RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

CS19442 SOFTWARE ENGINEERING CONCEPTS LAB

Laboratory Record
Note Book

Name :…… ……...

Year / Branch / Section : …….. …….

University Register No. : …….. ……

College Roll No. : ……

Semester : ……

Academic Year : …….

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name:_____

Academic Year: _____ Semester:____Branch:_____

Register No: | |

Certified that this is the bonafide record of work done by the above student in the CS19442-Software Engineering Concepts Laboratory during the year 2023- 2024

Signature of Faculty-in-charge

Submitted for the Practical Examination held on _____

Internal Examiner External Examiner

A REPORT ON PROJECT
WORK PLANNER SOFTWARE

INDEX

Ex.No	DATE	EXPERIMENT	PAGE.NO
1	20-02-2024	SRS	
2	01-03-2024	SCRUM METHODOLOGY	
3	12-03-2024	USER STORIES	
4	19-04-2024	USE CASE DIAGRAM	
5	29-04-2024	NFR	
6	09-04-2024	OVERALL PROJECT ARCHITECTURE	
7	19-04-2024	BUISNESS ARCHITECTURE	
8	30-04-2024	CLASS DIAGRAM	
9	10-02-2024	SEQUENCE DIAGRAM	
10	17-05-2024	ARCHITECTURAL PATTERN (MVC)	

OVERVIEW OF THIS PROJECT

In today's fast-paced work environments, individuals and teams often struggle to efficiently organize and manage their tasks, leading to missed deadlines, decreased productivity, and heightened stress levels. Traditional methods of task management, such as handwritten to-do lists or scattered digital notes, lack the flexibility and organization necessary to effectively prioritize and track work responsibilities. Additionally, many existing work planner software solutions suffer from significant drawbacks, including complex user interfaces, limited customization options, lack of integration with other productivity tools, and high costs associated with licensing or subscription fees. To address these challenges, there is a clear need for a comprehensive work planner that empowers individuals and teams to seamlessly organize, prioritize, and track their tasks and schedules in a unified digital platform. This work planner should offer intuitive task management features, customizable scheduling options, real-time collaboration capabilities, and insightful reporting functionalities, while also addressing the shortcomings of existing solutions. By providing users with a centralized hub for managing their work responsibilities, the proposed work planner aims to enhance productivity, streamline communication, and promote overall wellbeing in the workplace.

SOFTWARE REQUIREMENT SPECIFICATIONS

EX.NO: 1 DATE: 20-02-2024

PURPOSE :

Our project is a work planner that allows users to organize and manage their tasks and projects. It is designed to help individuals and teams manage their work, set priorities, and stay Organized. Our project aims to provide people a work-life balance by helping them manage their workflow and improve productivity by organizing their tasks and plans in a user friendly manner. This will help users manage their time efficiently.

PRODUCT SCOPE :

Task Manager: Users will able to create, edit, organize and track individual tasks efficiently. Integrated Calendar: Existing tasks and deadlines with users' schedules can be synced with an Integrated Calendar. Reminders and Notifications: The Website should provide timely reminders and notifications to users about upcoming deadlines, overdue tasks and important events. Accessibility and Cross platform Compatibility: The Website Should be Accessible across Various Devices and compatible with different web browsers.

INTENDED AUDIENCE :

The work planner software caters to a diverse audience, ranging from individual users to teams and organizations across various sectors. It provides a centralized platform for efficient task management and collaboration, making it ideal for professionals, freelancers, students, and businesses alike. Administrators can utilize the software to oversee team productivity and project management seamlessly. Its user-friendly interface appeals to both tech-savvy users and those with limited technical expertise, ensuring accessibility and ease

of use for all.

FUNCTIONAL REQUIREMENTS:

i) User Authentication:

The work planner system must allow users to create accounts securely. Upon account creation, users should be able to log in using their credentials. Additionally, the system should provide mechanisms for users to reset passwords and recover accounts if needed. Robust authentication protocols are essential to ensure data privacy and prevent unauthorized access.

ii) **Task Management:**

The work planner should facilitate efficient task management. Users must be able to create, edit, organize, and track individual tasks seamlessly. Each task should have relevant details, including due dates and priorities. Customizable task lists or boards will accommodate different workflows. Incorporating features like task dependencies (where one task is blocked until another is completed) and task assignment for effective team collaboration.

iii) Integrated Calendar:

The system should seamlessly sync existing tasks and deadlines with users' schedules. Users can view tasks alongside their appointments and events in a unified calendar. Provide customizable calendar views (daily, weekly, and monthly). Consider additional features such as color-coded events and the ability to set recurring tasks.

iv) Reminders and Notifications:

Timely reminders and notifications are essential for task management. The software should notify users about upcoming deadlines, overdue tasks, and important events. Customizable notification preferences (e.g., email, in-app notifications) enhance user experience.

v) Collaboration Capabilities:

Real-time collaboration features are crucial for team productivity. Users should be able to share tasks, comment on them, and collaborate seamlessly within the platform. This promotes effective communication and teamwork. The work planner should provide insightful reporting functionalities. Users should be able to generate reports on task completion, team performance, and overall productivity. Analytics can help users make informed decisions.

CONSTRAINTS:

It should be compatible with commonly used web browsers and operating systems. The work planner integrates with Third-party tools like calendars, email clients, project management software to ensure compatibility. The system architecture should accommodate scalability to expand design for future growth without compromising performance.

