SOFTWARE ENGINEERING & CONCEPTS – LAB MANUAL

Work Planner Application

Varun Kumar V, 220701311

BE CSE / II YEAR / E section

Software Concepts & Engineering - Lab Manual

Overview of the Project

The Juggle is Real

We've all been there. Juggling work deadlines, personal errands, and that nagging feeling we're forgetting something important. It's a constant battle to stay organized and on top of everything. This work planner aims to be your secret weapon in that battle.

Taking Control of Your Time

The problem is clear: without a structured plan, tasks pile up, priorities get muddled, and stress levels soar. Traditional methods often lack the flexibility to handle both personal and professional commitments effectively.
This planner tackles that head-on by leveraging data to create a powerful, yet user-friendly, organizational tool.

Data: The Power Behind the Plan

The planner gathers various types of data to provide a holistic view of your commitments:

Task details: Deadlines, priorities, dependencies – everything you need to know about what needs to be done.

Schedule data: Your calendar is integrated, ensuring tasks are realistically slotted within your available time.
Resource tracking: No more scrambling for files or tools. The planner helps you identify and access the resources needed for each task.

Progress tracking: Stay motivated and accountable with metrics on task completion, time spent, and overall productivity.

Features for a Smoother Workflow

Imagine a system that:

Unifies your task list: No more bouncing between personal to-do lists and work schedules. See everything in one place, with the option to filter by category.
Suggests smart schedules: Don't waste time figuring out when to tackle what. The planner analyzes your workload and suggests optimal scheduling based on priorities and deadlines.

Manages your resources: Need a specific document or contact for a task? The planner helps you locate and access them seamlessly.

Tracks your progress: See how you're doing with clear dashboards and reports. Analyze your productivity and identify areas for improvement.

Benefits You Can Feel

By implementing this work planner, you'll experience:

Increased productivity: Efficient time management and clear prioritization lead to getting more done in less time.
Reduced stress: Stay organized and focused, eliminating the mental strain of juggling tasks.

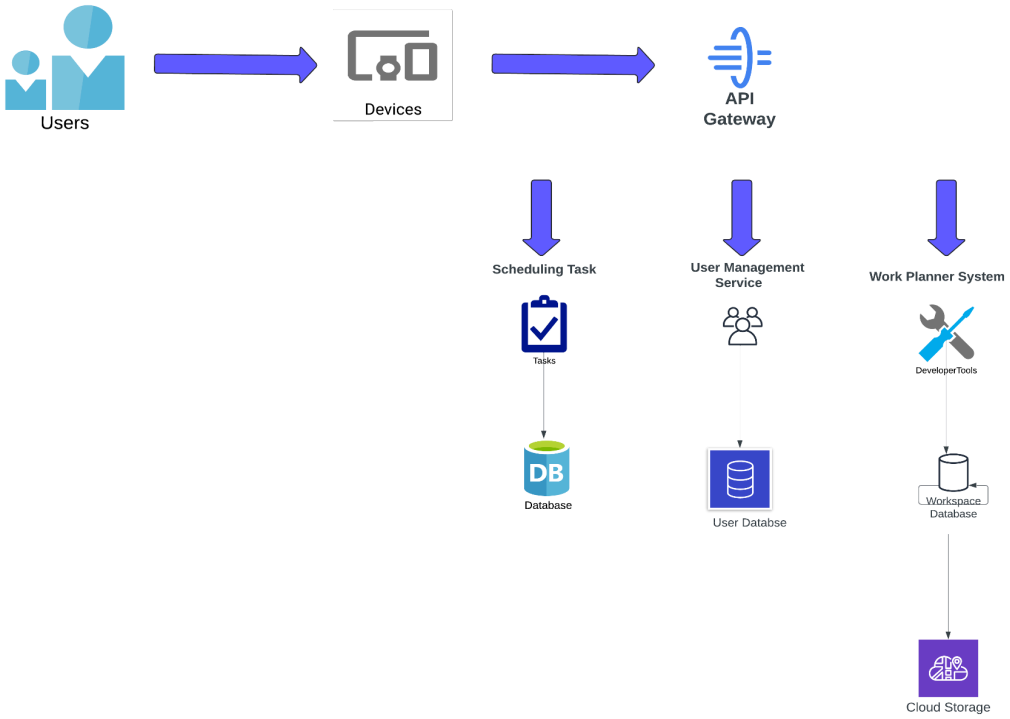
Improved work-life balance: Clearly separate personal and professional commitments, allowing you to reclaim control of your time.

Enhanced accountability: Progress tracking keeps you motivated and helps identify areas for growth.
Streamlined workflow: Easy access to resources ensures tasks are completed without unnecessary delays.

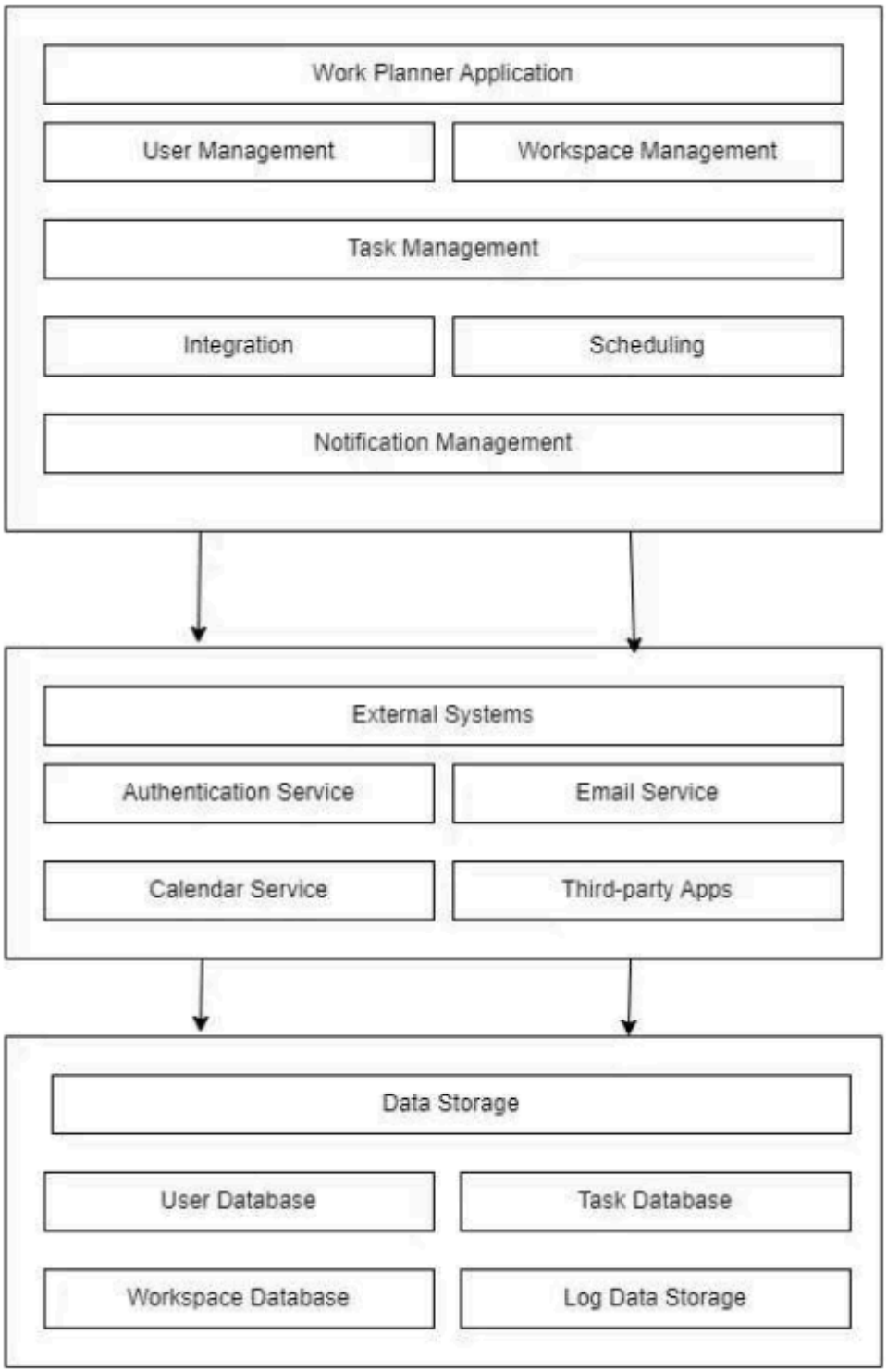
The Bottom Line

This work planner isn't just about managing tasks, it's about empowering you to manage your time effectively. It's about reducing stress, achieving a better work-life balance, and ultimately reaching your full potential.

