CheckMate

A To-Do List Application

Software Requirements Specification (SRS)

This document is an annotated outline intended for specifying software requirements and is adapted from IEEE 29148-2018.

Version 2.1

Prepared By:
Dayna Pessoa
Angela Bernabeo
Sam Thompson
Lauren Printy
Aydan Price
Kristen Rivas
Prasanna Lakshmi Guduru

Prepared For CSC 480 / HCI 521
Fall 2024

Table of Contents

Change Log

Name	Date	Reason For Changes	Version
All	9/20	Added initial functional requirements as specified in stakeholder interviews, and initial introductory documentation.	1.0
All	9/24	Further defined user characteristics to include personas and user stories.	1.1
KR	9/29	Updated / redesigned personas	1.2
All	9/30	Filled more info into section 1.3	1.3
DP	10/2	Added Introduction paragraph	1.4
ST	10/3	Added design constraints (3.6) and interface requirements (3.4)	1.5
ST	10/17	Added UML State Machine Diagrams (3.8.1) and logical database requirements (3.5)	1.6
All	10/31	Added Activity Diagrams to 3.8.2 UML Activity Diagram	1.7
DP	11/7	Reviewed and edited Introduction 1, 1.1, 1.2	1.8
DP	11/11	Reviewed & edited remainder of Introduction section 1.3 - 1.4	1.9
All	11/12	Reviewed and Edited Section 4.3 Usability Requirements and 4.4 Interface Requirements based on QA feedback	2.0
ST	11/14	Revised Activity & Sequence Diagrams in Section 3.8	2.1
All	12/2	Continuing final edits to the document - focus on Section 4.	2.2

1. Introduction

This document provides a description of the requirements for the CheckMate to-do list application created in HCI521 / CSC480 courses during Fall Semester 2024. CheckMate will provide a direct space for users to create, edit, and share to-do lists. The application will be available via the most popular browsers, and will be designed to be functionable through a desktop or laptop of different screen sizes. A high-fidelity prototype of Checkmate via figma will be tested on a number of users to obtain necessary information for reiteration prior to showcasing the final application.

1.1 Purpose

The CheckMate application is meant to provide individuals with access to a web based to-do list application that can assist with managing personal, and professional tasks. The primary audience are individuals who are wanting a to-do list application that can assist them with organizing, and tracking their daily tasks. Additionally, the application will assist in addressing issues related to memorizing, prioritizing, and monitoring tasks.

1.2 Scope

This document will serve as a detailed overview of the requirements for the CheckMate to-do list application being developed within the HCI521 / CSC480 course. The goal of this project is to deliver a minimum viable to-do list web based application by the final class, and present this to the client. The application “CheckMate” is being created for consumers to utilize for everyday tasks. At minimum, CheckMate should be accessible through Firefox and Google Chrome browsers and functionable on a desktop or laptop of different screen sizes.

Students are expected to use IBM Open Liberty product, JSON Web Token, and Microservice Architecture to develop this application. Privacy considerations should be provided to the users entering sensitive information via the to-do list application, therefore security measures will be taken to ensure data safety and a secured log in.

Based on the client’s stated requests, the following will be included in the scope of the application:

- Create a text-based web-application that allows the user to create, edit, and delete tasks.
- Create a log-in system that keeps the personal information of users secure.
- Sort tasks based on either priority, date, or both.
- Sync the application with a database for persistent storage.

- Move completed tasks to a separate part of the application that can still be viewed after the due date of the task has expired.
- Allow multiple users to edit both their own tasks as well as the shared tasks of others.
- Send notifications of both time-completed tasks and near time-completed tasks to the user when they log into the site.

1.3 Product Overview

1.3.1 Product Perspective

CheckMate is a web based to-do list application designed for a diverse group of users, who are seeking out a simple to-do list application that allows them to record, organize, and keep track of their personal and/or professional tasks. The application will allow users to create multiple lists also known as “projects”, and individual tasks. Each task will allow for customization of the task title, description, due date, and priority level making the experience individualized and unique for each user.

1.3.1.1 System Interfaces

Checkmate is a standalone web application embedded in a browser.

1.3.1.2 User Interfaces

Pages:

- Registration Page - Users will be able to create their account.
- Login Page - Users will be able to log in to their registered account.
- Homepage “My Tasks” - Users will see a collated list of all upcoming tasks among all to-do lists. Users will also be able to view a list of created projects on the left panel.
- Project Page(s) - Users will see tasks to do for each specific to-do list.
- Completed Page - Users will see a collated list of all completed tasks among all to-do lists.
- Recently Deleted - Users will see a collated list of all deleted tasks among all to-do lists.
- Shared With Me Page - Users will see a collated list of all to-do lists shared with them by other users.

Components:

- Dashboard (Create New Task and Filter Buttons)
- Table Items (Checkbox, Title Name, Project Select, Due Date, and Priority level)
- Menu Side Panel (My Tasks, Recently Deleted, My Projects, and Share with Me)
- Task Details Side Panel (Task title input field, Task description input field, Project selection, due date selection, and Priority selection)

1.3.1.3 Hardware Interfaces

Checkmate is a web-based to-do list application. There are no hardware interfaces that are applicable to this to-do list application. The application is expected to be accessible through a desktop or laptop to which may have specific hardware characteristics unrelated to the CheckMate to-do list application.

1.3.1.4 Software Interfaces

Software Interface requirements are as follows:

- OpenLiberty (Probably 24.0)
- JSON Web Tokens
- SQLite
- SSO v. Username, password, login

1.3.1.5 Communication Interfaces

The communication interfaces within the CheckMate application will be between the frontend user interface, the backend database, and the designated browser.

1.3.1.6 Memory Constraints

Memory Constraints are not applicable to the CheckMate to-do list application.

1.3.1.7 Site Adaptation

Site adaptation is not applicable to the CheckMate to-do list application.

1.3.2 Product Functions

The system shall provide the following functionality to its users:

- The ability to login and use the app using social login accounts (Github)
- The ability to create multiple to-do lists
- The ability to create and sort tasks within to-do lists

- The ability to view a collated list of all tasks within different to-do lists all at once or tasks within an individual list only
- The ability to sort tasks by priority level or due date or both
- The ability to add a due date to time sensitive tasks based on calendar dates
- The ability to be notified of time sensitive tasks while using the system
- The ability to share lists of tasks with other users
    - Share button

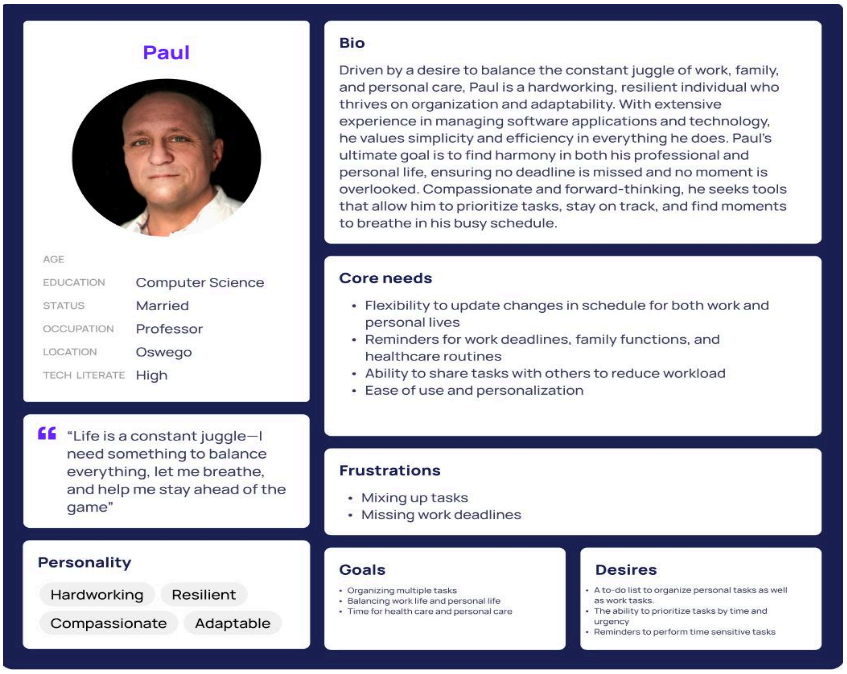
1.3.3 User characteristics

The generalized characteristics of our intended users are as follows:

- Everyday users of to-do list products who have a range of experience with computers, from the most minimal experience to more advanced experience
- No complex information is expected or needed to use this system.
- The users of the system are not strictly intended to use it for formal or workplace purposes.

1.3.3.1 Personas

Persona 1 - Paul



Paul

AGE	
EDUCATION	Computer Science
STATUS	Married
OCCUPATION	Professor
LOCATION	Oswego
TECH LITERATE	High

“Life is a constant juggle—I need something to balance everything, let me breathe, and help me stay ahead of the game”

Personality

Hardworking Resilient Compassionate Adaptable

Bio

Driven by a desire to balance the constant juggle of work, family, and personal care, Paul is a hardworking, resilient individual who thrives on organization and adaptability. With extensive experience in managing software applications and technology, he values simplicity and efficiency in everything he does. Paul's ultimate goal is to find harmony in both his professional and personal life, ensuring no deadline is missed and no moment is overlooked. Compassionate and forward-thinking, he seeks tools that allow him to prioritize tasks, stay on track, and find moments to breathe in his busy schedule.

Core needs

- Flexibility to update changes in schedule for both work and personal lives
- Reminders for work deadlines, family functions, and healthcare routines
- Ability to share tasks with others to reduce workload
- Ease of use and personalization

Frustrations

- Mixing up tasks
- Missing work deadlines

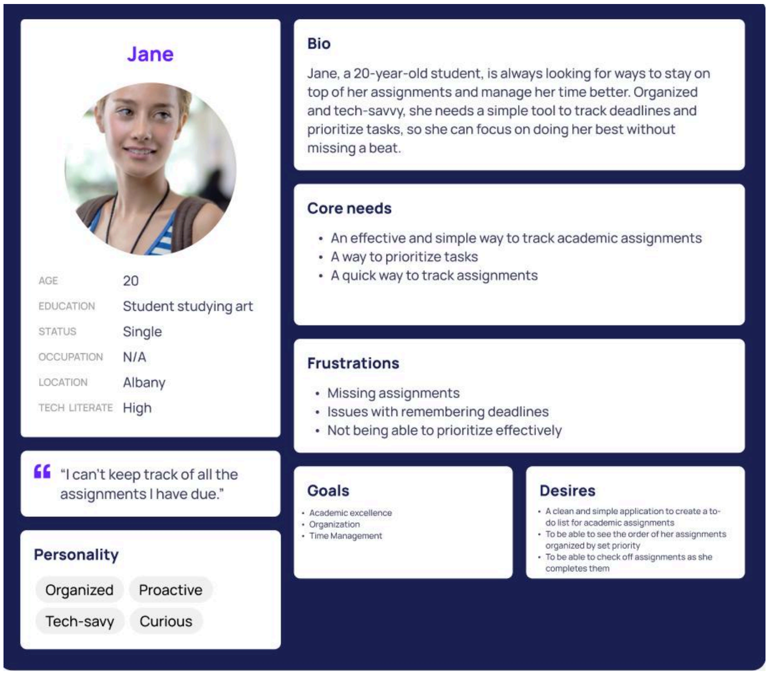
Goals

- Organizing multiple tasks
- Balancing work life and personal life
- Time for health care and personal care

Desires

- A to-do list to organize personal tasks as well as work tasks.
- The ability to prioritize tasks by time and urgency
- Reminders to perform time sensitive tasks

Persona 2 - Jane



Jane

AGE	20
EDUCATION	Student studying art
STATUS	Single
OCCUPATION	N/A
LOCATION	Albany
TECH LITERATE	High

"I can't keep track of all the assignments I have due."

