Department of Natural Applied Science
Texas A&M University Victoria

CollabTrack A Team & Personal Project Organizer

Team: Uche, Michael, Jonathan
Instructor: Professor Wenjuan Huang
Course: COSC 4320 Fall 2025

Table of Contents

1. Cover Page

2. Table of Contents

3. Introduction

4. System Requirements and Specifications

    A. Glossary

    B. User Requirements + System Specifications

    C. Functional Requirements

    D. Non-Functional Requirements

    E. Top 5 Main Requirements

    F. Scenarios (3)

    G. Use-Case Diagram

Introduction

CollabTrack is a streamlined project and task management platform designed to help individuals and small teams organize goals, manage time effectively, and collaborate efficiently. The system focuses on simplicity, clarity, and usability, providing only the essential features needed to plan and complete projects without the unnecessary complexity of enterprise-level tools. Users can create accounts, start projects, add tasks, assign responsibilities, set priorities, and communicate through an integrated activity log. This structure keeps all information centralized and ensures that collaboration happens directly within the context of the work being done.

Originally, CollabTrack was envisioned as a basic personal organizer that supported individual task tracking. Over time, the project has evolved significantly based on user feedback, testing, and design discussions. The most notable change since the previous phase is the expansion from a single-user tool into a **collaborative platform** that allows multiple users to share, track, and manage projects together. The system now includes secure user authentication using JWT and bcrypt for encrypted password storage, ensuring privacy and access control. Each project space features a live activity log, where updates, comments, and task changes are recorded automatically. This design keeps communication attached to each task and reduces the confusion caused by fragmented external chats or emails.

Another major refinement involves improvements to the user interface and dashboard layout. The updated dashboard now prioritizes “what matters today,” overdue tasks, and upcoming deadlines, helping users stay organized at a glance. These enhancements were guided by key usability goals such as enabling task creation in under ten seconds and simplifying navigation for new users. The interface remains lightweight and intuitive, suitable for both academic and real-world environments.

Technologically, CollabTrack maintains a practical and efficient architecture. The **Node.js and Express** backend powers routing and API logic, while **EJS templates** handle dynamic server-rendered views for the frontend. **SQLite** serves as the database for the prototype, with a planned transition to **PostgreSQL** for scalability in future versions. This layered structure balances simplicity and performance, ensuring a smooth development process while maintaining flexibility for future upgrades. The addition of Jest and Supertest for route testing, along with Prettier

and ESLint for code consistency, has also improved the project's reliability and maintainability.

From a professional growth perspective, this project provides valuable experience in full-stack development, teamwork, and version control practices. Each team member—Uche, Michael, and Jonathan—has clear responsibilities, mirroring a real-world software development environment. Together, these refinements and improvements demonstrate how CollabTrack has matured into a secure, collaborative, and user-friendly application that bridges the gap between individual task management and small-team project coordination.

Glossary

- **Project:** The main workspace or goal created by a user.
- **Task:** A smaller step or action item within a project.
- **Reminder:** A scheduled alert to notify users before due dates.
- **Private Project:** Only visible to the project owner.
- **Shared Project:** Visible to invited users for collaboration.
- **Owner:** The user who created the project and can manage members.

- **Member:** A user invited to collaborate on a project.

- **Status:** The progress stage of a task (To Do, In Progress, or Completed).

- **Activity Log:** A list of updates that show changes made by team members.

- **EJS:** The templating engine used to display dynamic HTML content.

- **API:** The backend routes that handle data and user actions.

User Requirements + System Specifications

UR1: Users shall be able to register and log in securely.
Spec: Passwords will be hashed with bcrypt, and user sessions handled with JWT tokens.

UR2: Users shall create, rename, and delete projects.
Spec: Each project will have a unique ID tied to the user in the database.

UR3: Users shall invite others to collaborate on projects.
Spec: The system includes a project members table to manage invites and roles.

UR4: The system shall allow the user to add, edit, or delete tasks.
Spec: Each task will have a title, due date, status, and priority.

UR5: The system shall allow users to set reminders for tasks.
Spec: A scheduler checks deadlines and displays or sends notifications.