DEPLOYMENT DIAGRAM:



Business Architecture Diagram



Business Need: The Work Planner Application is likely designed to streamline and optimize the planning, scheduling, and management of work tasks or projects within an organization. It aims to improve productivity, efficiency, and collaboration among teams or departments by providing a centralized platform for coordinating and tracking work activities.

Current Process (Manual or Automatic): The current process for work planning and scheduling might involve various manual steps, such as:

1. Manual task creation and assignment
2. Spreadsheet-based tracking of tasks and deadlines
3. Email or in-person communication for updates and coordination
4. Paper-based documentation or whiteboards for visualization

Alternatively, if an existing automated system is in place, it may suffer from limitations, such as lack of integration, poor user experience, or insufficient features to meet the organization's evolving needs.

Different Personas and Their Roles:

1. **Project Managers:** Responsible for creating and assigning tasks, setting deadlines, monitoring progress, and ensuring project completion.
2. **Team Leaders:** Oversee the work of their respective teams, assign tasks to team members, and track progress.
3. **Team Members:** Receive assigned tasks, update their progress, and collaborate with other team members.
4. **Executives or Stakeholders:** Monitor high-level project status, resource allocation, and overall performance.

Business Problems: The Work Planner Application likely aims to address several business problems, such as:

1. Inefficient task management and coordination across teams or departments.
2. Lack of visibility into project progress, bottlenecks, or resource constraints.
3. Difficulty in prioritizing tasks and managing deadlines effectively.
4. Limited collaboration and communication among team members.
5. Inability to adapt quickly to changing priorities or requirements.
6. Inefficient resource allocation and utilization.

Business Architecture Diagram: The business architecture diagram for the Work Planner Application would typically depict the various components, layers, and interactions within the system. It may include elements such as:

1. **User Interface Layer:** Representing the front-end interfaces for different user roles (e.g., project managers, team members).
2. **Business Logic Layer:** Handling the core functionality of task creation, assignment, scheduling, and progress tracking.
3. **Data Access Layer:** Interacting with the underlying data storage systems (e.g., databases) for persisting and retrieving project and task information.
4. **Integration Layer:** Facilitating integration with other systems or tools (e.g., communication tools, reporting tools).
5. **Security and Access Control:** Ensuring proper authentication, authorization, and data protection mechanisms.
6. **Reporting and Analytics:** Providing insights, dashboards, and reports for monitoring project performance and resource utilization.

Requirements as User Stories

Architecture Diagram depicting the

- different modules, their interactions, error handling, logging, data storage etc
- Architecture pattern used and why
- Design principles used and why

- Class diagrams
  - For all entities described in the Business Architecture Diagram, their relationships with other entities
  - Should depict the different attributes and its methods
- Sequence diagram for atleast 5 of the user stories documented above

Test Strategy

- Document the Test Plans
- Test cases for atleast 5 user stories showcasing the Happy Path and the Error Scenarios
- A view of the github repository showcasing the Project structure, their naming conventions
- A view of their DevOps Architecture for their respective project and the associated tools used in Azure

Deployment Architecture of the application

Usability: The Work Planner will be designed with a user-friendly interface to ensure ease of use. The application will feature intuitive navigation, clear instructions, and accessible design elements to cater to users of all technical

proficiency levels.

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to detail the requirements for the Work Planner application. The goal of the Work Planner is to facilitate efficient project management by providing features such as task creation, milestone tracking, and task analysis. This document serves as a guideline for developers and stakeholders to ensure the application meets user needs and expectations.

2. Scope

The Work Planner is designed to support users in managing tasks and projects efficiently. It will cater to individual users and teams across various industries, offering functionalities like task creation, task management, milestone tracking, and report analysis. The application will be accessible on both desktop and mobile platforms, ensuring that users can manage their projects anytime and anywhere.

3. Functional Requirements

3.1 User Authentication

The Work Planner will include a secure user authentication system to ensure that only authorized users can access the application. Users will register with their email and a password, and the system will support multi-factor authentication (MFA) for enhanced security. The application will also include a password recovery feature to help users regain access if they forget their login credentials.

3.2 Task Creation

Users will be able to create tasks easily within the Work Planner. Each

task will include details such as title, description, due date, priority level, and assigned team members. The task creation process will be streamlined to ensure users can quickly input the necessary information and start managing their tasks efficiently.

3.3 Task Editing and Management

The application will allow users to edit and manage their tasks seamlessly. Users can update task details, change due dates, adjust priorities, and reassign tasks to different team members as needed. This functionality ensures that tasks can be kept up-to-date and accurately reflect the current state of the project.

3.4 Export and Sharing

The Work Planner will offer export and sharing capabilities for tasks and reports. Users can export their data in various formats, including PDF, CSV, and Excel. Additionally, the application will allow users to share tasks and reports via email or direct links, facilitating collaboration and communication among team members.

3.5 Search and Filtering

The application will include robust search and filtering options to help users quickly locate specific tasks or projects. Users can search by keywords, due dates, priority levels, or assigned team members. Filtering options will enable users to narrow down their search results, making it easier to find relevant information and manage their workload effectively.

3.6 Help and Support

The Work Planner will provide comprehensive help and support to assist users in utilizing the application effectively. This will include a detailed user manual, FAQs, and a support ticket system for addressing technical issues. Users will also have access to live chat support and community forums to seek advice and share experiences with other users.

4. Non-Functional Requirements

4.1 Performance

The Work Planner must perform efficiently, ensuring quick response times and minimal latency. The application should handle a large number of users and tasks without significant performance degradation. Performance optimization will be a key focus to ensure the application remains fast and reliable under various conditions.

4.2 Security

Security is a critical aspect of the Work Planner. The application will implement robust security measures, including data encryption, secure communication protocols, and regular security audits. User data will be protected against unauthorized access and breaches, ensuring compliance with industry standards and regulations for data protection and privacy.

4.3 Usability

The Work Planner will be designed with a user-friendly interface to ensure ease of use. The application will feature intuitive navigation, clear instructions, and accessible design elements to cater to users of all technical proficiency levels. Continuous user feedback will be gathered and analyzed to make iterative improvements to the application's usability.

4.4 Compatibility

The application will be compatible with various operating systems and devices, including Windows, macOS, iOS, and Android. It will support major web browsers to ensure users can access the application from any device with an internet connection. Comprehensive compatibility testing will be conducted to verify seamless performance across different platforms and devices.

5. Constraints

The development of the Work Planner will be subject to constraints such as budget limitations, time constraints, and resource availability. The project will adhere to a defined timeline and budget to ensure timely and cost-effective delivery. Additionally, the application must comply with relevant legal and regulatory requirements, including data protection laws.

6. Conclusion

This SRS document outlines the essential requirements for the development of the Work Planner application. By adhering to these specifications, the development team can create a robust, secure, and user-friendly application that meets the needs of its users. The Work Planner aims to enhance productivity and streamline project management processes, ultimately contributing to the success of individuals and teams in various industries.

