

BAHRIA UNIVERSITY (KARACHI CAMPUS)

Software Design & Architecture (SEN-221)

ASSIGNMENT # 4 – Spring 2023

Based on: CLO-5

Class: **BSE-4B** Submission Deadline: **15th Jun 23**

Course Instructor: **ENGR. MAJID KALEEM** Max Marks: **04**

Name: **Muhammad Shoaib Akhter Qadri** Enrollment No: **02-131212-009**

Name: **Muhammad Mutayyab Imran**

1. In this task you are required to select the most relevant architectural model/style models to reflect implementation details from those requirements that you documented in SRS/user stories in your previous semester. You will submit your SRS/user stories as well along with this assignment.
2. You may complete this assignment in a group of maximum three members. Ideally those who were members in *Software Requirements Engineering*.
3. Viva/presentations regarding this assignment will be conducted individually. Each member must specify and present his/her role.
4. Produce *Use Cases* (diagrams) from the functional requirements (as mentioned in your SRS/user stories). In addition, use following use case template to describe your use cases.
5. Mention which type of *Software Architecture* (based on non-functional requirements) is suitable in the given scenario (as mentioned in your SRS/user stories). You are required to produce relevant diagram(s).
6. You are also required to specify the use of *Software Design Patterns* (with justifications) wherever applicable.
7. Justify why you would select a particular type of programing language to implement this.
8. **Architectural Model:**

We will use three Architectural Models in this scenario in which are following:

* **Model-View-Controller (MVC):**

The MVC architectural pattern will be apply to the user interface layer of the Hospital Management System. The model represents the system's data and business logic, the view handles the presentation and user interface, and the controller manages the interaction between the model and view. Implementing MVC can provide separation of concerns, modular development, and ease of maintenance for the user interface components.

* **Client-Server Architecture:**

The Client-Server architecture model will be use for the Hospital Management System as it involves multiple clients (users) accessing the system's services and resources. The clients, such as front desk staff, administrators, and users, would interact with a centralized server that manages the system's business logic and data. This model allows for centralized control, scalability, and efficient resource management.

* **Event-Driven Architecture:**

The Event-Driven architecture model can be utilized for handling asynchronous events and communication within the Hospital Management System. For example, when a new patient is added, an event can be triggered that notifies different modules or components to update their respective data. This model allows for loose coupling, scalability, and responsiveness to real-time events.

1. **Use Case Diagram:**

Diagram

Description automatically generated

1. **Template:**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | Hospital Management System | | |
| Created By: | Shoaib Akhter, Mutayyab Imran | Last Updated By: | Shoaib Akhter |
| Date Created: | 2/6/2023 | Date Last Updated: | 8/6/2023 |

|  |  |
| --- | --- |
| Actors: | Admin |
| Description: | The admin is responsible for managing the overall functioning of the hospital management system. They have access to all the modules and can perform administrative tasks. |
| Trigger: | Admin login. |
| Preconditions: | * The hospital management system is running and accessible. * The admin has valid login credentials. |
| Postconditions: | * The admin successfully logs in and gains access to the system. * The admin can perform various administrative tasks. |
| Normal Flow:  Sunny Day Scenario: | **Normal Flow:**  1. The admin opens the hospital management system.  2. The system prompts for admin credentials.  3. The admin enters the username and password.  4. The system verifies the credentials.  5. If the credentials are valid, the system grants access to the admin.  6. The admin can now perform administrative tasks, such as:  - Managing user accounts.  - Assigning roles and permissions.  - Generating reports.  - Configuring system settings.  - Managing system backups. |
| Alternative Flows:  Rainy Day Scenario: | **Alternative Flows:**  1. Invalid Credentials:  - If the admin enters invalid credentials, the system displays an error message.  - The admin can re-enter the correct credentials or request a password reset. |
| Exceptions: | - Network or server connectivity issues.  - System crashes or software errors.  - Security breaches or unauthorized access attempts.  - Data corruption or loss. |
| Includes: | - User Account Management  - Role and Permission Assignment  - Report Generation  - System Configuration  - System Backup Management |
| Priority: | High |
| Frequency of Use: | Daily |
| Business Rules: | - Only the admin has access to the administrative functions.  - User accounts must be created by the admin.  - All system changes and settings must be logged for auditing purposes. |
| Special Requirements: | - The hospital management system should have a robust security mechanism to protect sensitive patient data.  - The system should have a user-friendly interface for easy navigation and usability. |
| Assumptions: | - The admin has basic computer literacy and understands how to operate the hospital management system.  - The admin has been trained on the system's functionalities and administrative tasks. |
| Notes and Issues: | - Regular system maintenance and updates are essential to ensure the smooth operation of the hospital management system.  - Security measures, such as strong passwords and encryption, should be implemented to protect patient data. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | Hospital Management System | | |
| Created By: | Shoaib Akhter, Mutayyab Imran | Last Updated By: | Mutayyab Imran |
| Date Created: | 2/6/2023 | Date Last Updated: | 8/6/2023 |

