


# Software Requirements Specification for To Do List

Version 1.0

PREPARED BY EDA KOCAMAN  
202028003



## Table of Contents

<b>1.</b>	<b>1.Introduction .....</b>	<b>1</b>
1.1	Purpose .....	1
1.2	Scope.....	1
1.3	References.....	1
1.4	Overview.....	1
1.5	Assumptions & Dependencies .....	1
1.6	Design and Implementation Constraints.....	2
<b>2.</b>	<b>2.General Description .....</b>	<b>3</b>
2.1	Product Perspective .....	3
2.2	Product Functions .....	3
2.3	Users and Characteristics.....	4
2.4	General Constraints .....	4
2.5	Use Cases.....	5
2.5.1	Use case: Sign in/Register to system.....	5
2.5.2	Use case: Log in/Log out the system.....	6
2.5.3	Use case: Create a new task .....	7
2.5.4	Use case: Create a new list .....	8
2.5.5	Use case: Set task a deadline .....	9
2.5.6	Use case: Edit List.....	10
2.5.7	Use Case: Delete Task.....	11
<b>3.</b>	<b>3.The Specific Requirements.....</b>	<b>12</b>
3.1	External Interface specifications.....	12
3.1.1	User Interfaces.....	12
3.1.2	Hardware Interfaces .....	12
3.1.3	Software Interfaces .....	13
3.1.4	Communications Interfaces .....	13
3.2	Functional Requirements .....	14
3.2.1	Use Case Diagram .....	14
3.2.2	Activity Diagram.....	27
3.3	Detailed non-functional requirements .....	28
3.4	System Evolution.....	28

4.	<b>4. ....Other Non-functional Requirements</b>	<b>28</b>
4.1	Software Quality Attributes .....	28

**Revision History**

Name	Date	Reason For Changes	Version
SRS	29.05.2023	First Version created	1.0
			1.1.
			1.2

# **1. Introduction**

## **1.1 Purpose**

This Software Requirements Specification (**SRS**) document defines the specifications and requirements for version 1.0 of the “To Do List” project. The purpose of this application is to help users manage their daily tasks and responsibilities.

## **1.2 Scope**

Day by day, people's occupations and tasks that need to be done during the day are increasing. After a while, it becomes very difficult to follow the tasks. To do list application makes it easier for people to plan their day and manage the tasks they need to do more easily.

## **1.3 References**

1. "IEEE Guide for Software Requirements Specifications," in *IEEE Std 830-1984*, vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205.

## **1.4 Overview**

- Section 2 describes the general factors that affect the product and its requirements.
- Section 3 contains all the details the software developer needs to create the software.
- Section 4 defines technologies, infrastructure, performance, scalability, quality attributes, and other non-functional requirements.

## **1.5 Assumptions & Dependencies**

### **Assumptions:**

- Users will have access to compatible devices (e.g., smartphones, tablets) running supported operating systems (e.g., iOS, Android) to use the To-Do List App.
- Users will have a stable internet connection to access cloud-based services, perform backups, and synchronize their data.
- Users will have basic knowledge of how to operate mobile applications and navigate through the user interface.
- The user's device will have sufficient storage capacity to accommodate the To-Do List App and its associated data.

**Dependencies:**

- The To-Do List App may rely on external libraries, frameworks, or APIs for certain features or functionalities. The availability and compatibility of these dependencies should be ensured during the development process.
- The app may require access to device features like the camera, calendar, or notifications. The availability and proper functioning of these device features depend on the underlying operating system and hardware.
- Cloud-based storage and backup services are relied upon for securely storing and backing up user data. The app's functionality is dependent on the availability and reliability of these services.
- The To-Do List App may require periodic updates and maintenance to address issues, introduce new features, or ensure compatibility with new device models or operating system versions.

## **1.6 Design and Implementation Constraints**

Assumptions and dependencies in article 1.5 should be considered throughout the design and development process of the To-Do List App to ensure that the app functions as intended and meets the users' expectations.

## **2. General Description**

### **2.1 Product Perspective**

The system will be mobile-based application. User registration will be required.

### **2.2 Product Functions**

#### **Create To-Do Lists:**

Users can create multiple to do lists to organize their tasks based on different categories, projects, or priorities. Each list acts as a container for related tasks.