SCRUM METHODOLOGY for Work Planner Project

1. The Project Vision

Vision Statement

To create a comprehensive Work Planner that enables users to efficiently manage tasks and projects through a user-friendly interface with features such as task creation, milestone tracking, and task analysis.

Goals

- User-friendly interface
- Centralized project management
- Task creation and editing

- Milestone tracking
- Task timers
- Subtask management
- Task report analysis
- Task reminders and duration setup

2. The Product Backlog

The product backlog is a prioritized list of features, enhancements, and bug fixes required for the Work Planner project.

Product Backlog Items

- User Registration and Authentication
- Task Creation and Management
- Project Activity Management from a Central Place
- Milestone Tracking
- Task Timer Activation
- Subtask Management
- Task Report Analysis
- Task Reminders and Duration Setup
- Export Options for Task Reports
- User Feedback and Rating System
- Admin Panel for Managing Project Activities

3. The Scrum Team

Roles

- Product Owner: Defines the product features and prioritizes the backlog.
- Scrum Master: Ensures the Scrum process is followed, removes impediments, and facilitates meetings.
- Development Team: A cross-functional team responsible for delivering potentially shippable increments at the end of each sprint (includes developers, designers, QA, etc.).

4. Planning the Sprints

-Sprint Duration: Typically 2-4 weeks.

-Sprint Planning Meeting: The team selects items from the product backlog to commit to during the sprint.

5. Sprint Planning Meeting

-Goal: Define what will be delivered in the sprint and how it will be achieved.

-Input: Product backlog, team capacity, past performance.

-Output: Sprint backlog (tasks for the sprint), sprint goal.

6. Daily Stand-up Meetings

-Duration: 15 minutes

-Purpose : Discuss what was done yesterday, what will be done today, and identify any impediments.

7. Sprint Execution

-Development: Team works on the tasks in the sprint backlog.

-Testing: Continuous integration and testing of features.

8. Sprint Review

-Purpose: Demonstrate the working product increment to stakeholders.

-Activities: Team shows what was accomplished during the sprint. Stakeholders provide feedback.

9. Sprint Retrospective

-Purpose: Reflect on the sprint and identify improvements for future sprints.

-Activities: Discuss what went well, what didn't, and how to improve.

10. Release Planning

-Release Goal: Determine when and what features will be released to the users.

-Activities: Prioritize features, finalize the release date, prepare for deployment.

By adhering to these Scrum principles and processes, the Work Planner project aims to deliver a high-quality product that meets user needs and enhances project management efficiency.

Sprint Breakdown for Work Planner Project

Sprint 1: Basic Framework and User Authentication

- **Task 1:** Set up the project repository and initial project structure.
- **Task 2:** Implement user registration and authentication.
- **Task 3:** Create a basic user profile page.

Sprint 2: Task Creation and Project Management

- **Task 1:** Develop the task creation feature.
- **Task 2:** Implement project activity management from a central place.
- **Task 3:** Enhance the user profile with additional project management capabilities.

Sprint 3: Task Viewing and Milestone Tracking

- **Task 1:** Implement task viewing and management.
- **Task 2:** Develop milestone tracking.

- **Task 3:** Add subtasks functionality.

Sprint 4: Task Timers and Report Analysis

- **Task 1:** Implement task timer activation.
- **Task 2:** Develop task report analysis tools.
- **Task 3:** Implement real-time updates for tasks.

Sprint 5: Export Options and Reminder Setup

- **Task 1:** Implement export options for task reports.
- **Task 2:** Set up task reminders and duration settings.
- **Task 3:** Integrate with external tools and services.

Sprint 6: User Feedback and Admin Panel

- **Task 1:** Create a user feedback system.
- **Task 2:** Develop an admin panel for managing project templates.
- **Task 3:** Perform final testing and bug fixing.

REQUIREMENTS AS USER STORIES FOR WORK PLANNER APPLICATION

Work Planner Team

Backlog | Analytics | New Work Item | View as Board | Column Options | ... | Epics

Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
1	Epic	Good UI/UX	Active			Business	UI/UX
2	Epic	Login Page	New			Business	
3	Epic	Integration and Synchronization	New			Business	
4	Epic	Workspace Management	New			Business	
5	Epic	Backup/Restore data	New			Business	
6	Epic	Customization and Preferences	New			Business	
7	Epic	Interactive Schedule Management	New			Business	
8	Epic	Daily Schedule Visualization	New			Business	
9	Epic	Rewarding System	New			Business	

Work Planner Team

Backlog | Analytics | New Work Item | View as Board | Column Options | ... | Epics

Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
1	Epic	Good UI/UX	Active			Business	UI/UX
	Feature	Modern UI/UX	New			Business	
	User Story	As a user, I'm bored with cluttered and old UI's, so ...	New			Business	
	Task	Design UI/UX	New				
	Feature	The login page should have login details like usernam...	New			Business	
	User Story	As a user, I should be able to login using username ...	New			Business	
	Task	Design Login Layout	New				
	User Story	As a user, after logging in, I should have the option ...	New			Business	
	Task	Design Menu	New				
2	Epic	Login Page	New			Business	
	Feature	Forgot password/ password reset/ Register	New			Business	
	User Story	As a user, I sometime may forgot my password, So t...	New			Business	
	Task	Make forgot/Reset password option	New				
	Feature	Easy login	New			Business	
	User Story	As a user, I sometimes feel sign-in/sign-up options ...	New			Business	
	Task	Easy authorization	New				
3	Epic	Integration and Synchronization	New			Business	
	Feature	Extract meeting details	New			Business	

Work Planner Team

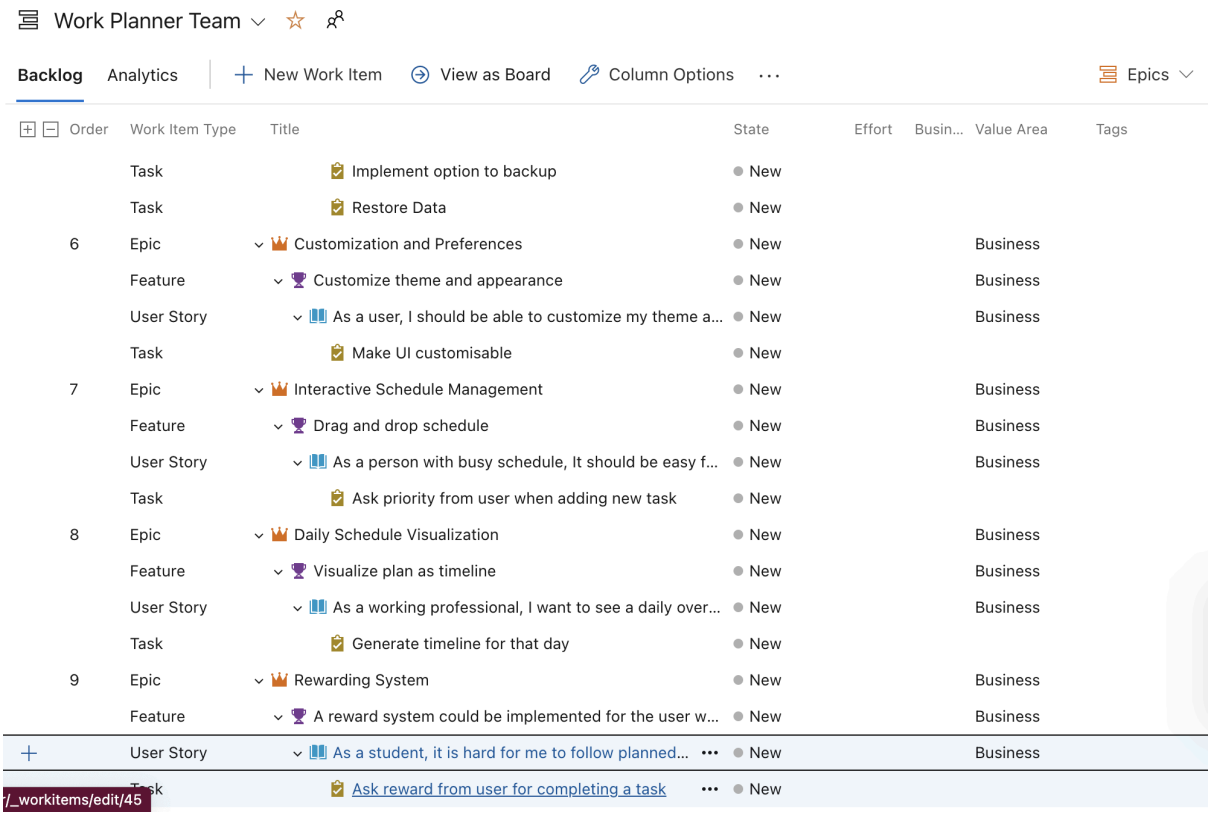
Backlog | Analytics | New Work Item | View as Board | Column Options | Epics

Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
3	Epic	Integration and Synchronization	New			Business	
	Feature	Extract meeting details	New			Business	
	User Story	As a person working in an organization, I want to att...	New			Business	
	Task	Extract meeting details from mails	New				
	Task	Notify User	New				
	Feature	Linking apps related to particular task	New			Business	
	User Story	As a user, I may find it difficult to navigate to va...	New			Business	
	Task	Link app	New				
	Task	Open App	New				
4	Epic	Workspace Management	New			Business	
	Feature	Create Workspace	New			Business	
	User Story	As a user, I want to create a new workspace so that ...	New			Business	
	Task	Implement backend API endpoint for creating a w...	New				
	Task	Develop frontend UI for creating a workspace	New				
	Task	Validate user input for workspace name and desc...	New				
	Task	Test creation of workspace by verifying it appear...	New				
	Feature	View Workspaces	New			Business	
	User Story	As a user, I want to view all the workspaces I've cre...	New			Business	

Work Planner Team

Backlog | Analytics | New Work Item | View as Board | Column Options | Epics

Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
	Feature	View Workspaces	New			Business	
	User Story	As a user, I want to view all the workspaces I've cre...	New			Business	
	Task	Fetch user's workspaces from the backend	New				
	Task	Display the list of workspaces in the frontend UI	New				
	Task	Handle cases where there are no workspaces to ...	New				
	Task	Test the display of workspaces by verifying their ...	New				
5	Epic	Backup/Restore data	New			Business	
	Feature	Option to take backup securely	New			Business	
	User Story	As a user, I may delete or my system may crash, So ...	New			Business	
	Task	Implement option to backup	New				
	Task	Restore Data	New				
6	Epic	Customization and Preferences	New			Business	
	Feature	Customize theme and appearance	New			Business	
	User Story	As a user, I should be able to customize my theme a...	New			Business	
	Task	Make UI customisable	New				
7	Epic	Interactive Schedule Management	New			Business	
	Feature	Drag and drop schedule	New			Business	
	User Story	As a person with busy schedule, It should be easy f...	New			Business	
	Task	Ask priority from user when adding new task	New				



User Story 1: As a User, I should be able to reset my password, choose "Forgot my Password", or register if new to the app**.**

Estimation of complexity using poker planning Methodology : 5 points

User Story 2: As a user, after my successful login or registration, I should be able to choose my workspace needed.

Estimation of complexity using poker planning Methodology : 3 points

User Story 3: As a user of this app, I should be able to select the app

to be integrated from dropdown options on the home page.

Estimation of complexity using poker planning Methodology : 3 points

User Story 4: As a user, after successful login and workspace selection, I should be able to see a home page where the scheduling page appears with work scheduled in a calendar.

Estimation of complexity using poker planning Methodology : 8 points

User Story 5: As a user, I should be able to create a new task and assign it to a team member.

Estimation of complexity using poker planning Methodology : 5 points

User Story 6: As a user, I should be able to update the status of a task (e.g., from "In Progress" to "Completed").

Estimation of complexity using poker planning Methodology : 3 points

User Story 7: As a user, I should receive notifications and reminders for upcoming tasks and deadlines.

Estimation of complexity using poker planning Methodology : 5 points

User Story 8: As a user, I should be able to view and edit my profile information.

Estimation of complexity using poker planning Methodology : 3

points

User Story 9: As a user, I should be able to view a dashboard that summarizes my tasks and schedule.

Estimation of complexity using poker planning Methodology : 8 points

User Story 10: As a manager, I should be able to view and generate reports on team performance and task completion.

Estimation of complexity using poker planning Methodology : 8 points

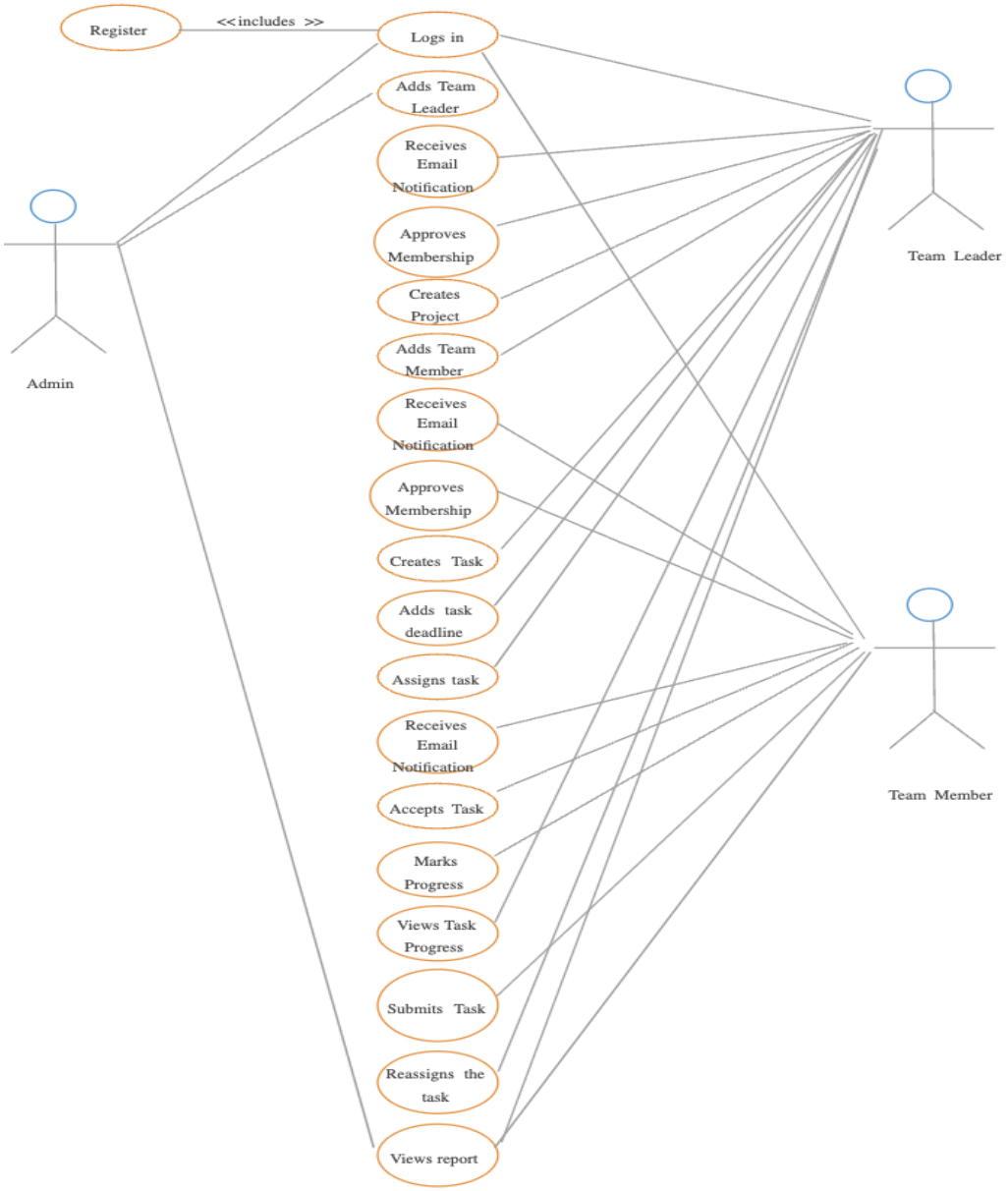
Non-Functional Requirements:

Performance: The system should handle up to 10,000 concurrent users with an average response time of less than 2 seconds for any operation.

