SOFTWARE

REQUIREMENTS SPECIFICATION

**For**

**Hospital Appointment Scheduler**

**Prepared by:-**

*LAXMAN P*

*LINGESHKUMAR P*

*KISHORE KUMAR K*

# Introduction

## Purpose

The primary objective of the Hospital Appointment Scheduler is to streamline and optimize the appointment booking process within the healthcare facility. This software solution aims to enhance overall operational efficiency, improve patient experience, and facilitate better resource management. By automating appointment scheduling and related tasks, the system intends to minimize scheduling conflicts, reduce waiting times, and ensure that healthcare providers can effectively manage their appointments. The Hospital Appointment Scheduler is designed to provide a user-friendly interface for both healthcare staff and patients, promoting accessibility, accuracy, and transparency in the appointment scheduling workflow. Additionally, the system aims to integrate seamlessly with existing hospital management systems, promoting data integrity and interoperability across various departments. Ultimately, the purpose of the Hospital Appointment Scheduler is to contribute to the overall enhancement of patient care and organizational efficiency within the healthcare institution.

This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

## Document Conventions

* + - Entire document should be justified.
    - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

The scope of development for the Java project includes creating a web-based system that allows patients to schedule appointments with doctors, access their medical records, and receive notifications and reminders. Additionally, the system should enable doctors to manage their schedules efficiently.

For the appointment scheduling feature, consider implementing a user-friendly interface for patients to request appointments and for doctors to confirm or reschedule them. Utilize a database to store appointment details and ensure data security.

Medical record access and management require a secure login system with role-based access control. Implement functionalities for patients to view their medical history and for doctors to update and manage patient records securely.

Notification and reminders can be achieved through email or SMS integration, sending alerts for upcoming appointments, changes, or new medical records added.

For understanding the importance of schedule management, conduct user interviews or surveys with both doctors and patients to gather their scheduling needs, preferences, and pain points. This insight will guide the system's design to cater to their specific requirements.

Access control for medical records and appointment management involves setting permissions based on user roles (patient, doctor, admin). Ensure that sensitive data is encrypted and accessible only to authorized individuals.

The development scope should encompass these elements, focusing on user experience, data security, and effective communication between patients and doctors while addressing the requirements gathered during the initial phase.

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

* + - Books

 Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson

Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers

Et.al, Osman, M. H. "Ambi Detect: An Ambiguous Software Requirements Specification Detection Tool." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, no. 3 (April 10, 2021): 2023–28. <http://dx.doi.org/10.17762/turcomat.v12i3.1066>.

* Mahalakshmi, K., Udayakumar Allimuthu, L. Jayakumar, and Ankur Dumka. "A Timeline Optimization Approach of Green Requirement Engineering Framework for Efficient Categorized Natural Language Documents in Non-Functional Requirements." *International Journal of Business Analytics* 8, no. 1 (January 2021): 21–37. http://dx.doi.org/10.4018/ijban.2021010102.
  + - Websites

[**http://www.slideshare.net/**](http://www.slideshare.net/)

