trigger	**Staff will log into account and manage OT information**	
Main Scenario	**Step**	**Action**
	1	**Staff log in his/her account**
	2	**System will show categories**
	3	**Staff will choose OT information category**
	4	**Staff chooses patient to see information and records**
	5	**Staff can insert/update/delete patient′ s OT information, team name and surgeon team**
	6	**Enter submit button**
	7	**Log out**
Extensions	**Step**	**Branching Action**
	5a	**Show error message if username/password is wrong**

Open Issues	**1**	**Should the system ask if the staff wants to see starting time and finishing time of patients meeting with doctors?**

Use case: Manage birth death reports
Diagram:

Brief Description

staff will manage birth & death report of the hospital

Initial step by step

1. **Staff log into his/her account**
2. **System will show categories**
3. **Staff will choose birth/death report category**
4. **Staff can enter the birth information or death information in required form**
5. **Enter submit button**
6. **Log out .**
7. **Show error message if username/password is wrong and at least one field is not filled and invalid.**

manage birth death report

Name	**Manage birth & death report**
Summary	**staff will manage birth & death report of the hospital**
Priority	**3**
Preconditions	**Staff has logged into his/her account**

Postconditions	**Staff manages birth & death information details**
Primary Actor(s)	**Staff**
Secondary Actor(s)	**Patient**

Trigger	**Staff will log into account and manage OT information**	
Main Scenario	**Step**	**Action**
	1	**Staff log in his/her account**
	2	**System will show categories**
	3	**Staff will choose birth/death report category**
	4	**Staff can insert/update/delete the birth information or death information**
	5	**Enter submit button**
	6	**Log out**
Extensions	**Step**	**Branching Action**
	4a	**Show error message if username/password is wrong and at least one field is not filled and invalid.**
Open Issues	**1**	

Use case: Manage drug stock
Diagram:

Brief Description

staff will manage the drug stocks of the pharmacy in the hospital.

Initial step by step

1. **Staff log into his/her account**
2. **System will show categories**
3. **Staff will choose "manage drug stock" category**
4. **Staff can insert/update/delete drug stock information of the pharmacy**
5. **Enter submit button**
6. **Log out .**
7. **Show error message if username/password is wrong and at least one field is not filled and invalid.**

manage drug stock

Name	**manage drug stock**
Summary	**staff will manage the drug stocks of the pharmacy in the hospital.**
Priority	**3**
Preconditions	**Staff has logged into his/her account**
Postconditions	**Staff manages drug stock information details**
Primary Actor(s)	**Staff**
Secondary Actor(s)	**database**

Trigger	**Staff will log into account and manage drug stock**	
Main Scenario	**Step**	**Action**

	1	Staff log in his/her account
	2	System will show categories
	3	Staff will choose "manage drug stock" category
	4	Staff can insert/update/delete drug stock information of the pharmacy
	5	Enter submit button
	6	Log out
Extensions	Step	Branching Action
	4a	Show error message if username/password is wrong and at least one field is not filled and invalid.
Open Issues	1	

Use case: Manage lab records
Diagram:

Brief Description

staff will manage the lab records in the hospital.

Initial step by step

1. **Staff log into his/her account**
2. **System will show categories**
3. **Staff will choose "mange Lab records" category**
4. **Staff can insert/update/delete lab result records of the patient.**
5. **Staff can view lab result records of the patient.**
6. **Enter submit button**

7. Log out .
8. Show error message if username/password is wrong and at least one field is not filled and invalid.

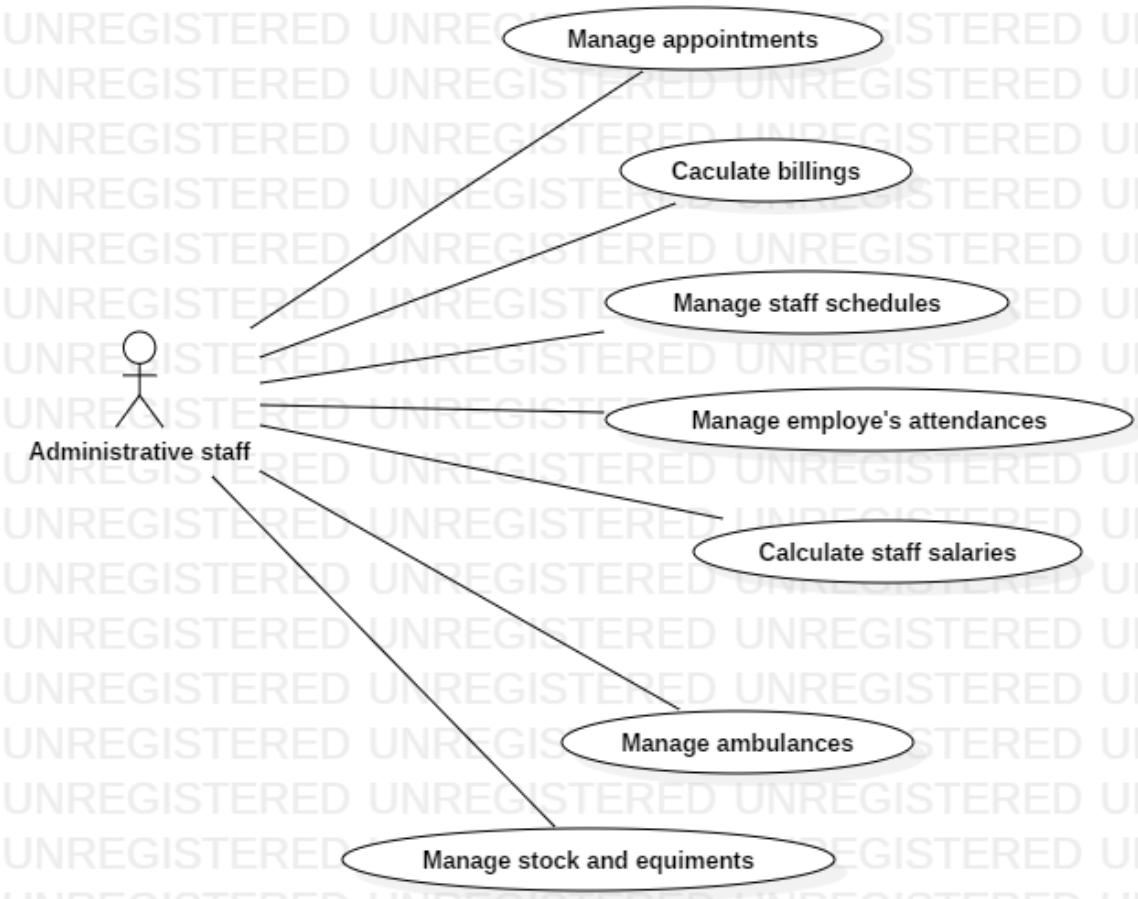
manage lab records

Name	manage lab records
Summary	staff will manage the lab records in the hospital.
Priority	3
Preconditions	Staff has logged into his/her account
Postconditions	Staff manages lab records in the hospital
Primary Actor(s)	Staff
Secondary Actor(s)	database