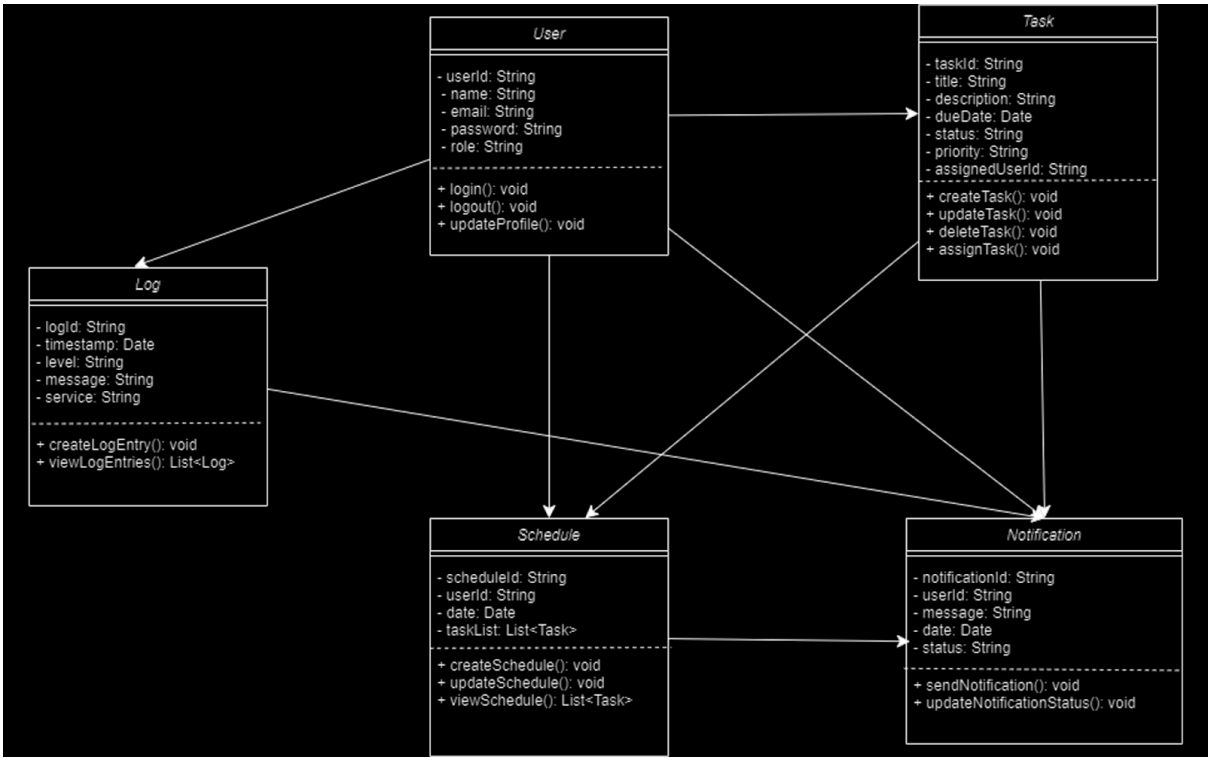
Security: All user data must be encrypted both at rest and in transit, and the system must comply with GDPR and other relevant data protection regulations.

Availability: The system should have an uptime of 99.9% to ensure it is accessible to users at all times.

Use case diagram:



CLASS DIAGRAM:



Class Diagram Overview for Work Planner Project

The class diagram depicts the structure of the Work Planner application, showcasing the classes involved, their attributes, methods, and relationships. It represents how different components of the system interact to create and manage tasks and projects.

Explaining Class Components

1. **WorkPlanner**
   - **Methods:** createTask(), manageProjectActivities()
   - **Purpose:** Main system class responsible for initiating task creation and managing project activities from a central place.
2. **Task**
   - **Attributes:** title, description, dueDate, priority, status, estimatedDuration
   - **Methods:**
     - **Task(String title, String description, Date dueDate, String priority):** Constructor.

        - **updateTask(String title, String description, Date dueDate, String priority, String status):** Updates task details.
        - **setReminder(Date reminder):** Sets a reminder for the task.
    - Purpose: Represents individual tasks within the work planner.
3. **Project**
    - **Attributes:** name, description, startDate, endDate
    - **Methods:**
        - **Project(String name, String description, Date startDate, Date endDate):** Constructor.
        - **addTask(Task task):** Adds a task to the project.
        - **getTasks():** Retrieves all tasks associated with the project.
    - **Purpose:** Represents a project that contains multiple tasks and manages project activities.
4. **User**
    - **Attributes:** id, username, password
    - **Methods:**
        - **User(String username, String password):** Constructor.
        - **createProfile(String username, String password):** Creates a user profile.
        - **login(String username, String password):** Logs in the user.
        - **updateProfile(String username, String password):** Updates user profile details.
    - **Purpose:** Represents users of the application who can create and manage tasks and projects.
5. **Milestone**
    - **Attributes:** name, description, dueDate
    - **Methods:**
        - **Milestone(String name, String description, Date dueDate):** Constructor.
        - **updateMilestone(String name, String description, Date dueDate):** Updates milestone details.
        - **getMilestoneStatus():** Retrieves the status of the milestone.
    - **Purpose:** Represents significant points in a project timeline to track progress.
6. **Subtask**
    - **Attributes:** title, description, status
    - **Methods:**
        - **Subtask(String title, String description):** Constructor.
        - **updateSubtask(String title, String description, String**

status): Updates subtask details.
    - **markAsComplete():** Marks the subtask as complete.
  - **Purpose:** Represents smaller tasks within a main task to break down work into manageable parts.

