# CSC 431 CheckMate Software Requirements Specification (SRS)

|  |  |
| --- | --- |
| Sergio Velikopoljski |  |
| Eugene Li |  |
| Nguyen Nguyen |  |
| Jason Betz |  |
| Carl Liu |  |
| Anmol Jena |  |
| Czavier Tan |  |
| Alejandra Zavala |  |
| Oliva Raymond |  |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1.0 | 2/10/20 | All | Preliminary draft (title and description) |
| 2.0 | 2/26/20 | All | More details |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[1.](#_heading=h.3dy6vkm) System Requirements 6

[1.1](#_heading=h.1t3h5sf) Functional Requirements 6

[1.1.1](#_heading=h.2grqrue) Requirement Title 6

[1.2](#_heading=h.2s8eyo1) Non-Functional Requirements 6

[1.2.1](#_heading=h.17dp8vu) Requirement Title 6

[2.](#_heading=h.3rdcrjn) System Constraints 7

[2.1](#_heading=h.26in1rg) Tool Constraints 7

[2.1.1](#_heading=h.lnxbz9) Requirement Title 7

[2.2](#_heading=h.35nkun2) Language Constraints 7

[2.2.1](#_heading=h.1ksv4uv) Requirement Title 7

[2.3](#_heading=h.44sinio) Platform Constraints 7

[2.3.1](#_heading=h.2jxsxqh) Requirement Title 7

[2.4](#_heading=h.z337ya) Hardware Constraints 7

[2.4.1](#_heading=h.3j2qqm3) Requirement Title 7

[2.5](#_heading=h.1y810tw) Network Constraints 7

[2.5.1](#_heading=h.4i7ojhp) Requirement Title 8

[2.6](#_heading=h.2xcytpi) Deployment Constraints 8

[2.6.1](#_heading=h.1ci93xb) Requirement Title 8

[2.7](#_heading=h.3whwml4) Transition & Support Constraints 8

[2.7.1](#_heading=h.2bn6wsx) Requirement Title 8

[2.8](#_heading=h.qsh70q) Budget & Schedule Constraints 8

[2.8.1](#_heading=h.3as4poj) Requirement Title 8

[2.9](#_heading=h.1pxezwc) Miscellaneous Constraints 8

[2.9.1](#_heading=h.49x2ik5) Requirement Title 8

[3.](#_heading=h.2p2csry) Requirements Modeling 10

[3.1.1](#_heading=h.147n2zr) Requirement Title 10

[4.](#_heading=h.3o7alnk) Evolutionary Requirements 11

[4.1](#_heading=h.23ckvvd) Functional Requirements 11

[4.1.1](#_heading=h.ihv636) Requirement Title 11

[4.2](#_heading=h.32hioqz) Non-Functional Requirements 11

[4.2.1](#_heading=h.1hmsyys) Requirement Title 11

# 

# 

# 

# 

# Table of Tables

<Generate table here>

**Table of Figures**

<Generate table here>

### System Requirements

### 1.1. Functional Requirements

*< List all functional requirements in the following example format >*

|  |  |
| --- | --- |
| ID | FR1 |
| Title | Login Screen / Registration |
| Description | Login menu for users so they can be validated and for access to personal accounts. |
| Priority | 0 |
| Precondition(s) | User clicks on the login option, prompting a modal or new page to open asking for username and password. Bring them to registration if they do not have an account yet. |
| Basic Flow | Once the user clicks on the login option, they enter their username and password for their account login. If they have forgotten their password, they can click on “forgot password” and will be asked for their email for confirmation. They then can set a new password and access their profile. |
| Postcondition(s) | The information is sent to the database to confirm that the username/email exists and the encrypted password matches. They then get access to their account. |
| Use Case Diagram | CheckMate Use Case Diagram |

|  |  |
| --- | --- |
| ID | FR2 |
| Title | Profile Page |
| Description | Page that displays current users profile. Contains content relevant to the users “mate” along with basic account information such as username, avatar, etc. |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 0 |
| Precondition(s) | The user logs into their account with username/email and password and is then taken to their profile page. |
| Basic Flow | The user is taken to their profile page and has access to their tasks and progress. They can check on their mate’s life and new tasks that must be completed in the given day. They also have options to edit basic information including the style of the mate and their username handle. |
| Postconditions(s) | The user interacts with the profile, if they choose to change information it is confirmed and sent to the database for storage. If they choose to complete a task, the mate responds in a positive way, promoting positive reinforcement to completing tasks. |
| Use Case Diagram | CheckMate Use Case Diagram |

