Software Requirements Specification

for

<Task Management System>

Version 1.0 approved

Prepared by <Group No. 4>

<Section No. 5215>

<SEP/26/2021>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 3

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 4

1.5 References 4

2. Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 5

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 7

4.1 Task Distribution 8

4.2 Create and Complete Tasks 11

4.3 Task Management………………………………………………………………………..14

4.4 Authentication……………………………………………………………………………18

4.5 View Tasks……………………………………………………………………………….21

4.6 Update Content…………………………………………………………………………...22

4.7 Verifications………………………………………………………………………………23

5. Other Nonfunctional Requirements 25

5.1 Performance Requirements 25

5.2 Safety Requirements 26

5.3 Security Requirements 26

5.4 Software Quality Attributes 26

5.5 Business Rules 27

6. Other Requirements 27

Appendix A: Glossary 27

Appendix B: To Be Determined List 28

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| TMS(1.0) | SEP-26-2021 | First Draft | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The software requirements specified in this document are for the task management system version 1.0. As the name of the product itself specifies, the scope of this product includes providing the user with facilities to manage their tasks by making different task lists. As well as other features which are described in the sub section 1.4. Furthermore, this software requirements specification document covers all aspects of the task management system.

## Document Conventions

IEEE template for System Requirements Specification documents has been followed.

Main Headings :-

Font : Times new Roman

Size: 18

Style: Bold

Sub-Headings :-

Font : Times new Roman

Size: 14/12

Style: Bold

Text :-

Font : Times new Roman

Size: 12

Style: none

## Intended Audience and Reading Suggestions

This document is intended for different types of readers mainly clients, users, developers, and the marketing staff. The software requirement specification document for version 1.0 of the task management system will include other sections namely, Overall Description, System Features, Nonfunctional Requirements, Other Requirements as well as Appendices A and B . It would be a good practice to go about the document in the order mentioned above however, specific readers can always jump to the sections or subsections they are interested in for example developers might want to check out the system requirements and the nonfunctional requirements and the users might want to check out the product scope.

## Product Scope

The main functionality of the task management system is to allow users to manage their different types of tasks effectively. Users can create task lists to list specific tasks under one general category for example household work, office work, university work etc. The users can then create tasks under a specific task list for example in the household work task list a user can create a task such as buy groceries etc. Users will also have an option to add subtasks under a certain task for example under the bring groceries task a user can create subtasks such as buy eggs, buy vegetables etc. Moreover, a user will have the option to add participants to divide a certain task given that the user who is trying to add participants to the task is the one who initiated the task in the first place i.e., that user will be the leader. After adding the participants, the leader can then distribute the subtasks of that specific task among the participants. A completion bar will be shown for each task within a task list depending upon the number of subtasks completed. There will be two types of tasks within a task list, those which are completed and those which are ongoing. An ongoing task can be marked as completed by a user and hence will be moved to the category of completed tasks. All the above-mentioned information can be considered as the scope of the task management system.

## References

* <https://docplayer.net/13534020-Software-requirements-specification-task-management-system-for-prepared-by-version-1-0-group-name-pink-and-purple-date.html>
* <https://cs.uwlax.edu/~mzheng/CS743Fall19/UseCaseDiagrams.html>
* <https://krazytech.com/projects/software-requirements-specification-report>
* <https://web.csulb.edu/~mopkins/cecs493/SystemRequirementsSpecificationExample.doc>
* <https://searchsoftwarequality.techtarget.com/answer/How-to-elicit-performance-requirements>

# Overall Description

## Product Perspective

The Task Management System (TMS) is a self-contained and a new product. The thought of developing such a system arose from the idea of how to make task managing an easier job with deadlines and reminders.

A Task Management System (TMS) is a program that tracks the progress of tasks as they are implemented in a program, helping to maintain the workflow of the tasks.

The Task Management System (TMS) module's primary objective is to provide a system for supporting repeated processes. As a result, productivity will be increased, beginner errors will be decreased, and business processes will be clearly defined.