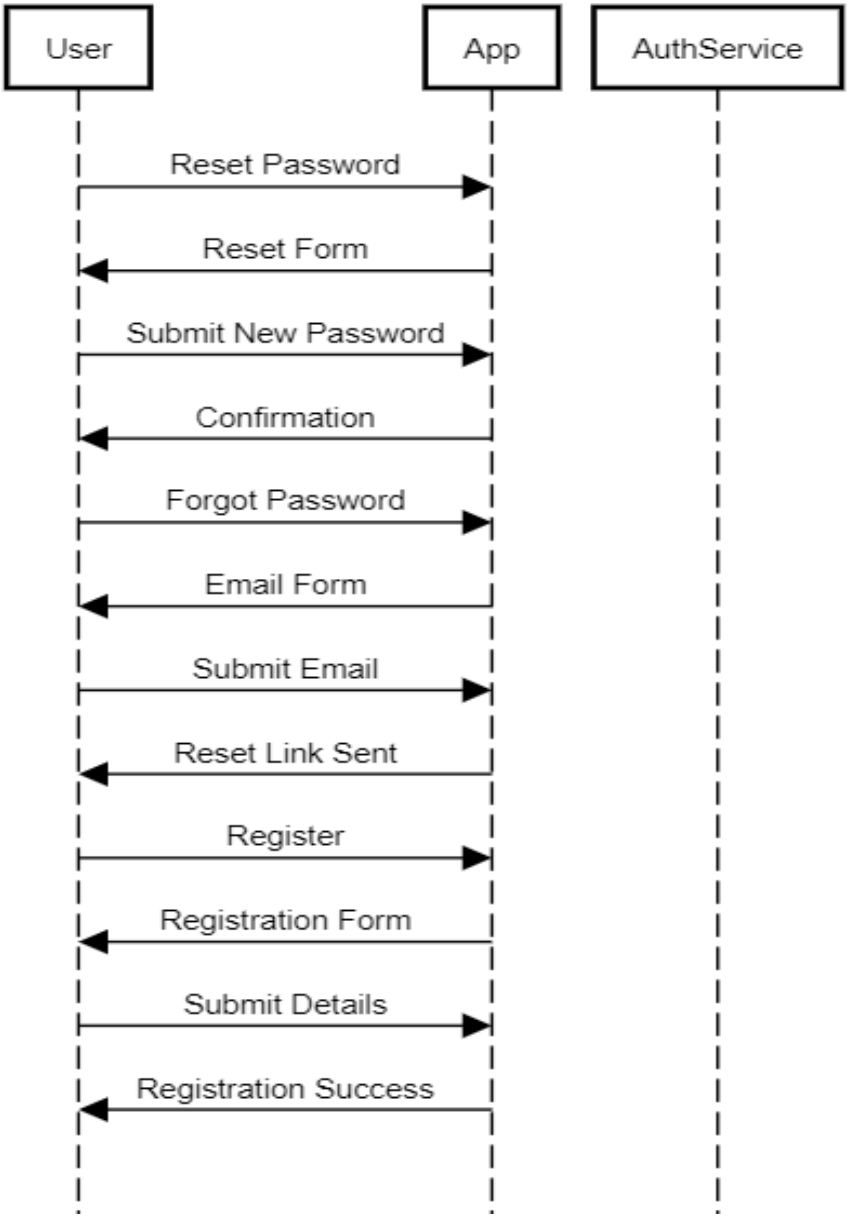
7. **Timer**
   - **Attributes: s**tartTime, endTime, duration
   - **Methods:**
     - **startTimer():** Starts the timer for a task.
     - **stopTimer():** Stops the timer and records the duration.
     - **getTimerDetails():** Retrieves details of the timer.
   - Purpose: Tracks the time spent on individual tasks.
8. **Report**
   - **Attributes:** reportId, reportData
   - **Methods:**
     - **generateReport(String reportType):** Generates a task report based on specified criteria.
     - **getReportDetails():** Retrieves details of the generated report.
   - **Purpose:** Analyzes task data and generates reports to evaluate productivity.

Relationships

- User to Task: Users can create, update, and manage multiple tasks.
- User to Project: Users can create, update, and manage multiple projects.
- Project to Task: A project aggregates multiple tasks.
- Task to Subtask: A task aggregates multiple subtasks.
- Task to Timer: A task can have an associated timer to track time spent.
- Project to Milestone: A project can have multiple milestones.
- Task to Report: Task data is used to generate various reports.

SEQUENCE DIAGRAM:

USER STORY 1:

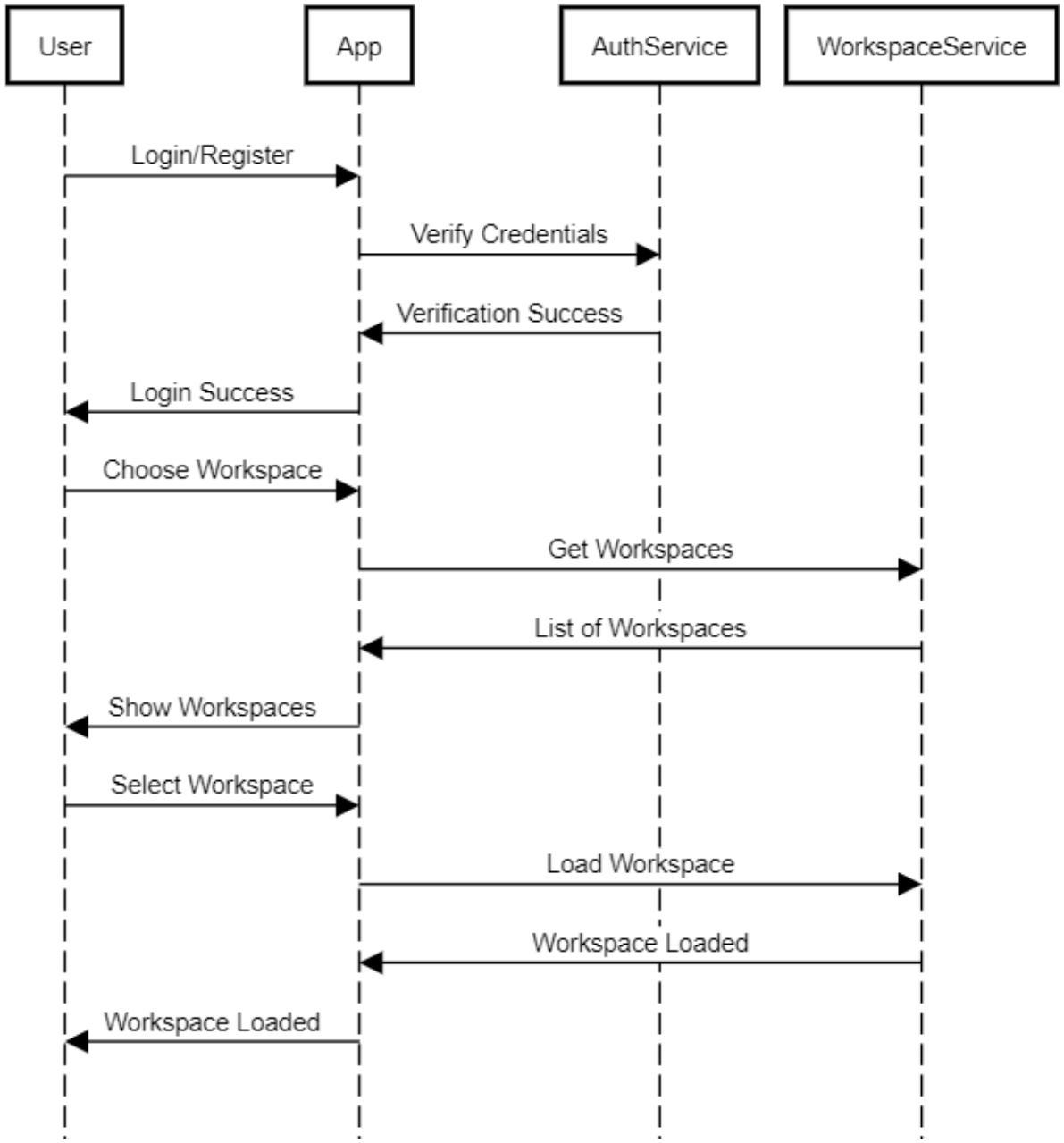


Sequence Diagram (User story 1) Explanation:

The sequence diagram illustrates the interactions between different components of the system during the user registration process. Here's an explanation of each step:

1. **User Initiates Sign-Up:**
   - The user accesses the Work Planner application and selects the "Sign Up" option.
2. **Display Sign-Up Form:**
   - The system displays a sign-up form to the user, which includes fields for a username and password.
3. **User Submits Sign-Up Form:**
   - The user fills out the sign-up form with a unique username and password and submits the form.
4. **System Receives Sign-Up Request:**
   - The sign-up request is received by the system's User Interface (UI) component.
5. **UI Validates Input:**
   - The UI validates the input fields to ensure they meet the required criteria (e.g., password strength, username uniqueness).
6. **UI Forwards Request to Authentication Service:**
   - The UI forwards the validated sign-up request to the Authentication Service.
7. **Authentication Service Checks for Username Uniqueness:**
   - The Authentication Service checks the database to ensure the username is unique.
   - If the username already exists, an error message is sent back to the UI, and the user is prompted to choose a different username.
8. **Authentication Service Creates User Account:**
   - If the username is unique, the Authentication Service creates a new user account in the database.
9. **Database Saves User Account:**
   - The user account details are saved in the database.

