Dr. Ambedkar Institute of Technology

[As Auseronous Institution, affiliated to VTU, Belgam and Aided by Government of Karnataka]

Near Jnana Bharathi Campus, Mallathahalli, Bangalore-560056.

DEPARTMENT OF MECHANICAL ENGINEERING

ENGINEERING ECONOMICS 21MET 6053

REPORT ON

"World Economy & Indian Economy"

Under Guidance of

Dr.SHRINIVAS

Assistant Professor

Visvesvaraya Technological University

Submitted By:

GROUP MEMBERS:

SI No:	Name	USN
01	Hajratali s mogalalli	1DA22CS409
02	Abhinav Kumar	1DA21IS001
03	BHARATKUMAR N MEDEGAR	1DA21CS033
04	Sumith sindhe	1DA21AE028
05	Karthik V Desai	1DA21CS073
06	PAVAN G B	1DA21CS099
07	MANJUNATH GOWDA R	1DA21CS084
08	YASHAS B Y	1DA21CS169
09	P Bharat Kumar Reddy	1DA21IS034
10	Abhishek H.	1DA21IS002
11	Mallikarjun	1DA21EC077

Table of Contents

DR. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Affiliated to VTU, Belgaum and Aided by Governmentof Karnataka)

Near Jnana Bharathi Campus, Bangalore-560056

Department of Mechanical Engineering

PROJECT REPORT ON

"Fake News Detection using Machine Learning"

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

2023-2024

SUBMITTED BY

SAHIL KHURSHEED [USN: 1DA21IS041]

ABHINAV KUMAR [USN: 1DA21IS001]

UNDER THE GUIDANCE OF:

Dr. SHILPA BIRADAR

Asst. Professor, Dept. of ISE

Dr. Ambedkar Institute of Technology.

Department of Information Science & Engineering

DR. AMBEDKAR INSTITUTE OF TECHNOLOGY

Outer Ring Road, Near Jnana Bharathi Campus

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the detailed requirements for the development and implementation of the Unified Seat Reservation System. This document serves several key objectives:

1. Requirement Clarification: It provides a clear and unambiguous description of the functionalities, features, and constraints that the system must adhere to. By documenting these requirements, it ensures that all stakeholders — including developers, designers, project managers, and end-users — have a shared understanding of what the system will deliver.
2. Basis for Agreement: The SRS document acts as a formal agreement or contract between the stakeholders involved in the project. It outlines the expectations and responsibilities of each party, thereby mitigating misunderstandings and conflicts during the development lifecycle.
3. Guidance for Development: By specifying the requirements in detail, the document serves as a roadmap for the development team. It provides them with clear directions on

what needs to be built and how different components of the system should interact to achieve the desired functionality.

4. Validation and Verification: It serves as a benchmark against which the final product can be validated and verified. Each requirement listed in the document acts as a criterion that the developed system must meet to be considered successful.
5. Scope Definition: It defines the scope of the Unified Seat Reservation System, outlining what functionalities and features will be included in the initial release. This helps in managing stakeholder expectations and prioritizing development efforts.
6. Risk Management: By identifying and documenting requirements upfront, the SRS document helps in identifying potential risks and dependencies early in the project lifecycle. This allows the project team to proactively address these challenges and minimize their impact on project timelines and budgets.

The purpose of this Software Requirements Specification (SRS) document is to articulate the functional and non-functional requirements necessary for the development of the Unified Seat Reservation System. By documenting these requirements in detail, this document aims to establish a comprehensive understanding among stakeholders regarding the system's capabilities, limitations, and operational characteristics.

This SRS serves as a foundational document that outlines the scope of the project, including the specific functionalities such as seat browsing, reservation creation, and booking management. It provides a structured framework for developers to design and implement the system in accordance with stakeholder expectations

and industry standards. Furthermore, this document acts as a contractual agreement between the project team and stakeholders, ensuring alignment on project goals and deliverables. It serves as a benchmark for validation and verification processes, allowing stakeholders to assess the system's compliance with defined requirements throughout its development lifecycle.

Ultimately, the SRS document aims to mitigate project risks by identifying potential challenges early on, such as technical constraints and dependencies on external systems. By addressing these factors proactively, we aim to deliver a robust and reliable Unified Seat Reservation

System that meets the needs of our users while adhering to industry best practices and regulatory requirements.

