Pillai Institute of Information Technology, Engineering, Media Studies & Research

SOFTWARE REQUIREMENTS SPECIFICATION

For

BLOG

Prepared by:-

Manoj kuamr R
Praveen VG
Rajeshwari M
Surendran S

Academic Year: *2023-2024*

Department of Information Technology
Pillai Institute of Information Technology, Engineering, Media Studies & Research

1. Introduction

1.1 Purpose

The primary purpose of a blog is to provide a platform for individuals, businesses, and organizations to share information, ideas, and opinions with a global audience. Blogs serve various purposes, including personal expression, education, entertainment, and marketing. They allow individuals to voice their thoughts and experiences, showcase their expertise, and connect with like-minded people. For businesses, blogs are an essential tool for promoting products and services, establishing authority in their industry, and engaging with customers. Blogs can also be used for journalism, reporting news, and offering in-depth analysis on various subjects. Overall, the purpose of a blog is to facilitate communication, information dissemination, and community building in the digital age.

1.2 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

1.3 Scope of Development Project

The scope of a blog development project entails a comprehensive plan to create, manage, and maintain a successful blog. It encompasses defining the blog's niche and target audience, selecting the appropriate platform, securing a domain name and hosting, customizing the design, producing high-quality content, optimizing for search engines, ensuring a seamless user experience, establishing a monetization strategy if applicable, building and engaging with the audience, implementing analytics tools for performance tracking, addressing security and maintenance needs, and ensuring legal compliance. The scope should also consider long-term goals and objectives for the blog, such as content planning and growth strategies. By defining these elements, a blog development project can provide a clear roadmap for achieving the desired outcomes.

1.4 Definitions, Acronyms and Abbreviations

L:	

JAVA :	
SQ	

For back-end development and platform independence.

For efficient database management.

SRS:	

UML :	
IDE:	

For modeling and documenting project components.

Such as Eclipse or IntelliJ IDEA for Java development. To define detailed project requirements and goals.

1.5 References

Books :

• Books on software development, project management, and productivity. • Online courses on web development, Java programming, React, and project management. Websites :

