Software Requirements Specification

for

<Task Management Application>

Version 1.0 approved

Prepared by <Group No. 4>

<Section No. 4>

<10/10/2021>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This SRS describes the software requirements of a task management application (Name undecided), [Application name] 1.0. It is a general use productivity application intended for all sorts of users. [The application name] will permit users to manage their everyday and professional tasks, keep track of deadlines, and distribute tasks among multiple users.

1.2 Document Conventions

In section 4, the Stimulus/Response Section, each Stimulus/Response is described individually for each feature.
The words project and list are used interchangeably throughout the SRS.

1.3 Intended Audience and Reading Suggestions

The SRS will serve as a reference to the project specifications and requirements. This report is to be used by members of the project team that will execute, implement, and validate the correct functioning of the system. Each team member will find all the sections of the SRS essential.

1.4 Product Scope

[Name] 1.0 is a desktop application that will provide effortless scheduling and time management for its users. The task management application will have an identical Android application to ensure that users can maintain their productivity on the go. The system will also allow effective task allocation and project planning to ensue so that stakeholders of all kinds are able to work with the [Name] 1.0.

1.5 References

Use-case diagram source code

2. Overall Description

2.1 Product Perspective

Name 1.0 is a new system that aids in diminishing the need to utilize handmade notes, pens, and papers to keep check of tasks. The application is designed to run on personal computers and mobiles for handling user assignments and productivity. The user is able to form tasks and lists, invite other users of the application and divide tasks between them, and maintain track of their deadlines through notifications generated by the application. The system is expected to evolve

over one release, for now, and would have updates when necessitated. The updates will allow other characteristics and functions to become a part of the application which will be added in the SRS if needed.

2.2 Product Functions

F.1: Create an account or log in (if existing user).
F.2: Create new tasks
F.3: Create a list of tasks/a project
F.4: Set deadlines
F.5: Assign priorities to a task
F.6: Delete tasks/lists
F.7: Invite different users to access the same list/project
F.8: Distribute tasks among different users
F.9: Mark tasks/lists as completed
F.10: See completed tasks/lists
F.11: Create subtasks of existing tasks
F.12: Two-factor Authentication
F.13: Show tasks under progress
F.14: Show list under progress
F.15: Reset deadline
F.16: Show people working on a list
F.17: Reassign tasks to people
F.18: Reset priority
F.19: Edit task contents

2.3 User Classes and Characteristics

.

2.3.1 Teachers:

Teachers can use this application to make their work easier as they can give assignments to students and can keep track of the deadline for that particular assignment.

2.3.2 Students:

Students can use this application to get their assignments done by dividing the assignment in small chunks until the day of deadline making it easier to do the assignment.

2.3.3 Project managers:

Project managers can use this application to keep track of their deadlines and it will also make their teamwork effective as they can easily divide tasks among the group members.
They will be the ones who will mostly use this application.

2.3.4 General users:

The user classes defined above are not the only ones who can use this application; it can be used by coordinators, HRs, CEOs to increase the productivity in their respective companies as they can collaborate with their employees using this application and can divide the workload accordingly.
People can also use the app for personal use and everyday tasks.

2.4 Operating Environment

This software will work on all hardware that runs on windows or android OS. It requires an internet connection to operate, users can use this software from anywhere around the globe. All data will be stored in the database and will be updated in Real time. Java/Python libraries will be used, and MySQL databases will be used to store data.

2.5 Design and Implementation Constraints

The system must be programmed in an object oriented language. The system must use an internet connection or the user will not be able to communicate with the database. The Task Management System must be portable and easy to download so that multiple computers may be used to look at the information.The SQL database must be attached to the system. The app will not be available for iOS for now. The project team is responsible for managing the software.

2.6 User Documentation

No user documentation information at this time. However, a small FAQ and user help manual will be created once the application is fully developed.

2.7 Assumptions and Dependencies

One assumption that could affect the design is that the user will have adequate internet connection; this could affect the speed with which the interface communicates with the database. Another assumption is that users must have an email account to access the application. This system will be written for users with a basic understanding of how computers work. Users with less computer experience may have a harder time.
The application is dependent on a sql database to store data and it is also dependent on java libraries for user interface.

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 Create an Account and Log-in

4.1.1 Description and Priority

Sign-up/Sign-in is one of the essential features of the system. Without this characteristic, the user will be incapable of accessing the contents of [Name] 1.0. This feature is of high priority and is especially required to keep the user data secure since [Name] 1.0 users will also include project managers and office workers handling their tasks and deadlines. These users cannot afford to have their data jeopardized.