CONCLUSION:

The proposed Work Planner software aims to address the challenges faced by individuals and teams in managing tasks efficiently. By incorporating user authentication, robust task management features, an integrated calendar, and adherence to non-functional requirements, we can create a comprehensive solution that enhances productivity, promotes collaboration, and ensures a positive user experience. Regular maintenance and updates will keep the system reliable and secure. As we move forward, close collaboration between development teams, stakeholders, and end-users will be essential to achieve the desired outcomes

SCRUM METHODOLOGY

EX.NO: 2 DATE: 01-03-2024

Sprint 1: Basic User Authentication and Task Management

User Authentication

User registration (5 story points)

User login (5 story points)

Password reset (3 story points)

Task Management

Create new tasks (8 story points)

Edit tasks (5 story points)

Delete tasks (3 story points)

Sprint 2: Enhanced Task Management and UI Improvements

Task Management

Mark tasks as complete/incomplete (5 story points)

Set task priorities (3 story points)

Assign tasks to users (8 story points)

UI Enhancements

Responsive design (8 story points)

Sprint 3: Project Management and Notifications

Project Management

Create projects (5 story points)

Edit projects (5 story points)

Delete projects (3 story points)

Notifications

Email notifications for tasks (8 story points)

Sprint 4: Calendar Integration and Collaboration

Calendar Integration

Daily calendar view (8 story points)

Weekly calendar view (8 story points)

Collaboration

Comment on tasks (5 story points)

Tag team members (3 story points)

Sprint 5: Reporting and Search

Reporting

Generate progress reports (13 story points)

Search and Filters

Search tasks by keyword (5 story points)

Filter tasks (8 story points)

Sprint 6: Optimization and Security

Performance Optimization

Optimize load times (8 story points)

Security

Two-factor authentication (8 story points)

USER STORIES

EX.NO: 3 DATE: 12-03-2024

Theme 1: User Management

Epic 1: Secure User Authentication and Authorization

User Story 1: As a new user, I want to be able to register for an account with the Work Planner website so that I can access its features.

- **Acceptance Criteria:**
    1. The registration form must include fields for name, email, and password.
    2. The email field must validate that the input is in email format.
    3. The password must meet security requirements (e.g., minimum 8 characters, including a number and a special character).
    4. An email confirmation must be sent to the user's email address with a link to activate the account.
    5. Users must only be able to log in after successfully verifying their email address.

User Story 2: As a registered user, I want to be able to log in to my account securely with my credentials so that I can manage my tasks.

- **Acceptance Criteria:**
    1. The login form must include fields for email and password.
    2. The system must authenticate the user's credentials against the stored data.
    3. After successful authentication, users must be redirected to their dashboard.
    4. The system must prevent access to the account after five consecutive failed login attempts and require a CAPTCHA verification.
    5. Users must have an option to reset their password via a "Forgot Password" link, which sends a password reset email.

Epic 2: Profile Management

User Story 1: As a user, I want to be able to update my profile information such as name, email address, and profile picture so that my profile reflects current information.

- **Acceptance Criteria:**
    1. The profile update form must include fields for name, email, and an option to upload a profile picture.
    2. The email field must validate that the input is in email format.

3. The profile picture upload must accept common image formats (e.g., JPEG, PNG) and enforce a maximum file size.
4. Changes to the profile information must be saved and reflected immediately after submission.
5. Users must receive a confirmation message upon successful profile update.

User Story 2: As a user, I want to have the ability to change my password so that I can protect my account from unauthorized access.

- **Acceptance Criteria:**
  1. The password change form must include fields for the current password, new password, and confirmation of the new password.
  2. The new password must meet security requirements (e.g., minimum 8 characters, including a number and a special character).
  3. The system must verify that the current password is correct before allowing a change.
  4. Users must receive a confirmation email after successfully changing their password.
  5. The system must log out the user from all active sessions after the password change.

Theme 2: Task Management

Epic 1: Task Creation and Organization

User Story 1: As a busy professional, I want to be able to create and organize my tasks efficiently so that I can prioritize my workload.

- **Acceptance Criteria:**
  1. Users must be able to create new tasks with fields for task name, description, deadline, priority.
  2. The task creation form must include an option to set task reminders.
  3. Tasks must be displayed in a list with sortable columns for name, deadline, priority, and category.
  4. Tasks must save successfully and appear immediately in the user's task list upon creation.

User Story 2: As a student, I want to be able to create and organize my study tasks efficiently so that I can manage my coursework effectively.

- **Acceptance Criteria:**
  1. Users must be able to categorize tasks specifically for study (e.g., assignments, exams, projects).
  2. The task creation form must include fields for task name, description, due date, and study category.
  3. Study tasks must be displayed in a dedicated section of the task list.
  4. Users must be able to filter study tasks by course or subject.

5. Tasks must save successfully and appear immediately in the user's study task list upon creation.

Epic 2: Task Filtering and Sorting

User Story 1: As a user, I want to have the option to sort my tasks by deadline so that I can prioritize tasks based on their due dates and ensure timely completion.

- **Acceptance Criteria:**
    1. The task list must include a sorting option for deadline.
    2. Users must be able to toggle between ascending and descending order for the deadline sort.
    3. The sorted task list must update immediately upon selection.
    4. The default sort option must be customizable in the user settings.
    5. The sorting preference must be saved for future sessions.

Theme 3: Collaboration

Epic 1: Task Assignment

User Story 1: As a project manager, I want to be able to assign tasks to specific team members so that I can track individual responsibilities within the team.

- **Acceptance Criteria:**
    1. The task creation and edit forms must include an option to assign tasks to team members.
    2. Only users within the project manager's team must be selectable for task assignments.
    3. Assigned tasks must display the assigned team member's name.
    4. The system must send a notification to the assigned team member upon task assignment.
    5. Users must be able to filter tasks by assigned team members.

Epic 2: Task Commenting, Notifications, and Real-time Updates

User Story 1: As a team member, I want to receive notifications when tasks are assigned or when there are updates or changes to shared tasks so that I can stay informed and on track.

- **Acceptance Criteria:**
    1. The system must send email and in-app notifications for task assignments and updates.
    2. Notifications must include details of the task and the changes made.
    3. Notifications must be sent in real-time or within a few seconds of the change.
    4. The notification history must be accessible within the user's account.