|  |  |
| --- | --- |
| Actors: | Patient |
| Description: | The patient is a user of the Hospital Management System who interacts with the system to book appointments, access medical records, make payments, and request medication. |
| Trigger: | The patient wants to perform a task related to their healthcare within the hospital. |
| Preconditions: | 1. The patient must have a valid account registered in the Hospital Management System.  2. The patient must be logged into their account. |
| Postconditions: | 1. The patient successfully completes their desired task.  2. The system records the patient's activity for future reference. |
| Normal Flow:  Sunny Day Scenario: | 1. The patient logs into the Hospital Management System.  2. The patient selects the desired option, such as booking an appointment, accessing medical records, making a payment, or requesting medication.  3. The system presents the necessary forms or information for the selected option.  4. The patient fills in the required details (e.g., appointment date and time, medical record access request, payment information, medication request).  5. The patient submits the form or request.  6. The system processes the request and provides a confirmation message to the patient.  7. The patient logs out of the system. |
| Alternative Flows:  Rainy Day Scenario: | 1. If the selected appointment slot is not available, the system displays an error message, and the patient selects a different appointment slot.  2. If the patient wants to access their medical records, they select the "Access Medical Records" option.  3. The system verifies the patient's identity and grants access to the medical records. |
| Exceptions: | 1. If the patient enters invalid login credentials, the system displays an error message and prompts the patient to enter valid credentials.  2. If the patient fails to fill in all the required information in a form, the system displays an error message and prompts the patient to complete the form. |
| Includes: | 1. Doctor  2. Receptionist  3. Lab Engineer  4. Quality Officer  5. Finance Officer  6. Pharmacist  7. Stockist |
| Priority: | High |
| Frequency of Use: | Daily |
| Business Rules: | 1. Only registered patients can access the Hospital Management System.  2. The patient can book appointments with available doctors.  3. The patient can access their medical records with proper authorization.  4. The patient can make payments for medical services.  5. The patient can request medication, which must be approved by a pharmacist. |
| Special Requirements: | 1. The Hospital Management System must ensure the security and confidentiality of patient data.  2. The system should provide a user-friendly interface for patients to interact with. |
| Assumptions: | 1. The Hospital Management System is already implemented and functional.  2. The patient has basic computer literacy and can navigate through the system.  3. The necessary hardware and software infrastructure is in place to support the system. |
| Notes and Issues: | - It is essential to regularly update and maintain the Hospital Management System to ensure its efficiency and security.  - Any feedback or issues encountered by the patient while using the system should be recorded and addressed promptly to improve user experience. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | Hospital Management System | | |
| Created By: | Shoaib Akhter, Mutayyab Imran | Last Updated By: | Shoaib Akhter |
| Date Created: | 2/6/2023 | Date Last Updated: | 8/6/2023 |