4.1.2 Stimulus/Response Sequences

Stimulus: The user opens the application.
Response: The front screen displays options for Sign-in/Sign-up

Stimulus: The user clicks the sign-in option.
Response: The application shows a form asking for user credentials (email and password)

Stimulus: The user inputs credentials and clicks the login button
Response: The system verifies the credentials, briefly shows a welcome screen before displaying the main screen of the application. The system will also request the user to set up two-factor authentication if they haven't done so. If the user enters incorrect credentials, the application displays the message 'incorrect email or password. Retry.'

Stimulus: The user clicks on the sign-up option.
Response: A form is displayed to the user requesting the user to enter their name, email, password, and re-enter the password. Once the user clicks the create account button, they will be directed to the main screen of the app. The user also will be prompted to set up two-factor authentication (small pop-up before the main screen loads), but that would be optional.

Stimulus: The user clicks the two-factor authentication option.
Response: The user will be prompted to set up the 2FA (method of 2FA is TBD)

4.1.3 Functional Requirements

1. ***Create an account:*** *The system will allow the user to and access the features of the application.*
2. ***Sign-in into an existing account:*** *The user will be allowed to access the features and existing contents of the application.*
3. ***Two-factor authentication:*** *For data security, the system will prompt the user to authenticate their account every time they log in.*

4.2 Create Tasks

4.2.1 Description and Priority

This feature is also one of the most important parts of the system. The user is able to manage everyday tasks such as students keeping track of deadlines etc. It also is of high priority like the previously mentioned feature. It is one of the basic features where the user is able to keep track of their deadlines and priorities. It can only be accessed by the user of the application and no one else.

4.2.2 Stimulus/Response Sequences

Stimulus: User Clicks create task button
Response: A screen to add details to a new task is opened. Two text boxes and a button that says "Next". The user can add the main heading of the title and the details of the task in the text box below the heading.

Stimulus: The user presses the 'Next' button.
Response: The screen shows options for setting a deadline and setting priority to the task.

Stimulus: The user presses the 'Set Deadline' button.

Response: The user is shown a Calendar where they can set a date for the deadline and a clock through which they can set a time for the deadline.

Stimulus: The user presses 'Done' button
Response: The system takes the user back to the screen where the buttons of set priority and set deadline exist. With a deadline set. The timer will start once the user confirms the task creation.

Stimulus: The user presses the 'Set priority' button.
Response: The system shows three options to the user. High, medium, low priority. These buttons will decide the placement of the task in the list and the number of times the user gets reminder notification. However, this is not a mandatory function and the user has the choice to not set the deadline. The system will assume that the task is of low priority.

Stimulus: The user presses 'Done' button
Response: The system takes the user back to the screen where the buttons of set priority and set deadline exist.

Stimulus: The user presses 'Back' button
Response: The system takes the user back to the part where they can add details of the task.

Stimulus: The user presses the 'Back' button again.
Response: The system takes the user back to the main page with no task created.

Stimulus: The user presses 'Create Task' button (after setting deadline and/or priority)
Response: The system creates the task, starts the timer for the deadline and places the task in a queue according to its priority.

4.2.3 Functional Requirements

1. ***Create Task:*** *The system will allow the user to create a task, and establish details of the task.*
2. ***Set Deadline:*** *The system will allow the user to set a time and date for the task.*
3. ***Set Priority:*** *The system will allow the user to set a priority (High, medium, low) to a task*

4.3 Create Subtask

4.3.1 Description and Priority

Creating subtasks is a medium priority part of the system. The user is able to divide major tasks into different parts so that they are manageable and easy to keep track of. This may be particularly useful for students, teachers or office workers to keep track of their tasks through subdivisions of a major task.

4.3.2 Stimulus/Response Sequences

Stimulus: User Clicks create subtask button

Response: A screen to add details to a new subtask is opened. Two text boxes and a button that says "Next". The user can add the main heading of the title and the details of the task in the text box below the heading.

Stimulus: The user presses the 'Next' button.
Response: The screen shows options for setting a deadline and setting priority to the subtask.

Stimulus: The user presses the 'Set Deadline' button.
Response: The user is shown a Calendar where they can set a date for the deadline and a clock through which they can set a time for the deadline.

Stimulus: The user presses 'Done' button
Response: The system takes the user back to the screen where the buttons of set priority and set deadline exist. With a deadline set. The timer will start once the user confirms the task creation.

Stimulus: The user presses the 'Set priority' button.
Response: The system shows three options to the user. High, medium, low priority. These buttons will decide the placement of the task in the list and the number of times the user gets reminder notification. However, this is not a mandatory function and the user has the choice to not set the deadline. The system will assume that the task is of low priority.

Stimulus: The user presses 'Done' button
Response: The system takes the user back to the screen where the buttons of set priority and set deadline exist.

