Software Requirements Specification

for

Task Management Application

Version 1.0 approved

Prepared by Group No. 4

Section No. 5217 (Class 3)

14 October, 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Initial Release	Oct, 2021	N/A	1.0
N/A	N/A	N/A	N/A

1. Introduction

1.1 Purpose

This document consists of software requirements of a Task Management System that will help to create, delete, organize and distribute tasks effectively. This product can be used by those individuals who want to manage tasks effectively, ranging from project managers to students. The scope of the project includes task creation and maintenance that can be used by individuals or as a team to keep a check on daily tasks or professional projects. Furthermore, this SRS will provide insight into each component of the product.

1.2 Document Conventions

This document has been written in font 'Times' with the font size 12. Each section titles and subsections are bold and given specific numbers. Requirements are given in bullet points whereas the description is given in paragraphs.

1.3 Intended Audience and Reading Suggestions

The document will be read and reviewed by the development team, testers, documentation writers, and stakeholders to have a better idea of this product and its requirements. This SRS will contain the introduction followed by an overall detailed description, interface requirements along with system features and non-functional requirements of this product. The SRS has been organized approximately in order of increasing specificity.

The system features section is more pertinent to testers, as they need to have a clear understanding of the system features to test the software and provide meaningful feedback to the developers. Interface requirements, functional and non-functional requirements are targeted towards the development team so that they can create a user-friendly interface, and follow optimized techniques to develop the components of this product.

1.4 Product Scope

The Task Management System will provide an organized platform for the user to keep a track of all kinds of daily tasks as well as to manage the professional projects effectively. This product can be used by any user and isn't targeted to any specific audience. The features of this product will minimize the efforts of the user by creating a systemized way to define any processes or tasks. The greater emphasis on the user interface by the development team will create an easy-to-use application, this will not only reduce the complexities in any form but will also enable any application user to experience this product freely without any biases of ages. Along with its major objectives, the product can provide an authority to the user to keep an account of the group members working on the same tasks which can be an effective way to maximize productivity. The feature of dividing tasks into subtasks will help the users to manage time effectively.

Furthermore, the user will have the ability to manage several tasks easily. To ensure the privacy of the user, an authentication system would also be added to prevent any unauthorized access. Professional institutes and corporations can also utilize this platform to keep track of their project timelines. It will help them to increase productivity due to the teamwork and the involvement of various departments.

1.5 References

1. Reference SRS Document: https://cs.gmu.edu/~dfleck/classes/cs421/spring08/SampleProject/FINAL%20SRS.pdf
2. SRS Document Guide:
3. https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

2. Overall Description

2.1 Product Perspective

The task management application is aimed towards those users who want to maintain and monitor a list of their to-do tasks. It provides them with a platform to create, edit and delete tasks they need to complete. For users working in teams, it allows them to create sub-tasks and distribute these tasks among their colleagues/workers and monitor how much of each task has been completed.
The application is intended to be an enclosed source software, supporting a range of devices and operating systems. It will follow the client-server model, with each user device being a client who requests data from a server. The server will be an Oracle 19c database running on a Windows 10 desktop environment for demonstration purposes. The application software will be closely linked to the server at all times to ensure that the correct information is displayed and the correct changes are made to the database anytime a user updates any information on their respective device. For this reason the application will need to be connected to the internet at all times to ensure the correctness and consistency of data.

2.2 Product Functions

Our Task Management Applications contains the following key features:

1. Allows the creation of a distinct user account and provides two-factor authentication to ensure no accidental login occurs
2. Allows a user to view and edit their account details and change them if needed (excluding their email address)
3. Allows the user to create a main task and then create sub-tasks within the main task. The user can then assign each of these tasks and sub-tasks to different users by adding them using their email address
4. Allows the task creator to manage editors and viewers of a distributed task through their email address
5. Monitors the completion of each parent task in the users to-do list through a progress bar
6. Automatically moves a task to the completed list once the root task has been flagged as completed
7. Allows the user to change the font, size and color of any task they have created
8. Allows the user to color code different tasks for ease of navigation.

2.3 User Classes and Characteristics

The following users are anticipated to use this product:

1. **Office Workers:**
   Project managers assigned to overlook a team working on any sort of project will use this application to ensure smooth management of all pending tasks. They will be able to monitor who is assigned to what task as well as monitor the completion of each parent task separately. These users will be professionals of their industry and will know how to properly manage a team. The user interface will be simple enough for them to map their everyday management practices onto our application.
2. **Personal Users:**
   Users who want to keep track of their everyday to-do list will use this application to list down their various pending tasks and then complete them in whichever order they desire. These users will range from students to parents or even highly skilled professionals. In order to allow almost anyone to easily make use of our application, the user interface will be simple and easy to navigate through. These people will rarely ever need to create sub-tasks or distribute tasks among other people.

2.4 Operating Environment

The task management application will be operating on both a desktop environment and a mobile phone by multiple different users in multiple different locations. The operating systems that it will be running on initially will be Windows 10 for the desktop and Android 11.0 on mobile devices. After the initial full release of the software operating systems such as Linux, MacOS X and iOS will be taken into consideration to extend the software onto. In order to use the software, each device connected will need to have access to the internet and thus the database server. For demo purposes the database being used will be Oracle 19c, which will be running on a Windows 10 operating system acting as a server for the application.
To run and operate the task management software basic input devices such as a keyboard, a mouse, a monitor or a touch screen are necessary.

2.5 Design and Implementation Constraints

1. **Technical Skillset** - The developers of this software are undergraduate students hence have a limited amount of technical know-how. These students are not very well-versed in any one particular programming language and hence will encounter several difficulties while developing the software
2. **Database Environment** - The system will be connected to a relational database system (Oracle 19c) which has its limitations in the free version. This might cause some difficulty when programming some specific functionalities in the system.
3. **Budget** - Due to budget limitations, the developers may not be able to buy pre-built components to implement in their software. The database being used will also be the free version and due to budget constraints the paid version cannot be bought.

4. **Time** - The deadline to complete this project means that the developers have to ensure that each functionality is implemented within the time constraint which can lead to some code being implemented less efficiently.
5. **Hardware Limitations** - Because there is a wide variety of mobile and desktop device configurations, the developers have to ensure that the software makes use of as little RAM and storage. They have to make sure the software can be built to adapt to any display size and pixel density.
6. **Internet Constraints** - Because the connection speeds can vary from place to place and time to time, the software must be built in a way that it ensures no data is lost or corrupted in the case of a sudden internet outage.

2.6 User Documentation

This product will include a simple user manual. It will include a product overview, a complete configuration of the database software (Oracle 19c), technical details and contact information of the developers. There will not be any online help for this software and neither will there be any tutorials.

2.7 Assumptions and Dependencies

The list of assumptions are:

1. There should be no errors in the coding
2. The user-interface should be simple yet detailed so that a range of technically skilled users can adapt to it easily
3. The system should provide fast access to the database
4. The system should be secure and prevent accidental logins
5. The system should prevent data corruption due to internet outage
6. The database should be up and running twenty four hours a day
7. Each user should have a unique email ID associated with their account
8. The users should be able to access their account securely from any device that has an internet connection

The list of dependencies are:

- The users should have the knowledge to run and operate a simple desktop/mobile application
- The information of all users should be stored in a database which is accessible to the application at all times
- The users should have internet access on their device to ensure connectivity to the database
- The application should be running on the minimum specified hardware or anything above

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

4.1 Account Management

4.1.1 Description and Priority

Account management is the feature which enables a user to create an account. It allows the user to log-in to an existing account or log-out of a currently logged-in account. It gives the user the option to create a new password in case they have forgotten their current one. It provides the user with an additional layer of security known as the two-factor authentication. This is a high priority feature as an account is required to use the application.

