

REPUBLIC OF ALBANIA

FACULTY OF COMPUTER SCIENCES AND IT

Software Engineering Program

**Software Analysis and Design**

Accepts: Igli Hakrama Presenting: Dervin Canta, Keti Stafa, Darling Selita

1 Executive Summary

1.1 Project Summary.

With the large amounts of data, people involved and innumerable processes, a hospital is definitely an ideal candidate for data management software. If hospitals are to run efficiently, provide top line care, ensure patient and other data confidentiality, and work seamlessly – they cannot hope to do so without an effective [Hospital Management System Software](https://www.karexpert.com/hospital-information-management-system?utm_source=Website%20Organic&utm_medium=https://www.google.com/&referer=https://www.google.com/&origin_referer=https://www.google.com/). Reduced human intervention for paperwork, less paperwork, reduced staff headcount for jobs that can be easily managed within the HMS, speedier processes, reduction of errors, and data privacy and safety – are just some of the benefits of what we are offering throughout our web based application “Hospiware”.

1.2 Purpose and Scope of this Specification.

For the hospitals, HMS translates to being able to track patient history, provide better care, keep track of appointments, save patient insurance and payment data, enable doctors and clinicians to check patient history, maintain patient care continuity, and save time and effort on unnecessary tedious manual tasks. This [Electronic Medical Record (EMR)](https://www.karexpert.com/saas/ehr-emr-software/?utm_source=Website%20Organic&utm_medium=https://www.google.com/&referer=https://www.google.com/&origin_referer=https://www.google.com/) or Electronic Health Record (EHR) is the journey of a patient with the hospital – keeping track of the date of every visit, doctor consulted, medicines and advice prescribed, and other information for the patient. This ensures that even if a patient

visits after a long break, the patient and hospital will not require going through the registration process again.

2 Product/Service Description.

2.1 Product Context.

Hospiware is a hospital-only dedicated system which can be of use in every department/sector of a hospital, be that the Neurology department, Medicine department, Surgery Department, Plastic Surgery, Infectious Disease department ect. It makes it easier for doctors and surgeons to know the patient’s medical history beforehand and helps them deflect prescriptions or drugs that a patient’s allergic to.

2.2 User Characteristics.

System Administrator(ADMIN).

Update or change the system based on user’s request, manage permissions for users, create or delete user accounts, approve the deletion of patient’s data.

Receptionist.

Schedule appointments, check doctor’s and surgeon’s timetable, add new patient files by searching them up in the database, alert the nurse or a doctor about a patient coming from the emergency room, keep track of financial costs for every patient in the hospital.

Nurse.

Check future appointments, alter the medical file of a patient, retrieve medical file of a patient from database.

Doctor.

Delay, remove or schedule a future appointment, give medication to a patient, alter the medical file of a patient.

Surgeon.

Delay, remove or schedule a future surgery, check patient’s medical file prime to the surgery, add notes to the database inquiring the surgery (usually surgeries don’t always go the way they’re supposed to, so surgeons can write the complicalities and how they dealt with them in order for other surgeons to learn from them and be ready when a similar complication happens on their surgeries).

Head of Department.

Assign patient cases to doctors/ surgeons, send an automated system message regarding his decisions to the users.

HR Agent.

Accept complaints from patients and internal hospital staff, write a report regarding said problem, send it to Head of Department for the final decision.

Patients.

Create accounts, fill out online form to make an appointment, send a complaint to HR.

2.3 Assumptions.

It is assumed that every user that interacts with the system has the proper training, computer skills and professional knowledge to interact with the system and be able to use its features.

It is assumed that every user has a phone/tablet/desktop which has a stable internet connection.

It is assumed that every employee in the hospital has his/her own account created by the system admin.

It is assumed that appointments are made from patients by filling out a form online.

It is assumed that the hospital has a working bank account to distribute all paychecks of its employees.

It is assumed that the hospital has a template document for the patients who want to make an appointment.

It is assumed that the hospital directors have email accounts for the employees on the following email providers: Google, Microsoft.

It is assumed that the hospital has the legal means to protect hospital staff in the case of lawsuit.

2.4 Constrictions.

The system can only be accessed through a stable internet connection.

The system can only be integrated with a limited number of third-party applications

The system is designed in different sections where each user which has access to the corresponding section should have prior knowledge or training to systems of this kind.

The system does not allow the users to make customizations regarding the system’s main components in order to prevent anomalies.

The system has assigned roles for every type of user so not all users have the same accessibility to it.

2.5 Dependencies.

The Secretary should alert any available doctors in case a patient comes from the emergency room.

The Secretary should send every new case to the Head of Department in order for him/her to make the decisions.

The nurse should always access the database and retrieve a patient’s medical file when he comes from the emergency room.

The Doctors and Surgeons should make sure to clear their schedule before accepting a new patient case incase of an overlap.

The Head of Department should consider very carefully when assigning cases based on their severity.

The Doctors and Surgeons should carefully edit the patient’s file in case there has been a change of his conditions.

2.5 Legislation Constrictions.

Health Insurance Portability and Accountability Act (HIPAA): This law sets standards for protecting the privacy and security of patients' health information, including electronic health records (EHRs).

Food and Drug Administration (FDA): The FDA regulates medical devices, including software, that are used in the diagnosis, treatment, or prevention of medical conditions.

European Union Medical Devices Regulation (MDR): This regulation sets the requirements for the design, manufacture, and performance of medical devices in the European Union.

Clinical Laboratory Improvement Amendments (CLIA): CLIA sets standards for laboratory testing and regulatesthe use of software in clinical laboratory settings.

General Data Protection Regulation (GDPR): This regulation governs data privacy and protection for individuals in the European Union.

Interoperability: The software should be able to easily exchange information with other healthcare systems and software, such as electronic health record (EHR) systems and laboratory information systems.

User interface and experience: The software should be intuitive and user-friendly for healthcare providers, patients, and administrators.

Data security and privacy: The software should comply with regulations such as HIPAA and GDPR, and have robust security measures in place to protect patients' personal and health information.

Scalability: The software should be able to handle increasing amounts of data and users as the hospital grows.

Clinical decision support: The software should provide clinical decision support to healthcare providers, such as drug

interaction warnings, treatment guidelines, and alerts for abnormal test results.

Analytics and reporting: The software should provide real-time data analytics and customizable reports to help hospital administrators make informed decisions.

Integration with other systems: The software should be able to integrate with other systems used in the hospital, such as billing and scheduling systems, to provide a seamless experience for users.

3. Functional and Non-Functional Requirements.

3.1 Functional requirements of Outpatient.

The Hospital Management system project must be able to handle Patient Registration

The Hospital Management system project must be able to handle Patient’s complete lab tests history

The authentic registered user can view, delete, update, insert and search Thank you (Get well soon), text message once patient is discharged.

Patient’s previous visits including doctor’s detail with time stamp

The authentic registered user can view, delete, update, insert and search Patient’s complete radiology history.

The authentic registered user can view, delete, update, insert and search Patient’s allergy detail

The Hospital Management system project must be able to handle Patient Queue System

The Hospital Management system project must be able to handle Doctor’s view of waiting patients

The Hospital Management system project must be able to handle Patient’s complete medical history

Calculate each time a doctor serves a patient so payment can be made based on number of patients served.

