
Software Requirements Specification

for

Task Management Application

Version 1.0 approved

Prepared by Class 2 Group 4

14th October 2021

Table of Contents

1.	Introduction	1
1.1	Purpose	1
1.2	Document Conventions	1
1.3	Intended Audience and Reading Suggestions	1
1.4	Product Scope	1
1.5	References	2
2.	Overall Description	2
2.1	Product Perspective	2
2.2	Product Functions	2
2.3	User Classes and Characteristics	5
2.4	Operating Environment	6
2.5	Design and Implementation Constraints	6
2.6	User Documentation	6
2.7	Assumptions and Dependencies	6
3.	External Interface Requirements	7
3.1	User Interfaces	7
3.2	Hardware Interfaces	7
3.3	Software Interfaces	7
3.4	Communications Interfaces	7
4.	System Features	8
4.2	Room Management	15
4.2.1	Description and Priority	15
4.2.2	Stimulus/Response Sequences	15
4.2.3	Functional Requirements	15
4.3	Job Management	19
4.4	Task Management	22
5.	Other Nonfunctional Requirements	27
5.1	Performance Requirements	27
5.2	Safety Requirements	28
5.3	Security Requirements	28
5.4	Software Quality Attributes	28
5.5	Business Rules	29

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this project is to make task management easier for whoever uses this software. From daily tasks to organizational and team-based tasks, the user will have the options to set deadlines, organize their tasks according to priority, assign tasks to their teammates and much more. It mainly targets students and managers, team leaders who have trouble designating and managing tasks. The privacy of our users is of the highest priority as two factor authentication and Secure Sockets Layer will be used to maintain security.

1.2 Document Conventions

By default, all text is written in Arial font size 12 with headings in size 18 and subheadings in size 14. Subheadings are also in **bold** and **bold** is also used to lay emphasis on specific features. The document is divided in to 6 sections which are ranked by complexity.

1.3 Intended Audience and Reading Suggestions

Ideally, the users should know about the basics of how a task management software operates and it is better that they have used task management software.

The document is intended for:

- **Developers:** Should go through everything except section 1
- **Project Managers:** Should go through the whole document.
- **Clients:** Clients should go over all the sections, so they get an in depth understanding of the software.
- **Testers:** Should start from section 3 as the information after section 3 is relevant for them

1.4 Product Scope

The purpose of our task management application is to make life of our users easier. Tracking tasks from beginning to end, delegating subtasks to teammates, dividing a specific task and setting deadlines and progress bars to make sure the project a team is working on is submitted in time.

Our software:

- **Empowers** teams to work more productively and efficiently.
- **Task prioritization** will help our users to be sure of what tasks to do for the whole week and the sequence of these tasks.
- As all the tasks of the users, be it personal and professional will be present in the same place, **scheduling** the whole day will be easier for our user.
- **Collaboration** between the user and their team will improve as effective feedback will be shared among the members and ideas can be pooled in easily.
- **Arranging** of our tasks in different perspectives will aid the users to have an in depth understanding of the task.

Primarily, the scope pertains to the digitalization of everyday tasks and proper management of these tasks with fixed deadlines and progress bars. It is centered on users, the company or organization the users are a part of and the stakeholders of that organization.

1.5 References

Russ Miles, K. H. (2006). Learning UML 2.0 A Pragmatic Introduction to UML. O'Reilly.

Heather A. Johnson, MLIS, ude.htuomtrad@nosnhoj.a.rehtaeh, Dartmouth College, Hanover, NH

[S Dubrow](#), [DM Wallace](#) - APS Observer, 2017 - dev.psychologicalscience.org

[Software Requirements Specification document with example - Krazytech](#)

[SRSEExample-webapp.doc \(live.com\)](#)

[Example Software Requirements Specification \(SRS\) | ReqView Documentation](#)

<https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fhomeweb.csulb.edu%2F~mopkins%2Fcecs493%2FSystemRequirementsSpecificationExample.doc&wdOrigin=BROWSELINK>

2. Overall Description

2.1 Product Perspective

Our product includes a Task Management Application as requested by the client, and it is based on the combination and improvement of various existing Task Management Application features from Trello, Microsoft To Do, ToDoist, Asana and Tick Tick. The user can manage their projects and list of tasks through our Task Management interface. The user is provided with rooms to make a project and then create jobs option to create multiple sub projects inside a room and then later create tasks within a job. All the data in this application will be stored in a separate database and login authentication will also be verified through user credentials database.

2.2 Product Functions

Account Management

1- Sign-In

- Continue with Google
- Continue with Microsoft
- Continue with Apple
- With Email/Username and Password
- Remember me

- 2- Sign-Up
 - Signup with Google
 - Signup with Microsoft
 - Signup with Apple
 - Manual sign-up
- 3- Reset Password (if forgotten)
 - Email/Username
- 4- Two Factor Authentication
 - Enter OTP
 - By Email
- 5- Account Options
 - Update Username
 - Update Email
 - Change Password
 - Logout

Room Management

- 6- Create Room
 - Overall progress bar
 - Room title
 - Room description
 - Add members
- 7- Edit Room
 - Title
 - Description
 - Add/Remove members
 - Delete
 - Invite Members (Username/email)
 - Delete Members
- 8- View my account
 - Settings
 - Logout
 - Help

Job Management

9- Create Job

- Progress bar
- Title
- Description

10- Edit Jobs

- Title
- Description
- Delete

Task Management

11- Create Tasks

- Repeat tasks (daily, weekly, monthly)
- Due Date
- Description

12- Sort Tasks

- By due date
- By creation date
- By Priority
- Alphabetically

13- Edit Tasks

- Title
- Description
- Due date
- Assign members
- Remove Members
- Repeat tasks (daily, weekly, monthly)
- Delete
- Hide (Goes to Completed Task section)

14- Filter Tasks

- By due date
- By Priority
- By color

View Tasks

15- View

- Calendar view
- List view
- Board view

Admin Options

16- Roles assignment

- Admin
- Subordinate
- Member

17- Theme

- Dark mode
- Light mode

2.3 User Classes and Characteristics

The app allows for a **single user to fit in various user classes** as tasks related to projects, study, work, home etc. can be managed alongside in the software.