Stimulus: The user presses 'Back' button
Response: The system takes the user back to the part where they can add details of the task.

Stimulus: The user presses the 'Back' button again.
Response: The system takes the user back to the main task page with no subtask created.

Stimulus: The user presses 'Create Subtask' button (after setting deadline and/or priority)
Response: The system creates the subtask, starts the timer for the deadline and places the subtask in a queue (of task) according to its priority.

4.3.3 Functional Requirements

1. ***Create subtasks of existing tasks:*** *The system allows users to divide the task into easy manageable tasks.*
2. ***Set Deadline:*** *The system will allow the user to set a time and date for the task.*
3. ***Set Priority:*** *The system will allow the user to set a priority (High, medium, low) to a task.*

4.4 Create List/Project

4.4.1 Description and Priority

Create list/project is a high priority system feature that allows users to create and keep track of multiple tasks and their subtasks and divide tasks between multiple users. This is essential for users such as project managers, students, teachers, etc who would find it easier to manage group projects, fix deadlines for other users and set priorities with ease.

4.4.2 Stimulus/Response Sequences

Stimulus: User Clicks create list button
Response: A screen to add details of the new list is opened. Two text boxes and a button that says "Next". The user can add the main heading of the title and the details of the project in the text box below the heading.

Stimulus: The user presses the 'Next' button.
Response: The screen shows option adding tasks And invite people to the list.

Stimulus: User Clicks add task button
Response: A screen to add details to a new task is opened. Two text boxes and a button that says "Next". The user can add the main heading of the title and the details of the task in the text box below the heading.

Stimulus: The user clicks 'invite other users'
Response: The system will allow the user to invite other users, the other users will only be able to access the contents of the list after they have accepted the invite.

Stimulus: The user presses the 'Next' button.
Response: The screen shows options for setting a deadline and setting priority to the task.

Stimulus: The user presses the 'Set Deadline' button.
Response: The user is shown a Calendar where they can set a date for the deadline and a clock through which they can set a time for the deadline.

Stimulus: The user presses 'Done' button
Response: The system takes the user back to the screen where the buttons of set priority and set deadline exist. With a deadline set. The timer will start once the user confirms the task creation.

Stimulus: The user presses the 'Set priority' button.
Response: The system shows three options to the user. High, medium, low priority. These buttons will decide the placement of the task in the list and the number of times the user gets reminder notification. However, this is not a mandatory function and the user has the choice to not set the deadline. The system will assume that the task is of low priority.

Stimulus: The user presses 'Done' button

Response: The system takes the user back to the screen where the buttons of set priority and set deadline exist.

Stimulus: The user presses the 'Done' button again.
Response: The system takes the user back to the screen where the user can add more tasks.

Stimulus: The user presses 'Back' button
Response: The system takes the user back to the part where they can add details of the task.

Stimulus: The user presses the 'Back' button again.
Response: The system takes the user back to the main page with no task created.

Stimulus: The user presses 'Create List' button (after setting deadline and/or priority)
Response: The system creates the task, starts the timer for the deadline and places the task in a queue according to its priority.

4.4.3 Functional Requirements

1. ***Create List:*** *The user will be allowed to create a project where they can add multiple tasks and their subtasks.*
2. ***Create Task:*** *The system will allow the user to create a task, and establish details of the task.*
3. ***Set Deadline:*** *The system will allow the user to set a time and date for the task.*
4. ***Set Priority:*** *The system will allow the user to set a priority (High, medium, low) to a task.*
5. ***Invite other users:*** *The system allows users to invite other users to see the contents of the tasks.*

4.5 Show Completed Tasks

4.5.1 Description and Priority

It is a medium priority system feature that enables users to access their previously completed tasks, check their contents and delete them from the list if needed.

4.5.2 Stimulus/Response Sequences

Stimulus: The user clicks ‘show the completed’ task button.
Response: The system shows a list of all completed tasks of the user.

Stimulus: The user clicks a task.
Response: The system shows details of the task including deadline and priority and subtasks.

Stimulus: The user clicks 'Delete task'
Response: The system removes the task from the queue permanently including its subtasks.

Stimulus: The user clicks a subtask.
Response: The system shows details of subtask.

Stimulus: The user clicks 'Delete Subtask'
Response: The system removes the subtask only from the queue.

4.5.3 Functional Requirements

1. ***Show completed tasks:*** *The system allows the user to display all completed tasks.*
2. ***Delete task:*** *The system removes task from the list*

4.6 Show Completed Lists

4.6.1 Description and Priority

Similar to the Show Completed Tasks, it is a medium priority feature that allows users to access their previous projects and delete them if they wish to.