#### **Add Tasks:**

Users can add new tasks to their to do lists by providing a title, description, and optional details such as due dates, priority levels, or labels. This function allows users to capture and document their tasks effectively.

#### **Set Due Dates:**

Users can assign due dates or deadlines to tasks to keep track of their time-sensitive responsibilities. This helps users prioritize their tasks and ensures that they are completed within the desired timeframe.

#### **Mark Tasks as Completed:**

Users can mark tasks as completed once they are finished. This function allows users to track their progress and provides a sense of accomplishment as tasks are checked off the list.

#### **Edit and Delete Tasks:**

Users have the ability to make changes to existing tasks, such as modifying task details, updating due dates, or revising task descriptions. They can also delete tasks that are no longer relevant or necessary.

#### **Organize Tasks:**

The app provides features to help users organize their tasks within each to do list. This may include options to reorder tasks, group them by priority or category, or apply labels or tags for easy identification and filtering.

#### **Reminders and Notifications:**

The app may include a reminder feature that allows users to set notifications or alerts for upcoming or overdue tasks. This helps users stay on top of their responsibilities and ensures that important deadlines are not missed.

## **2.3 Users and Characteristics**

The app is suitable for individuals from various occupations who need to organize their personal tasks and responsibilities. Professionals, students, homemakers, and anyone seeking a tool to manage their daily to do lists can use it.

## **2.4 General Constraints**

**User Interface:** The application should have an intuitive and user-friendly interface that allows users to easily add, edit, and manage their tasks. The interface should be responsive, visually appealing, and accessible on different devices and platforms.

### **Task Management:**

The application should support basic task management functionalities such as creating tasks, setting due dates, assigning priorities, categorizing tasks into different lists or categories, and marking tasks as completed.

### **Synchronization and Data Storage:**

To do list applications often need to synchronize tasks and data across multiple devices. This requires a reliable data storage solution and a mechanism to synchronize changes made on one device with all other devices connected to the user's account.

### **Reminders and Notifications:**

Users should have the ability to set reminders for important tasks and receive notifications at specified times or intervals. The application should provide options for customizing reminder settings and supporting various notification channels (e.g., email, push notifications).

### **Collaboration and Sharing:**

Some to do list applications offer collaboration features, allowing users to share tasks or lists with others, assign tasks to team members, and track progress collectively. This constraint involves implementing user permissions, real-time updates, and communication features.

### **Data Privacy and Security:**

As with any application handling personal data, maintaining data privacy and security is crucial. The application should employ secure authentication mechanisms, encryption protocols, and follow best practices to protect user information.

### **Performance and Scalability:**

The application should be capable of handling a large number of tasks and users without significant performance degradation. Optimizations should be implemented to ensure quick response times, efficient data retrieval, and smooth user experience.

### **Integration and Interoperability:**

To do list applications may need to integrate with other platforms or services, such as calendar apps, email clients, or project management tools. APIs or plugins should be available to facilitate these integrations and enable data exchange between systems.

### **Offline Functionality:**

Providing offline functionality allows users to access and modify their task lists even when they do not have an internet connection. Changes made offline should synchronize seamlessly once the connection is restored.

### **Customization and Personalization:**

Offering customization options allows users to tailor the application to their preferences. This may include themes, colour schemes, task-sorting options, and the ability to create custom tags or labels.

## **2.5 Use Cases**

### **2.5.1 Use case: Sign in/Register to system**

Actor: User

Description:

- Users should enter their e-mail, password, gender, date of birth, and level of education when creating an account.

Preconditions:

- The user has downloaded and installed the To-Do List App on their device.
- The app is accessible and running on the user's device.

Post-conditions:

- The user is successfully signed in or registered and gains access to their personalized account.

Main Flow:

1. The user launches the To-Do List App on their device.
2. The app displays the login/register screen.
3. If the user already has an account:
  - 3.1. The user enters their registered email address or username in the provided input field.
  - 3.2. The user enters their associated password in the password field.
  - 3.3. The user selects the "Sign In" button.
4. If the user does not have an account:
  - 4.1. The user selects the "Register" or "Sign Up" option on the login/register screen.