Personality

Organized Proactive Tech-savy Curious

Bio

Jane, a 20-year-old student, is always looking for ways to stay on top of her assignments and manage her time better. Organized and tech-savvy, she needs a simple tool to track deadlines and prioritize tasks, so she can focus on doing her best without missing a beat.

Core needs

- An effective and simple way to track academic assignments
- A way to prioritize tasks
- A quick way to track assignments

Frustrations

- Missing assignments
- Issues with remembering deadlines
- Not being able to prioritize effectively

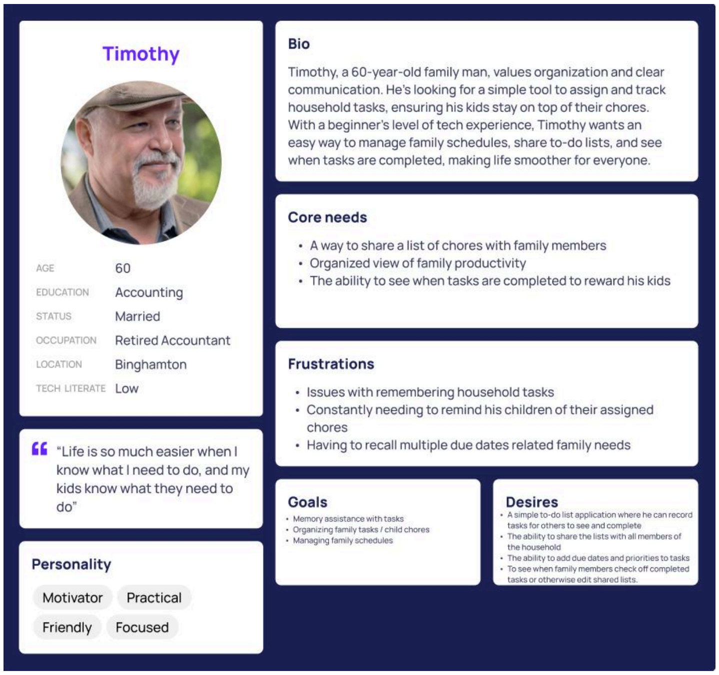
Goals

- Academic excellence
- Organization
- Time Management

Desires

- A clean and simple application to create a to-do list for academic assignments
- To be able to see the order of her assignments organized by set priority
- To be able to check off assignments as she completes them

Persona 3 - Timothy



Timothy

AGE	60
EDUCATION	Accounting
STATUS	Married
OCCUPATION	Retired Accountant
LOCATION	Binghamton
TECH LITERATE	Low

“Life is so much easier when I know what I need to do, and my kids know what they need to do”

Personality

Motivator Practical Friendly Focused

Bio

Timothy, a 60-year-old family man, values organization and clear communication. He's looking for a simple tool to assign and track household tasks, ensuring his kids stay on top of their chores. With a beginner's level of tech experience, Timothy wants an easy way to manage family schedules, share to-do lists, and see when tasks are completed, making life smoother for everyone.

Core needs

- A way to share a list of chores with family members
- Organized view of family productivity
- The ability to see when tasks are completed to reward his kids

Frustrations

- Issues with remembering household tasks
- Constantly needing to remind his children of their assigned chores
- Having to recall multiple due dates related family needs

Goals

- Memory assistance with tasks
- Organizing family tasks / child chores
- Managing family schedules

Desires

- A simple to-do list application where he can record tasks for others to see and complete
- The ability to share the lists with all members of the household
- The ability to add due dates and priorities to tasks
- To see when family members check off completed tasks or otherwise edit shared lists.

1.3.3.2 User Stories

User Story 1:

Paul is a hardworking family man, and professional with a multitude of daily tasks. He is interested in finding a tool to assist him in finding balance between his personal, and professional life. He is most interested in having a to-do list that he can use in all areas of his life to assist with keeping track of deadlines, family functions, and healthcare appointments. He would like the ability to share tasks with others, and prefers simplified applications that allow for personalization and easy use.

- “As a working person with many home responsibilities, I need a way to organize tasks between work and home life.”
- “As a busy worker, I need a way to prioritize and organize the many tasks I need to complete in a day.”
- “As a forgetful person, I need to be reminded of time sensitive tasks”
- “I’d really like the option to be able to share lists with different people, it helps with collaboration”

User Story 2:

Jane is a young adult student who appreciates learning new ways to stay organized with her class assignments and improve her time management. She is tech savvy and looking to find a to-do list application that can assist her in keeping track of deadlines and prioritizing her tasks so that she can find a great balance between doing well in her studies and other hobbies.

- "As a student, I need an efficient way to document all the tasks I need to complete."
- "It would be so nice to see all the things I need to do, all together in one list so I can plan my week better"
- "To do lists only work well for me if I can add a due date, otherwise I may forget"
- "I need it to be simple, I don't want to add to my workload"

User Story 3:
Timothy is a late middle aged father who prides himself in being organized and an excellent communicator. He is looking for a simple tool that would allow him to create, assign, and track household tasks to ensure his children are well aware of their chores. With minimal tech experience, Timothy, would like a consistent and easy way of managing his families schedules, share lists amongst each other, view when tasks are completed to ensure smoother days for everyone in his household.

- "As a parent, I need an easy way to share tasks with my children and remind them of their responsibilities."
- "As someone who is bad at managing passwords, I need the ability to login using accounts I've already created for other services."
- "I want to be able see what chores/tasks my kids complete, so I can reward them"
- "The tool needs to be simple, I am not tech savvy. I don't want to feel like I am learning too much"

1.3.4 Limitations

The limitations regulating the development of this system are as follows:

- The system must be completely developed before December 6, 2024.
- The system must be hosted on the existing server hardware provided by the university.
- The system must comply with WCAG accessibility standards.
- The system must be created on a limited budget of $0.

1.4 Definitions

Term	Definition
Example: *context of use*	*users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used*
System Under Development	The system which is actively being developed.
FR	Functional Requirement - Requirements which describe a distinct function performed by the system.
UR	Usability Requirement - Requirements which relate to the enforcement of usability guidelines within the system.
DR	Database Requirement - Requirements which outline the structure and function of the database used in the system.
CR	Constraint Requirement - Requirements which constrain the development process, such as guidelines and restrictions that must be met.
SOR	Solution Oriented Requirement - A requirement which directly enforces a development decision used in the implementation of the system.
WCAG	Web Content Accessibility Guidelines
SMD	State Machine Diagram - A diagram which depicts the transitions and state space of the system.
UML	Unified Modeling Language - A language in which diagrams that illustrate the functioning of a system can be depicted.

2. References

References related to this project are listed below:

- WCAG Guidelines for Accessibility
- Design Specifications Document - Version 1.0
- Wireframes
- Interactive Prototype
- User Manual

3. Requirements

3.1 Functional Requirements

ID	Natural Language Requirement	Artifact Type	Requirement Type	Quality Property	Source
FR1	The user shall be able to create and manipulate multiple to-do lists.	Goal	Functional	N/A	Stakeholder Interview 1
FR1.1	The user shall be able to view created lists on separate individual pages.	Goal	Functional	N/A	Stakeholder Interview 1
FR2	The user shall be able to create, edit, complete, and delete individual tasks within a list.	Goal	Functional	N/A	Stakeholder Interview 1
FR2.1	The user shall be able to create recurring tasks.	Goal	Functional	N/A	Stakeholder Interview 1
FR3	The user shall be able to share to-do lists with others.	Goal	Functional	N/A	Stakeholder Interview 1
FR4	The user should be able to move completed items from the “Upcoming” main page to the “Completed” page.	Goal	Functional	N/A	Stakeholder Interview 1
FR4.1	The system shall separate completed tasks from tasks that	Goal	Functional	N/A	Stakeholder Interview 1

	are still pending.				
FR5	The user should have the ability to tag tasks as high, medium, and low priority.	Goal	Functional	N/A	Stakeholder Interview 1
FR5.1	The system shall provide the user with the ability to sort tasks by priority, due date, or both.	Goal	Functional	N/A	Stakeholder Interview 2
FR6	The system shall provide the user the ability to add due dates to tasks.	Goal	Functional	N/A	Stakeholder Interview 1
FR7	The system shall provide a login system to the user.	Goal	Functional	N/A	Stakeholder Interview 1
FR7.1	The system shall provide the user the ability to use social login.	Goal	Functional	N/A	Stakeholder Interview 3
FR 7.1.2	The system shall allow users to login using their GitHub credentials via social login.	Goal	Functional	N/A	Stakeholder Interview 3
FR7.2	The system shall provide the user the ability to register an account using their email and a password.	Goal	Functional	N/A	Stakeholder Interview 3
FR7.3	The system shall provide the user with the ability to login using an account they have previously registered using the system.	Goal	Functional	N/A	Stakeholder Interview 3

FR8	The system shall display currently active tasks in the main view of the user interface.	Goal	Functional	N/A	Stakeholder Interview 1
FR9	The system shall provide the user with the ability to view all active tasks from all lists at the same time.	Goal	Functional	N/A	Stakeholder Interview 2

3.2 Performance Requirements

Not applicable to CheckMate application.

3.3 Usability Requirements

ID	Natural Language Requirement	Artifact Type	Requirement Type	Quality Property	Source
UR1	The user shall be able to access the system from a laptop or desktop PC.	Goal	Functional	N/A	Stakeholder Interview 1
UR1.1	The user shall be able to access the system through Firefox & Chrome.	Goal	Functional	N/A	Stakeholder Interview 1
UR2.1	The user shall be able to use “minimal effort” when sorting a task.	Goal	Quality	Operability	Stakeholder Interview 1
UR2.2	The system shall maintain a simple and clean interface.	Goal	Quality	Operability / Usability / Attractiveness	Stakeholder Interview 1
UR3	The system shall resize graphical elements of the user	Goal	Quality	Operability / Usability / Attractiveness	Stakeholder Interview 2

	interface to fit any size display.				
UR4	The system shall depict casual tones in the graphical elements of its user interface	Goal	Quality	Attractiveness	Stakeholder Interview 2
UR5	The system shall comply with WCAG accessibility guidelines as detailed in the Design Specification document. (See 3.6 - Design Constraints)	Goal	Constraint Requirement	Accessibility	Stakeholder Interview 1

3.4 Interface Requirements

Interface Elements:

- See Section 1.3.1.2

ID	Natural Language Requirement	Artifact Type	Requirement Type	Quality Property	Source
IR1	The system's GUI shall display a login interface to the user upon first access.	Goal	Functional	N/A	Stakeholder Interview 1
IR1.1	The login interface shall contain text fields for the user's email address, password.	SOR	Functional	N/A	Stakeholder Interview 1
IR1.2	The login interface shall contain a social login button.	SOR	Functional	N/A	Stakeholder Interview 1
IR1.3	The login interface shall contain buttons with the option of "Sign Up" and "Log	SOR	Functional	N/A	Stakeholder Interview 1

	In”.				
IR 1.3.1	When the “Sign Up” button is pressed, it shall take the user to the registration screen				
IR 1.3.2	When the “Sign Up” button is pressed nth time, and a valid email and confirmed password are provided by the user, the system shall create an account for the user as further detailed in Section 3.5 of this document.	SOR	Functional	N/A	Goal Refinement
IR 1.3.2.1	When the user completes a valid registration as detailed in IR1.3.2, the system shall change interfaces to the My Tasks view.	SOR	Functional	N/A	Goal Refinement
IR 1.3.3	When the “Login” button is pressed, and valid credentials are provided by the user, the system shall change interfaces to the My Tasks view.	SOR	Functional	N/A	Goal Refinement
IR2	The main interface shall contain a menu panel on the left, a view of tasks in the center, and a task editor panel on the right	SOR	Functional	N/A	Usability
IR3	The system shall provide the user with the ability to select different task views using buttons in the menu side panel.	SOR	Functional	N/A	Usability

