Software Requirements Specification

for

TGHM

Version 1.0 approved

Prepared by
Abhishek Gandhi | 19CS10031
Shashvat Gupta | 19CS30042
Sajal Chhamunya | 19CS10051

Indian Institute of Technology Kharagpur

18 March 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
TGHM	18 march 2021	First version	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to build travel in good health management system, which is a food delivery system aimed at delivering food to train passengers. It would take orders from passengers inside a running train or before passenger boards a train, and provide them food they order from nearby restaurants. This document specifies the software requirements for version 1.0.

1.2 Document Conventions

This document was created based on the IEEE template for System Requirement Specification Documents.

Abbreviation - TGHM	Travel in Good Health Management
Abbreviation - ER	Entity - Relationship
Bold	Noun

1.3 Intended Audience and Reading Suggestions

This project is a prototype for travel in the good health management system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the management team of the product,

the concerned restaurants, and agents, and as well as to the passengers.

1.4 Product Scope

The purpose of the travel in good health management system is to provide a better alternative to the unsatisfactory food provisions provided by the pantry car while traveling in a train by creating a convenient and easy to use application that delivers food to the customer's train. We will have a database server supporting thousands of **restaurants** in all possible major **cities**, covering hundreds of trains and **train** stations. Above all, we hope to bring major innovations and revolutionize the food sector under the **railway** industry.

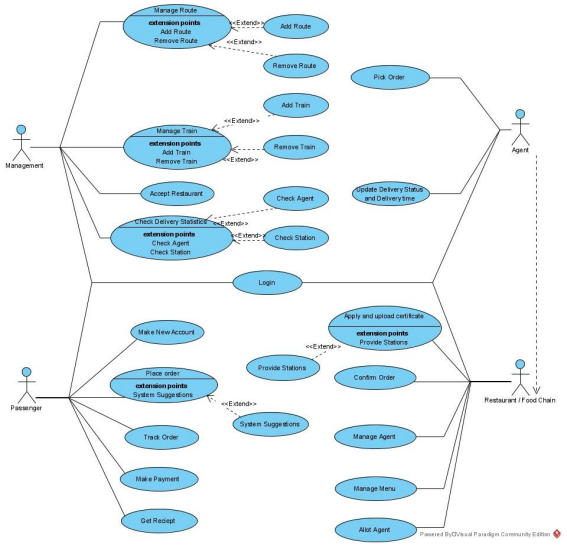
1.5 References

- https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database
- https://gephi.org/users/gephi_srs_document.pd

2. Overall Description

2.1 Product Perspective

Passengers traveling via train are often not satisfied with the food services provided by the pantry car due to limitations of choices. This product creates a bridge between railway passengers and **restaurant/food stores**. It would take orders from passengers inside a running train or before a passenger board a train and provide them food they order from nearby **restaurants**.
The product being discussed is a new self-contained project. Its origin can be traced back to our college professors under whose guidance this is being implemented.



The diagram shows the layout of travel in good health management system –ER model

2.2 Product Functions

- **Restaurant, Customer** can register on our site, Restaurant will submit an authentication certificate
- **Management, Restaurant, Customer** can log in through the login portal
- **Management** can add remove **Train, Station**, accept or deny **Restaurant** application
- **Restaurants** that can deliver to the next subsequent stations are displayed.
- **Customers** can place a food order.
- **Food Orders** are sent to Restaurants.
- **Restaurants** allot delivery agents.
- **Customers** can track the status and location of the food order.

2.3 User Classes and Characteristics

The user classes for this product include customers, restaurants and delivery agents, and management.
Each user class needs basic information on running the user interface to interact with the product.

Customer: This class needs basic web surfing skills to interact with the product. They are given basic permissions related to ordering. The customer should be able to do the following functions:

- Make a new account
- Login
- Place an order, our system will suggest few changes in filter taking into account their current selections
- Download Receipt
- Track order
- Cancel an existing order

Restaurants: This class needs basic web surfing skills to interact with the product. They are given restaurant manager permissions. The Restaurant should be able to do the following functions:

- Get a list of orders
- Accept, Reject orders
- Allot Agents
- Edit, Add menu

Delivery Agents: This class needs basic web surfing and map using skills to interact with the product. They are given Agent Permissions with access to change delivery status and constant location tracking for giving status updates.

Management: This class needs basic web surfing and map using skills to interact with the product. Management will manage the entire system. They have been permitted to:

- Add/Remove Train
- ADD/Remove Station
- Accept/Deny Restaurant application
- Check order statistics for all stations and Agents.