|  |  |
| --- | --- |
| ID | FR3 |
| Title | To-do List |
| Description | Menu for users to add tasks and display current tasks. |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 0 |
| Precondition(s) | The user selects “Tasks List” on the navigation bar or from their own profile. |
| Basic Flow | On this page, the user has the ability to check off any new tasks they have completed in the given day. If something new has occurred in their life, they have the option of creating a new task with title, description, importance, and due date. They can also scroll through popular tasks and add them to their list like “walk in the park” or “meditation”. Users can also create different task lists, star one as the default task list, rename lists, delete lists, and set different priorities for tasks. |
| Postconditions(s) | The user checks off a task or creates a new task, adding it to their database and updating the status of their mate and feed. |
| Use Case Diagram | CheckMate Use Case Diagram |

|  |  |
| --- | --- |
| ID | FR4 |
| Title | Mates |
| Description | Interactive pixel avatar that is customizable. Status of the Mate is dependent on and proportional to the amount of completed tasks. |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 0 |
| Precondition(s) | The user must create an account and input tasks in order to have access to their mate. |
| Basic Flow | Once the user creates their account and tasks, they have the opportunity to style their mate. The mate can be human, animal, etc. and wear different props. Their mate is reliant on the completion of tasks, so a healthy mate will be achieved when tasks are completed daily. Different props can be added whenever the user wants, and customization could have a currency that is earned through continuous interaction with the mate. The mate page will show the current health of the mate, what tasks must be completed to help it live, and how long it has been alive for. There should be a specific mate for each task list. |
| Postconditions(s) | The mate’s life and appearance is stored in the database, and this status can be shared via social media. |
| Use Case Diagram | Mate Use Case Diagram |

|  |  |
| --- | --- |
| ID | FR5 |
| Title | Notification |
| Description | The app will notify users of any updates to the status of their Mate via email. It will also notify them on the status of any other mates that they might be following through the social aspect of the app. |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 5 |
| Precondition(s) | The user must make an account and include their email for the notifications to be active. |
| Basic Flow | If the status of the mate changes drastically, or if any of the user’s followers send a message or interact with the mate, an email will be sent summarizing the event with a link to the profile page. The user can turn off notifications as well. On the web app, notifications will appear on the navigation bar. |
| Postconditions(s) | Email is sent to the user’s email giving a summary of the event, and the notification is also registered in the database for showcase once the user logs in again. |
| Use Case Diagram | CheckMate Use Case Diagram |

|  |  |
| --- | --- |
| ID | FR6 |
| Title | Feed Page |
| Description | Page used by users to share content and view content of other users (think facebook feed, insta feed, or reddit feed page). |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 2 |
| Precondition(s) | The user selects “feed” from the navigation bar and is taken to the Feed Page. |
| Basic Flow | The user enters the feed page and can see all progress and activity from the users they follow. They can scroll through the feed and find a mate or user that they want to interact with. They can comment on tasks that were completed or give a “like” to a mate that is doing well. Clicking on the username or mate will also give more information about the data they selected. |
| Postconditions(s) | The user scrolls through the feed and interacts with other users, saving the interaction in the database and displaying likes or comments on the other users’ activity. |
| Use Case Diagram | Feed Page Use Case Diagram |

|  |  |
| --- | --- |
| ID | FR7 |
| Title | Social Media Sharing |
| Description | Button that allows user to share the status of their mates on social media sites (like Facebook or Twitter). |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 1 |
| Precondition(s) | The user clicks the share button and logins in to their social media. |
| Basic Flow | CheckMate uses the social media API to generate a post sharing the status of their mate and inviting their friends to join the application. |
| Postconditions(s) | Other users can join via the social media post. Original user is returned to their main page. |
| Use Case Diagram | Feed Page Use Case Diagram |

|  |  |
| --- | --- |
| ID | FR8 |
| Title | Mate Map |
| Description | Overlooking map of all the mates (each one representing a task list), so user can monitor their overall progress. |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 1 |
| Precondition(s) | The user switches to the map button, the screen “zooming out” to an island of all their mates. |
| Basic Flow | All mates can be seen chilling on some background. The user can select a mate to go to that specific task list. There may or may not be a graveyard of past mates. Other statistics can be included (total time spent, number of total tasks completed, etc.). |
| Postconditions(s) | User can admire their progress, query more information about their mates, or return to their task list page. |