IR 3.1	The system shall provide a list of buttons in the menu side panel corresponding to the user’s created lists	SOR	Functional	N/A	Usability
IR 3.2	When the My Tasks button in the menu side panel is pressed, the system shall display all currently active and recently completed tasks in the task list panel.	SOR	Functional	N/A	Stakeholder Interview 2
IR 3.3	When the Completed tab is pressed, the system shall display all archived completed tasks in the task list panel corresponding to the current task list view.	SOR	Functional	N/A	Stakeholder Interview 2
IR4	The system shall display a list of tasks in the center panel of the main interface.	SOR	Functional	N/A	Usability
IR 4.1	The system shall display the task name, due date, and priority for each visible task.	SOR	Functional	N/A	Usability
IR 4.2	The system shall provide a checkbox next to each listed task to be toggled upon task completion.	SOR	Functional	N/A	Stakeholder Interview 1
IR4.3	The system shall provide a delete button next to each listed task to allow the user to delete tasks.	SOR	Functional	N/A	Stakeholder Interview 1

IR4.4	The system shall display tasks in the list with their data organized by columns.	SOR	Functional	N/A	Usability
IR4.5	At the top of the task list, the system shall provide sorting buttons above each data column for the user to sort by due date, project name, and priority.	SOR	Functional	N/A	Stakeholder Interview 1 / 2 / Usability
IR5	The system shall provide a task editor panel on the right side of the main interface.	SOR	Functional	N/A	Usability
IR5.1	The task editor panel shall contain text fields for the user to input data for the task name, description, and project name.	SOR	Functional	N/A	Usability
IR5.2	The task editor panel shall provide a date selector for the user to input a due date for the current task.	SOR	Functional	N/A	Usability
IR5.3	The task editor panel shall contain drop down lists enabling the user to select task priority, as well as the task's repeat schedule.	SOR	Functional	N/A	Usability

3.5 Logical Database Requirements

ID	Natural Language Requirement	Artifact Type	Requirement Type	Quality Property	Source
DR1	The database shall	Goal	Functional	N/A	Engine Team

	store information based on tables for tasks, lists, and users.				Collaboration
DR1.1	The database shall use supporting tables to link data as necessary for features such as sharing lists.	Goal	Functional	N/A	Engine Team Collaboration
DR2	The database shall store all data in SQL tables.	SOR	Functional	N/A	Engine Team
DR2.1	The system shall use SQLite to facilitate SQL data storage and retrieval.	SOR	Functional	N/A	Engine Team
DR3	The database shall store all provided information for each task.	Goal	Functional	N/A	Engine Team / Stakeholder Collaboration
DR3.1	The database shall store task information in an SQL table with columns for task name, list ID, task ID, due date, priority, repeat schedule, description, and deletion status.	SOR	Functional	N/A	Engine Team / Stakeholder Collaboration
DR3.2	The database shall use the task ID column as the primary key for the task data table, to prevent entering rows into the database with duplicate task IDs.	SOR	Functional	N/A	Engine Team Collaboration
DR4	The database shall store all provided information for each list of tasks.	Goal	Functional	N/A	Engine Team / Stakeholder Collaboration

DR4.1	The database shall store list information in an SQL table with columns for list name, list ID, list owner ID, and deletion status.	SOR	Functional	N/A	Engine Team / Stakeholder Collaboration
DR4.2	The database shall use the list ID column as the primary key for the list data table, to prevent entering rows into the database with duplicate list IDs.	SOR	Functional	N/A	Engine Team Collaboration
DR5	The database shall store all provided user data.	Goal	Functional	N/A	Engine Team / Stakeholder Collaboration
DR5.1	The database shall store user data in an SQL table with columns for email and password.	SOR	Functional	N/A	Engine Team Collaboration
DR5.2	The database shall use the email column as the primary key for the user data table, to prevent entering rows into the database with duplicate emails.	SOR	Functional	N/A	Engine Team Collaboration
DR5.3	In the case where a user registers their account using social login, the database shall store a record of their email along with an empty password field, to prevent users from registering an account if they have already created one using social login.	SOR	Functional	N/A	Engine Team Collaboration

3.6 Design Constraints

ID	Natural Language Requirement	Artifact Type	Requirement Type	Quality Property	Source
CR1	The system shall comply with WCAG accessibility guidelines as detailed in the Design Specification document.	Goal	Constraint Requirement	Accessibility	Stakeholder Interview 1
CR1.1	The system shall space lines of text apart by at least 1.5 times the size of the font used	SOR	Functional / Quality	Attractiveness Usability Accessibility	WCAG Guidelines
CR1.2	The system shall space paragraphs apart by at least 2 times the size of the font used	SOR	Functional / Quality	Attractiveness Usability Accessibility	WCAG Guidelines
CR1.3	The system shall space individual letters apart by at least 0.12 times the size of the font used	SOR	Functional / Quality	Attractiveness Usability Accessibility	WCAG Guidelines
CR1.4	The system shall space words apart by at least 0.16 times the size of the font used	SOR	Functional / Quality	Attractiveness Usability Accessibility	WCAG Guidelines
CR1.5	The system must not distract or harm users with animations	Goal	Quality	Usability Accessibility	WCAG Guidelines
CR 1.5.1	The system shall use minimal animations	SOR	Functional / Quality	Usability Accessibility	WCAG Guidelines
CR 1.5.2	The system shall provide the user with the ability to reduce motions and animations	SOR	Functional / Quality	Usability Accessibility	WCAG Guidelines

CR1.6	The system must not use color as the sole signifier to distinguish between tasks and/or lists	Goal	Constraint Requirement	Usability Accessibility	WCAG Guidelines / Usability Team

3.7 Software System Attributes

To ensure data security for users, the application is being developed with SQLite and JSON Web tokens (JWTs). To reduce the possibility of a hacker being able to gain access to the user data, the developer can use "personal statements" which is a SQL query that allows the developer to separate a SQL command from the user input data. This reduces the chances of a SQL injection happening.

Also, JSON Web Tokens will allow for a secure transmission of user information between the server and the backend database. For example: when a user is logging into Checkmate it will be verified by the backend database; the server would then generate a JWT (secret key known by the server) to ensure the information is not compromised.

3.8 Supporting Information

[Add any additional information needed to understand the SRS, include things like background information, problem descriptions, packaging instructions for code, sample input/output formats, etc.]

3.8.1 UML State Machine Diagrams

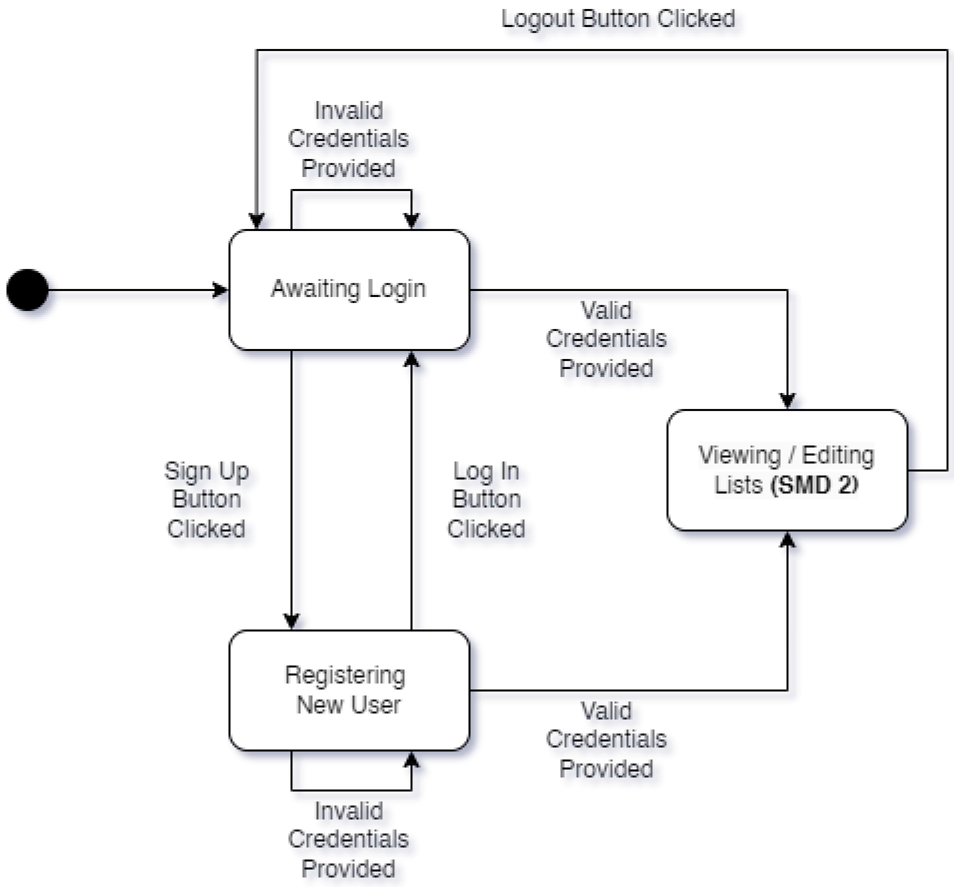
UML state machine diagrams depict the many states of a system and the transitions between these states. From these diagrams, every possible interaction between the user and the system can be described through a sequence of states and transitions. These state machine diagrams will provide context and understandability to the documented requirements of the system.

Syntax:

- Rounded rectangles depict states.
- Arrows depict events which cause transitions between states.
- Solid circles depict the start state of the diagram.
- Solid circles with a circle around them depict the end state of the diagram.

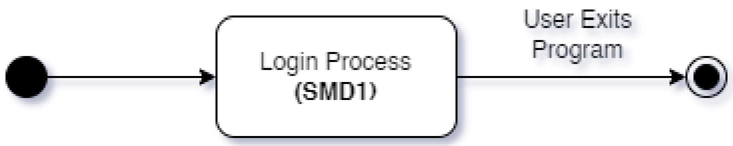
- A transition out of a state that has inner states can be triggered from any inner state.
- A transition into a state which has inner states continues at the inner start state.
- Transitions from a state into itself depict events which do not cause state changes.