Trigger	Staff will log into account and manage lab records	
Main Scenario	Step	Action
	1	Staff log in his/her account
	2	System will show categories
	3	Staff will choose "mange Lab records" category
	4	Staff can insert/update/delete lab result records of the patient.
	5	Enter submit button
	6	Log out
Extensions	Step	Branching Action

	4a	Show error message if username/password is wrong and at least one field is not filled and invalid.
Open Issues	1	

2.4 Administrative staff Use Case



General Use Cases
Use case: Manage appointments
Diagram:

Brief Description

Receptionist has to login his/her account and manage appointments.

Initial step by step

1. **. Receptionist enters username and password first.**
2. **Enter the login button.**
3. **Staff will choose OT information category**
4. **He can manage patient′ s appointments.**
5. **He could continue other patients like the above process.**
6. **Log out his account.**
7. **Show error message if username/password is wrong**
8. **He won′ t find the patient′ s appointments with patient′ s wrong id/password.**

Manage appointments

Name	**Manage Appointments**
Summary	**In this function, the receptionist has to manage appointments for patients**
Priority	**3**
Preconditions	**Receptionist has to login his/her account and manage appointments**
Postconditions	**Receptionist has logout his account.**
Primary Actor(s)	**Receptionist**

Secondary Actor(s)	Patient

Trigger	Receptionist has to login his/her account and manage appointments.	
Main Scenario	Step	Action
	1	Receptionist enters username and password first.
	2	Enter the login button.
	3	He can manage patient′ s appointments.
	4	He could continue other patients like the above process.
	5	Log out
Extensions	Step	Branching Action
	1a	Show error message if username/password is wrong
	3a	He won′ t find the patient′ s appointments with patient′ s wrong id/password
Open Issues		

Use case: Calculate billings

Diagram

Brief Description

Receptionist and Pharmacist can calculate billings.

Initial step by step

1. **Receptionist and Pharmacist enter username and password first.**
2. **Enter the login button.**

3. They can calculate billings of appointments and pharmacies for their patient.
4. They clicks the finish button and show the alert box including the finish message.
5. They could continue other patients like the above process
6. Log out his account.
7. . Show error message if username/password is wrong

Calculate billings

Name	Calculate billings
Summary	Receptionist and Pharmacist calculate bills for apppointments and pharmacies of patients.
Priority	3
Preconditions	Receptionist and Pharmacist have to log in their accounts.
Postconditions	Receptionist and Pharmacist have to log out their accounts.
Primary Actor(s)	Receptionist and Pharmacist
Secondary Actor(s)	Patient

Trigger	Calculate billings	
Main Scenario	Step	Action
	1	Receptionist and Pharmacist enter username and password first.
	2	Enter the login button.
	3	They can calculate billings of appointments and pharmacies for their patient.
	4	They clicks the finish button and show the alert box including the finish message.

	5	They could continue other patients like the above process.
	6	Log out
Extensions	Step	Branching Action
	1a	Show error message if username/password is wrong
Open Issues	1	

Use case: Manage staff schedules
Diagram

Brief Description

HR Department manages staff schedules

Initial step by step

1. **HR Department has to log in.**
2. **.Lists of schedules of staffs will be shown.**
3. **HR Department can manage schedules of each and everyone of staffs.**
4. **They can click the finish button to complete managing**
5. **. Log out.**
6. **HR Department can filter the list of staffs by type.**

Manage staff schedules

Number	3
Name	Manage staff schedules.
Summary	HR Department manages staff schedules.
Priority	3

Preconditions	**HR Department has to log in.**
Postconditions	**HR Department has to logout his account.**
Primary Actor(s)	**HR Department**
Secondary Actor(s)	**Staff**

Trigger	**HR Department manages staff schedules.**	
Main Scenario	**Step**	**Action**
	1	**HR Department has to log in.**
	2	**Lists of schedules of staffs will be shown.**
	3	**HR Department can manage schedules of each and everyone of staffs.**
	4	**They can click the finish button to complete managing.**
	5	**Log out.**
Extensions	**Step**	**Branching Action**
	2a	**HR Department can filter the list of staffs by type.**
Open Issues	**1**	

Use case: Manage employee's attendance
Diagram

Brief Description

HR Department manages employee' s attendances.

Initial step by step

1. HR Department has to log in.
2. Lists of attendances of employees will be shown.
3. HR Department can manage schedules of each and everyone of employees.
4. They can click the finish button to complete managing
5. Log out.
6. HR Department can filter the list of staffs by type.

Manage employee's attendance

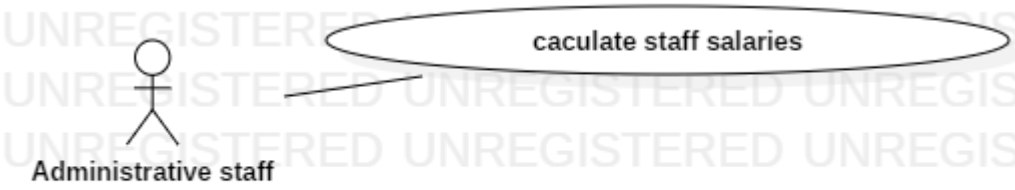
Number	4
Name	Manage Employee′ s Attendences
Summary	HR Department manages employee′ s attendences.
Priority	3
Preconditions	HR Department has to log in.
Postconditions	HR Department has to logout.
Primary Actor(s)	HR Department
Secondary Actor(s)	Employee

Trigger	HR Department manages employee′ s attendences.	
Main Scenario	Step	Action
	1	HR Department has to log in.
	2	Lists of attendences of employees will be shown.
	3	HR Department can manage schedules of each and everyone of employees.

	4	They can click the finish button to complete managing.
	5	Log out.
Extensions	Step	Branching Action
	2a	HR Department can filter the list of employees by type.
Open Issues	1	

Use case: Calculate staff salaries

Diagram



Brief Description

The HR Department calculate salaries of staffs

Initial step by step

1. **HR Department has to log in.**
2. **Lists of types and salaries of staffs will be shown.**
3. **HR Department can calculate salaries of each and everyone of staffs according to their rank and type.**
4. **They can click the finish button to complete calculating.**
5. **Log out.**
6. **HR Department can filter the list of staffs by type and rank.**

Calculate staff salaries

Number	5
Name	Calculate staff salaries.