User Story 2: As a team member, I want to be able to add comments to tasks to provide context, updates, or feedback to my team members and facilitate collaboration on shared tasks.

- **Acceptance Criteria:**
    1. Each task must have a comment section where users can post comments.
    2. Comments must support text formatting (e.g., bold, italics, bullet points).
    3. Users must be able to edit or delete their own comments.
    4. The system must send notifications to relevant team members when a new comment is added.
    5. Comments must display the date and time they were posted and the name of the commenter.

Theme 4: Task Completion and Progress Tracking

Epic 1: Task Status Tracking

User Story 1: As a user, I want to be able to mark tasks as completed manually so that I can control over updating the task status based on my progress.

- **Acceptance Criteria:**
    1. Users must have an option to mark tasks as completed from the task list or task detail view.
    2. The system must prompt for confirmation before marking a task as completed.
    3. Completed tasks must move to a separate section or have a clear visual distinction.
    4. Users must be able to undo the completion of a task if needed.
    5. The completion status must be saved and persistent across sessions.

User Story 2: As a user, I want a visual indicator on each task so that I can check the status of my tasks.

- **Acceptance Criteria:**
    1. Each task must display a visual status indicator (e.g., not started, in progress, completed).
    2. The status indicator must update automatically based on user actions (e.g., marking a task as completed).
    3. Users must be able to manually update the status of their tasks.
    4. The visual indicator must be easily distinguishable (e.g., color-coded).

Epic 2: Real-time Progress Monitoring

User Story 1: As a project manager, I want a percentage indicator for each task, so that I can see the status of work completed as a visual representation.

- **Acceptance Criteria:**
    1. Each task must include a field for the percentage of completion.

2. The percentage must be displayed as a progress bar.
3. Users must be able to update the percentage manually or automatically based on sub-tasks.
4. The progress bar must update in real-time as changes are made.
5. Tasks must display a summary of progress when viewed in list format.

Epic 3: Task Completion Metrics and Reporting

User Story 1: As a user, I want to generate reports on task completion rates, average completion times, and overdue tasks, so that I can assess my productivity and performance.

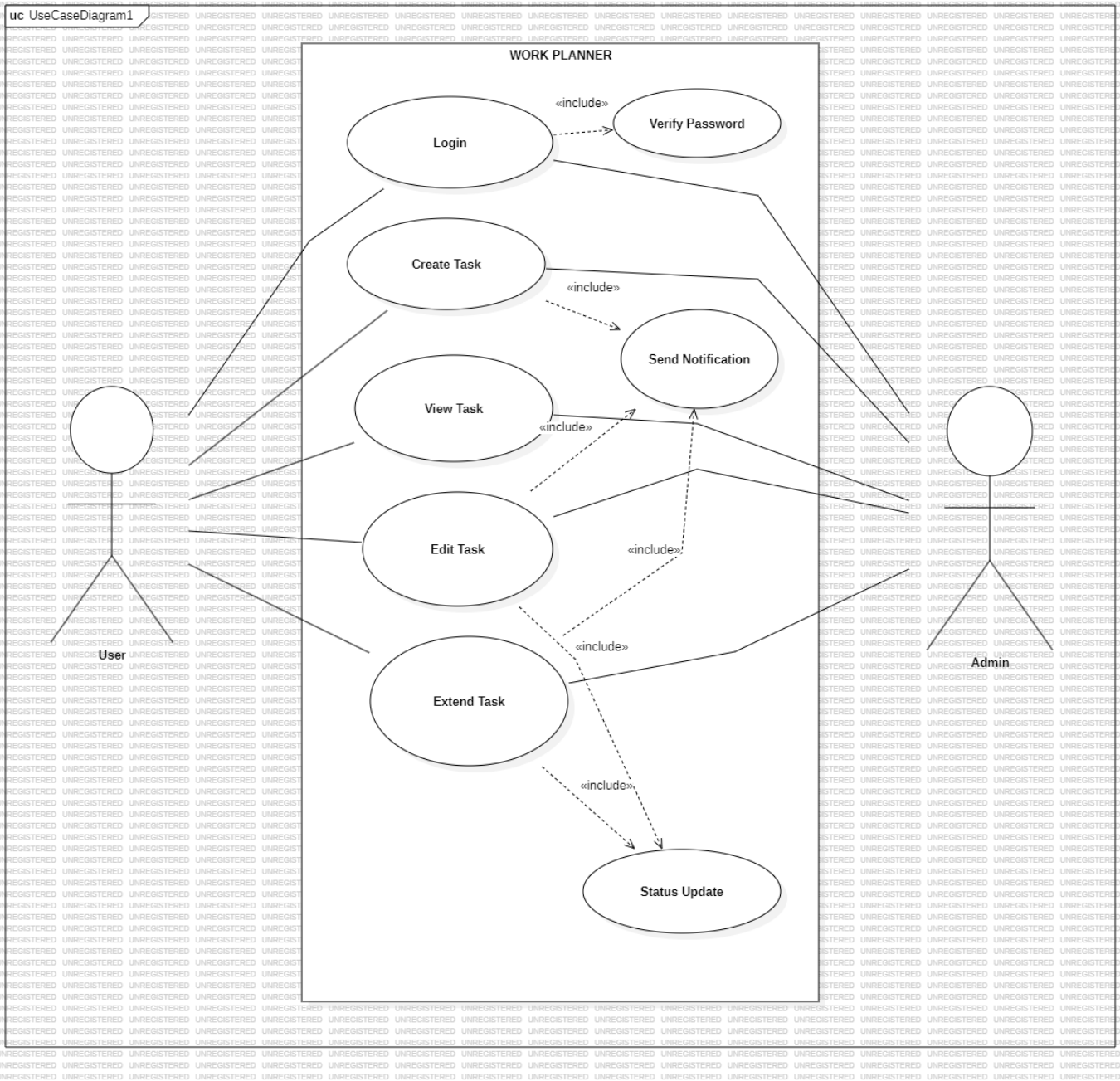
- **Acceptance Criteria:**
  1. The system must provide a report generation tool with options to select metrics such as completion rates, average completion times, and overdue tasks.
  2. Users must be able to specify a date range for the report.
  3. Reports must be downloadable in common formats (e.g., PDF, Excel).
  4. The generated report must include visual summaries (e.g., charts, graphs).
  5. Users must receive an email notification when the report is ready for download.