3.1.1 Functional requirements of Inpatient

The authentic registered user can view, delete, update, insert and search Prescription

The authentic registered user can view, delete, update, insert and search Test request

The authentic registered user can view, delete, update, insert and search Doctor’s note including previous notes

The authentic registered user can update, insert, view, delete, and search Discharge summary notes

The Hospital Management system project must be able to handle Receipt

The authentic registered user can update, insert, view, delete, and search Integrated Pharmacy

The authentic registered user can update, insert, view, delete, and search Integrated laboratory

The authentic registered user can update, insert, view, delete, and search Integrated Radiology

The Hospital Management system project must be able to handle Billing

The authentic registered user can update, insert, view, delete, and search Another Doctor’s call management (for Inpatient)

The authentic registered user can update, insert, view, delete, and search Patient Discharge

The Hospital Management system project must be able to handle Patient Admission System

Ward Management System

The authentic registered user can view, delete, update, insert and search Bed Management System

The Hospital Management system project must be able to handle Doctor’s visit to ward management

The Hospital Management system project must be able to handle Doctor’s visit detail including doctor’s detail, ward detail with time stamp

The Hospital Management system project must be able to handle Doctor’s visits added to total number of visits so doctor could be paid based on number of patients visited/served

The Hospital Management system project must be able to handle Patient Transfer/move management

The authentic registered user can update, insert, view, delete, and search Comprehensive patient search within each ward

3.1.2 Functional requirements of Operation Theatre Management System

The authentic registered user can view, delete, update, insert and search Check relevant surgeon’s availability

The authentic registered user can view, delete, update, insert and search Schedule an Operation

The authentic registered user can view, delete, update, insert and search Check anaesthetist availability

The authentic registered user can view, delete, update, insert and search Define type of operation.

3.2 Non-Functional Requirements.

3.2.1 Maintainability

The Hospital Management System must have high level of Maintainability.

3.2.2 Serviceability

If issue arises in the Hospital Management System, then the project must be programmed in such a way that developer can service it again.

3.2.3Environmental

The Hospital Management System must be working in latest operating system environments like windows 7, windows 8, windows 10 and on linux.

3.2.4 Data Integrity

All the data in the Hospital Management System must be accurate and reliable.

3.2.5 Usability

The Hospital Management System must have a good looing user friendly interface.

3.2.6 Recoverability

The Hospital Management System must have a proper data backup mechanism.

3.2.7 Interoperability

The Hospital Management System must work with or use the parts or equipment of another system.

3.2.8 Capacity

The Hospital Management System must fulfill on storage requirements, today and in the future. The Hospital Management System must be scale up for increasing volume demands.

3.2.9 Performance

The Hospital Management System must perform well in different scenarios.

3.2.10 Security

The Hospital Management System must be secured with proper user name and passwords.

3.2.11 Regulatory

The Hospital Management System must obey all the governmental requirements and  constraints.

3.2.12 Availability

The Hospital Management System must be available 24 hours a day with no bandwidth issues.

3.2.13 Manageability

The Hospital Management System must Alerts when the system suffers from a recoverable interruption.

4. User Cases .

4.1 Scenario Title: User logs in the system.

The user opens the application on the tablet/PC.

Puts in the credentials which he/she is registered with.

The system verifies these credentials.

The user is logged in the system.

4.2 Scenario Title: User logs in with the wrong credentials.

The user opens the application on the tablet/PC.

Writes in the credentials needed.

The system checks for these credentials and finds nothing.

The user is prompted with a warning to rewrite the credentials or check with the admin for a new account if he/she isn’t registered.

4.3 Scenario Title: The admin wants to put in a new user ( A new surgeon, nurse or receptionist).

The admin opens the system administrative tool.

Creates a new account with a temporary password

Gives it to the new user to have as his/her own account and tells him/her to change the password.

4.4 Scenario Title: A patient comes in from the emergency room.

User connects to a large database containing medical files about every person in the country.

He identifies the patient by their personal ID.

The system checks for the file related to said ID and brings out detailed medical information.

The user registers him/her as a new patient.

4.5 Scenario Title: A nurse/medic (user) wants to find the best way to treat a patient coming from the emergency room.

The user primarily logged in and put in the credentials of the patient to pull the medical information.

The user uses the AI by inputting every medical constriction and the problem at hand

The system generates the best way to treat the patient.

4.6 Scenario Title: The user wants to register a new appointment for a returning patient

The user logs in the system using the credentials.

The user checks to see if a doctor is available on a certain date.

The user responds to said patient telling him/her the date of the appointment.

4.7 Scenario Title: A returning patient comes for a routine checkup

The user logs in using the credentials.

The user checks the system using the returning patient’s ID credentials.

The user checks to see if the date of the appointment is in the system.

The user refreshes the system if the appointment date is found.

4.8 Scenario Title: The user wants to edit/fix an error in a patient’s file.

The user opens the application using the credentials.

He finds the said patient’s file.

Presses the buton labeled edit.

Then makes the changes.

The system automatically saves the changes after pressing the button “Save Changes”.

4.9 Scenario Title: A patient wants to schedule a surgery with one of the surgeons.

The user logs in using the credentials.

Then, checks the calendar for a surgeon.

Informs the patient on a date available.

Fills the date in the system which prompts a message in the other user’s account ( the surgeon) that they have a scheduled surgery in “X” date.

4.10 Scenario Title: The user wants to see the patient’s surgery history. (In this case the user is a surgeon)

The user opens the database.

Searches up the patient’s file by using his ID credentials.

Asks the system to generates any kind of surgery done primar to the patient’s arrival at the hospital.

4.11 Scenario Title: The user wants to add an operation procedure to a patient’s medical record.

The user asks the system to generate the patient’s file from the database.

The system promptly generates the file and gives the user the possibility to edit the medical records.

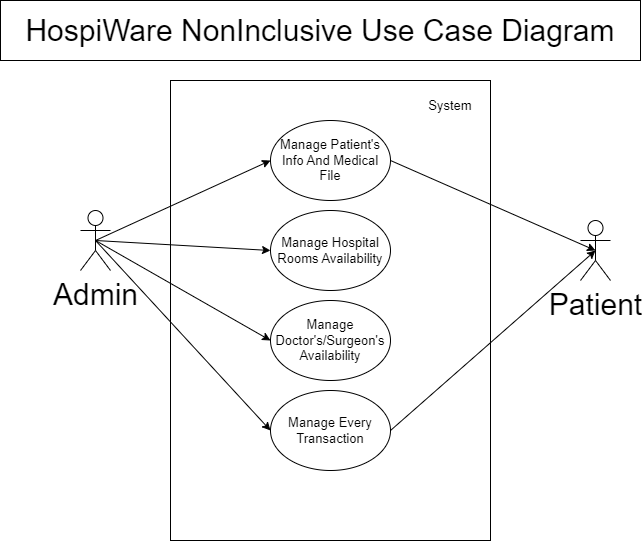
The user presses the button labeled “Edit” and after being done presses the button labeled “Save Changes”.