The most important User classes needed to be satisfied are the Customer user class as well as the Restaurants user class.

2.4 Operating Environment

The application runs in the latest version of **Chrome or Firefox browser** on **Windows**, **Linux,** and **Mac** as well as **Android**.

2.5 Design and Implementation Constraints

Some of the factors that limit the option available to developers are:

- Hardware limitations: Unavailability of public server for hosting
- **Database** to be used to store all information.

2.6 User Documentation

A user manual with a pictorial flow of steps on how to use the product will be provided in the product itself. The manual will have sections for different kinds of users of the product, which includes **Customers, Restaurant Managers,** and **Delivery Agents**.

2.7 Assumptions and Dependencies

The product will be a **website** hence it will require a server to host the code. The backend needs a vast **database** to store all data. On the user's side, they will need a web browser to run the application.

3. External Interface Requirements

3.1 User Interfaces

The format of the GUI of the application is in a **website** format and can be navigated through using a keyboard, mouse on computers, and by touch in android. The GUI contains standard buttons and forms for easy control by **computers** and **android phones**.

3.2 Hardware Interfaces

Hardware requirements include having a computer or an android device, on which one can surf the **website** and appropriate tools (**Mouse** and **keyboard** for **computer**) to facilitate navigation within the product GUI.

3.3 Software Interfaces

The product requires a **web browser** installed in the system and software capable of

internet surfing. The web browser can access the product **website** and navigate the same.

3.4 Communications Interfaces

The product needs a constant network connection and a web browser in the system to communicate with restaurants, agents, and other components of the system. **HTTP** communication standards will be used throughout.

4. System Features

4.1 Database management system

4.1.1 Description and Priority

The travel in good health management system maintains information on **trains** and their **routes, passenger seat number, restaurants** based around every **train station** and their **menus**, and orders. This feature has the highest priority (9) as this is the main requirement for any other feature to work.

4.1.2 Stimulus/Response Sequences

1. Add and remove **trains**, along with editing their **routes**, timings, etc.
2. Add or remove **stations** from the existing database.
3. Add and remove **restaurants** and edit their menus along with adding and removing **food items**.
4. Upon receiving input about the **passenger's train** details, show restaurants of the next stations.
5. Show menus of the **restaurants** that can deliver at a particular station.
6. Statistics on orders carried out **per agent** or **restaurant** or **train station** will also be available.

4.1.3 Functional Requirements

REQ-1: Database for storing all the information.

4.2 Placing orders

4.2.1 Description and Priority

Accept orders by **customers** and deliver the order by an **agent**. Meanwhile, the user can see the progress of his/her **order**. This is the main functionality as applicable to a user and hence has a high priority (8) but lower than the first feature as this feature can work only if the first feature is implemented.

4.2.2 Stimulus/Response Sequences

1. Upon receiving an **order**, the **order** is forwarded to the concerned restaurant.
2. The **restaurant** confirms the order and an **agent** is assigned to carry the order out.
3. If the **restaurant** Denies the order then-new suggestions with an apology message are given to the **customer**.
4. Each such stage is updated on the main database and can be seen by the user in real-time.
5. Suggestions if constraints set by the user give fewer results will also be given.

4.2.3 Functional Requirements

REQ-1: a server to communicate with the users, restaurant and the agent.

4.3 User interface

4.3.1 Description and Priority

A User interface, where one would be able to create an account, log in to one's account and then access all the **order** options. A payment portal will be available too. As this will be the main face of the product, it has a high priority (7) and will require attention over its aesthetics too.

4.3.2 Stimulus/Response Sequences

1. User authentication with password encryption.
2. Secured payment portal so that pre-payment for any order is also available.

4.3.3 Functional Requirements

REQ-1: A login and a sign-up page, which includes an I forgot my **password** button and uses **phone number** verification after signing up.

REQ-2: **Database** for user's username and passwords.

REQ-3: A **payment portal** page, including a merchant affiliation.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system does require a **web browser** and a stable **internet connection**. Internet Connection is crucial to put orders as well as pay for them. The amount of data that can be stored in the backend depends on the database constraint available.

5.2 Safety Requirements

The product comes with downsides of a possible information leak in case of breakdown of **server** or intruder attacks. Any other harm or damage is much less likely to occur and good quality **food** must be ensured so that we provide good and healthy food to our **customers.**

5.3 Security Requirements

This product requires security measures to keep it from malfunctioning. Appropriate security measures to lock out intruders or other users from the administrator permissions as well as keeping intruders out from **restaurant manager** permissions is absolutely necessary. Security measures to secure **customer ID** and **agent ID** are also required to protect the privacy of the **users**.