## Product Functions

The System at minimum must provide the following functions to the user:

* User can register themselves via a Signup option and create their account
* User can then login to their account using their credentials.
* User can only access his/her data and tasks.
* User can view the task-lists/tasks/subtasks
* User should be able to create a task list
* User should be able to delete a task list
* User should be able to edit the task list
* User can any time create a task
* User can any time delete a task
* User can any time edit a task
* User can any time create a subtask
* User can any time delete a subtask
* User can any time edit a subtask
* User can add multiple subtasks under a task
* Users when working on a big project can add multiple participants to their project
* The User should be able to accept or reject whether to be part of a task.
* Users when working on a big project can distribute tasks among the participants
* There should be a completion bar showing what percentage of task has been completed
* Lastly the user can logout via logout option
* Any changes made by the user in the task management system will trigger an update in the content database

## User Classes and Characteristics

Users of the system would be able to create, manage, complete, and delete a task. The task app would therefore allow users to create distinct accounts and allow them to manage their daily tasks effectively. The users would be able to add more users to a task and distribute the tasks amongst other users. The system will support two types of user privileges, leader, and a member, incase multiple users are working on a big project. Leader would be able to add multiple members to a task with their consent. This would grant the members an access to that specific task and then the members can contribute to that task by working on different subtasks.

The user should be able to do the following functions:

* Create multiple tasks

1. Individual task
2. Group task

* Manage a task
* Add more users to a specific task
* Divide the task among participants
* Complete a task

The tasks & tasks list would have the following functionalities:

* Each task/task-list would have a completion bar indicating the percentage of task done.
* The task can be created, edited, deleted at any time.
* Tasks of a user would only be accessible to that specific user unless shared.
* Tasks can be prioritized using a color.

## Operating Environment

Operating environment for the task management system is as listed below:

* Distributed database
* Client/server system
* Operating system: Windows, Linux, etc.
* Database: Oracle
* Frontend: angular-js / react-js
* Backend: Nodejs
* Hardware: Desktop/Laptop/Mobile.
* Browsers: Chrome, Opera, Internet explorer, etc.

## Design and Implementation Constraints

* The system is a web-app designed using html, java script and CSS.
* The system will use an internet connection, or the user will not be able to

communicate with the database.

* The Task Management System must be portable so that multiple computers may be used to look at the information.
* The computers must be equipped with web browsers such as Internet explorer.
* A general knowledge of basic computer skills is required to use the product.
* Users must have their correct usernames and passwords to enter their accounts and do actions.
* The memory capacity of the server should be sufficient.

## User Documentation

A detailed user manual will be provided that will entail all the instructions and guidelines on how to use the system and other than that GUI will be intuitive and easy to use so with minimal effort one will get used to the system

## 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.
* This system will be written for users with a basic understanding of how computers work.
* The user already has a working e-mail address.
* The server is of high quality.
* The user has a web browser.

# External Interface Requirements

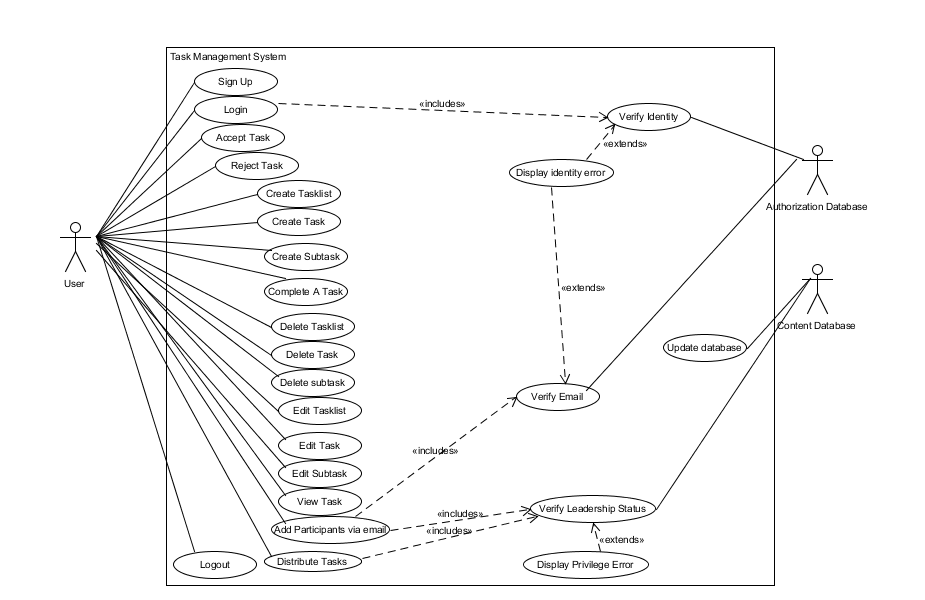
## User Interfaces

## Hardware Interfaces

## Software Interfaces

## Communications Interfaces

# System Features

**Use Case Diagram**

## Task Distribution

### Description and Priority

This feature allows users of Task Management System to add participants and distribute subtasks in a Task. The initiator of a Task can add multiple participants to his/her Task using their e-mail address. The added participants can then be allotted to specific Subtasks within that Task.