4.6.2 Stimulus/Response Sequences

Stimulus: The user clicks the 'show completed tasks' button.
Response: The system shows a list of all completed lists of the user.

Stimulus: The user clicks a list
Response: The system shows all the tasks in the particular list.

Stimulus: The user clicks a task.
Response: The system shows details of the task including deadline and priority and subtasks.

Stimulus: The user clicks 'Delete List'
Response: The system removes the list from the queue permanently including its tasks and subtasks.

Stimulus: The user clicks 'Delete task'
Response: The system removes the task from the queue permanently including its subtasks.

Stimulus: The user clicks a subtask.
Response: The system shows details of subtask.

Stimulus: The user clicks 'Delete Subtask'
Response: The system removes the subtask only from the queue.

4.6.3 Functional Requirements

1. ***Show completed list:*** *The system allows the user to display all completed tasks.*
2. ***Show completed tasks:*** *The system allows the user to display all completed tasks.*
3. ***Delete task:*** *The system removes tasks from the list.*
4. ***Delete project:*** *The system removes projects from the list.*
5. ***Show completed subtasks:*** *The system allows the user to display all completed subtasks.*

6. ***Delete subtask:*** *The system removes subtask from the list.*

4.7 Create an Account and Log-in

4.7.1 Description and Priority

Show Tasks Under Progress is also a high priority system feature where the user will be able to access and manipulate the contents of the task, reset deadlines or priority according to the changes made in their real life schedules.

4.7.2 Stimulus/Response Sequences

Stimulus: The user clicks the 'show tasks under progress' button.
Response: The system shows a list of all tasks that have not reached their deadline yet.

Stimulus: The user clicks a task.
Response: The system shows details of the task including deadline and priority and subtasks.

Stimulus: The user clicks 'Add subtask'
Response: The system shows/goes through the process of the feature 'Create Subtask' mentioned above.

Stimulus: The user clicks 'Reset Deadline'
Response: The system takes the user to set a deadline screen where they can reset their deadlines.

Stimulus: The user clicks 'Delete task'
Response: The system removes the task from the queue permanently including its subtasks.

Stimulus: The user clicks a subtask.
Response: The system shows details of subtask.

Stimulus: The user clicks 'Delete Subtask'
Response: The system removes the subtask only from the queue.

Stimulus: The user clicks the check option besides the task.
Response: The Task will be removed from the progress list to completed list

Stimulus: The user clicks the check option beside the subtask.
Response: The subtask will be marked completed but will move to the completed list once the main task is marked completed.

Stimulus: The user clicks reset priority for a task.
Response: The system takes the user to the set priority page where they can reset the task priority. (Same stimulus/response for subtasks)

Stimulus: The user clicks the Edit button.
Response: The user is given permission to re-edit the task.

4.7.3 Functional Requirements

1. ***Show tasks under progress:*** *The user is allowed to see the contents of tasks under progress.*
2. ***Create subtask:*** *The system allows the user to create a subtask.*
3. ***Reset Deadline:*** *The system allows the user to change their task/subtask deadlines.*
4. ***Delete task:*** *The system removes tasks from the list.*
5. ***Delete subtask:*** *The system removes subtask from the list.*
6. ***Reset priority:*** *The system will allow users to reset priorities for any task/subtask they want to alter.*
7. ***Edit contents:*** *The system will allow users to make changes in the contents of tasks/subtasks.*

4.8 Show Lists/Projects Under Progress

4.1.1 Description and Priority

Show Projects Under Progress is a system feature that would allow the user to keep track of all the projects they are managing without a hassle. They would be aware which team member is assigned what task and what deadlines without having to write them down on paper. They can also modify the contents and redistribute work when needed. It is also a high priority feature.

4.1.2 Stimulus/Response Sequences

Stimulus: The user clicks show lists/projects under progress button.
Response: The system shows a list of all of all projects that have not reached their deadline yet.

Stimulus: The user clicks 'Invite people'
Response: The system shows the screen where user can invite people to work on the same project (The method of sending invite is TBD however it is most likely to be through emails)

Stimulus: The user clicks 'Show people working'
Response: The system shows the user the number of people working on the project and what tasks they are assigned.

Stimulus: The user clicks 'Reassign Tasks'
Response: The system will allow the user to redistribute tasks among the people who are working currently on the project.

Stimulus: The user clicks a task.
Response: The system shows details of the task including deadline and priority and number of people working on it and subtasks of those tasks.

Stimulus: The user clicks 'Add Task'

Response: The system shows/goes through the process of the feature 'Create Task' as mentioned above.

Stimulus: The user clicks 'Assign Task'
Response: The system will allow the user to assign tasks to one or multiple people.