**https://software-engineering-book.com/slides/**

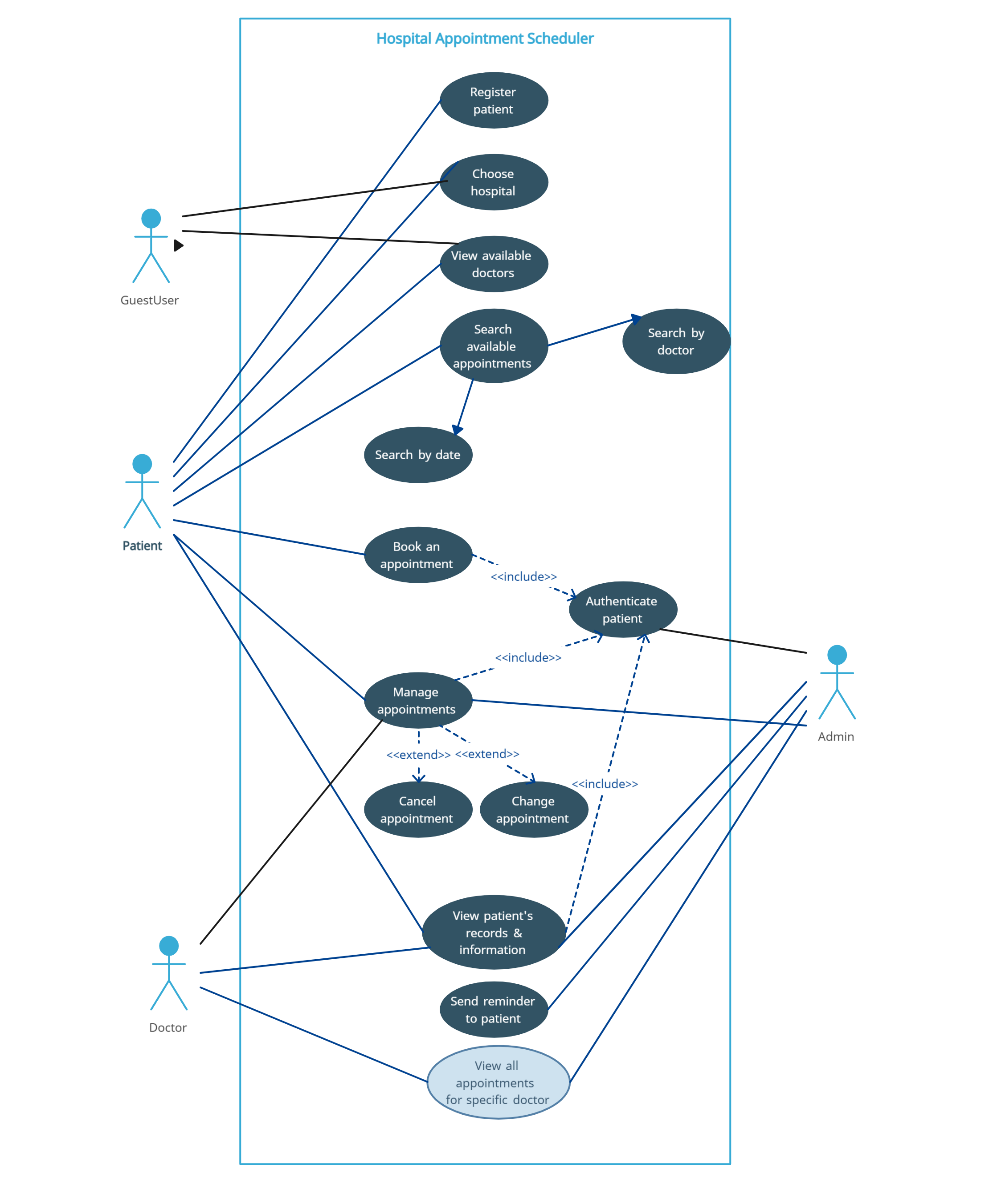
* [**https://www.scribd.com/document/637849619/hos**](https://www.scribd.com/document/637849619/hos)

# Overall Descriptions

## Product Perspective

Use Case Diagram of Hospital Appointment Scheduler

This is a broad level diagram of the project showing a basic overview. The users can be either Doctor or patient and may occasionally include GuestUser.This System will provide a search functionality to facilitate the search of doctors