Summary	**The HR Department calculate salaries of staffs**
Priority	**3**
Preconditions	**HR Department has to log in.**
Postconditions	**HR Department has to log out.**
Primary Actor(s)	**HR Department**
Secondary Actor(s)	**Staff**

Trigger	**The HR Department calculate salaries of staffs**	
Main Scenario	**Step**	**Action**
	1	**HR Department has to log in.**
	2	**Lists of types and salaries of staffs will be shown.**
	3	**HR Department can calculate salaries of each and everyone of staffs according to their rank and type.**
	4	**They can click the finish button to complete calculating.**
	5	**Log out.**
Extensions	**Step**	**Branching Action**
	5a	**HR Department can filter the list of staffs by type and rank.**
Open Issues	**1**	

Use case: Manage ambulances
Diagram

Brief Description

The administrator has to manage ambulances for emergency.

Initial step by step

1. **The administrator has to log in.**
2. **Enter the login button.**
3. **He can manage ambulances for emergemcy and other purposes.**
4. **He could continue other ambulances like the above process.**
5. **Log out.**
6. **Show error message if username/password is wrong**
7. **He won′ t find the ambulances′ details with patient′ s wrong id/password**

Manage ambulances

Number	**6**
Name	**Manage ambulances.**
Summary	**The administrator has to manage ambulances for emergency.**
Priority	**3**
Preconditions	**The administrator has to log in.**
Postconditions	**The administrator has to log out.**
Primary Actor(s)	**The administrator**
Secondary Actor(s)	

Trigger	**The administrator has to manage ambulances for emergency.**	
Main Scenario	**Step**	**Action**
	1	**The administrator has to log in.**

	2	Enter the login button.
	3	He can manage ambulances for emergemcy and other purposes.
	4	He could continue other ambulances like the above process.
	5	Log out
Extensions	Step	Branching Action
	1a	Show error message if username/password is wrong
	3a	He won′ t find the ambulances′ details with patient′ s wrong id/password
Open Issues		

Use case: Manage stock and equipment
Diagram

`

Brief Description

The administrator has to manage stock and equipment of the hospital.

Initial step by step

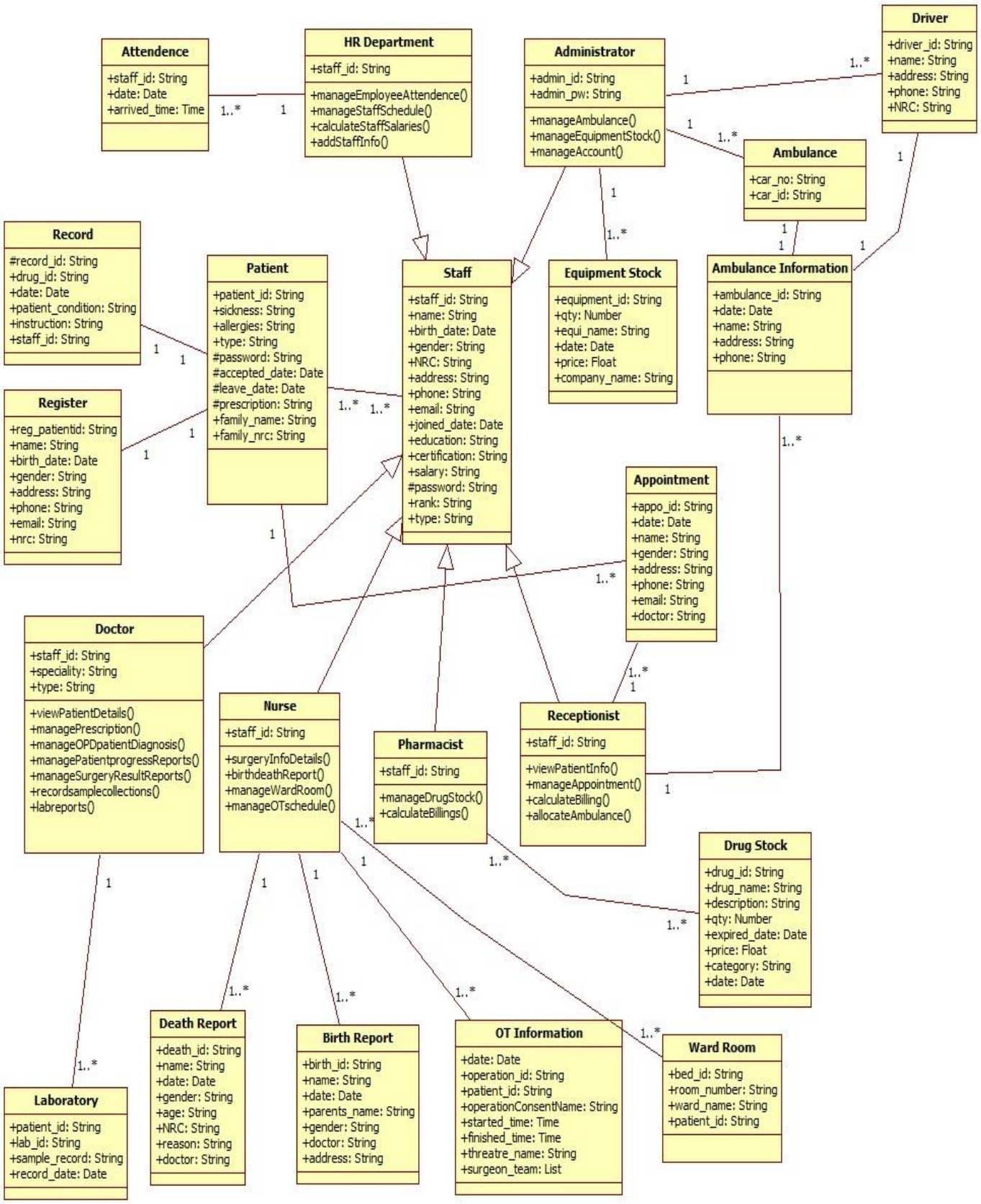
1. **The administrator has to log in.**
2. **Enter the login button.**
3. **He can manage stock and equipment of the hospital.**
4. **Log out.**
5. **Show error message if username/password is wrong**
6. **He won′ t find the stock and equipment details with equipment′ s wrong id**

