CEN 4010 Principles of Software Engineering Summer 2023

Milestone 3: More Detailed Requirements, Architecture and a Vertical Software Prototype

# The 5 Bytes: TaskMaster

# Team 16

Sekai Wynn, [swynn2022@fau.edu]

Kaela Kent, [kkent2018@fau.edu]

Israel Andrade, [iandrade2021@fau.edu]

Joel Cardenas, [cardenasj2017@fau.edu]

Devon Purification, [dpurificatio2021@fau.edu]

<https://github.com/IsReal010/The-5-Bytes.git>

July 18, 2023

Revision History: 7/18/23 11:41pm

Vertical Prototyping feature to be evaluated:

demonstrates the users ability to create multiple entries, and have a list of tasks created

### 1 - Executive Summary

TaskMaster is an innovative task manager app designed to enhance productivity and streamline task management for individuals and teams across various market sectors.

Developed with a focus on simplicity, efficiency, and user-friendliness, TaskMaster provides a comprehensive set of features to help users stay organized, prioritize tasks, and accomplish their goals effectively.

Its key advantages would be an intuitive interface offering users a clean and user friendly interface allowing them to quickly navigate and manage tasks. Smart task organization where the user can create, categorize, and prioritize tasks, collaboration and teamwork which provides progress tracking and task sharing with other users, and reminders and notifications that will help keep the user on track.

TaskMaster stands out in the task management landscape by offering a combination of user-friendly design, intelligent organization, and collaboration features. It benefits professionals in various industries, including project management, marketing, sales, education, and freelancing.

Additionally, students, entrepreneurs, and individuals seeking personal organization and productivity tools can also greatly benefit from TaskMaster's capabilities.

### 

### 2 - Competitive analysis

| Competitor Features | Tick Tick | TimeCamp Planner | MeisterTask | TaskMaster (Planned) |
| --- | --- | --- | --- | --- |
| Free Registration |  | x | x | x |
| Reminders | x | x | x | x |
| Collaboration | x | x |  | x |
| Task Organization |  | x | x | x |
| Beginner Friendly | x |  | x | x |

Summary: By leveraging these planned advantages, TaskMaster aims to provide a comprehensive and feature-rich task management solution that surpasses the existing competitors in terms of:

Reminders: TaskMaster includes reminder functionality, similar to Tick Tick, TimeCamp Planner, and MeisterTask, ensuring users stay on top of their tasks and deadlines.

Collaboration: TaskMaster supports collaboration features, just like Tick Tick and TimeCamp Planner, enabling users to work together on tasks and projects.

Task Organization: TaskMaster offers efficient task organization capabilities, similar to Tick Tick, TimeCamp Planner, and MeisterTask, allowing users to categorize and prioritize their tasks effectively.

Beginner Friendly: TaskMaster prioritizes user-friendliness, making it accessible for beginners, similar to Tick Tick and MeisterTask, ensuring a smooth onboarding experience.

### 3 - Data definition

| **Name** | **Meaning** | **Usage** | **Comment** |
| --- | --- | --- | --- |
| User | Actor | Use Case scenarios | The person who will be using the website. |
| Entry | Data | Activity type | The content a user adds to their note or list. |
| Home Page | User interface | User interface | The initial page the user will see when they go to the website. |
| Create Page | User interface | User interface | The page the user will see when they are starting a new entry. |
| About Page | User interface | User interface | The page a user can read to learn how their info is stored and used. |
| View Page | User interface | User interface | The page a user will see to scroll through previously-made entries. |
| Pop-up | User interface | User interface | A form that will hover over the current page without taking up the entirety of the screen. |
| Share Pop-up | User interface | Activity type | A form the user will see when they want to share their entry. |

### 

### 4 - Overview, scenarios and use cases

Overview: The Task Master App is a productivity and task management tool designed to help individuals and teams effectively organize and track their tasks and projects. The app provides a user-friendly interface and a range of features to enhance task management and collaboration.

Task Master Key Features:

Task Creation and Organization:

-Users can easily create and organize tasks, specifying details such as due dates, priorities, labels, and descriptions.

-Tasks can be categorized into different projects or groups, allowing for better organization and segmentation.

Task Tracking and Progress Monitoring:

-Users can track the progress of their tasks, mark them as completed, and set reminders to stay on top of deadlines.

-Progress indicators and status updates provide a clear overview of task completion and pending items.

Collaboration and Teamwork:

-The app supports collaboration among team members, enabling users to assign tasks, share project updates, and communicate within the app.

-Team members can leave comments, provide feedback, and collaborate seamlessly on shared tasks and projects.

Calendar Integration:

-The app integrates with users calendars, allowing them to view and manage tasks alongside their other appointments and commitments.

-Calendar synchronization ensures that task deadlines and reminders are reflected in the users’ overall schedule.

Notifications and Reminders:

-Users receive timely notifications and reminders for upcoming deadlines, task assignments, and updates from team members.

-Customizable notification settings ensure that users stay informed without feeling overwhelmed.