It is intended for everyone who would prefer to stay organized and managed.

Specific use classes include:

1. General Public (includes hobbies)

- Outdoor & indoor home chores, paying bills, buying groceries, going to the gym etc.
- Frequency of use: daily/weekly/monthly.

2. Working Professionals

- Meeting reminders, assigning work, taking updates etc.
- Frequency of use: weekly/monthly.

3. Academics

- Set deadlines for research papers, assign work etc.
- Frequency of use: monthly.

4. Students

- Home assignment deadlines, group projects, exam prep etc.
- Frequency of use: weekly/monthly.

5. Employees

- Project deadlines, due date for submitting reports etc.
- Frequency of use: weekly/monthly.

2.4 Operating Environment

- **Mobile application** supports **Android & IOS**.
- Minimum android requirement: **5.0 Lollipop**.
- Minimum IOS requirement: **IOS 11**.

- **Desktop application** supports **MAC & Windows**.
- Minimum Windows requirement: **Windows 7**.
- Minimum MAC requirement: **macOS 10.15 Catalina**

- **Database:** TBD

2.5 Design and Implementation Constraints

- The information of the user and tasks will be stored in a database which will be accessible by the Task Management Application
- Oracle Database and Java will be used for database management.
- The Task Management System should be running 24 hours a day.
- Users may access the task management application through mobile phones or desktop PC that has internet browsing capabilities and internet connection.
- User must have their correct usernames and passwords to enter their task management application.
- All interfaces and data inside the application will be written in English Language.
- Minimizing memory and processing power usage by optimizing the data inside the application and reducing redundancies through database to make the application lightweight and easy to use.
- Ensuring that all the code and functions are error and bug free.

2.6 User Documentation

A complete user manual will be provided with more in-depth information and instructions. Along with that we will provide online instruction videos available on YouTube explaining the functions of our Task Management application.

2.7 Assumptions and Dependencies

The software has several features that are dependent on other systems:

Login & Signup:

Links with **Facebook, Google & Apple** requires their services to be functioning.

Two Factor Authentication:

Active cellular service that can deliver OTP to all global country codes.

Email Service:

We assume that the email application of Google and other services that user uses to sign up are working perfectly.

User Expertise:

We assume that the users have a basic understanding of using computer and mobile phone apps.

User Internet Connection:

We assume that user has an **adequate internet connection** because database and interface need to communicate over the internet.

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

This feature allows for registering a new account through various channels, logging into the software via an existing account and the option to reset your password (if forgotten). Two factor authentication is also built in the login process.

This feature is of **High priority**.

4.1.2 Stimulus/Response Sequences

User clicks on the sign-up button -> sign up page loads displaying multiple sign-up options

User either:

- **Fills in required information** -> system verifies entered email address and proceeds to successful sign-up page. Gives an error message if an account with the entered email address already exists.
- **Chooses to register through Google, Microsoft or Apple** -> system proceeds to successful sign up page

User opens the application -> the default sign-in page loads up

User either:

- **Fills in login credentials** -> system verifies entered credentials and proceeds to application home page. Gives an error message if credentials are not valid.
- **Chooses to login through Google, Microsoft, or Apple** -> system loads application home page. If no account is associated, then creates an account automatically.
- **User tick marks 'remember me' option or leaves it blank** -> system reads input and stores sign-in information in local application if option tick marked.

User logs in with difference device -> system loads two-factor authentication page and sends OTP (one time passcode) to registered phone number against user account.

User either:

- **Enters OTP** -> system proceeds to application home page if OTP is correct otherwise returns an error message.
- **Clicks on 'don't have phone; use alternate email' option** -> system sends email on alternate email address. Link has to be clicked for access.

User clicks on 'Forgot Password' option -> system loads new window asking for username or email address.

User enters username or email address -> system verifies if username or email address exists in accounts database. System sends reset password email if account exists otherwise returns an error message.

User clicks on logout button -> system saves current application state to the database and logs out the user

User updates account details -> system updates database with relevant details

4.1.3 Functional Requirements

REQ-1: Manual Sign-in
REQ-2: Sign in with Google
REQ-3: Sign in with Microsoft
REQ-4: Sign in with Apple
REQ-5: Remember me (check field)

REQ-6: Manual Sign-up
REQ-7: Sign up with Google
REQ-8: Sign up with Microsoft
REQ-9: Sign up with Apple

REQ-10: Option to enter email/username for password reset (if password forgotten)
REQ-11: Option to enter OTP or alternate email for Two Factor Authentication

REQ-12: Option to log out from account
REQ-13: Option to update username
REQ-14: Option to update email address
REQ-15: Option to change password

USE CASES

Use case name		Sign up
Related requirements		REQ-6 REQ-7 REQ-8 REQ-9
Goal in context		To create a new user account
Preconditions		1. User should be on sign-up page 2. Username & email must be unique. 3. Password meets the system's criteria. 4. User must fill in all fields 5. Phone Number must be unique
Successful end condition		The user account is successfully created and is redirected to the login page.
Failed end condition		New user account is not created if any one of the pre-conditions is not met.
Primary actors		User
Secondary actors		None
Trigger		User taps on sign up button
Included cases		Sign-up Verification
Main flow	Step	Action
	1	User loads the sign-up page which has the sign-up form to be filled manually or user chooses to sign up through Google, Microsoft, Apple.
	2	User fills in the required details.
	3	

	Include: Verification Check	System verifies the entered email address and username. Both should be unique.
	4	A new user account is created.
	5	Login page is returned.

Use case name		Log-in
Related requirements		REQ-1 REQ-2 REQ-3 REQ-4 REQ-5
Goal in context		Existing User can log in and access the application
Preconditions		User must exist.
Successful end condition		User successfully logs in to the account and main page is shown.
Failed end condition		The log in attempt is failed due incorrect login credentials.
Primary actors		User.
Secondary actors		Task management database.
trigger		Existing user taps on the log in button.
Included cases		Check Identity.
Main flow	Step	Action
	1	An existing user loads log-in page; the login form is displayed.
	2	User enters the credentials and clicks on login button.
	3 Include: Verification Check	Credentials are verified from the database.
	4	User is logged in to his account and main page is returned giving user access to the application.