#### Non-Functional Requirements

*< List all non-functional requirements in the following example format >*

|  |  |
| --- | --- |
| ID | NF1 |
| Title | Web and mobile framework |
| Description | The app will be built using React, which is a multi-platform framework that allows for one codebase to be used to deploy the app to the web and iOS and Android. |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 0 |
| Applicable FR(s) | FR1, FR2, FR3, FR4, FR5, FR6 |

|  |  |
| --- | --- |
| ID | NF2 |
| Title | Non-relational database |
| Description | The application will use the MongoDB database to store the users information and data. Data such as: username, passwords, mates, mate customizations, etc. |
| Source Scenario | <Code for associated scenario in SCD> |
| Priority | 0 |
| Applicable FR(s) | FR1, FR2, FR3, FR4, FR5, FR6 |

|  |  |
| --- | --- |
| ID | NF3 |
| Title | Animation framework |
| Description | The application will use Aseprite to create animations and pixel art that will be displayed throughout the application. |
| Source Scenario |  |
| Priority | 0 |
| Applicable FR(s) | FR5 |

### System Constraints

#### 3.1. Tool Constraints

*< List all tool constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 3.2. Language Constraints

*<List all language constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 3.3. Platform Constraints

*< List all platform constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 

#### 3.4. Hardware Constraints

*<List all hardware constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 3.5. Network Constraints

*<List all network constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 3.6. Deployment Constraints

*<List all deployment constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 3.7. Transition & Support Constraints

*<List all transition & support constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 3.8. Budget & Schedule Constraints

*<List all budget & schedule constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### 3.9. Miscellaneous Constraints

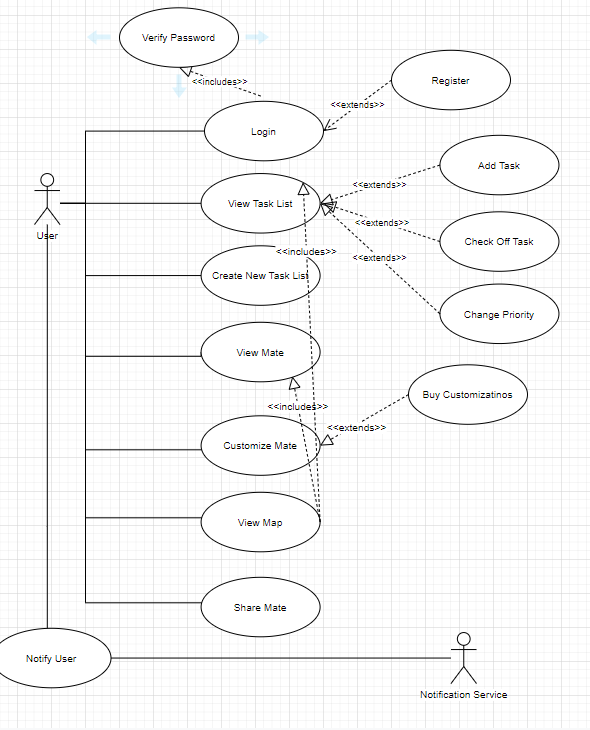
*<List all miscellaneous constraints in the following example format>*

|  |  |
| --- | --- |
| ID | <a unique ID for the constraint, e.g., C1)> |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