- 4.2. The app presents a registration form with required fields such as name, email address, and password.
- 4.3. The user enters the necessary information in the registration form.
- 4.4. The user submits the registration form.

## **2.5.2 Use case: Log in/Log out the system**

Actor: User

Description:

- This use case describes the process of logging in to and logging out of the To-Do List App.

Preconditions:

- The user has downloaded and installed the To-Do List App on their device.
- The user has a registered account in the app.

Post-conditions:

- For Logging In:  
The user gains access to their personalized account and can start managing their to do lists and tasks.
- For Logging Out:  
The user is successfully logged out and no longer has access to their account and app features.

Main Flow - Log In:

1. The user launches the To-Do List App on their device.
2. The app displays the login screen.
3. The user enters their registered email address or username in the provided input field.
4. The user enters their associated password in the password field.
5. The user selects the "Log In" or "Sign In" button.

Main Flow - Log Out:

1. The user is currently logged in to their account within the To-Do List App.
2. The user navigates to the account or profile settings section of the app.
3. The user selects the "Log Out" or "Sign Out" option.
4. The app prompts the user for confirmation to log out.
5. The user confirms their intent to log out.
6. The app terminates the user session and logs the user out of the app.
7. The app displays the login screen, signalling that the user has been successfully logged out.

### **2.5.3 Use case: Create a new task**

Actor: User

Description:

- This use case describes the process of creating a new task in the To-Do List App.

Preconditions:

- The user has successfully logged into the app.
- The user has accessed the appropriate screen or section for creating a new task.

Post-conditions:

- A new task is created and added to the users to do list.

Main Flow:

1. The user selects the "Add Task" button or option in the app.
2. The app presents a form or input fields to capture task details.
3. The user enters the required information for the task, such as the task title and optional details (description, due date, priority, etc.).
4. The user submits the task details.
5. The app validates the entered information for completeness and consistency.
6. If any errors or missing information are found, the app displays appropriate error messages and prompts the user to correct them.
7. If the information is valid, the app creates a new task with the provided details.
8. The app adds the new task to the user's existing to do list or creates a new list if none exists.

## **2.5.4 Use case: Create a new list**

Actor: User

Description:

- This use case describes the process of creating a new list in the To-Do List App.

Preconditions:

- The user has successfully logged into the app.
- The user has access to the appropriate screen or section for managing lists.

Post-conditions:

- A new list is created and added to the user's list of to do lists.
- The user can start adding tasks to the newly created list.

Main Flow:

1. The user selects the "Create New List" button or option in the app.
2. The app presents a form or input field to capture the details of the new list.
3. The user enters a unique name or title for the new list.
4. The user may optionally provide a description or additional details for the list.
5. The user submits the list details.
6. The app validates the entered information for completeness and uniqueness.
7. If any errors or missing information are found, the app displays appropriate error messages and prompts the user to correct them.
8. If the information is valid, the app creates a new list with the provided details.
9. The app adds the new list to the user's existing collection of lists.
10. The user can start adding tasks to the newly created list or perform other actions within the app.

### **2.5.5 Use case: Set task a deadline**

Actor: User

Description:

- This use case describes the process of assigning a deadline to a task in the To-Do List App.

Preconditions:

- The user has successfully logged into the app.
- The user has created or accessed a task in their to do list.

Post-conditions:

- The task is updated with the specified deadline.

Main Flow:

1. The user selects the task to which they want to assign a deadline.
2. The app displays the details of the selected task, including its title, description, and current status.
3. The user selects the "Set Deadline" option or an associated icon/button for the task.
4. The app presents a date and time picker or an input field to capture the desired deadline.
5. The user selects a date and, if applicable, a time for the deadline.
6. The user submits the selected deadline.
7. The app validates the entered deadline to ensure it is a future date and time.
8. If the entered deadline is invalid (e.g., past date, conflicting with other tasks), the app displays an error message and prompts the user to select a valid deadline.
9. If the entered deadline is valid, the app updates the task with the specified deadline.
10. The app confirms the successful assignment of the deadline and notifies the user.
11. The user can view the updated task details, including the assigned deadline.