Use case name		Log out
Related requirements		REQ-12
Goal in context		A logged-in user is able to log out from the system.
Preconditions		User must be logged in.
Successful end condition		User is logged out of his account.
Failed end condition		User is unable to log out.
Primary actors		User.
Secondary actors		None.
Trigger		Logged in user taps on logout button.
Included cases		None.
Main flow	Step	Action
	1	A logged in user taps on logout button.
	2	Any changes or details are stored in the application database.
	3	The user is logged out, and log in form is returned.

Use case name		Forgot Password
Related requirements		REQ-10
Goal in context		Password is reset successfully
Preconditions		User account must exist.
Successful end condition		A password reset link is sent via email.
Failed end condition		User is unable to reset his or her password.
Primary actors		User
Secondary actors		Null
trigger		User taps on the forgot password button.
Included cases		Verification check
Main flow	Step	Action
	1	User taps on the forgot password button.
	2	The system prompts the user to enter email address or username used for their existing account.
	3	User enters the email address or username
	4 Include: Verification Check	Email address is verified by the database.
	5	If account exists, an reset password email is sent otherwise an error message is displayed.
	6	User enters the new credentials and is returned to the login page of the application.

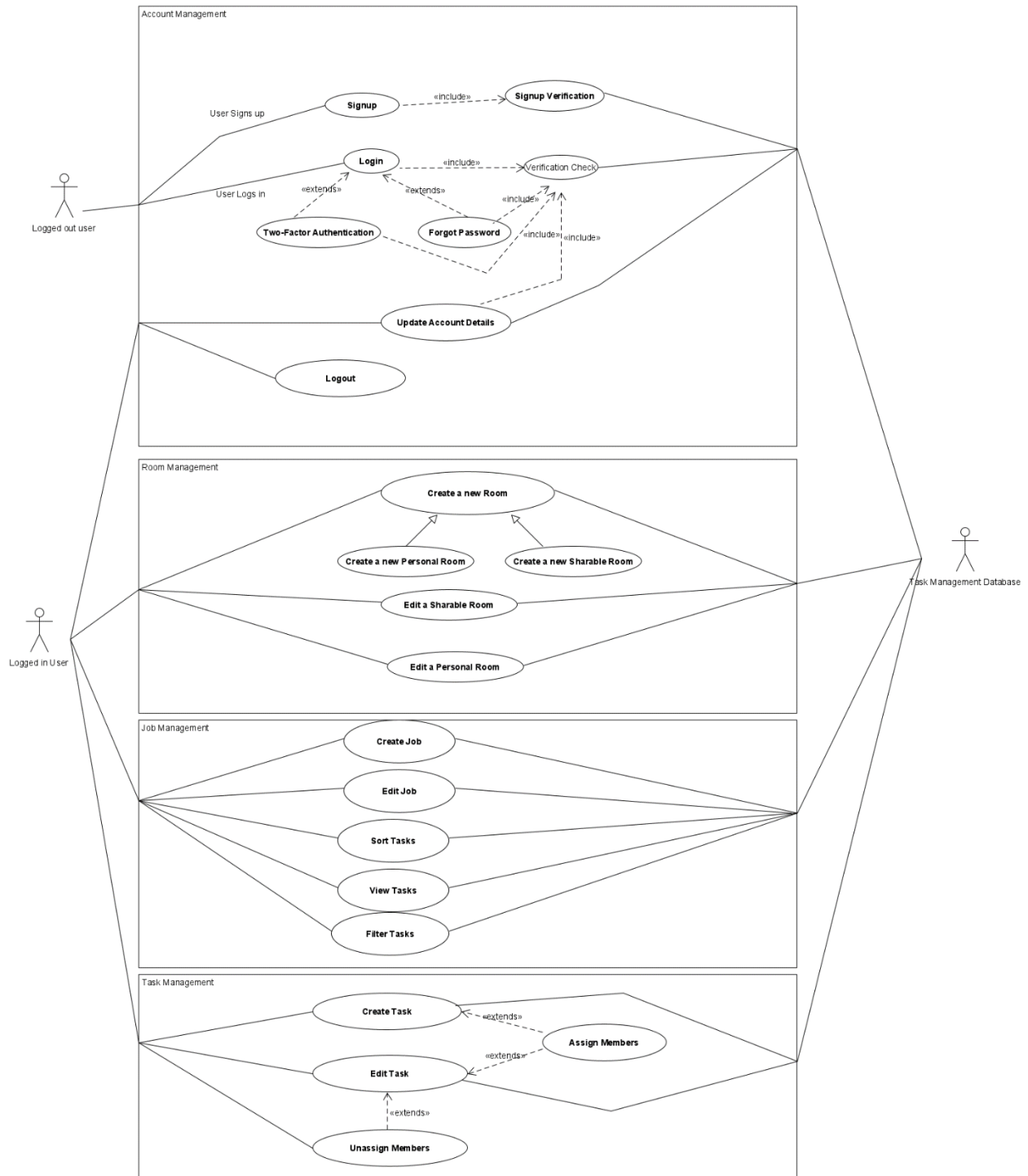
Use case name		Two Factor Authentication
Related requirements		REQ-11
Goal in context		User is successfully logged in after additional verification.
Preconditions		User phone number or alternate email must exist.
Successful end condition		User is logged in to the application
Failed end condition		User is unable to verify through phone number or alternate email address.
Primary actors		User
Secondary actors		Null
trigger		User taps on the login button. System redirects to Two Factor Authentication if login is from new device.
Included cases		Verification check
Main flow	Step	Action
	1	User taps on the login button with a new device.
	2	The system prompts the user to choose verification through email address or phone number.
	3	User chooses between phone number & email address.
	4	User enters OTP.
	5 Include: Verification Check	OTP is verified by the Database
	6	User successfully logs in