Stimulus: The user clicks 'Reset Deadline'
Response: The system takes the user to set a deadline screen where they can reset their deadlines of specific tasks/subtask or the whole list itself.

Stimulus: The user clicks 'Add subtask'
Response: The system shows/goes through the process of the feature 'Create Subtask' mentioned above.

Stimulus: The user clicks 'Delete task'
Response: The system removes the task from the queue permanently including its subtasks.

Stimulus: The user clicks a subtask.
Response: The system shows details of subtask.

Stimulus: The user clicks 'Delete Subtask'
Response: The system removes the subtask only from the queue.

Stimulus: The user clicks the check option beside the list/project.
Response: The list will be moved to the completed lists/projects.

Stimulus: The user clicks the check option besides the task.
Response: The task will be marked completed but will move to the completed list once the main project.is marked completed.

Stimulus: The user clicks the check option beside the subtask.
Response: The subtask will be marked completed but will move to the completed list once the main project is marked completed.

Stimulus: The user clicks reset priority for a task/project/subtask.
Response: The system takes the user to the set priority page where they can reset the task priority. (Same stimulus/response for subtasks)

Stimulus: The user clicks Edit button
Response: The user is given permission to re-edit the task/project/subtask.

4.1.3 Functional Requirements

1. ***Show lists under progress:*** *The user will be allowed to access the contents of the list.*
2. ***Show tasks under progress:*** *The user is allowed to see the contents of tasks under progress.*
3. ***Create subtask:*** *The system allows the user to create a subtask.*
4. ***Reset Deadline:*** *The system allows the user to change their task/subtask deadlines.*
5. ***Delete task:*** *The system removes tasks from the list.*

6. ***Delete subtask:*** *The system removes subtask from the list.*
7. ***Reset priority:*** *The system will allow users to reset priorities for any task/subtask they want to alter.*
8. ***Edit contents:*** *The system will allow users to make changes in the contents of tasks/subtasks.*
9. ***Invite other users:*** *The system allows users to invite other users to see the contents of the tasks.*
10. ***Reassign Tasks:*** *The system will allow the user to switch tasks between users or change who works on what task/subtask.*

4.9 Use Case Tables

Use Case Name	Create project	
Related Requirements	P.1 Users can combine to work on a project of tasks as a project in teams. They will be assigned individual responsibilities and deadlines, and the group will be able to collectively see the progress of all tasks/projects, both ongoing and completed.	
Goal in Context	User adds a grouped task in the form of a project.	
Precondition	User(s) is/are logged in/signed up	
Successful end condition	project Successfully created	
Failed end condition	Request timeout, server disconnect	
Primary Actors	User(s)	
Secondary Actors	None	
Trigger	User clicks the 'Create project' button	
Included Cases		
Main Flow	**Step 1:**	Users Sign up
	Step 2:	Users Log in
	Step 3 **Include:: Add deadline**	Set a deadline for each user's task in project
	Step 4:	Task Added
	Step 5: **Include:: distribute**	Distribute the tasks among the users/team.
	Step 6 **Include:: Add deadline**	Set a deadline for all users
	Step 7:	Tasks shown, assigned, and added

Use Case Name	Add deadlines (Projects/teams)
Related Requirements	P.2 The user/supervisor/manager can set a deadline(s) of the task or members. The deadline can be extended, altered, or reset by the Manager.
Goal in Context	User assigns deadlines to a tasks from Calendar.
Precondition	User has invited members and assigned tasks.
Successful end condition	Deadlines successfully assigned.
Failed end condition	Invalid date, server timeout, User unavailability
Primary Actors	User (Manager)
Secondary Actors	None
Trigger	User clicks the 'add Deadline to tasks' button
Included Cases	
Main Flow	Step 1: User creates a project Step 2: Picks a date and time for tasks Step 3 Include:: Add deadline Sets deadlines Step 4: deadlines Assigned

Use Case Name	Invite other users (Projects/teams)
Related Requirements	p.3 The user/supervisor/manager can invite users to collaborate on a project. The manager then assigns deadlines and tasks to every user.
Goal in Context	manager assigns deadlines and tasks to invitees..
Precondition	User has created a project.
Successful end condition	Users successfully invited
Failed end condition	Invalid invite method, no project exists, User unavailability
Primary Actors	User (Manager)
Secondary Actors	None
Trigger	User clicks the 'invite other users' button
Included Cases	
Main Flow	Step 1: User creates a project Step 2: User invites other users through a certain medium (TBD) Step 3: Invitation method successful Step 4: Invited users join.