|  |  |
| --- | --- |
| Actors: | Doctor |
| Description: | This use case describes the scenario where a doctor prescribes medication for a patient, and the prescription is sent to the pharmacist for fulfillment. |
| Trigger: | The doctor decides to prescribe medication for a patient during a consultation. |
| Preconditions: | 1. The doctor is logged into the Hospital Management System.  2. The patient's medical records are available in the system.  3. The doctor has the necessary authority to prescribe medication. |
| Postconditions: | 1. The prescription is recorded in the patient's medical records.  2. The prescription is sent to the pharmacist for fulfillment. |
| Normal Flow:  Sunny Day Scenario: | 1. The doctor selects the patient's record from the system.  2. The doctor reviews the patient's medical history and current condition.  3. The doctor determines the appropriate medication and dosage for the patient.  4. The doctor enters the prescription details into the system, including the medication name, dosage, frequency, and duration.  5. The system validates the prescription details and records them in the patient's medical records.  6. The system sends the prescription to the pharmacist for fulfillment. |
| Alternative Flows:  Rainy Day Scenario: | - If the system detects a potential drug interaction or allergy based on the patient's medical history, the doctor is alerted and may need to modify the prescription accordingly.  - If the prescribed medication is not available in the hospital's stock, the system may notify the doctor and/or the stockist to arrange for its procurement. |
| Exceptions: | - If the doctor encounters an issue with the Hospital Management System or experiences a technical problem, the doctor may need to seek assistance from the IT department to resolve the issue. |
| Includes: | - Doctor accessing patient's medical records  - Doctor entering prescription details  - System validating prescription details  - System sending prescription to pharmacist |
| Priority: | High |
| Frequency of Use: | Frequent (Multiple times per day for each doctor) |
| Business Rules: | - Only authorized doctors can prescribe medication.  - Prescription details must be accurately recorded in the patient's medical records.  - Prescription should be sent to the pharmacist promptly. |
| Special Requirements: | - The Hospital Management System should have a comprehensive database of medications and their details.  - The system should be able to handle potential drug interactions and allergies. |
| Assumptions: | - The doctor has received proper training on prescribing medication and understands the potential risks and benefits associated with different drugs.  - The pharmacist has the necessary knowledge and expertise to fulfill the prescription accurately. |
| Notes and Issues: | - Integration with the pharmacy module of the Hospital Management System is crucial for seamless prescription fulfillment.  - The Hospital Management System should ensure data security and privacy, especially when dealing with sensitive patient information such as prescriptions. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | Hospital Management System | | |
| Created By: | Shoaib Akhter, Mutayyab Imran | Last Updated By: | Mutayyab Imran |
| Date Created: | 2/6/2023 | Date Last Updated: | 8/6/2023 |

|  |  |
| --- | --- |
| Actors: | Receptionist |
| Description: | This use case describes the process of managing patient registration within the hospital management system. |
| Trigger: | A new patient arrives at the hospital and needs to be registered. |
| Preconditions: | - The receptionist is logged into the hospital management system.  - The patient has provided the necessary information for registration. |
| Postconditions: | - The patient's information is stored in the hospital management system.  - The patient is assigned a unique identification number. |
| Normal Flow:  Sunny Day Scenario: | 1. The receptionist greets the patient and initiates the registration process.  2. The receptionist enters the patient's personal information (e.g., name, age, contact details) into the system.  3. The receptionist verifies the patient's insurance information and enters it into the system if applicable.  4. The receptionist schedules an appointment for the patient, if necessary, by checking the doctor's availability and assigning a time slot.  5. The receptionist generates a unique identification number for the patient.  6. The system saves the patient's information, appointment details, and assigns the identification number.  7. The receptionist prints out an identification card or receipt containing the patient's details and hands it to the patient.  8. The receptionist provides any additional information or instructions to the patient, if required.  9. The registration process is completed. |
| Alternative Flows:  Rainy Day Scenario: | - If the patient is already registered in the system:  1. The receptionist searches for the patient's existing record.  2. The receptionist updates any necessary information, such as contact details or insurance information.  3. The receptionist verifies the appointment details and makes changes if needed.  4. The receptionist provides the updated identification card or receipt to the patient. |
| Exceptions: | - If the patient refuses to provide necessary information, the receptionist informs the patient about the requirement and asks for cooperation.  - If there are technical issues with the system, the receptionist notifies the IT department for assistance. |
| Includes: | - Notify Doctor: After completing the registration, the system sends an automated notification to the assigned doctor about the new patient. |
| Priority: | High |
| Frequency of Use: | Multiple times per day, depending on the number of new patients. |
| Business Rules: | - Patient information must be accurately recorded and securely stored.  - Patient appointments must be scheduled based on doctor availability.  - Identification numbers must be unique for each patient. |
| Special Requirements: | - The hospital management system must have a user-friendly interface for easy registration.  - The system should support quick search and retrieval of existing patient records. |
| Assumptions: | - The receptionist is trained in using the hospital management system.  - The necessary hardware and software infrastructure are in place for the registration process. |
| Notes and Issues: | - It is important to ensure data privacy and comply with applicable regulations (e.g., HIPAA) when handling patient information.  - Integration with other modules of the hospital management system (e.g., billing, appointment scheduling) may be required for a seamless workflow. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | Hospital Management System | | |
| Created By: | Shoaib Akhter, Mutayyab Imran | Last Updated By: | Shoaib Akhter |
| Date Created: | 2/6/2023 | Date Last Updated: | 8/6/2023 |