Manage stock and equipment

Number	6
Name	Manage stock and equipment
Summary	The administrator has to manage stock and equipment of the hospital. .
Priority	3
Preconditions	The administrator has to log in.
Postconditions	The administrator has to log out.
Primary Actor(s)	The administrator
Secondary Actor(s)	Database

Trigger	The administrator has to manage stock and equipment.	
Main Scenario	Step	Action
	1	The administrator has to log in.
	2	Enter the login button.
	3	He can manage stock and equipment of the hospital.
	5	Log out
Extensions	Step	Branching Action
	1a	Show error message if username/password is wrong
	3a	He won′ t find the stock and equipment details with equipment′ s wrong id
Open Issues		

2.3 Class Diagram



2.4 Non-Functional Requirement

i. Reliability
- **24 hours.**
- **Accurate schedule**
- **Easy to understand**

ii. Performance
- **The system response time will be less than 5 seconds for any types of websites.**
- **More than 2000 people can use the system at the same time.**

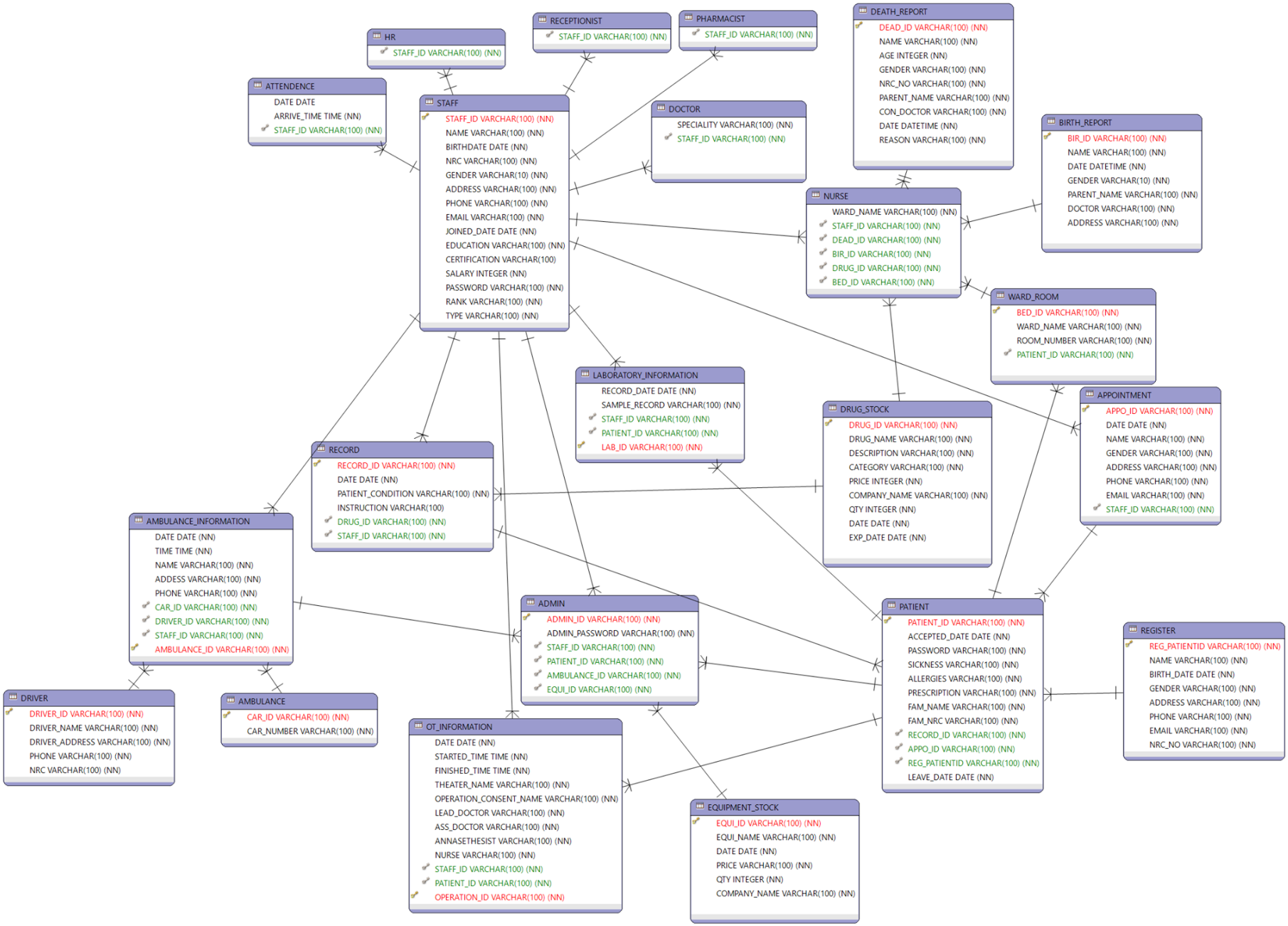
iii. Security
- **patient identification**
- **Login ID**
- **Modification**
- **Reception view**

