جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

Imam Abdulrahman Bin Faisal University
College of Computer Science & Information Technology
Department of Computer Science

CS 411 – Software Engineering
Term 1 – 2024/2025

Software Design Specifications

Task tracker

Instructor Dr. Muawiya Abdul majid

Submission date:25 November 2024

This Software Design Specification was prepared and provided as a deliverable for [Course Name, number, term], and it will be used by [name of end user].

This document is based in part on the IEEE Recommended Practice for Software Design Descriptions.

Group members:

#	Name	ID	Role
1	Ritaj Alhamli	2210002809	Leader
2	Maram Alnabrees	2210009014	Member
3	Anwar Aldahan	2220004414	Member
4	Fatimah Alwarsh	2220003876	Member
5	Shahad alshehab	2220003900	Member
6	Fatima Alawami	2220005142	Member
7	Fatimah Mohsen Alageel	2220003313	Member

Table of Content

Revision History

Name	Date	Reason For Changes	Version
All members	Oct 31, 2024	Prepared initial version	0.1
All members	Nov. 3, 2024	Updated section 1	0.2
All members	Nov. 3, 2024	Updated section 2	0.2
All members	Nov. 6, 2024	Updated section 3	0.3
All members	Nov. 8, 2024	Numbering the tables and figures	0.4
All members	Nov. 8, 2024	Checking of grammar and spelling	0.4
All members	Nov. 9, 2024	Complete review - Final version	1.0

Table of Tables

Table of Figures

1. Introduction

This subsection of the document is an overview of the Task Tracker Management System (TTMS) project and contains information about its purpose and scope, definitions, and acronyms used in the project, as well as references. This document will present a clear vision of TTMS specification, the target users and expected capabilities of the system to the stakeholders like the developers, the team leaders and the end users.

1.1 Purpose

This SRS is composed for the identification of some of the key aspects of the Task Tracker Management System. The present document describes the capabilities of the system, which is necessary to achieve the project objectives, as well as the required and desired characteristics of the system. The following stakeholders will benefit from this document:

- Development Team: This document will help the development team in defining the features and interfaces of the software to keep in check as they continue designing and implementing this project so that they can produce software that meets the required quality of features and interfaces of the software.
- Project Supervisor: This SRS will be useful for the project supervisor to evaluate the progress of the system and suggest some inputs for enhancement of the work.
- End Users: Using the goal and need analysis, users such as individual contributor, students, and team managers will comprehend the design purpose, the tasks from creating to completing and tracking.

1.2 Scope

The main goal of Task Tracker Management System is to enhance the general organization and efficiency of the tasks by providing a more adaptable and easier environment. Therefore, this section covers the activities and scope of TTMS, explaining how their respective functions will be implemented to address the projects intended goals. Showing that our TTMS will support diverse users by offering the following functionalities:

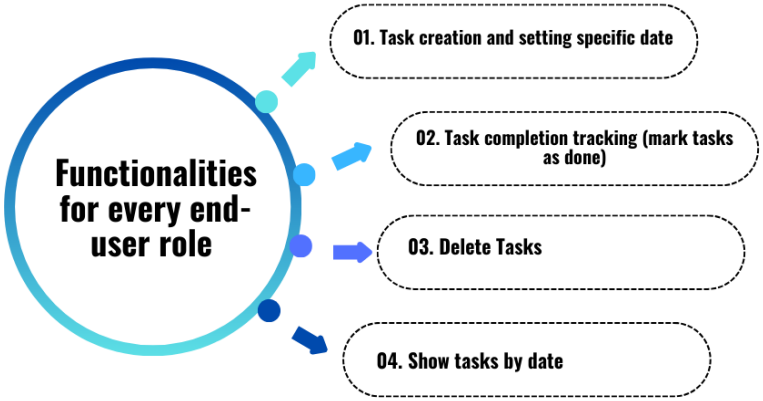

Figure 1: Provided Functionalities

The primary purpose of the TTMS is to optimize the effectiveness of work-flows by applying the principle of automating task handling procedures.

1.3 Definitions, Acronyms, and Abbreviations

Terminology	Definition
Usability	how well the users can address their needs by navigating the programs to solve problems.
Task Tracking	The act of organizing and distributing activities and ensuring they are effectively accomplished within a given timeframe.
Functionality	The specific functions provided by the software and the general ability of the software to meet the need of the user.
Adaptability	the extent to which a system, or portions of it, can be modified when there are minor variations in demands.
Maintainability	A maintainable system is a system that can be modified to accommodate new requirements, overcome difficulties and changing needs with the least effort.

Table 1: Terminologies Definitions

Acronyms	Meaning
UI	User Interface
UX	User Experience
IEEE	Institute of Electrical and Electronics Engineers
SRS	Software Requirements Specification
TTMS	Task Tracker Management System
GUI	Graphical User Interface

Table 2: Acronyms Definitions

1.4 References

[1] Institute of Electrical and Electronics Engineers (IEEE). (1998, June 25). IEEE Recommended Practice for Software Requirements Specifications. Retrieved October 22, 2023, from https://standards.ieee.org/standard/830-1998.html

[2] Bennett, S., McRobb, L., & Farmer, R. (2014). Object-Oriented Systems Analysis and Design (5th ed.). McGraw-Hill Education.

2. System overview

The Task Tracker Management system was built to assist end users in managing their tasks. It enables the users to create an account, log-in to their accounts, or simply continue as guests. Enable the users to prioritize their responsibilities and tasks, enhance organization and productivity, and simplify their daily tasks.

A general explanation of the system's functionality will be given in this section.

System functionalities:

- Account creation/ log-in
- Assign tasks as a guest
- Task Creation and Date Setting
- Display Tasks by Date
- Delete Tasks
- Tracking Task Completion

3. Design Considerations

In this section we will cover the assumptions, dependencies, and general limitations affecting our task tracker system.

The following list can be used to classify the software's assumptions and dependencies:

- Related software or hardware
- Operating systems
- End-user characteristics
- Possible and/or probable changes in functionality

3.1 Assumptions and Dependencies

Related software or hardware:

The application can run on any mobile device. The database for the program will be built with MySQL and will have a straightforward architecture to prevent data loss and protect the application's data. We will use MySQL Workbench to retrieve and store data, and the NetBeans IDE to create and test the application. Both should be installed on the developers' PCs.