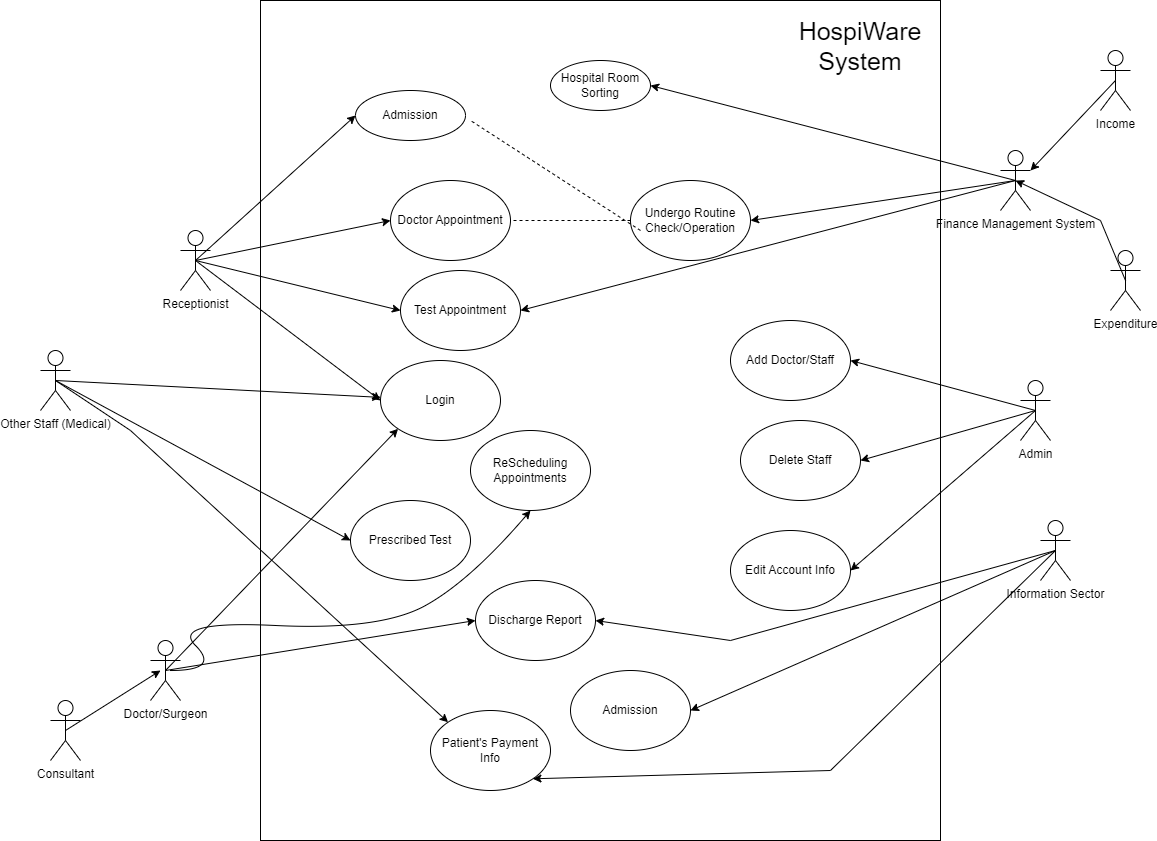
4.12 Scenario Title: The user wants to add a new patient file.

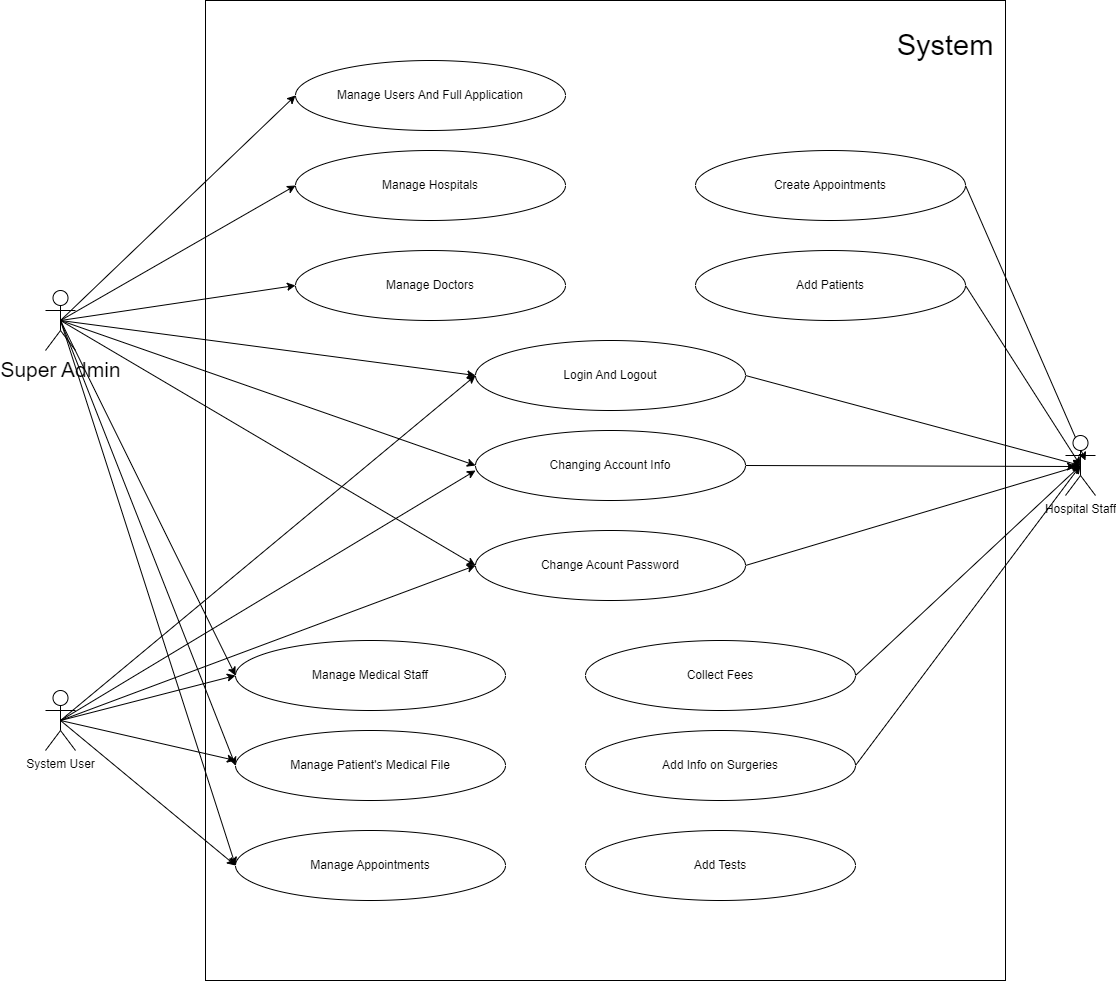
The user opens the application by using the credentials to log in.

Asks for special permission from the admin to add a new file to the database.

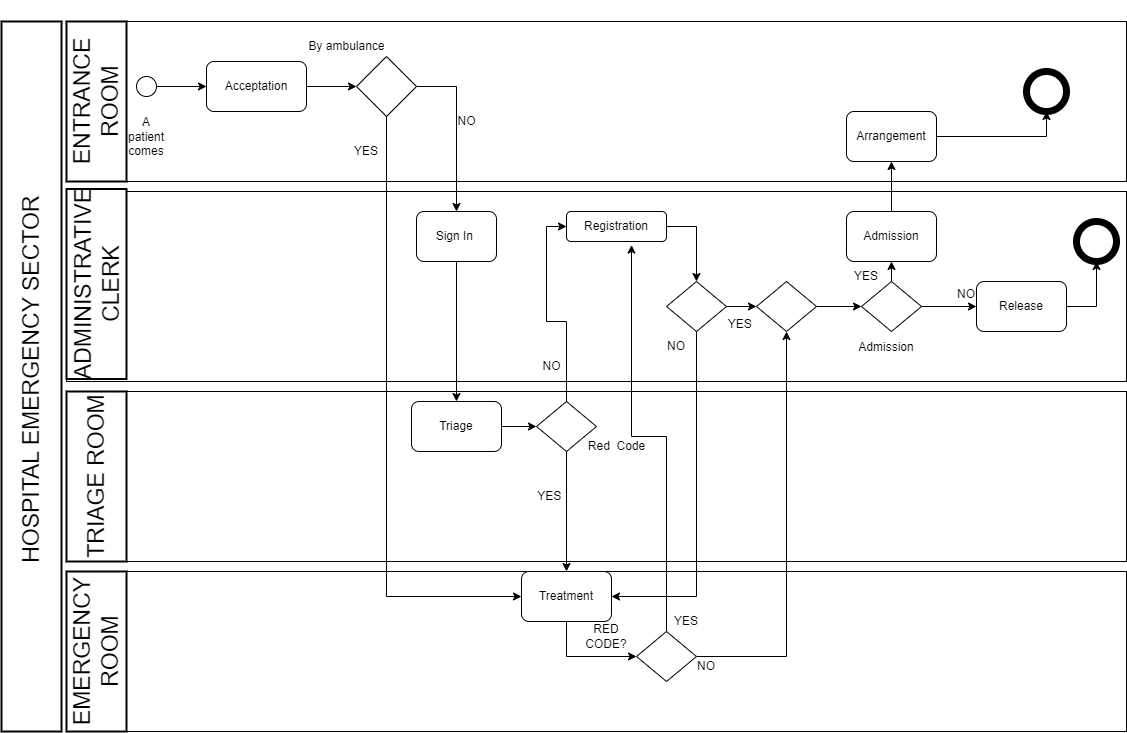
Starts to fill out a new file for the new patient.