SMD 1: Login Process


Logout Button Clicked
Invalid Credentials Provided
Awaiting Login
Valid Credentials Provided
Sign Up Button Clicked
Log In Button Clicked
Viewing / Editing Lists **(SMD 2)**
Registering New User
Valid Credentials Provided
Invalid Credentials Provided

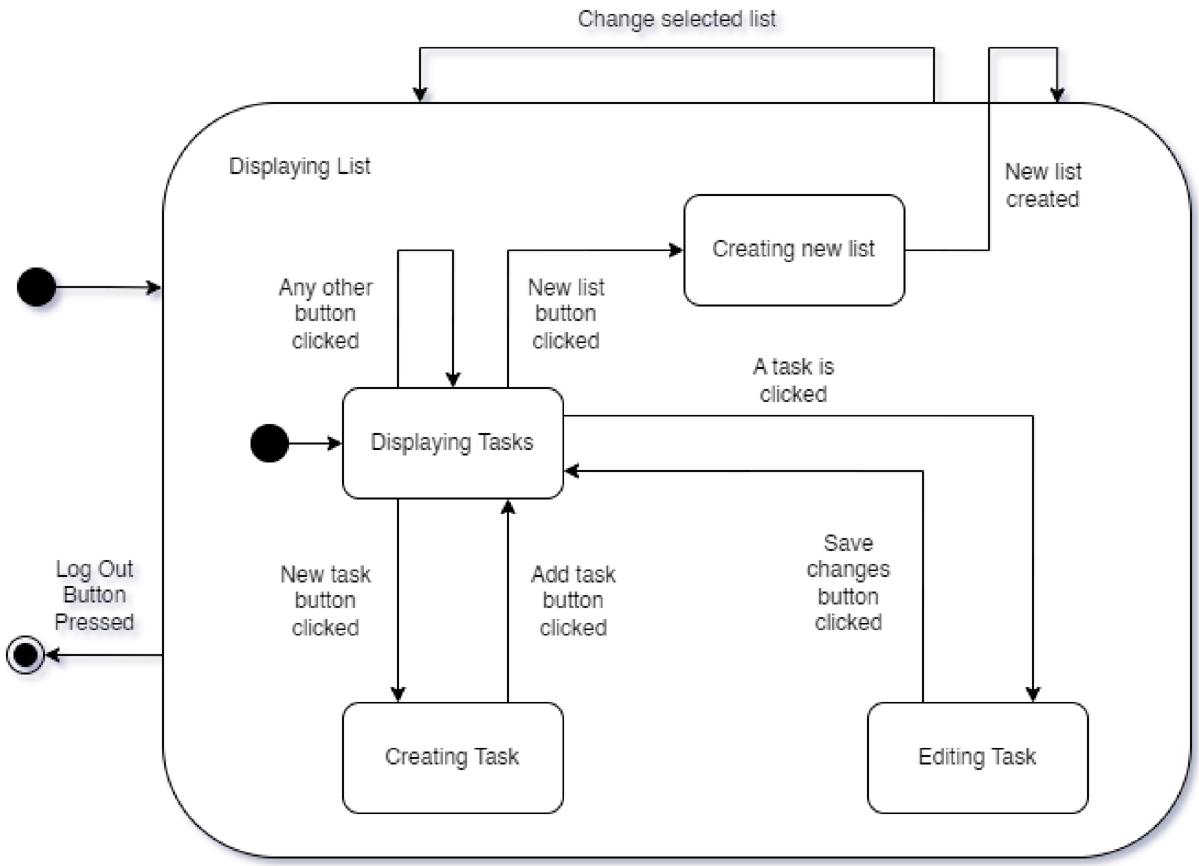
- This figure depicts a UML State Machine diagram of the login process as previously described in Interface Requirements, and how the state of the system changes in response to the inputs given by the user. See section **3.4** for detailed requirements specifying these interactions.

SMD 1.1: Global Program State



- This supporting figure depicts the global external state of the system, providing context that from any state within figure **SMD 1**, or any nested states therein, the user can exit that state by closing the program.

SMD 2: Viewing / Editing Lists



- This figure depicts the state machine diagram of the system while the user is logged in and viewing their lists and tasks. From any state within "displaying lists" the system can exit and return to **SMD 1** by pressing the log out button. See section **3.4** for detailed requirements specifying these interactions. For context, any transition from an inner state to the outer state, or from the outer state onto itself, continues at the start state within the outer state.

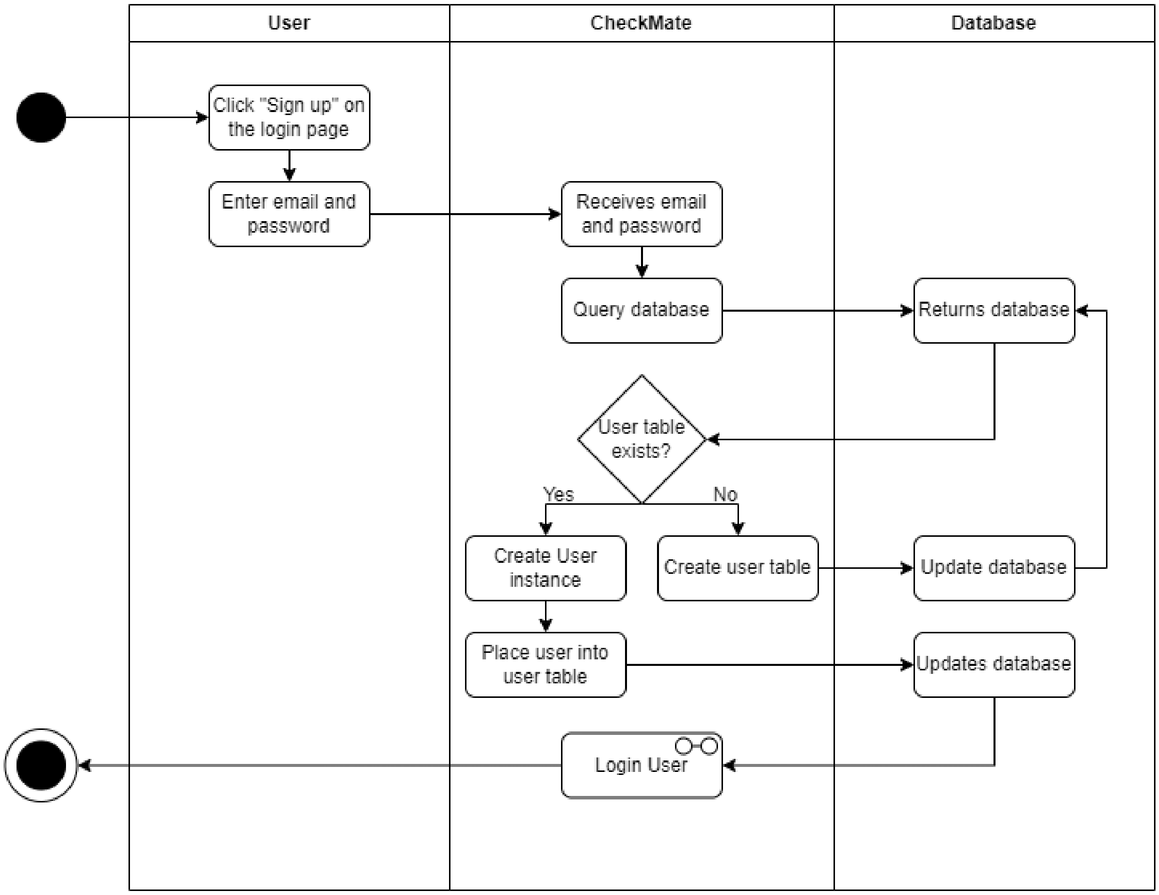
3.8.2 UML Activity Diagrams

UML (unified modeling language) activity diagrams depict the many interactions a user may have with the system. From these diagrams, many distinct sequences of interactions can be seen through the perspective of the user. These activity diagrams will make the workflow of the system's users explicit and understandable.

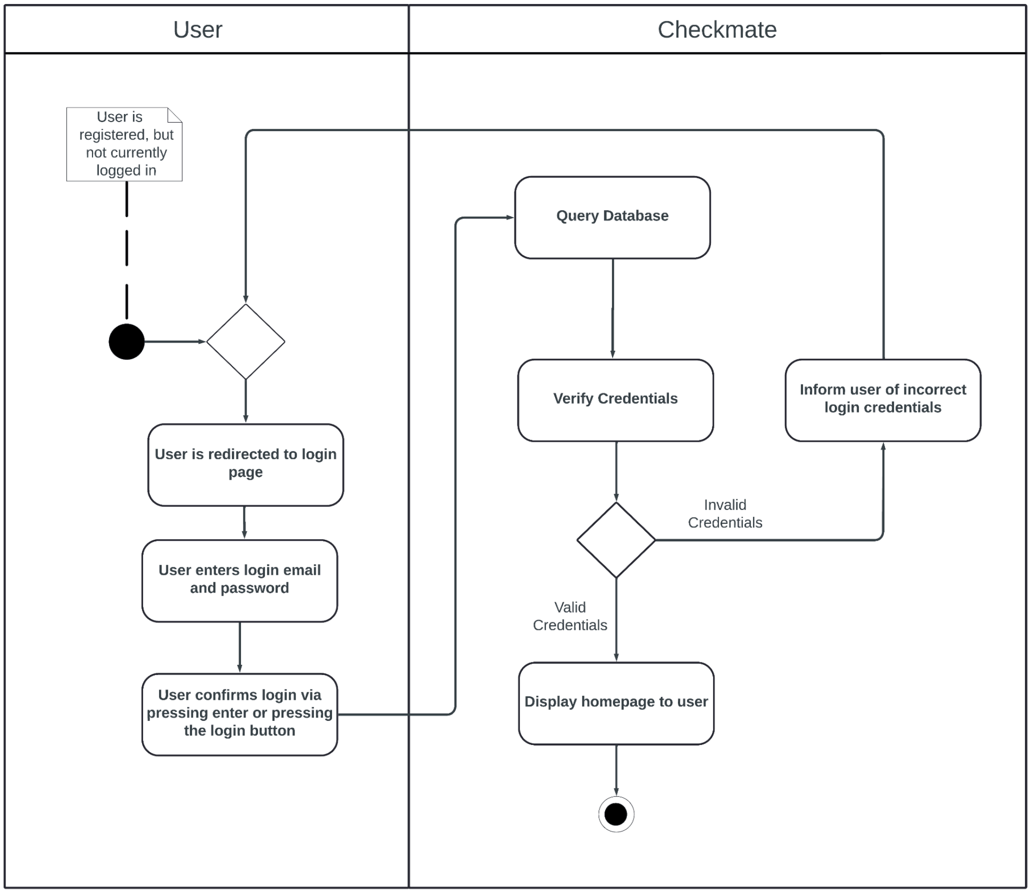
Syntax:

- Rounded rectangles depict activities.
- Arrows depict the workflow from one activity into another.
- Solid circles depict the start of the diagram.
- Solid circles with a circle around them depict the end of the diagram.
- Diamond nodes represent decision branches and merges, where the flow will continue one way out of the diamond based on the guard labels.
- One branch splitting into two without a decision node depicts a flow split, where the two branches will continue in parallel until merged, or until all parallel branches are killed.
- Swimlanes depict which context entity within the system is actively performing a given activity