4.1.2 Stimulus/Response Sequences

Sign-up Case: When a user runs the application the first screen prompts two buttons. One button leads to the sign-up form and the other leads to a log-in form. If the user taps the sign-up button, they are led to a new screen with a sign-up form. This form will contain a few fields to gather the users information, such as their email and their password along with other information. The system checks the application database to ensure an account with the same email doesn't already exist. If it does, the system shows an error message. If the details are valid, it stores the newly given information in the database table and directs the user to the login page.

Log-in Case: When a user runs the applications the first screen prompts two buttons. One button leads to the sign-up form and the other leads to a log-in form. When the user taps the log-in button, they are led to a new screen with a log-in form. This form contains two fields, the email and the password field. Once the user presses the enter button, the email and the password are checked from the database to ensure that the correct password has been entered. If either the email or the password is incorrect, the user is prompted with an error message and a button to change their password if need be. If the email and password are authentic the user is directed to a new screen where they are required to enter a one time password which is sent to the user at their email. If the OTP that the user has entered is correct, they are directed to the application home page.

Forgot-Password Case: In the log-in page, the user is shown a button underneath the log-in form, which displays the text "Forget Password?". If the user clicks this button they are directed to a screen which asks them to enter their email. If the email exists in the database, an OTP is sent to the given email and the user is directed to a new page which asks them to enter the OTP. If the OTP is verified, the user is then directed to a page where they are required to enter their new password. If the password matches the required criteria, the database is updated with the new password. The user is then redirected to the log-in form.

Log-out Case: On any given page the user has a menu bar available to them at all times. If the user wishes to end their session and log-out of their account on one particular device, they press the button which shows a drop down menu. The last element in the menu is a red button with the text “Log-out”. If the user presses this button they are shown a pop-up which asks them to confirm if they would like to log-out. If they select the confirm button, their session ends and the user is directed to the start-up page with the log-in and sign-up buttons.

Edit Details Case: On any given page the user will have a drop down menu available to them. In this menu will be a button to view and edit their account details. When a user clicks this button they will be directed to a screen which displays personal details the user entered while creating their account. The screen will have two buttons, one to edit their information and the second to save their edited information. If the user clicks the edit information button they will be redirected to a form in which all fields other than their email can be modified. Once the user clicks the save button, the system will check to ensure that the given information matches the required criteria. If it does not, an appropriate error message will be displayed. If it does, the system will update the information in the required database table and prompt the user to a successful change.

4.1.3 Functional Requirements

1. Log-in and sign-up buttons shall be shown on the startup page
2. The system shall display a form based when sign-up is pressed
3. The system shall ensure the email is not already taken at the time of sign-up
4. The system shall verify that all fields match the required criteria
5. The system shall display a form when log-in is pressed
6. The system shall ensure that the password matches the email in the database
7. The system shall prompt the user to enter an OTP sent via their email
8. The system shall verify that the OTP entered matches the OTP generated
9. The system shall display a forgot password button on the log-in form
10. The system shall show a form when this button is pressed
11. The system shall verify that the email exists in the database
12. The user shall be prompted to enter an OTP sent via their email
13. The system shall verify that the OTP matches the OTP generated
14. The system shall display a change password form
15. The system shall verify that the new password matches the given criteria and shall then update the database accordingly
16. The user shall have a log-out button available to them on all pages
17. The system shall ensure the users work is saved to the database before ending the session
18. The system shall ask the user to confirm if they want to log-out
19. The user shall have a view/edit button available to them on all pages
20. The system shall show all user details selected from the database
21. An edit button shall be displayed underneath the information
22. The system shall show a form to change all fields excluding the email
23. A save button shall be displayed underneath the form

24. The system shall verify that all fields match the given criteria and then update them in the database
25. Appropriate error messages should be displayed in case of any error

4.1.4 Use Case Tables

Use Case Name	**Sign-up**
Related requirements	1, 2, 3, 4, 25
Goal in context	A new account will be created
Preconditions	1. Email must be in the correct format 2. All fields must meet the criteria 3. The email must not already exist
Successful End Condition	Correct information is input and the user details are saved in the database. The log-in page is returned.
Failed End Condition	Account creation fails for various reasons and an error message is output
Primary Actors	User
Secondary Actors	Null
Triggers	Sign-up button is pressed
Included Cases	Null
Main Flow	1. User presses sign-up button and an appropriate form is shown 2. User entered details 3. System verifies details and stores them in the database 4. New account is created 5. Log-in page is returned

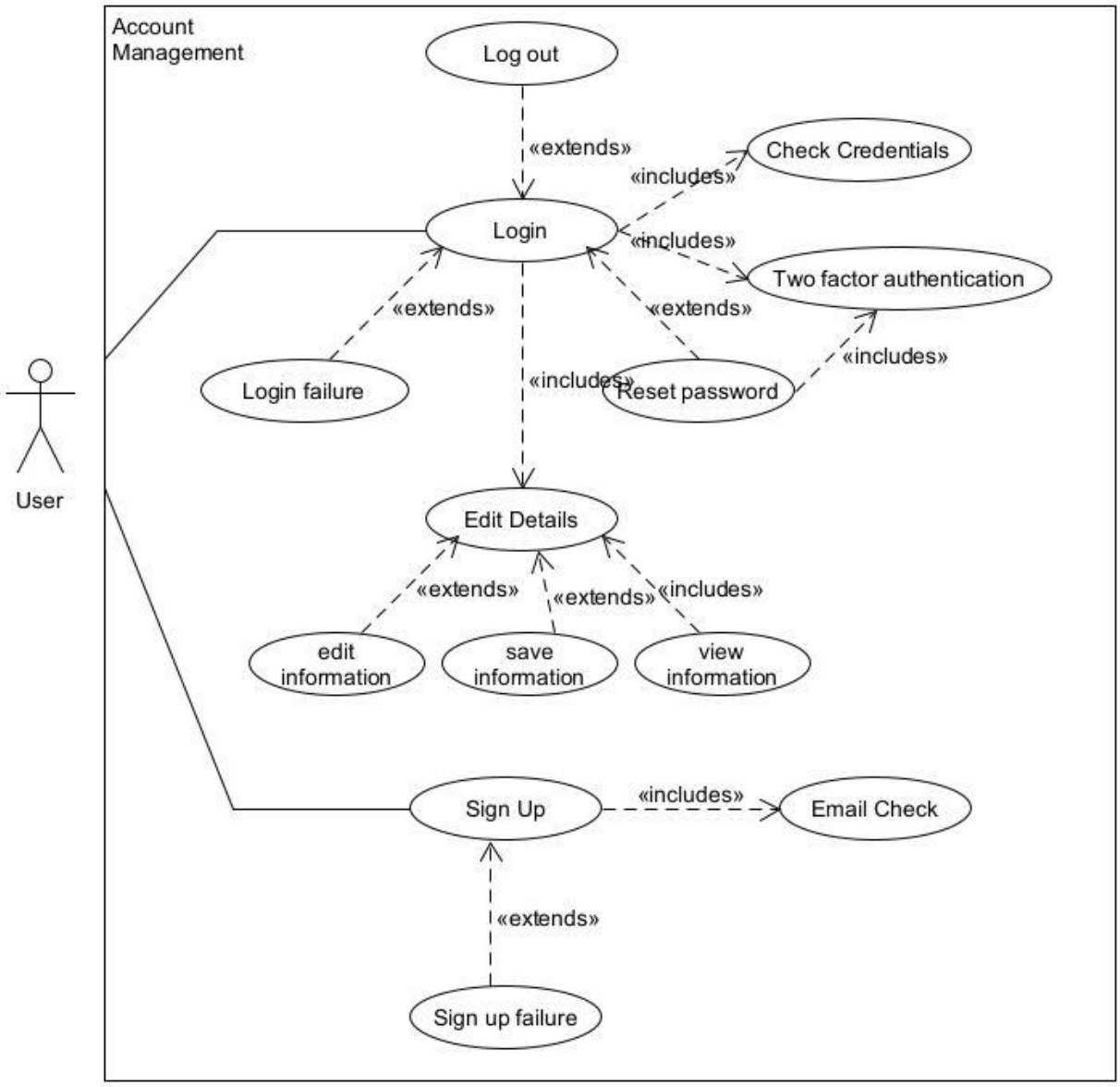
Use Case Name	**Log-in**
Related requirements	5, 6, 7, 8, 25
Goal in context	An existing user will be allowed to access their data
Preconditions	1. User must exist 2. User must have access to their email
Successful End Condition	User logs in and the home page is returned
Failed End Condition	Log-in attempt is failed and an error message is returned
Primary Actors	User
Secondary Actors	Authorization database
Triggers	Log-in button is pressed
Included Cases	Verify Credentials, Two-Factor Authentication
Main Flow	1. User presses the log-in button and a form is returned 2. User enter their credentials 3. Credentials are verified from the database 4. An OTP is sent via email and entered in the appropriate text box and then verified by the system 5. User is logged in and directed to their home page