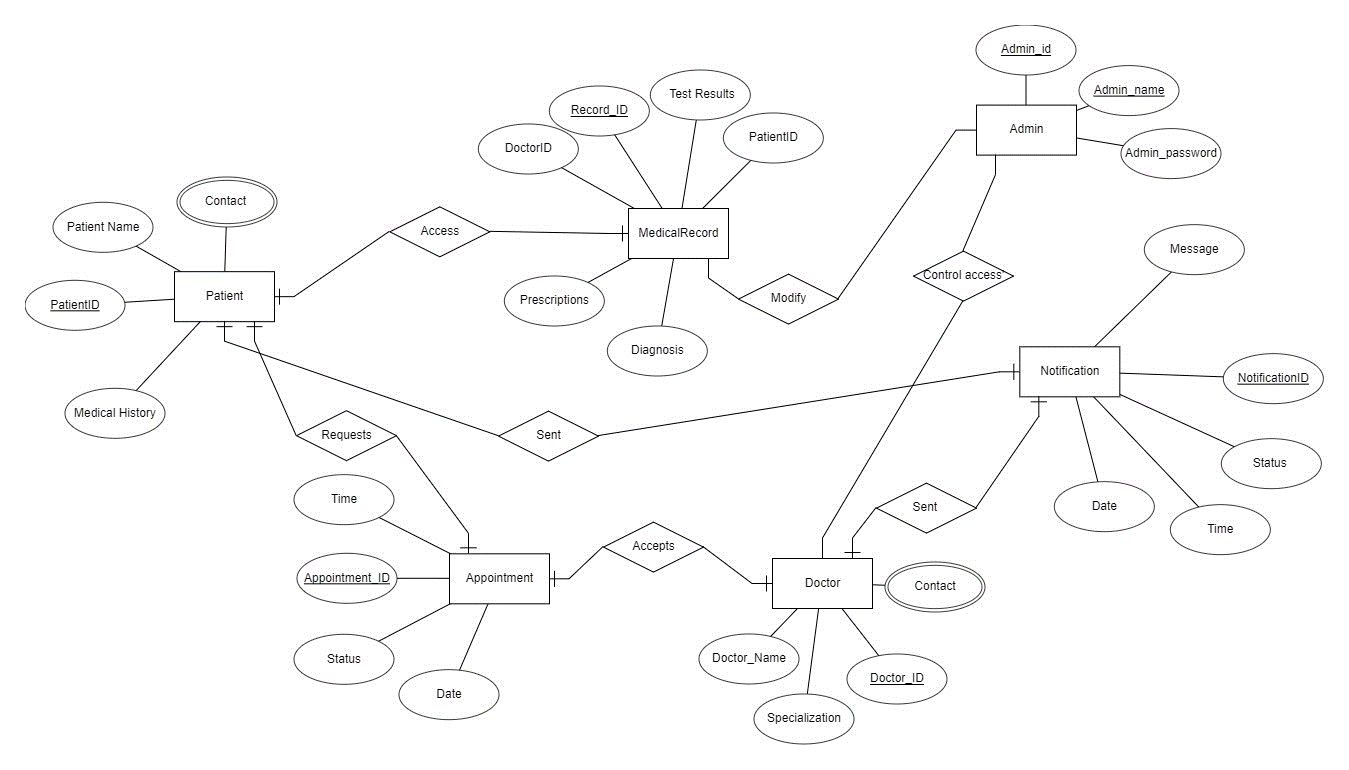
This search will be based on various categories viz. Doctor’s specialization. Further the user can also search appointments based upon the date of appointment or the specialization of doctor. The patient can also register on the application and choose the hotel from the application. The patient can book an appointment and manage his appointments that is either he can delete an appointment or change the appointment. These all functionalities require the authentication of the patient to be able to be executed.

A doctor on the other hand can view the patient’s records and previous appointments and can also view his or her all appointments. He can also manage his appointments which can be further extended to deleting and changing a particular appointment. The admin on the other hand can authenticate patient manage appointments and also have access to the history of appointments of both users that is the doctor and the patient.

.

## Product Function

Entity Relationship Diagram of Hospital Appointment Scheduler



. The Hospital Appointment Scheduler automates and simplifies the management of medical appointments and patient records. The software allows patients to schedule, modify, and view their appointments online. Healthcare professionals can efficiently confirm and manage appointments, access patient records, and receive automated reminders. Admins have control over user accounts and permissions. This system aims to reduce manual work, enhance efficiency, and ensure seamless coordination in healthcare scheduling and record-keeping.

## User Classes and Characteristics

The system provides different types of services based on the type of users [Patient/Doctor/Administrator/GuestUsers]. The Patient , Doctor and GuestUser will be acting as the member and Administrator have the authority to manage all the privileges provided to members. The member can access the Hospital Appointment Scheduling System through Online.

The features that are available to the Patients are:-

* Can view and manage their own profiles including personal information and contact details.
* Can view and manage their medical records securely.
* Can request appointments with available doctors.
* Can set preferences for receiving appointment remainders.

The features that are available to the Doctors are:-

* Can view and manage appointment schedules , accept or reject the appointment requests.
* Can access and update patient medical records.
* Can set preferences for receiving appointment notifications.

The features that are available to the GuestUsers(Non-Registered) are:-

* Can access publicly available information without logging in.
* Can check general information about available appointments and services.

The features that are available to the Administrators are:-

* Can create , modify and deactivate user accounts for patients and doctors.
* Can configure system settings including access controls and notification parameters.
* Can monitor and review system logs and audit trails for security and accountability.