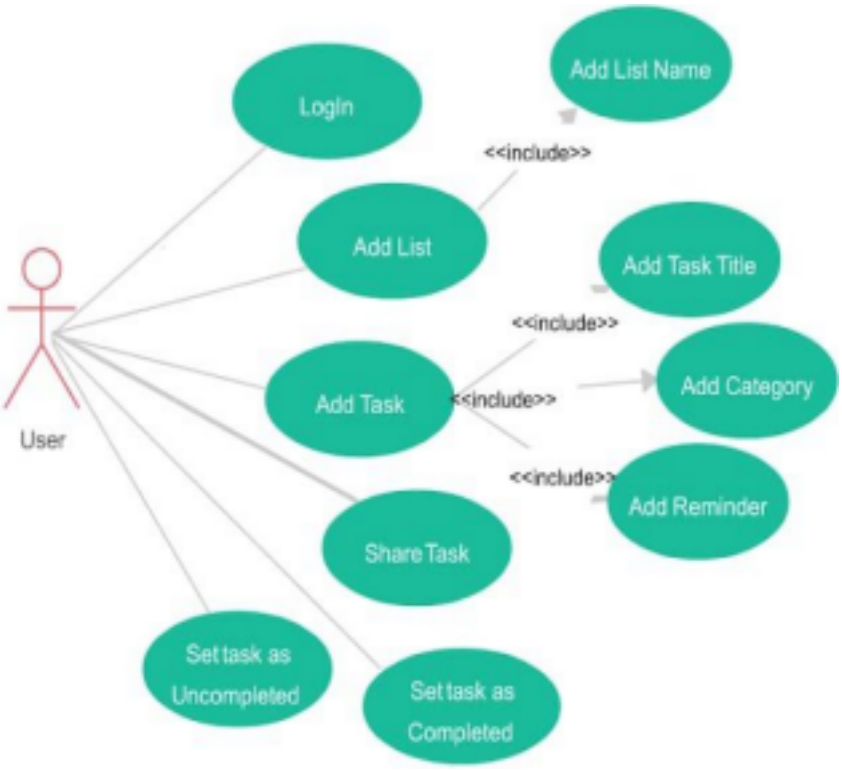
• •

Coursera	

2. Overall Descriptions

2.1 Product Perspective

Use Case Diagram

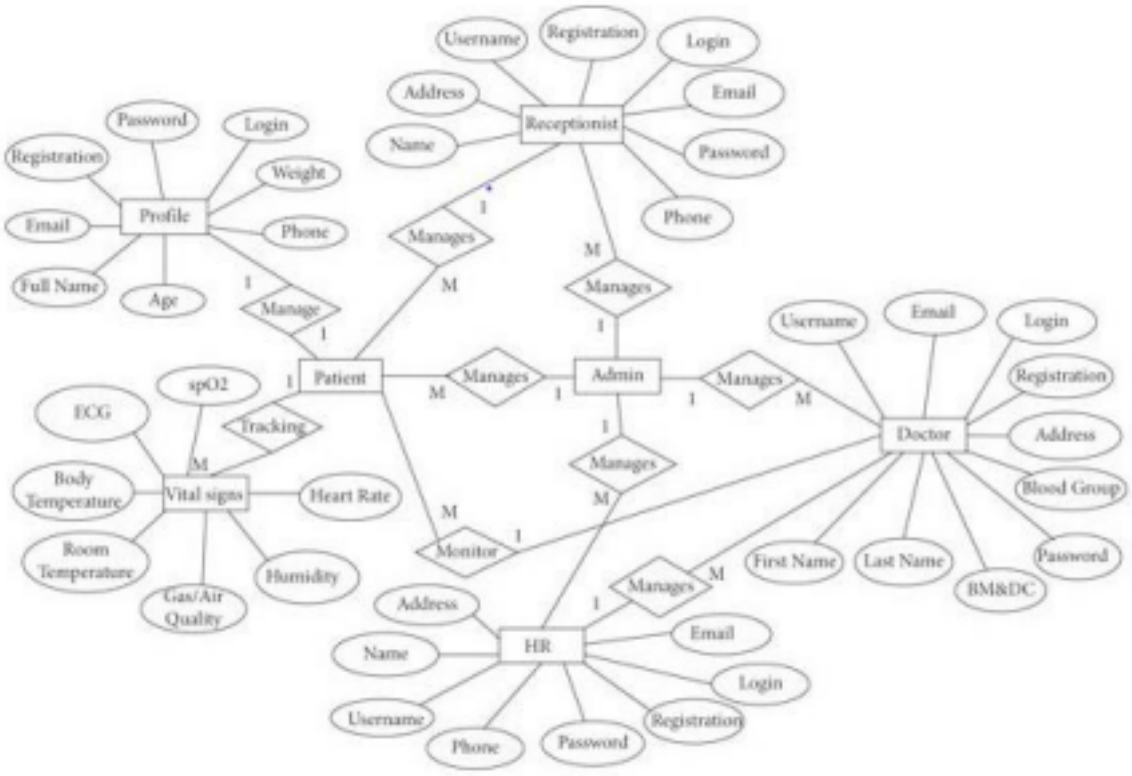


The use case diagram for an activity schedule management To-Do Planner illustrates interactions between actors and the system. The primary actor, "User," represents those utilizing the To-Do Planner. The system comprises key use cases like "Create Task," "Edit Task," "Delete Task," "Set Priority," "Set Due Date," "View Task List," "Organize Tasks," and "Receive Notifications." These use

cases encompass fundamental functionalities, facilitating efficient task and schedule management. Lines connecting the actor and use cases depict user actions and interactions, providing a visual framework to understand system functionalities and user roles, aiding in design and development processes.

2.2 Product Function

Entity Relationship Diagram



2.3 User Classes and Characteristics

Individual Users:

Char

These users are the core audience, managing personal tasks and schedules.

Nee ds:

Features for task creation, priority setting, due date management, and efficient organization of personal tasks are essential.

Team Users:

Char

Team users collaborate on tasks and projects with colleagues or team members.