1.2 Scope

The scope section of the Software Requirements Specification (SRS) document delineates the boundaries of the Unified Seat Reservation System, specifying what the system will and will not do. This section is crucial for setting clear expectations among stakeholders and ensuring that the project team focuses on the defined objectives. A comprehensive scope description helps prevent scope creep and ensures that resources are appropriately allocated.

Scope Statement:

The Unified Seat Reservation System is designed to facilitate the process of reserving seats for various types of events, such as concerts, theatre performances, sports events, and conferences. The system will provide a unified platform that allows users to browse available seats, make reservations, and manage their bookings through an intuitive interface. The system will support multiple venues and events, offering a seamless experience for users and administrators.

Example Scope Statement:

The scope of the Unified Seat Reservation System includes functionalities for user registration and authentication, event and venue management, interactive seat browsing and selection, reservation and booking processes, and administrative functions for system management. The system will ensure security and privacy compliance, offer support and helpdesk features, and provide a seamless user experience for booking seats across various events and venues. The system will not include functionalities for event scheduling, external ticket sales, transportation and accommodation bookings, physical ticket printing, complex loyalty programs, detailed financial accounting, or marketing and promotional activities. By clearly defining these boundaries, we aim to deliver a focused and effective solution for seat reservations, while managing stakeholder expectations and project resources effectively.

In-Scope:

1. User Registration and Authentication:
    - Users can create accounts, log in, and manage their profiles.

- Authentication mechanisms to ensure secure access.

2. Event and Venue Management:
    - Administrators can create and manage events, including setting dates, times, and locations.
    - Venues can be defined with specific layouts and seating arrangements.
3. Seat Browsing and Selection:
    - Users can browse available events and venues.
    - Interactive seating maps allowing users to view available and reserved seats.
    - Filter options for events based on date, location, and category.
4. Reservation and Booking:
    - Users can select seats and make reservations.
    - Integration with payment gateways for processing payments.
    - Generation of electronic tickets with unique identifiers.
5. Booking Management:
    - Users can view, modify, or cancel their reservations.
    - Notifications and reminders for upcoming events.
6. Administrative Functions:
    - Administrators can manage user accounts, view booking statistics, and generate reports.
    - Tools for monitoring system performance and managing content.
7. Security and Privacy:
    - Implementation of data encryption for sensitive information.
    - Compliance with relevant data protection regulations.
8. Support and Help:
    - Helpdesk features for user support.
    - FAQ section and user guides.

Out of Scope:

1. Event Scheduling:
    - The system will not handle the creation of event schedules or timetables.
    - Integration with external event scheduling systems is not included.
2. External Ticket Sales:

- The system will not manage ticket sales through third-party platforms.
- Integrations with external ticketing services are beyond the scope of this project.

3. Transportation and Accommodation Booking:
    - The system will not provide functionalities for booking transportation or accommodation related to events.
4. Physical Ticket Printing:
    - The system will not support physical ticket printing services.
    - Users will receive electronic tickets only.
5. Complex Loyalty Programs:
    - While basic user profiles and preferences will be managed, complex loyalty or rewards programs are not included.
6. Detailed Financial Accounting:
    - Advanced financial accounting and auditing features beyond basic transaction logging are excluded.
7. Marketing and Promotions:
    - Management of marketing campaigns and promotional activities are not part of this system.

Scope Diagram:

A scope diagram (e.g., a context diagram) can visually represent the boundaries of the system, illustrating the system's interaction with external entities such as users, administrators, payment gateways, and external databases. This diagram can help stakeholders quickly grasp the extent and limits of the system's functionalities.

The scope section would be detailed, specifying the boundaries of the system comprehensively. It would include a discussion on what functionalities are within scope (e.g., seat browsing, reservation management, payment processing) and what functionalities are explicitly excluded (e.g., event scheduling, external ticket sales).

1.3 Definitions, Acronyms, and Abbreviations

This section provides definitions for terms, acronyms, and abbreviations used throughout the Software Requirements Specification (SRS) document. Understanding these terms is essential

for ensuring clear communication among all stakeholders involved in the Unified Seat Reservation System project.