User Story 2: As a user, I want to view charts and graphs visualizing task completion trends and patterns, so that I can identify bottlenecks and optimize my workflow.

- **Acceptance Criteria:**
  1. The system must include a dashboard with charts and graphs visualizing task completion trends.
  2. Visualizations must include metrics such as completion rates, average completion times, and overdue tasks.
  3. Users must be able to customize the dashboard to display specific metrics of interest.
  4. The dashboard must support real-time updates as new data is entered.
  5. Users must be able to export visualizations for reporting and analysis.

USE CASE DIAGRAM

EX.NO: 4 **DATE: 19-03-2024**



This image is a use case diagram for a "Work Planner" system. Use case diagrams are used in software engineering to represent the functional requirements of a system, illustrating the interactions between users (actors) and the system itself. Here's a detailed explanation of the key components and their interactions in this diagram:

Actors:

User

Represents a regular user of the Work Planner system.

Admin

Represents an administrative user with additional privileges in the system.

Use Cases:

Login

This use case allows both Users and Admins to access the system by logging in.

Includes the "Verify Password" use case to ensure the credentials are correct.

Create Task

This use case enables Users and Admins to create new tasks within the Work Planner.

It includes the "Send Notification" use case to notify relevant parties about the new task.

View Task

This use case allows Users and Admins to view existing tasks.

Edit Task

Users and Admins can modify details of existing tasks.

It includes the "Send Notification" use case to inform relevant parties about the changes.

Extend Task

This use case allows Users and Admins to extend the deadline or duration of a task.

It includes the "Send Notification" use case to update relevant parties about the extension.

Status Update

Users and Admins can update the status of tasks, for example, marking them as complete or in progress.

Included Use Cases:

Verify Password

A subprocess of the "Login" use case, ensuring user authentication.

Send Notification

A common subprocess included in "Create Task," "Edit Task," and "Extend Task" use cases to keep relevant users informed about actions taken on tasks.

Relationships:

Include Relationships:

Shown with dashed arrows and "<<include>>" labels.

Indicates that the base use case (e.g., "Login," "Create Task") relies on the included use case (e.g., "Verify Password," "Send Notification") to complete its function.

Interactions:

The diagram illustrates that both Users and Admins can log in, create, view, edit, extend tasks, and update task statuses.

Each of these interactions often involves verifying passwords during login and sending notifications when tasks are created, edited, or extended.

This use case diagram provides a clear overview of the main functionalities of the Work Planner system and the interactions between users (both regular and administrative) and these functionalities. It helps in understanding the requirements and scope of the system, ensuring that all necessary features are included and properly documented.

NFR

EX.NO: 5 DATE: 29-03-2024

Performance:

The Work Planner should be responsive and have minimal loading times, even when handling large volumes of tasks and users concurrently. Users should not experience delays or lags when navigating through the application or performing tasks.

Scalability:

The Work Planner should be designed to handle future growth in terms of the number of tasks, users, and concurrent sessions. It should be scalable to accommodate increasing data volumes and user loads without degradation in performance or functionality.

Availability:

The Work Planner should have high availability, with minimal downtime for maintenance or updates. It should be accessible to users whenever they need to access it, ensuring uninterrupted workflow and productivity.

Security:

The Work Planner should adhere to industry-standard security practices to protect sensitive data and ensure user privacy. This includes measures such as encryption of data in transit and at rest, role-based access control, and secure authentication mechanisms.

Reliability:

The Work Planner should be reliable and robust, with mechanisms in place to prevent data loss or corruption. It should have built-in backup and recovery features to restore data in case of system failures or disasters.

Compliance:

The Work Planner should comply with relevant regulations and standards governing data privacy, such as GDPR, HIPAA, or ISO 27001. It should also adhere to any internal policies or guidelines related to data handling and security.

OVERALL PROJECT ARCHITECTURE

EX.NO: 6 DATE: 09-04-2024

Frontend

Login Page:

This is where users enter their credentials to log in to the application.

Registration Page:

This is where users can create new accounts.

Backend

Authentication Service:

This service authenticates users and verifies their credentials.

User Management Service:

This service manages user accounts, including creating, updating, and deleting them. It likely interacts with the User table in the database.

Task Service:

This service manages tasks, including creating, updating, and deleting them. It likely interacts with the Task table in the database.

Calendar Service:

This service manages calendars, including creating, updating, and deleting them. It likely interacts with the Calendar table and Project table in the database.

Project Service:

This service manages projects, including creating, updating, and deleting them. It likely interacts with the Project table in the database.

Notification Service:

This service sends notifications to users, such as reminders about upcoming tasks or deadlines. It likely interacts with the Notification table in the database.

Database:

This stores all of the application's data, such as users, tasks, calendars, and projects.

User Table:

This table stores information about users, such as their names, usernames, and passwords.

Task Table:

This table stores information about tasks, such as their titles, descriptions, and due dates.

Calendar Table:

This table stores information about calendars, such as their names and the events that are scheduled on them.

Project Table:

This table stores information about projects, such as their names, descriptions, and deadlines.

Notification Table:

This table stores information about notifications, such as the recipient, the sender, and the message.