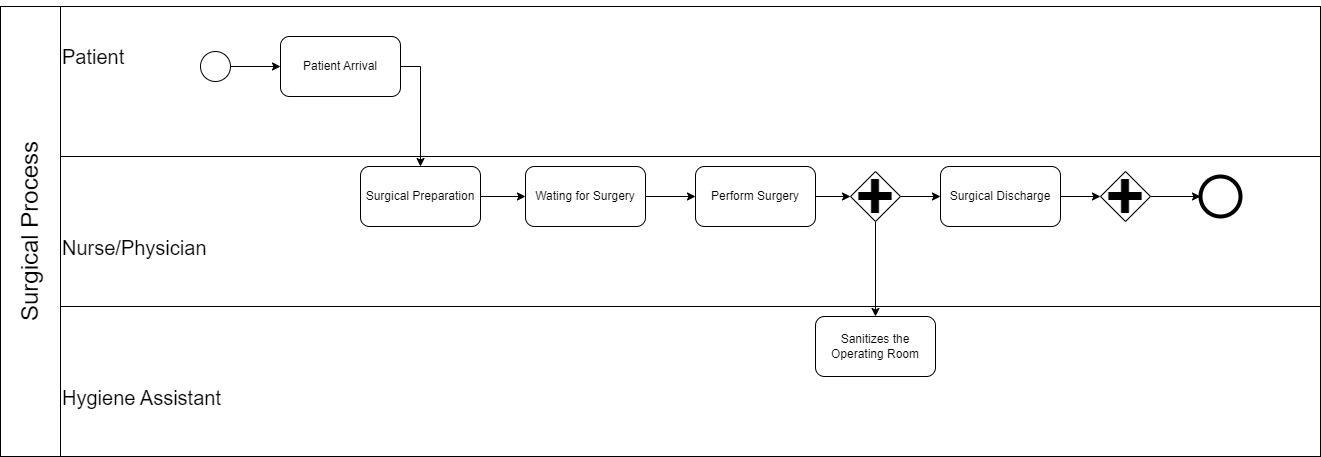
5. Use Case Scenarios

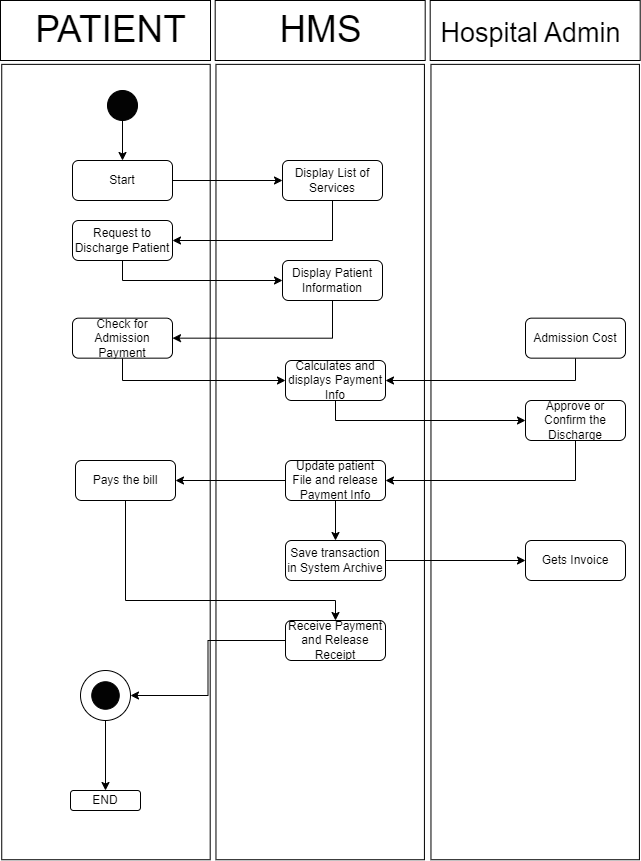




6. BPMN

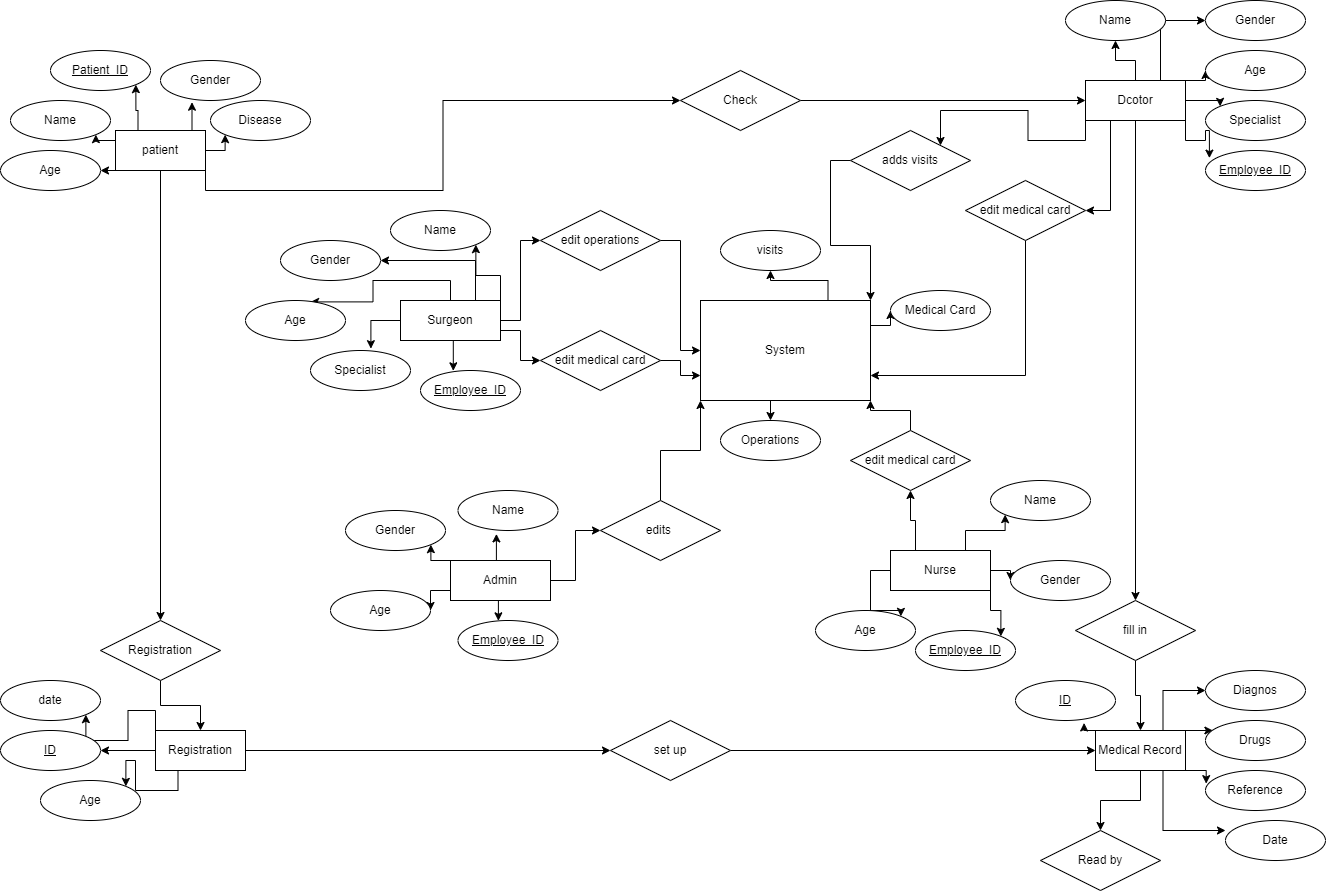


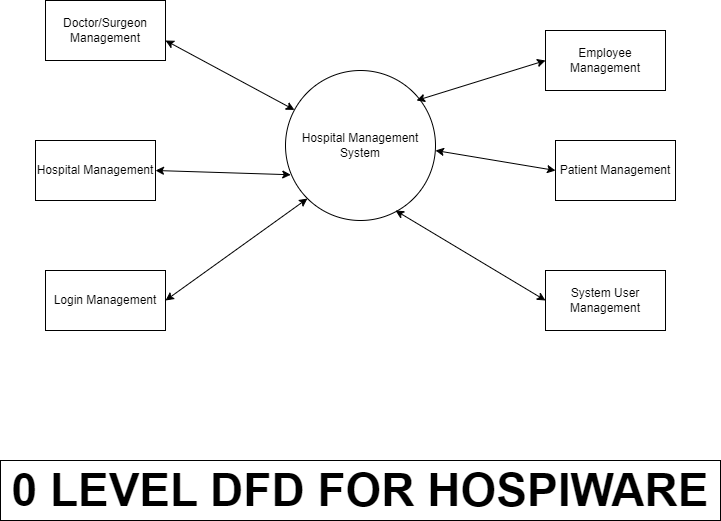


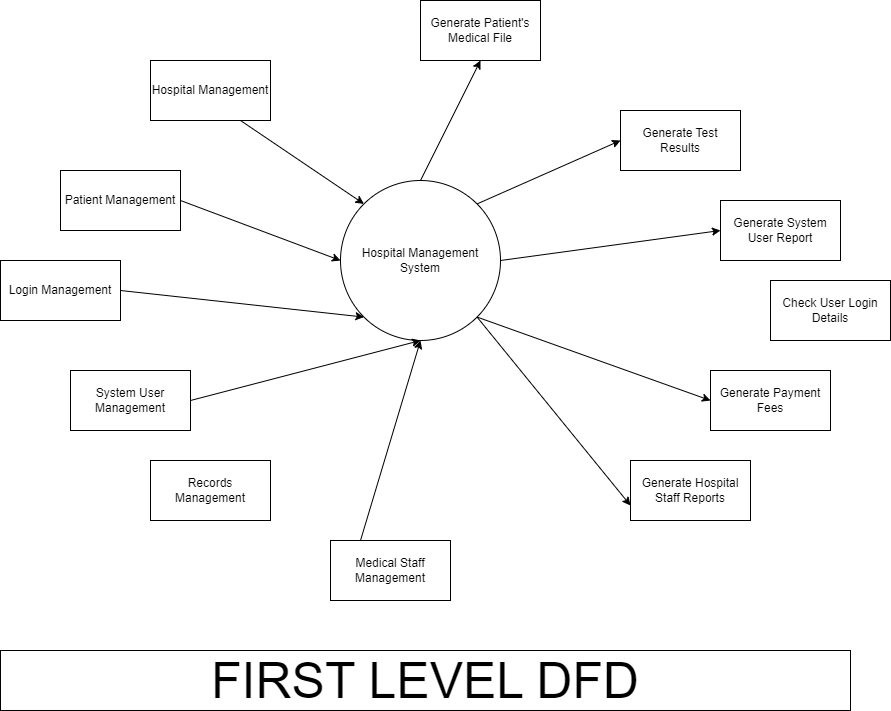