This feature has a high priority.

Benefit: 9

Cost: 8

Risk: 8

### Stimulus/Response Sequences

1. Add participants via email
2. Distribute Tasks
3. Accept Task

4. Reject Task

### Functional Requirements

TD-REQ-1: Add participants via email

The leader of a Task is allowed to add multiple participants to that task. The user must first prove that he is the leader. The email of the participant provided by the user must be verified. The “Verify Leadership Status” use case makes sure that the user who is trying to call this use case is actually the leader. If the user who called this use case is not the leader, then an error is generated which is then handled by the “Display Privilege Error” use case. The “verify email” use case is used to check whether the provided email is legitimate or not. If the provided email is not authentic then an error occurs which is handled by the “display identity error” use case.

TD-REQ-2: Distribute tasks

The leader of a Task is allowed to distribute subtasks among different participants. The user must first prove that he is the leader. There must be at least one participant involved in the Task. The “Verify Leadership Status” use case makes sure that the user who is trying to call this use case is actually the leader. If the user who called this use case is not the leader, then an error is generated which is then handled by the “Display Privilege Error” use case.

TD-REQ-3: Accept task

A user can accept or reject an invitation via email to participate in a task.

TD-REQ-4: Reject task

A user can accept or reject an invitation via email to participate in a task

## Create and Complete tasks

### Description and Priority

This feature refers to the ability of the Task Management System to allow users to create and mark as complete, different task lists, tasks, and subtasks. The users will create as many task-lists as they want. They will then be able to create individual tasks within these lists. Furthermore, the users can also create subtasks for bigger tasks in their lists. The user can also mark tasks as complete, and they will then be moved to “completed tasks” list.

The priority for this feature is high.

Benefit: 9

Cost: 7

Risk: 4

### Stimulus/Response Sequences

1. Create Tasklist
2. Create Task
3. Create Subtask
4. Complete a Task



### Functional Requirements

C-REQ-1: Create Task List

The task management system should allow users to create multiple task-lists in which they can create different Tasks.

C-REQ-2: Create Task

The Task Management System should allow the user to create different individual tasks within the task lists that they have created for themselves. These tasks may or may not contain subtasks and they may or may not be shared with other users.

C-REQ-3: Create Subtask

The Task Management System allows user to create subtasks within their tasks to help them divide a major task into small sized minor tasks. These subtasks can be shared with the participants within a task. The addition of participants and the distribution of subtasks is done by the leader of the task to which these subtasks belong to.

C-REQ-4: Create Subtask

The task management system should allow users to complete on going tasks they have created. These tasks will then be moved to a list where all the completed tasks are kept.

## Task Management

### Description and Priority

This feature allows users of Task Management System to delete and edit tasks, subtasks in a Task list. Only the initiator or leader of the Task or subtask can edit them and as for deletion of tasks and subtasks a user is given a prompt of whether the task/subtask is to be deleted for everyone or them only, if the option for everyone is chosen then it is verified if they were the initiators or leaders of the task/subtask.

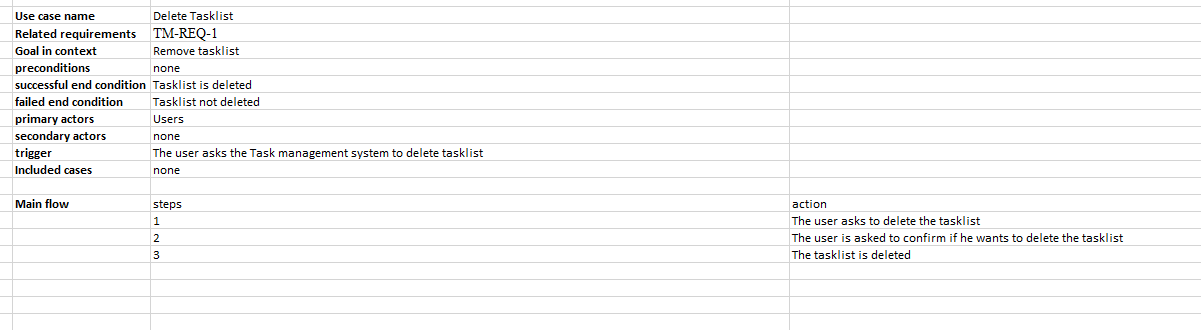
This feature has a high priority.