USER STORY 2:



Sequence Diagram (User Story 2) Explanation:

The sequence diagram illustrates the interactions between different components of the system during the user login and workspace selection process. Here's an explanation of each step:

User Initiates Login/Register:

The user accesses the Work Planner application and selects the "Login" or "Register" option.

System Receives Login/Register Request:

The login/register request is received by the system's User Interface (UI) component.

UI Forwards Request to Authentication Service:

The UI forwards the login/register request to the Authentication Service.

Authentication Service Verifies Credentials:

The Authentication Service checks the database to verify the user's credentials.

Authentication Service Sends Verification Success:

If the credentials are valid, the Authentication Service sends a

verification success message back to the UI.

If the credentials are invalid, an error message is sent back to the UI, prompting the user to try again.

System Sends Login Success:

Upon receiving the verification success message, the system sends a login success message to the user.

User Chooses Workspace:

The user selects the "Choose Workspace" option from the UI.

UI Forwards Request to Workspace Service:

The UI forwards the request to get available workspaces to the Workspace Service.

Workspace Service Retrieves List of Workspaces:

The Workspace Service retrieves the list of available workspaces from the database.

Workspace Service Sends List of Workspaces:

The Workspace Service sends the list of available workspaces

back to the UI.

System Displays Workspaces:

The system displays the list of available workspaces to the user.

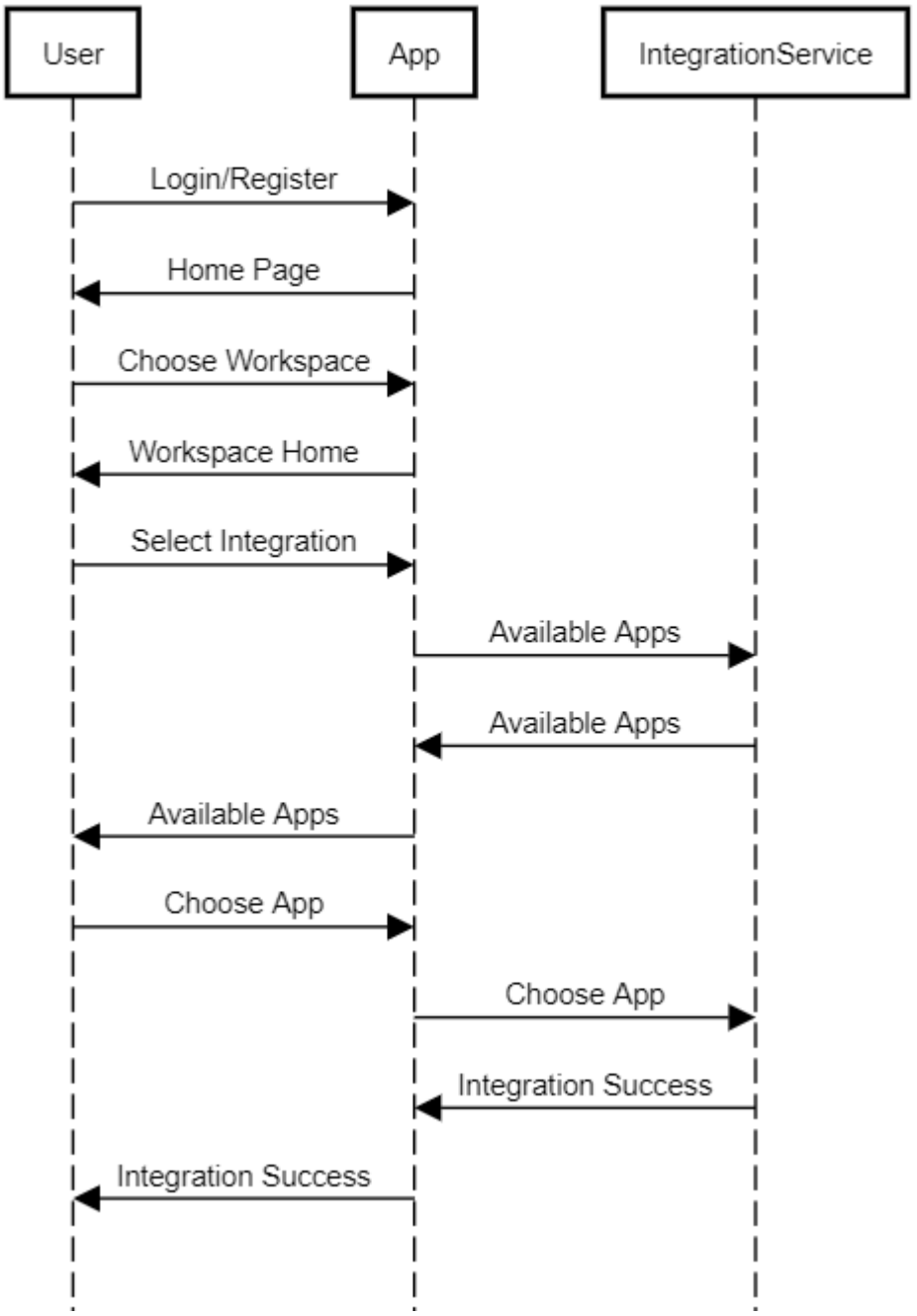
User Selects Workspace:

The user selects a specific workspace from the list.

UI Forwards Request to Load Workspace:

The UI forwards the request to load the selected workspace to the Workspace Service.

USER STORY 3:


User
App
IntegrationService
Login/Register
Home Page
Choose Workspace
Workspace Home
Select Integration
Available Apps
Available Apps
Available Apps
Choose App
Choose App
Integration Success
Integration Success

1.User->UI: Open Home Page:

- The user opens the home page of the Work Planner application.

2.UI->User: Display Dropdown:

- The user interface displays a dropdown menu with available app integration options.

3.User->UI: Select App from Dropdown:

- The user selects an app from the dropdown menu.

4.UI->IntegrationService: Process Selection:

- The user interface sends the selected app information to the integration service for processing.

note over IntegrationService,Database/ExternalService: Integration Service retrieves app data:

- A note indicating that the integration service retrieves the necessary app data from the database or an external service.

5.IntegrationService->Database/ExternalService: Retrieve App Data:

- The integration service requests the app data from the database or an external service.

6.Database/ExternalService-->IntegrationService: Return App Data:

- The database or external service returns the app data to the integration service.