Use Case Name	**Forgot Password**
Related requirements	9, 10, 11, 12, 13, 14, 15, 25
Goal in context	Password is changed on the user's request
Preconditions	1. The user account must exist 2. User must have access to their email
Successful End Condition	The password is accepted and the user is redirected to the log-in page
Failed End Condition	The password change fails and an error message is prompted
Primary Actors	User
Secondary Actors	Null
Triggers	Forgot Password button is pressed
Included Cases	Two-Factor Authentication
Main Flow	1. User presses the forget password button and a form is displayed 2. User enters their email address 3. Email address is verified 4. An OTP is sent via email and entered in the appropriate text box and then verified by the system 5. Change password form is displayed 6. The user enters the new password which is then verified 7. The password is changes the user is redirected to the log-in page

Use Case Name	**Log-out**
Related requirements	16, 17, 18. 25
Goal in context	A logged in user is able to log-out
Preconditions	1. User must be logged in
Successful End Condition	The user is logged out and the startup page is returned
Failed End Condition	The user is unable to log-out
Primary Actors	User
Secondary Actors	Null
Triggers	Log-out button is pressed
Included Cases	Null
Main Flow	1. A logged in user presses the log-out button 2. All changes are saved in the database 3. The user is asked to confirm if they want to log-out 4. The user is logged out and the startup page is returned

Use Case Name	**Edit Details**
Related requirements	19, 20, 21, 22, 23, 24, 25
Goal in context	A logged in user is able to change their account details
Preconditions	1. User must be logged in
Successful End Condition	The user details are changed and the home page is returned
Failed End Condition	The user details are not updated and an error message is displayed
Primary Actors	User
Secondary Actors	Null
Triggers	Edit Details button is pressed
Included Cases	Null
Main Flow	1. A logged in user presses the edit details button 2. A form is displayed with editable values 3. The user enters the new information 4. The user presses the save button and the details are verified 5. The details are saved and the user is directed to the home page

4.1.5 Use Case Diagram



4.2 Main Task Management

4.2.1 Description and Priority

Task creation is the feature which enables a user to create a task. It allows the user to maintain a list of tasks. This feature allows the user to create subtasks for any given task as well. Each parent task has a completion bar which helps the user monitor the overall progress. Each task and subtask has a checkbox next to it indicating its completion. Completed tasks are automatically moved to a completed list and can be manually moved back to the to-do task list. This is a high priority feature as without task management the application has no functionality.

4.2.2 Stimulus/Response Sequences

Navigate Lists Case: After a user has logged into their account they are directed to the home page. This page displays a small welcome message along with two buttons leading to two separate lists of tasks. The To-Do button leads to a new screen on which the user is shown a list of pending tasks and sub-tasks. The Completed button leads the user to a list of tasks which have been marked as completed. Each screen has its own functions and features as discussed below

Add Task Case: With-in the To-Do list, the system displays a large "+" symbol indicating that if this button is pressed it will allow the user to create a new task. Once the button is pressed the user is directed to a form in which a user can add the task heading, task details, set the task deadline and delete the task. The information is then stored in the appropriate database table and the task is given a unique ID and a reference to the user who created the task.

Delete Task Case: Within the To-Do list, the system displays a delete button. Once this delete button is pressed the display is adjusted to allow the user to select from the list of tasks that are visible to them. After selection, the user presses the Confirm Delete button. A prompt is shown to the user, warning them about the action they are about to take and asking them to confirm if they would like to continue. Once confirmation is received, the task along with its information is removed from the database table and the To-Do list.

View Task Case: A user will have the option to click on any of the tasks within their task list. When a task is clicked the display will update to show the details of that specific task. A completion bar will also be visible if the particular task contains subtasks within it. There will be three buttons available within each task page. The edit information button will allow the user to change the information they entered while creating the task or color code their task based on priority. The assign people button will allow the task creator to invite people to view the task as well as assign the task to any number of people they have invited. The add subtask button will allow the task creator to add subtasks with their own title, description and date of completion. These subtasks can also be color coded and assigned to different users.

Edit Task Case: When a user clicks the edit task details button in the task view page the system should display a form with all the information that was entered when the task was created. The fields in this form should be editable. A user can change any information they want and then press the save button. Once the save button is pressed the information is checked to see if it matches the required criteria and then updated in the appropriate database table. The user is then directed back to the task view page with the update task information

Complete Task Case: Each task within the To-Do list will be led by a checkbox to mark its completion. Once a task has been completed the user simply needs to click on the check box. It task will be flagged as completed in the database and will be moved to the completed list in the user's account. If a user wishes to revert this action, they must go to the completed task list and uncheck the checkbox. The flag will be removed from the completed list and be added back to the To-Do list with all its information.

4.2.3 Functional Requirements

1. A welcome message with the user's name must be shown on the home page
2. Two buttons shall be shown on the homepage(To-Do List and Completed List)
3. To-Do list shall contain a list of tasks the user has created and are pending
4. Completed list shall contain a list of tasks the user has created and are completed
5. To-Do list shall have an Add Task button
6. The system shall display an add task form containing fields for the Title, description and due date of the task
7. The text fields should allow the user to change the font style of the text
8. The system shall verify that the data meets the required criteria and should add the new task to the appropriate database table with a reference to the user who created the task, and the To-Do list
9. The default color code for any task upon creation should be transparent
10. The To-Do list shall have a Delete Task button
11. Once clicked the user should have the option to select any number of tasks from the given list
12. All selected tasks should be highlighted to indicate their selection
13. A Confirm Deletion button should be available
14. Once pressed a warning prompt should be shown to the user asking for confirmation to continue
15. Once the confirmation is pressed the selected tasks shall be removed from the database table and from the To-Do list
16. Each task shall be clickable
17. Once a task is clicked the system shall return a task view page
18. The task view page shall have the task details as well as any subtasks created within the that specific task
19. The task view page shall have a button to edit the task information
20. The system shall display a form to update the task information

21. The information shall be verified and saved to the database once the save changes button is pressed
22. The task view page shall have the option to color code the task
23. Once clicked a popup should show with a set number of color swatches to choose from.
24. The chosen color should be shown as a colored square on the right hand side of the task in the main task list
25. The task view page shall have a button to create subtasks (explained in 4.3)
26. The task view page shall have a button to assign people (explained in 4.4)
27. The task view page shall show a completion bar only if subtasks exist
28. The completion percentage shall be automatically computed based on the number of completed subtasks within the task
29. The To-Do list shall have a checkbox before each task
30. Once the checkbox is clicked the task shall be flagged as completed in the database
31. A completed task shall be moved to the Completed list
32. A completed task shall be able to be moved back to the To-Do list by unchecking the checkbox in the Completed list
33. Appropriate error messages should be displayed in case of any error

