**SW Engineering CSC648 Section 1**

Project title: TaskFlow

### Team #4

#### Team Leader: Bisum

#### Other Members: Josh, Zoe, Anisah, Sanna, Jun

#### MILESTONE 2

#### 10/21/2024

#### History table

| M2 Version 1 | October 21, 2024 |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

#### App: <https://csc648-01-fa24-csc648-01-fall24-team04.vercel.app/>

## Table of contents

[**1. Data Definitions 3**](#_730fnq3geghn)

[**2. Functional Requirements 6**](#_pky6qku5wgk5)

[**3. UI Mockups and UX Flows 8**](#_js7b4ojy6ub2)

[**4. High-level Architecture, Database Organization 14**](#_dqbkkgq2co0z)

[**5. High Level UML Diagrams 17**](#_xi5h78iawnd7)

[**6. Identify actual key risks for your project at this time 17**](#_brm6onx545wy)

[**7. Project Management 17**](#_j1iyskw5keyc)

# Data Definitions

Sanna

1. **User**
   1. **Definition:** An individual who interacts with the TaskFlow platform.
   2. **Types:** Users can be categorized into different roles, such as:
      1. **Basic User:** Can create, manage, and track their own tasks and schedules
      2. **Admin User:** Has additional privileges, such as managing groups or assigning tasks to others
   3. **Usage:** users log into the system to access their personalized tasks and schedules, which syncs across devices
2. **Task**
   1. **Definition:** a single action item or event created by a user that needs to be completed or tracked
   2. **Attributes:** 
      1. **Title:** a brief description of the task
      2. **Due Date:** the deadline by which the task should be completed
      3. **Priority:** a classification that indicates the urgency or importance of the task (high, medium, low)
      4. **Status:** the current state of the task (e.g. In progress, completed, pending)
      5. **Category:** A label or grouping (e.g. work, personal, school) used to organize tasks
   3. **Usage:** users create tasks with the appropriate attributes, manage them over time, and mark them as completed when done
3. **Calendar View**
   1. **Definition:** A visual representation of a user’s tasks and events, organized by day, week, or month
   2. **Types:**
      1. **Daily View:** displays tasks scheduled for a single day
      2. **Weekly View:** shows tasks for an entire week
      3. **Monthly View:** Provides a high-level view of all tasks for the month
   3. **Usage:** Users can switch between different calendar views to see their tasks over specific periods of time
4. **Category**
   1. **Definition:** A label assigned to tasks that helps users organize and sort them based on context (e.g. work, personal, school)
   2. **Attributes:** 
      1. **Name:** The name of the category
      2. **Color Code:** A color associated with the category to visually differentiate it from others
   3. **Usage:** Categories allow users to group related tasks for easier tracking and organization
5. **Priority Level**
   1. **Definition:** A color-coded design to measure of how important or urgent a task is
   2. **Levels:**
      1. **High:** tasks that need immediate attention + critical deadlines
      2. **Medium:** moderate importance + urgency
      3. **Low:** less urgent + can be completed at a later time
   3. **Usage:** Users assign priority levels to tasks to help manage their workload and focus on what needs to be done first.
6. **Reminder**
   1. **Definition:** A notification or alert set by the user to be reminded of a task/deadline before it’s due
   2. **Attributes:** 
      1. **Reminder Time:** the time at which the user receives the reminder (e.g. 1 hr before, 1 day before, etc.)
      2. **Recurring Reminder:** Option to set repeating reminders for ongoing tasks
   3. **Usage:** Users set reminders to ensure they are notified of important tasks ahead of time, helping them stay on track
7. **Point System**
   1. **Definition:** A gamification feature that rewards users for completing tasks with points, which can be tracked for motivation + productivity
   2. **Attributes:**
      1. **Points Awarded:** the number of points earned per task, which may vary based on difficulty and/or priority level
      2. **Total Points:** The cumulative points a user has earned for completed tasks
   3. **Usage:** Users are awarded points when they complete tasks, motivating them to stay on top of their work and providing a sense of accomplishment
8. **User Account**
   1. **Definition:** A personalized account for each user, containing their login credentials and associated data (tasks, preferences, schedules)
   2. **Attributes:** 
      1. **Username:** a unique identifier for the user
      2. **Password:** a secure key to access the user’s account
      3. **Profile Settings:** Customizable options like light/dark mode, notification preferences, and time zones
   3. **Usage:** User accounts store individual data and allow users to securely log in, access, and manage their tasks across multiple devices
9. **Time Zone** 
   1. **Definition:** The specific geographic time setting associated with a user’s tasks and reminders
   2. **Attributes:**
      1. **User’s Local Time Zone:** the time zone where user is located or prefers their tasks to be managed
      2. **Event Time Zone:** time zones can be assigned to specific events or tasks when they take place in different locations
   3. **Usage:** Users set a default time zone for their tasks, and the system adjusts accordingly for accurate scheduling and reminders
10. **Task Status**
    1. **Definition:** The current condition or progress of a task
    2. **States:**
       1. **Pending:** the task is created but not started
       2. **In Progress:** task is currently being worked on
       3. **Completed:** task has been finished
       4. **Overdue:** task passed its due date without being completed/marked as completed
    3. **Usage:** Task statuses help users track the lifecycle of a task and understand what still needs to be done
11. **Color Scheme**
    1. **Definition:** An option for users to set a color scheme that they like. Can be light/dark/pastel/inverted mode
    2. **Attributes**
       1. **scheme\_id (Primary Key) -** A unique identifier for each color scheme.
       2. **name -** The name of the color scheme (e.g., Light, Dark, Pastel, Inverted).