UR6: Users shall leave comments or updates on a project.
Spec: Store user comments and updates in an activity log table.

UR7: Users shall choose between private or shared projects.
Spec: The database will include a privacy flag to limit visibility.

UR8: Users shall view project progress and completed task percentages.
Spec: Progress is calculated based on completed tasks vs total tasks.

C. Functional Requirements and Priority Explanation

Functional Requirement	Priority	Description

FR1. User Authentication (Login/Logout/Register)	High	Essential for system access and data security; implemented first to enable user-specific actions.
FR2. Project Creation and Collaboration	High	Core functionality; users must create and share projects to organize tasks collectively.
FR3. Task Management (CRUD operations)	High	Fundamental to the system's purpose; ensures users can manage all task details.
FR4. Activity Log and Comments	Medium-High	Enhances communication and transparency within teams; developed after core CRUD operations.
FR5. Dashboard Visualization (Deadlines, Priorities, Today's Tasks)	Medium	Improves usability and user engagement through visual task summaries.
FR6. Reminders and Notifications	Medium	Adds task accountability but depends on the completion of core task and project modules.
FR7. UI Responsiveness and Accessibility	Medium	Ensures the interface adapts to devices and meets accessibility standards.

FR8. Analytics and Reports (Optional)	Low	Useful for future versions but not essential for MVP demonstration.

Non-Functional Requirements

- **High Priority (Core System Qualities)**

- **NFR1: Security & Privacy** – Protect user data with password hashing (bcrypt) and JWT authentication; restrict access to private projects.
- **NFR2: Reliability** – Ensure reminders trigger accurately and data saves consistently.
- **NFR3: Performance** – Maintain quick response time (under 3 seconds per page).

-

- **Medium Priority (Enhancement Qualities)**

- **NFR4: Usability** – Keep the interface clean, simple, and easy to understand for all users.
- **NFR5: Maintainability** – Organize backend code (routes/controllers/views) for easier updates and debugging.
- **NFR6: Portability** – App should run locally or deploy easily to services like Render or Heroku using MySQL.
- **NFR7: Scalability** – Designed so the database and system can expand to handle more users or switch to a stronger DB (PostgreSQL).

- **Low Priority (Future Improvements)**

- **NFR8: Accessibility** – Optimize for mobile and assistive technologies (screen readers, keyboard navigation).
- **NFR9: Data Backup & Recovery** – Automatic daily backups or export feature for data recovery.
- **NFR10: Analytics Integration** – Support future tracking for performance and usage stats.

- **Priority Method:**

- Based on the **MoSCoW Method** (Must, Should, Could, Won't).
  Focus first on "must-have" qualities like security, reliability, and speed before enhancements and future expansions.

-

Top 5 Requirements

1.

Function	Task Creation
Inputs	Details on task name, description, due date, urgency, importance
Outputs	A task added to the user's calendar
Requires	A user account with CollabTrack
Description	This creates a new task that the user wants to keep track of. The new task will be added to the user's general task calendar
Action	Users will be prompted for information regarding the new task. Upon inputting task information like name, description and deadline, the system will create a new task and add it to the user's general task calendar
Precondition	There must not be an exact same task with the same name, due date and description
Postcondition	The new task has been added to and is viewable on the user's general calendar
Side Effects	None

2.

Function	Group Creation
Inputs	Two CollabTrack accounts (including the user creating the group)
Outputs	A new CollabTrack group with two members
Requires	Two valid CollabTrack accounts
Description	Creates a new CollabTrack group board with a multiple or one single account
Action	Upon clicking the 'New Group' button, a user will be prompted with information about the group like name and description. The user will then need to select one other user to add to the group at creation. When the user has input all selections, the new group is created and added to the list of groups the user is in.
Precondition	The only precondition is that the two accounts that will be in the group at its creation must be valid (Valid = Account exists)
Postcondition	The new group has been created, and group members can access the group board
Side Effects	None

Top 5 Requirements

3.