Use Case Name	Project Added (Create Project)
Related Requirements	p.5 After the user has created a project, assigned deadlines, and added the contents, the project is created. There can be multiple editing or options that can be done to a task after it is added.
Goal in Context	Show the users that their project is added.
Precondition	User has successfully filled the required details to add and assign the tasks.
Successful end condition	Tasks are created and displays progress for all users.
Failed end condition	Invalid input in fields, task duplication
Primary Actors	User (Manager)
Secondary Actors	Users (Team) (Read-only)
Trigger	User successfully creates project.
Included Cases	Check created projects
Main Flow	Step 1: User makes a project Step 2: User invites the team and assign tasks. Step 3 Extends:: Edit priorities: The Manager can edit the priorities of tasks. Step 4: Extends:: edit deadline: The deadlines can be edited to an earlier or later date. Step 5: Extends:: edit contents: The contents of the task can be edited by manager if the requirements change. Step 6 Include:: delete task: Option to delete task if it no longer required

Use Case Name	Tasks in Progress
Related Requirements	T.1 After the user has created a task, assigned deadline to it, and added the contents, the task is created. The purpose is to show the task in this section as an in-progress incomplete task.
Goal in Context	Show the user their task(s) which are incomplete/in-progress.
Precondition	User has successfully filled the required details to add the tasks.
Successful end condition	Task(s) is/are created and shown as incomplete.
Failed end condition	Invalid input in fields, task doesn't exist
Primary Actors	User
Secondary Actors	None
Trigger	User successfully creates task.
Included Cases	Check created tasks
Main Flow	Step 1: User adds a task Step 2: The task is yet to be completed. Step 3: The deadline hasn't approached yet. Step 4: Added to a list with all the tasks still in progress

Use Case Name	Distribute Tasks (Projects/teams)
Related Requirements	p.4 The user is required to assign tasks to group members, so he does so by distributing tasks among the team and sets individual deadlines.
Goal in Context	Manager successfully assigns tasks to all users.
Precondition	User has created a project.
Successful end condition	Users assigned tasks.
Failed end condition	Invalid invite method, no project exists, User unavailability
Primary Actors	User (Manager)
Secondary Actors	None
Trigger	User clicks the 'invite other users' button
Included Cases	Check existing project
Main Flow	Step 1: User creates a project Step 2: User has invited other users and they have joined the team. Step 3: Assign tasks to individuals with deadlines Step 4: Tasks successfully assigned.

Use Case Name	Projects in Progress
Related Requirements	T.2 After the manager has created a project, distributed it to users, assigned deadlines to it, and added the contents, the project is created. The purpose is to show the project in this section as an in-progress incomplete project.
Goal in Context	Show the user their project(s) which are incomplete/ in-progress.
Precondition	User has successfully filled the required details to add and assigned the tasks.
Successful end condition	projects(s) is/are created and shown as incomplete.
Failed end condition	Invalid input in fields, project doesn't exist, deadline eexpired
Primary Actors	User (Manager)
Secondary Actors	Users (Team) (Read-only)
Trigger	Manager successfully creates task.
Included Cases	Check created projects
Main Flow	Step 1: User makes a project Step 2: The tasks are yet to be completed by **'Any'** user(s). Step 3: The deadline hasn't approached yet. Step 4: Added to a list with all the projects still in progress

Use Case Name	Delete Task (Add task→ Tasks in progress
Related Requirements	T.1.1 After the user has assigned deadlines and added the contents, the possibility exists that the task is no longer required and hence needs to be deleted.
Goal in Context	Delete a task no longer required
Precondition	User has successfully added a task
Successful end condition	Task is from current tasks in progress
Failed end condition	Task already completed
Primary Actors	User
Secondary Actors	None
Trigger	User successfully creates task and now needs to delete it.
Included Cases	
Main Flow	Step 1: User adds task Step 2: User reviews the task's progress Step 3: User doesn't require the task anymore. Step 4: User clicks the 'Delete task' option. Step 5: Task successfully deleted

Use Case Name	Delete Project (Create Project→ projects in progress
Related Requirements	T.2.1 After the user has assigned deadlines and added the contents, the possibility exists that the project is no longer required by the manager/organization, and hence needs to be deleted.
Goal in Context	Delete a project no longer required
Precondition	User has successfully added a project
Successful end condition	Task is from current projects in progress
Failed end condition	project already completed
Primary Actors	User (Manager)
Secondary Actors	None
Trigger	Manager successfully creates project and now needs to delete it.
Included Cases	
Main Flow	Step 1: User adds Project Step 2: User reviews the project's progress Step 3: User doesn't require the project anymore. Step 4:: User clicks the 'Delete Project' option. Step 5: project successfully deleted

Use Case Name	Show Completed Task
Related Requirements	C.1 The tasks which were completed before deadline expiration are shown.
Goal in Context	Show all the tasks successfully completed
Precondition	User has completed all the objectives of task(s).
Successful end condition	Show the tasks completed before deadline expiration.
Failed end condition	-
Primary Actors	User
Secondary Actors	None
Trigger	User successfully completes tasks.
Included Cases	
Main Flow	Step 1: User adds Task Step 2: User completes the required objectives of the task. Step 3: User finishes the task before deadline. Step 4:: The task is added to the 'show completed tasks' list.