Benefit: 9

Cost: 8

Risk: 9

### Stimulus/Response Sequences

1. Delete Task list

1. Graphical user interface, text, application, email

   Description automatically generatedDelete Task
2. Graphical user interface, text, application, email

   Description automatically generatedDelete Subtask
3. Edit Task

Application

Description automatically generated with medium confidence

1. Edit Task list

Application

Description automatically generated

1. Edit Subtask

Graphical user interface, text, application, email

Description automatically generated

### Functional Requirements

TM-REQ-1: Delete Task list

Every user can create their own personalized task list which consists of tasks assigned to them. The user can decide to delete this task list if he/she deems it unwanted or not required anymore.

TM-REQ-2: Delete Task

Every user also has the ability to create tasks, and tasks can be assigned to multiple people. Only the creator of the task has the ability to delete the task for everyone, the participants can only delete the tasks for themselves. Thus, when a user asks to delete a task, the “Verify Leadership Status” use case makes sure that the user who is trying to call this use case is actually the leader and then gives the option for delete for everyone, otherwise the only option given is for delete for me.

TM-REQ-3: Delete Subtask

Every user also has the ability to create subtasks, and subtasks can be assigned to multiple people. Only the creator of the subtask can delete the subtask for everyone, the participants can only delete the subtasks for themselves. Thus, when a user asks to delete a subtask, the “Verify Leadership Status” use case makes sure that the user who is trying to call this use case is actually the leader and then gives the option for delete for everyone, otherwise the only option given is for delete for me.

TM-REQ-4: Edit Task

Every user also has the ability to edit tasks, in case changes are needed to be made to the tasks. Only the creator of the task can make changes to it therefore whenever a user tries to edit a task the “Verify Leadership Status” use case makes sure that the user who is trying to call this use case is actually the leader and then gives the right to edit the task. If the user who called this use case is not the leader, then an error is generated which is then handled by the “Display Privilege Error” use case. The user will not be able to edit the task if they are not the creator.

TM-REQ-5: Edit Task list

Every user will have the ability to edit their personalized task list for themselves if it is deemed that changes are to be made to the task list. No verification of leadership is required since task list is already personalized for everyone, and they will only be editing it for themselves.

TM-REQ-6: Edit Subtask

Every user also has the ability to edit subtasks, in case changes are needed to be made to the subtasks. Only the creator of the subtask can make changes to it therefore whenever a user tries to edit a subtask the “Verify Leadership Status” use case makes sure that the user who is trying to call this use case is actually the leader and then gives the right to edit the subtask. If the user who called this use case is not the leader, then an error is generated which is then handled by the “Display Privilege Error” use case. The user will not be able to edit the subtask if they are not the creator.

## Authentication

### Description and Priority

This feature allows a new user who does not exist in the database to create a new account or an existing user to log in or out of their account. A new account with the email of an existing user can not be created. A user can only be displayed the option of logging out when they are logged into the system. A user will only be displayed the option of logging in when they are not currently logged in to any account. A user will only be displayed the option to sign up if they are not logged in to an existing account.

This feature has a high priority.

Benefit: 9

Cost: 8

Risk: 9

### Stimulus/Response Sequences

1. Sign up

|  |  |  |
| --- | --- | --- |
| Use case name | | Sign up |
| Related requirements | | A-REQ-1 |
| Goal in context | | A new user can sign up |
| preconditions | | User does not exist |
| Successful end condition | | New User created |
| Failed end condition | | New user not created |
| Primary actors | | User |
| Secondary actors | | Authorization database |
| trigger | | A new user clicks on sign up |
| Included cases | | - |
| Main flow | step | action |
|  | 1 | A new user asks to sign up |
|  | 2 | User enters his details |
|  | 3 | User details are stored in authorization database |
|  | 4 | New user is created with login details entered in previous step |
|  |  |  |

1. Log in

|  |  |  |
| --- | --- | --- |
| Use case name | | Log in |
| Related requirements | | A-REQ-2 |
| Goal in context | | An existing user can Log in |
| preconditions | | User already exists |
| Successful end condition | | User gets logged into his account |
| Failed end condition | | Login attempt is rejected |
| Primary actors | | User |
| Secondary actors | | Authorization database |
| trigger | | An existing user asks to log in |
| Included cases | | Verify identity |
| Main flow | step | action |
|  | 1 | An existing user asks the system to log in |
|  | 2 | User enters his details |
|  | 3 | Details are verified from the authorization database |
|  | 4 | User is logged in to his account |
|  |  |  |