iv. Usability
- **Only English language is support**

v. Safety
- **If there is a security attack, the system should not be operate.**

3.0 Database Management System

3.1 ER Diagram


HR
STAFF_ID VARCHAR(100) (NN)
RECEPTIONIST
STAFF_ID VARCHAR(100) (NN)
PHARMACIST
STAFF_ID VARCHAR(100) (NN)
DEATH_REPORT
DEAD_ID VARCHAR(100) (NN)
NAME VARCHAR(100) (NN)
AGE INTEGER (NN)
GENDER VARCHAR(100) (NN)
NRC_NO VARCHAR(100) (NN)
PARENT_NAME VARCHAR(100) (NN)
CON_DOCTOR VARCHAR(100) (NN)
DATE DATETIME (NN)
REASON VARCHAR(100) (NN)
ATTENDENCE
DATE DATE
ARRIVE_TIME TIME (NN)
STAFF_ID VARCHAR(100) (NN)
STAFF
STAFF_ID VARCHAR(100) (NN)
NAME VARCHAR(100) (NN)
BIRTHDATE DATE (NN)
NRC VARCHAR(100) (NN)
GENDER VARCHAR(10) (NN)
ADDRESS VARCHAR(100) (NN)
PHONE VARCHAR(100) (NN)
EMAIL VARCHAR(100) (NN)
JOINED_DATE DATE (NN)
EDUCATION VARCHAR(100) (NN)
CERTIFICATION VARCHAR(100)
SALARY INTEGER (NN)
PASSWORD VARCHAR(100) (NN)
RANK VARCHAR(100) (NN)
TYPE VARCHAR(100) (NN)
DOCTOR
SPECIALITY VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
BIRTH_REPORT
BIR_ID VARCHAR(100) (NN)
NAME VARCHAR(100) (NN)
DATE DATETIME (NN)
GENDER VARCHAR(10) (NN)
PARENT_NAME VARCHAR(100) (NN)
DOCTOR VARCHAR(100) (NN)
ADDRESS VARCHAR(100) (NN)
NURSE
WARD_NAME VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
DEAD_ID VARCHAR(100) (NN)
BIR_ID VARCHAR(100) (NN)
DRUG_ID VARCHAR(100) (NN)
BED_ID VARCHAR(100) (NN)
WARD_ROOM
BED_ID VARCHAR(100) (NN)
WARD_NAME VARCHAR(100) (NN)
ROOM_NUMBER VARCHAR(100) (NN)
PATIENT_ID VARCHAR(100) (NN)
LABORATORY_INFORMATION
RECORD_DATE DATE (NN)
SAMPLE_RECORD VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
PATIENT_ID VARCHAR(100) (NN)
LAB_ID VARCHAR(100) (NN)
DRUG_STOCK
DRUG_ID VARCHAR(100) (NN)
DRUG_NAME VARCHAR(100) (NN)
DESCRIPTION VARCHAR(100) (NN)
CATEGORY VARCHAR(100) (NN)
PRICE INTEGER (NN)
COMPANY_NAME VARCHAR(100) (NN)
QTY INTEGER (NN)
DATE DATE (NN)
EXP_DATE DATE (NN)
APPOINTMENT
APPO_ID VARCHAR(100) (NN)
DATE DATE (NN)
NAME VARCHAR(100) (NN)
GENDER VARCHAR(100) (NN)
ADDRESS VARCHAR(100) (NN)
PHONE VARCHAR(100) (NN)
EMAIL VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
RECORD
RECORD_ID VARCHAR(100) (NN)
DATE DATE (NN)
PATIENT_CONDITION VARCHAR(100) (NN)
INSTRUCTION VARCHAR(100)
DRUG_ID VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
AMBULANCE_INFORMATION
DATE DATE (NN)
TIME TIME (NN)
NAME VARCHAR(100) (NN)
ADDESS VARCHAR(100) (NN)
PHONE VARCHAR(100) (NN)
CAR_ID VARCHAR(100) (NN)
DRIVER_ID VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
AMBULANCE_ID VARCHAR(100) (NN)
ADMIN
ADMIN_ID VARCHAR(100) (NN)
ADMIN_PASSWORD VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
PATIENT_ID VARCHAR(100) (NN)
AMBULANCE_ID VARCHAR(100) (NN)
EQUI_ID VARCHAR(100) (NN)
PATIENT
PATIENT_ID VARCHAR(100) (NN)
ACCEPTED_DATE DATE (NN)
PASSWORD VARCHAR(100) (NN)
SICKNESS VARCHAR(100) (NN)
ALLERGIES VARCHAR(100) (NN)
PRESCRIPTION VARCHAR(100) (NN)
FAM_NAME VARCHAR(100) (NN)
FAM_NRC VARCHAR(100) (NN)
RECORD_ID VARCHAR(100) (NN)
APPO_ID VARCHAR(100) (NN)
REG_PATIENTID VARCHAR(100) (NN)
LEAVE_DATE DATE (NN)
REGISTER
REG_PATIENTID VARCHAR(100) (NN)
NAME VARCHAR(100) (NN)
BIRTH_DATE DATE (NN)
GENDER VARCHAR(100) (NN)
ADDRESS VARCHAR(100) (NN)
PHONE VARCHAR(100) (NN)
EMAIL VARCHAR(100) (NN)
NRC_NO VARCHAR(100) (NN)
DRIVER
DRIVER_ID VARCHAR(100) (NN)
DRIVER_NAME VARCHAR(100) (NN)
DRIVER_ADDRESS VARCHAR(100) (NN)
PHONE VARCHAR(100) (NN)
NRC VARCHAR(100) (NN)
AMBULANCE
CAR_ID VARCHAR(100) (NN)
CAR_NUMBER VARCHAR(100) (NN)
OT_INFORMATION
DATE DATE (NN)
STARTED_TIME TIME (NN)
FINISHED_TIME TIME (NN)
THEATER_NAME VARCHAR(100) (NN)
OPERATION_CONSENT_NAME VARCHAR(100) (NN)
LEAD_DOCTOR VARCHAR(100) (NN)
ASS_DOCTOR VARCHAR(100) (NN)
ANNASETHESIST VARCHAR(100) (NN)
NURSE VARCHAR(100) (NN)
STAFF_ID VARCHAR(100) (NN)
PATIENT_ID VARCHAR(100) (NN)
OPERATION_ID VARCHAR(100) (NN)
EQUIPMENT_STOCK
EQUI_ID VARCHAR(100) (NN)
EQUI_NAME VARCHAR(100) (NN)
DATE DATE (NN)
PRICE VARCHAR(100) (NN)
QTY INTEGER (NN)
COMPANY_NAME VARCHAR(100) (NN)

3.2 Data directory

TABLE	ATTRIBUTE	TYPE	REQUIRED	PK/FK	FK REFERENCE TABLE
STAFF	STAFF_ID	VARCHAR(100)	YES	PK	
	NAME	VARCHAR(100)	YES		
	BIRTHDATE	DATE	YES		
	NRC	VARCHAR(100)	YES		
	GENDER	VARCHAR(10)	YES		
	ADDRESS	VARCHAR(100)	YES		
	PHONE	VARCHAR(100)	YES		
	EMAIL	VARCHAR(100)	YES		
	JOINED_DATE	DATE	YES		
	EDUCATION	VARCHAR(100)	YES		
	CERTIFICATION	VARCHAR(100)			
	SALARY	INTEGER	YES		
	PASSWORD	INTEGER	YES		
	RANK	VARCHAR(100)	YES		
	TYPE	VARCHAR(100)	YES		
DOCTOR	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
	SPECIALITY	VARCHAR(100)	YES		
NURSE	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
	WARD-NAME	VARCHAR(100)	YES		
	DEAD_ID	VARCHAR(100)	YES	FK	DEATH_REPORT
	BIRTH_ID	VARCHAR(100)	YES	FK	BIRTH_REPORT
	DRUG ID	VARCHAR(100)	YES	FK	DRUG_STOCK
	BED ID	VARCHAR(100)	YES	FK	WARD_ROOM
PHARMACIST	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
RECEPTIONIST	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
PATIENT	PATIENT_ID	VARCHAR(100)	YES	PK	
	REG_PATIENTID	VARCHAR(100)	YES	FK	REGISTER
	ACCEPTED_DATE	DATE	YES		
	LEAVED_DATE	DATE	YES		
	PASSWORD	VARCHAR(10)	YES		
	RECORD_ID	VARCHAR(100)	YES	FK	RECORD
	APPO_ID	VARCHAR(100)	YES	FK	APPOINTMENT
	SICKNESS	VARCHAR(100)	YES		
	ALLEGRIES	VARCHAR(100)	YES		
	PRESCRIPTION	VARCHAR(100)	YES		
	FAM_NAME	VARCHAR(100)	YES		
	FAM_NRC	VARCHAR(100)	YES		
RECORD	RECORD_ID	VARCHAR(100)	YES	PK	
	DRUG_ID	VARCHAR(100)	YES	FK	DRUG_STOCK