Definitions

- Reservation: The act of booking one or more seats for an event through the system.
- Event: An organized occasion such as a concert, theatre performance, sports match, or conference for which seats can be reserved.
- Venue: The location where an event takes place, typically characterized by a layout of seats that can be reserved.
- Seat Map: A visual representation of the seating arrangement within a venue, showing available and reserved seats.
- Booking: The process of selecting and reserving seats for a specific event.
- User: An individual who interacts with the system, which can include customers, administrators, venue managers, and support staff.
- Customer: A user who browses events and makes reservations.
- Administrator: A user with elevated privileges responsible for managing events, venues, user accounts, and system settings.
- Venue Manager: A user responsible for managing the seating arrangements and logistical details of events held at a specific venue.
- Support Staff: Users who provide customer support and handle inquiries and issues related to the system.
- Authentication: The process of verifying the identity of a user attempting to access the system.
- Authorization: The process of granting or denying a user access to specific features and functionalities based on their role.

Acronyms

- SRS: Software Requirements Specification
- GUI: Graphical User Interface
- API: Application Programming Interface
- HTTP: Hypertext Transfer Protocol

- HTTPS: Hypertext Transfer Protocol Secure
- JSON: JavaScript Object Notation
- XML: Extensible Markup Language
- GDPR: General Data Protection Regulation
- PCI DSS: Payment Card Industry Data Security Standard
- UAT: User Acceptance Testing

Abbreviations

- Req: Requirement
- UI: User Interface
- UX: User Experience
- DB: Database
- SQL: Structured Query Language
- SSL: Secure Sockets Layer
- TLS: Transport Layer Security
- CSV: Comma-Separated Values
- CRUD: Create, Read, Update, Delete

Example Usage:

- Reservation: "Users can make a reservation for multiple events through the unified platform."
- Event: "Each event will have a unique identifier and associated metadata such as date, time, and venue."
- Venue: "The system will support multiple venues, each with its unique seat map."
- Seat Map: "An interactive seat map will allow users to view available seats in real-time."
- Booking: "The booking process includes selecting seats, entering payment details, and receiving a confirmation."
- User: "Different user roles, such as customer and administrator, will have specific access rights within the system."
- Authentication: "The system will implement multi-factor authentication to enhance security."

- Authorization: "Authorization levels will ensure that only administrators can modify event details."

This section helps to standardize the terminology used in the SRS document, ensuring all stakeholders have a common understanding of the terms and abbreviations used throughout the project documentation.

1.4 References

This section lists all the documents, standards, and resources referenced in this Software Requirements Specification (SRS) document for the Unified Seat Reservation System. These references provide additional context, guidelines, and standards that are essential for understanding and implementing the system requirements.

Internal References

1. Unified Seat Reservation System Design Document
    - Document ID: USRS-DD-001
    - Description: Detailed design specifications, including architectural diagrams, data flow diagrams, and design patterns.
    - Location: Internal project repository
2. User Interface Mock-ups and Wireframes
    - Document ID: USRS-UI-002
    - Description: Visual representations and design mock-ups of the user interface for different system components.
    - Location: Internal project repository

External References

1. ISO/IEC 27001:2013 - Information Security Management Standards

- Description: International standard for managing information security. Provides guidelines for implementing and maintaining an effective information security management system (ISMS).
- Publisher: International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)
- URL: ISO/IEC 27001:2013

2. PCI DSS - Payment Card Industry Data Security Standard
   - Description: Security standard for organizations that handle branded credit cards from major card schemes. Provides a framework for secure payment processing.
   - Publisher: PCI Security Standards Council
   - URL: PCI DSS
3. GDPR - General Data Protection Regulation
   - Description: Regulation in EU law on data protection and privacy for individuals within the European Union. Addresses the transfer of personal data outside the EU and EEA areas.
   - Publisher: European Union
   - URL: GDPR
4. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
   - Description: Provides recommended practices for writing software requirements specifications. Defines content and qualities of a good SRS document.
   - Publisher: Institute of Electrical and Electronics Engineers (IEEE)
   - URL: IEEE Std 830-1998
5. W3C Web Content Accessibility Guidelines (WCAG) 2.1
   - Description: Guidelines for making web content more accessible to people with disabilities. Covers web content, web applications, and web tools.
   - Publisher: World Wide Web Consortium (W3C)
   - URL: WCAG 2.1
6. RESTful Web Services: Design and Development
   - Author: Leonard Richardson, Sam Ruby
   - Description: Book that covers the principles and practices of designing RESTful web services.