Function	Task deadline notifications
Inputs	An existing tasks deadline time stamp, the current time stamp
Outputs	A notification will be sent to the option the user selected to receive notifications through
Requires	A CollabTrack account, an existing task, the current time stamp, an option selected for notification 'pathway' (The option for how or what users will use to receive notifications)
Description	When a task deadline has passed/arrived, the system will send a notification to the user to inform them of the task
Action	The system keeps track of all users created and group tasks. The system compares the current time stamp to the deadline (will be stored as a time stamp) for a task. If the deadline has passed, the system will send a notification via the option, the user selected as their notification pathway. The notification will be sent through either email, text, or app notification.
Precondition	An existing task with a time stamp that has not been passed already, an option selected for how the user will receive notifications
Postcondition	A notification sent through the users desired pathway ('Pathway' as in text, email or app notification)
Side Effects	None

4.

Function	Group Task Creation
Inputs	Details on task name, description, due date, urgency, importance
Outputs	A new task that is viewable on those specific group calendars
Requires	A CollabTrack account, an existing group that you are in
Description	Creates a new task on the group calendar that all group members can view
Action	Users will be prompted for information regarding the new group task. Upon inputting task information like name, description, and

	deadline, the system will create a new task and add it to the group's task calendar for all group members to view.
Precondition	Only precondition is that a group exists, and the user is a part of it
Postcondition	A new group task that all group members can view
Side Effects	None

5.

Function	Group Invitation
Inputs	An existing group, a CollabTrack account that is not in the provided group
Outputs	A CollabTrack account that has joined the provided group
Requires	An existing group, a CollabTrack account that is not in the provided group
Description	Sends an invitation to a user to join a group user can accept or deny
Action	A user selects a specific or multiple CollabTrack account to invite to an existing group they are in. Once the invitations have been sent, the receiving users can choose to deny or accept the invitation. If they accept, they will be added into the CollabTrack group
Precondition	An existing group must exist in order to invite another account to it
Postcondition	The account that was invited is either in the group if they accepted it or not if they denied the invitation
Side Effects	None

Scenarios

1.

Initial Assumption:

John, Jane, and Brandon are all students in a group for their biology final project. The group of students are planning on doing a group project on ecosystems. They all have accounts with CollabTrack and are all in a group within CollabTrack. The

group hasan important date coming up when they will present a rough draft of their project.

Normal:

One member of the group creates a task in the group calendar within the CollabTrack app. They note down the task's name, urgency, importance, and add extra info to the description of the task.

What can go wrong:

A task may have already been created with that date and time. Users should be prompted to continue with the creation of the task. If they respond yes, the task will be created and the app will remember to notify the user when the new and already created tasks are coming up. If the user decides to not continue with the task creation, the task will be deleted and not be added to the group calendar.

Other activities: N/A

System state on completion:

The task has been created on the group calendar. Group members will be notified of when the task is coming up.

2.

Initial Assumption:

John, Jane, and Brandon are all students in a group for their biology final project. The group of students are planning on doing a group project on ecosystems. They all have accounts with CollabTrack and are all in a group within CollabTrack. The group has an important date coming up where they will present a rough draft of their project.

Normal:

One of the group members, John, needs to access the CollabTrack app to inform the other group members about not being able to make the presentation date and to reschedule. John has not accessed the CollabTrackapp on his laptop, so he'll need to log in again.

What can go wrong:

John might have forgotten the password to his account. He will need to go through the CollabTrack password reset where users authenticate with an email that they provided at account creation. The system will send code to the user, which they can then use to reset the password to their account.

Other activities: N/A

System state of completion:

Once John has received the code sent by the CollabTrack system, he can then use it to reset and create a new password for his account and get logged in to send the message to his group members.

3.

Initial Assumption:

John, Jane, and Brandon are all students in a group fortheir biology final project. The group of students are planning on doing a groupproject on ecosystems. They all have accounts with CollabTrack and are all in agroup within CollabTrack. They recently found a fourth person they want to inviteto the CollabTrack group they made.

Normal:

The new group member can create a new CollabTrack account, then someone from inside the group can invite the new member by searching their CollabTrack username up and inviting them. Once the new member gets the invite notification sent to them, they can accept or deny the invite.