- Needs:

Collaboration features such as task sharing, assignment, and real-time updates are crucial to facilitate teamwork.

Administrators:

- Characteristics:

Administrators manage user accounts and system settings.

- Needs:

User management tools, access control settings, and system configuration options are vital for effective system administration.

Guest Users:

- Characteristics:

Guest users have limited access without creating full accounts, often for external collaboration.

- Needs:

Ability to view shared tasks and some interaction capabilities are important to involve external parties.

Mobile Users:

• Characteristics:

Mobile users access the To-Do Planner through mobile apps for on-the-go task management.

•

Nee
ds:

Responsive design, mobile app availability, and mobile-optimized features are critical for their user experience.

Power Users:

•

Characteristi
cs:

Power users heavily rely on the To-Do Planner for complex task management.

•

Nee
ds:

Advanced features, extensive customization options, and integrations with other tools are essential to support their complex task management needs.

Notification-Dependent Users:

•

Characteristi
cs:

These users depend on notifications to stay updated on task deadlines and changes. •

Needs:

Customizable notification settings and timely alerts are crucial to help them manage their tasks effectively.

2.4 Operating Environment

The operating environment for a blog refers to the conditions and factors that influence its functionality and performance. Key elements of the operating environment include the choice of blogging platform (e.g., WordPress, Blogger) and the underlying technologies it relies on. The platform's compatibility with different web browsers and devices is crucial to ensure a consistent user experience. Hosting services play a vital role, impacting a blog's loading speed, uptime, and security. The availability of reliable internet connectivity and servers is essential for the blog's accessibility. Additionally, the regulatory and legal landscape, including copyright and privacy laws, can shape how content is created and shared on the blog. Understanding and adapting to this operating environment is critical for bloggers to create a successful and sustainable online presence.

2.5 Assumptions and Dependencies

Assumptions:

- Users have reliable internet access.

- Users use modern web browsers.
- Users have devices meeting basic requirements.
- Data privacy is a user expectation.

Dependencies:

- Third-party services for features.
- Browser updates for compatibility.
- ISP reliability for user access.
- Regulatory compliance for data protection.
- Hardware suppliers for equipment.
- Security updates for system integrity.
- User adoption for project success.

2.6 Requirement

Software

Configuration:

Front-End Tech supported by Sun M	
Integrated Development Environment (ID 7.0.1	

development.

Back-End Databa SQL Server fo	

• Operating System Compatibility: Windows NT, Windows 98, Windows XP.

Hardware Configuration:

• Processor: Pentium(R) Dual-core CPU for efficient processing.
• Hard Disk: A minimum of 40GB storage capacity for data and software.

• RAM: 256 MB or more for smooth application performance.

2.7 Data Requirement

Data requirements for a blog encompass the essential information and elements necessary to establish and maintain a functional and effective blog. These include content data, such as text, images, and multimedia materials, which form the core of blog posts. User data, including comments, interactions, and user-generated content, helps engage with the audience and tailor content to their needs. Metadata, such as titles, tags, and descriptions, aids in content organization and search engine optimization. Analytics data provides insights into blog performance and user behavior. Security data, like login credentials and security logs, is crucial for protecting the blog and user information. Legal data ensures compliance with copyright and privacy laws. Regular backups

of content and data are essential to prevent data loss. Monetization data, if applicable, helps track revenue and performance metrics for financial management. These data requirements collectively support the creation, management, and growth of a successful blog, enabling bloggers to connect with their audience and maintain the blog's integrity and security.

3. External Interface Requirement

1. User Data:

- User Profiles: Store user information, including usernames, passwords, and email addresses.
- User Preferences: Capture user-specific settings and preferences, such as notification preferences and time zone settings.

2. Task and Activity Data:

• Task Details: Record task names, descriptions, priorities, due dates, and categories. • Activity Schedules: Store details of scheduled activities, including dates, times, locations, and descriptions.

- Task/Activity Status: Track the status of tasks and activities (e.g., pending, completed, in progress).

3. Collaboration Data (if applicable):

- User Collaborations: Maintain information about shared tasks or activities, including collaborators' details.
- Real-Time Updates: Record real-time updates on shared tasks to ensure collaboration transparency.