### **2.5.6 Use case: Edit List**

Actor: User

Description:

- This use case describes the process of editing an existing list in the To-Do List App.

Preconditions:

- The user has successfully logged into the app.
- The user has created or accessed the list they want to edit.

Post-conditions:

- The list is updated with the edited details.

Main Flow:

1. The user selects the list they want to edit from their list of to do lists.
2. The app displays the details of the selected list, including its name, description, and associated tasks.
3. The user selects the "Edit List" option or an associated icon/button for the list.
4. The app presents a form or input fields populated with the current details of the list.
5. The user makes the necessary changes to the list's name, description, or any other editable fields.
6. The user submits the edited details.
7. The app validates the entered information for completeness and consistency.
8. If any errors or missing information are found, the app displays appropriate error messages and prompts the user to correct them.
9. If the information is valid, the app updates the list with the edited details.
10. The app confirms the successful update of the list and notifies the user.
11. The user can view the updated list details, including the edited information.

### **2.5.7 Use Case: Delete Task**

Actor: User

Description:

- This use case describes the steps involved in deleting a task from the To-Do List App.

Preconditions:

- The user must be logged in to the app.
- The user must have access to the task they want to delete.

Post-conditions:

- The task is permanently deleted from the user's task list.
- Any associated data or attachments related to the task are also deleted.

Main Flow:

1. The user navigates to the specific task they want to delete.
2. The user selects the task and chooses the "Delete" option.
3. The app prompts the user to confirm the deletion.
4. The user confirms the deletion by selecting the "Yes" or "Delete" option.
5. The app removes the task from the user's task list and updates the user interface accordingly.
6. The app displays a confirmation message indicating that the task has been successfully deleted.
7. The user can continue using the app or perform other actions.

## **3. The Specific Requirements**

### **3.1 External Interface specifications**

#### **3.1.1 User Interfaces**

- **Homepage:** When users open the application, they should see a homepage. This page can provide an overview, for example information such as the number of unfinished tasks, calendar view or important reminders.
- **Task Lists:** Users should be able to create new task lists and view existing lists. Lists should allow users to group tasks by a particular category, project, or job. Easily switch between lists
- **Tasks:** Users should be able to create, edit and delete tasks in each task list. Features such as title, description, priority level, assigned person and deadline should be able to be entered for tasks.
- **Calendar View:** Users should be able to schedule and track their tasks in the calendar view based on the due dates of the tasks. In this view, users should be able to access detailed information, with tasks highlighted in colours and being clickable.
- **Reminders:** Users should be able to add reminders to a specific task and receive notifications when reminders are due. This ensures that users do not forget their tasks and complete them on time.
- **User Account:** Users should be able to sign up for the app, log in, and manage their account settings. In this way, users can create custom lists, edit profile information and set preferences.

#### **3.1.2 Hardware Interfaces**

- **Supported device types:** To-Do List App should be able to run on different devices. This allows it to be used on different platforms such as desktop computers, laptops, tablets and smartphones. It is important that users can install and use the application on their preferred devices.
- **Storage:** The application needs device memory or disk space to store users' task lists and other data. This is important so that users can save tasks, sync data and access locally.
- **Other External Devices (optional):** Integration with external devices can be provided if the application has some special requirements. For example, features such as synchronization with smartwatches, microphone or barcode reader to support voice commands.

### **3.1.3 Software Interfaces**

- **Menu Bar:** A menu bar located at the top of the application screen provides users with the main navigation options. This menu bar can contain options such as accessing task lists, creating new lists, accessing account settings.
- **Home:** The home page, which will greet users when the application is opened, provides an overview. This page can show information such as the number of unfinished tasks, calendar view or important reminders.
- **Task Lists:** Users are directed to a task list that they choose from the home page or menu. The task list page allows users to add new tasks to the list, view and edit existing tasks.
- **Task Details:** When users select a particular task, they can access the details of the task. This page displays information such as the task's title, description, priority, assignee, deadline, and users can edit this information.
- **Calendar View:** Users can schedule and track their tasks in the calendar view based on the due dates of the tasks. In this view, users can access detailed information as tasks are highlighted by colours and are clickable.
- **Notifications:** The app can send important reminders and notifications to users. These notifications can inform the user about topics that interest the user, such as task deadlines, updates from the assignee, or unfinished tasks.