4.2.4 Use Case Tables

Use Case Name	Add Task
Related requirements	5, 6, 7, 8, 9, 33
Goal in context	To be able to add a task in the To-Do list
Preconditions	The user must be logged in.
Successful End Condition	The user can add tasks successfully which can be viewed in the To-Do list.
Failed End Condition	The criteria is not fulfilled or the task has not been added to the database completely. In both conditions, the an error message will be displayed
Primary Actors	User
Secondary Actors	Null
Triggers	Add Task button is pressed
Included Cases	Null
Main Flow	1. The add task button is pressed and the user is directed to a form 2. The user enters the required information 3. The default color code is set to transparent 4. After the criteria is met, the user presses the add button and the data is added to the database. 5. The user is then directed back to the main task list

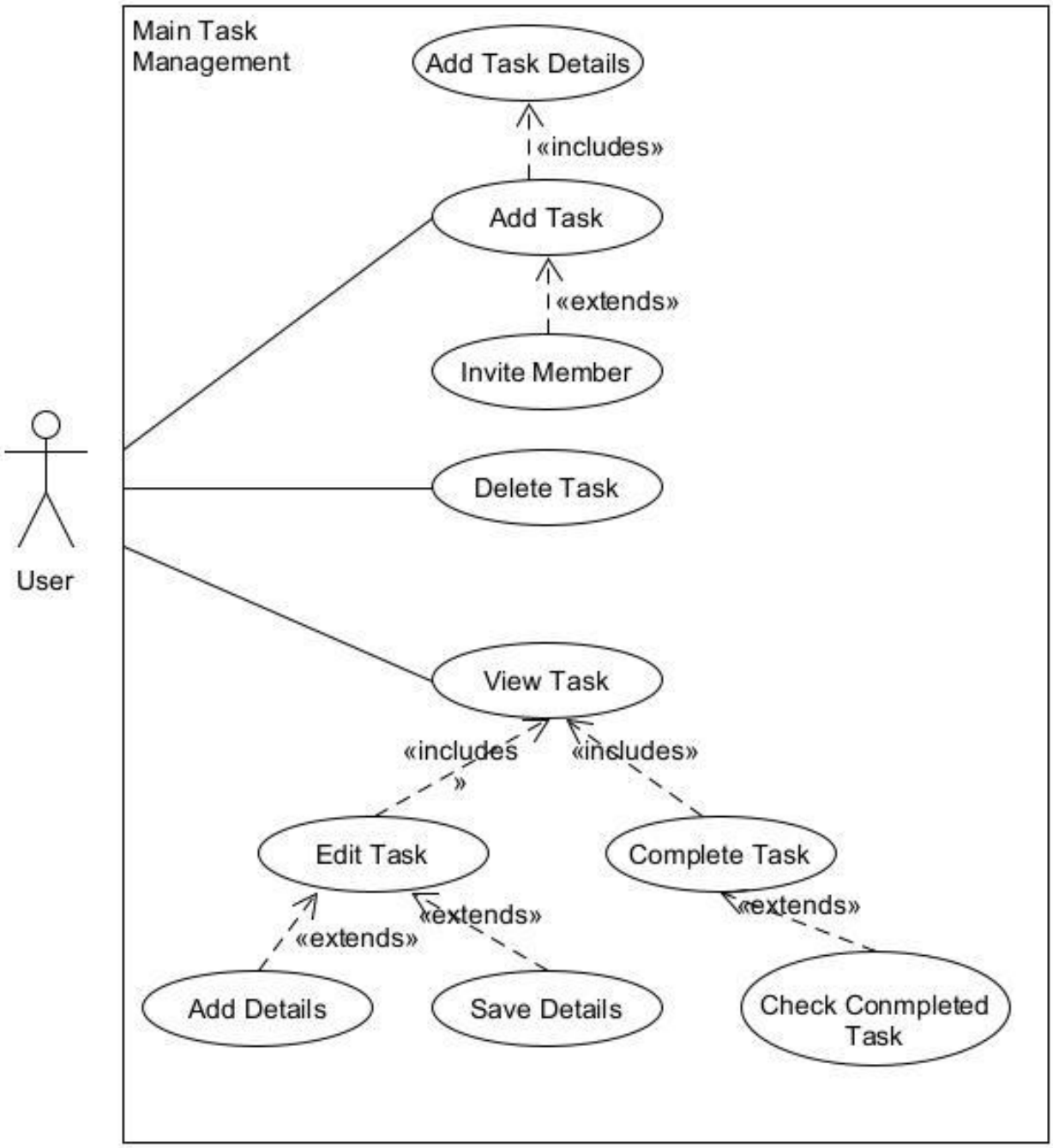
Use Case Name	**Delete Task**
Related requirements	10, 11, 12, 13, 14, 15, 33
Goal in context	To be able to delete tasks from the To-Do List
Preconditions	1. The user must be logged into their account 2. There should be at least one task in the list 3. The user should have the rights to delete the task
Successful End Condition	The user confirms the actions and the data is deleted from the database
Failed End Condition	The user cancels the delete actions and the data is not deleted
Primary Actors	User
Secondary Actors	Null
Triggers	1. The delete button is pressed 2. The confirmation button is pressed
Included Cases	Null
Main Flow	1. User presses the delete button 2. User select one or more task from list which will be highlighted 3. User presses Confirm Delete button 4. Warning popup will be displayed that will ask the user for confirmation. The confirm and cancel button will be displayed. 5. If the user presses cancel, the user will be directed to the main task lists window. 6. If the user presses confirm, the task will be deleted from the database and the user will be returned to the main task list

Use Case Name	**View Task**
Related requirements	16, 17, 18, 25, 26 , 27, 28, 33
Goal in context	To be able to view and edit the task information
Preconditions	1. The user must be logged in 2. There must be at least one task in the task list
Successful End Condition	The user can view the task details and interact with the various buttons
Failed End Condition	Database connection failure results in the user not being able to view the task
Primary Actors	Users
Secondary Actors	Null
Triggers	The user clicks on a task in To-Do/Completed list
Included Cases	Null
Main Flow	1. The user clicks on any task in the task list 2. The user is redirected to the task view page where they will be able to see the task details 3. The user can interact with the various buttons available in the task view page

Use Case Name	**Edit Task**
Related requirements	19, 20, 21, 22, 23, 24, 33
Goal in context	To be able to edit the task information
Preconditions	1. The user must be logged in 2. There must be at least one task in the main task list 3. The user must be in the specific task view page 4. The user must have rights to edit the task
Successful End Condition	Changes made to the task are successfully added in the database and the user is redirected to the task view page
Failed End Condition	The changes were canceled by the user and the user is redirected to the task view page
Primary Actors	User
Secondary Actors	Null
Triggers	Edit Task Information button is pressed
Included Cases	Null
Main Flow	1. The user selects a task and presses the edit task information button in the task view page 2. The system returns a form with the required fields 3. The user updates the fields 4. The system validates the information and updates the database accordingly 5. The user presses the change color code button 6. The system shows a pallet of colors to choose from 7. The user selects the required color 8. The system updates the color in the database

Use Case Name	**Complete Task**
Related requirements	29, 30, 31, 32, 33
Goal in context	To be able to mark the completion of a given task
Preconditions	1. The user must be logged in 2. There must be at least one task in the main task list 3. The user must have the rights to mark a task as completed
Successful End Condition	The task is checked as completed and is successfully moved from the To-Do list to the Completed list
Failed End Condition	1. To-Do list task is not moved into completed list and pop up will be displayed with error message. 2. Database gets disconnected.
Primary Actors	User
Secondary Actors	Null
Triggers	Checkbox is clicked which is placed beside each task in the To-Do list.
Included Cases	Null
Main Flow	1. Task checkbox is clicked in To-Do list. 2. The database will be updated and the task moved from To-Do list to Completed list. 3. User Interface will be updated and tasks will be moved from To-Do list to Completed list.

4.2.5 Use Case Diagram



4.3 Subtask Management

4.3.1 Description and Priority

This feature allows a user to create divisions of each main task/subtask. A main task can be subdivided into any number of subtasks. Each subtask can then be subdivided into even more subtasks. Each subtask will be listed under the view page of its parent task. Each subtask will have its own title, description and due date. The subtask will have all the functionalities of a main task, except it will always be in reference to a parent task. This is a medium priority feature as the application, although not complete, will be able to run without this functionality as well.