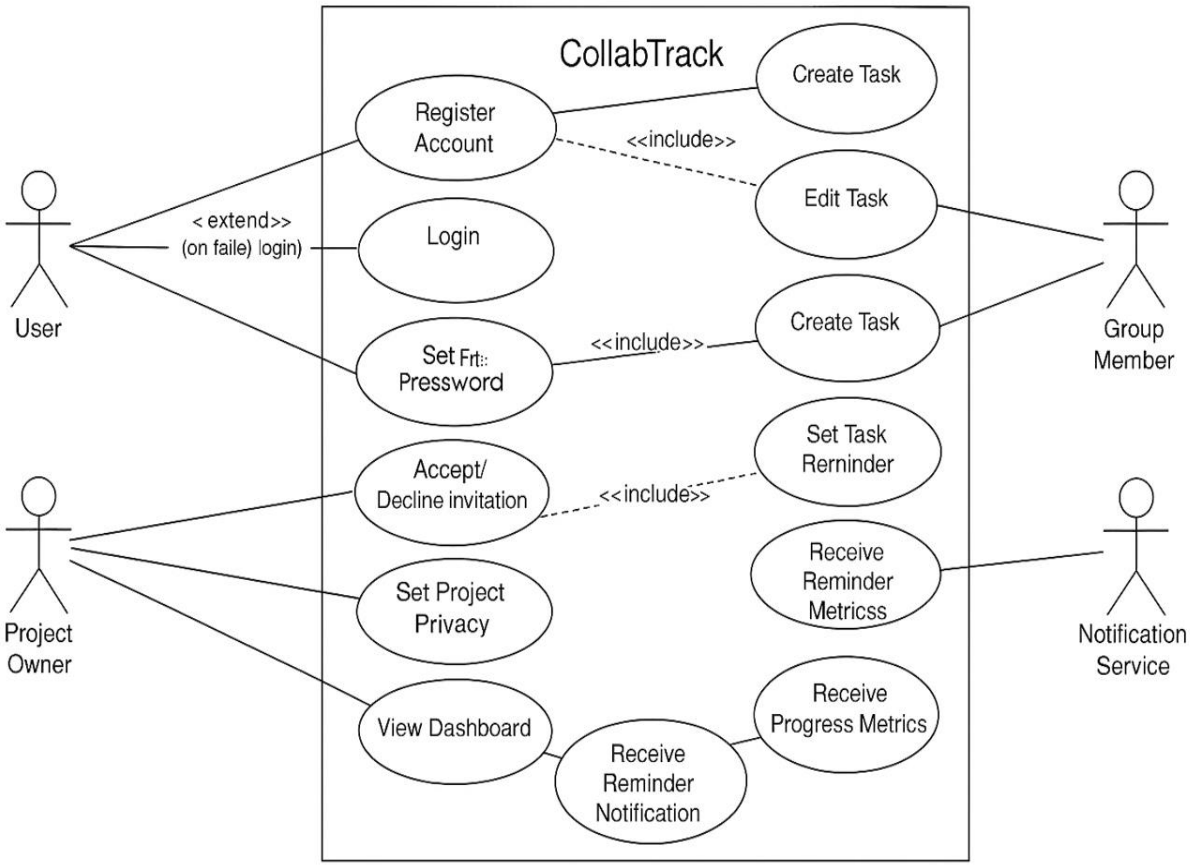
What can go wrong:

When attempting to invite the new member, the new member's account may not be able to be found by searching with their username. Instead, users can use the email or phone number linked to an account to find other accounts. The user could have already been in the group or have already been invited. In that case the invite will not be sent, and the user will be informed in the app that the intended user to add has already been invited to the group or already is in the group.

Other activities: N/A

System state of completion: The new member should be added to the group

CollabTrack Use Case Diagram


CollabTrack
Register Account
Login
Set Frt:: Pressword
Accept/ Decline invitation
Set Project Privacy
View Dashboard
Receive Reminder Notification
Create Task
Edit Task
Create Task
Set Task Rerninder
Receive Reminder Metricss
Receive Progress Metrics
<<include>>
<<include>>
<<include>>
< extend>>
(on faile) login)
User
Project Owner
Group Member
Notification Service