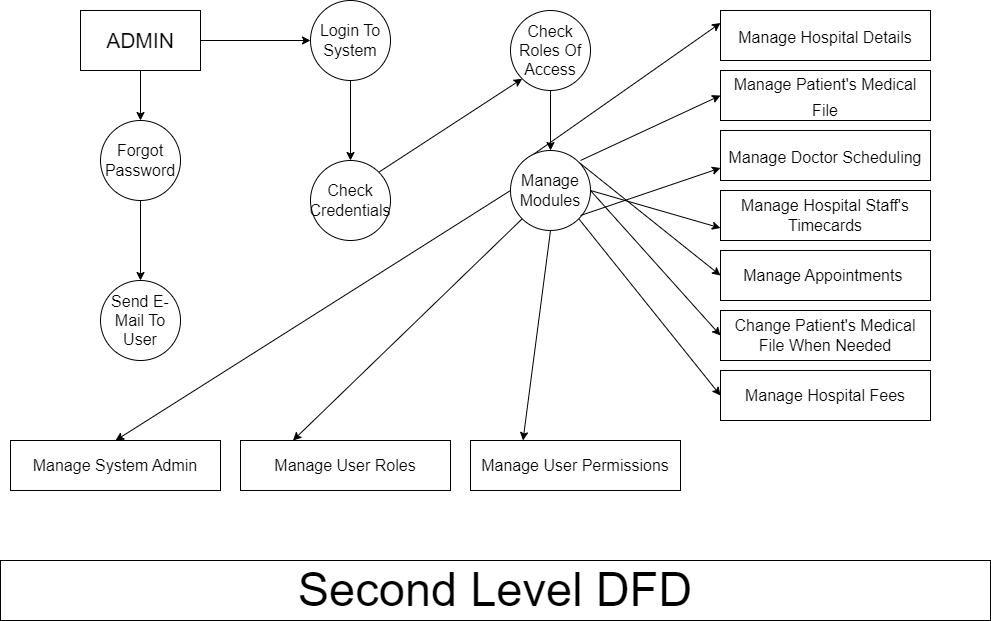
7. Database Diagrams

7.1 EFD.



7.2 DFD (0,1,2)

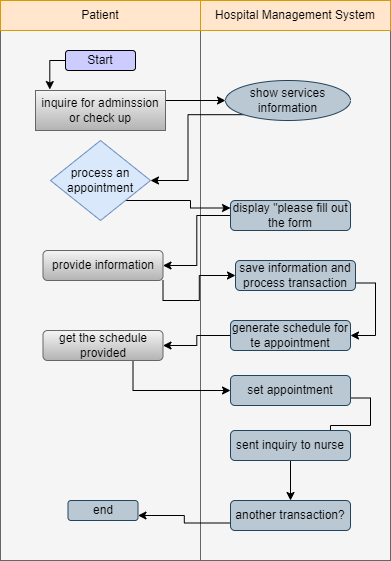


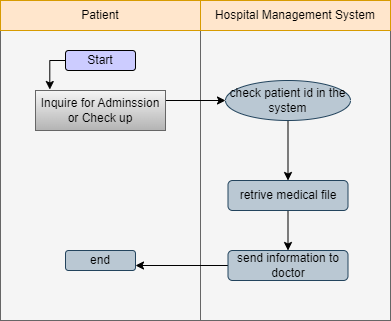


`

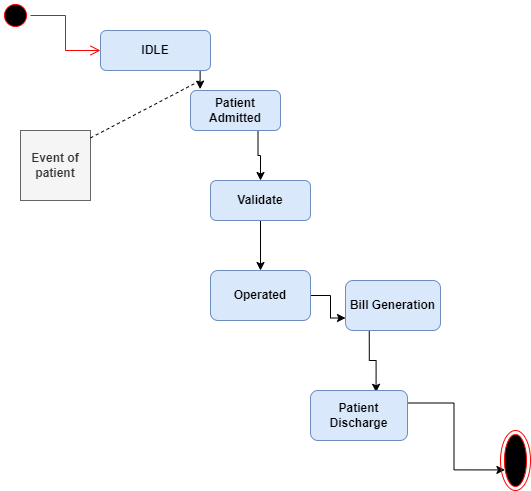
8. UML Diagrams

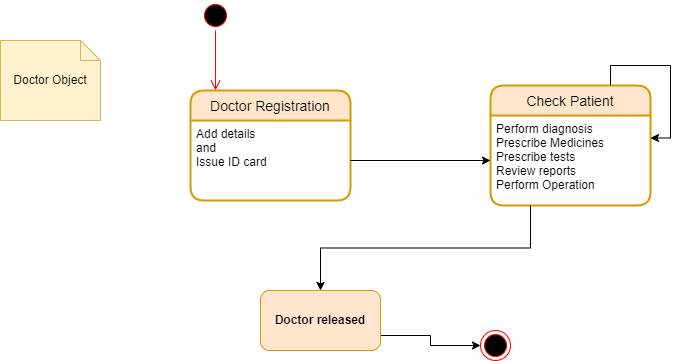