1. Log out

|  |  |  |
| --- | --- | --- |
| Use case name | | Log out |
| Related requirements | | A-REQ-3 |
| Goal in context | | A logged in user can log out of his account |
| preconditions | | User is already logged in |
| Successful end condition | | User is logged out of his account |
| Failed end condition | | Log out attempt fails |
| Primary actors | | User |
| Secondary actors | | - |
| trigger | | Logged in user clicks on logout |
| Included cases | | - |
| Main flow | step | action |
|  | 1 | A logged in user clicks on logout button |
|  | 2 | The user is logged out |
|  |  |  |
|  |  |  |
|  |  |  |

### Functional Requirements

A-REQ-1: Sign up

A new user who wishes to manage their tasks can user their email to sign up. If an account with that email address is already an existing account, then the user will be prompted to log in with their existing account.

A-REQ-2: Log in

An existing user can log in to their account by using their credentials If the user enters the incorrect credentials, then an error will be displayed prompting the user that their entered credentials do not match the ones in the authorization database.

A-REQ-3: Log out

A logged in user can log out of their account. This option will only be displayed if the user is logged in to their account.

## View Tasks

### Description and Priority

With this feature, the task management system allows the user to view subtasks within a certain task. The user can then manage the task accordingly like adding participants to the task or marking it complete etc. Whenever a user logs in, they will be shown their respective task lists and a preview of all the tasks in that task list will also be shown. The user will then click on one of the previews of their choice to view full details of that task and the subtasks that are inside that task. User will also be able to edit the opened tasks and the subtasks in them.

This feature is of high priority since it allows users to view the details of the tasks they have created, and it opens the interface where users will be able to edit their tasks and subtasks within those tasks.

Benefit: 9

Cost: 8

Risk: 9

### Stimulus/Response Sequences

* 1. View Task Details

### Functional Requirements

1. VT-REQ-1: View Task Details

Users can access the details of their tasks that they have created inside task lists. In addition to these details the users will also be able to see the subtasks that are a part of the task which is being viewed. Users can also edit the task after viewing it as well as edit the subtasks that are a part of that task.

## Update Content

### Description and Priority

This feature of the task management system is used to make sure that any work that the user performs doesn’t get lost and that the data/changes are saved into the database. For example, if a user creates a new task list and adds some tasks to that task list then this feature will make sure that all of this is saved. The same applies for other operations that the user performs. This feature ensures concurrency in the database and allows users to make changes dynamically.

This feature is of High priority since saving data is an essential part of the task management system.

Benefit: 9

Cost: 9

Risk: 9

### Stimulus/Response Sequences

1. Update Database

### Functional Requirements

1. UD-REQ-1: Update Database

The task management system will update the content database in accordance with the operations performed by the user. This is done to store any new data, update or delete existing information etc.

## Verifications

### Description and Priority

This system feature emphasizes on the integrity and security of the task management system. The functionalities that this feature provides include authenticating the user upon login, verifying the email address of the intended participants of a task in order to send them an invitation and lastly when a user tries to add participants in a task or distribute subtasks of a certain task among the participants, it is important for the task management system to verify the leadership status of the user since only the user who initiated a task can perform these operations on that task. These seemingly basic functionalities play an important role towards making the task management system coherent and secure.

This feature is of medium to high priority.

Benefit: 7

Cost: 6

Risk: 6

### Stimulus/Response Sequences

1. Verify Identity
2. Verify Email



1. Verify Leadership Status

### Functional Requirements

1. V-REQ-1: Verify Identity

When the user tries to login, the task management system needs to verify that whether the user is legitimate or not by checking the Authorization database.

1. V-REQ-2: Verify Email

When the leader of a task tries to add participants to that task via their email, then the email must first be verified by checking the Authorization Database to make sure that that email belongs to an existing user and that the leader isn't making any mistakes while providing the email of the person that needs to be added (or the user provides a wrong email on purpose to test our application ☹)

1. V-REQ-3:

When a user tries to add participants to a certain task via email or when a user distributes the subtasks of that task among the participants, the task management system needs to check whether that user is the leader or not i.e., whether that user is the one who initiated the task in the first place to which he/she intends to add participants or whose subtasks he/she intends to distribute. This will be done by checking the content database

# Other Nonfunctional Requirements

## Performance Requirements

Feature wise Performance requirements:

* Task Distribution feature:

The system should support concurrency; hence it should allow multiple users to use the system simultaneously and participate in tasks. It should keep teams on the same page. The request to participate in a task should not take more than 10 seconds to reach a user.