Use Case Name	Show Completed Projects
Related Requirements	D.1 The projects which were completed before deadline expiration by all the users are shown.
Goal in Context	Show all the projects successfully completed
Precondition	Manager checks and evaluates all tasks as complete.
Successful end condition	Show the project 'completed' before deadline expiration.
Failed end condition	-
Primary Actors	User-Manager (view-only)
Secondary Actors	None
Trigger	Manager checks the project as complete.
Included Cases	
Main Flow	Step 1: Manager adds project. Step 2: Users complete the required objectives of their respective tasks. Step 3: Users finish all the tasks before deadline. Step 4: The project is added to the 'show completed projects' list.

Use Case Name	Add Task
Related Requirements	A.1The user is able to keep track of the tasks they are working on, see the tasks they have completed and the progress on the ongoing ones this helps the user to lead a smooth observation of project(s).
Goal in Context	User adds a new task to be completed.
Precondition	User is logged in/signed up
Successful end condition	Task Successfully created
Failed end condition	Request timeout
Primary Actors	User
Secondary Actors	None
Trigger	User clicks the add task button
Included Cases	
Main Flow	Step 1: User Signs up Step 2: User Logs in Step 3 Include:: Add deadline: Set a deadline Step 4: Task Added Step 5: Include:: delete: Option to delete task Step 6: Task shown and added

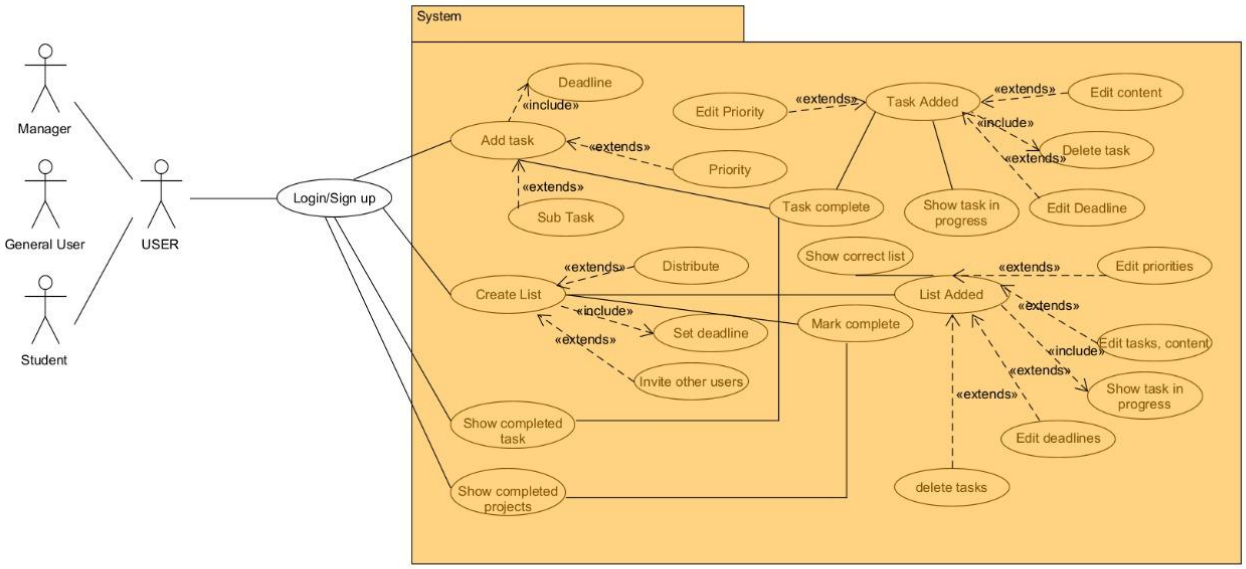
Use Case Name	Add deadline for (Tasks)
Related Requirements	A.2The user can set a deadline of the task at hand. The deadline can be extended, altered, or reset.
Goal in Context	User assigns a deadline to a task from Calendar.
Precondition	User has started creating a task.
Successful end condition	Deadline successfully assigned.
Failed end condition	Invalid date, server timeout
Primary Actors	User
Secondary Actors	None
Trigger	User clicks the 'add Deadline to task' button
Included Cases	
Main Flow	Step 1: User creates a task Step 2: Picks a date and time Step 3 Include:: Add deadline Sets a deadline Step 4: deadline Assigned