8.1 Activity Diagrams.



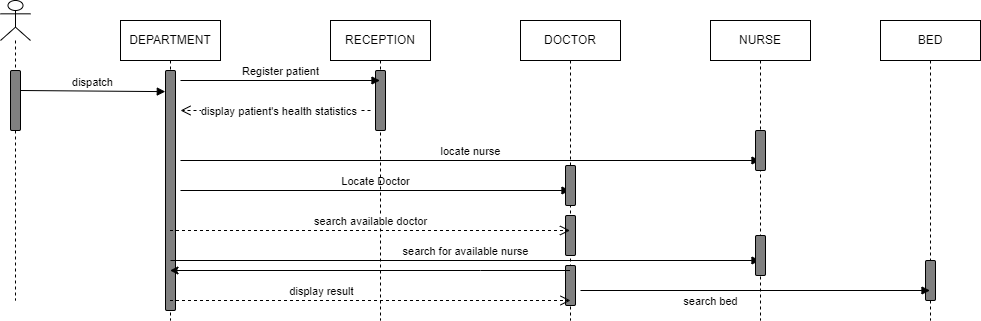


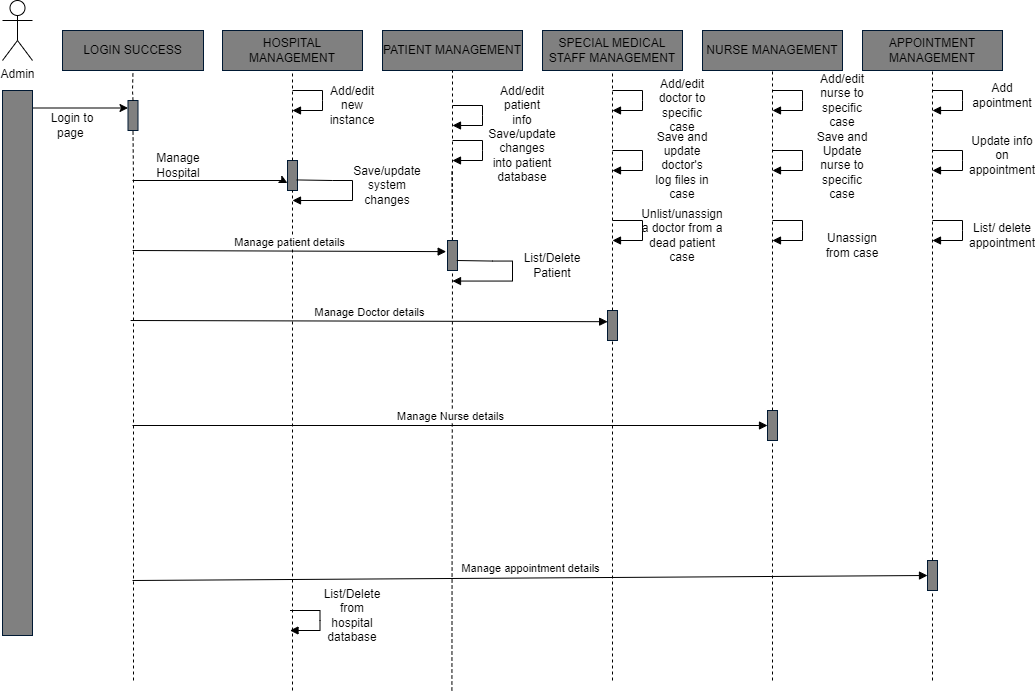
8.2 State Diagrams.



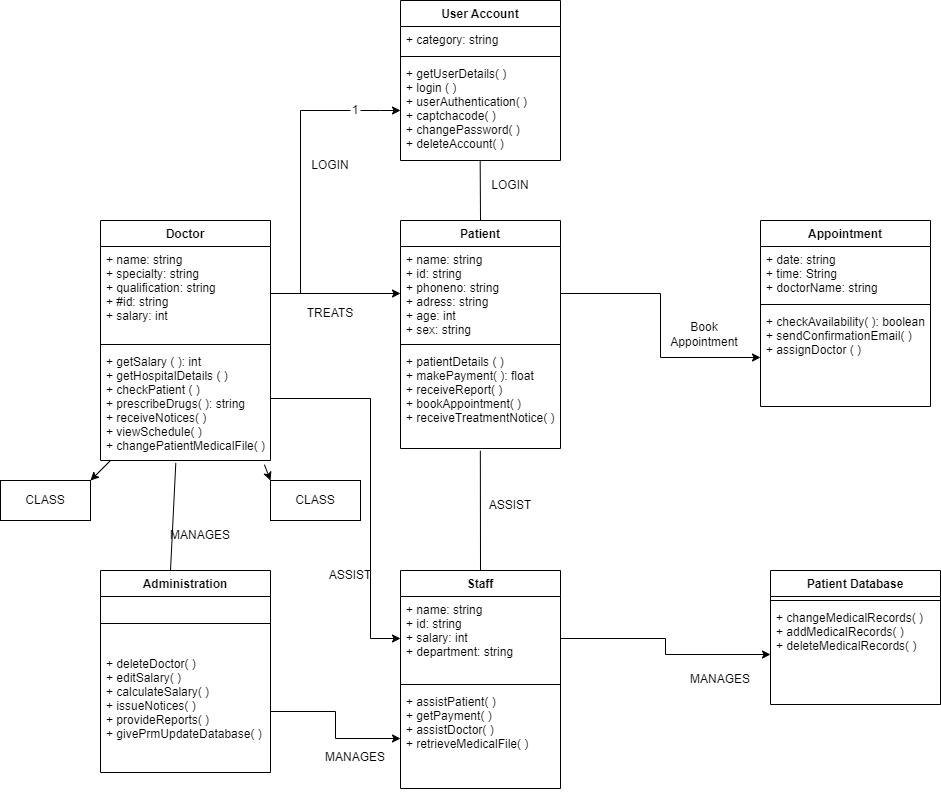


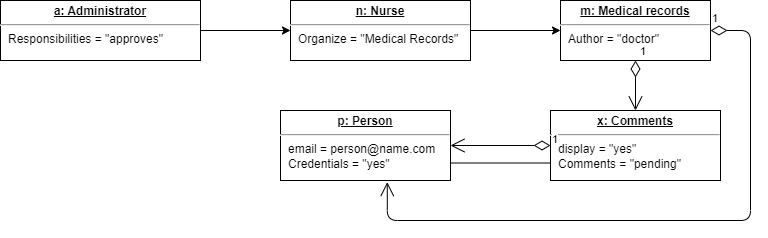
8.3 Sequence Diagrams.