# 

# 

# 

# 2. Functional Requirements

Sanna

**User Accounts & Authentication**:

* Users can create accounts, log in, and manage their personal profiles (basic and admin roles).
* Secure login system with password protection, possibly using NextAuth.
* Sync tasks and preferences across multiple devices.

**Task Management**:

* Users can create, edit, and delete tasks.
* Tasks should have attributes like title, due date, priority, status (pending, in progress, completed, overdue), and category.
* Tasks can be categorized into color-coded groups such as work, personal, or school.

**Task Views**:

* Users can switch between daily, weekly, and monthly views to see their tasks over different periods of time.
* Option to customize task view preferences (e.g., filter by priority, status, or category).

**Task Comparison & Productivity Tracking**:

* Users can compare task completion on a weekly basis, tracking productivity over time.
* A progress bar or streak tracker displays ongoing task completion and productivity.

**Gamification (Points & Rewards System)**:

* Users earn points for completing tasks. Points vary based on task difficulty or priority.
* Accumulated points can unlock badges, themes, emojis, or customization options.
* Display a visual progress bar showing points earned and progress towards unlocking rewards.

**Celebratory Animations & Positive Reinforcement**:

* Upon task completion, celebratory animations (e.g., confetti burst, fireworks) and sound effects are triggered.
* Users receive positive reinforcement messages to encourage continued productivity.

**Daily Accomplishment Summary**:

* At the end of each day, users receive a visually engaging summary of their accomplishments (e.g., celebratory graphics summarizing completed tasks).

**Customizable Task Settings**:

* Users can customize time settings, deadlines, and reminders based on time zones and preferences (e.g., recurring reminders).
* Tasks can be set with reminders that notify users before a deadline (e.g., 1 hour, 1 day before).

**Color Scheme & Light/Dark Mode**:

* Users can switch between light and dark modes for a more personalized visual experience.
* Customizable color schemes to differentiate tasks visually by categories and priority levels.

**Role-Based Task Assignment**:

* Admin users (e.g., parents or teachers) can assign tasks to other users (e.g., kids or students) and track their progress.
* Users can collaborate and share tasks across accounts (e.g., parents assigning tasks to kids or teachers scheduling tasks for students).

**Calendar Integration**:

* Tasks and events are integrated into a calendar view, which syncs with Google Calendar, iCal, or similar platforms.
* Ability to add recurring events and tasks (e.g., weekly quizzes, regular meetings).