4.3.2 Stimulus/Response Sequences

Add Subtask Case: In the main task view page, the user will see a button to add a subtask to that particular task. When the user clicks this button they will be directed to a form similar to the one they are shown when they add a main task. The user will enter the task information such as the task title, description and due date. This information will be stored in the appropriate table in the database with its own unique ID as well as a reference to the parent task under which it was created. The user will be able to click on this subtask the same way they would click on a main task and will be prompted with the same information as explained in the View Task Case.

Delete Subtasks Case: When the user is in the task view page, they will have the option to click a button to delete certain subtasks. Once the delete button is pressed, the user will be able to select any number of subtasks from the list. After this, when the user presses the confirm delete button, they will be prompted with a warning. The user will be asked to confirm if they want to continue, and the selected tasks will then be removed from the database and from the subtask list.

View Subtasks Case: When the user clicks on any parent task in the main task list, they will be able to view the information as explained in the View Task Case. Along with that information they will also be able to view a list of all the subtasks created under the parent task. Each subtask will be clickable and upon clicking any one subtask it will show the same information as in the View Task Case. If a particular subtask is further divided into more tasks then the parent subtask will also show a completion bar and a list of further subdivided tasks created under it.

Edit Subtasks Case: When a user clicks the edit subtask details button in the subtask view page the system should display a form with all the information that was entered when the subtask was created. The fields in this form should be editable. A user can change any information they want and then press the save button. Once the save button is pressed the information is checked to see if it matches the required criteria and then updated in the appropriate database table. The user is then directed back to the subtask view page with the update task information

Complete Subtasks Case: Each subtask will be led by a checkbox marking its completion. Once a subtask is completed the user simply needs to click the checkbox. The task will then be greyed out and the checkbox will be ticked. This subtask will not be moved to the completed list, but would rather stay in the subtask list. If the user wishes to revert this action, they simply uncheck the checkbox and the subtask will no longer be greyed out indicating that it is not yet finished.

4.3.3 Functional Requirements

1. The task view page shall have a button to create subtasks
2. The add subtask button, once pressed shall display a form with text fields such as title, description and due date of the subtask
3. The text fields should allow the user to change the font styles
4. The system shall verify that the data meets the required criteria and shall add the new subtask to the appropriate database with a reference to the task under which it was created
5. The default color code for the subtasks should be transparent
6. The task view list should have a Delete Subtasks button
7. Once clicked the user should be able to select any number of subtasks from the given list and then press the Confirm Deletion button
8. The selected subtasks should be highlighted to indicate their selection
9. A warning prompt will be shown to the user asking him for confirmation to continue
10. The selected subtasks shall be removed from the database and the task view page under which they were created
11. Each subtask shall be clickable
12. Upon click the subtask view page should be displayed where the subtask information along with any further subdivisions should be displayed
13. The subtask view page shall have an edit information button (same working as 4.2) and an assign people button (explained in 4.4)
14. The subtask view page shall have a button to create subtasks (section 4.3)
15. The subtask view page shall have a completion bar only if further subtasks exist
16. The completion percentage shall be automatically calculated based on the number of completed tasks created under that specific subtask
17. The completion percentage of the subtask shall contribute to the overall task competition percentage
18. The task view list shall have a checkbox before each subtask
19. Once the checkbox is clicked the subtask shall be flagged as completed in the database
20. A completed task shall be greyed out and can be ungreyed by unchecking the checkbox.
21. Appropriate error messages should be displayed in case of any error

4.3.4 Use Case Tables

Use Case Name	**Add Subtask**
Related requirements	1, 2, 3, 4, 5, 23
Goal in context	To be able to create subtasks under a particular parent task
Preconditions	1. The user must be logged in 2. There must be at least one subtask under particular main parent task 3. The user must be in the specific task view page 4. The user must have rights to edit the subtask
Successful End Condition	The subtask is created and added to an existing parent task.
Failed End Condition	The subtask creation failed and is not added to the parent tasks.
Primary Actors	User
Secondary Actors	Null
Triggers	The Add Subtask button is pressed.
Included Cases	Null
Main Flow	1. User will press the Add Subtask button. 2. A window will popup with text fields such as title, description and due date of the subtask. 3. As soon as the user fills the text fields and clicks the Add Subtask button the task will be added to the parent task in the database. 4. And the user interface will be refreshed with the updated database that will show the added subtask under the parent task.

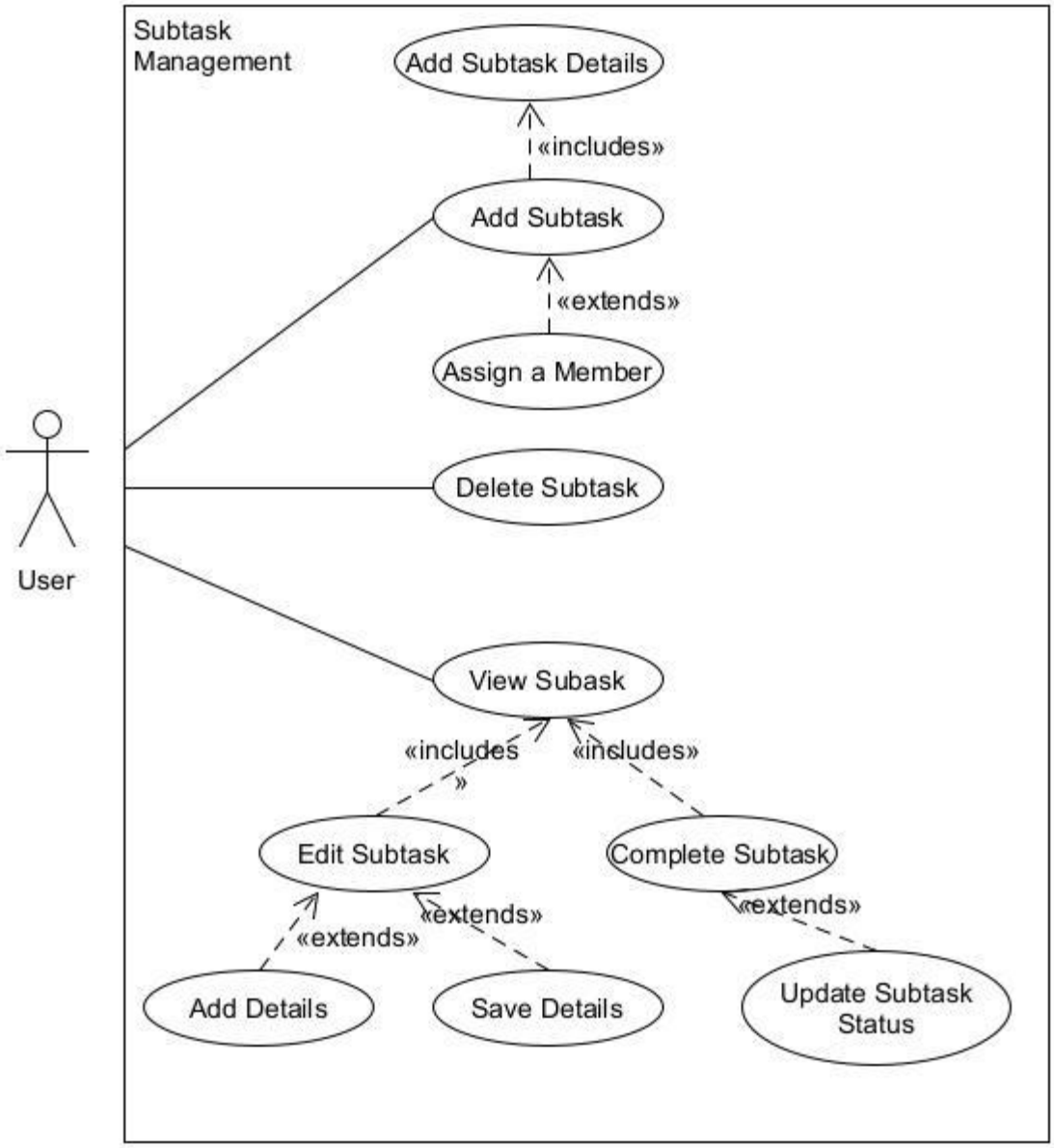
Use Case Name	**Delete Subtask**
Related requirements	6, 7, 8, 9, 10, 23
Goal in context	To be able to delete subtasks from under a given parent task
Preconditions	1. The user must be logged in 2. There must be at least one subtask created under the parent task 3. The user must have the rights to delete subtasks
Successful End Condition	The subtasks is removed from the database and the updated subtask list is shown in the subtask view page
Failed End Condition	The subtask deletion is canceled or fails and the subtasks are not removed from the database
Primary Actors	User
Secondary Actors	Null
Triggers	The delete subtasks button is pressed
Included Cases	Null
Main Flow	1. User presses the delete subtasks button 2. User select one or more subtasks from list which will be highlighted 3. User will press the Confirm Delete button 4. Warning popup will be displayed that will ask the user for confirmation. The confirm and cancel button will be displayed. 5. If the user presses cancel, the user will be directed to the main task lists window. 6. If the user presses confirm, the subtasks will be deleted from the database and the user will be returned to the parent task view page