## Operating Environment

The product will be operating in windows environment. The Hospital Appointment Scheduler is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error free
    - The system should be user-friendly so that it is easy to use for the users
    - The information of all users, books and libraries must be stored in a database that is accessible by the website
    - The system should have more storage capacity and provide fast access to the database
    - The system should provide search facility and support quick transactions
    - The Hospital Appointment Scheduler is running 24 hours a day
    - Users may access from any computer that has Internet browsing capabilities and an

Internet connection

* + - Users must have their correct usernames and passwords to enter into their online accounts and do actions

The dependencies are:-

* + - The specific hardware and software due to which the product will be run
    - On the basis of listing requirements and specification the project will be developed and run
    - The end users (admin) should have proper understanding of the product
    - The system should have the general report stored
    - The information of all the users must be stored in a database that is accessible by the Hospital Appointment Scheduler.
    - Any update regarding the medical history or appointment from the Hospital is to be recorded to the database and the data entered should be correct

## Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP Language: Java Runtime Environment, Net beans 7.0.1 (front end) Database: MS SQL Server (back end)

Hardware Configuration:- Processor: Pentium(R)Dual-core CPU Hard Disk: 40GB

RAM: 256 MB or more

## Data Requirement

The Hospital Appointment Scheduler requires comprehensive data inputs and outputs to facilitate its core functionalities. For appointment scheduling, users input patient information, preferred appointment details, and doctor preferences, receiving confirmation notifications and detailed appointment information as outputs. Medical record access and management involve inputs such as patient identifiers and medical data, yielding outputs comprising comprehensive medical records, prescriptions, and diagnostic results. Notification and reminders rely on inputting appointment schedules and patient preferences to generate outputs like appointment reminders, medication alerts, and emergency notifications. Additionally, user account details involve inputs of credentials and personal information, resulting in outputs confirming account creation, access permissions, and detailed user profiles, ensuring a secure and efficient healthcare management system.

# External Interface Requirement

## GUI

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing and monitoring the details of the medical records .

* + - It allows user to Request , view and cancel appointments .
    - The user interface must be customizable by the administrator
    - All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined
    - The design should be simple and all the different interfaces should follow a standard

template

* + - The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

Login Interface:-

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears. If the user forgot the username or password then there will be a availability of account recovery option.

Patient DashBoard:-

In this dashboard it allows patients to manage their accounts and access medical records including

appointment request form , appointment history .

Doctor DashBoard:-

In this dashboard it provides doctors with tools to manage schedules and access patient information.

It provides components like Schedule management interface , patient list and medical record access.

Administrator Console:-

It is used to enable administrators to manage the user accounts and configure system settings.

It provides components like user management interface and system configuration settings.

Appointment Scheduling Interface:-

It is used to facilitate the scheduling of appointments by patients and confirmation by doctors.

It provides components like Calendar for appointment selection , Doctor availability display and

Appointment request form.

Medical records viewer:-

It is used to allow doctors and patients to view and update medical records securely.

It provides components like patient information display , medical history timeline and

Editable fields for updates.

Notification preference settings:-

It is used to enable users to customize their notification preferences.

It provides components like Notification Type selection(SMS , Email).

Guest User Interface:-

It is used to allow non-registered users to access limited information.

It provides components like general information display and publicly services overview.

Help and Documentation Links:-

It is used to assist users in understanding system functionalities.

It provides components like FAQ section and Help and Support options with User Manuals link.

# System Features

The users of the system should be provided the surety that their account is secure and to meet the needs of the patients, doctors and administrators .This is possible by providing:-

* User Authentication and Authorization.
* Profile Management and Appointment Scheduling and Confirmation.
* Medical record access and Management.
* Notification and Remainders.
* Specialized DashBoard for doctors to manage schedules and view patient information.
* User management by Administrator.
* Audit Trails and Logging.
* Responsive User Interface.
* Security Measures.
* Help and Documentation.
* Scalability and Performance Optimization.
* Testing , Deployment and Support.

# Other Non-functional Requirements

## Performance Requirement

## The envisioned Hospital Appointment Scheduler is intended to serve as the primary scheduling system across various departments of the healthcare facility, facilitating interactions between medical staff and patients. The database is anticipated to seamlessly meet all functional requirements outlined by the hospital, ensuring efficient appointment scheduling, medical record management, and timely notifications for both healthcare providers and patients.