**Mobile & Desktop Responsiveness**:

* The platform should be fully responsive, accessible via mobile devices and desktops.

# 3. UI Mockups and UX Flows

Zoe

## 3.1 Prototype

<https://www.figma.com/design/hSCHfzSrdezrfxDIILo86u/TaskFlow?node-id=0-1&t=ZoXaz0YVS7Gt7tZh-1>

| **Functional Included** | |
| --- | --- |
| Log in |  |
| Display Calendar (Monthly, weekly, daily) |  |
| Add Tasks |  |
| Color Theme |  |
| Light/Dark mode |  |
| Point System |  |

## 3.2 Seven UX Quality Criteria

### 3.2.1 Useful

* The purpose of the tool is to help users manage their tasks and schedules effectively.
* **Features**:
  + Monthly and Weekly calendar views provide flexibility for planning at different granularities.
  + Users can add tasks with customizable attributes like priority, category, and description.
  + Theme customization offers personalization, making the app more engaging.
  + The night mode option supports users working in low-light environments.
  + Point system keeps user motivated to finish their Tasks

### 3.2.2 Usable

* **Intuitive Structure:** Each screen is focused on a particular function (e.g., Add Task, Overview), which minimizes cognitive load.
* **Form Inputs:** Fields in the Add Task section (start time, category, priority) are clearly labeled, ensuring the input process is straightforward.
* **Consistency:** The same brown color scheme is applied throughout, creating visual continuity.

3.2.3 Findable

* Each section (e.g., "Overview," "Add Task") is clearly labeled, and the menus are easy to find.
* The calendar views are well-organized by time periods (weekly and monthly).
* Users can quickly find specific tasks by searching
* Users can filter tasks by category (e.g., "Work," "Personal") within views that would enhance findability.

### 3.2.4 Desirable

* The soft brown color palette feels welcoming and user-friendly.
* Icons like the "+" button and toggle switches make the interface more interactive.
* Night mode enhances aesthetics while supporting user needs during nighttime.

### 3.2.5 Accessible

* **Color Themes**: Offering both light and night modes improves accessibility for people sensitive to light.
* **Readable Fonts:** The chosen fonts seem clear and legible. Ensure that font sizes are large enough, especially for users with vision impairments.
* **Interaction Hints**: Include hover effects or focus outlines for better accessibility.

### 3.2.6 Credible

* **Professional Design**: The clean and minimal layout suggests reliability.
* **Help Section**: We would add a help section or tooltips within the app to guide users and increase credibility.

## 3.3 TaskFlow Prototype Documentation

### **Main Features Overview**

This prototype is designed to make task management easier by offering a few key features:

* **Sign-Up Page**: Users can create accounts either with their email or through Google/Facebook.
* **Overview Page**: Shows your progress and achievements
* **Add Task Page**: A simple form where you can input new tasks with details like priority, category, and time.
* **Monthly & Weekly Views**: Two different ways to look at your schedule. Monthly gives you the big picture, while Weekly helps with more detailed planning.
* **Color Themes**: Lets you adjust the colors for different task categories and you can have different color groups.
* **Night Mode**: Switches the interface to a dark theme, making it more comfortable to use in low-light situations.

### **How to Use the Prototype**

Here’s a quick guide on how to get started with the app and make the most of its features.

1. Sign Up / Login

* Open the Sign-Up Page and enter your details.
* You can use your email, Google, or Facebook to sign in.

2. Navigating Calendar Views

* Monthly View: shows all your tasks over the month
* Weekly View: Focuses on tasks within a specific week, helping you break things down into smaller chunks.

3. Adding a Task

* Go to the Add Task Page and fill in the task details, such as Start Time, Category, and Priority.
* Click Add Task, and it’ll show up in your calendar.

4. Customizing Themes

* Go to My Color Theme to personalize the colors for different types of tasks (e.g., blue for Work or green for Meetings).
* Once you save your changes, the new colors group will be saved in your collection and you can select one of them to use.

5. Switching to Night Mode

* Use the toggle switch at the top to switch to night mode.
* This is useful when you’re working late or in dark environments.