- Publisher: O'Reilly Media
- ISBN: 978-0596529260

7. JavaScript: The Good Parts
    - Author: Douglas Crockford
    - Description: Book that explores the core features of JavaScript and provides best practices for using the language effectively.
    - Publisher: O'Reilly Media
    - ISBN: 978-0596517748
8. The Mythical Man-Month: Essays on Software Engineering
    - Author: Frederick P. Brooks Jr.
    - Description: Collection of essays on software engineering, including the challenges of project management and team coordination.
    - Publisher: Addison-Wesley
    - ISBN: 978-0201835953
9. Agile Software Development: Principles, Patterns, and Practices
    - Author: Robert C. Martin
    - Description: Comprehensive guide to agile development methodologies and practices.
    - Publisher: Prentice Hall
    - ISBN: 978-0135974445

Example Citation

In the SRS document, references can be cited as follows:

- "The system must comply with the GDPR for data protection (see GDPR)."
- "Design guidelines follow the best practices outlined in RESTful Web Services by Richardson and Ruby (2007)."

2. Overall Description

The overall description provides a high-level overview of the Unified Seat Reservation System, including its context, primary functions, user characteristics, constraints, and assumptions. This section aims to give a clear understanding of what the system is, what it aims to achieve, and the environment in which it will operate.

2.1 Product Perspective

The Unified Seat Reservation System is a web-based application designed to facilitate the reservation of seats for various events such as concerts, theater performances, sports events, and conferences. It integrates with existing venue management systems and external payment gateways to provide a seamless experience for users. The system is built to be compatible with multiple devices, including desktops, tablets, and smartphones.

System Interfaces

- External Payment Gateways: Integration with third-party payment processing services for handling transactions securely.
- Venue Management Systems: Interfaces with venue management software for real-time seat availability and updates.
- Email and SMS Services: For sending booking confirmations, reminders, and notifications to users.

User Interfaces

- Web Interface: Accessible through modern web browsers, designed for ease of use on both desktop and mobile devices.
- Administrative Dashboard: Provides tools for administrators to manage events, venues, and user accounts.

2.2 Product Functions

The Unified Seat Reservation System supports the following key functions:

- User Registration and Authentication: Allows users to create accounts, log in, and manage their profiles.
- Event and Venue Management: Enables administrators to create, modify, and delete events and manage venue details, including seating layouts.
- Interactive Seat Browsing and Selection: Provides users with a visual map of available seats and allows them to select and reserve their desired seats.
- Reservation and Booking: Facilitates the booking process, including seat selection, payment processing, and booking confirmation.
- Booking Management: Allows users to view their booking history, modify existing bookings, and cancel reservations.
- Payment Processing: Integrates with payment gateways to handle transactions securely, supporting multiple payment methods.
- Administrative Tools: Includes functionalities for monitoring system performance, generating reports, and managing user accounts.

2.3 User Characteristics

The system is designed to accommodate the following types of users:

- Customers: Individuals who browse events, reserve seats, and manage their bookings. They expect an intuitive interface and secure transaction processing.
- Administrators: Users with elevated privileges responsible for managing events, venues, and user accounts. They require comprehensive tools for system oversight and management.

- Venue Managers: Users responsible for the logistical aspects of events, including seating arrangements and availability updates. They need accurate and real-time information about seat reservations.
- Support Staff: Individuals providing customer support and resolving user issues. They require access to user and booking information to assist customers effectively.

2.4 Constraints

The development and operation of the Unified Seat Reservation System are subject to the following constraints:

- Technical Constraints: The system must be compatible with modern web browsers (e.g., Chrome, Firefox, Safari) and mobile devices (iOS and Android). It must also integrate with existing venue management systems and payment gateways.
- Regulatory Constraints: Compliance with GDPR for data protection and PCI DSS for secure payment processing is mandatory.
- Performance Constraints: The system must be capable of handling high traffic volumes, especially during peak booking periods for popular events.
- Security Constraints: The system must implement robust security measures to protect user data and ensure secure transactions.