Use Case Name	Add Priority For (tasks)
Related Requirements	A.3The user can assign priorities to the task. This way the user can prioritize which task to complete first and assign the appropriate deadline to.
Goal in Context	User assigns priorities to tasks.
Precondition	User is logged in/signed up
Successful end condition	Priority to a task assigned
Failed end condition	Task Expired, Server disconnect
Primary Actors	User
Secondary Actors	None
Trigger	User clicks the 'Assign priority' button
Included Cases	
Main Flow	Step 1: User adds task Step 2: User analyzes the task list Step 3 Sets priorities to tasks and deadline Step 4: Task assigned priorities. Step 5: Option to alter priority Step 6 Task shown and prioritized

Use Case Name	Add sub task (task)
Related Requirements	A.2.1 The user can add sub tasks to a main task. This way the user can segmentize a task into small sub tasks and an easily track progress.
Goal in Context	User assigns sub tasks to main tasks.
Precondition	User is logged in and created a task.
Successful end condition	Sub tasks assigned
Failed end condition	Task Expired, Server disconnect
Primary Actors	User
Secondary Actors	None
Trigger	User clicks the 'Assign subtask' button
Included Cases	
Main Flow	Step 1: User adds task Step 2: User analyzes the task list Step 3 Adds a sub task with their own deadlines Step 4: Sub tasks created. Step 5: Option to delete or merge sub tasks Step 6 Sub tasks added

Use Case Name	Task Added (Add task)
Related Requirements	A.4 After the user has assigned deadlines and added the contents, the task is added. There multiple editing or options that can be done to a task after it is added.
Goal in Context	Show the user that their task is added.
Precondition	User has successfully filled the required details to add task
Successful end condition	Task is created and displays progress
Failed end condition	Invalid input in fields, task duplication
Primary Actors	User
Secondary Actors	None
Trigger	User successfully creates task.
Included Cases	
Main Flow	Step 1: User adds task Step 2: User reviews the task's progress Step 3 Extends:: Edit priority: The trigger to edit the priority of task. Step 4: Extends:: edit deadline: The deadline can be edited to an earlier or later date. Step 5: Extends:: edit contents: The contents of the task can be edited if the requirements change. Step 6 Include:: delete task: Option to delete task if it no longer required

4.10 Use Case diagram



5 Other Nonfunctional Requirements

5.1 Performance Requirements

- *The system shall perform basic operations such as add, remove, or update tasks quickly without freezing. There will be no delay time for the user.*
- *The user interface shall be easy to grasp.*
- *The system shall run on the Windows 10 and android operating system.*
- *Tasks will be queued priority-wise (high to low) so that the user can keep track of their tasks in order of importance.*
- *The system shall communicate between Java/Python libraries and SQL databases*

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

- *The safety of this software is mostly related to the user and their data. A User-Password system shall be placed to ensure security and not more than 4 attempts shall be given in case of an incorrect entry.*
- *The user shall be reminded to set up a two-factor authentication every day if they haven't done so.*
- *Only the programmer shall have access to the database which holds the user To-Do List with a confidential agreement in place.*
- *The users invited to the list will only have read-only access to the contents of the list.*

5.4 Software Quality Attributes

5.4.1 Usability:

Since it is a general productivity app. The GUI will be made in a user-friendly way. The user will be able to use the software easily right on the first go. It will be able to perform add, delete, or other specific tasks easily.

5.4.2 Maintainability:

The system code will be written in such a way that it is easier for programmers to modify it according to the dynamic requirements of the user. It shall be composed keeping the benefit of other programmers to whom the software is sold.

5.4.3 Portability:

The system is portable as it will run on windows and android hence can be used by anyone with a smartphone and a laptop.

5.5 Business Rules

The application is valid for 13+ users. The application can be used by people of all professions and other general users.

6 Other Requirements

The application will also show a bar that displays the percentage of tasks completed. The percentage of task completion will be divided based on the number of subtasks in a task and all subtasks and tasks in lists.The team sees it as something that will be a part of the code rather than a functional requirement.

Appendix A: Glossary

- ***Tasks****: a piece of work to be done or undertaken by the user.*
- ***Subtasks****: a task that is part of a more complex task created by the user to manage more efficiently.*
- ***Priority****: condition of being regarded or treated as more important than others.*
- ***User Credentials****: a username and password combination used for logging in to online accounts.*
- ***Queued****: a collection of entities that are maintained in a sequence.*

Appendix B: Analysis Models

No Analysis Models available at this time.

Appendix C: To Be Determined List

- *Method of Two factor authentication.*
- *Method of inviting other users.*
- *Name of the application.*