4. Notifications and Reminders:

- Notification Records: Store information about notifications, including content and delivery timestamps.
- Reminder Settings: Capture user-defined reminder settings for tasks and activities.

5. Search and Filter Data:

• Search Queries: Record user search queries and results for tasks and activities. • Filter Criteria: Store user-defined filtering criteria for organizing tasks and activities.

6. Calendar Integration (if applicable):

• Calendar Events: Store data related to synchronized events with external calendars.

7. Audit and Logging (for system monitoring):

• Activity Logs: Maintain logs of user actions and system events for monitoring and troubleshooting.

8. System Configuration and Settings:

• System Configuration: Store system-wide settings and configurations.
• Security Settings: Capture security-related configurations and access controls.

9. Backup and Recovery Data:

• Backup Records: Maintain records of system backups, including timestamps and data integrity checks.

4. System Features

Task Management:

➢ Users can create, edit, and organize tasks, setting due dates and priorities for effective organization.

Activity Scheduling:

➢ Schedule and manage activities with details like dates, times, locations, and descriptions, viewable in a calendar format.

➢ Facilitate task sharing, assignment, and real-time

User Collaboration (if applicable):

updates for teams collaborating on projects.

➢ Automated notifications keep users informed about

Notifications and Reminders:

upcoming tasks and activities.

➢ Users can search for tasks and activities by keywords and apply

Search and Filtering:

custom filters for efficient organization.

➢ Regular data backups and a recovery mechanism are in

Data Backup and Recovery:

place to safeguard against data loss.

5. Other Non-functional Requirements

➢

Performance:

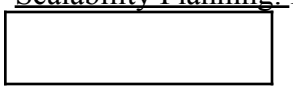
• Load Balancing: Implement load balancing to distribute user requests evenly across

multiple servers, optimizing performance.

• Caching Mechanisms: Utilize caching for frequently accessed data to reduce database

load and improve response times.

• Scalability Planning: Develop a clear plan for scaling resources as user numbers grow,

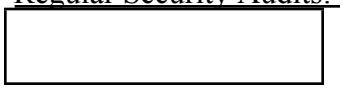


including provisioning additional servers or cloud resources.

➢

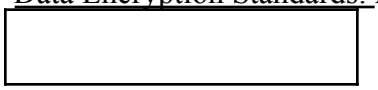
Securi
ty:

• Regular Security Audits: Conduct periodic security audits and vulnerability assessments



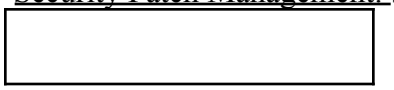
to identify and mitigate potential threats.

• Data Encryption Standards: Adhere to industry-standard encryption protocols (e.g.,



SSL/TLS) for data in transit and at rest.

• Security Patch Management: Stay up-to-date with security patches and updates for all

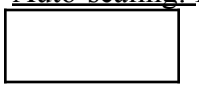


system components to address known vulnerabilities.

➢

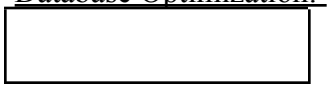
Scalabili
ty:

• Auto-scaling: Implement auto-scaling mechanisms that automatically adjust resource



allocation based on real-time usage metrics.

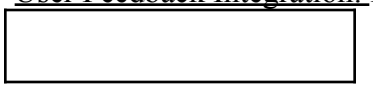
• Database Optimization: Optimize database queries and indexing to ensure efficient data



retrieval and storage as the user base expands.

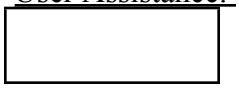
➢ User-Friendly Interface:

• User Feedback Integration: Provide a feedback mechanism within the interface to gather



user input for continuous improvement.

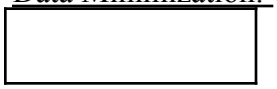
• User Assistance: Offer contextual help, tooltips, and user guides to assist users in



navigating and utilizing the system effectively.