Use case name		Verification Check
Related requirements		REQ-6 REQ-7 REQ-8 REQ-9
Goal in context		Entered details must be checked and verified as accurate
Preconditions		Appropriate request by the system
Successful end condition		The details are verified .
Failed end condition		The details are not verified.
Primary actors		User
Secondary actors		Task Management Database
trigger		Relevant details are provided for verification.
Included cases		none
Main flow	Step	Action
	1	The details are provided to the System
	2	The database verifies the details.
	3	The details are returned as verified

Extensions	5	The databased does not verify the details.
	6	The details are returned as unverified

Use case name		Update Account Details
Related requirements		REQ-13 REQ-14 REQ-15
Goal in context		User changes account details: username or email or password
Preconditions		User must be logged in.
Successful end condition		All changes are incorporated.
Failed end condition		Changes are not made.
Primary actors		User
Secondary actors		Task Management Database
trigger		User clicks update button.
Included cases		Verification Check
Main flow	Step	Action
	1	User enters updated information in relevant fields.
	2 Include: Verification Check	The database verifies the email or username for uniqueness.
	3	On successful verification, changes are incorporated else an error message is displayed

USE CASE DIAGRAM



4.2 Room Management

4.2.1 Description and Priority

This feature allows user to create a room from his or her account. The user can either create a personal room which only they can access or a shared room in which they can add more members. The user can set the attributes of the room according to themselves and have the option of adding members and assigning them roles. The features of the room can also be edited, and members can be removed as well. This feature is of **High Priority**.

4.2.2 Stimulus/Response Sequences

User clicks on the “Create room” button – A new displays various options of setting attributes.

1. System verifies that the room name is unique. It gives an error message if the name is not unique. The user will enter the room description and the system will check it.
2. The user selects the type of room “shareable” or “private”.
3. If the type is shareable, the user can add and remove members in the room and assign them roles.
4. If the type is private, the user cannot add or remove members.
5. User clicks “save” and the room is created.
6. An overall progress bar is created for the room.

User clicks on “edit room” – The edit room page loads up.

User can:

1. Edit the title of the room
2. Edit the description of the room
3. Add or remove members in the room if the room is shareable
4. Delete the room
5. Invite new members in the room via username if the room is shareable
6. System checks if all the details inputted by the user are unique and then saves the changes when the “save” button is clicked.

4.2.3 Functional Requirements

- REQ-1: User should be able to set a unique title for the room.
- REQ-2: User can select room type.
- REQ-3: User can set a description for the room to show its purpose
- REQ-4: An overall progress bar is created and is set to 0%.
- REQ-5: The room is not created if there already exists a room with the same title.
- REQ-6: User should be able to set a unique title for the room.
- REQ-7: User will select **Shareable** room type.
- REQ-8: User can set a description for the room to show its purpose
- REQ-9: User can add members in the room to assign tasks.

REQ-10: User can assign roles to the members to be added in the room.
 REQ-11: An overall progress bar is created and is set to 0%.
 REQ-12: The room is not created if there already exists a room with the same title.

REQ-13: User should be able to set a unique title for the room.
 REQ-14: User will select **Private** room type.
 REQ-15: User can set a description for the room to show its purpose
 REQ-16: An overall progress bar is created and is set to 0%.
 REQ-17: The room is not created if there already exists a room with the same title.

REQ-18: User should be able to edit the title of the **Shareable** room.
 REQ-19: User can edit the description of the room.
 REQ-20: User can invite members in the room to through their username.
 REQ-21: User can add and remove members from the room.
 REQ-22: User can assign role to the members in the room.
 REQ-23: User can delete the room.
 REQ-24: The room is not edited if there already exists a room with the same title.

REQ-25: User should be able to edit the title of the **Private** room.
 REQ-26: User can edit the description of the room.
 REQ-27: User can delete the **Private** room.
 REQ-28: The room is not edited if there already exists a room with the same title.

USE CASES

Use Case Name	Create room	
Related Requirements	REQ1, REQ2, REQ3, REQ4, REQ5	
Goal in context	User asks to create a new room from the system.	
Preconditions	User must be logged in and must have an account.	
Successful end condition	A new Room is created.	
Failed end condition	The request to create a new room is rejected.	
Primary Actor	User	
Secondary Actor	None.	
Trigger	User clicks on “ Create room ” button.	
Included Cases	None	
Main flow	Step	Actions
	1	User clicks on the Create room button.
	2	User selects room type .
	3	User will enter all the details.
	4	User clicks on “ save ” button.
	5	The details of the room are checked.
	6	A new room is created.
	7	Progress bar is created.
Extensions	Step	Branching Action
	1	The user is not allowed to create a new room.
	2	A room with the same title already exists.
	3	The user is prompted on this rejection.

Use Case Name	Create Shareable room
---------------	-----------------------

Related Requirements	REQ6, REQ7, REQ8, REQ9, REQ10, REQ11, REQ12	
Goal in context	A user requests to create a new Shareable room from the system.	
Preconditions	User must be logged in and must have an account.	
Successful end condition	A new Shareable Room is created.	
Failed end condition	The request to create a new Shareable room is rejected.	
Primary Actor	User	
Secondary Actor	None	
Trigger	User asks the system to create a new Shareable room.	
Included Cases	Create Room	
Main flow	Step	Actions
	1	User clicks on the Create room button.
	2	User selects the “ Shareable room ” type.
	3	User enters title of the room and description.
	4	User adds or invites members to be a part of the room and assigns roles to them.
	5	User clicks on save.
	6	The details of the room are checked
	7	A new shareable room is created.
	8	Progress bar is created.
Extensions	Step	Branching Action
	1	The user is not allowed to create a new Shareable room.
	2	A room with the same title already exists.
	3	The user is prompted on this rejection.

Use Case Name	Create Private room	
Related Requirements	REQ13, REQ14, REQ15, REQ16, REQ17	
Goal in context	User asks to create a private room from the system.	
Preconditions	User must be logged in and must have an account.	
Successful end condition	A new Private room is created.	
Failed end condition	The request to create a private room is rejected.	
Primary Actor	User	
Secondary Actor	None	
Trigger	User asks the system to create a new Private room .	
Included Cases	Create Room	
Main flow	Step	Actions
	1	User clicks on the Create room button
	2	User selects the Private room type.
	3	User will enter title of the room and description.
	3	User clicks on “save” button.
	4	The details of the room are checked.
	5	A new private room is created.
	6	Progress bar is created.
Extensions	Step	Branching Action
	1	The user is not allowed to create a new Private room.
	2	A room with the same title already exists.
	3	The user is prompted on this rejection.