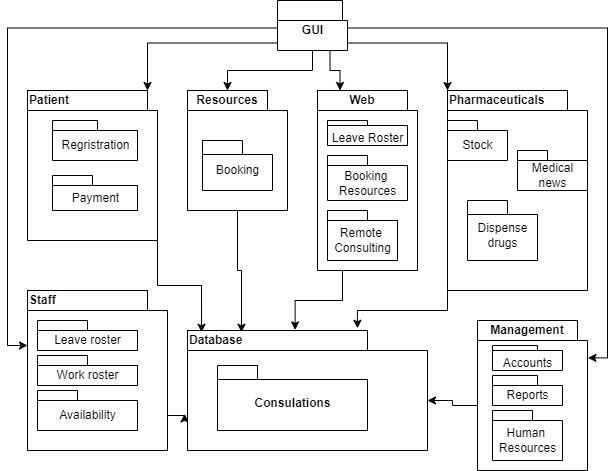


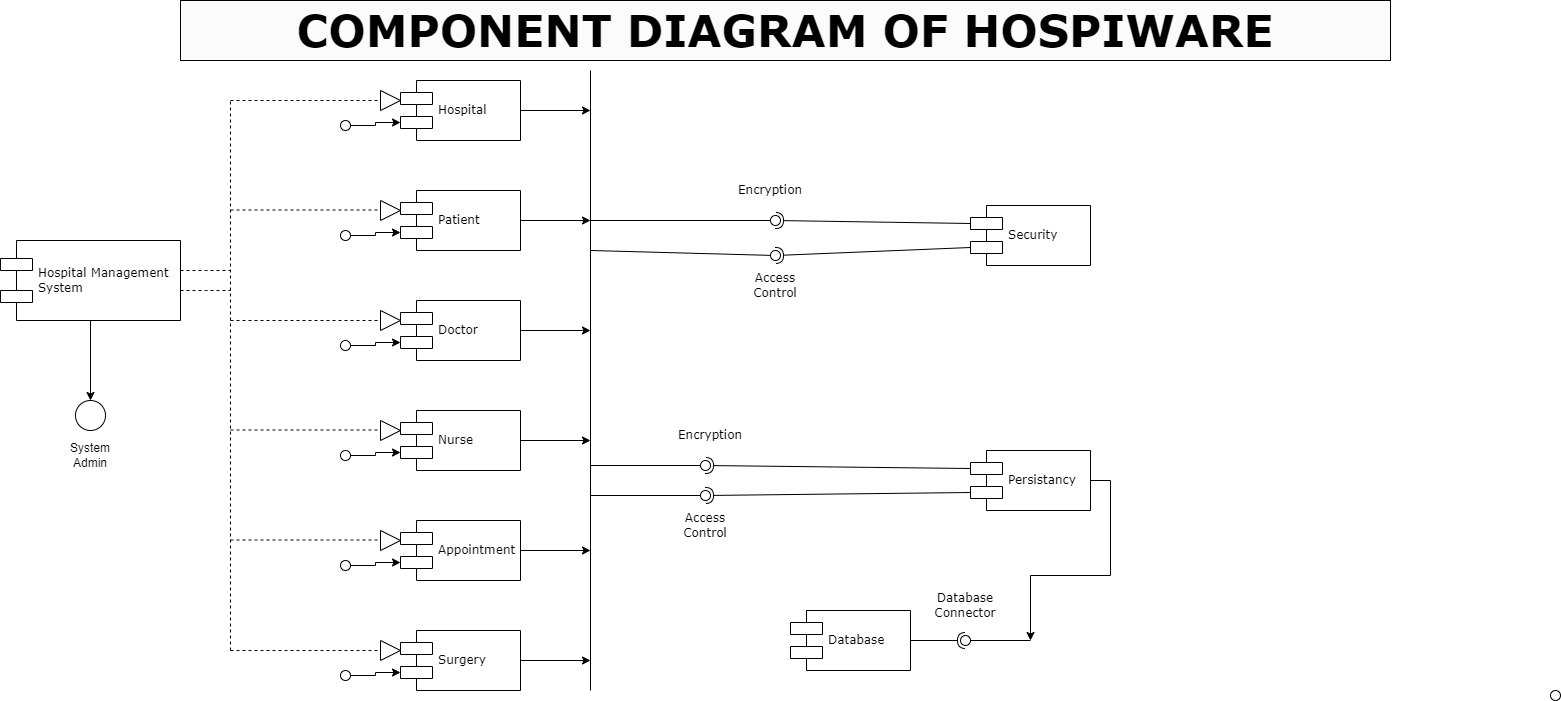
8.4 Class Diagram.



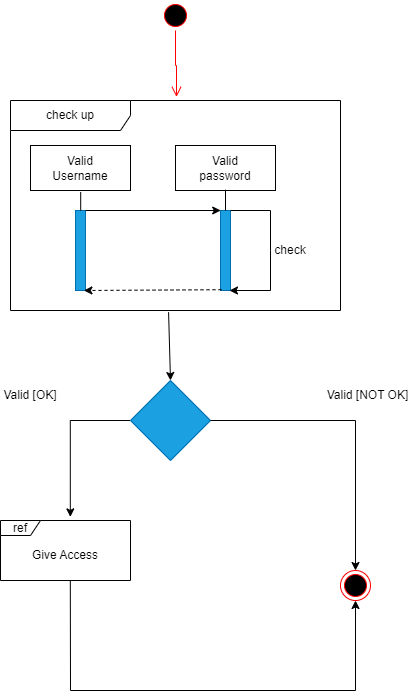
8.5 Object Diagram.