External Integrations:

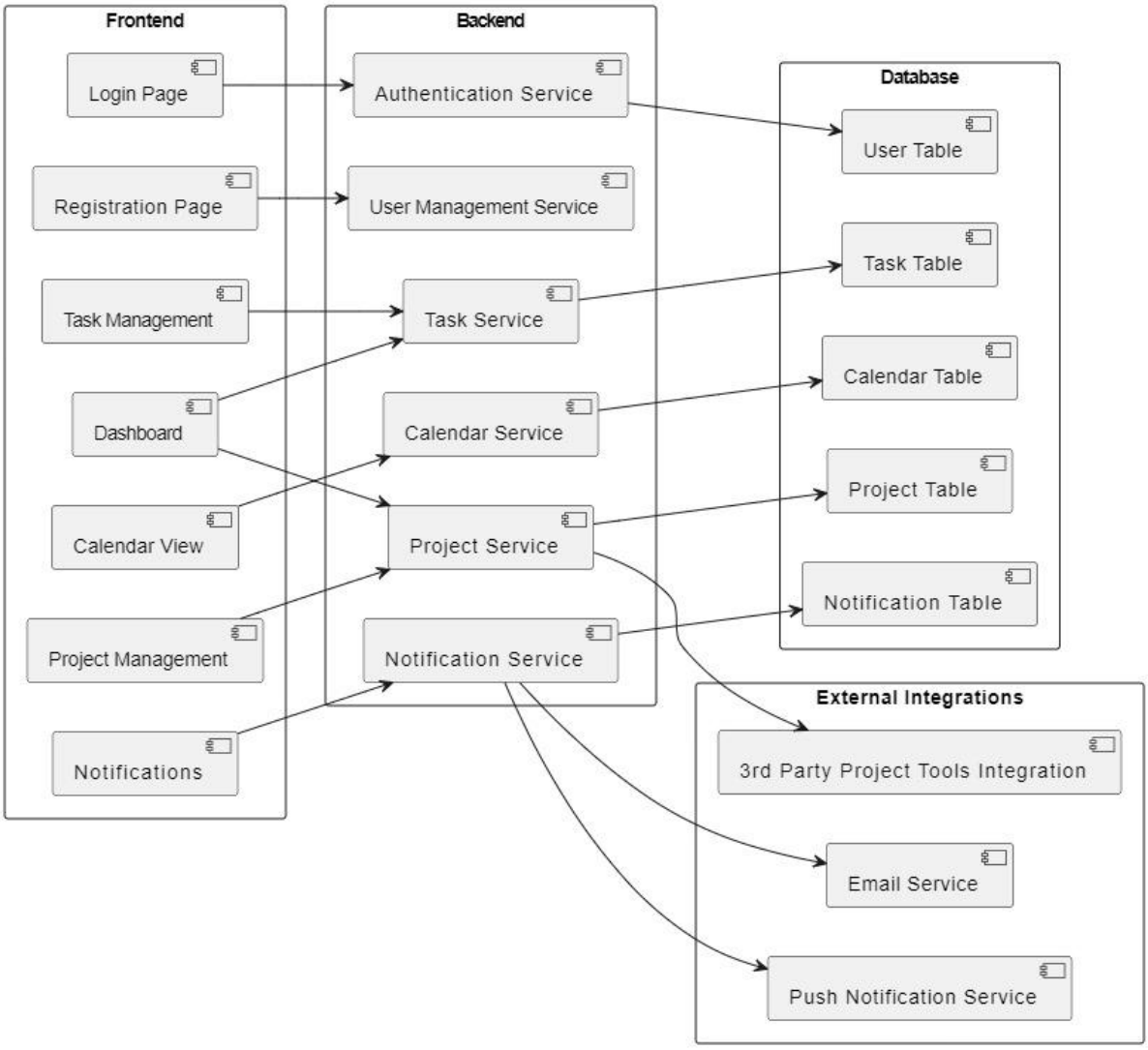
These are integrations with third-party tools and services. For example, the work planner application might integrate with a project management tool or a time tracking tool.

Email Service:

This service sends emails to users.

Push Notification Service:

This service sends push notifications to users' devices.



BUISNESS ARCHITECTURE

EX.NO: 7 DATE: 19-04-2024

Business Needs

The business needs involve:

- Improved efficiency and productivity through streamlined task management and scheduling.
- Enhanced collaboration and communication among team members.
- Better visibility and control over workload distribution.
- Reduced manual effort in work allocation and scheduling.

Current Process

The diagram doesn't reveal details on the current process, but it's manual and likely involves:

- Spreadsheets or physical planners for task allocation
- Email or physical memos for communication and updates
- Difficulty in tracking progress and dependencies

Personas and their pain points

The diagram identifies three user groups:

- **Administrator:** Sets up user accounts, manages access levels, and customizes the application. Their pain points might include difficulty in managing a large number of users and access rights.

- **Team Member:** Assigns tasks, sets deadlines, tracks progress, and collaborates with colleagues. Their pain points could be wasting time scheduling tasks and communicating updates.
- **User:** Receives assigned tasks, updates progress, and communicates with team members. Their pain points might include difficulty keeping track of assigned tasks and deadlines.