### **Best Practices for Using the App**

* Use the Weekly View for detailed planning, and check the Monthly View to get an overview of your schedule.
* Organize tasks by color to make them easier to recognize at a glance.
* Overview Page shows all the tasks you finished and the points you got, which can help you stay motivated.

## **Accessibility Guidelines**

* Switch between light and night modes to make the interface easier on your eyes.
* Keyboard shortcuts will be included in future updates to make navigation even smoother.
* The app uses high-contrast text to help people with visual impairments.

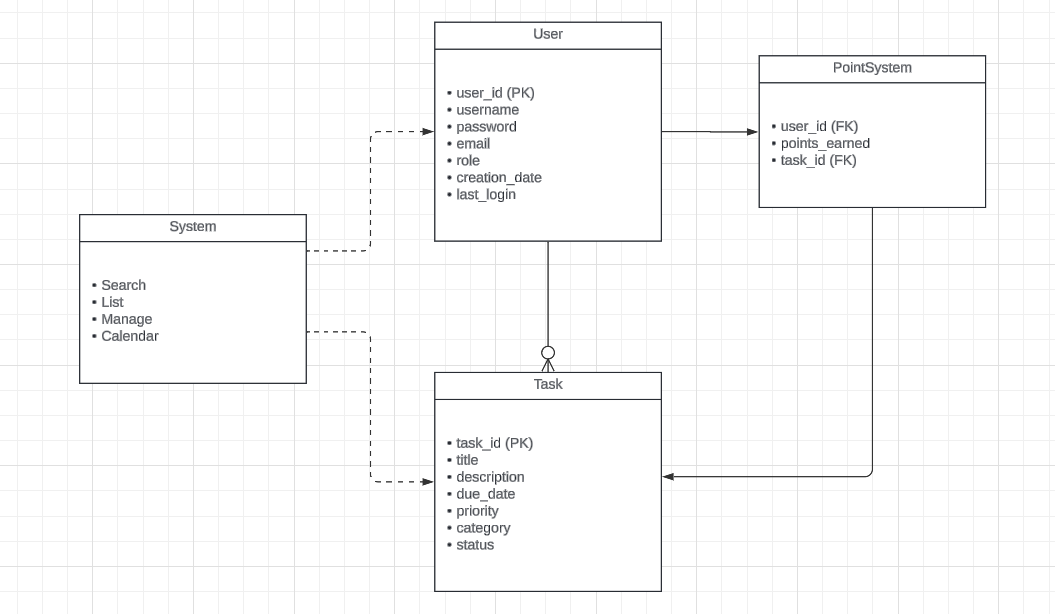
# 4. High-level Architecture, Database Organization