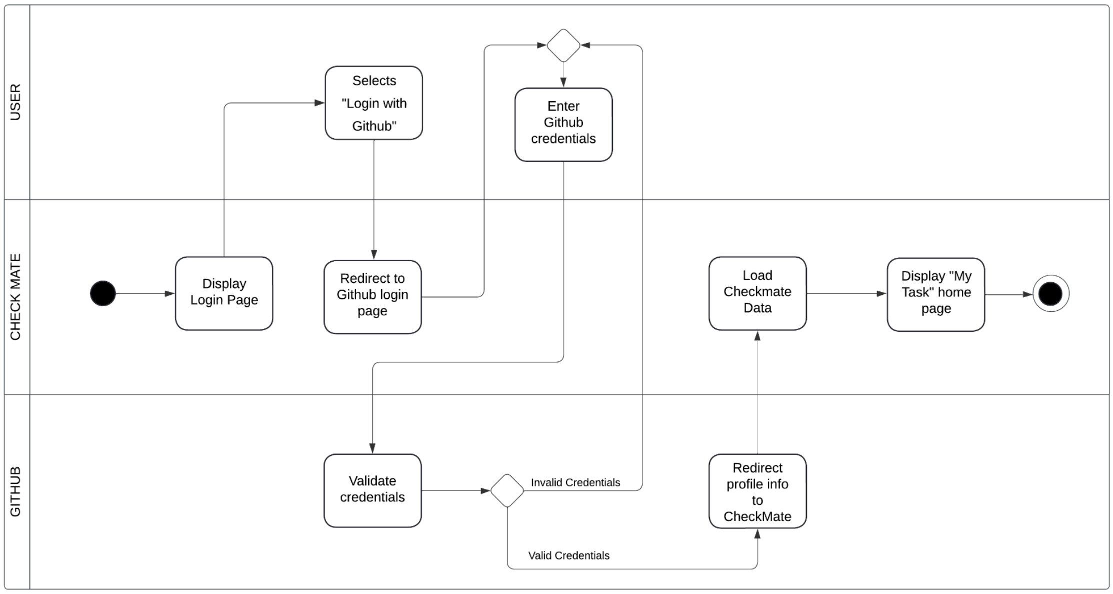
Activity: Create an Account



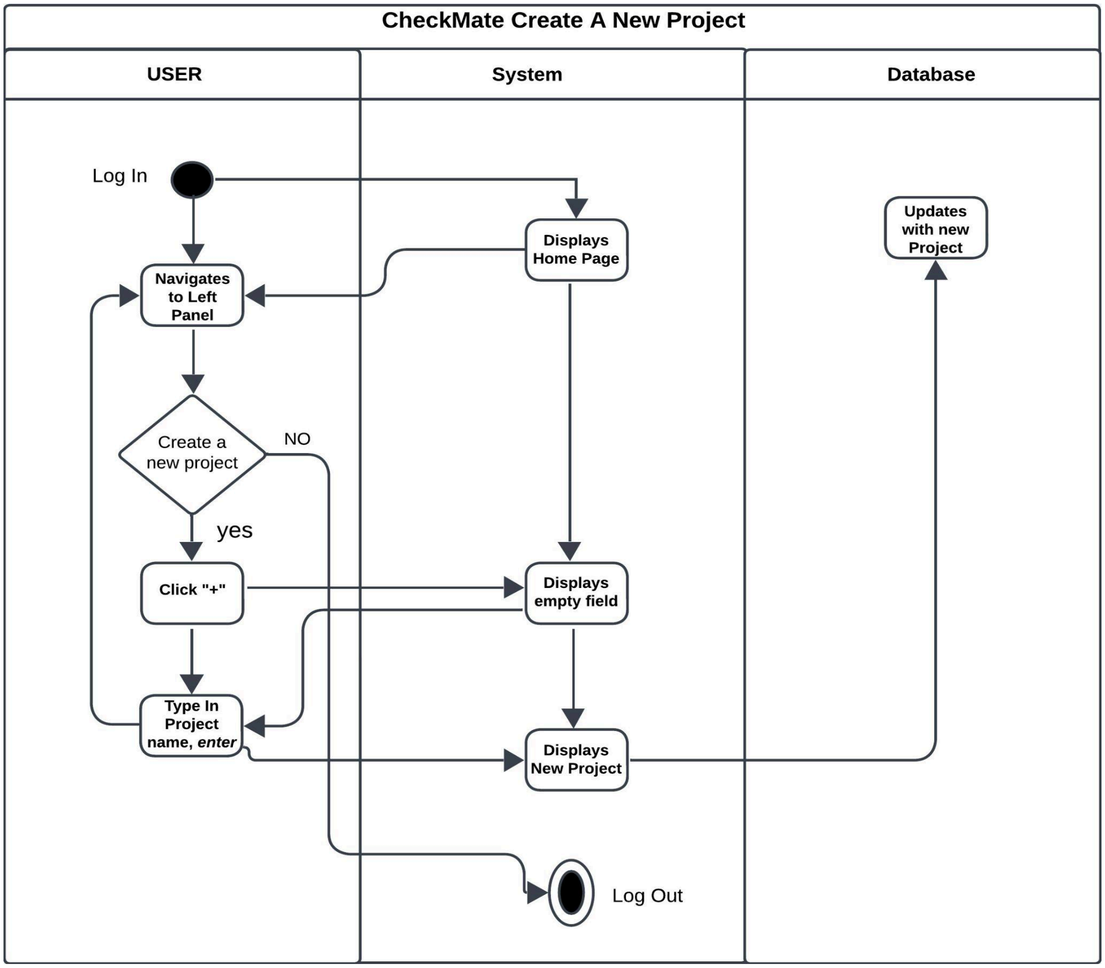
Activity: Native Login


User
Checkmate
User is registered, but not currently logged in
User is redirected to login page
User enters login email and password
User confirms login via pressing enter or pressing the login button
Query Database
Verify Credentials
Inform user of incorrect login credentials
Invalid Credentials
Valid Credentials
Display homepage to user

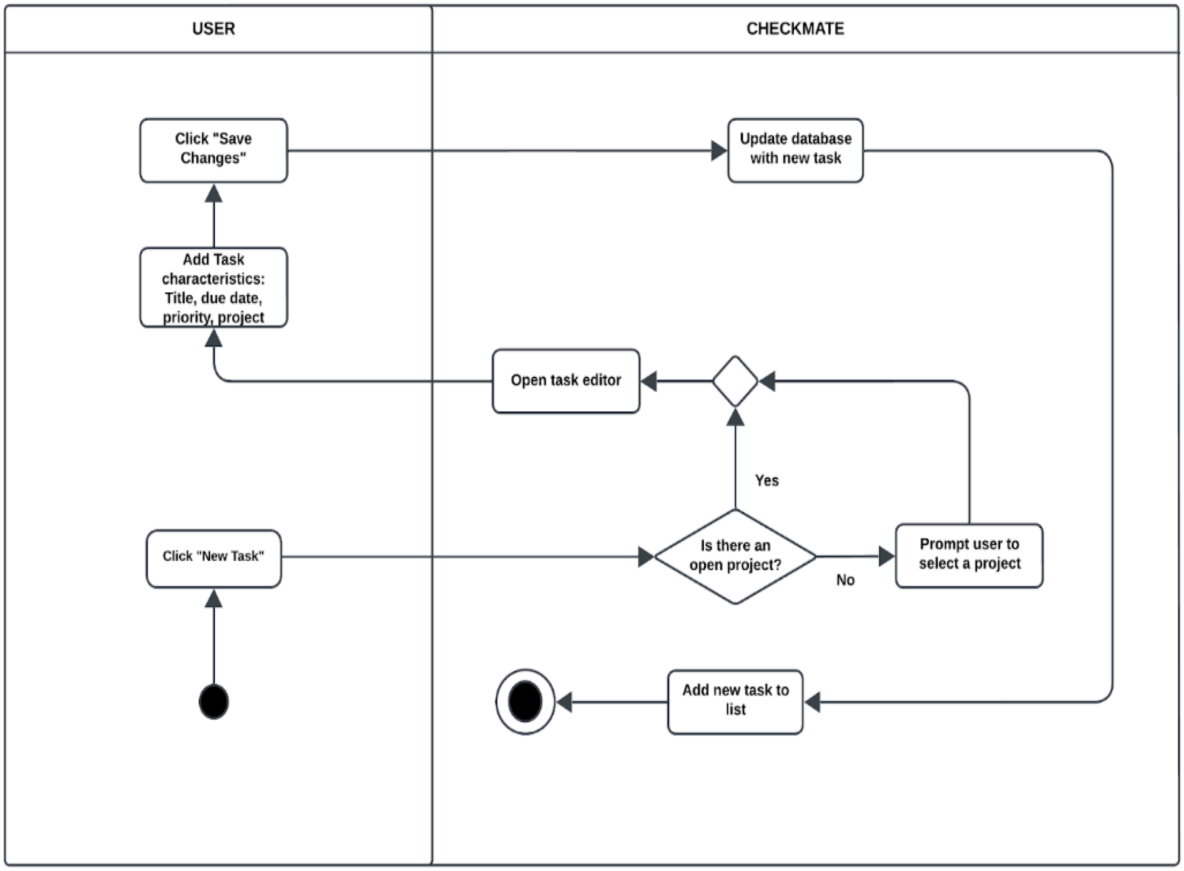
Activity: Social Login (Github)



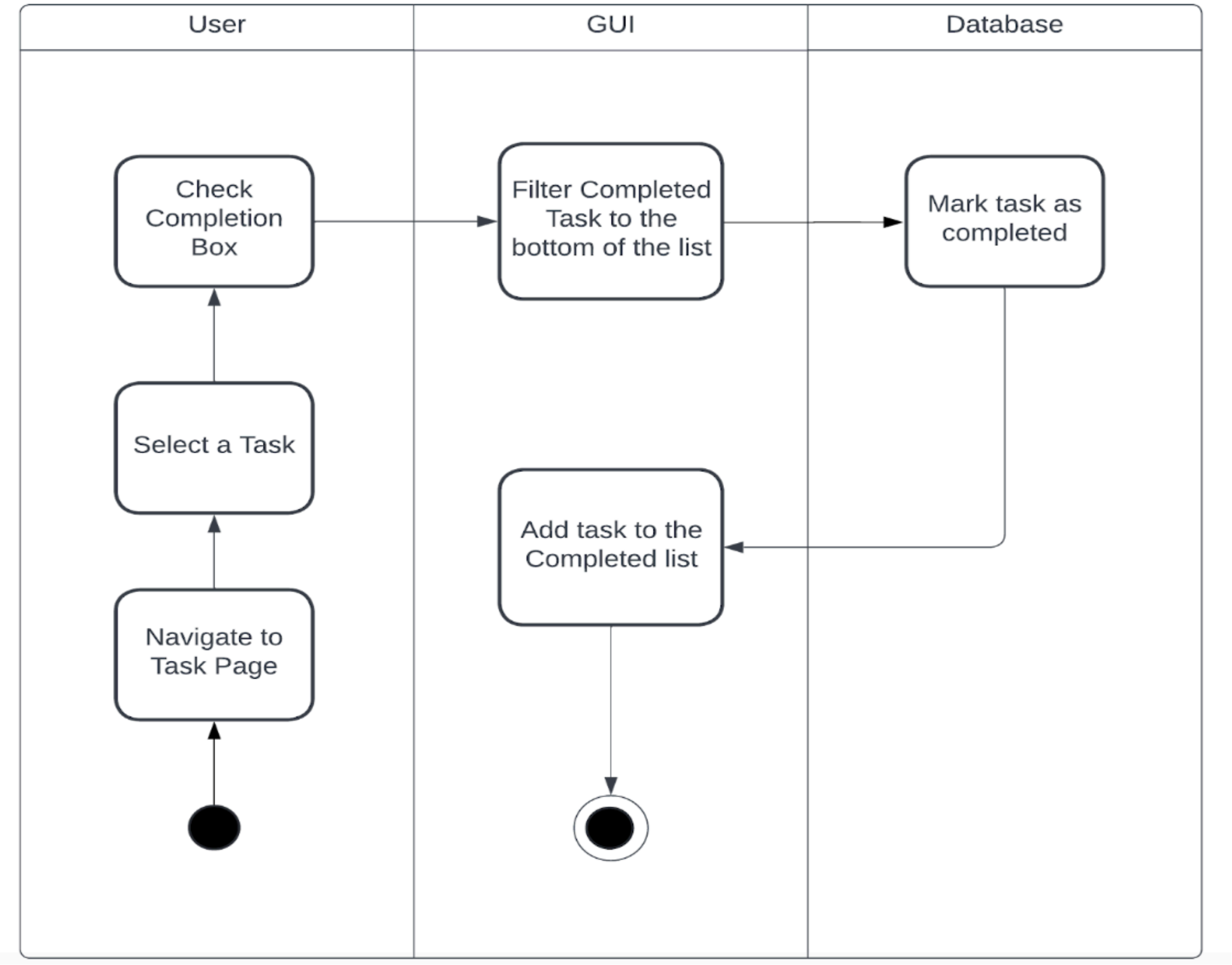
Activity: Create a Project



Create a Task


USER
CHECKMATE
Click "Save Changes"
Update database with new task
Add Task characteristics: Title, due date, priority, project
Open task editor
Yes
Click "New Task"
Is there an open project?
No
Prompt user to select a project
Add new task to list