	DATE	DATE	YES		
	PATIENTCONDITION	VARCHAR(100)	YES		
	INSTRUCTION	VARCHAR(100)	YES		
	STAFF_ID	VARCHAR(100)	YES		

REGISTER	**REG_PATIENTID**	**VARCHAR(100)**	**YES**	**PK**	
	NAME	VARCHAR(100)	YES		
	BIRTHDATE	VARCHAR(100)	YES		
	GENDER	VARCHAR(10)	YES		
	NRC	VARCHAR(100)	YES		
	ADDRESS	VARCHAR(100)	YES		
	PHONE	VARCHAR(100)	YES		
	EMAIL	VARCHAR(100)	YES		
LABORATORY_ INFORMATION	LAB_ID	VARCHAR(100)	YES	PK	
	RECORDER-DATE	DATE	YES		
	PATIENT_ID	VARCHAR(100)	YES	FK	PATIENT
	SAMPLE_RECORD	VARCHAR(100)	YES		
	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
DEATH_REPORT	DEAD_ID	VARCHAR(100)	YES	PK	
	NAME	VARCHAR(100)	YES		
	AGE	INTEGER	YES		
	GENDER	VARCHAR(10)	YES		
	NRC_NO	VARCHAR(100)	YES		
	PARENT_NAME	VARCHAR(100)	YES		
	CON_DOCTOR	VARCHAR(100)	YES		
	DATE	DATETIME	YES		
	REASON	VARCHAR(100)	YES		

BIRTH_REPORT	BIR_ID	VARCHAR(100)	YES	PK	
	NAME	VARCHAR(100)	YES		
	DATE	DATETIME	YES		
	GENDER	VARCHAR(10)	YES		
	PARENT_NAME	VARCHAR(100)	YES		
	DOCTOR	VARCHAR(100)	YES		
	ADDRESS	VARCHAR(100)	YES		
OT_INFORMATION	OPERATION_ID	VARCHAR(100)	YES	PK	
	PATIENT_ID	VARCHAR(100)	YES	FK	PATIENT
	DATE	DATE	YES		
	STARTED_TIME	TIME	YES		
	FINISHED_TIME	TIME			
	THEATER-NAME	VARCHAR(100)	YES		
	OPERATIONCONSENTNAME	VARCHAR(100)	YES		
	LEADDOCTOR	VARCHAR(100)	YES		
	ASSDOCTOR	VARCHAR(100)	YES		
	ANNASETHESIST	VARCHAR(100)	YES		
	NURSE	VARCHAR(100)	YES		
	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
WARD_ROOM	BED_ID	VARCHAR(100)	YES	PK	
	WARD NAME	VARCHAR(100)	YES		
	ROOM_NUMBER	VARCHAR(100)	YES		
	PATIENT_ID	VARCHAR(100)	YES		
DRUG_STOCK	DRUG_ID	VARCHAR(100)	YES	PK	
	DRUG_NAME	VARCHAR(100)	YES		

	DESCRIPTION	VARCHAR(100)	YES		
	CATEGORY	VARCHAR(100)	YES		
	PRICE	INTEGER	YES		
	COMPANYNAME	VARCHAR(100)	YES		
	QTY	INTEGER	YES		
	DATE	DATE	YES		
	EXPIREDATE	DATE	YES		
APPOINTMENT	APPO_ID	VARCHAR(100)	YES	PK	
	DATE	DATE	YES		
	NAME	VARCHAR(100)	YES		
	GENDER	VARCHAR(100)	YES		
	ADDRESS	VARCHAR(100)	YES		
	PHONE	VARCHAR(100)	YES		
	EMAIL	VARCHAR(100)	YES		
	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
AMBULANCE_INFORMATION	AMBULANCE_ID	VARCHAR(100)	YES	PK	
	CAR_ID	VARCHAR(100)	YES	FK	AMBULANCE
	DRIVER_ID	VARCHAR(100)	YES	FK	DRIVER
	DATE	DATE	YES		
	TIME	TIME	YES		
	NAME	VARCHAR(100)	YES		
	ADDRESS	VARCHAR(100)	YES		
	PHONE	VARCHAR(100)	YES		
	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
AMBULANCE	CAR_ID	VARCHAR(100)	YES	PK	

	CAR_NUMBER	VARCHAR(100)	YES		
DRIVER	DRIVER_ID	VARCHAR(100)	YES	PK	
	DRIVER_NAME	VARCHAR(100)	YES		
	DRIVER_ADDRESS	VARCHAR(100)	YES		
	PHONE	VARCHAR(100)	YES		
	NRC	VARCHAR(100)	YES		
ADMIN	ADMIN_ID	VARCHAR(100)	YES		
	ADMIN_PASSWORD	VARCHAR(100)	YES		
	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
	PATIENT_ID	VARCHAR(100)	YES	FK	PATIENT
	AMBULANCE_ID	VARCHAR(100)	YES	FK	AMBULANCE_INFORMATION
	EQUI_ID	VARCHAR(100)	YES	FK	EQUIPMENT_STOCK
EQUIPMENT_STOCK	EQUI_ID	VARCHAR(100)	YES	PK	
	EQUI_NAME	VARCHAR(100)	YES		
	DATE	DATE	YES		
	PRICE	VARCHAR(100)	YES		
	QTY	INTEGER	YES		
	COMPANYNAME	VARCHAR(100)	YES		
	ADMIN_ID	VARCHAR(100)	YES		
HR	STAFF_ID	VARCHAR(100)	YES		
ATTENDANCE	DATE	DATE	YES		
	STAFF_ID	VARCHAR(100)	YES	FK	STAFF
	ARRIVEDTIME	TIME	YES		