2.5 Assumptions and Dependencies

The development and successful operation of the Unified Seat Reservation System depend on the following assumptions and dependencies:

- Reliable Internet Connectivity: Users and administrators are assumed to have stable internet connections to access the system.
- Basic Internet Navigation Skills: It is assumed that users have basic skills to navigate web applications.
- Third-Party Services: The system's functionality relies on the availability and reliability of external payment gateways and venue management systems.
- Regulatory Compliance: The system assumes adherence to regulatory requirements such as GDPR and PCI DSS.

- Operational Support: The presence of support staff to assist users and handle issues is assumed

could illustrate the layout and design of user interfaces (UIs) for different screens (e.g., home page, seat selection, booking confirmation). Each UI component would be described in terms of functionality and usability considerations.

3. Specific Requirements

This section details the specific requirements for the Unified Seat Reservation System, including external interface requirements, functional requirements, and non-functional requirements. Each requirement is identified by a unique identifier for easy reference.

3.1 External Interface Requirements

3.1.1 User Interfaces

The system must provide the following user interfaces:

- Customer Interface:
  - Home Page: Displays a list of upcoming events and promotions.
  - Event Details Page: Provides detailed information about a selected event, including date, time, venue, and available seats.
  - Interactive Seat Map: Allows users to view and select available seats in a graphical format.
  - Booking Page: Guides users through the booking process, including seat selection, payment details, and confirmation.
  - User Account Page: Allows users to manage their profile, view booking history, and modify or cancel reservations.
- Administrative Interface:
  - Dashboard: Provides an overview of system status, upcoming events, and recent activities.
  - Event Management Page: Allows administrators to create, modify, and delete events.

- Venue Management Page: Enables the management of venue details and seat maps.
- User Management Page: Provides tools for managing user accounts and permissions.
- Reports Page: Generates reports on bookings, revenue, and system performance.

3.1.2 Hardware Interfaces

The system is a web-based application and does not have specific hardware interface requirements beyond standard server and networking infrastructure to host the application and ensure reliable internet access for users.

3.1.3 Software Interfaces

The system must interface with the following software components:

- Payment Gateways: Integration with third-party payment processors like PayPal, Stripe, and credit card networks to handle transactions securely.
- Venue Management Systems: Interface with existing venue management software to sync seat availability and event details.
- Email and SMS Services: Integration with services like SendGrid or Twilio to send booking confirmations, reminders, and notifications.

3.1.4 Communication Interfaces

The system will use standard web communication protocols:

- HTTP/HTTPS: For secure communication between clients and the server.
- RESTful APIs: For interaction with external services such as payment gateways and venue management systems.
- WebSocket's: For real-time updates on seat availability and booking status.

3.2 Functional Requirements

The functional requirements specify the actions the system must be able to perform. They are categorized based on different functionalities of the system.

3.2.1 User Registration and Authentication

- FR1: The system shall allow users to create an account using their email address and a password.
- FR2: The system shall allow users to log in using their registered email address and password.
- FR3: The system shall provide a password recovery option through email.

3.2.2 Event and Venue Management

- FR4: The system shall allow administrators to create, modify, and delete events.
- FR5: The system shall allow administrators to specify event details such as date, time, venue, and pricing.
- FR6: The system shall allow administrators to manage venue details, including seat maps and capacity.

3.2.3 Interactive Seat Browsing and Selection

- FR7: The system shall display an interactive seat map for each event.
- FR8: The system shall allow users to select available seats from the seat map.
- FR9: The system shall update the seat map in real-time to reflect current availability.

3.2.4 Reservation and Booking

- FR10: The system shall guide users through the booking process, including seat selection and payment details.
- FR11: The system shall process payments securely through integrated payment gateways.
- FR12: The system shall send booking confirmation to users via email and/or SMS.

3.2.5 Booking Management

- FR13: The system shall allow users to view their booking history.
- FR14: The system shall allow users to modify or cancel their reservations before a specified cut-off time.

3.2.6 Administrative Tools

- FR15: The system shall provide a dashboard for administrators to monitor system status and activities.
- FR16: The system shall generate reports on bookings, revenue, and system performance.

3.3 Non-functional Requirements

The non-functional requirements specify the quality attributes of the system, such as performance, security, and usability.