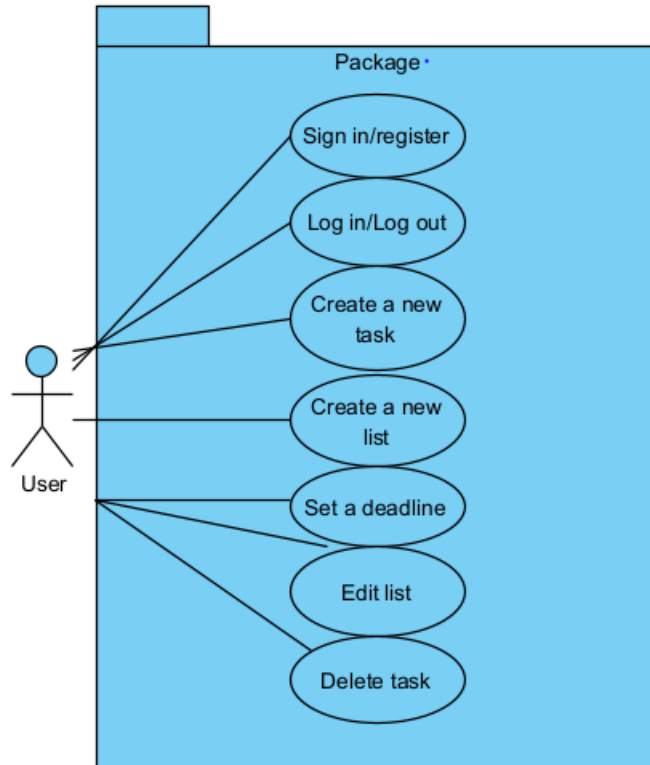
### **3.1.4 Communications Interfaces**

- **Notification Services:** The application can communicate with notification services to send push notifications to users. This can be used for task reminders, important updates or other user notifications. The communication interface ensures proper integration with notification services.
- **Cloud Storage Services:** To-Do List App can back up users' data to cloud-based storage services. This allows users to synchronize their data between different devices. The communication interface ensures that the To-Do List App can communicate with the cloud storage service securely and accurately.



## 3.2 Functional Requirements

### 3.2.1 Use Case Diagram



**3.2.1.1 Use case: Sign in/Register to system**

Use Case Name	Sign in/Register to system
Priority	Medium
Trigger	The user open the application.
Precondition	<ul style="list-style-type: none"> <li>The user has downloaded and installed the To-Do List App on their device.</li> <li>The app is accessible and running on the user's device.</li> </ul>
Basic Path	<ol style="list-style-type: none"> <li>The user launches the To-Do List App on their device.</li> <li>The app displays the login/register screen.</li> <li>If the user already has an account:</li> <li>The user enters their registered email address or username in the provided input field.</li> <li>The user enters their associated password in the password field.</li> <li>The user selects the "Sign In" button.</li> <li>If the user does not have an account:</li> <li>The user selects the "Register" or "Sign Up" option on the login/register screen.</li> <li>The app presents a registration form with required fields such as name, email address, and password.</li> <li>The user enters the necessary information in the registration form.</li> <li>The user submits the registration form.</li> </ol>
Alternate Path	<ul style="list-style-type: none"> <li>If the user decides to cancel the task creation:</li> </ul>

	<ul style="list-style-type: none"><li>• The user can select a "Cancel" or "Back" button to abort the task creation process.</li><li>• The app navigates the user back to the previous screen or section without creating the task.</li></ul>
Post-condition	The user is successfully signed in or registered and gains access to their personalized account.
Exceptional Path	<ul style="list-style-type: none"><li>• If there are technical issues or errors during the task creation process:</li><li>• The app displays an error message indicating the problem and advises the user to try again later.</li><li>• The user can choose to retry the task creation or exit the app.</li></ul>
Other	
Assumption	The user entered the valid information