|  |  |
| --- | --- |
| Actors: | Finance Officer |
| Description: | The Finance Officer plays a crucial role in the Hospital Management System by handling financial transactions, budgeting, and financial reporting within the hospital. |
| Trigger: | The Finance Officer needs to perform financial tasks and maintain accurate financial records. |
| Preconditions: | 1. The Finance Officer must be logged into the Hospital Management System.  2. The Finance Officer must have appropriate permissions and access rights. |
| Postconditions: | 1. Financial transactions and records are updated and stored in the system.  2. Reports and statements are generated accurately. |
| Normal Flow:  Sunny Day Scenario: | 1. The Finance Officer logs into the Hospital Management System.  2. The system presents the Finance Officer with a dashboard or menu displaying various financial tasks and options.  3. The Finance Officer enters the necessary information for the selected task, such as transaction details, financial figures, or budget allocations.  4. The system validates the input and updates the financial records accordingly.  5. If required, the system generates reports or statements based on the entered data.  6. The Finance Officer reviews the generated reports for accuracy and completeness.  7. The Finance Officer saves or prints the reports as necessary.  8. The Finance Officer logs out of the system. |
| Alternative Flows:  Rainy Day Scenario: | 1. If the entered information is invalid or incomplete:  a. The system displays an error message.  b. The Finance Officer corrects the information and resubmits.  c. The system validates the corrected input and updates the financial records accordingly.  2. If there are discrepancies or inconsistencies in the financial records:  a. The Finance Officer investigates the issue.  b. The Finance Officer may consult other relevant personnel or departments.  c. Once the issue is resolved, the Finance Officer updates the records accordingly.  3. If there are technical issues or system failures:  a. The Finance Officer reports the problem to the IT department or system administrators.  b. The Finance Officer may temporarily switch to manual methods or alternative systems until the issue is resolved. |
| Exceptions: | 1. Unauthorized access attempts: If someone other than the Finance Officer tries to access the system, appropriate security measures are triggered, such as login restrictions or alerts.  2. Non-compliance with financial regulations: If the Finance Officer attempts to perform tasks that violate financial regulations or internal policies, the system may restrict access or display warnings. |
| Includes: | - Interaction with the hospital staff for clarifications or additional information related to financial tasks.  - Integration with external financial systems or banks for seamless financial transactions and reporting. |
| Priority: | High |
| Frequency of Use: | Daily or as required for financial tasks and reporting. |
| Business Rules: | - The Finance Officer must follow financial regulations, accounting principles, and internal policies.  - Financial transactions must be accurately recorded and updated in real-time.  - Reports and statements must be generated promptly and accurately. |
| Special Requirements: | - The Hospital Management System must have robust security measures to protect financial data and prevent unauthorized access.  - The system should provide relevant financial analytics and forecasting capabilities to aid decision-making.  - The system should support integration with external financial systems or banks for efficient financial transactions. |
| Assumptions: | - The Finance Officer has the necessary knowledge and expertise to perform financial tasks and interpret financial information.  - The Hospital Management System is accessible and available for use during the Finance Officer  's working hours.  - The system has been properly configured with accurate financial data and parameters. |
| Notes and Issues: | - It is essential to ensure data accuracy and integrity when updating financial records or generating reports.  - Regular backup and data protection measures should be in place to prevent data loss or corruption.  - Compliance with applicable financial regulations and internal policies is crucial to maintain the integrity of the financial management process. |

1. ***Software Architecture:***

Based on the given non-functional requirement of security for the Hospital Management System scenario, a suitable software architecture would be a **three-tier architecture**.

**Three-tier architecture** separates the system into three layers: presentation layer, application layer, and data layer. Each layer has its own responsibilities, which helps to enhance security and maintainability.

1. **Presentation Layer**:

This layer handles the user interface and interaction with the system. It is responsible for validating user inputs, including the PIN and login credentials, and ensuring secure communication between the user and the system.