5.4 Software Quality Attributes

The product provides the users with both simple and advanced features. Due to its well-designed and easy-to-use interface, it can be used by both experts and typical users. The product is easy to maintain as well as portable. The user is required to know how to surf the web.

5.5 Business Rules

Some roles included in the product include:

- **Restaurant Managers** can create and update **menus**, assign **agents** and accept **food orders**.
- **Customers** can view **menus** and place orders.
- **Agents** can view customer status and can provide their location to facilitate delivery.

6. Other Requirements

- **Database Requirement:** The project requires a database to store, update and fetch all information regarding Stations, Trains, Restaurants, Agents and Customers

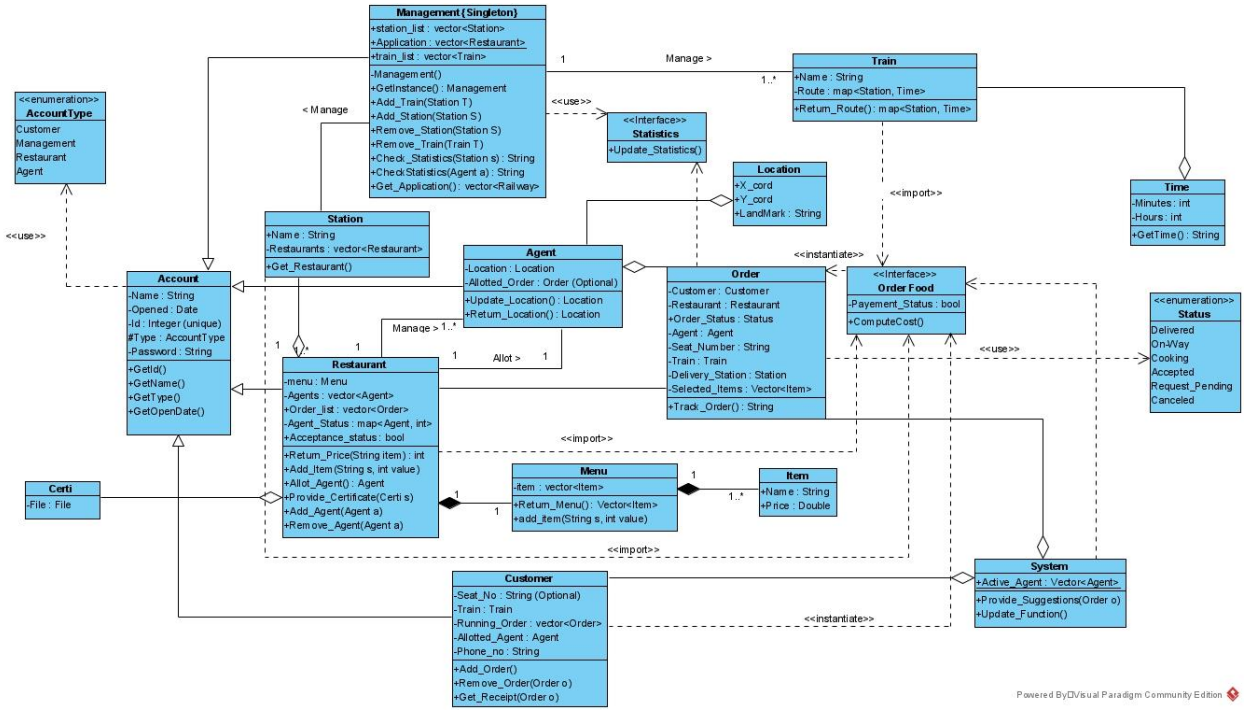
Appendix A: Glossary

Derived Class: A class that is created from an existing class. The derived class inherits all members and member functions of a base class. The derived class can have more functionality with respect to the Base class and can easily access the Base class. A Derived class is also called a child class or subclass.

Static Data Members: Members of class that are the same in all instances and can be used initialized outside the class.

Singleton Class: a singleton class is a class that can have only one object (an instance of the class) at a time. After the first time, if we try to instantiate the Singleton class, the new variable also points to the first instance created.

Appendix B: Analysis Models



Class Diagram Representing OOP structure used in this project.

Appendix C: To Be Determined List

Few things which we need to determine beforehand are

- Qualification criteria for a restaurant.
- Password Strength Requirement.