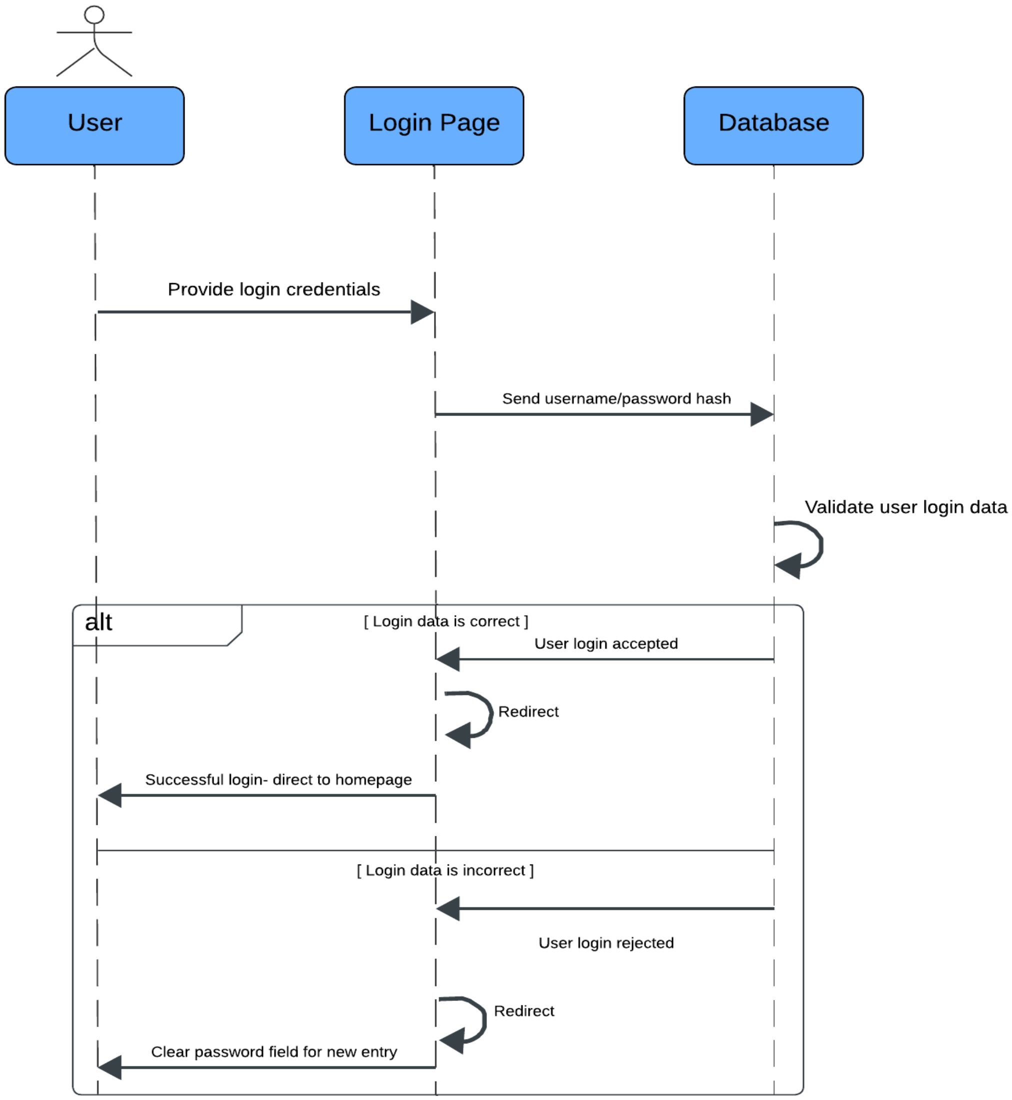
Activity: Completing a Task



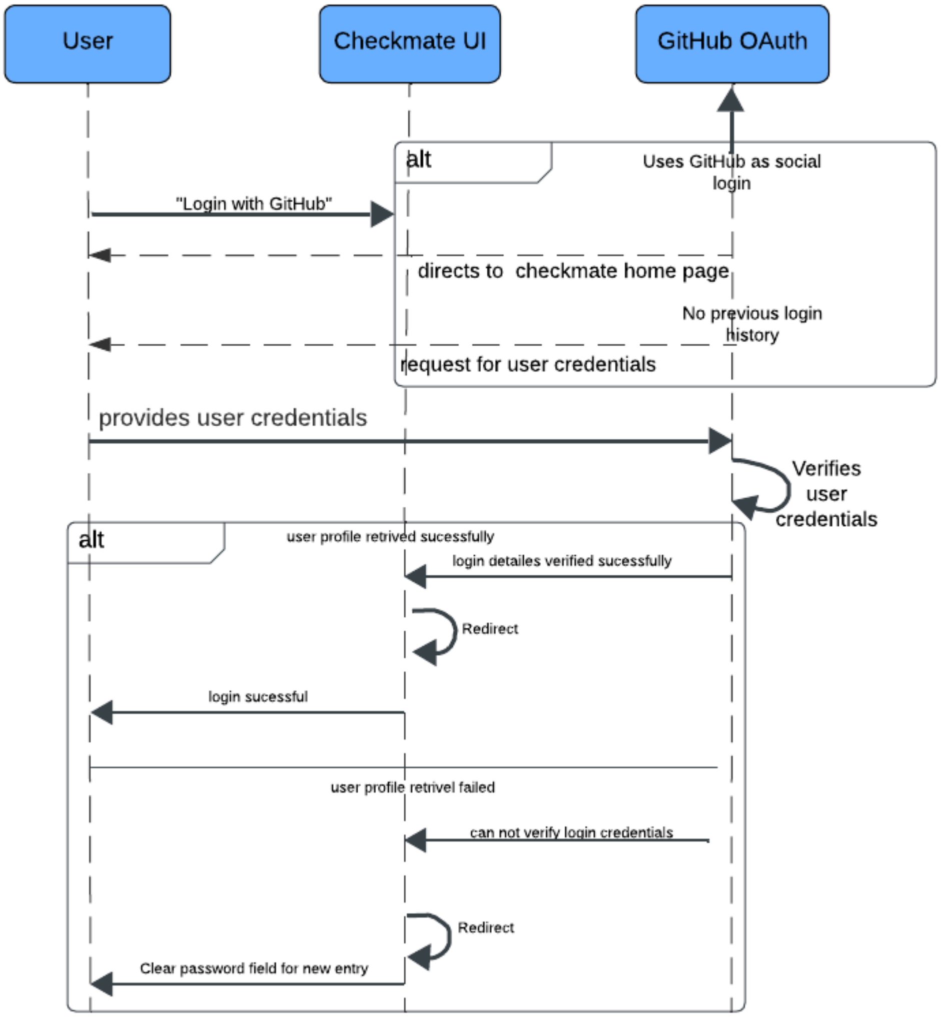
3.8.3 UML Sequence Diagrams

UML sequence diagrams depict an abstraction of the control / data flow of the system. From these diagrams, the general operational flow of the system, including its inputs and outputs, can be graphically depicted. Specifically, sequence diagrams are helpful for demonstrating interactions between the context entities of the system, for example, the user provides an input to the user interface, which sends that input to the database. This interaction is not only depicted as the separate events, as would be in an activity diagram, but as exchanges between context entities.

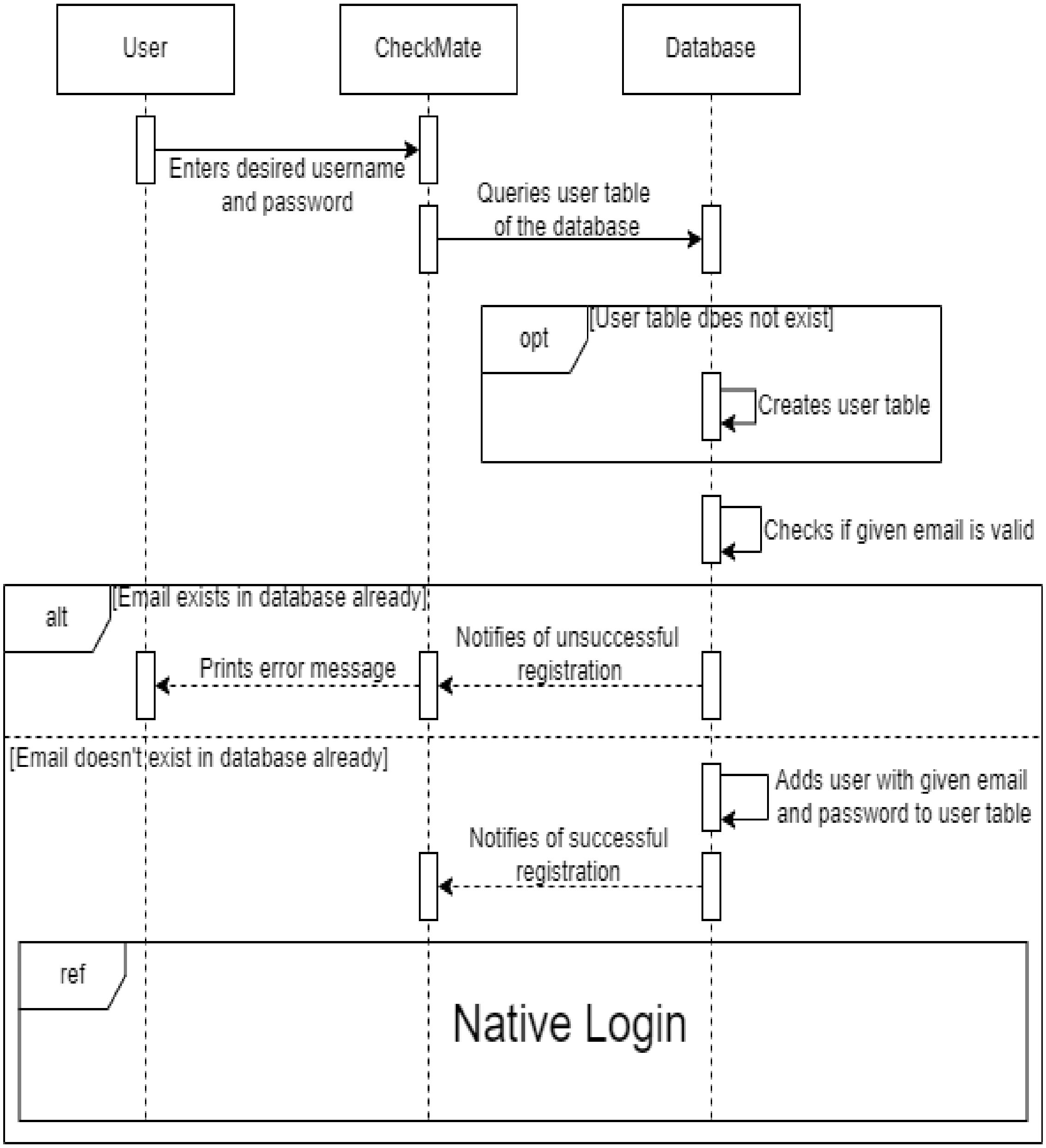
Sequence Diagram: Login Native


User
Login Page
Database
Provide login credentials
Send username/password hash
Validate user login data
alt
[Login data is correct]
User login accepted
Redirect
Successful login- direct to homepage
[Login data is incorrect]
User login rejected
Redirect
Clear password field for new entry