1. **Application Layer:**

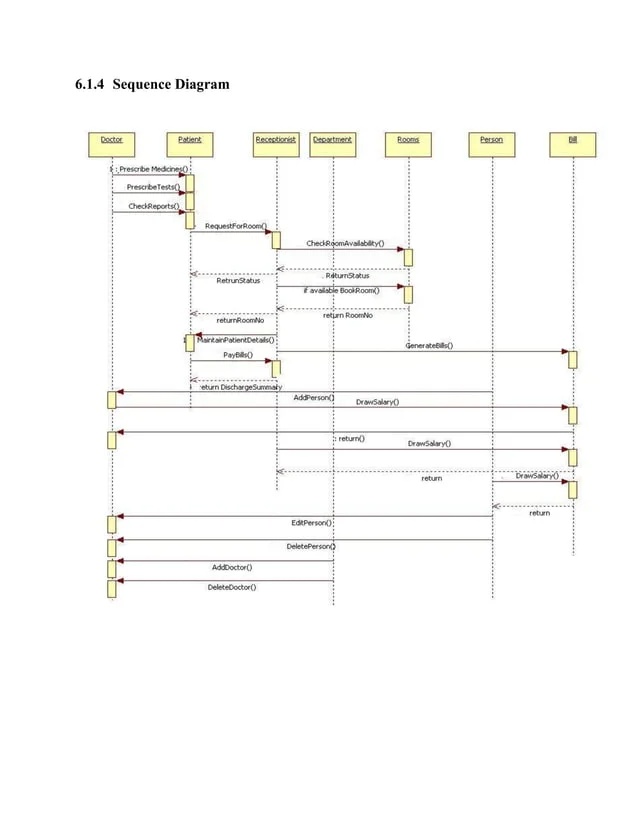
This layer contains the business logic and controls the system's functionality. It enforces the security requirements, such as access control and user rights management. Only authorized administrators should have the ability to modify information in the system, while front desk staff should only have read access.

1. **Data Layer:**

This layer manages the storage and retrieval of data. It should implement robust security measures, including encryption, to protect sensitive patient information. The database should be accessible only to authorized personnel, such as administrators, and any modifications should be synchronized and performed by the administrator in the ward.

By implementing a three-tier architecture, the Hospital Management System can achieve the necessary security measures, such as user authentication, access control, and secure data storage, while also providing scalability, maintainability, and reliability.

**Diagram**:



1. **Software Design Pattern:**

Here are some software design patterns that can be applied in the Hospital Management System scenario, along with their justifications:

Singleton Pattern: The Singleton pattern can be applied to ensure that there is only one instance of certain classes throughout the system. For example, the DatabaseConnection class that handles database access can be implemented as a Singleton to ensure that there is a single, shared connection object. This helps in managing resources efficiently and maintaining data integrity.

Proxy Pattern: The Proxy pattern can be employed to provide controlled access to sensitive resources or operations. In the Hospital Management System, the Proxy pattern can be used to restrict access to certain functionalities based on user roles. For example, a UserAccessProxy can be implemented to verify the user's role and permissions before granting access to modify patient information. This helps enforce the principle of least privilege and enhances security.

Factory Method Pattern: The Factory Method pattern can be employed when creating different types of objects based on a common interface or superclass. In the Hospital Management System, a Factory Method pattern can be used to create different types of users based on their roles, such as creating instances of FrontDeskStaff or Administrator classes. This allows for flexibility in adding new user types in the future and promotes code reusability.

1. **Programming Languages:**

We will select C# programming languages for this purpose because of these non-functional requirements:

1. **Security**:

Since security is a critical requirement for the system, it is important to choose a programming language that has robust security features and a strong track record in terms of security vulnerabilities. Languages like Java and C# provide built-in security features and have mature frameworks for implementing secure web applications. They offer features such as strong type checking, input validation, and memory management that can help mitigate security risks.

1. **Scalability**:

The system needs to support a large number of users and handle concurrent requests. Therefore, a language with good support for concurrent programming and scalability is desirable. Languages like Java and C# have frameworks and libraries that facilitate building scalable applications. Additionally, languages with support for asynchronous programming, such as JavaScript with Node.js, can help handle a high number of concurrent requests efficiently.

1. **Integration**:

The Hospital Management System may need to integrate with various external systems, such as payment gateways, medical devices, or other hospital systems. Choosing a programming language that has extensive libraries, APIs, and support for integration can simplify the development and maintenance of these integrations. Popular languages like Java and Python have a wide range of libraries and frameworks for integration with external systems.

1. **Maintainability**:

Given the long-term nature of the Hospital Management System, it is crucial to choose a programming language that promotes maintainability. This includes factors such as code readability, community support, availability of tools and frameworks, and ease of debugging. Languages like Python and Java have strong community support, well-established coding conventions, and rich ecosystems that offer a wide range of tools and frameworks to aid in development and maintenance tasks.

1. **Performance**:

The system needs to respond quickly and handle a large number of transactions. Choosing a programming language that offers good performance can help meet these performance requirements. Languages like Java, C++, and Go are known for their performance capabilities. Additionally, leveraging caching mechanisms, optimizing database queries, and employing

efficient algorithms can further enhance performance.

Good Luck!