**Yaşar University**

**Spring, 2023 - 2024**

**SE 2224 - Software System Analysis**

**Final Project Report: Software Requirements Specifications Document (SRS)**

|  |  |
| --- | --- |
| **Student Name:** | **Lara Özduman** |
| **Student No:** | **22070006044** |
| **Department Name:** | **Software Engineering** |
| **Course Section No:** | **Section 1** |

This template is prepared based on the IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998).

**Table of Contents (Do not change the Section Names!)**

[**1** **Introduction** 3](#_Toc165879141)

[**1.1** **Purpose** 3](#_Toc165879142)

[**1.2** **Scope** 3](#_Toc165879143)

[**1.3** **Definitions, acronyms, and abbreviations** 3](#_Toc165879144)

[**1.4** **References** 3](#_Toc165879145)

[**1.5** **Overview** 3](#_Toc165879146)

[**2** **Design and Implementation Constraints** 3](#_Toc165879147)

[**3** **Specific Requirements** 3](#_Toc165879148)

[**3.1** **Functional Requirements** 4](#_Toc165879149)

[**3.2** **Performance Requirements** 4](#_Toc165879150)

[**3.3** **Software System Attributes** 4](#_Toc165879151)

[**3.4** **Use Case Analysis** 4](#_Toc165879152)

[**3.4.1** **Actors** 4](#_Toc165879153)

[**3.4.2** **Scenarios** 4](#_Toc165879154)

[**3.4.3** **Use Case Forms** 4](#_Toc165879155)

[**3.4.4** **Relationships among Actors and Use Cases** 5](#_Toc165879156)

[**3.4.5** **Use Case Diagram** 5](#_Toc165879157)

[**4** **Behavioral Models** 5](#_Toc165879158)

[**4.1** **Sequence Diagram** 5](#_Toc165879159)

[**5** **Structural Models** 5](#_Toc165879160)

[**5.1** **Class Diagram** 5](#_Toc165879161)

[**6** **Process Modeling** 5](#_Toc165879162)

[**6.1** **Data Flow Diagram (DFD)** 5](#_Toc165879163)

[**7** **Graphical User Interface(s) (GUIs)** 5](#_Toc165879164)

[**8** **Conclusion and Future Work** 5](#_Toc165879165)

# **Introduction**

## **Purpose**

Purpose of this SRS document is to document and provide a description of the Java program with it requirements, functions and constraints as the SRS document provides a better understanding of the system and the purpose of it.

## **Scope**

a) This software product is called “MyFavoriteSites”

b) MyFavoriteSites is a Java GUI program that allows users to keep track of the countries that they have visited as they are able to add, delete ,edit , display their visit details and share their visit details with a friend.

c) The goal of this system is to provide an tracking application that stores their travelling information and has the ability to share it with a friend. With the racking system it helps to be able to look into their ratings and best features of the visit to plan accordingly to their future visits.

## **Definitions, acronyms, and abbreviations**

SRS: Software Requirements Specification

GUI: Graphical User Interface

SQL: Structured Query Language

JDBC: Java Database Connectivity

MySQL: A database management system

Java: A programming Language

Intellij IDEA: A java integrated development environment

Visual Paradigm: A software tool for modeling and designing systems

JFrame: A class in java Swing Library which is used to create a new window

JButton: A class in Java Swing Library which is used to create a new button

CRUD: Create, read, update, delete

## **References**

Java Swing Documentation: <https://docs.oracle.com/javase/8/docs/technotes/guides/swing/>

GeeksForGeeks: <https://www.geeksforgeeks.org/>

Stackoverflow: <https://stackoverflow.com/>

## **Overview**

This SRS contains the sections:

**Design and implementation constraints:** This part describes the constraints of the project including hardware constraints, Interfaces and other applications, specific technologies tools and databased that were used and language requirements.

**Functional requirements:** Details of the functionalities that the software should perform

**Non-Functional requirements:** Portrays system attributes

**System Features:** Shows the features of the system and use cases.

# **Design and Implementation Constraints**

**Hardware Limitations:**

* The system must run on a standard computer with a minimum 3GB RAM.