Use Case Name	**View Subtask**
Related requirements	11, 12, 13, 14, 15, 16, 17, 18, 23
Goal in context	To be able to view and edit the subtask information
Preconditions	1. The user must be logged in 2. There must be at least one task in the task list 3. There must be at least one subtask under the main parent task
Successful End Condition	The user can view the subtask details and interact with the various buttons
Failed End Condition	Database connection failure results in the user not being able to view the task lists
Primary Actors	User
Secondary Actors	Null
Triggers	The user clicks on a task in To-Do/Completed list
Included Cases	Null
Main Flow	1. The user clicks on any task in the task list 2. The user is redirected to the task view page where they will be able to see the task details as well as the related subtasks 3. The user clicks on any subtask from parent's subtask list 4. The user is redirected to the subtask view page where they will be able to see the subtask details along with different interactive button will be displayed

Use Case Name	**Edit Subtask**
Related requirements	11, 12, 13, 23
Goal in context	To be able to edit the Subtask information
Preconditions	1. The user must be logged in 2. There must be at least one Subtask in the main task. 3. The user must be in the specific Subtask view page 4. The user must have rights to edit the Subtask
Successful End Condition	Changes made to the subtasks are successfully added in the database and the user is redirected to the task view page
Failed End Condition	The changes were canceled by the user and the user is redirected to the task view page
Primary Actors	User
Secondary Actors	Null
Triggers	Edit Subtask button is pressed
Included Cases	Null
Main Flow	1. The user selects a task and presses the edit Subtask information button in the subtask view page 2. The system returns a form with the required fields 3. The user updates the fields 4. The system validates the information and database is updated accordingly 5. The user presses the change color code button 6. The system shows a pallet of colors to choose from 7. The user selects the required color 8. The system updates the color in the database 9. The user is returned to the task view page

Use Case Name	**Complete Subtask**
Related requirements	19, 20, 21, 22, 23
Goal in context	To be able to mark the completion of a given subtask
Preconditions	1. The user must be logged in 2. There must be at least one subtask under the main parent task 3. The user must have the rights to mark a subtask as completed
Successful End Condition	The subtask checked as completed is greyed out
Failed End Condition	The subtask marked as completed is not changing the color to grey
Primary Actors	User
Secondary Actors	Null
Triggers	Checkboxes beside each subtask is clicked
Included Cases	Null
Main Flow	1. Subask checkbox is clicked in the task view page. 2. The task view page will be updated with checked subtasks which will be greyed out . 3. Parent task completion bar will be updated.

4.3.5 Use Case Diagram



4.4 Task Distribution

4.4.1 Description and Priority

Task distribution is a feature which allows the users to distribute the tasks created in a list. These tasks can either be a main task or a subtask of any main task. The user will have a privilege to distribute the task to their team members and monitor them. He can also edit or delete the tasks or assign a single task to multiple group members. This feature will allow the users to work effectively in a team and maximize productivity. This is a medium priority feature as it's not expected for every user to work in a team and they can use this application even without this functionality.

4.4.2 Stimulus/Responses

Invite Task Case: Once a task has been created, the user will have an option to assign any team members to that task. To use this functionality the user will be prompted to enter an email address. The creator will need to select a specific role for the invited member.. An email will then be generated with an invite link and sent to the specified email. The invited member must accept the invite link and have an account to work as a team member within that specific task.

Edit Role Case: Upon invitation, the creator will have to specify the role that the invited member will play. This role will either be that of an editor or a viewer. An editor will be able to mark any subtask within the main task as complete whereas the viewer will only be able to see the main task and subtasks and not be able to mark their completion. Adding, deleting, editing and assigning member privileges will only be in the hands of the creator of the main task itself.

Assign Task Case: A creator will have the option to click on any subtask within a parent task and select, from a drop down list of invited members, who they want the task to be assigned to. The invited members will be able to see their name written next to the subtask to indicate that they are assigned to that particular subtask. This will only be a visual functionality available to the team members.

Remove Member Case: The creator of a task would have an option to remove a participating member from that task. Once removed, the shared task will be removed from the invited member's main task list and they will not be able to view it and its subtasks until they are invited again

Remove Assignment Case: Once the creator removes a member assigned to a specific subtask it will remove their name from that subtask. However, they will still be able to see the tasks and its subtasks within their own main task list.

4.4.3 Functional Requirements

1. There should be a button to invite people to a specific task
2. Once pressed a prompt will ask the user to input an email address and specify the role the invited member will have
3. The system will generate an email and send it to the specified email
4. Upon accepting the invite link, the user will be redirected to workspace if he already has an account.
5. If the invited member doesn't have an account, he would be redirected to the sign up page.
6. When the desired member has accepted the link he will be able to view the task and its subtasks within his own main task list
7. All members shall be able to see the people invited and assigned to the the task and its subtask
8. There should be two different roles that can be assigned to an invited member.
9. The creator should be able to select and change the role that is specified for an invited member
10. The creator will have the option to assign an invited team member to a particular subtask using the button of ‘ Assign a task’.
11. The selected member will then gain the rights to the task based on the particular role that has been specified for them.
12. The task creator can remove any member from a task or remove a member assignment from subtask using the ‘Remove member’ or “Remove Assignment” button.
13. A person removed from a task would not be able to view the task or its subtasks anymore
14. A person removed from a subtask would still be able to view the task and its subtasks.
15. Appropriate error messages should be displayed in case of any error

4.4.4 Use Case Tables

Use Case Name	**Invite Task**
Related requirements	1, 2, 3, 4, 5, 6, 7, 15
Goal in context	Invite a user to an existing task
Preconditions	1. Creator must have a task 2. User must know the email address of particular user that needs to be added
Successful End Condition	Invited user is added to a particular task
Failed End Condition	User is not added to a task and an error message is displayed
Primary Actors	User (Task Creator)
Secondary Actors	Invited user
Triggers	Invite Member button is pressed
Included Cases	Null
Main Flow	1. User presses an invite button. 2. User will enter an email in a popup window asking about the email 3. Email is generated which will be send to invited user 4. The invited user will press the link send it via email 5. If an invited user has an account it will be redirected to the login window and then the invited user will follow the login steps. 6. After login the invited user is redirected to the main task lists window 7. If invited user does not have a account it will be redirected to the sign up page and after sign up it will be redirected to main task lists window

