

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

CS 411 – Software Engineering
Term 1 – 2023/2024
Group 3

Software Requirements Specification

Task tracker



Instructor Dr. Muawiya Abdul majid

Submission date: 10 November 2024

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	4
1. Introduction.....	5
1.1 Purpose.....	5
1.2 Scope.....	5
1.3 Definitions, Acronyms, and Abbreviations.....	6
1.4 References	6
2. Overall description.....	7
2.1 Product perspective	7
2.2 Product functions	8
3. Specific requirements	9
3.1 External interface requirements	9
3.1.1 User interfaces	9
3.2 Non-Functional Requirements.....	10
3.2.1 Performance Requirements.....	10
3.2.2 Design Constraints.....	10
3.2.3 System Software Attributes	11
3.2.3.1 Usability.....	11
3.2.3.2 Security	11
3.2.3.3 Maintainability.....	11
3.2.3.4 Availability	11
3.2.3.5 Portability	11

Table of figures

Figure 1 Provided Functionalities	5
---	---

Table of tables

Table 1 Terminologies Definitions	6
Table 2 Acronyms Definitions	6
Table 3 Add tasks interface structural description.....	9
Table 4 performance properties and their measurements	11
Table 5 constraints and their description along with source.....	11

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

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:

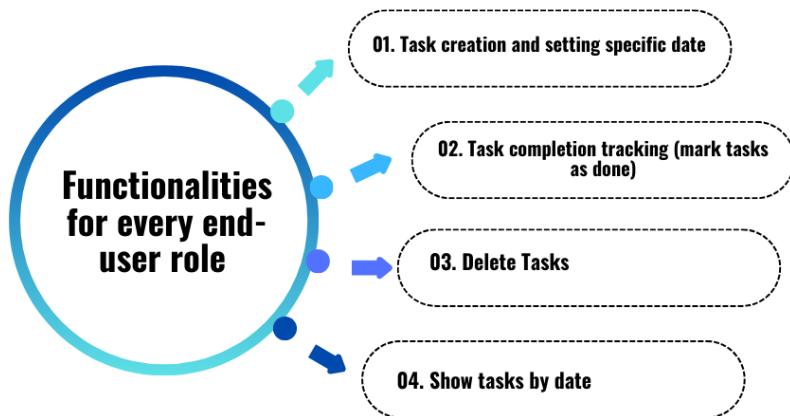


Figure 1 Provided Functionalities

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. Overall description

This section illustrates the product via two subtopics, product perspective as well as product functions.

2.1 Product perspective

The Task Tracker Management application was created to help any end user manage their tasks, especially in light of many distractions that might make them forget about work that must be done by a specific day. Additionally, this system has different interfaces, and each interface represents a service to the user.

System users will have the ability to add tasks and assign them to a specific date. To differentiate between each task's priority and make tasks that are close to being delivered or completed more obvious to the user, as well as there is a feature to order tasks by date.

Moreover, once the user has completed tasks, they can mark them as completed and they will be marked as checked, even delete them.

All the tasks will be saved with the help of using SQL database unless the user deletes them by clicking on each task that are not needed to be in the to do list screen anymore.

2.2 Product functions

Our system “Task track” will provide various features that help the user to manage their tasks and responsibilities based on their importance, a task management solution designed to improve organization and productivity. With emphasis on simplifying everyday tasks.

Task Creation and Date Setting:

With the help of this feature, users can recognize tasks in the program and give them deadlines. Task information and the deadline for completion can be entered by users, this aids in task prioritization and organization according to due dates.

Tracking Task Completion (Mark task as done):

After completing a task, users can mark it as finished. This feature gives customers a clear picture of what tasks are still unfinished or have been completed, helps them keep organized, and tracks their progress.

Delete Tasks:

Users can remove tasks that are no longer required or important. By enabling the elimination of unnecessary activities from the task list, this feature aids users in keeping their task management system updated and specific.

Display Tasks by Date:

This feature lets users see tasks arranged according to their due dates. Filtering and displaying tasks according to dates, users can more easily concentrate on jobs that are nearing their due dates or require urgent attention. It helps users efficiently prioritize their work by giving them a chronological perspective of their assignments.

3. Specific requirements

3.1 External interface requirements

This section lists all the methods you can use the program and illustrates the advantages each one provides. It works like a map, showing you where to go in the application's different sections and what happens when you click on or use each feature.

3.1.1 User interfaces

Add task interface

Here, users can add/create new tasks. Once the user has written the task name and click save, the task will be created and added to the application.

Field	Format	Level	Input/Output
Task name	Text	required	input
Task date	Number	required	input

Table 3 Add tasks interface structural description

Assigning a date to each task

Every time user adds a task, they must choose a specific date, otherwise a default date will be assigned.

Delete task

Users will have the ability to delete/remove no longer required tasks from the task list.

Show tasks by date

Tasks are going to be filtered and displayed based on their due dates, from nearest to farthest.

Mark done

In the mark done interface, once the user finishes any task, he/she can mark it as completed/finished.

3.2 Non-Functional Requirements

The non-functional requirements, such as the system software attributes, design details, and performance requirements will be covered in detail in this part.

3.2.1 Performance Requirements

Property	Measurement
Response time	The system should respond to user actions (e.g., task creation, deletion) within a maximum of 2 seconds.
Throughput	It can support up to 1000 jobs without degradation, thus allowing for heavy task management use cases to run smoothly.
Availability	The system shall be up 99.9% and therefore always available to the user during the day.
Fault Recovery	The system should automatically recover in the event of system crashes or otherwise unforeseen failures, restoring user data in less than 10 seconds.

Table 4 performance properties and their measurements

3.2.2 Design Constraints

The design constraints will undoubtedly have an impact on the overall architecture and evolution of the system. The limits imposed on the Buzz application are largely organizational and product constraints.

Constraint	Description	Source
Timeline Constraints	The project deliverables must be submitted within the predefined deadlines to ensure timely delivery.	Organizational
Budget Constraints	The project has to be developed on the mutually agreed upon and approved budget and cannot involve costs other than those planned in the beginning.	Organizational
Programming Language	The software has to be written in any particular programming language; Python or Java are options.	Organizational
Database Constraints	The data of the tasks should be persisted in a relational or document-based database that supports efficient querying.	Organizational
Platform Constraints	This needs to run on all desktop and mobile platforms since it aims to target a wide audience.	Product
Hardware Limitations and Compatibility Constraints	The system will be able to run on devices with minimal processor capability and memory	Product

Table 5 constraints and their description along with source

3.2.3 System Software Attributes

3.2.3.1 Usability

The task management application should remain user friendly: the users of the application should be allowed to add, edit, and delete tasks without much hustle. The interface should be intuitive, with instructions present but not being too complex for end-users of any level of computer experience. The filtering and status tracking of tasks should be easy to learn and perform.

3.2.3.2 Security

Although this is a single-user project management tool, this does not mean that security aspects can be fully avoided. In case there is a need to store the credentials of the user or other sensitive data regarding the tasks, these should be encrypted in order to avoid unauthorized access. Any kind of user data should be stored and protected securely. Furthermore, task information should be available only to the authenticated user; this will further enable expansion easily to multi-users later.

3.2.3.3 Maintainability

The system should be regularly updated and maintained to ensure an optimal run of programs. Bug reporting and performance monitoring shall be integrated in such a manner that developers can identify issues as quickly as possible. Any updates or fixes must be implemented without disrupting the user much. Maintain well-documented, modular codebase with ease of future implementations and updates, feature expansions.

3.2.3.4 Availability

It should be up at any time, meaning users should have access to their tasks at any given time. The system is not supposed to experience much downtime, and there should be backup mechanisms that allow users not to lose their data. The system should recover fast with seamless restoration of all task data when there is an unexpected failure or crash.

3.2.3.5 Portability

The task management program should be portable on multiple platforms, including both desktop and mobile environments. It will run seamlessly on several operating systems: Windows, macOS, iOS, and Android. Cross-platform technologies, such as web technologies or platform-agnostic programming languages, make the application accessible to users without requiring configuration changes for the device in use.