**3.2.1.2 Use case: Log in/Log out the system**

Use case name	Log in/Log out the system
Priority	Medium
Trigger	The user wants to log in system or log out from the system
Precondition	User must be sign in to system first.
Basic Path	<ol style="list-style-type: none"><li>1. The user selects the "login" button in the application.</li><li>2. The user enters the e-mail and password.</li><li>3. System checks the password and e-mail.</li></ol>
Alternate Path	<ol style="list-style-type: none"><li>1. The user selects the "log out" button in the application.</li><li>2. The system log out the user until log in again.</li></ol>
Post-condition	The user selects emergency type and send call.
Exceptional Path	
Other	
Assumption	Log in and log out will be completed successfully

**3.2.1.3 Use case: Create a new task**

Use case name	Create a new task
Priority	Medium
Trigger	The user clicks "create new task" button
Precondition	<ul style="list-style-type: none"> <li>The user has successfully logged into the app.</li> <li>The user has accessed the appropriate screen or section for creating a new task.</li> </ul>
Basic Path	<ol style="list-style-type: none"> <li>The user selects the "Add Task" button or option in the app.</li> <li>The app presents a form or input fields to capture task details.</li> <li>The user enters the required information for the task, such as the task title and optional details (description, due date, priority, etc.).</li> <li>The user submits the task details.</li> <li>The app validates the entered information for completeness and consistency.</li> <li>If any errors or missing information are found, the app displays appropriate error messages and prompts the user to correct them.</li> <li>If the information is valid, the app creates a new task with the provided details.</li> <li>The app adds the new task to the user's existing to do list or creates a new list if none exists.</li> </ol>
Alternate Path	<ul style="list-style-type: none"> <li>If the user decides to cancel the task creation:</li> </ul>

	<ul style="list-style-type: none"><li>• The user can select a "Cancel" or "Back" button to abort the task creation process.</li><li>• The app navigates the user back to the previous screen or section without creating the task.</li></ul>
Post-condition	<ul style="list-style-type: none"><li>• The user has successfully logged into the app.</li><li>• The user has accessed the appropriate screen or section for creating a new task.</li></ul>
Exceptional Path	<ul style="list-style-type: none"><li>• If there are technical issues or errors during the task creation process:</li><li>• The app displays an error message indicating the problem and advises the user to try again later.</li><li>• The user can choose to retry the task creation or exit the app.</li></ul>
Other	
Assumption	The user created a valid task.

**3.2.1.4 Use case: Create a new list**

Use case name	Create a new list
Priority	Medium
Trigger	The user clicks "create new list" button.
Precondition	<ul style="list-style-type: none"> <li>• The user has successfully logged into the app.</li> <li>• The user has access to the appropriate screen or section for managing lists</li> </ul>
Basic Path	<ol style="list-style-type: none"> <li>1. The user selects the "Create New List" button or option in the app.</li> <li>2. The app presents a form or input field to capture the details of the new list.</li> <li>3. The user enters a unique name or title for the new list.</li> <li>4. The user may optionally provide a description or additional details for the list.</li> <li>5. The user submits the list details.</li> <li>6. The app validates the entered information for completeness and uniqueness.</li> <li>7. If any errors or missing information are found, the app displays appropriate error messages and prompts the user to correct them.</li> <li>8. If the information is valid, the app creates a new list with the provided details.</li> <li>9. The app adds the new list to the user's existing collection of lists.</li> <li>10. The user can start adding tasks to the newly created list or perform other actions within the app.</li> </ol>

Alternate Path	<ul style="list-style-type: none"><li>• If the user decides to cancel the list creation:</li><li>• The user can select a "Cancel" or "Back" button to abort the list creation process.</li><li>• The app navigates the user back to the previous screen or section without creating the list.</li></ul>
Post-condition	<ul style="list-style-type: none"><li>• A new list is created and added to the user's list of to do lists.</li><li>• The user can start adding tasks to the newly created list.</li></ul>
Exceptional Path	<ul style="list-style-type: none"><li>• If there are technical issues or errors during the list creation process:</li><li>• The app displays an error message indicating the problem and advises the user to try again later.</li><li>• The user can choose to retry the list creation or exit the app.</li></ul>
Other	
Assumption	The user created a valid list.



**3.2.1.5 Use case: Set task a deadline**

Use case name	Set task a deadline
Priority	Medium
Trigger	The user choose set deadline option.
Precondition	<ul style="list-style-type: none"><li>• The user has successfully logged into the app.</li><li>• The user has created or accessed a task in their to do list.</li></ul>
Basic Path	<ol style="list-style-type: none"><li>1. The user selects the task to which they want to assign a deadline.</li><li>2. The app displays the details of the selected task, including its title, description, and current status.</li><li>3. The user selects the "Set Deadline" option or an associated icon/button for the task.</li><li>4. The app presents a date and time picker or an input field to capture the desired deadline.</li><li>5. The user selects a date and, if applicable, a time for the deadline.</li><li>6. The user submits the selected deadline.</li><li>7. The app validates the entered deadline to ensure it is a future date and time.</li><li>8. If the entered deadline is invalid (e.g., past date, conflicting with other tasks), the app displays an error message and prompts the user to select a valid deadline.</li><li>9. If the entered deadline is valid, the app updates the task with the specified deadline.</li></ol>

	<p>10. The app confirms the successful assignment of the deadline and notifies the user.</p> <p>11. The user can view the updated task details, including the assigned deadline.</p>
Alternate Path	<ul style="list-style-type: none"><li>• If the user decides not to set a deadline for the task:</li><li>• The user can select a "Cancel" or "Back" button to abort the deadline assignment process.</li><li>• The app navigates the user back to the task details without assigning a deadline.</li></ul>
Post-condition	The task is updated with the specified deadline.
Exceptional Path	<ul style="list-style-type: none"><li>• If there are technical issues or errors during the deadline assignment process:</li><li>• The app displays an error message indicating the problem and advises the user to try again later.</li><li>• The user can choose to retry assigning the deadline or exit the app.</li></ul>
Other	
Assumption	The user add a deadline.

**3.2.1.6 Use case: Edit List**

Use case name	Edit list
Priority	Medium
Trigger	The user choose edit list option
Precondition	<ul style="list-style-type: none"> <li>• The user has successfully logged into the app.</li> <li>• The user has created or accessed the list they want to edit.</li> </ul>
Basic Path	<ol style="list-style-type: none"> <li>1. The user selects the list they want to edit from their list of to do lists.</li> <li>2. The app displays the details of the selected list, including its name, description, and associated tasks.</li> <li>3. The user selects the "Edit List" option or an associated icon/button for the list.</li> <li>4. The app presents a form or input fields populated with the current details of the list.</li> <li>5. The user makes the necessary changes to the list's name, description, or any other editable fields.</li> <li>6. The user submits the edited details.</li> <li>7. The app validates the entered information for completeness and consistency.</li> <li>8. If any errors or missing information are found, the app displays appropriate error messages and prompts the user to correct them.</li> <li>9. If the information is valid, the app updates the list with the edited details.</li> </ol>