3.3 Data Definition Language

3.3.1 Creating databases and tables for the system into Mysql

```
/* Create Tables */

CREATE TABLE ADMIN
(
	ADMIN_ID VARCHAR(100) NOT NULL,
	ADMIN_PASSWORD VARCHAR(100) NOT NULL,
	STAFF_ID VARCHAR(100) NOT NULL,
	PATIENT_ID VARCHAR(100) NOT NULL,
	AMBULANCE_ID VARCHAR(100) NOT NULL,
	EQUI_ID VARCHAR(100) NOT NULL,
	PRIMARY KEY (ADMIN_ID)
);


CREATE TABLE AMBULANCE
(
	CAR_ID VARCHAR(100) NOT NULL,
	CAR_NUMBER VARCHAR(100) NOT NULL,
	PRIMARY KEY (CAR_ID)
);


CREATE TABLE AMBULANCE_INFORMATION
(
	DATE DATE NOT NULL,
	TIME TIME NOT NULL,
	NAME VARCHAR(100) NOT NULL,
	ADDESS VARCHAR(100) NOT NULL,
	PHONE VARCHAR(100) NOT NULL,
```

```
    CAR_ID VARCHAR(100) NOT NULL,
    DRIVER_ID VARCHAR(100) NOT NULL,
    STAFF_ID VARCHAR(100) NOT NULL,
    AMBULANCE_ID VARCHAR(100) NOT NULL,
    PRIMARY KEY (AMBULANCE_ID)
);


CREATE TABLE APPOINTMENT
(
    APPO_ID VARCHAR(100) NOT NULL,
    DATE DATE NOT NULL,
    NAME VARCHAR(100) NOT NULL,
    GENDER VARCHAR(100) NOT NULL,
    ADDRESS VARCHAR(100) NOT NULL,
    PHONE VARCHAR(100) NOT NULL,
    EMAIL VARCHAR(100) NOT NULL,
    STAFF_ID VARCHAR(100) NOT NULL,
    PRIMARY KEY (APPO_ID)
);


CREATE TABLE ATTENDENCE
(
    DATE DATE,
    ARRIVE_TIME TIME NOT NULL,
    STAFF_ID VARCHAR(100) NOT NULL
);


CREATE TABLE BIRTH_REPORT
(
    BIR_ID VARCHAR(100) NOT NULL,
    NAME VARCHAR(100) NOT NULL,
    DATE DATETIME NOT NULL,
```

```
    GENDER VARCHAR(10) NOT NULL,
    PARENT_NAME VARCHAR(100) NOT NULL,
    DOCTOR VARCHAR(100) NOT NULL,
    ADDRESS VARCHAR(100) NOT NULL,
    PRIMARY KEY (BIR_ID)
);


CREATE TABLE DEATH_REPORT
(
    DEAD_ID VARCHAR(100) NOT NULL,
    NAME VARCHAR(100) NOT NULL,
    AGE INTEGER NOT NULL,
    GENDER VARCHAR(100) NOT NULL,
    NRC_NO VARCHAR(100) NOT NULL,
    PARENT_NAME VARCHAR(100) NOT NULL,
    CON_DOCTOR VARCHAR(100) NOT NULL,
    DATE DATETIME NOT NULL,
    REASON VARCHAR(100) NOT NULL,
    PRIMARY KEY (DEAD_ID)
);


CREATE TABLE DOCTOR
(
    SPECIALITY VARCHAR(100) NOT NULL,
    STAFF_ID VARCHAR(100) NOT NULL
);


CREATE TABLE DRIVER
(
    DRIVER_ID VARCHAR(100) NOT NULL,
    DRIVER_NAME VARCHAR(100) NOT NULL,
    DRIVER_ADDRESS VARCHAR(100) NOT NULL,
```

```
        PHONE VARCHAR(100) NOT NULL,
        NRC VARCHAR(100) NOT NULL,
        PRIMARY KEY (DRIVER_ID)
);


CREATE TABLE DRUG_STOCK
(
        DRUG_ID VARCHAR(100) NOT NULL,
        DRUG_NAME VARCHAR(100) NOT NULL,
        DESCRIPTION VARCHAR(100) NOT NULL,
        CATEGORY VARCHAR(100) NOT NULL,
        PRICE INTEGER NOT NULL,
        COMPANY_NAME VARCHAR(100) NOT NULL,
        QTY INTEGER NOT NULL,
        DATE DATE NOT NULL,
        EXP_DATE DATE NOT NULL,
        PRIMARY KEY (DRUG_ID)
);


CREATE TABLE EQUIPMENT_STOCK
(
        EQUI_ID VARCHAR(100) NOT NULL,
        EQUI_NAME VARCHAR(100) NOT NULL,
        DATE DATE NOT NULL,
        PRICE VARCHAR(100) NOT NULL,
        QTY INTEGER NOT NULL,
        COMPANY_NAME VARCHAR(100) NOT NULL,
        PRIMARY KEY (EQUI_ID)
);


CREATE TABLE HR
(
```

```
        STAFF_ID VARCHAR(100) NOT NULL
);


CREATE TABLE LABORATORY_INFORMATION
(
        RECORD_DATE DATE NOT NULL,
        SAMPLE_RECORD VARCHAR(100) NOT NULL,
        STAFF_ID VARCHAR(100) NOT NULL,
        PATIENT_ID VARCHAR(100) NOT NULL,
        LAB_ID VARCHAR(100) NOT NULL,
        PRIMARY KEY (LAB_ID)
);


CREATE TABLE NURSE
(
        WARD_NAME VARCHAR(100) NOT NULL,
        STAFF_ID VARCHAR(100) NOT NULL,
        DEAD_ID VARCHAR(100) NOT NULL,
        BIR_ID VARCHAR(100) NOT NULL,
        DRUG_ID VARCHAR(100) NOT NULL,
        BED_ID VARCHAR(100) NOT NULL
);


CREATE TABLE OT_INFORMATION
(
        DATE DATE NOT NULL,
        STARTED_TIME TIME NOT NULL,
        FINISHED_TIME TIME NOT NULL,
        THEATER_NAME VARCHAR(100) NOT NULL,
        OPERATION_CONSENT_NAME VARCHAR(100) NOT NULL,
        LEAD_DOCTOR VARCHAR(100) NOT NULL,
        ASS_DOCTOR VARCHAR(100) NOT NULL,
```

```
        ANNASETHESIST VARCHAR(100) NOT NULL,
        NURSE VARCHAR(100) NOT NULL,
        STAFF_ID VARCHAR(100) NOT NULL,
        PATIENT_ID VARCHAR(100) NOT NULL,
        OPERATION_ID VARCHAR(100) NOT NULL,
        PRIMARY KEY (OPERATION_ID)
);


CREATE TABLE PATIENT
(
        PATIENT_ID VARCHAR(100) NOT NULL,
        ACCEPTED_DATE DATE NOT NULL,
        PASSWORD VARCHAR(100) NOT NULL,
        SICKNESS VARCHAR(100) NOT NULL,
        ALLERGIES VARCHAR(100) NOT NULL,
        PRESCRIPTION VARCHAR(100) NOT NULL,
        FAM_NAME VARCHAR(100) NOT NULL,
        FAM_NRC VARCHAR(100) NOT NULL,
        RECORD_ID VARCHAR(100) NOT NULL,
        APPO_ID VARCHAR(100) NOT NULL,
        REG_PATIENTID VARCHAR(100) NOT NULL,
        LEAVE_DATE DATE NOT NULL,
        PRIMARY KEY (PATIENT_ID)
);


CREATE TABLE PHARMACIST