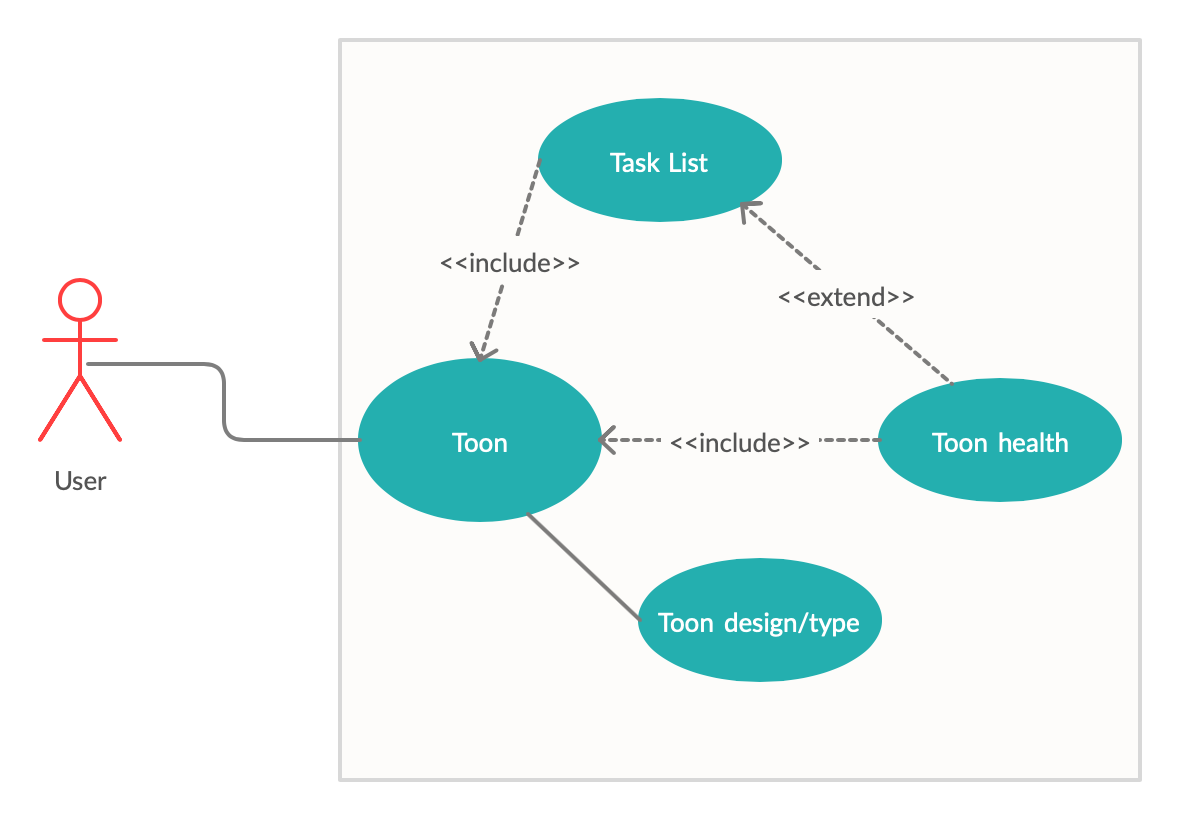
### Use Case Modeling

*< List all Use-case diagrams for the functional requirements in the following format>*