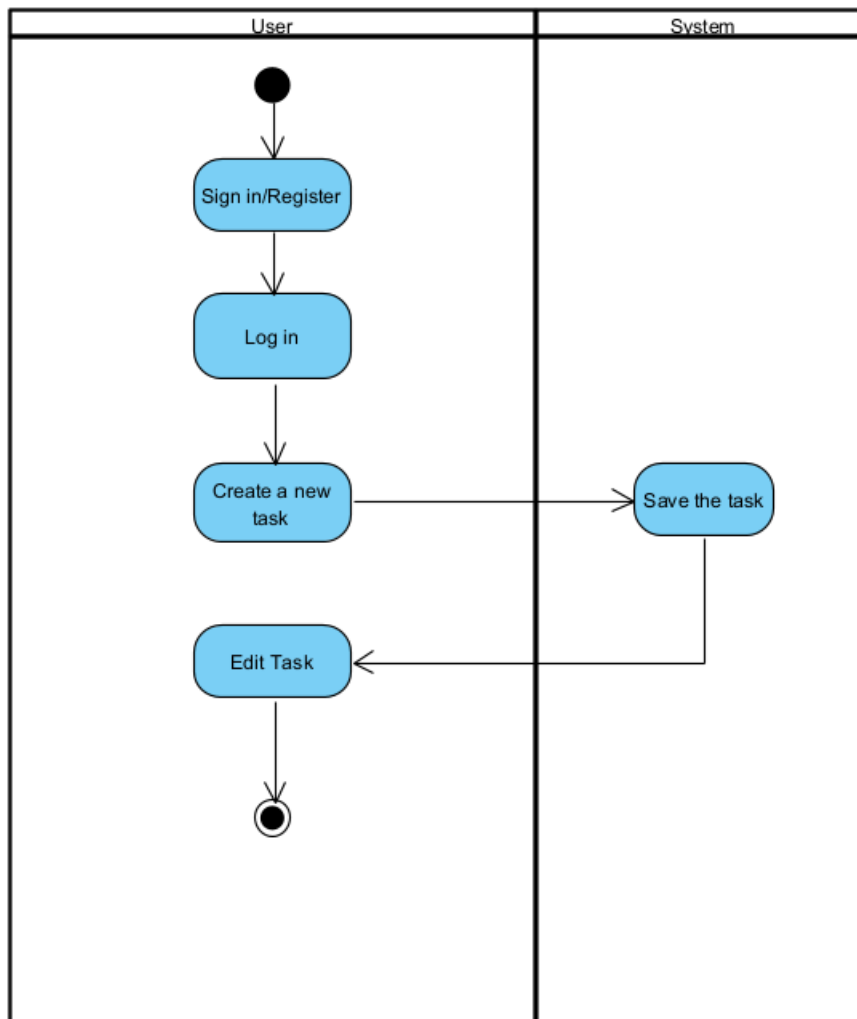
	<p>10. The app confirms the successful update of the list and notifies the user.</p> <p>11. The user can view the updated list details, including the edited information</p>
Alternate Path	<ul style="list-style-type: none"><li>• If the user decides to cancel the list editing:</li><li>• The user can select a "Cancel" or "Back" button to abort the editing process.</li><li>• The app navigates the user back to the list details without saving the changes.</li></ul>
Post-condition	The list is updated with the edited details.
Exceptional Path	<ul style="list-style-type: none"><li>• If there are technical issues or errors during the list editing process:</li><li>• The app displays an error message indicating the problem and advises the user to try again later.</li><li>• The user can choose to retry the list editing or exit the app.</li></ul>
Other	
Assumption	The user change the list.

**3.2.1.7 Use Case: Delete Task**

Use case name	Delete Task
Priority	Medium
Trigger	The user choose delete task option
Precondition	<ul style="list-style-type: none"> <li>The user must be logged in to the app.</li> <li>The user must have access to the task they want to delete.</li> </ul>
Basic Path	<ol style="list-style-type: none"> <li>The user navigates to the specific task they want to delete.</li> <li>The user selects the task and chooses the "Delete" option.</li> <li>The app prompts the user to confirm the deletion.</li> <li>The user confirms the deletion by selecting the "Yes" or "Delete" option.</li> <li>The app removes the task from the user's task list and updates the user interface accordingly.</li> <li>The app displays a confirmation message indicating that the task has been successfully deleted.</li> <li>The user can continue using the app or perform other actions.</li> </ol>
Alternate Path	
Post-condition	<ul style="list-style-type: none"> <li>The task is permanently deleted from the user's task list.</li> <li>Any associated data or attachments related to the task are also deleted.</li> </ul>
Exceptional Path	
Other	
Assumption	The user delete the task.

## 3.2.2 Activity Diagram

### 3.2.2.1



### 3.3 Detailed non-functional requirements

Attribute Name	Attribute Type	Attribute Size
LastName	String	30
FirstName	String	30
emailAddress	String	30
userPassword	String	50
username	String	30
dateOfBirth	Date	8
gender	String	varchar(30)
levelOfEducation	String	30

### 3.4 System Evolution

In the future, this system will be updated according to user’s feedback. For example for login the app, face ID can be used. In addition, maybe some AI features can be added to be helpful for disabled and old people.

## 4. Other Non-functional Requirements

### 4.1 Software Quality Attributes

- Product should have following quality attributes:
- Ability to achieve the purpose with the least effort and time,
- Easy to use,
- Easy to remember,
- Satisfying design in the first use,
- Allowing the users to take back their mistakes,
- Preventing mistakes.
- A pleasant user experience.