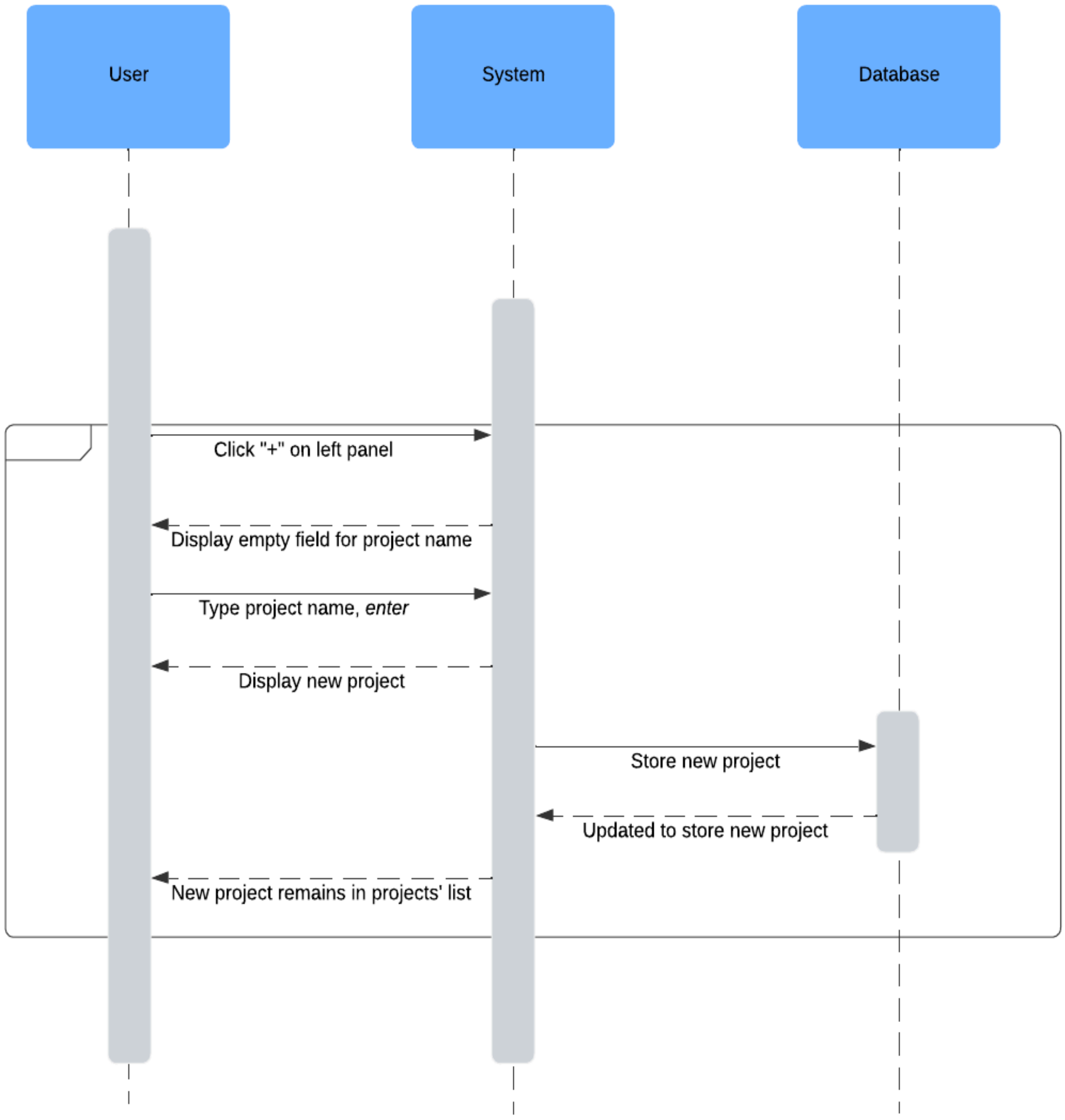
Sequence Diagram: Login Github


User
Checkmate UI
GitHub OAuth
alt
Uses GitHub as social login
"Login with GitHub"
directs to checkmate home page
No previous login history
request for user credentials
provides user credentials
Verifies user credentials
alt
user profile retrived sucessfully
login detailes verified sucessfully
Redirect
login sucessful
user profile retrivel failed
can not verify login credentials
Redirect
Clear password field for new entry

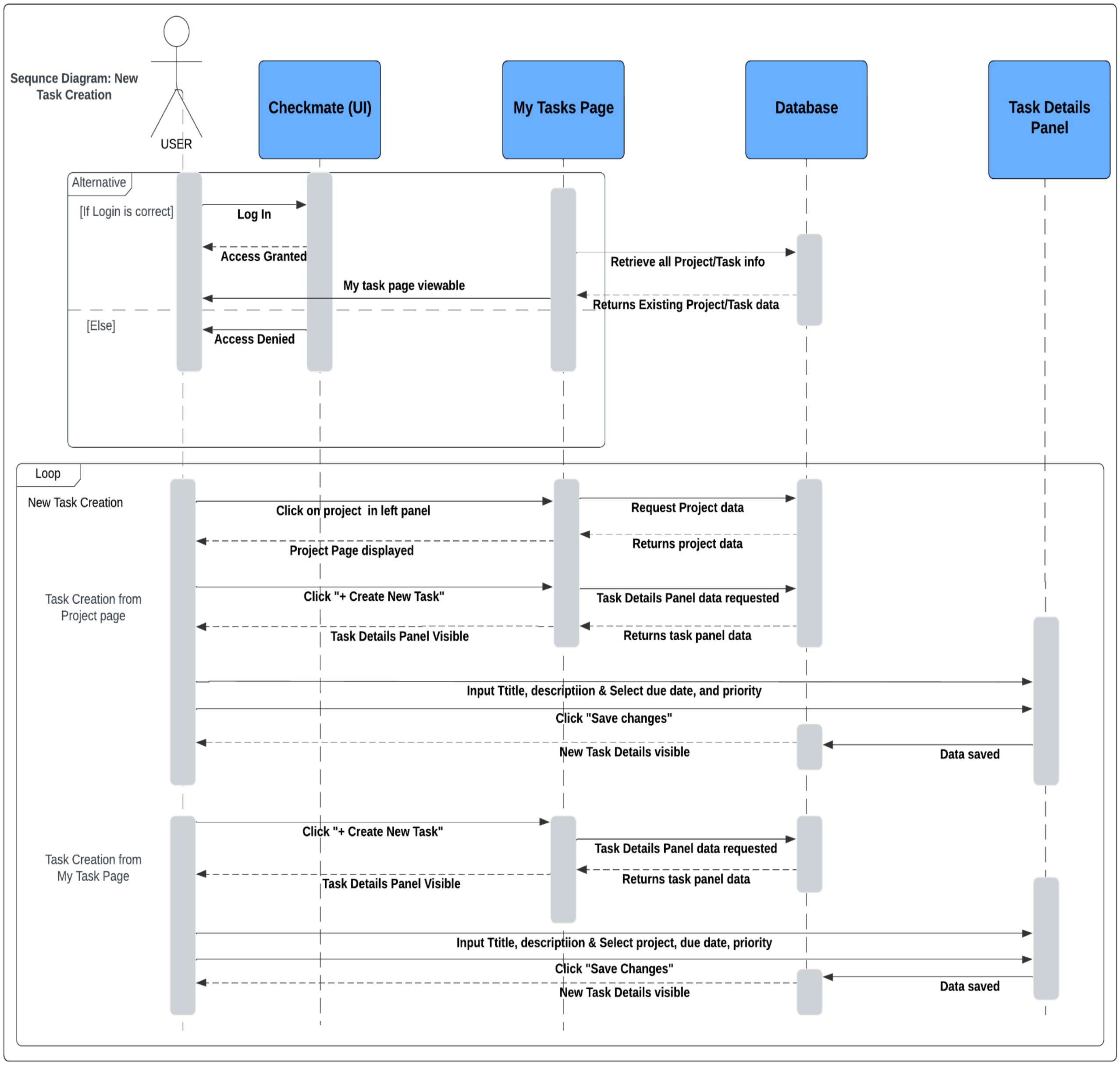
Sequence Diagram: Create Account


User
CheckMate
Database
Enters desired username and password
Queries user table of the database
opt
[User table does not exist]
Creates user table
Checks if given email is valid
alt
[Email exists in database already]
Notifies of unsuccessful registration
Prints error message
[Email doesn't exist in database already]
Adds user with given email and password to user table
Notifies of successful registration
ref
Native Login

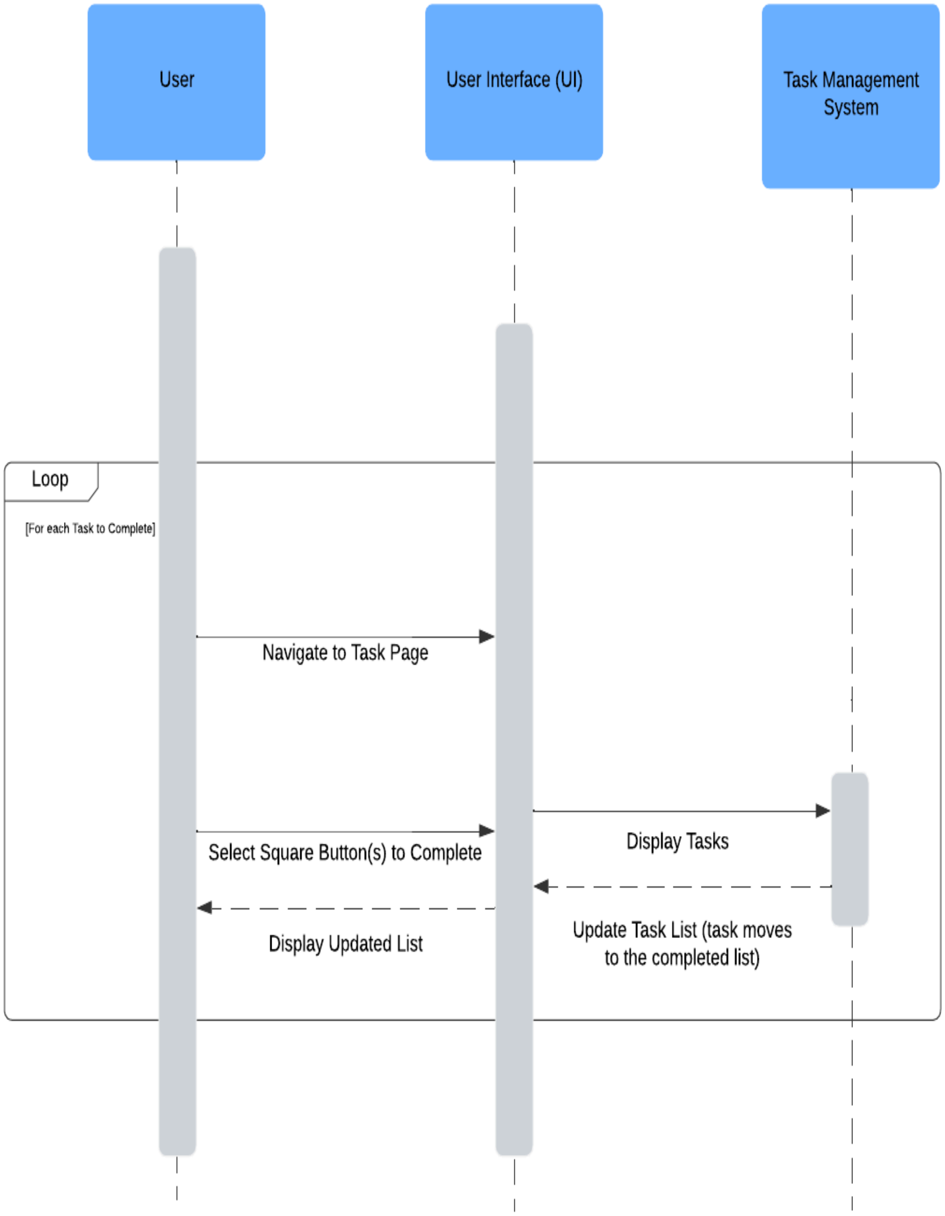
Sequence Diagram: Create Project


User
System
Database
Click "+" on left panel
Display empty field for project name
Type project name, *enter*
Display new project
Store new project
Updated to store new project
New project remains in projects' list

Sequence Diagram: Create New Tasks


Sequnce Diagram: New Task Creation
USER
Checkmate (UI)
My Tasks Page
Database
Task Details Panel
Alternative
[If Login is correct]
Log In
Access Granted
Retrieve all Project/Task info
My task page viewable
Returns Existing Project/Task data
[Else]
Access Denied
Loop
New Task Creation
Click on project in left panel
Request Project data
Returns project data
Project Page displayed
Task Creation from Project page
Click "+ Create New Task"
Task Details Panel data requested
Task Details Panel Visible
Returns task panel data
Input Ttitle, descriptiion & Select due date, and priority
Click "Save changes"
New Task Details visible
Data saved
Click "+ Create New Task"
Task Details Panel data requested
Task Creation from My Task Page
Returns task panel data
Task Details Panel Visible
Input Ttitle, descriptiion & Select project, due date, priority
Click "Save Changes"
New Task Details visible
Data saved

Sequence Diagram: Completing Task


User
User Interface (UI)
Task Management System
Loop
[For each Task to Complete]
Navigate to Task Page
Display Tasks
Select Square Button(s) to Complete
Display Updated List
Update Task List (task moves to the completed list)

4. Verification

4.1 Functions

[See sections 4.0, 3.1 for specific directions about what outputs should be included here.]