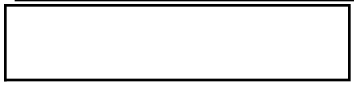
➢ Data Privacy and Compliance:

• Data Minimization: Collect and store only the minimum necessary user data required



for system functionality.

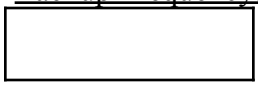
• User Data Access Control: Implement user-controlled data access settings, allowing



users to manage who can view their information.

➢ Data Backup and Recovery:

• Backup Frequency: Specify the frequency of backups based on data change rates,



ensuring minimal data loss in case of failures.

• Backup Storage: Safely store backups in geographically distributed locations to mitigate

data loss risks.

➢ Accessibility:

• Screen Reader Compatibility: Ensure compatibility with popular screen reader software

for users with visual impairments.

• Keyboard Navigation: Enable full keyboard navigation for users who rely on keyboard

input.

➢ Reliability and Availability:

• Service Level Agreements (SLAs): Define SLAs that guarantee a certain level of uptime

and performance for users.

• Proactive Monitoring: Implement proactive monitoring tools that detect and address

issues before they impact users

6. Other Requirements

6.1 Data and Category Requirements

The system will have various categories of users, including individuals managing activities, team members, administrators, and others. Each category of users will have specific access rights. Administrators will have the authority to modify, delete, and append data, while other users, except for specific roles like managers, will generally have read-only rights for data retrieval. Additionally, the system will include different categories of activities and tasks, and the relevant data associated with each category will be presented in a structured format.

6.2 Appendix

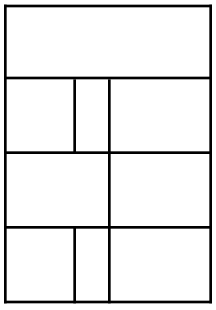
A: Admin, Abbreviation, Acronym, Assumptions				
C: Categories, Classifications, Client				
D: Data Requirements, Dependencies				
G: GUI (Graphical User Interface)				
K: Key Features				
U: User, User Classes and Characteristics, User Requirements				

-
-
-
-
-
-

6.3 Glossary

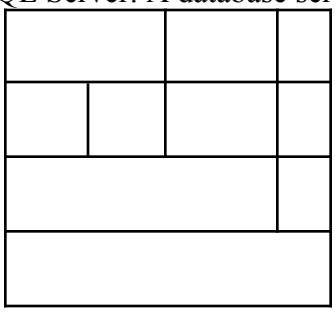
The following glossary provides definitions and explanations for key terms and acronyms used in this document and the project:

- Administrator: A user role with administrative privileges for system management.

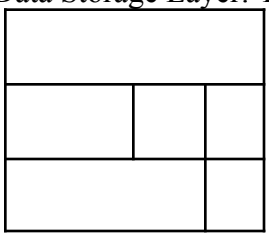


- User: A general login role assigned to most system users.
- Client: Refers to the intended users or stakeholders of the system.

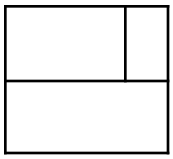
• SQL: Abbreviation for Structured Query Language, used for querying and retrieving data from databases.
• SQL Server: A database server used for organized data storage.



• Layer: Represents different sections or components of the project.
• User Interface Layer: The part of the system that users directly interact with. • Application Logic Layer: The web server component responsible for processing computations and logic.
• Data Storage Layer: The component where data is stored and managed.



• Use Case: A high-level diagram illustrating the system's functionality and interactions. • Class Diagram: A static structure diagram depicting the system's classes, attributes, and relationships.
• Interface: A means of communication between different system components.



• Unique Key: A database attribute used to distinguish entries uniquely.

6.4 Class Diagram

Department of Information Technology
Pillai Institute of Information Technology, Engineering, Media Studies & Research

A class diagram represents the structure of a system by showing classes, their attributes, and relationships between classes. In the context of "Activity Schedule Management," certain main classes, such as "Tasks," "Activities," and "Users," are central to the system's functionality. Relationships between these classes, including associations, aggregations, and generalizations, are

depicted with role names and multiplicities to illustrate how data is organized and accessed within the system.