Operating systems:

Our application is compatible with Java programming language so, it will operate on Android and IOS. The development team will use Windows 10/ Windows 11.

End-user characteristics:

Our application has one user. He/she can create an account/ log-in to existing account, and delete, prioritize, display tasks.

Possible and/or probable changes in functionality:

Since changes are unavoidable, we might make some changes to the original design to accommodate the user's new needs.

3.2 General Constraints

In this part, we will outline certain global restrictions or limits that significantly affect the "task tracker" design. Such as:

- Hardware or software environment
- Interface requirements
- Data repository and distribution requirements
- Security requirements
- Verification and validation requirements (testing)

Hardware or software environment:

Our application will be run on both IOS and Android. For the developers they will work on Windows 10/ windows 11. Also, they will need NetBeans IDE and MySQL workbench to store and retrieve data.

Interface/protocol requirements:

Our task tracker system will have a user-friendly interface that is clear, simple, and accessible.

Data repository and distribution requirements:

All data must be securely kept in a relational database that includes efficient data retrieval capabilities. MySQL is the database technology we have selected since it is dependable and facilitates effective querying.

Security requirements:

Unauthorized access must be prevented. So, every user will have a strong password and a distinct username. Additionally, if the user choose to continue as a guest all the tasks will be deleted when he/she close the application.

Verification and validation requirements (testing):

The system must be thoroughly tested to verify that it satisfies both functional and non-functional requirements. In other words, the primary purpose here is to test for bugs and find application specification failures early on and resolve them.

4. User Interface Design

4.1 Overview of User Interface

Users will be able to register to create an account, and that feature saves their work (the tasks they have applied). They can then sign in with their saved information (username and password) at any time and find their tasks with their respective dates. Some users, especially those who want to use the app temporarily, may not want to create an account. However, they can continue as guests and use the app without saving their work. As well as, any user will have the ability to sort their tasks by date to know which tasks to do first. Additionally, they can delete tasks or mark finished tasks as done.

4.2 Interface Design Rules

- **Consistent Layout**: Maintain a consistent layout throughout the app to provide a sense of familiarity and make it easier for users to predict where to find information.
- **Color and Contrast**: Colors and contrast effectively to highlight important elements and improve readability. Ensuring that the color scheme is visually appealing and accessible.
- **Feedback and Alerts**: Provide clear feedback to users when actions are taken.
- **Intuitive Navigation**: Keeping the interface simple and intuitive with clear navigation paths to allow users to easily move between different sections of the app.

4.3 Screen Images

4.3.1. home page interface

In the home page interface there are three options, log-in for already registered users, register for new users willing to have an account, and continue as guest option for temporary users.

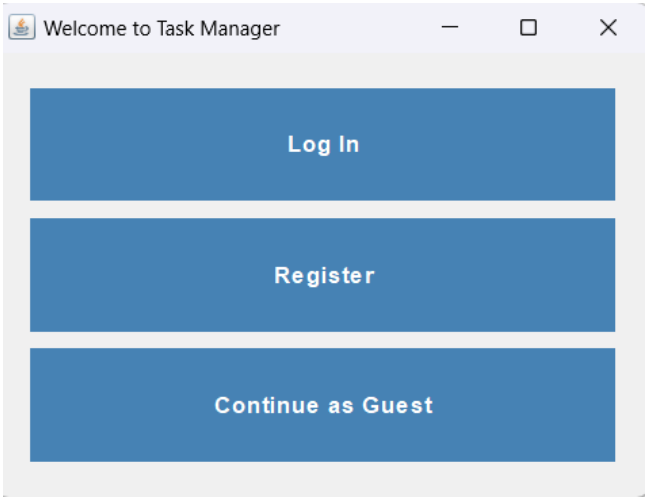

Figure 2: home page interface

4.3.2. Register interface for new users

New users can make an account by choosing a unique username and password. However, there are a restrictions to passwords (each password should contains digits and capital, small letters, as well as a special character and to be 8 characters long) to address the security for each user.

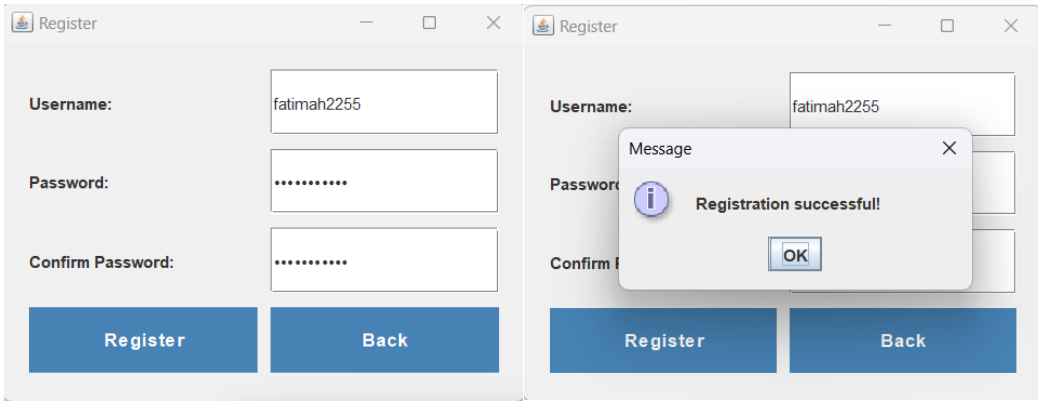

Figure 3: Register interface

4.3.3. log-in interface for users they have already registered

For already registered users, they can sign-in into the app with their saved username and password at any time to complete their tasks as all of them will be permanently saved.

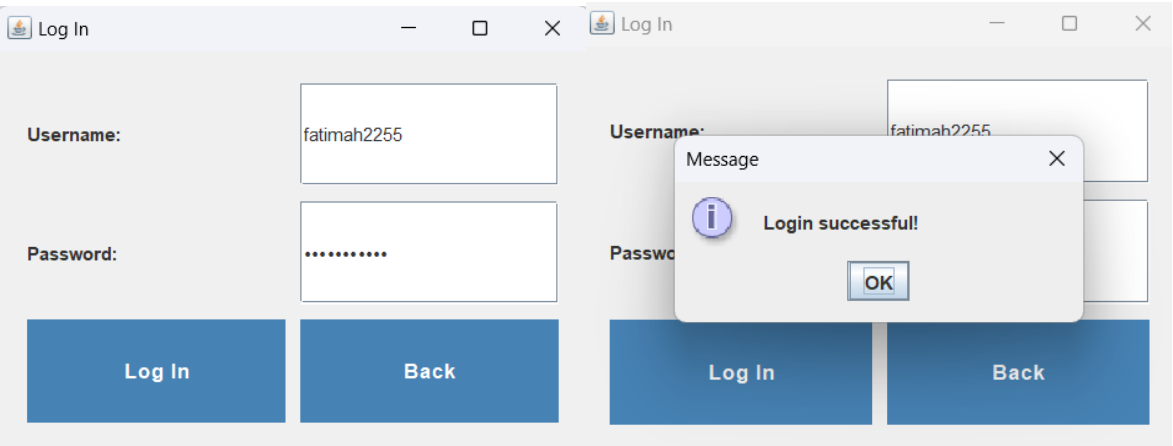

Figure 4: log-in interface

4.3.4. Tasks board

Guests and users will have the tasks board, they must specify a date for each task. Additionally, sort by date, delete task features are provided to sort all the tasks based on the date, and delete not needed anymore tasks. As well as users can mark each completed task as done.

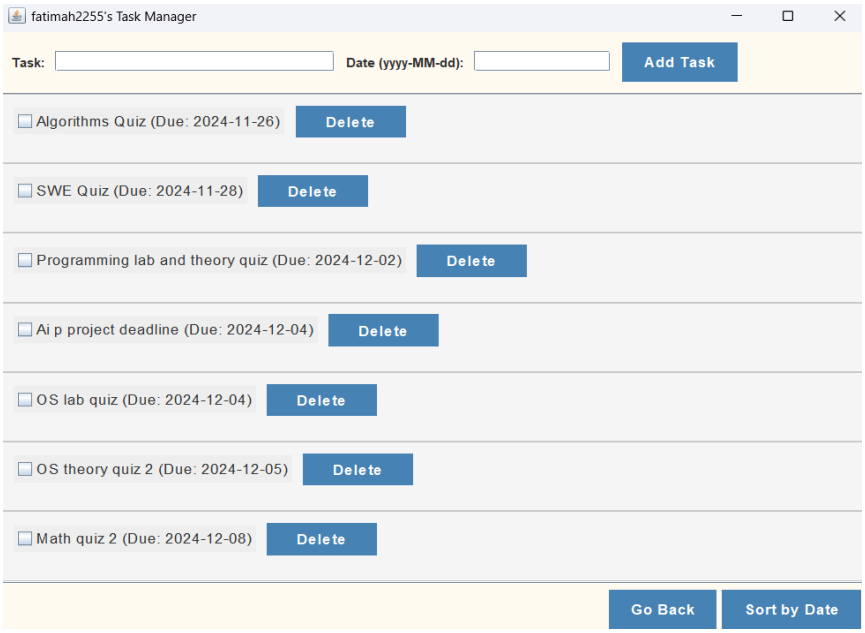

Figure 5: Tasks board

4.4 Screen Objects and Actions

No.	Object	Type	Action
	Home page interface		
1	Log-in	Button	Moves to log-in interface.
2	Register	Button	Moves to register interface.
3	Continue as Guest	Button	Moves to the task board interface.
	Log-in interface		
4	Log In	Button	Checking if the provided information is saved in the database to allow the user to access their task board.
5	Back	Button	To move back to homepage interface.
	Register interface		
6	Register	Button	Saving user's username and password in the database.
7	Back	Button	To move back to homepage interface.
	Tasks board interface		
8	Add task	Button	Adding the task with their date to the board.
9	Delete	Button	Deleting a specific task.
10	Sort by date	Button	Sorting all tasks based on date.
11	Go back	Button	Moving back to homepage.
12	Mark as done	Checkbox	Marking completed tasks as done.

Table 3: Screen Objects and Actions

5. System Architecture

The architecture of our program will be discussed carefully in this section, this section explores the architecture that underpins the Task Tracking system, the architectural design will discussed first, then the architectural design, followed by the subsystem architecture.

Task Track have main role: Tasks management, and it have the login feature.

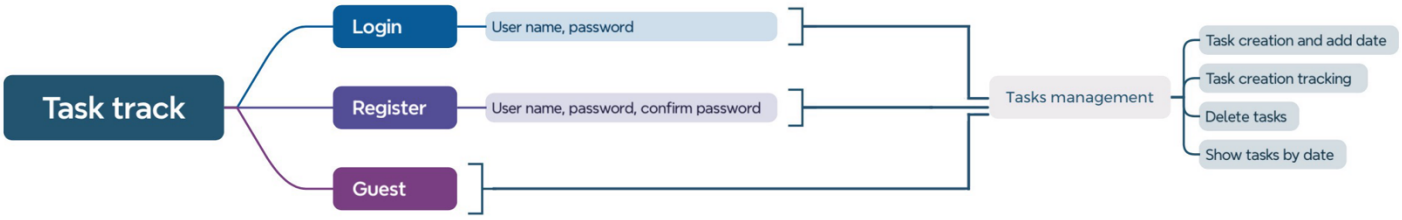

Figure 6: Task track architectural design

5.1 Architectural Design Approach

For our system we find that the most appropriate architectural design approach is the layered architecture, because it will be easier for us in our further development to add more features and enhance the system abilities, also the changes will be applied to the system easily.
Which consists of 3 layers:

- User interface layer: the first layer of the application displays information in an understandable way that users can interact with it directly.
- Application service layer: the Business Logic Layer, this layer acts as the bridge between the user interface and the data layer, by handling tasks and performing the functions.
- Data layer: the lowest layer that responsible for managing data storage, retrieval, and manipulation within the application.

5.2 Architectural Design

- User interface layer: in this layer we need to use GUI to display the screen and its related data to the user which takes data from the user as an input and send it to the application service layer to produce the wanted output.
- Application service layer: the connection between user interface layer and the data layer, it will take the actions made by the user in the user interface layer to perform the application functions also to store retrieve from data layer.
- Data layer: the layer that can access the database server, it can update retrieve and delete data by using queries.

The relationship between all three layers:

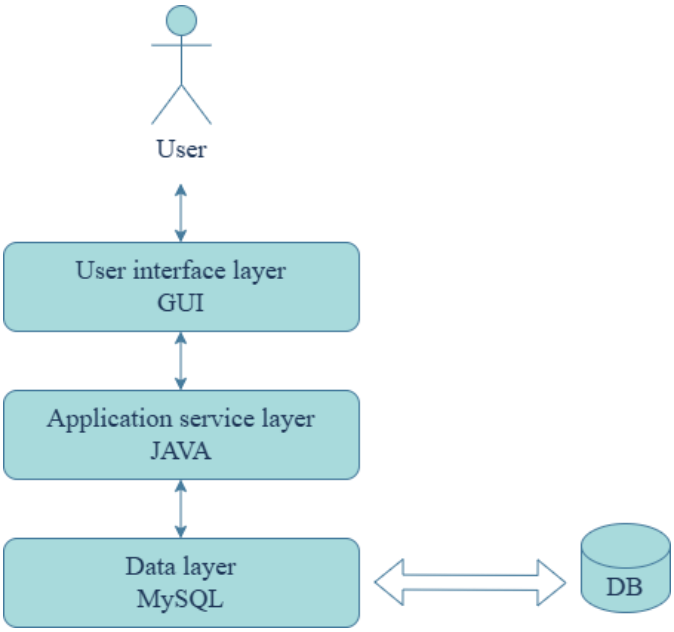

Figure 7: Layered software architecture.

5.3 Subsystem Architecture

This section will break down the system into smaller and more manageable parts, outlining the main operations, the functions of the system, and how to store the data in the server.

5.3.1 User's Subsystem

This following level 0 data flow diagram illustrates user interactions and data interchanges between processes for the shared functionalities:

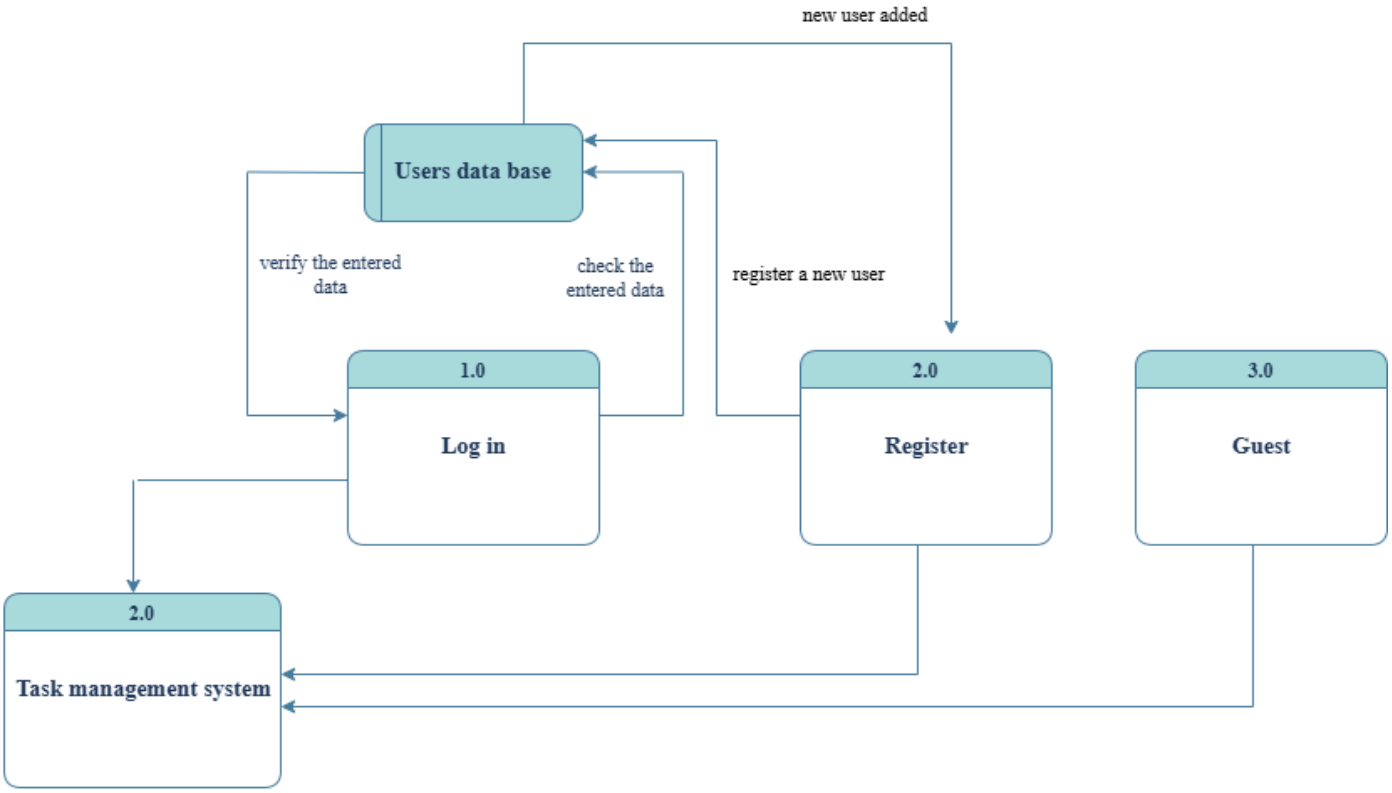

Figure 8: User subsystem.

5.3.2 Task management Subsystem

The following subsystem of level 0 describes the data flow between the functionalities of task management task system:

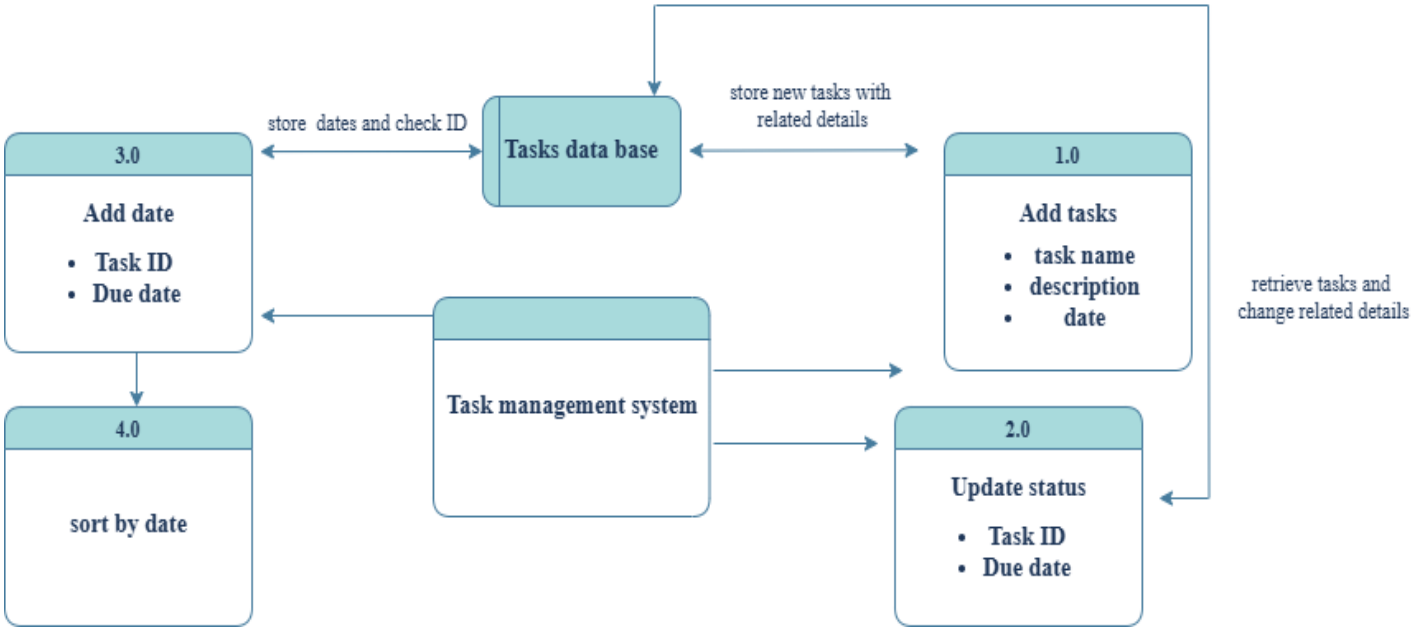

Figure 9: Task management Subsystem.

6. Data Design

This section details how information in TTMS is organized, processed, and stored. It contains the data description, dictionary, and database design that will give a clear framework for understanding and implementing the data handling aspects of the system.

6.1 Data Description

The Task Tracker Management System converts the information domain into data structures in order to enable functionalities like task creation, tracking, deletion, and displaying. The main data entities and their organization are as follows:

Entity	Field	Type	PK	FK	Constraints
Tasks	Task_ ID	INT (8)	✓		NOT NULL,uniqe
	User_ID	INT (8)		✓	NOT NULL,uniqe, Referencing Users.User_ID
	Description	VARCHAR (50)			NOT NULL
	Due_date	DATE			NOT NULL
	Completed	ENUM('0', '1')			NOT NULL, defaults to '0'
Users	User_ID	INT (8)	✓		NOT NULL, uniqe
	Username	VARCHAR(30)			NOT NULL, uniqe
	Password	VARCHAR(40)			NOT NULL

Table 4: Data Description

6.2 Data Dictionary

Below is a detailed description of the major data entities and their attributes:

Entity	Field	Description
Tasks	Due_date	Deadline for task completion.
	Task_ ID	Unique identifier for each task.
	Description	Name/description of the task.
	Completed	Indication of the task completion (1 (Completed)/0(Not completed)).
	User_ID	Referencing the User_ID in Users table
Users	Password	User's password.
	User_ID	Unique identifier for each user.
	Username	User's login name.

Table 5: Data Dictionary

6.3 Database Description

The database for TTMS will be implemented as a relational database using SQL. Below is an overview of the ER diagram and the relational mapping of the database structure:

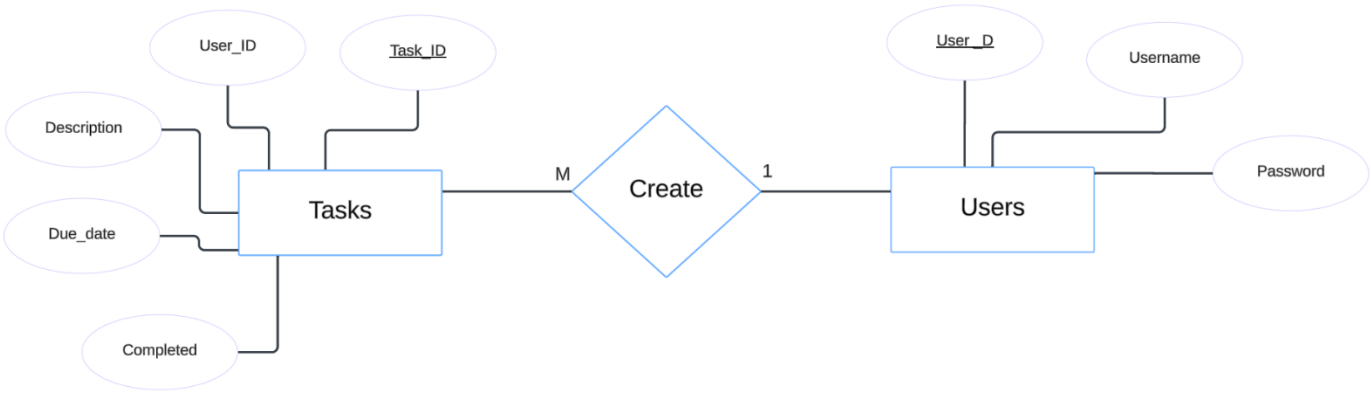

Figure 10: The ER diagram

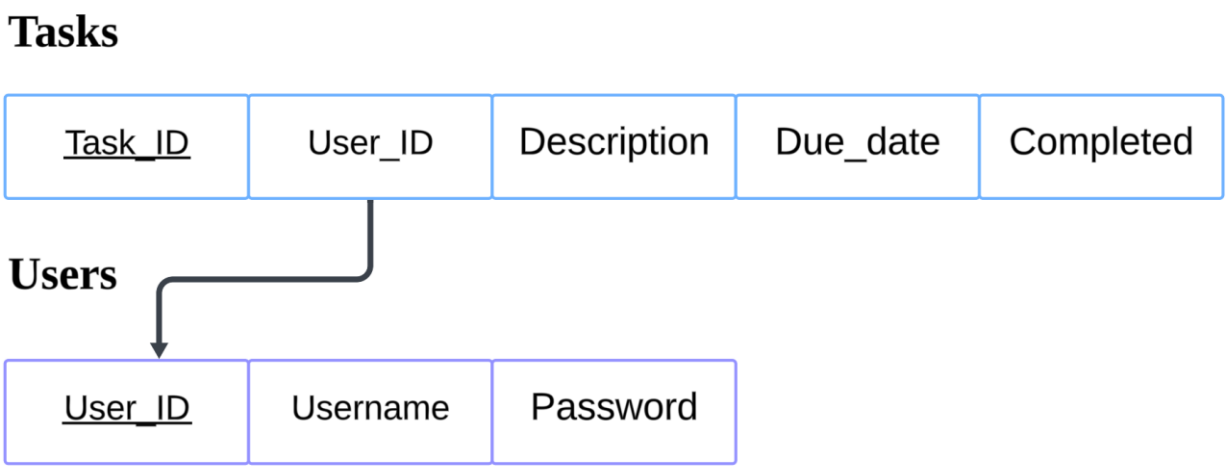

Figure 11: Relational Mapping

7. Component Design

This section includes the pseudocode for the task tracker system, detailing each component in the system algorithmically.

7.1 Common functions

7.1.1 Login

Function Login() {

Enter username and password

If username exists in the database AND the password matches:

Redirect to the dashboard

Else:

Show error message: "Incorrect username or password. Please try again!"}

7.1.2 Register

Function Register () {

Enter username, password, and confirm password

If username is empty OR already exists in the database:

Show error message: "Username is taken or invalid. Please choose a different one."

Else If password does not match confirm password:

Show error message: "Passwords do not match. Please try again."

Else:

Save username and password in the database}

7.1.3 Continue as Guest

Function continueAsGuest() {

Show a message: "Tasks will not be saved in the database during this session."

Redirect to the dashboard

}

7.2 Task Management

7.2.1 Create Task

Function CreateTask() {

Enter task title and due date

If input fields are valid:

If user is registered:

Save task in the database

Show success message: "Task created successfully!"

Else:

Show message: "Task created but not saved. Register to save your tasks."

Else:

Show error message: "Please fill in all required fields correctly."

}

7.2.2 View Tasks

Function showTaskManager() {

If user is registered:

Query tasks from the database

Sort tasks by due date

If tasks are found:

Display tasks in a list format

Else:

Show message: "No tasks found."

Else:

Display tasks created in the current session

}

}

7.2.3 Mark Task as Completed

Function updateTaskCompletionStatus (taskID) {

If user is registered:

Update the task status in the database to "Completed"

Show success message: "Task marked as completed and saved!"

Else:

Show message: "Task marked as completed but not saved."

}

7.2.4 Delete Task

Function deleteTaskFromDatabase (taskID) {

If user is registered:

Delete the task from the database using taskID

Show success message: "Task deleted successfully!"

Else:

Remove the task from the session list

Show message: "Task deleted for this session."

}

7.2.5 Sort Tasks by Date

Function SortTasks() {

If user is registered:

Query tasks from the database

Sort tasks by due date in ascending order

Display sorted task list

Else:

Sort tasks created in the current session by date

Display sorted task list

}

}

7.3 Guest Mode

7.3.1 Create Guest Task

Function CreateGuestTask() {

Enter task title and due date

If input fields are valid:

Add task to the session list (not saved in the database)

Show message: "Task created for this session."

Else:

Show error message: "Please fill in all required fields correctly."

}

7.3.2 View Guest Tasks

Function ViewGuestTasks() {

Display tasks created during the session

If no tasks are found:

Show message: "No tasks found in this session."

}

8. Detailed System Design

8.1 Classification, Definition and Responsibilities:

This section will include the classification, definition and responsibilities of each component.

Components	Classification and Definition	Responsibilities and composition
Log IN	In the Starting Interface. This function will allow the user to access their account and their task history.	Wil ask the user to write their name and their password, and they have to be correct to continue. And allow the user to go back to choose another choice.
Register	In the Starting Interface. This function will allow the user to make new account so their information will be saved their.	The user will enter their name and password and confirm the password to make an account, then they click Register to continue or back to choose another thing. Error when repeated username or weak password or empty.
Continue as guest	In the Starting Interface, This choice allow the user to write their task and see them only as temporary.	The user can write their task based on some dates and sort the tasks by date or go back to choose another thing but there in no database for this user.
Tasks	The interface after login that allow the user to write their own tasks.	This interface allow the user to write their own tasks and choose a date for each task then add these tasks then you also can sort the tasks by the date from oldest once also you can delete tasks or check them or go back.
Database	Will present the saved data from the program.	Will have two tables one for the account and the other one for tasks .

Table 6: Classification, Definition and Responsibilities

8.2 Constraints:

- We have designed the username to be primary key so it shouldn't be repeated or null.
- The Password has some constraints so it shouldn't be null or weak based on some rules.
- When we click sort the data based on some function will be ordered from the oldest to the newest
- Check if the password is correct and the user can log in it has to match the password for that username
- When the user confirms their password at the confirmation it has to match the password.
- Each user can see only their own tasks.
- When task deleted or added the database should be updated directly.

8.3 Composition:

A description of the use and meaning of the subcomponents that are a part of this component. Mention in detailed in the table.

8.4 Uses/Interactions (sequence diagrams):

This section describes the components' relationships. Sequence diagrams illustrate component relationships and interactions. It shows the links and order in which processes function.

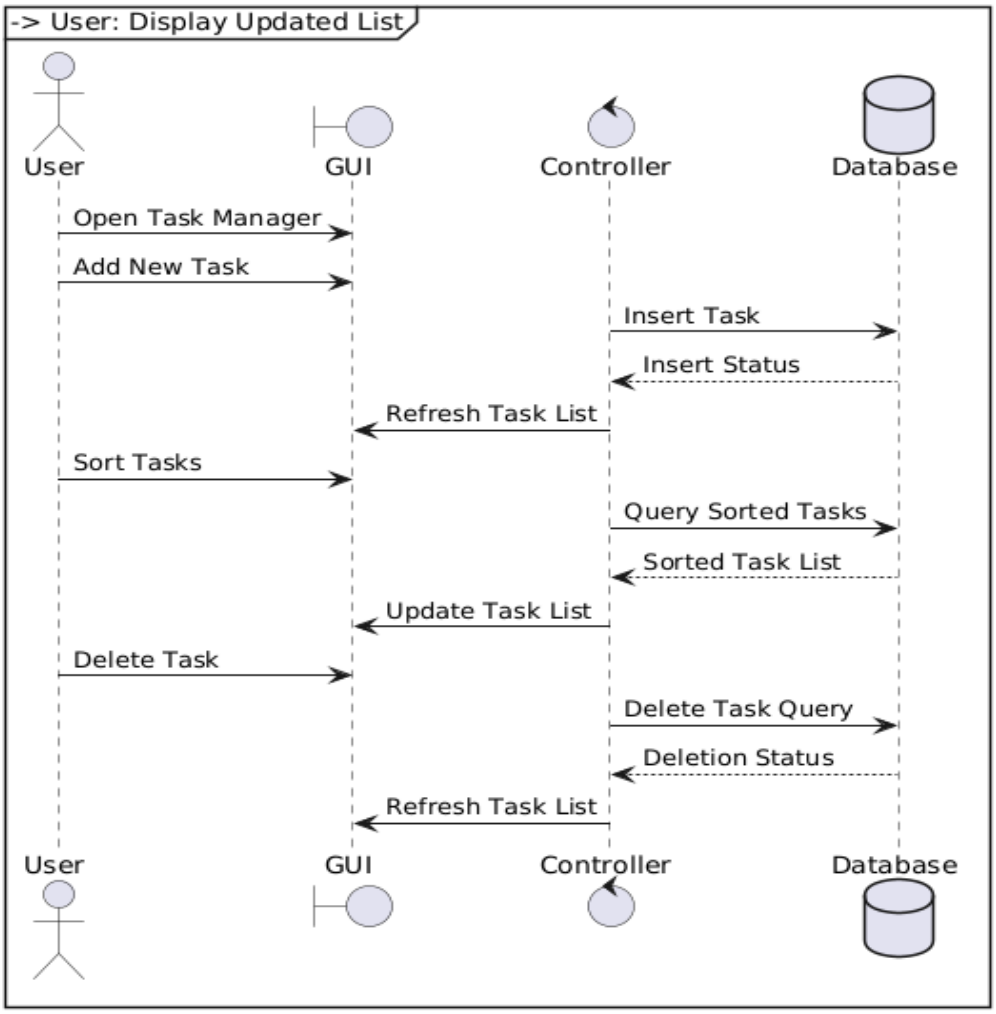

Figure 12: Task manager sequence diagram

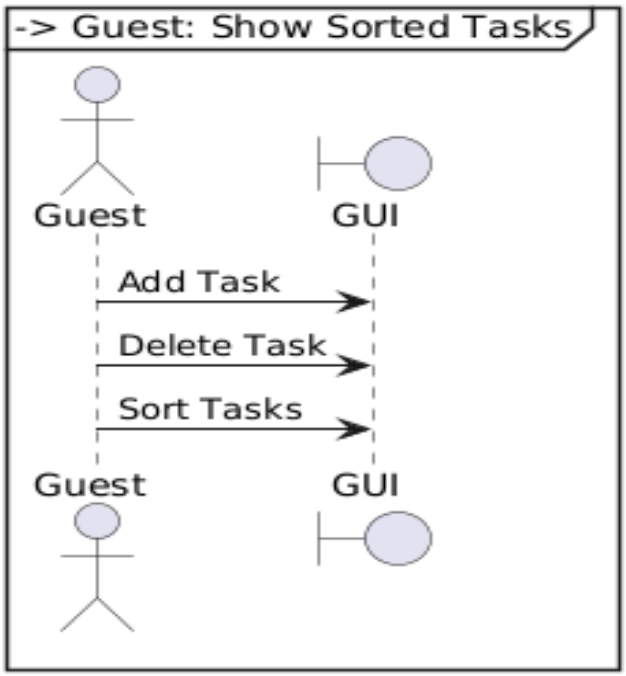

Figure 13:Guest sequence diagram

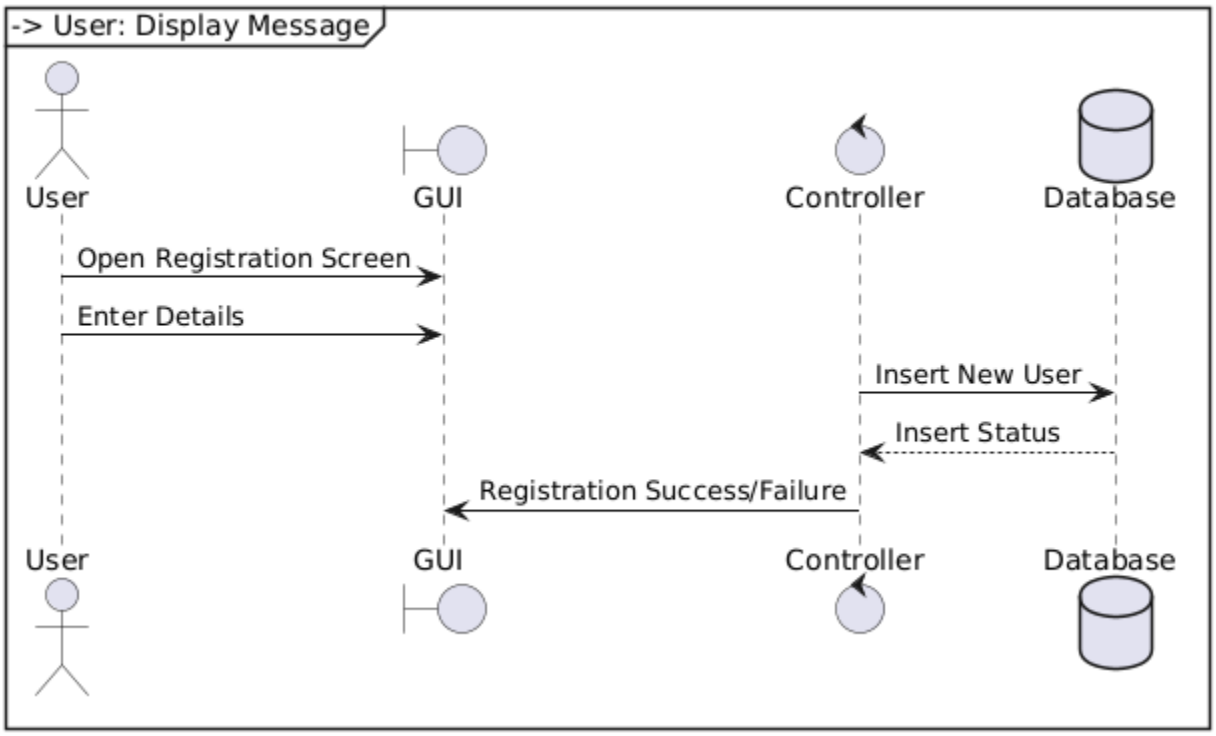

Figure 14: Registration sequence diagram

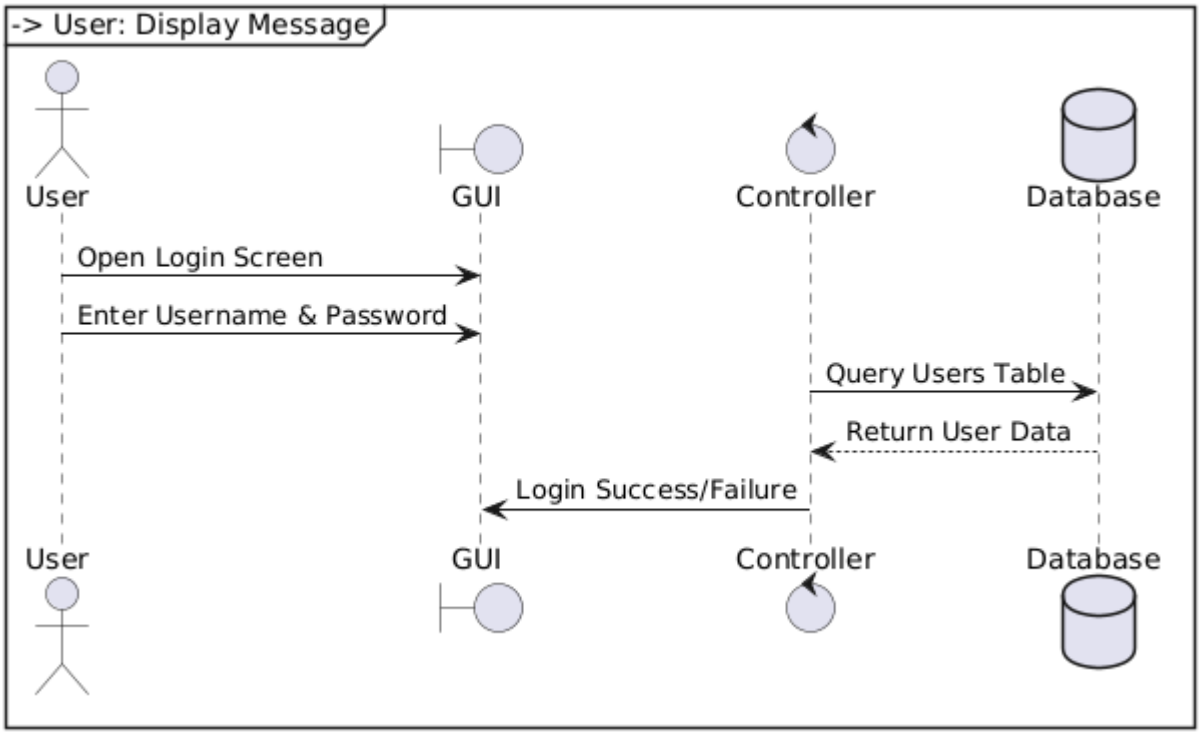

Figure 15: Login sequence diagram

8.5 **Resources:**

A description of all resources that are managed, affected, or needed by this entity. Resources are entities external to the design such as memory, processors, printers, databases, or a software library. This should include a discussion of any possible race conditions and/or deadlock situations, and how they might be resolved.

Type	**Resourses**
Operating System	Windows could work for android and ios also
Database	JDBC
Memory Storage	depends on the data amount

Table 7: resources specification

Deadlock situation	**Description**	**Solution**
Database connection problem cause of the Wi-Fi	Wi-Fi is needed to have a database connection and the tasks cannot be shown because there is no database connection.	Fix the Wi-Fi connection so the database will be connected to the system and the system can work.

Table 8: Deadlock situation

8.6 **Processing:** A description of precisely how this component goes about performing their duties and the deception and the input and output for them.

✓ **Starting Interface**

Description	**Input**	**Output**
Chose the thing that need to be done	Register Log in Continue as guest	Other interfaces

Table 9: Starting Interface

✓ **Register interface**

Description	Input	Output
For the register interface the user will be able to enter their username and password and confirm the password,	Username Password Confirm Password	The Account

Table 10: Description of (Register) component

✓ **Log In Interface**

Description	Input	Output
This is how the user login to the system by entering the correct username and password	Username Password	Tasks page

Table 11: Description of (Login) component

✓ **Tasks Interface**

Description	Input	Output
The users can add tasks, see their tasks, click them or either delete them.	Tasks Date Add them Click them Delete them	The result will be updated in the database and the user interface.

Table 12: Description of (add task) component

9. **Interface/Exports:** demonstrated in the previous sections. in components classification, definitions, and responsibilities, constraints. And, in sequence diagrams.
10. **Detailed Subsystem Design:** Previous sections provided a thorough discussion of software components, including their structure and function, as well as information flow demonstrations.

9. Other Design Features:

All the design features of the task tracker system were mentioned and detailed in the previous sections.

10. Requirements Traceability Matrix

Associated ID in SRS	Technical Assumption(s) and/or Customer Need(s)	Functional Requirement	System Component
Common functionalities	The user can log in by providing a valid username and password.	Login	Login
	The user can register with a username (not taken or empty) and password.	Register	Register
	The user can continue as a guest without logging in, but their tasks will not be saved in the database.	Guest Mode	Continue as Guest
User 1: Registered User	The user can create tasks by specifying the task name and date.	Add a task	Task Management System
	The user can view tasks and mark them as completed. Completed tasks will be saved in the database.	View/Mark tasks as complete	Task Management System
	The user can sort tasks by date.	Sort tasks	Task Management System
	The user can delete tasks from the task list.	Delete tasks	Task Management System
User 2: Guest User	The guest user can create tasks by specifying the task name and date, but these tasks will not be saved in the database.	Add a task	Guest Task Management System
	The guest user can view tasks and mark them as completed during the session.	View/Mark tasks as complete	Guest Task Management System

Table 13: Requirement Traceability matrix