* Create and Complete Tasks feature:

The task lists, tasks and subtasks should be created immediately after user enters their details and should not take more than 3 seconds. The completion bar should be updated within 5 seconds of marking a task complete for all the participants to see.

* Task management feature(delete & edit):

The database should be updated quickly and frequently to reflect the changes made by users. Each time a change occurs the database will be updated to maintain concurrency.

* Authentication feature:

It should take no more than 10 seconds to open the system. It should take no more than 10 seconds to log in or log out.

* Viewing Feature:

Whenever a user asks for the system to view the details of a specific task, it should take no more that 10 seconds for the system to fetch the details of a task.

## Safety Requirements

The database will not accidently lose user account information. The system will keep backup of database and log file to allow recovery of database transactions. Use a stable internet connection to avoid loss of data and in case of a power failure use UPS.

## Security Requirements

* AUTHORIZATION REQUIREMENTS:

A particular user’s data will only be accessible to him/her, and an authentication system is used to safeguard the account from unauthorized access or accidental login. Distribution of subtasks and addition of new participants is allowed only to the Leader.

* AUTHENTICATION REQUIREMENTS:

To protect user's data, a secure database will be used to store records and the system will require users to create distinct accounts to access records. The system grants access to accounts only when users enter the correct username and password. The system will ensure that the users use strong passwords that are at least 8 characters long and include at least a number and a capital letter.

## Software Quality Attributes

* AVAILABILITY:

 The database for this system will be hosted on a server therefore the system will only be available when the internet connection is present. It shall provide actors with minimum operational availability of 99%.

* CORRECTNESS:

The system will correctly keep track of all the task lists, tasks, and subtasks that a user creates as well as the distributions of tasks among users. It will also correctly show the changes that are being made in real time.

* USABILITY:  
   It will be easy for users to interact with the app and manage their tasks. The interface will be intuitive, and its navigation will be easy to learn such that users can easily figure out what a feature is and what it can do. For example, predicting that tapping the “+” button creates a task.

The system will satisfy a maximum number of user needs. The priority of tasks will be indicated using colors. Red for a task that is due within 3 hours, yellow for due within 24 hours and green for due after more than 24 hours.

System notifications for due dates of tasks will be shown to users periodically.

* SCALABILITY:

The TMS will be scalable to allow a maximum number of user accounts, and simultaneous user access.

* ROBUSTNESS:

System will continue to function under a certain number of abnormal circumstances and handle invalid inputs or invalid access with proper error messages.

* INTEROPERABILITY:  
   The TMS shall interoperate with Oracle Database and the following OS:

1) Windows

2) Linux

3) Android

## Business Rules

All users need a valid email address. The sign-up feature shall include a mandatory email input. Only the leader of a task should be able to add other users to that task via their email and only he shall be able to distribute tasks among them.

# Other Requirements

* A server should be available.
* Reasonable space to store data on the server
* No copyrighted material/code used
* The web link for the application will be available to users.
* The application will be open source
* Oracle database will be used

Appendix A: Glossary

* TMS: Task Management System.
* Use Case: A situation where the system is used to fulfill user’s requirements. It captures a piece of functionality that the system provides.
* Client: Intended users of the system.
* Oracle: The database service provider used.
* Non-functional requirements: targets ensuring effectiveness of the software.
* Functional requirements: targets essential for the software.
* Distributed Database: integrated collection of databases that is physically distributed across multiple sites in a computer network.
* Client/Server system: Distributed application structure that partitions tasks or workloads between the providers of a service called servers, and service requesters, called clients.
* GUI: Graphical User Interface.
* Extend: The <<extend>> relationship shows that a use case might completely reuse another use case's behavior, but this reuse is optional.
* Include:The <<include>> relationship declares that the use case at the head of the dotted arrow completely reuses all the steps from the use case being included.
* Leader: The initiator of a task.
* HTML/CSS/JAVA Script: programming language used.

Appendix B: To Be Determined List

* Some new features to add:
  + Feedback option for multiuser tasks.
  + Calendar view that shows due dates.
  + Comment option for users in multiuser tasks.