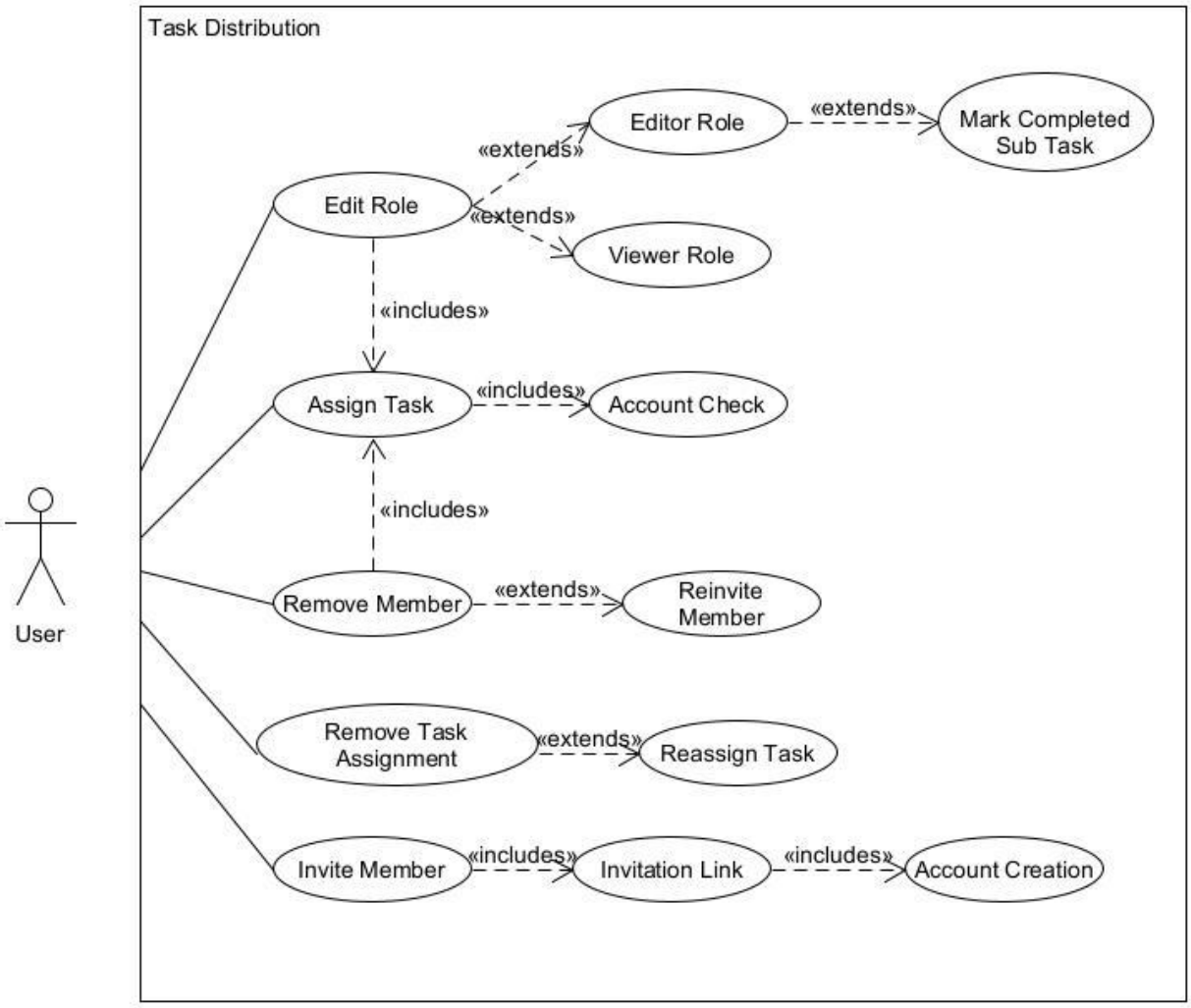
Use Case Name	**Edit Role**
Related requirements	8, 9, 15
Goal in context	Edit the role of an incited member
Preconditions	1. The user should be invited to the task
Successful End Condition	The user role is changed and the invited user's rights are implemented successfully
Failed End Condition	The user role is not changed. The task view page is returned
Primary Actors	User (Task Creator)
Secondary Actors	Invited user
Triggers	The change role button is pressed
Included Cases	Null
Main Flow	1. User can edit the role of invited user by choosing from a drop down menu placed before each invited user in invited users list belonging to one particular task 2. Once the user chooses the role option of an invited user they would be granted rights based on the role

Use Case Name	**Assign Task**
Related requirements	10, 11, 15
Goal in context	Assign a subtask to an invited user
Preconditions	1. The user should be invited to the task
Successful End Condition	Invited User is assigned to specific subtask and they are able to see their name next to the particular subtask
Failed End Condition	Invited user is not assigned to specific subtask and the creator is returned to the view task page
Primary Actors	User (Task Creator)
Secondary Actors	Invited user
Triggers	The Assign Subtask To button is pressed
Included Cases	Null
Main Flow	1. User will click the assign task button of a particular subtask 2. All the invited user of that task will be displayed in a popup window 3. The creator will select the user that they want to assign to that subtask 4. The added user will be able to see their name next to the subtask assigned to them

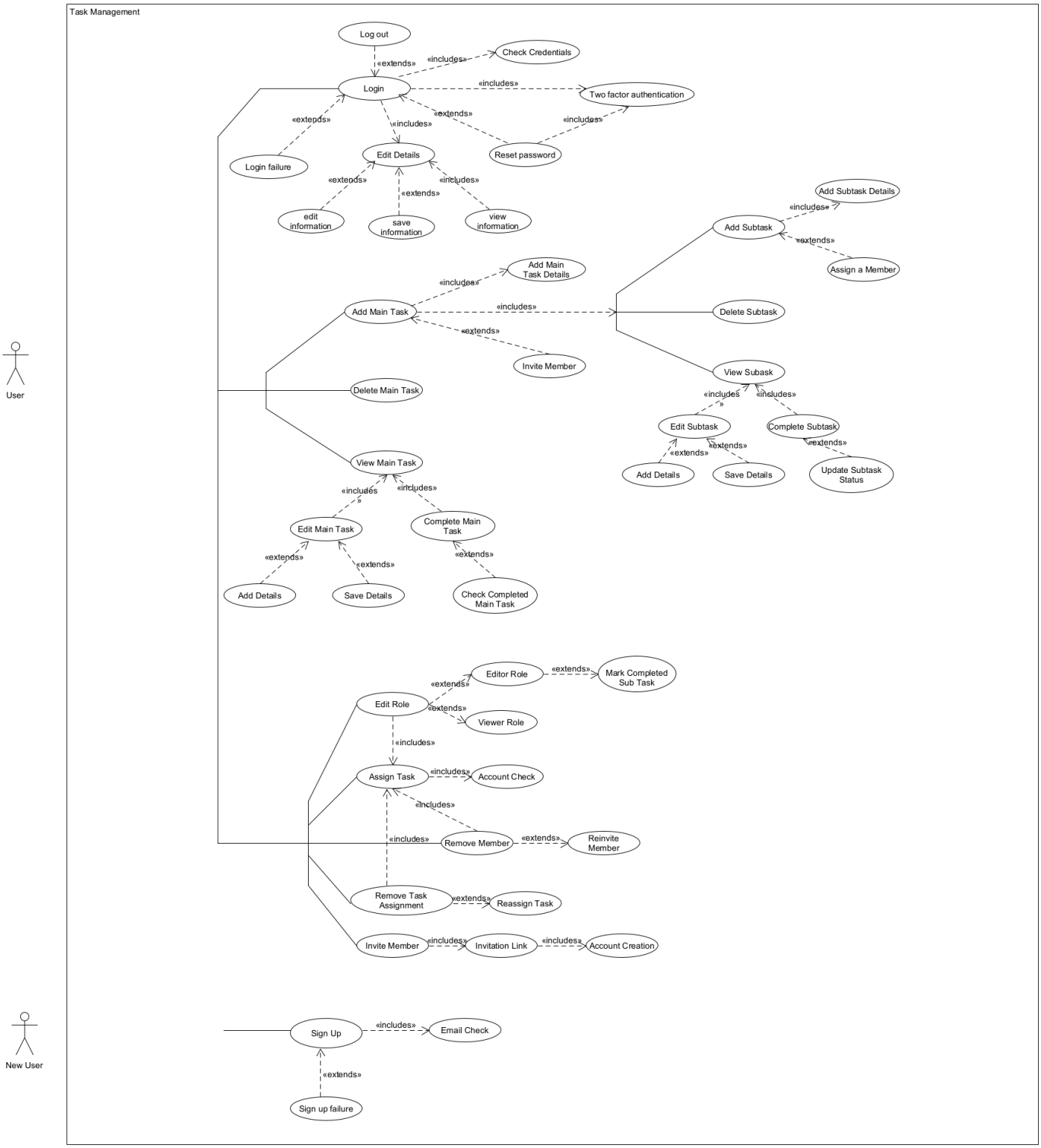
Use Case Name	**Remove Member**
Related requirements	12, 13, 15
Goal in context	An invited user is removed from the task
Preconditions	1. The user must be invited to the task 2. The user to be removed must not be the creator themself
Successful End Condition	The selected user is removed from the task and they are no longer able to access/view the task in their main task list
Failed End Condition	User is not added to a task
Primary Actors	User (Task Creator)
Secondary Actors	Invited user
Triggers	Remove Member button is pressed
Included Cases	Null
Main Flow	1. User will click the remove member button 2. The list of user will be displayed invited to that task 3. The user will click the user that it want to remove from the task 4. The pop up window will ask for a confirmation 5. If user press confirm the user will be removed 6. Removed user cannot view the task until they are invited back