(
        STAFF_ID VARCHAR(100) NOT NULL
);


CREATE TABLE RECEPTIONIST
(
```

```
    STAFF_ID VARCHAR(100) NOT NULL
);


CREATE TABLE RECORD
(
    RECORD_ID VARCHAR(100) NOT NULL,
    DATE DATE NOT NULL,
    PATIENT_CONDITION VARCHAR(100) NOT NULL,
    INSTRUCTION VARCHAR(100),
    DRUG_ID VARCHAR(100) NOT NULL,
    STAFF_ID VARCHAR(100) NOT NULL,
    PRIMARY KEY (RECORD_ID)
);


CREATE TABLE REGISTER
(
    REG_PATIENTID VARCHAR(100) NOT NULL,
    NAME VARCHAR(100) NOT NULL,
    BIRTH_DATE DATE NOT NULL,
    GENDER VARCHAR(100) NOT NULL,
    ADDRESS VARCHAR(100) NOT NULL,
    PHONE VARCHAR(100) NOT NULL,
    EMAIL VARCHAR(100) NOT NULL,
    NRC_NO VARCHAR(100) NOT NULL,
    PRIMARY KEY (REG_PATIENTID)
);


CREATE TABLE STAFF
(
    STAFF_ID VARCHAR(100) NOT NULL,
    NAME VARCHAR(100) NOT NULL,
    BIRTHDATE DATE NOT NULL,
```

```
	NRC VARCHAR(100) NOT NULL,
	GENDER VARCHAR(10) NOT NULL,
	ADDRESS VARCHAR(100) NOT NULL,
	PHONE VARCHAR(100) NOT NULL,
	EMAIL VARCHAR(100) NOT NULL,
	JOINED_DATE DATE NOT NULL,
	EDUCATION VARCHAR(100) NOT NULL,
	CERTIFICATION VARCHAR(100),
	SALARY INTEGER NOT NULL,
	PASSWORD VARCHAR(100) NOT NULL,
	RANK VARCHAR(100) NOT NULL,
	TYPE VARCHAR(100) NOT NULL,
	PRIMARY KEY (STAFF_ID)
);


CREATE TABLE WARD_ROOM
(
	BED_ID VARCHAR(100) NOT NULL,
	WARD_NAME VARCHAR(100) NOT NULL,
	ROOM_NUMBER VARCHAR(100) NOT NULL,
	PATIENT_ID VARCHAR(100) NOT NULL,
	PRIMARY KEY (BED_ID)
);



/* Create Foreign Keys */

ALTER TABLE AMBULANCE_INFORMATION
	ADD FOREIGN KEY (CAR_ID)
	REFERENCES AMBULANCE (CAR_ID)
	ON UPDATE RESTRICT
	ON DELETE RESTRICT
;
```

```
ALTER TABLE ADMIN
        ADD FOREIGN KEY (AMBULANCE_ID)
        REFERENCES AMBULANCE_INFORMATION (AMBULANCE_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE PATIENT
        ADD FOREIGN KEY (APPO_ID)
        REFERENCES APPOINTMENT (APPO_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE NURSE
        ADD FOREIGN KEY (BIR_ID)
        REFERENCES BIRTH_REPORT (BIR_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE NURSE
        ADD FOREIGN KEY (DEAD_ID)
        REFERENCES DEATH_REPORT (DEAD_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE AMBULANCE_INFORMATION
```

```
        ADD FOREIGN KEY (DRIVER_ID)
        REFERENCES DRIVER (DRIVER_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE NURSE
        ADD FOREIGN KEY (DRUG_ID)
        REFERENCES DRUG_STOCK (DRUG_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE RECORD
        ADD FOREIGN KEY (DRUG_ID)
        REFERENCES DRUG_STOCK (DRUG_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE ADMIN
        ADD FOREIGN KEY (EQUI_ID)
        REFERENCES EQUIPMENT_STOCK (EQUI_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE ADMIN
        ADD FOREIGN KEY (PATIENT_ID)
        REFERENCES PATIENT (PATIENT_ID)
        ON UPDATE RESTRICT
```

```
        ON DELETE RESTRICT
;


ALTER TABLE LABORATORY_INFORMATION
        ADD FOREIGN KEY (PATIENT_ID)
        REFERENCES PATIENT (PATIENT_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE OT_INFORMATION
        ADD FOREIGN KEY (PATIENT_ID)
        REFERENCES PATIENT (PATIENT_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE WARD_ROOM
        ADD FOREIGN KEY (PATIENT_ID)
        REFERENCES PATIENT (PATIENT_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE PATIENT
        ADD FOREIGN KEY (RECORD_ID)
        REFERENCES RECORD (RECORD_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;
```

```
ALTER TABLE PATIENT
        ADD FOREIGN KEY (REG_PATIENTID)
        REFERENCES REGISTER (REG_PATIENTID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE ADMIN
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE AMBULANCE_INFORMATION
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE APPOINTMENT
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE ATTENDENCE
        ADD FOREIGN KEY (STAFF_ID)
```

```
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE DOCTOR
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE HR
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE LABORATORY_INFORMATION
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;

ALTER TABLE NURSE
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
```

```
;


ALTER TABLE OT_INFORMATION
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE PHARMACIST
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE RECEPTIONIST
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;


ALTER TABLE RECORD
        ADD FOREIGN KEY (STAFF_ID)
        REFERENCES STAFF (STAFF_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;
```

```
ALTER TABLE NURSE
        ADD FOREIGN KEY (BED_ID)
        REFERENCES WARD_ROOM (BED_ID)
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
;
```