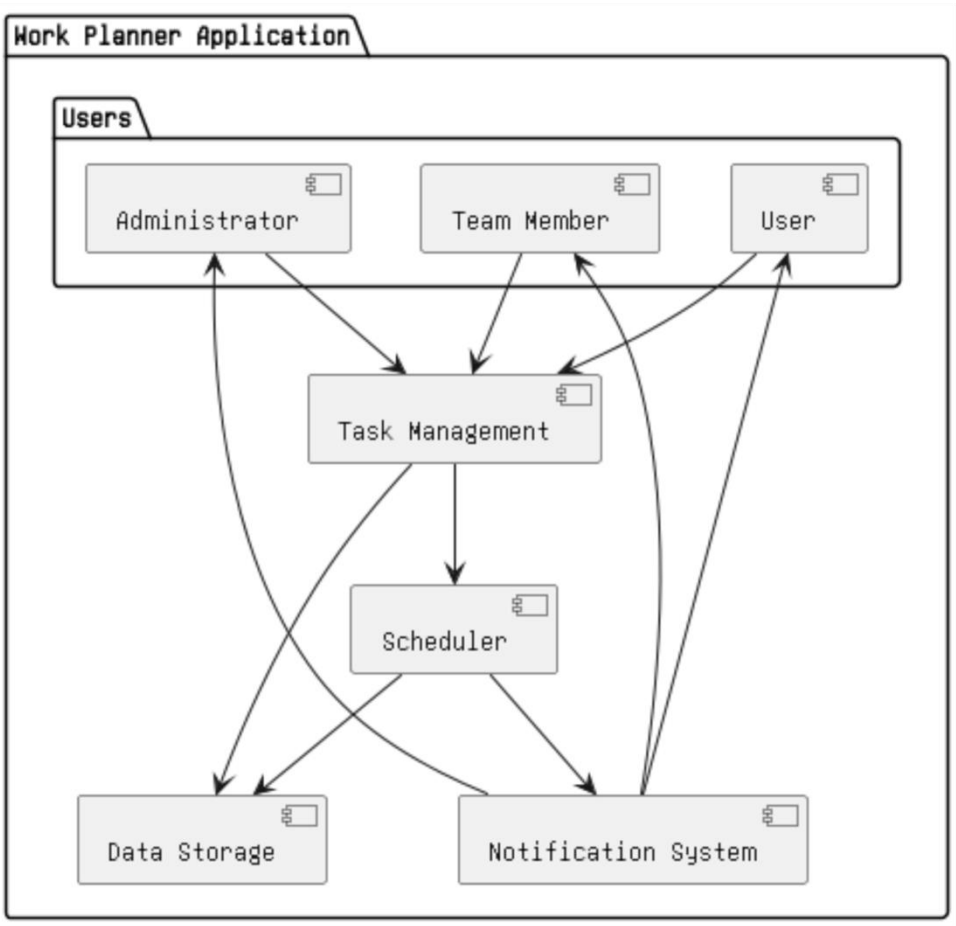
Business Problems

By automating task management and scheduling, the work planner application aims to address inefficiencies in the current process, such as:

- Inaccurate or outdated information due to manual data entry
- Difficulty in managing and tracking task dependencies
- Lack of visibility into team workload and progress
- Poor communication and collaboration among team members

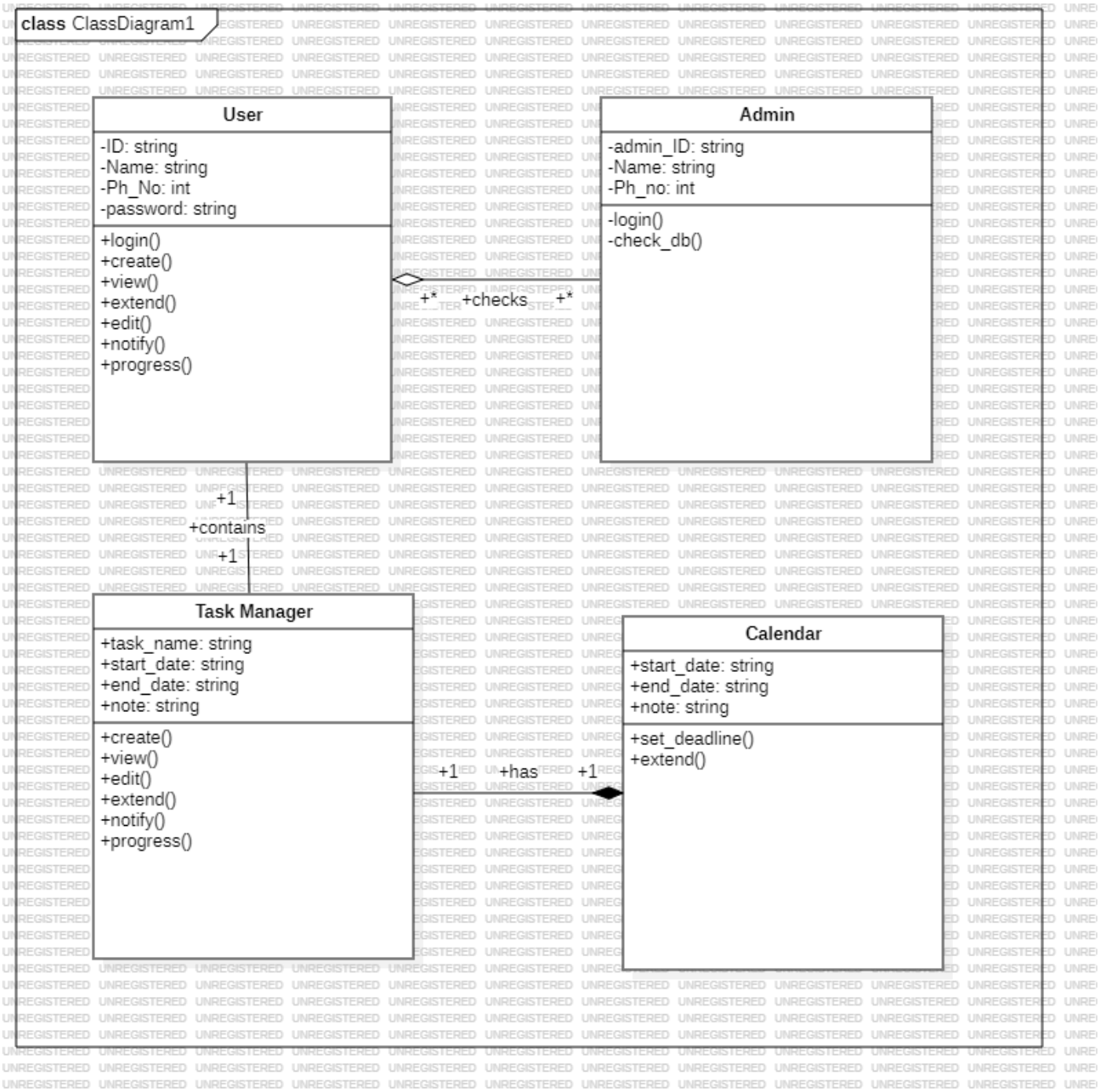
The application can potentially improve business efficiency and productivity by:

- Providing a centralized platform for task management and scheduling
- Streamlining communication and collaboration
- Improving visibility into workload and progress
- Reducing manual effort and errors


Work Planner Application
Users
Administrator
Team Member
User
Task Management
Scheduler
Data Storage
Notification System

CLASS DIAGRAM

EX.NO: 8 **DATE: 30-04-2024**



This image represents a class diagram for a "Work Planner" system, depicting the system's main classes, their attributes, methods (operations), and the relationships between them. Class diagrams are a fundamental part of object-oriented design and are used to visually represent the structure of a system by showing its classes and the relationships between them.

Classes and Their Details:

User

Attributes:

ID: string

Name: string

Ph_No: int

password: string

Methods:

login()

create()

view()

extend()

edit()

notify()

progress()

Relationship:

A User checks (checks) the Admin.

Admin

Attributes:

admin_ID: string

Name: string

Ph_no: int

Methods:

login()

check_db()

Relationship:

An Admin is checked (checks) by a User.

Task Manager

Attributes:

task_name: string

start_date: string

end_date: string

note: string

Methods:

create()

view()

edit()

extend()

notify()

progress()

Relationship:

A Task Manager contains (contains) multiple tasks and is related to Users.

Calendar

Attributes:

start_date: string

end_date: string

note: string

Methods:

set_deadline()

extend()

Relationship:

A Calendar is associated (has) with a Task Manager.

Relationships:

User and Admin:

The User class has a dependency on the Admin class, shown by the checks relationship. This indicates that a User might perform operations that involve checking or interacting with the Admin (e.g., validation or approval processes).

User and Task Manager:

The User class interacts with the Task Manager class to perform various operations related to tasks (creating, viewing, editing, extending, notifying, and tracking progress).

Task Manager and Calendar:

The Task Manager class is associated with the Calendar class, indicating that tasks managed by the Task Manager are scheduled and possibly extended within the Calendar. This relationship is represented by the has and contains notation, showing the hierarchical relationship where a Task Manager has one or more Calendar entries associated with tasks.

Summary:

The User and Admin classes handle user-related and admin-related functions, respectively, with methods for logging in and managing tasks.

The Task Manager class is central to managing tasks, containing methods for creating, viewing, editing, extending, notifying, and tracking the progress of tasks.

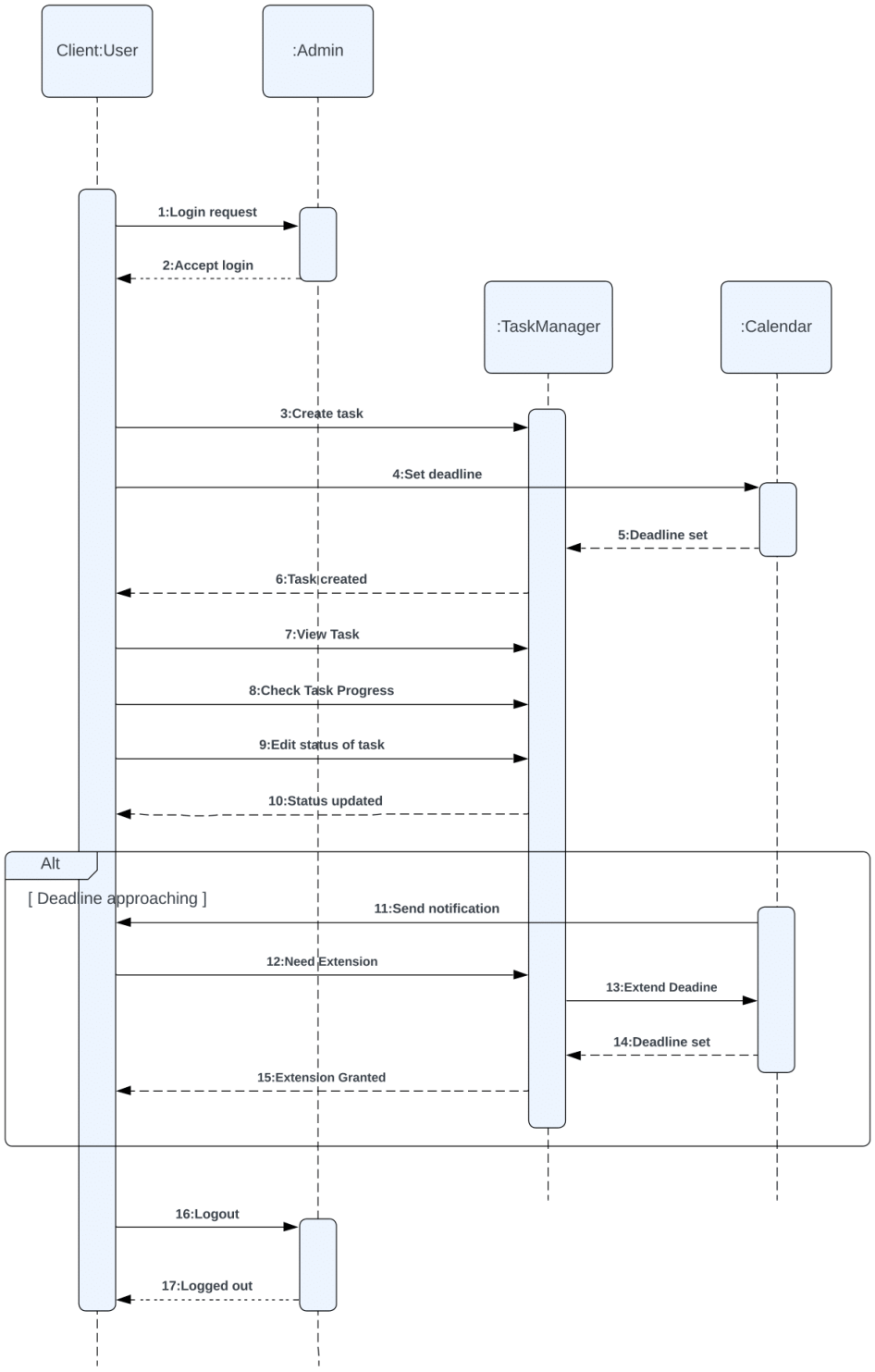
The Calendar class manages scheduling and deadlines of tasks, with methods to set deadlines and extend them.