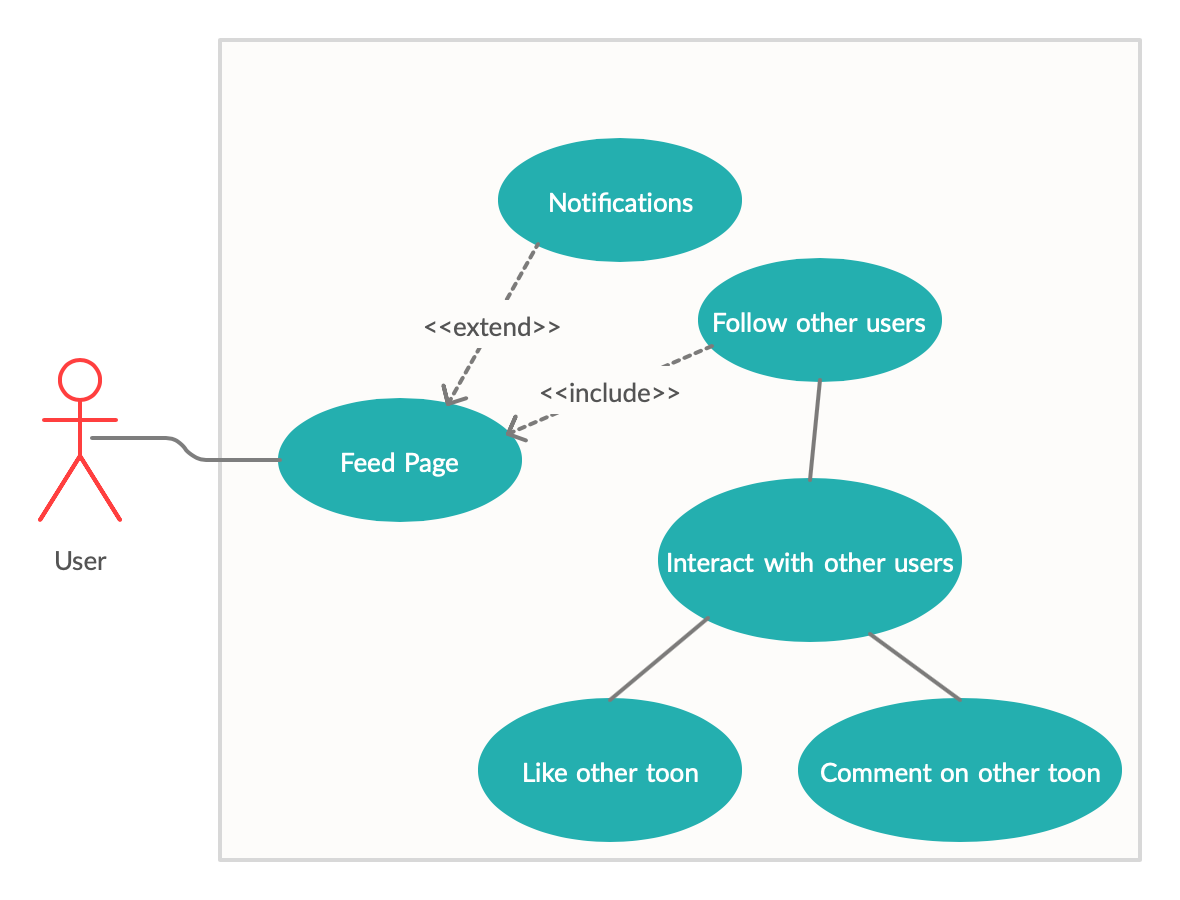
|  |  |
| --- | --- |
| Title | CheckMate Use Case Diagram |
| Description | This diagram is the complete interaction of the web app CheckMate. It includes all functional requirements and displays the flow of the user and options they can take when they log on. |
| Priority | 0 |
| Precondition(s) | The user needs to make an account. |
| Postconditions(s) | The user finishes interacting with their mate and followers, and signs off for the day. |
| Use Case Diagram | 1 |



|  |  |
| --- | --- |
| Title | Mate Use Case Diagram |
| Description | This diagram illustrates the purpose of the Mate in relation to the task list. The user creates their mate along with their task list, and can customize the type of mate. The mate health is dependent on the task list and completion of tasks. |
| Priority | 0 |
| Precondition(s) | The user needs to make a task list and create a mate |
| Postconditions(s) | As the user completes tasks the mates health is affected |
| Use Case Diagram | 2 |



|  |  |
| --- | --- |
| Title | Feed Page Use Case Diagram |
| Description | This diagram illustrates how users can interact with other users’ mates via following, commenting on, and liking their mates. |
| Priority | 4 |
| Precondition(s) | The user needs an established account and mate to share |
| Postconditions(s) | The user receives interaction such as likes and comments on their own page |
| Use Case Diagram | 3 |



### Evolutionary Requirements

#### 5.1. Functional Requirements

*< List all functional requirements in the following example format >*

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |
| Precondition(s) | <What needs to happen before> |
| Postconditions(s) | <What happens as a result> |
| Use Case Diagram | <Link or number, if present> |

#### 5.2. Non-Functional Requirements

*< List all non-functional requirements in the following example format >*

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |
| Applicable FR(s) | <Which functional requirement(s) is this applicable to?> |