Use Case Name	**Remove Assignment**
Related requirements	12, 14, 15
Goal in context	Remove the user from a specific subtask
Preconditions	1. The user to be removed must have a subtask assigned to them
Successful End Condition	User is un-assigned from the subtask and the removed user is no longer able to see his name next to the subtask.
Failed End Condition	User is not un-assigned and their name is still visible next to the particular subtask
Primary Actors	User
Secondary Actors	Invited user
Triggers	Remove Assignment button is pressed
Included Cases	Verify Credentials, Two-Factor Authentication
Main Flow	1. User press remove assignment button next to the subtask 2. The assigned user list will be displayed in popup window 3. User will then select the member it want to a remove from the subtask 4. Pop up window will ask for a confirmation 5. If user confirm the continuation of above task the user will be remove from the assignment task list 6. The removed user will still be able to view the task and its subtasks but will have their name removed from the particular subtask

4.4.5 Use Case Diagram



4.5 Overall Use Case Diagram



5. Other Nonfunctional Requirements

5.1 Performance Requirements

The initial load time of the product would be dependent on the internet connection as well as on the medium used to access the product. The performance of the application will have a greater dependency on the hardware capability of the user's device that means a user will be able to use the application, but if the hardware resources of the desktop or mobile device do not meet the criteria, it will compromise the usability.

5.2 Safety Requirements

A user must make sure that he invites only credible members to his workspace to avoid unnecessary disturbance. The user can have the functionality to choose a supervisor in the task list containing multiple members. To encounter the issue of the deletion of the task list, the creator of the task list can only delete it, or he can make another reliable person as a supervisor to have this authority. It will prevent damage and maintain the integrity of tasks. A warning would also be shown while sharing the invite link of a workspace mentioning that it can cause possible loss or damage to the user's workspace.

5.3 Security Requirements

To make sure that the user's data is only accessible to them, a two-factor authentication system would be implemented to prevent unauthorized access. The two-factor authentication will provide an extra layer of security. A user cannot log in on the application by just providing their username and password. They will be responsible for issuing extra information such as verification from email. The email will be used to send an authentication code to enhance security and overcome vulnerabilities.

5.4 Software Quality Attributes

Correctness: The application will consistently provide specified functionality and objectives that were discussed with the client, such as the task should be added to the task list correctly and in order, as specified by the user. This would be achieved by following different testing techniques.

Usability: An interactive user interface would provide ease of use and will help the users to manage tasks easily. The users would be able to organize tasks in different workspaces which can also be sorted based on the user's priority. Users would also be able to customize their workspace theme according to their preferences. Lastly, the application would be easy to learn for any user so that they can maximize its productivity.

Maintainability: As the product includes many functionalities, the code should be easy to understand and manageable. To ensure the maintainability of the project, the code will include comments and object oriented programming techniques will be emphasized. These will help the development team to debug and test the product without any difficulties. Furthermore, if the client wants to integrate functionalities or expand the product, it would be feasible for the developers to refer to the code and modify it.

Reliability: Multiple users can use the application at the same time without any interference or errors. The mutually exclusive processes cannot interfere with other processes regardless of any circumstances except if the client has specified it. The developers will make sure to minimize the error rate to enhance the reliability of the product.

Portability: The project would be built as a Mobile as well as a Desktop application. The user would be able to access it from both of the devices having consistent experience and functionalities.

Availability: The application and server would be available 24/7 for the users to access and manage their tasks.

5.5 Business Rules

1. Only the user who has created the task, is allowed to assign a user or edit that task.
2. The creator assigns the rights of task to a user.
3. User's personal information is strictly confidential and can only be accessed by the user.
4. Only developers have access to the database.
5. Active email is required for signup and/or password reset.
6. The password cannot be the name of the user, it should be unique.
7. The application is a freeware software, but not open-source.
8. The application will not be integrated into a third-party application.
9. The application will not host any advertisement of another application or organisation.

6. Other Requirements

1. Software protected by Copyright law.
2. Database requirement: Oracle 19c and multiple dedicated systems which will work 24/7.
3. Continuous monitoring of database, system and internet connectivity.
4. Product overview, a complete configuration of the database software (Oracle 19c), technical details and contact information of the developers would be provided.
5. Multiple high speed internet with static IP connection for the server

Appendix A: Glossary

- **SRS** - Software Requirements Specification
- **Functional Requirements** - The services provided by the software application without which its goal for origin cannot be fulfilled; they define the specific behaviour of the application
- **Non-Functional Requirements** - These are requirements that essentially can be used to judge the operation of the system, for example, performance
- **Root Task** - The task that was created first in a chain of created subtasks
- **Parent Task** - The task that is immediately above in linkage to a task
- **Child Task** - The task that is an immediate subtask of another task
- **Two-Factor Authentication** - An extra security feature that is activated after a user successfully logs in using their email and password; this can be a code sent to an added cell number or a code to another added email address
- **OTP** - One-Time Password is the code which will be sent to the user's email after a successful login attempt or after a forget password case; the code will be useless after its first use.
- **Prompt** - A message in a dialog box that appears immediately in response to an action the user performs in the application
- **Main Task List** - The ultimate tasks listed which are in midst of completion
- **View Task Page** - The dialog box that appears once a main task is clicked; the box displays all the details of the task
- **View Subtask Page** - The dialog box that appears once a subtask is clicked; the box displays all the details of the subtask
- **Redirect** - Automated navigation to a webpage as the result of an action on the current one
- **Role** - A state assigned to a user's account in relation to a task which allows the user to either edit, or watch the updates on the task
- **Supervisor** - A member from among the members of a common task which has the full control over the manipulation of task properties and rights
- **Database** - A combination of, the collection of records that hold the details of user accounts, and the software package that allows the task management application to use those details
- **Sign-up Page** - The webpage or form where a user will add all the relevant information that is needed by the application in the desired format before the application allows the said user to create an account
- **Log-in Page** - The form or webpage where a user with an already existing account enters their credentials to successfully enter into their account on the application
- **Stimulus/Response Sequence** - A definition of the required actions that will need to be performed by the user in relation to a particular goal within the service sphere of the software, followed by the immediate response from the software of that action

Appendix B: To Be Determined List

1. Lock button to suspend all activities within a task
2. Feature of attaching files or images
3. Recurring task (daily, weekly, monthly or yearly)
4. Automatically flag the main task as completed once all the subtasks are marked complete
5. Dedicated desktop and mobile application.