* + - The performance of the system should be fast and accurate
    - Hospital Appointment Scheduler shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus it should have inbuilt error testing to identify invalid username/password
    - The system should be able to handle large amount of data. Thus it should accommodate high number of records and users without any fault

## Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

## Security Requirement

* + - System will use secured database
    - User Authentication: Implement secure login functionality using encrypted passwords and multi-factor authentication (if possible) for patients, doctors, and admin roles.
    - Medical Record Confidentiality: Ensure all medical records are encrypted both in transit and at rest to prevent unauthorized access.
    - Limited Data Exposure: Implement mechanisms to restrict the exposure of sensitive patient information only to authorized medical professionals.
    - Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
    - System will have different types of users and every user has access constraints.
    - Proper user authentication should be provided.
    - No one should be able to hack user’s password.
    - There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

## Requirement attributes

* + - There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
    - The project should be open source
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
    - The user be able to easily download and install the system

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data.This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## User Requirement

The users of the system are patients and Doctors of the hospital and admin who act as administrator to maintain the system. The members are assumed to have basic knowledge of the computers and internet browsing. The administrators of the system should have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The admin provides certain facilities to the users in the form of:-

* + - Backup and Recovery
    - Forgot Password
    - Data migration i.e. whenever user registers for the first time then the data is stored in the server
    - Data replication i.e. if the data is lost in one branch, it is still stored with the server
    - Auto Recovery i.e. frequently auto saving the information
    - Maintaining files i.e. File Organization
    - The server must be maintained regularly and it has to be updated from time to time

# Other Requirements

## Data and Category Requirement

**User Categories and Access Rights:** The system comprises three primary user categories—patients, doctors, and administrators—each assigned specific access rights based on their roles within the system. Patients have access to schedule appointments, view their medical history, and receive notifications. Doctors possess the capability to manage their schedules, confirm appointments, and update patient medical records. Administrators hold comprehensive access rights, enabling them to modify, delete, and manage all aspects of the system, including user accounts and data.

**Data Categorization and Access Privileges:** The system segregates data into distinct categories, aligning with the diverse needs of the users. Medical records, appointment schedules, and notifications are categorized to ensure streamlined access and management. Access privileges are granted accordingly, allowing patients to retrieve their medical history and schedule appointments. Doctors are empowered to access and update patient records and manage appointment schedules, while administrators have complete access to all system data for comprehensive management and oversight.

**Coded Data Formats:** The system employs specific coding formats to categorize and present data effectively. Appointment schedules are organized chronologically and can be accessed and modified by doctors and patients based on their respective roles. Medical records are structured in a format that ensures secure storage and accessibility, allowing doctors to update patient information while granting patients read-only access to their records. Notifications and reminders are formatted to convey essential information to patients and doctors in a clear and timely manner.

**User-Centric Data Presentation:** Based on user categories and their associated access rights, the system presents data in a user-centric manner. Patients are presented with a simplified interface to schedule appointments and view their medical history. Doctors access a more comprehensive dashboard allowing them to manage schedules and update patient records. Administrators have privileged access to all system data for efficient monitoring, management, and maintenance.

## Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

## Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

* + - Administrator: A login id representing a user with user administration privileges to the software
    - User: A general login id assigned to most users
    - Client: Intended users for the software
    - SQL: Structured Query Language; used to retrieve information from a database
    - SQL Server: A server used to store data in an organized format
    - Layer: Represents a section of the project
    - User Interface Layer: The section of the assignment referring to what the user interacts with directly
    - Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
    - Data Storage Layer: The section of the assignment referring to where all data is recorded
    - Use Case: A broad level diagram of the project showing a basic overview
    - Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
    - Interface: Something used to communicate across different mediums
    - Unique Key: Used to differentiate entries in a database

## Class Diagram

In the Hospital Appointment Scheduler class diagram, several key classes interact to facilitate the scheduling, management, and access of appointments and medical records. The central class, 'Appointment Scheduler,' orchestrates the entire system and connects various essential classes. It contains attributes such as appointment List and patient List, indicating the aggregation relationship between the Scheduler and Appointments/Patients. The 'Doctor' class, having attributes like schedule and specialization, is associated with the 'Appointment' class to confirm and manage appointments. The 'Patient' class, with medical History and contact Info attributes, requests appointments and accesses medical records. Furthermore, the 'Notification' class, with preferences and notification Type, manages reminders for appointments, associating with both 'Patient' and 'Doctor' classes. Access control for medical records and appointment management is enforced through 'Access Control' or 'Permission' classes, ensuring secure data handling. These classes collectively represent the static model of the system, depicting relationships through associations, aggregations, and possibly generalization to portray the connections and responsibilities among the classes.