Test ID	Description	Input	Expected Output	Actual Output	Verified Requirement IDs
6	Login via GitHub (Windows)	User selects “Log in with GitHub” on login page	User is successfully logged in	User is successfully logged in	FR7 FR7.1 FR7.1.2
7	Login via GitHub (Unix)	User selects “Log in with GitHub” on login page	User is successfully logged in	User is successfully logged in	FR7 FR7.1 FR7.1.2
33	Sort by Date/priority Functioning (Moxie) (Windows) (Chrome)	User selects multiple sort buttons from a list view page	Successful compound sorting of tasks	Successful compound sorting of tasks	FR5 FR5.1 FR6
34	Sort by Date/priority Functioning (Gui-init-branch) (Windows)	User selects multiple sort buttons from a list view page	Successful compound sorting of tasks	Successful compound sorting of tasks	FR5 FR5.1 FR6
45	Creating a Task	User selects “Create New Task,” inputs data in relevant fields and then selects “Add New Task”	Successful creation, storage, and display of a new task	Successful creation, storage, and display of a new task	FR1 FR1.1 FR2 FR8 FR9
46	Checking Tasks	User selects the checkbox next to an existing task from any list view page	Task is marked as completed and moved to the completed tab	Task is marked as completed and moved to the completed tab	FR2 FR4 FR4.1 FR8 FR9

52	Registration (Local)	User selects “Sign Up” from the log in page, provides a valid email and password, and selects “Sign Up” once more	Successful registration of native login account	Successful registration of native login account	FR7 FR7.2
53	Login (Local)	User inputs a valid and registered email and password in the login page and selects “Log in”	Successful login using native login account	Successful login using native login account	FR7 FR7.3
54	Create Task with Low Priority (Local)	User selects Create New Task” from any task view page, selects “Low Priority” from the priority drop down menu, and selects “Add New Task”	Successful creation of a task with the Low Priority tag	Successful creation of a task with the Low Priority tag	FR2 FR5
55	Create Task with Medium Priority (Local)	User selects Create New Task” from any task view page, selects “Medium Priority” from the priority drop down menu, and selects “Add New Task”	Successful creation of a task with the Medium Priority tag	Successful creation of a task with the Medium Priority tag	FR2 FR5
56	Create Task with High Priority (Local)	User selects Create New Task” from any task view page, selects “High Priority” from the priority drop down menu, and selects “Add New Task”	Successful creation of a task with the High Priority tag	Successful creation of a task with the High Priority tag	FR2 FR5

4.2 Performance Requirements

[See sections 4.0, 3.2 for specific directions about what outputs should be included here.]

ID	Description
N/A	The performance requirements are not applicable for this particular application. (Low Risk)

4.3 Usability Requirements

[See sections 4.0, 3.3 for specific directions about what outputs should be included here.]

ID	Description	Input	Expected Output	Actual Output
UR1	The user shall be able to access the system from a laptop or desktop PC.	User utilizes a desktop or laptop to access the application.	Users should access the browser and site url using a desktop or laptop.	Users can successfully access the system using a desktop or laptop.
UR1.1	The user shall be able to access the system through Firefox & Chrome.	User inputs the application url into Firefox & Chrome, and is able to access the login page.	Users should use Firefox or Chrome to visit the URL address and access the web application.	Users can successfully access the application from Chrome, MacOS, and Firefox.
UR2	The user shall be able to use the sort arrows to sort tasks by name, due date, and priority simultaneously.	User clicks on sorting icon next to task title or due date or priority to sort tasks by designated category	User should sort by task title, task date, and task priority	Users can sort tasks by all three categories: task title, due date, and priority.
UR2.2	The system shall maintain a simple and clean interface.	Users with a range of technology experience	Users should easily navigate the system with simple	Users of a diverse age range and technological

		should be able to navigate through the application.	navigation.	knowledge can navigate through the application design.
UR3	The system shall resize graphical elements of the user interface to fit any size display.	Users utilize different size displays (laptops or desktops) to access the application as graphical elements adjust according to device size.	Users should see a resized graphical display.	The system is resizing graphical elements to fit any size display.
UR4	The system shall depict neutral colors in the graphical elements of its user interface.	User accesses the application and is able to view all graphical design elements as intended.	Users should be able to view colors used in graphical elements of the user interface.	Users are able to view neutral colors used in graphical elements.
UR5	The system shall comply with WCAG accessibility guidelines as detailed in the Design Specification document. (See 3.6 - Design Constraints)	Users are able to access the application and view or interact with all content specified by the WCAG accessibility guidelines.	Users of different abilities should be able to see and engage with all content in a tab target order that follows accessibility guidelines and WCAG Constraints.	Users are able to see and engage with content that follows the WCAG accessibility guidelines.

4.4 Interface Requirements

[See sections 4.0, 3.4 for specific directions about what outputs should be included here.]

Test ID	Description	Input	Expected Output	Actual Output	Verified Requirement IDs

6	Login via GitHub (Windows)	User selects “Log in with GitHub” on login page	User is redirected to GitHub’s login interface	User is redirected to GitHub’s login interface	IR1 IR1.1
52	Registration (Local)	User selects “Sign Up” from the log in page.	User is provided a sign up screen with fields for email, password, and confirm password	User is provided a sign up screen with fields for email, password, and confirm password	IR1 IR1.2.1 IR1.3.2 IR1.3.2.1
53	Login (Local)	User inputs a valid and registered email and password in the login page and selects “Log in”	Users are presented a login screen with both Email and Password fields, as well as Github Log in	Users are presented a login screen with both Email and Password fields, as well as Github Log in	IR1 IR1.2 IR1.3.3
46	Checking Tasks	User navigates to any task view page using the menu panel and selects the check box next to a task in the center task view panel	The menu panel buttons successfully navigates to the chosen list and the tasks are displayed checkable in the center view	The menu panel buttons successfully navigates to the chosen list and the tasks are displayed checkable in the center view	IR2 IR3 IR3.1 IR3.2 IR3.3 IR4 IR4.1 IR4.2
47	Delete Newly Created Task	User navigates to any task view page using the menu panel and selects the delete button next to a task in the center task view panel	The task is successfully deleted and visible in the “Recently Deleted” section	The task is successfully deleted and visible in the “Recently Deleted” section	IR4.3

58	Task Sort by Name	User selects the sort button on the name column within any task list view	The tasks are sorted by name in ascending or descending order	The tasks are sorted by name in ascending or descending order	IR4.4 IR4.5
45	Creating a Task	User selects “Create New Task” button from any task list view	A task details menu opens on the right side of the window with fields for task name, description, project, due date, and priority	A task details menu opens on the right side of the window with fields for task name, description, project, due date, and priority	IR5 IR5.1 IR5.2 IR5.3

4.5 Logical Database Requirements

[See sections 4.0, 3.5 for specific directions about what outputs should be included here.]

Test ID	Description	Input	Expected Output	Actual Output	Verified Requirement IDs
45	Creating a Task	User selects “Create New Task,” inputs data in relevant fields and then selects “Add New Task”	A new task with the inputted information is inserted into the tasks table in the database	A new task with the inputted information is inserted into the tasks table in the database	DR1 DR1.1 DR2 DR2.1 DR3 DR3.1 DR3.2
52	Registration (Local)	User selects “Sign Up” from the log in page, provides a valid email and password, and selects “Sign Up”	The provided user information is stored in the user account data table in	The provided user information is stored in the user account data	DR1 DR1.1 DR2 DR2.1 DR5 DR5.1

		once more	the database	table in the database	DR5.2
7	Login via GitHub (Unix)	User selects “Log in with GitHub” on login page	The data sourced from GitHub is stored in the user data table	The data sourced from GitHub is stored in the user data table	DR1 DR1.1 DR2 DR2.1 DR5 DR5.3

4.6 Design Constraints

[See sections 4.0, 3.6 for specific directions about what outputs should be included here.]

4.7 Software System Attributes

[See sections 4.0, 3.7 for specific directions about what outputs should be included here.]

4.8 Supporting Information

[See sections 4.0, 3.8 for specific directions about what outputs should be included here.]

5. Appendix A – Tailoring Policies

5.1 Assumptions and dependencies

[Identify any and all factors that may impact the implementation and execution of the requirements written below. These factors do not add a constraint but may impact development if they are changed. Example: a major update to an operating system(OS) on which the SUD is intended to run impacts the implementation of one of the core features. The version of the OS that the system had intended to run on should be listed in this section.]

5.2 Acronyms and Abbreviations

Term	Definition	Abbreviation	Source (if applicable)
Example: context of use	*users, tasks, equipment (hardware, software and materials), and the physical*	*COU*	*[SOURCE: ISO/IEC 25000:2014, 4.2]*

	and social environments in which a product is used		
System Under Development	The system which is actively being developed.	SUD	

5.3 Tailoring Policies

Tailoring is not a requirement to bring the document into compliance with the standards set by IEEE 29148-2018. Tailoring should only occur when conformance to the standard is not possible or practical. The act of tailoring is the modification and/or removal of one of the content sections outlined in this document, adding additional information items for organization is not considered tailoring. Tailoring should only occur when factors or circumstances:

- surround an organization that is using the document
- influence a project using this document to meet an agreement
- reflect the needs of an organization.

When tailoring the document, the following activities shall be implemented:

- Identify and document the circumstances that may influence tailoring.
  - novelty, size and complexity
  - stability of operating environments
  - variety in operating environments
  - starting date and duration
  - emerging technology
  - availability of services of enabling systems
  - other standards with which the document needs to conform.
- Identify and get input from all parties impacted by the tailoring process.
  - Such as stakeholders, contributors, and other interested parties
- Delete the information contents that require tailoring.

6. Appendix B – Copyright

This document is based on a template meeting the ISO/IEC/IEEE 29148-2018 standard, available at https://www.iso.org/standard/72089.html. Template authors are:

Dr. rer. nat. Bastian Tenbergen,
Associate Professor of Software Engineering
bastian@tenbergen.org

Mikayla Conner-Spagnola, MA
Independent Consultant

mconner@oswego.edu

Department of Computer Science
State University of New York at Oswego
Oswego, NY 13126, United States