Use case name	Edit Shareable room	
Related requirements	REQ18, REQ19, REQ20, REQ21, REQ22, REQ23, REQ24	
Goal in context	User asks to edit the contents of a shareable room from the system.	
Preconditions	User must be logged in. The room must exist. User must have admin or subordinate access of the room.	
Successful end condition	The contents of the room are edited including adding and deleting members and changing their roles.	
Failed end condition	The request of editing room title is rejected.	
Primary Actor	User	
Secondary Actor	None	
Trigger	User clicks on the “edit room” button	
Included Cases	None	
Main flow	Step	Action
	1	Admin clicks the “ edit room ” button.
	3	Admin edits the attributes of the room.
	4	Admin adds/invites or remove members.
	5	Admin changes the role of the members.
	6	Admin presses “save” button.
	7	The system checks the details and changes are saved.
Extensions	Step	Branching Action
	1	The user is not allowed to edit a Shareable room.
	2	A room with the same title already exists.
	3	The user is prompted on this rejection.

Use case name	Edit Private room	
Related requirements	REQ25, REQ26, REQ27, REQ28	
Goal in context	User asks to edit the contents of a private room from the system.	
Preconditions	User must be logged in. The room must exist. User must be the owner of the room.	
Successful end condition	The contents of the room are edited.	
Failed end condition	The request of editing room title is rejected.	
Primary Actor	User	
Secondary Actor	None	
Trigger	User clicks on the “edit room” button.	
Included Cases	None	
Main flow	Step	Action
	1	Admin clicks the “edit room” button
	3	Admin edits the attributes of the room
	6	Admin presses “save” button.
	7	The system checks the details and changes are saved.
Extensions	Step	Branching Action
	1	The user is not allowed to edit a private room.
	2	A room with the same title already exists.

	3	The user is prompted on this rejection.
--	---	---

4.3 Job Management

4.3.1 Description and Priority

This feature allows user with admin or subordinate access to create a job from his or her account. The user can create multiple jobs within a room and set the attributes of the jobs according to themselves. The features of the job can also be edited, and the admin or subordinate can add/remove tasks within a job.

4.3.2 Stimulus/Response Sequences

- User clicks on the “Create Job” button – A new displays various options of setting attributes.
- System verifies that the job name is unique. It gives an error message if the name is not unique else the user will then be able to enter the job description.
- An overall progress bar is created for the job.
- User clicks on “edit job” – The edit job page loads up.

Admin or Subordinate user can:

- Edit the title of the job
- Edit the description of the job
- Delete the job
- System checks if all the details entered by the user are unique and then saves the changes when the “save” button is clicked.

The users can:

- Click on sort tasks within a job.
- The user will be shown with 4 methods of sorting.
- The user can sort tasks according to priority, creation date, due date and alphabetically.

4.3.3 Functional Requirements

Create Job:

REQ-1: An Admin or Subordinate within a room can create a job

REQ-2: An Admin or Subordinate within a room should be able to set a unique title for the job.

REQ-3: User can set a description for the job to show its purpose

REQ-4: An overall progress bar is created and is set to 0%.

REQ-5: The job is not created if the title is not unique.

Edit room:

REQ-6: An Admin or Subordinate within a room can edit job title.

REQ-7: An Admin or Subordinate within a room can edit job description.

REQ-8: An Admin or Subordinate can delete the job.

REQ-9: The job is not edited if the job title already exists.

Sort Tasks:

REQ-10: A user within a job can sort the list of tasks.

REQ-11: A user can sort tasks alphabetically

REQ-12: A user can sort tasks according to priority

REQ-13: A user can sort tasks according to due date

REQ-14: A user can sort tasks according to creation date.

View Tasks:

REQ-15: A user within a job can view the list of tasks.

REQ-16: The user can view tasks in calendar view

REQ-17: The user can view tasks in list view.

USE CASES

Use case name	Create Job(s)
Related Requirements	REQ-1, REQ-2, REQ-3, REQ-4, REQ-5
Goal in context	<ul style="list-style-type: none">• To create a job which has a unique name, a concise and straightforward description.• To create a job progress bar.
Preconditions	<ul style="list-style-type: none">• A room is already created within which a new job will be created.• The user has admin or subordinate access within the room
Successful end condition	<ul style="list-style-type: none">• A new job is created within a room• A progress bar for job is created.
Failed end condition	<ul style="list-style-type: none">• The request for new job is rejected because another job with the same name already exists within that room.
Primary Actors	Room Admin and Room Subordinate
Secondary Actors	None
Trigger	The Admin or Subordinate asks the TMS to create a new job.
Included Cases	None
Main Flow	Step 1: The Admin or Subordinate clicks to create a new job. Step 2: The create job interface will be visible Step 3: The Admin or Subordinate enters the title and details of the new job. Step 4: The TMS checks if a job with the given name already exists within the room. Step 5: If the job name is unique then a new job is created. Step 6: The job progress bar is shown with 0%.

Use case name	Edit Job(s)
Related Requirements	REQ-6, REQ-7, REQ-8, REQ-9
Goal in context	To edit existing job details.
Preconditions	<ul style="list-style-type: none"> • A room is already created within which a job to be edited exists. • The user has admin or subordinate access within the room.
Successful end condition	An existing job is edited within a room
Failed end condition	The request for editing a job is rejected because another job with the edited name already exists within that room.
Primary Actors	Room Admin and Room Subordinate
Secondary Actors	None
Trigger	The Admin or Subordinate asks the Task Management System to edit an existing job.
Included Cases	None
Main Flow	<p>Step 1: The Admin or Subordinate clicks to edit an existing job.</p> <p>Step 2: Edit job interface will be visible.</p> <p>Step 3: The Admin or Subordinate edits the title and details of the existing job.</p> <p>Step 4: The TMS checks if a job with the edited name already exists within the room.</p> <p>Step 5: If the job name is unique then the job is edited successfully.</p>

Use case name	Sort Task(s)
Related Requirements	REQ-10, REQ-11, REQ-12, REQ-13, REQ-14
Goal in context	The user requests to sort the existing list of tasks.
Preconditions	A room is already created within which a job and list of tasks exist.
Successful end condition	The list of tasks is sorted according to the method.
Failed end condition	None
Primary Actors	All users with a list of tasks.
Secondary Actors	None
Trigger	The user asks the Task Management System to sort the list of the tasks within a job.
Included Cases	None

Main Flow	<p>Step 1: The user clicks to on sort tasks.</p> <p>Step 2: The user is prompted with a list of sorting methods.</p> <p>Step 3: The user selects a particular method.</p> <p>Step 4: The TMS sorts the list of tasks according to the desired method and shows the sorted list.</p>
-----------	---

Use case name	View Task(s)
Related Requirements	REQ-15, REQ-16, REQ-17
Goal in context	The user can view the existing list of tasks.
Preconditions	A room is already created within which a job and list of tasks exist.
Successful end condition	The list of tasks appears within a job.
Failed end condition	None
Primary Actors	All users with a list of tasks.
Secondary Actors	None
Trigger	When user opens a job they are shown with an interface showing list of tasks.
Included Cases	None
Main Flow	<p>Step 1: The user clicks to on job</p> <p>Step 2: The TMS shows the list of tasks according to the default sorting by due date.</p>

4.4 Task Management

CREATE TASK

4.1.1 Description and Priority

- This feature is used to create a task that is requested by the user for different purposes.
- At the time of creation of this task, the user identifies some details like the title of the task, the description of the task and the due date of the task.
- Priority: HIGH
- Reason: This is a HIGH priority feature as without the creation of task, the main purpose of the application is not fulfilled.

4.1.2 Stimulus/Response Sequences

- As the user clicks on the create task feature, the system will generate some fields that the user must fill in order to proceed like title of the task, description of the task and the due date of the task.
- After filling these details, the system gives the user a choice whether he wants to assign members to the task (applicable in shareable workspace only).
- After this step, the system creates a new task in the database that is ready to serve its purpose.

4.1.3 Functional Requirements

REQ-1: The system will allow the users to add new tasks

REQ-2: The system will allow the users to set a title of the task

REQ-3: The system will allow the users to set the description of each task

REQ-4: The system will allow the users to set a due date of each task

REQ-5: The system will allow the users to set priority (high, medium, low) of each task

EDIT TASK

4.2.1 Description and Priority

- This feature is used to edit the already created task. A need may arise for user to edit certain details such as changing title, edit description, and change the due date of the task.
- Priority: LOW
- This is LOW priority feature since user is not necessarily required to use this feature. For example, if everything goes correctly, the user may not edit anything at all.

4.2.2 Stimulus/Response Sequences

- As the user clicks on the edit task feature, the system will offer some options to user such as if he wants to edit title, he may click on 'Edit Title' and further options include 'Edit description' and 'edit due date'.
- After choosing desired option, the user will perform the editing and system will check if these editing conforms to the constraints.
- Once the editing is complete, the system updates the edited task in the database that is ready to serve its purpose.

4.2.3 Functional Requirements

REQ-1: The system will allow the users to edit tasks

REQ-2: The system will allow the users to edit the title of the task

REQ-3: The system will allow the users to edit the description of each task

REQ-4: The system will allow the users to change the due date of each task

ASSIGN MEMBERS

4.3.1 Description and Priority

- This feature is used to assign the members to the task.
- Priority: HIGH

- In sharable workspace, the priority is HIGH since there must be at least one member assigned to do the task otherwise the concept that members can share a task will lose its credibility.
- In personal workspace, user cannot assign members.

4.3.2 Stimulus/Response Sequences

- The user will click on the option of 'Assign members' and system will present list of members available that were previously added when the room was being created by the user.
- The user will select the required members.
- The system will assign the selected members to the workspace.

4.3.3 Functional Requirements

REQ-1: The system will allow the users to assign member to his workspace.

UNASSIGN MEMBERS

4.3.1 Description and Priority

- This feature is used to unassign the members to the task.
- Priority: LOW
- In sharable workspace, the priority is LOW since user may not unassign members if he assigned them correctly or it is optional.

4.3.2 Stimulus/Response Sequences

- The user will click on the option of 'Unassign members'
- A list will appear in which we can see the members that are assigned to any task
- The user will select the members that he wants to unassign from any task
- The system will unassign the selected members.

4.3.3 Functional Requirements

REQ-1: The system will allow the users to unassign member to his workspace.

FILTER TASKS

4.3.1 Description and Priority

- This feature is used to filter the tasks based on a property (due date, color or priority)
- Priority: LOW
- The reason for low priority is because the user may not filter the tasks. For ex: it is not necessary to filter if the number of tasks are manageable.

4.3.2 Stimulus/Response Sequences

- The user will click the FILTER TASK button
- A screen will appear from which the user has to select the property against which the tasks are supposed to be filtered.
- After selection of the property, the user will click on APPLY FILTER button so that the command is transferred to the system to filter the task
- The system filters the task and display them according to the property selected.

4.3.3 Functional Requirements

REQ-1: The system will allow the users to filter the task.

REQ-2: The system will allow the users to select a property for the filter task feature

REQ-3: The system will not allow the users to filter the task if the selected property is unavailable

USE CASES

Use Case name	Create Task	
Related Requirements		
Goal in Context	The user creates a new task	
Preconditions	Already logged in, Room already created, Job already created	
Successful End condition	A new task is created for the users	
Failed End condition	The request to create a new task is rejected	
Primary Actors	User	
Secondary Actors		
Trigger	The user asks the system to create a new task	
Included Cases	TBD	
Main Flow	1	The user asks the system to create a task by providing details like task name, description, due date and priority
	2	The system checks whether the details conform to the constraints or not
	3	A new task is created

Use Case name	Edit Task	
Related Requirements		
Goal in Context	The user wants to edit an already created task	
Preconditions	Already logged in Room already created Job already created Task already created	
Successful End condition	Editing of task is completed successfully	
Failed End condition	Failed to edit the task	
Primary Actors	User	
Secondary Actors		
Trigger	The user asks the system to edit a task	
Included Cases	TBD	
Main Flow	1	The user asks the system to edit a task
	2	The system checks whether the task is present in the database or not
	3	The system checks whether the field that needs to be edited is valid or not

	4	The editing of task is completed by the system

Use Case name	Assign Members	
Related Requirements	Create Task Edit Task	
Goal in Context	The user wants to assign members to any specific task	
Preconditions	Already logged in Room already created Job already created Task already created	
Successful End condition	Members are assigned to a task	
Failed End condition	Failed to assign a member to a task	
Primary Actors	User	
Secondary Actors		
Trigger	The user asks the system to assign members to a task	
Included Cases	TBD	
Main Flow	1	The user asks the system to assign members to a task
	2	The system checks the type of workspace
	3	The system checks the name of the member that needs to be assigned
	4	The system has assigned members to the task depending upon the type of workspace

Use Case name	Unassign Members	
Related Requirements	Edit Task	
Goal in Context	The user wants to Unassign members to any specific task	
Preconditions	Already logged in Room already created Job already created Task already created	
Successful End condition	Members are Unassigned from any task	
Failed End condition	Failed to Unassign a member from a task	
Primary Actors	User	
Secondary Actors		
Trigger	The user asks the system to Unassign members to a task	
Included Cases	TBD	
Main Flow	1	The user asks the system to Unassign members to a task
	2	The system checks the type of workspace
	3	The system checks the name of the member that needs to be removed

	4	The system has unassigned the member from the task depending upon the type of workspace

Use Case name	Filter Task	
Related Requirements		
Goal in Context	The user wants to filter the available tasks by due date, color or priority	
Preconditions	Already logged in Room already created Job already created Task already created A task must have a due date or a priority	
Successful End condition	The tasks are successfully filtered based on due date, color or priority.	
Failed End condition	The task is not filtered based on due date, color or priority.	
Primary Actors	User	
Secondary Actors	-	
Trigger	The user asks the system to filter the tasks	
Included Cases	TBD	
Main Flow	1	The user first selects the property from which the tasks are supposed to be filtered
	2	The user asks the system to filter the tasks based on the selected property (due date, color, priority)
	3	The system checks whether the tasks contain that property or not
	4	The system filters the task successfully

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must not delay the basic operations which includes:

- The system shall load and update User Interface quickly.
 - Switching between forms shall be very quickly (< 1 or 2 sec)
- Inter application communication:
 - System shall facilitate inter application communication between users with no delay.
- Login verification:
 1. System shall process all the relevant information required for login verification in a very short time duration. (< 2 seconds)
 2. In case of logging in with other applications such as Google, the system shall interact with relevant server without any observable delay. (< 3 seconds)

3. If user has forgotten his password, the system must send email in less than 2 to 3 seconds.
- Two Factor Authentication:
 4. System shall send OTP to user's phone number within the given time period.
 5. In case of email verification, the System shall send the verification link within the given time.

5.2 Safety Requirements

- The system shall be equipped with Exception Handling techniques to cope up with exceptions that might cause system to crash unexpectedly.
- To ensure the proper use of system, system shall have **help** option to apprise the users about using product safely.
- There should backup facility in case of database failure.
- Copies of data are being made on regular basis

5.3 Security Requirements

- Privacy between users.
No users can gain an improper access to another's profile.
- Strong password is required from the user with minimum 8 character, one uppercase letter, one special symbol and one numeric character.
- 3 wrong attempts at logging in with wrong password will disable the login for that user.
- In case of logging in from different device, the system will perform **two factor authentication**.
OTP will be sent to the user on his mobile phone or email address.
- The system shall automatically log out all customers after a period of inactivity.
- The system shall not leave any cookies on the customer's computer containing the user's password.
- The system shall not leave any cookies on the customer's computer containing any of the user's confidential information.
- The system will be closed source.

5.4 Software Quality Attributes

Usability:

- The system will have user friendly User Interface such that anyone with basic computing sense can use it.
- The UI will be clearly labelled, easily navigable and there will be a **help** option to guide the user if he face any difficulty in usage.

Portability:

- The task management system can be used on any system that supports Java Virtual Machine.
- The system is able to run on different platforms such as Windows and IOS.

Availability:

- The system is available to offer services whenever requested.

Maintainability:

- The system will be open to future upgradation
- Coding maintainability:
 - Variables will be meaningful.
 - There will be comments describing the functionality of different subroutines and variables.

Interoperability:

- The system will share data with database to store relevant information such as login info.
- The system can work with other applications such as Google and Apple during sign up process.

5.5 Business Rules

- All users need to have valid email address to be able to use this application.
- Users must know their username and password to log in.
- The administrator will have the absolute authority over other users (subordinate and member)
- Only administrator can create a room.
- To create jobs, room should be created first.
- To create tasks, job should be created inside room first.
- Administrator and subordinate can create jobs and create tasks.
- Administrator and subordinate can assign and unassign members to task.
- The user should be notified through notification bar about task completion.
- User can add and remove members to his room

6. Other Requirements

Glossary

Preconditions

Pre-condition is a statement or set of statements that outline a condition that should be true when an action is called. The precondition statement indicates what must be true before the function is called.

Goal In Context goal of the use case in context of the scope and level.

Trigger This is the event that causes the use case to be initiated

Primary Actors The primary actor of a use case is the stakeholder that calls on the system to deliver one of its services.

Secondary Actors

A secondary actor is one from which the system requires assistance to complete the use case.

Stimulus

Enables system architects to simulate requirements and to detect ambiguous, incorrect, missing, or conflicting requirements

Response Sequence Diagram

A sequence diagram is structured in such a way that it represents a timeline which begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows.

TMS Task Management System

OTP One Time Passcode

Cookies Track, personalize, and save information about each user's session.

User Classes User classes will differ based on the type of software you are writing, and the audience it is designed for.

Product perspective

Product perspective defines how the product contributes to fulfilling stakeholder needs and adjacent systems assumptions

Product scope Describe the software being specified. And include benefits, objectives, and goals.

Communication Interfaces Designed to a specific standard, that enables one machine to telecommunicate with another machine.

Software Quality Attributes

Describes the characteristics of the product such as size, complexity, design features, performance, and quality level.

Performance Requirement

Performance requirements typically comprise a set of criteria which stipulate how things should perform or the standards that they must achieve in a specific set of circumstances.

External Interfaces

External interfaces are typically a product's lifeline to the outside world. Such interfaces may be used for several purposes, including connecting to peripherals, field programming, or testing during product manufacturing.

Interoperability

It is the ability of different systems, devices, applications, or products to connect and communicate in a coordinated way, without effort from the end user.

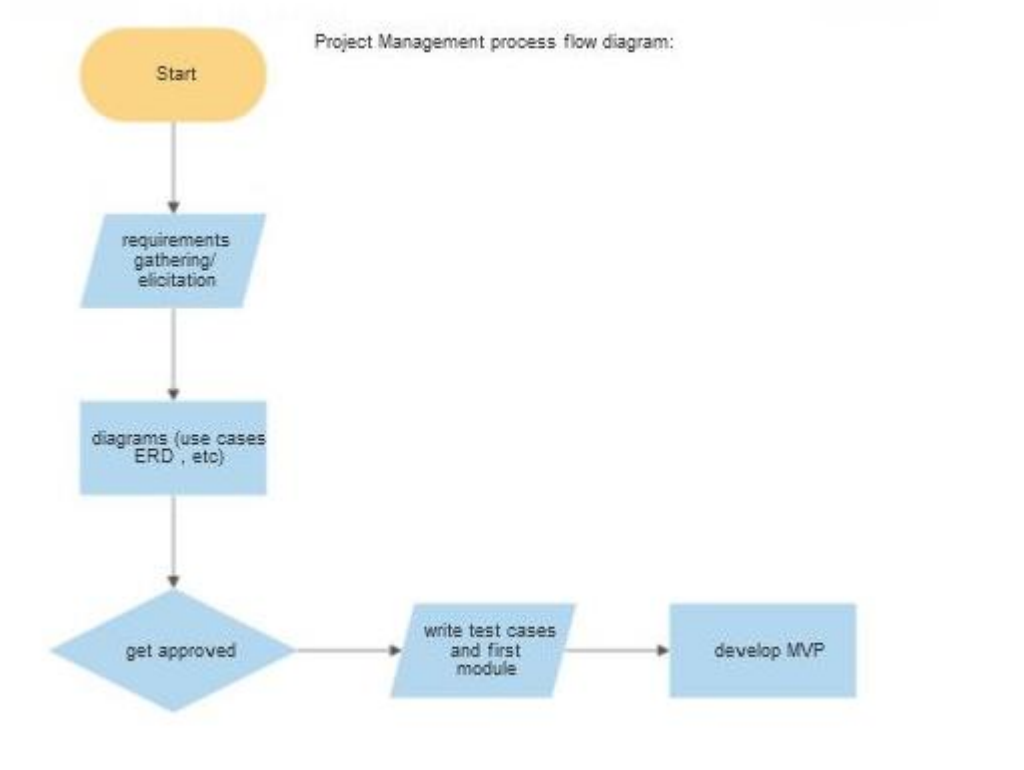
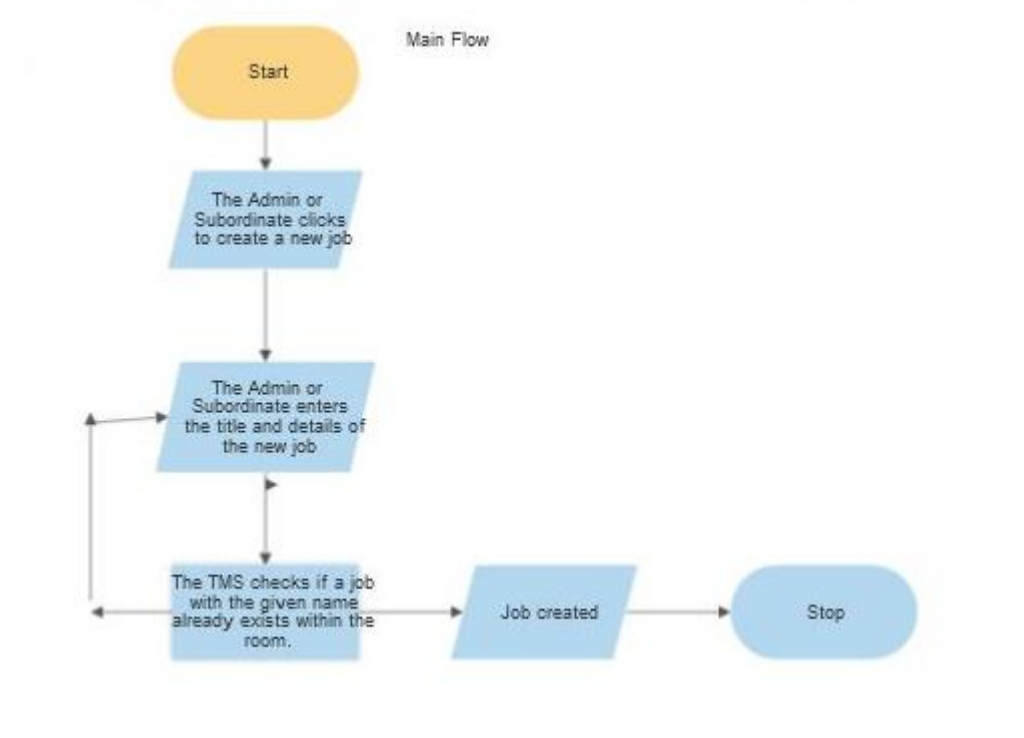
Custom Attribute

Additional requirement property capturing additional requirements properties such as requirements source, status, priority, verification method.

Document:

A structured requirements specification capturing textual requirements for a given product or service.

Appendix B: Analysis Models



Appendix C: To Be Determined List

Following is the list of TBD:

1. Database Platform to be implemented
2. Backend Programming Language
3. Tool for Front end designing