3.3.1 Performance Requirements

- NFR1: The system shall handle up to 10,000 concurrent users without performance degradation.
- NFR2: The system shall process transactions within 2 seconds under normal load conditions.

3.3.2 Security Requirements

- NFR3: The system shall comply with GDPR for data protection.
- NFR4: The system shall implement SSL/TLS for all data transmissions to ensure secure communication.
- NFR5: The system shall perform regular security audits and vulnerability assessments.

3.3.3 Reliability Requirements

- NFR6: The system shall have an uptime of 99.9% excluding scheduled maintenance.
- NFR7: The system shall provide failover mechanisms to ensure continuous availability.

3.3.4 Availability Requirements

- NFR8: The system shall be available 24/7, except during scheduled maintenance windows.
- NFR9: The system shall provide real-time updates on seat availability.

3.3.5 Maintainability Requirements

- NFR10: The system shall use modular architecture to facilitate easy updates and maintenance.
- NFR11: The system shall provide comprehensive documentation for administrators and developers.

3.3.6 Portability Requirements

- NFR12: The system shall be accessible via major web browsers (Chrome, Firefox, Safari, Edge) and on mobile devices (iOS and Android).
- NFR13: The system shall support easy deployment on different cloud platforms (AWS, Azure, Google Cloud).

4. Appendices

The appendices provide additional information to support the main content of the Software Requirements Specification (SRS) document for the Unified Seat Reservation System. This section includes a glossary, index, and references, offering clarity on terminology and further resources for understanding and implementing the system requirements.

A. Glossary

- API (Application Programming Interface): A set of rules and protocols for building and interacting with software applications.
- Authentication: The process of verifying the identity of a user or system.
- Authorization: The process of determining what actions an authenticated user or system can perform.
- CRUD (Create, Read, Update, Delete): Basic operations for managing data in a database.
- GDPR (General Data Protection Regulation): European Union regulation on data protection and privacy.
- GUI (Graphical User Interface): A visual way of interacting with a computer using items such as windows, icons, and menus.

- HTTPS (Hypertext Transfer Protocol Secure): An extension of HTTP for secure communication over a computer network.
- PCI DSS (Payment Card Industry Data Security Standard): A set of security standards designed to ensure that all companies that accept, process, store, or transmit credit card information maintain a secure environment.
- Seat Map: A graphical representation of the seating arrangement in a venue.
- SSL (Secure Sockets Layer): A standard security technology for establishing an encrypted link between a server and a client.
- TLS (Transport Layer Security): A cryptographic protocol designed to provide communications security over a computer network.
- UAT (User Acceptance Testing): A phase of software development in which the software is tested in the real world by the intended audience.

B. Index

    - HTTPS (3.3.2)
- I
    - Interactive Seat Map (3.2.3)
- N
    - Non-functional Requirements (3.3)
- P
    - PCI DSS (3.3.2)
    - Payment Processing (3.2.4)
- R
    - References (4.C)
    - Reports (3.2.6)
- S
    - Seat Map (A. Glossary)
    - Security Requirements (3.3.2)

- U
    - User Interface (3.1.1)
    - User Management (3.2.6)
    - UAT (A. Glossary)

C. References

1. ISO/IEC 27001:2013 - Information Security Management Standards
    - URL: ISO/IEC 27001:2013
    - Description: International standard for managing information security, providing guidelines for implementing and maintaining an effective ISMS.
2. PCI DSS - Payment Card Industry Data Security Standard
    - URL: PCI DSS
    - Description: Security standard for organizations handling branded credit cards from major card schemes, providing a framework for secure payment processing.
3. GDPR - General Data Protection Regulation

- URL: GDPR
- Description: Regulation in EU law on data protection and privacy for individuals within the European Union, addressing the transfer of personal data outside the EU and EEA areas.

4. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
   - URL: IEEE Std 830-1998
   - Description: Provides recommended practices for writing software requirements specifications, defining content and qualities of a good SRS document.
5. W3C Web Content Accessibility Guidelines (WCAG) 2.1
   - URL: WCAG 2.1
   - Description: Guidelines for making web content more accessible to people with disabilities, covering web content, web applications, and web tools.
6. RESTful Web Services: Design and Development
   - Author: Leonard Richardson, Sam Ruby
   - Publisher: O'Reilly Media
   - ISBN: 978-0596529260
   - Description: Book covering the principles and practices of designing RESTful web services.
7. JavaScript: The Good Parts
   - Author: Douglas Crockford
   - Publisher: O'Reilly Media
   - ISBN: 978-0596517748
   - Description: Book exploring the core features of JavaScript and providing best practices for using the language effectively.
   -
8. The Mythical Man-Month: Essays on Software Engineering
   - Author: Frederick P. Brooks Jr.
   - Publisher: Addison-Wesley
   - ISBN: 978-0201835953
   - Description: Collection of essays on software engineering, including the challenges of project management and team coordination.

9. Agile Software Development: Principles, Patterns, and Practices
    - Author: Robert C. Martin
    - Publisher: Prentice Hall
    - ISBN: 978-0135974445
    - Description: Comprehensive guide to agile development methodologies and practices.

Assignment

Design classes

The analysis model defines a set of analysis classes (chapter 10). Each of these classes describes some element of the problem domain, focusing on aspects of the problem that are user visible. The level of abstraction of an analysis class is relatively high.

As the design model evolves, you will defi ne a set of design classes that refi ne the analysis classes by providing design detail that will enable the classes to be implemented, and implement a software infrastructure that supports the business solution. Five different types of design classes, each representing a different layer of the design architecture, can be developed [amb01].

1. User interface classes: these classes define abstractions necessary for human-computer interaction (hci). They often implement the hci using metaphors and manage how users interact with the system.
2. Business domain classes: these classes refine the analysis classes related to elements of the business domain. They identify attributes and services (methods) needed to implement specific business functionalities.
3. Process classes: these classes implement lower-level business abstractions that manage the behaviour and interactions between business domain classes. They are crucial for the operational flow and logic of the application.

4. Persistent classes: these classes represent data stores such as databases. They manage data persistence beyond the execution of the software and encapsulate the logic for storing and retrieving data.
5. System classes: these classes implement software management and control functions necessary for the system to operate within its computing environment and interact with the outside world. They handle system-level concerns like communication protocols, hardware interfaces, etc.

Each type of design class plays a specific role in translating high-level analysis models into detailed design specifications that can be implemented. They encapsulate different aspects of the system's functionality and technical implementation details, ensuring that the system is both well-structured and maintainable over time.

Example related to a hypothetical software application for managing a library:

User Interface Classes:

- Example: Library UI, BookSearchDialog, MemberManagementForm
- Explanation: These classes define how users interact with the library management system. For instance, LibraryUI might handle overall navigation and display of different modules like book search, member management, and borrowing processes. BookSearchDialog could manage the interface for searching books by title, author, or category. MemberManagementForm might handle adding, editing, and deleting library members.

Business Domain Classes:

- Example: Book, Member, Loan, LibraryCard
- Explanation: These classes represent core entities and concepts within the library domain. For example, Book class would encapsulate attributes like title, author, ISBN, etc., and methods for managing books such as borrowing and returning. Member class would store member details like name, address, and methods for managing memberships. Loan class could handle loan transactions between members and books. LibraryCard might encapsulate the information related to a member's library card.

Process Classes:

- Example: LoanManager, ReservationProcessor, NotificationService
- Explanation: These classes implement the business logic and workflows of the library system. LoanManager could manage the process of borrowing and returning books, ensuring availability and due dates. ReservationProcessor might handle book reservations and notifications to members when reserved books become available. NotificationService could handle sending email or SMS notifications to members regarding due dates or overdue items.

Persistent Classes:

- Example: BookRepository, MemberRepository, LoanRepository
- Explanation: These classes interact with the data storage mechanisms (e.g., databases) to persist and retrieve data. BookRepository would handle CRUD (Create, Read, Update, Delete) operations for books in the database. MemberRepository would manage member data persistence. LoanRepository could handle storing loan records and their state (active, returned, overdue).

System Classes:

- Example: EmailService, DatabaseManager, SecurityManager
- Explanation: These classes provide system-level functionalities and integrations. EmailService manages sending emails for notifications. DatabaseManager encapsulates database connection and transaction management. SecurityManager handles authentication and authorization within the system, ensuring secure access to sensitive operations like borrowing books or managing member accounts.