Relationships between classes ensure that users interact with the system to manage tasks, while tasks are scheduled and tracked through the Calendar.

This class diagram provides a structural overview of the Work Planner system, helping in understanding how different components (classes) interact and their responsibilities within the system.

SEQUENCE DIAGRAM

EX.NO: 9 **DATE: 10-05-2024**



Client: User:

This is the user of the application who interacts with the system.

Admin:

This is the administrator of the application who has additional privileges such as creating tasks.

TaskManager:

This component is responsible for managing tasks. It interacts with the Client and the Calendar to create tasks, set deadlines, and update the status of tasks.

Calendar:

This component is responsible for storing deadlines and sending notifications. It interacts with the TaskManager to receive deadlines and send notifications to the Client when deadlines are approaching.

The sequence diagram illustrates a typical workflow where a user logs in, creates a task, sets a deadline, and views the task. The diagram also shows an alternate scenario where a deadline is approaching and the user requests an extension.

ARCHITECTURAL PATTERN (MVC)

EX.NO: 10 **DATE: 17-05-2024**

MVC (Model-View-Controller) is a software design pattern that promotes separation of concerns in application development. It structures the application into three distinct parts:

- **Model:**

  Represents the application's core data and business logic. It encapsulates data access and manipulation, ensuring data integrity.

- **View:**

  Deals with how the application presents information to the user. It translates the model's data into a user-friendly format on the chosen interface (mobile, web, desktop).

- **Controller:**

  Acts as an intermediary between the user and the model. It receives user input from the view, interacts with the model to perform actions based on that input, and updates the view with any changes.

-

Components in the Work Planner MVC

1. Model:

- **Task Service:**

  This service handles the logic behind tasks. It creates new tasks, modifies existing ones, retrieves tasks based on various criteria (user, status, etc.), and deletes tasks when needed.

- **Reporting Service:**

  Generates reports based on work planner data. This could involve reports on task progress, user activity, workload distribution, etc.

- **Notification Service:**

  Manages notifications related to tasks and deadlines. It sends alerts or reminders to users based on pre-defined rules or user preferences.

- **Authentication Service:**

  Validates user credentials during login and manages user access levels within the application.

- **User Service:**

  Handles user data and user accounts. It creates new accounts, manages user

profiles, and potentially controls access permissions based on user roles.

- **Calendar Service:**

  Integrates with or manages calendar applications. This service allows tasks to be scheduled on calendars, ensuring better visibility and time management.

- **Data Access Objects (DAOs):**

  These are classes responsible for interacting with the database. They encapsulate database access logic, providing a layer of abstraction for the services. There are likely specific DAOs for Tasks, Reports, Notifications, Users, and Calendars.

2. View:

- **User Interface (UI):**

  This encompasses the visual components that users interact with. The work planner application might have separate UIs designed for different devices (mobile, web, desktop). Each UI would provide functionalities for creating, viewing, modifying, and managing tasks, reports, notifications, and potentially user accounts, depending on user roles. Here are some potential additional components within the View:

  - Task lists: Displaying assigned tasks, team tasks, upcoming deadlines, etc.
  - Task details view: Showing details of a specific task, including description, attachments, progress updates, and collaborators.
  - Calendar view: Integrating with the calendar application to visualize scheduled tasks and deadlines.

- Reports view: Presenting generated reports in a user-friendly format (charts, tables).

3. Controller:

- **Task Controller:**

  This controller listens to user actions related to tasks. For example, if a user creates a new task through the UI, the Task Controller would capture that action, interact with the Task Service to create the task in the model, and then update the View (task list) to reflect the new task. Similar functionalities exist for editing, deleting, and retrieving tasks.

- **Report Controller:**

  Listens to user actions related to reports, such as generating reports by date range, user, or task status. It interacts with the Reporting Service to generate the report and updates the View to display it.

- **Notification Controller:**

  Listens to user actions related to notifications, such as setting reminders for tasks or deadlines. It interacts with the Notification Service to schedule those notifications and updates the View to reflect the changes.

-

- **User Controller:**

  Listens to user actions related to user accounts, such as login, logout, and potentially managing user profiles (depending on user roles). It interacts with the Authentication Service and User Service to process these actions and updates the View accordingly (login success/failure, displaying user profile information).

- **Calendar Controller:**

  Listens to user actions related to calendars, such as scheduling tasks or syncing with existing calendar applications. It interacts with the Calendar Service to perform these actions and updates the View to reflect the changes (e.g., task added to calendar).