Anisah, Jun

* DB Organization
  + Main DB Table
    - User:
      * user\_id(PK) : Unique identifier for each user
      * username : User’s login name
      * password : User’s password
      * email : User’s email address
      * role : If user’s admin or basic user
      * creation\_date : Timestamp of when the user signed up.
      * last\_login : Timestamp of user’s last login
    - Task:
      * task\_id(PK) : Unique identifier of each task
      * title : Task name
      * description : Task description
      * due\_date : Task Due Date
      * priority : Importance of task(low, medium, high)
      * category : Category of the task(school, work, personal)
      * status : Current status of task(pending, in progress, completed)
    - Point System:
      * user\_id(FK) : Refers user
      * points\_earned : Total points earned by user
      * task\_id(FK) : Refers the task so points can be added when the task status becomes completed.
    - Notification:
      * notif\_id(PK): Unique identifier for each notification
      * message: notification message to user
      * user\_id(FK): Which user it is being sent to
      * task\_id(FK): Which task the notification is for
* Add/Delete/Search Architecture
  + Add
    - **Users** can be created by inserting a new row into the User table. This can only be done with a unique email.
    - **Tasks** can be created by inserting new rows in the Task table. Users should enter the task title, due date, priority, and category.
    - **Points** can be added into the Point System table, but it is dependent on the tasks the user creates.
    - **Notifications** can be inserted into the Notification table, but it is solely dependent on the Task due date.
  + Delete
    - **Users** may also be deleted but this is only possible via the backend. Customers may not deactivate their account.
    - **Tasks** can be deleted (can indicate whether it’s removed or completed) from the Task table. This should be displayed on the User end.
    - **Points** should not be able to be removed.
    - **Notifications** are stored even after the due date, it is up to the backend for when it should be removed to create space.
  + Search
    - **Users** may only access their own account and can view their personal information
    - **Tasks** may be searchable and displayed for the user
    - **Points** are displayed to the user but is not searchable because it's only a number
    - **Notifications** may only be displayed a certain amount of time before the task due date. Notifications are also searchable ONLY if they have been sent to the user already.
* Backend Endpoint APIs : We used mongoDB as our database solution and Mongoose, an Object Data Modeling (ODM) library, to manage the connection between my Node.js application and MongoDB. Mongoose simplified database interactions by allowing us to define schemas and models, which ensured consistency in how data is stored and retrieved. To verify and test the API endpoints created, we used Postman, a tool that made it easy to send HTTP requests and check the responses from the server. With Postman, we tested various CRUD operations (Create, Read, Update, Delete) to ensure my API was interacting with MongoDB correctly. This streamlined the development process and gave us confidence that the backend was working as expected. Below are the endpoints for each collection.
  + User
    - Register (POST /api/users/register)
      * Allow new users to create an account by providing a username, password, email, and role.
    - Login (POST /api/users/login)
      * Allow users to login by providing their email/user and password
    - Details (GET /api/users/:user\_id)
      * Retrieve user details by providing user\_id
  + Task
    - Create (POST /api/tasks)
      * Allow user to create a task with the necessary information
    - Search (GET /api/tasks/:taskid)
      * Allow user to search up specific tasks by entering task\_id
      * This is logically inefficient as users will not remember task\_id, but we will refine this functionality to allow more ways of task searching.
  + Notification
    - Create (POST /api/notifications)
      * Allow user to create notifications with the necessary information
    - Retrieve (GET /api/notification/:user\_id)
      * Get all the notifications that apply to a single user

# 5. High Level UML Diagrams

Josh



# 6. Identify actual key risks for project at this time

*Identify only actual and specific risks in current work*

* ***Skills*** risks and mitigation plan
  + Risk
    - The team members are not familiar with some of the techstack we are using for this project such as MongoDB, typescript, etc.
  + Mitigation Plan
    - We can have skill-sharing sessions with the team and study together and help each other. Furthermore, each member should allocate time for self-study to learn what it takes to make the project.
* ***Schedule*** risk
  + Risk
    - There’s a potential risk of delays in delivering task due to heavy school work, member’s health issues, availability, etc.
  + Mitigation Plan
    - Set mini-deadlines to ensure everybody can stay on track and help other members if they don’t meet mini-deadlines and figure out the issues. Weekly reviews could also be a great idea.
* ***Teamwork*** risk
  + Risk
    - Some members might miss work because they are sick or busy, then there might be a communication breakdown or coordination lacking, especially when the tasks are dependent on other teammate’s work.
  + Mitigation Plan
    - Encourage members to increase the frequency of checking group chat and summarize the meeting result and upload it on group chat and leave a mark such as @here or make another channel to write it down, so the team members who missed it can check it out later.
* ***Legal/Content*** Risk
  + Risk
    - If the team decides to use third-party content, there might be a risk of legal issues regarding copyright issues or IP right.
  + Mitigation Plan
    - Ensure that all the resources and codes we use comply with copyright regulation properly. Although using our own code and idea is the best, but if we need reference, we should ensure that we are using free or open-source resources.

# 7. Project Management

Bisum

Our M2 tasks are managed by spitting tasks according to the team members' roles. However we aren’t strictly bound to each task as we are helping each other out with each other's tasks as well. We are meeting twice a week and a third time if necessary depending on the progress made. We are also keeping in touch by messaging each other at all times in a group discussion whenever needed.