Use Case - Creating Entries

The user will head to their view page and select a create button that will lead them to the create page. Here, the entry contents can be written, posted, and added to the view page.

Use Case - Editing Entries

The user will head to their view page and select the entry they would like to edit. Once in the desired entry, the user may select the edit button to alter their work. When they are finished, they may post their entry as they did when creating it.

Use Case - Viewing Entries

The user will start at the home page and select a button that leads them to the view page. Here, the user may search and scroll through their previous entries.

Use Case - Sharing Entries

The user will come onto their view page and select the entry they would like to share. Once in the desired entry, the user may select the share button. The share pop-up will open so the user may copy a link and send it to those they wish to collaborate with.

Use Case - Categorizing Entries

The user may be on the create page to categorize their new entry by selecting a label button. If an entry the user wants to categorize has already been made, they may select the desired entry from the view page and select a label button.

### 

### 5 - High-level functional requirements

1.1 The user will be able to create and edit entries with a basic sticky note format.

1.2 The user will be able to create and edit entries with a list-like format (featuring bullets and/or numbering).

1.3 The user will be able to sort their entries.

1.4 The user will be able to scroll through their entries.

1.5 The user will be able to set due dates for their entries.

2.1 The user will be able to add attachments to their entries.

2.2 The user will be able to search for entries with a search bar.

3.1 The user will be able to share their entries.

### 6 - List of non-functional requirements

1.1 The user can save their entries and come back to them later.

1.2 The user can create customizations on their entries.

1.3 The app should respond to user interactions within two seconds.

2.1 Simple to open with your username and password and can run efficiently.

2.2 User data will be stored securely.

2.3 The app can provide autocorrect and suggested ideas for you to customize.

3.1 The app can allow users to invite friends to collaborate with the user’s entries.

3.2 The user can send notifications and reminders to oneself or others.

3.3 The app will be compatible with major web browsers.

### 7- High-level system architecture and database organization

1) High level Architecture of the code must be consistent with UML class diagram Classes: Each class in the code should have a corresponding class in the UML diagram. The class should have the same name, attributes, and methods.

Relationships: The code should take into account the connections between classes.

2) DB organization: Describe the main database schema/organization (high level), User Table: Contains details about users. Attributes might include UserID (primary key), Username, Password.

3) Media storage: We will store the media files directly on the server's file system, and only keep a reference (typically the file path) in the database.

4) Search/filter architecture and implementation: Determine the criteria by which users would like to filter tasks. This will be the due date (e.g., display all tasks with due dates within the next week), priority (e.g., display all tasks with a high priority), assigned user, state (e.g., display all tasks with unfinished status), etc.

Determine the characteristics that users would wish to utilize to order tasks. This will be the task name, due date, priority, or creation date.

5) Your own APIs:

- User Authentication API: For user registration, login, and authentication.

- Project API: CRUD operations for projects.

- Task API: CRUD operations for tasks, including assignment, completion, and updates.

-Search API: Endpoint for searching and retrieving tasks or projects based on search terms.

6)One very important non-trivial algorithm or process could be automatic prioritizing of tasks based on factors such as due date, priority, and user preferences. This algorithm could analyze task attributes, user behavior, and predefined rules to determine task priorities dynamically. The prioritization algorithm could consider factors like level of urgency, importance, and task dependencies to automatically assign priority levels to tasks, helping users focus on critical and time-sensitive activities.

### 8 - High-Level UML diagrams

| SIGN IN PAGE | VIEW PAGE | ENTRY |
| --- | --- | --- |
| Username  Password  → | Entries | Name  Description  Due date  Status  ← |
|  |  |  |

### 9 - Identify actual key risks for your project at this time

Identify only actual and specific risks in your current work such as (list those that apply: 1) Skills risks (do you have the right skills)

Most team mates have hands-on experience with front end development experience, product owner knowledge, project management knowledge and some with backend development experience which will help contribute to the final project.

2) Schedule risks (can you make it given what you committed and the resources),

Some teammates may have important tasks to complete outside of class which may cause scheduling conflicts with meetings.

3) Technical risks (any technical unknowns to solve),

Sometimes there could be some code errors that may seem impossible to debug, or software application technical difficulties which causes the application not to function properly.

4) Teamwork risks (any issues related to teamwork)

Some time conflicts upon teammates may occur where teammates may have Work or other class tasks such as exams

5) Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright).

YES

Tell us how you plan to resolve risks? The key is to resolve risks as soon as possible. Categorizing risk as above helps a lot in managing them. Be brief: identify the risk and explain (2-3 lines), list how will you address these issues (2-3 lines)

Skills Risks:

To reduce this risk, team members can collaborate, share knowledge, and support each other in areas where they may have less experience.

Schedule Risks:

Effective communication and coordination among team members will help ensure timely completion of deliverables.

Technical Risks:

Identify potential technical risks and find time for research and experimentation. Utilizing online resources can help address technical challenges.

Teamwork Risks:

Regular team meetings and open discussions can help address and resolve any teamwork issues.

Legal/Content Risks:

Ensuring proper thorough research and adherence to licensing agreements.

### 