8.6 Package Diagram.

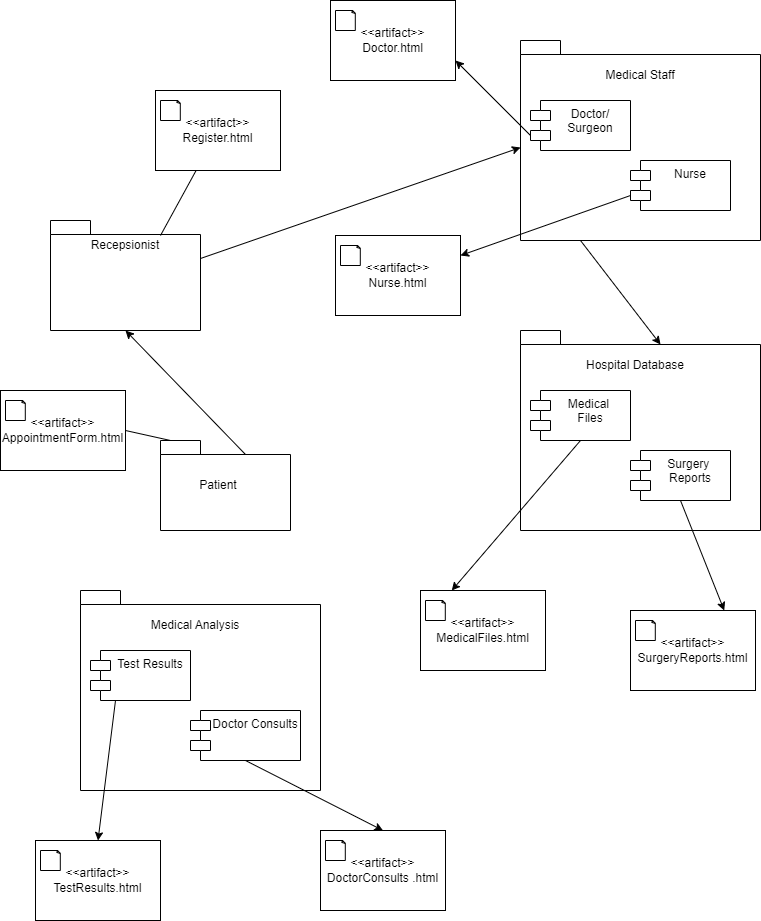


8.7 Component Diagram.

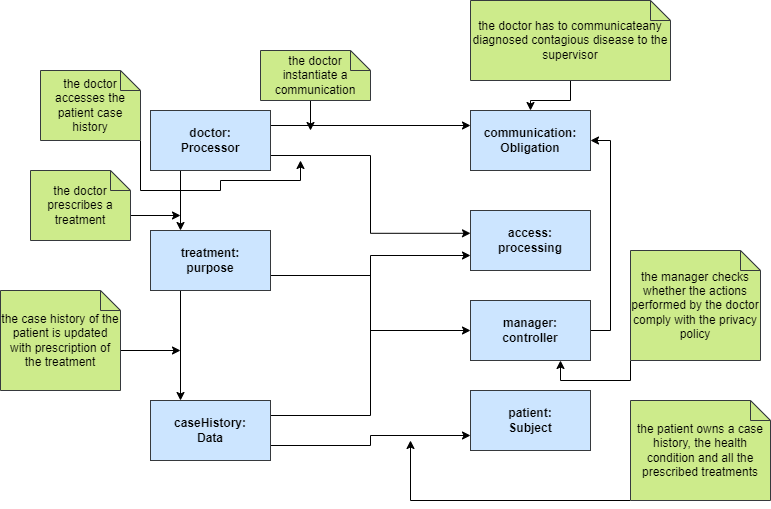
8.8 interaction Overview Diagram.



8.9 Deployment Diagram.

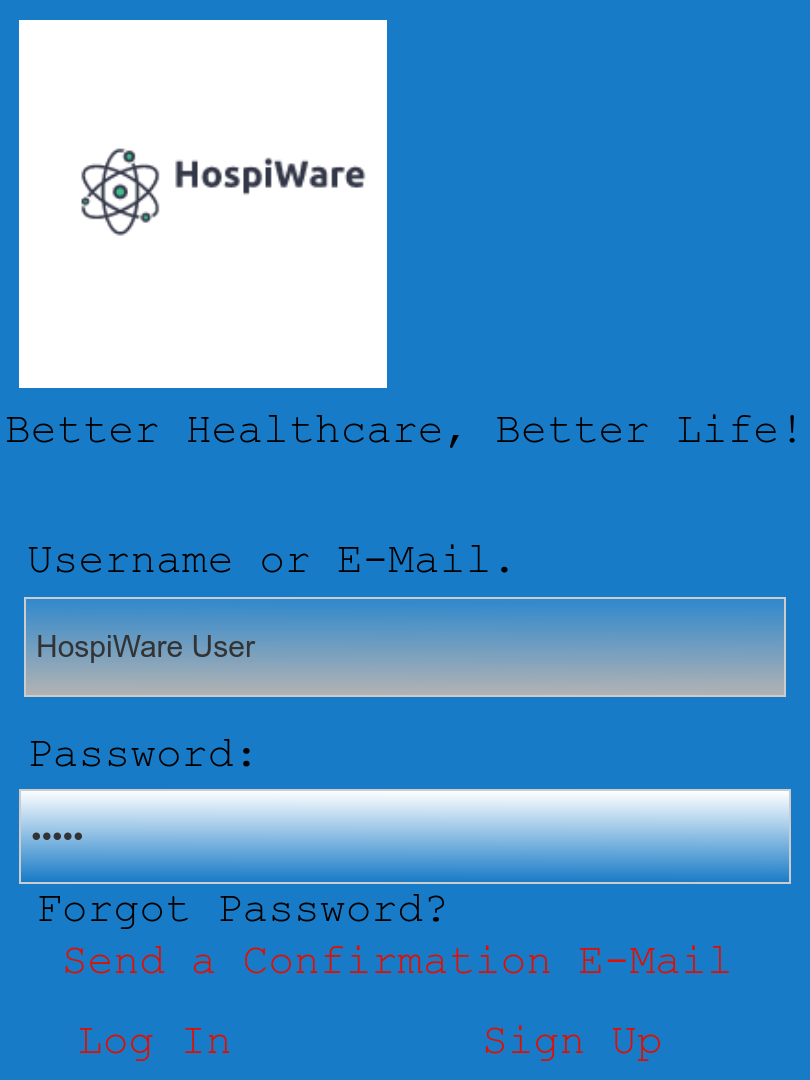


8.10 Composite Structure Diagram.



9. User Interface.

In user interface we start to get an Idea about how our product is going to look like when being used by its users as per the example below.



-WIKI’s.

**Week 1.**

The first week was an introduction to the project, after the group was formed we started brainstorming for ideas the project. We came up with a lot of good ones but the one that stuck was the Hospital Management System. We also decided to give it the name HospiWare because the user should always be aware of his health.

**Week 2.**

The second week we were introduced to Functional Requirements. Therefore, we started interviewing future users of our system. We started to gather data from different users and different experiences and came up with a set of functional requirements which would satisfy most of the user's needs. We also started on writing a project summary and scope.

**Week 3.**

In this week we learned about Non-Functional Requirements. They include Safety, Legal Constrictions, Maintainability, Efficiency.

**Week 4.**

Now we started working on Use cases and use case diagrams. Use cases include the following as: Logging in the system, Changing Credentials, Creating new accounts, Creating new instances etc.

**Week 5.**

At this point we learned about the first type of diagrams, BPMN's. We started building three different diagrams in order to better conclude how our system is built.

**Week 6.**

In week 6 we learned about ERD (Entity Relationship Diagram) and DFD (Data Flow Diagram). Me and my group members sketched a primitive ERD in order for us to be more flexible. We did the same for DFD's.

**Week 7.**

In this week we were introduced to activity and sequence diagrams. We divided our work and worked together to come up with both types of diagrams.

**Week 8.**

In this week divided our jobs because there was a lot of new information to process. We each had 2 types of diagrams to do because of the workload.

**Week 9.**

We started on finishing and detailing the last parts of the project. We got 3 new types of diagrams which we divided them one by one and finished them, also adding them to our final work product. We also started working on our user interface to make sure the user has a clear idea of what our end product is going to look when developed into code.

**Week 10.**

This is the final week and also the most important one. We made the final part of the documentation and compiled the document in Microsoft Word. We also gave a finishing hand to the UI prototypes using proto.io and reference images.