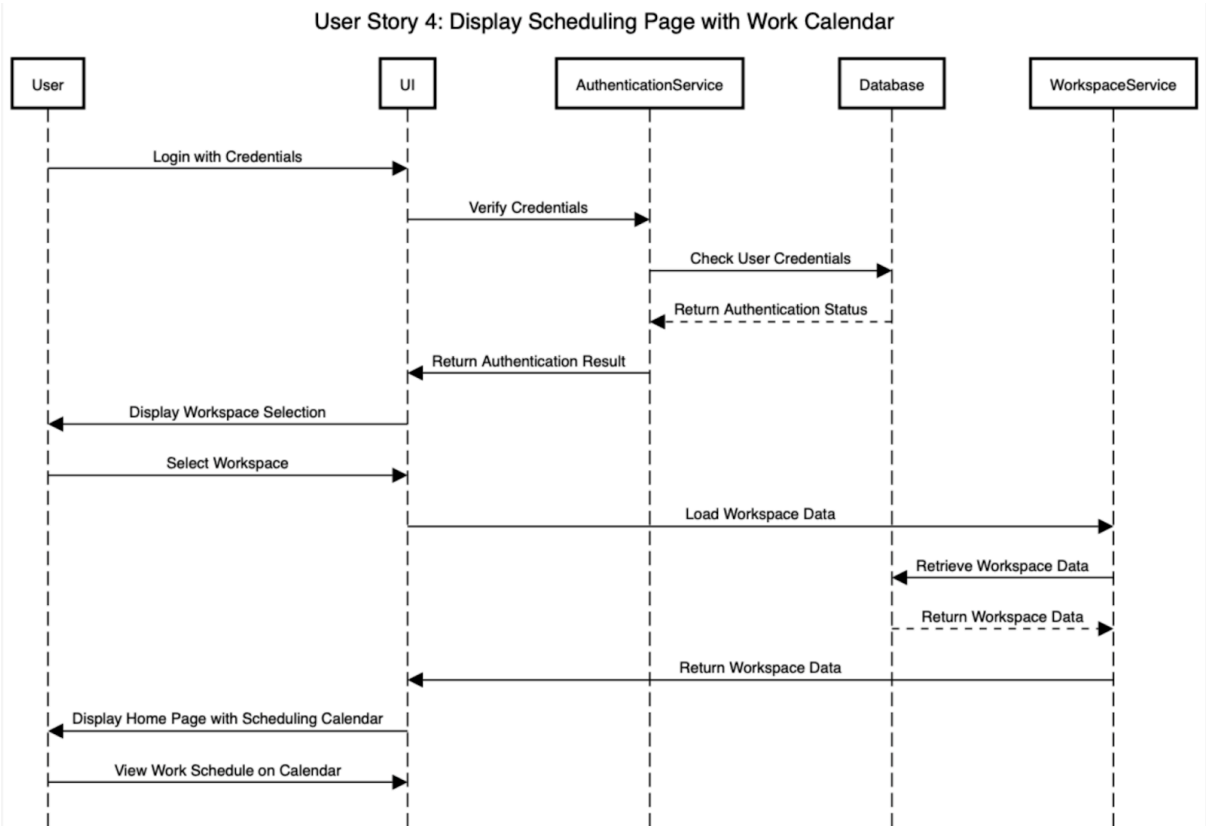
7.IntegrationService->UI: Integrate App:

- The integration service processes the app integration and updates the user interface.

8.UI->User: Confirm Integration:

- The user interface confirms the successful integration of the app and notifies the user.

USER STORY 4:



1.User->UI: Login with Credentials:

- The user logs in by entering their credentials on the login page.

2.UI->AuthenticationService: Verify Credentials:

- The user interface sends the credentials to the authentication service for verification.

3.AuthenticationService->Database: Check User Credentials:

- The authentication service checks the user credentials against the database.

4.Database-->AuthenticationService: Return Authentication Status:

- The database returns the authentication status to the authentication service.

5.AuthenticationService->UI: Return Authentication Result:

- The authentication service returns the result of the authentication to the user interface.

6.UI->User: Display Workspace Selection:

- If authentication is successful, the user interface displays the workspace selection options to the user.

7.User->UI: Select Workspace:

- The user selects a workspace from the available options.

8.UI->WorkspaceService: Load Workspace Data:

- The user interface sends the selected workspace information to the workspace service to load the relevant data.

9.WorkspaceService->Database: Retrieve Workspace Data:

- The workspace service retrieves the workspace data from the database.

10.Database-->WorkspaceService: Return Workspace Data:

- The database returns the workspace data to the workspace service.

11.WorkspaceService->UI: Return Workspace Data:

- The workspace service sends the workspace data to the user interface.

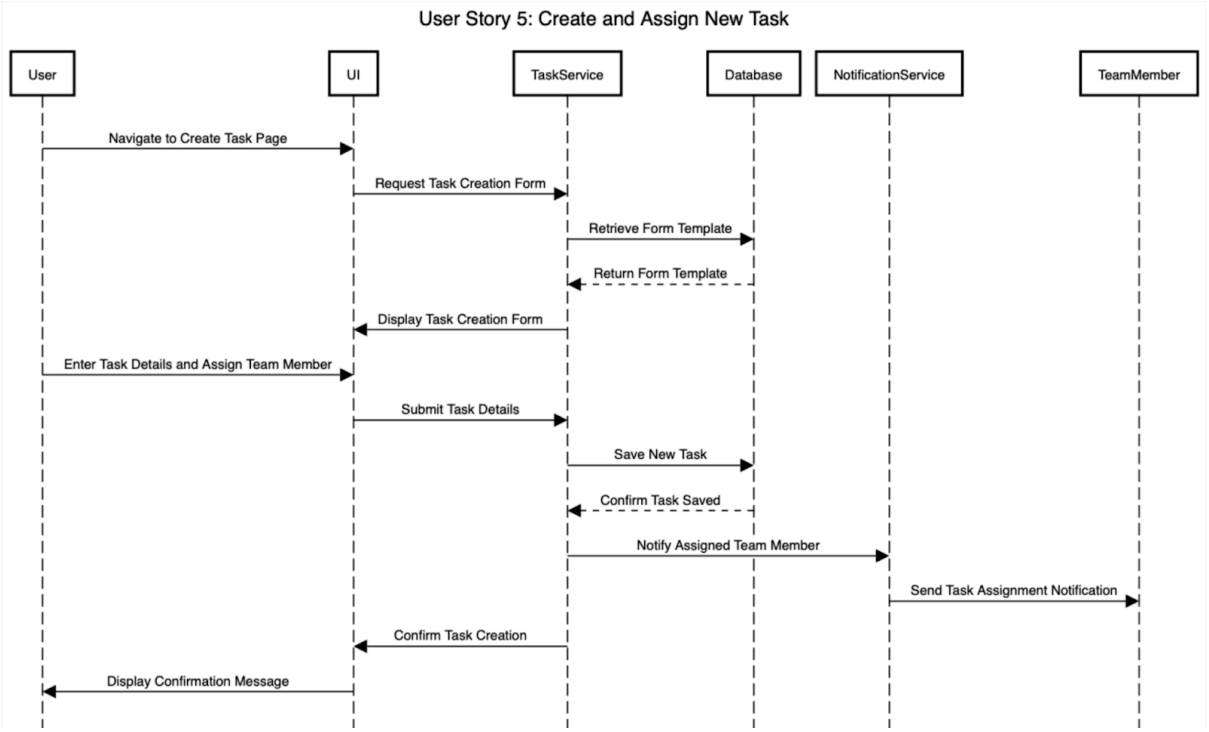
12.UI->User: Display Home Page with Scheduling Calendar:

- The user interface displays the home page, including a scheduling calendar populated with the user's work schedule.

13.User->UI: View Work Schedule on Calendar:

- The user views their work schedule displayed on the calendar on the home page.

USER STORY 5:



1. **User->UI: Navigate to Create Task Page:**
    - The user navigates to the page where they can create a new task.
2. **UI->TaskService: Request Task Creation Form:**

- The user interface requests the task creation form from the task service.

3. **TaskService->Database: Retrieve Form Template:**
   - The task service retrieves the task creation form template from the database.
4. **Database-->TaskService: Return Form Template:**
   - The database returns the form template to the task service.
5. **TaskService->UI: Display Task Creation Form:**
   - The task service sends the form template to the user interface to be displayed to the user.
6. **User->UI: Enter Task Details and Assign Team Member:**
   - The user enters the details of the new task and assigns it to a team member.
7. **UI->TaskService: Submit Task Details:**
   - The user interface submits the entered task details to the task service.
8. **TaskService->Database: Save New Task:**
   - The task service saves the new task details in the database.
9. **Database-->TaskService: Confirm Task Saved:**
   - The database confirms that the new task has been saved.
10. **TaskService->NotificationService: Notify Assigned Team Member:**
    - The task service sends a notification request to the notification service to inform the assigned team member.
11. **NotificationService->TeamMember: Send Task Assignment Notification:**
    - The notification service sends a notification to the team member about the new task assignment.
12. **TaskService->UI: Confirm Task Creation:**
    - The task service confirms the task creation to the user interface.
13. **UI->User: Display Confirmation Message:**
    - The user interface displays a confirmation message to the user indicating that the task has been successfully created and assigned.

ARCHITECTURAL PATTERNS: