Software Requirements Specification

for

TrackIt

Version 1.0

Prepared by

Group #: 9 Group Name: localghosts:3000

|  |  |  |
| --- | --- | --- |
| Sarthak Kalankar | 210935 | sarthakk281103@gmail.com |
| Maurya Aryan Swaminath | 210595 | [aryanmaurya383@gmail.com](mailto:aryanmaurya383@gmail.com) |
| Saugat Kannojia | 210943 | [saugatkannojia@gmail.com](mailto:saugatkannojia@gmail.com) |
| Harsh Oza  Padulkar Rohan Ravikumar  Sujal Lalawat  Depanshu Sahu  Rashi Gupta  Shrilakshmi S K  Chirayush Mohanty | 210411  210689  211066  210316  190690  211012  210289 | [2003ozaharsh@gmail.com](mailto:2003ozaharsh@gmail.com)  [rohan1.padulkar@gmail.com](mailto:rohan1.padulkar@gmail.com)  [sujallalawat223@gmail.com](mailto:sujallalawat223@gmail.com)  [depanshusahu057@gmail.com](mailto:depanshusahu057@gmail.com)  [grashi253@gmail.com](mailto:grashi253@gmail.com)  [shrilakshmisk13@gmail.com](mailto:shrilakshmisk13@gmail.com)  chirayushimo@gmail.com |

|  |  |
| --- | --- |
| Course: | CS253 |
| Mentor TA: | Abhinav |
| Date: | 27/01/2023 |

Content

Contents ii

Revisions ii

1 Introduction 1

1.1 Product Scope 1

1.2 Intended Audience and Document Overview 1

1.3 Definitions, Acronyms and Abbreviations 1

1.4 Document Conventions 1

1.5 References and Acknowledgments 2

2 Overall Description 2

2.1 Product Overview 2

2.2 Product Functionality 3

2.3 Design and Implementation Constraints 3

2.4 Assumptions and Dependencies 3

3 Specific Requirements 4

3.1 External Interface Requirements 4

3.2 Functional Requirements 4

3.3 Use Case Model 5

4 Other Non-functional Requirements 6

4.1 Performance Requirements 6

4.2 Safety and Security Requirements 6

4.3 Software Quality Attributes 6

5 Other Requirements 7

Appendix A – Data Dictionary 8

Appendix B - Group Log 9

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# 

# Introduction

## Product Scope

You would have seen your parents or elders keeping track of how much money they spent on monthly or yearly basis to get an idea of their monthly or yearly savings. Keeping this in mind we have made a web app to ease the hassle of keeping the record or tracking physically or manually typing the entries in excel or similar apps and then using formulas to calculate the total sum spent and all. Using the app any common person whether adult or student will be able to upload their entry on the web app and get the analytics of how much amount they spent and on what like Food, Travel, Groceries, Shopping, etc. monthly.

One of the other benefits of this app would be that the person can access it from any device as it’s a web app and will save time in making entries and keeping records. They won’t have to keep saving the receipts or tickets for their purchases. They would be able to directly enter it. We hope that **TrackIt** will help most people in keeping a track of their spending.

## Intended Audience and Document Overview

The document is mainly intended for the product end-users (to understand product requirements and give suggestions) and the developers (to build the product as per requirements). The user should start with the introduction, then he/she could read the product overview section (2.1) followed by specific requirements section (3). The developers would be mainly interested in specific requirements section (3) and non-functional requirements section (4).

## Definitions, Acronyms and Abbreviations

**1.3.1 Abbreviations**

● IITK- Indian Institute of Technology, Kanpur

● OTP- One Time Password

● SQL- Structured Query Language

● UI- User Interface

● DB - Database

## Document Conventions

● General Text: Font-Arial Size-12

● Heading: Font-Arial Size-18

● Sub headings: Font-Arial Size-14 Style-Bold

● Margin-1’’

## References and Acknowledgments

None

# Overall Description

## Product Overview

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.

TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level.>

## Product Functionality

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. These can be at the level given in the project description.>

TO DO:

1. Provide a bulleted list of all the major functions of the system

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software). You can be creative here to some degree.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

TO DO: Provide a short list of some major assumptions that might significantly affect your design.>

# Specific Requirements

## External Interface Requirements

### User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. For your project, you only need to be concerned with the main thermostat (not the mobile app) and can use the graphic from the project description as the basis for your user interface..

TO DO: Provide the graphic for the user interface and provide a basic description as to how users will interact (e.g. menus, etc.).>

### 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. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. This can simply be a list of the devices you must interact with at this point. >

### Software Interfaces

<Describe the connections between this product and other specific software components (in your case, just the mobile app that can send commands).>

## Functional Requirements

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions.*

### F1: The system shall …

### <Functional Requirement or Feature #2>

…

## Use Case Model

TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.

### Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram

**Author –** Identify team member who wrote this use case

**Purpose** - What is the basic objective of the use-case. What is it trying to achieve?

**Requirements Traceability –** Identify all requirements traced to this use case

**Priority** - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is deployed

**Preconditions** - Any condition that must be satisfied before the use case begins

**Post conditions** - The conditions that will be satisfied after the use case successfully completes

**Actors** – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

**Exceptions** - Exceptions that may happen during the execution of the use case

**Includes** (other use case IDs)

**Notes/Issues** - Any relevant notes or issues that need to be resolved

### Use Case #2

…

# Other Non-functional Requirements

**4.3.1   Flexibility**

The web app is to be designed so that it is flexible. Also, it allows the incorporation of new requirements in any module of the system. The application will be designed in a modular format such that any future changes (additions or deletions) will be easily incorporated into the system. Each functionality is divided into several folders so make it flexible.

**4.3.2   Portability**

The application will be easily portable on any window-based system. The website front is designed using React JS, making it a responsive and progressive web app, which ensures that the application can run on different platforms.

**4.3.3   Maintainability**

The architecture, design, implementation, and documentation of the product should minimize the maintenance costs of the software system. The maximum person-time required to fix a security defect (including regression testing and documentation update) must not exceed two persons per day.

Otherwise, the software system must be taken offline or the offending feature disabled which should only be done in case of emergencies. The average person-time required to make a minor enhancement (including testing and documentation update) must not exceed one person per week.

**4.3.4   Usability**

The frontend of the software application is designed to be user-friendly such that the user can utilize the system effectively. It will be based on the well-known principle of usability -As Simple as Possible. Also, after logging in the user can view his account without logging in again even after the website is closed and reopened.

**4.3.5   Security**

The application is password protected and also any upgradation of new entries and deletions is done by only privileged users. All the users will be able to log on to their accounts using their CC username and custom password (created by them). This password will be saved in an encrypted(hashed and salted) format in the database to ensure security. Also, the password is ensured to be strong enough while signing up, so as to ensure if someone gets an encrypted password, it will too difficult to decrypt.

**4.3.6   Reliability and Availability**

Web app will store data in mongoDb Altas Cloud which is realiable and always available

The system will always be running  and ready to carry out its task whenever the user needs it.

Also the time to exchange data between web app and database is less which will give users a smooth experience.

# Other Requirements

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

|  |  |
| --- | --- |
| **Abbreviation** | **Abbreviation definition** |
| **hashed** | It is a method of encryption of data such it converts in into alpnumberic string constant length. |
| **Salting** | A string which is added to data during its encryption to make hashed string difficult to decrypt. |
| **Server** | It is a computer that can be accessed remotely. |
| **Database(DB)** | It is an organized collection of structured information stored in the server. |
| **User Interface(UI)** | By which the user and the computer system interact , by use of input and output devices. |
| **OTP** | One Time Password , string which is send to user for verying user. |

Appendix B - Group Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. No.** | **Date** | **Timings** | **Venue** | **Description** |
| 1. | 06/01/2023 | 6:00 PM –7:00 PM | CCD | Brainstormed various project ideas |
| 2. | 11/01/2023 | 6:00 PM –7:00 PM | CCD | Finalised our project idea |
| 3. | 16/01/2023 | 6:00 PM –7:00 PM | CCD | Discussed the features, requirements and softwares we will work on. |
| 4. | 17/01/2023 | 6:30 PM –7:00 PM | Microsoft Teams | Discussed our project with our mentor Abhinav |
| 5. | 23/01/2023 | 6:00 PM –7:00 PM | CCD | Distributed work and decided UI of the project |
| 7. | 24/01/2023 | 6:00PM-  7:00PM | Microsoft  Teams | Discussed about the specific work each is doing in the SRS document. |
| 8. | 25/01/2023 | 6:00PM-6:30PM | CCD | Discussed about SRS document. |