**Interfaces and Other Applications:**

* This application interfaces with MySQL to store and fetch data
* It uses JDBC to connect and use CRUD operations

**Specific Technologies, Tools and Databases that were used:**

* IntelliJ IDEA: This application is developed with IntelliJ IDEA version 2022.3.2
* MySQL: This application uses MySQL version 8.0
* Java: This application is programmed with Java language
* Visual Paradigm: Version 17.1 Visual Paradigm will be used as a UML modeling tool

**Language Requirements:**

* Programing language of this application is Java
* SQL will be used for databases  
     
  **Specific Requirements**

## **Functional Requirements**

* The system shall allow the user to create a new account with a username and password.
* The system shall allow the user to log into their account with username and password.
* The system shall allow the user to add a new visit by entering information such as country name, city name, year visited, season, best feature, comment and rating.
* The system shall allow the user to edit an existing visit.
* The system shall allow the user to delete an existing visit.
* The system shall display a list of users visits
* The system shall allow the user to share a visit with a friend
* The system shall allow the user to view shared visits by their friends
* The system shall provide a graphical user interface (GUI)
* The system shall provide a form for inputting visit details
* The system shall display error messages for invalid inputs
* The system shall store user information in the MySQL database

## **Performance Requirements**

Write 3 nonfunctional requirements from the performance category.

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

a) The system shall optimize database queries and reduce system resource usage

b) The system shall load the main frame within 5 seconds of the user logging in.

c) The system shall provide meaningful error messages for invalid inputs to users to understand the error better

## **Software System Attributes**

For each category given below, write a nonfunctional requirement that can be suitable for your project. In total, there should be 5 nonfunctional requirements in this subsection.

Reliability: The system shall detect and handle any errors in the code by sending out informative alerts in order to discover issues

Availability: The system maintenance shall not take more than 2 hours per month

Security: The system shall implement a role based access to limit accessing administrative functions

Maintainability: The system shall adhere to the coding standards in order to guarantee code readability

Usability: The system shall have an easy to use user interface with clear instructions

## **Use Case Analysis**

### **Actors**

User: The primary actor who uses with the system to manage and keep their visits stored

User’s friend: An actor who can share visits with the user and see the users shared visits

### **Scenarios**

Scenario 1:  
Scenario name: addDestination

Participating actor instances: Alice: User

Flow of events:

1. Alice has recently visited a new destination and wants to add it to her list of destinations
2. Alice logs into the system with her username and password
3. Alice navigates to the “Add” section in the system
4. Alice enters the details of the destination which includes country name, city name, year visited, season, comment, best feature, and rating.
5. Alice confirms and submits the form by clicking the add button
6. The system processes the input and stores the information in Alices profile

Scenario 2:

Scenario name: deleteDestination

Participating Actor Instances: John: User

Flow of events:

1. John decides to remove a destination from his list of destinations
2. John logs into the system with her username and password
3. John navigates to the “Delete” section in the system
4. John enters the visit ID of the destination that he wants to delete
5. John confirms and clicks the delete button
6. The system processes the input and deletes the information from Johns profile

### **Use Case Forms**

1. Add Destination

Use Case Name: Add New Location

Participating actors: User

Description: This use case describes the process of adding a visited location to the user’s profile

Trigger: User selects the “Add” button on the main frame

Preconditions: User must be logged into the system

Normal Flow:

1. User clicks the “Add” button
2. Add panel opens
3. User enters the required information such as country name, city name, year visited, season, comment, best feature, and rating into the Add panel
4. User submits the form by clicking the Add button
5. Successful submission message is displayed

Postconditions: The location is added to the database and user’s profile

Exceptions:

* + If user does not enter the necessary information an error message is displayed

1. Delete Destination

Use Case Name: Delete Visit

Participating Actors: User

Description: This use case describes the process of deleting an existing visited location from the user’s profile

Trigger: User selects the “Delete” button

Preconditions: User must be logged into the system

Normal Flow:

1. User enters the visit ID of the visit that they want to delete
2. User selects the “Delete” button on the delete panel
3. Successful delete message is displayed

Postconditions: The selected destination is removed from the database and the user’s profile

Exceptions

* + If user enter a visit ID which is not in the database an error message is displayed

1. Display Destination

Use Case Name: Add New Location

Participating actors: User

Description: This use case describes the process of displaying all the visited locations of the user

Trigger: User selects the “Display” button on the main frame

Preconditions: User must be logged into the system

### **Relationships among Actors and Use Cases**

1. User with Login

Relationship: To gain access to the system the user initiates the login use case

Description: The user enters their username and password to login

1. User with Add Destination

Relationship: The user interacts with the use case to add a destination to their profile

Description: User enters the information of the destination

1. User with Delete:

Relationship: The User interacts with this use case to remove a destination from their favorites list.

Description: The User selects a destination from their list and confirms the deletion.

1. User with View Favorite Destinations

Relationship: The User interacts with this use case to view their current list of favorite destinations.

Description: The User can browse through the list and see details of each destination.

1. User with Share Favorite Destinations

Relationship: The User interacts with this use case to share their list of favorite destinations with a Friend.

Description: The User selects the destinations to share and provides the Friend's contact information.

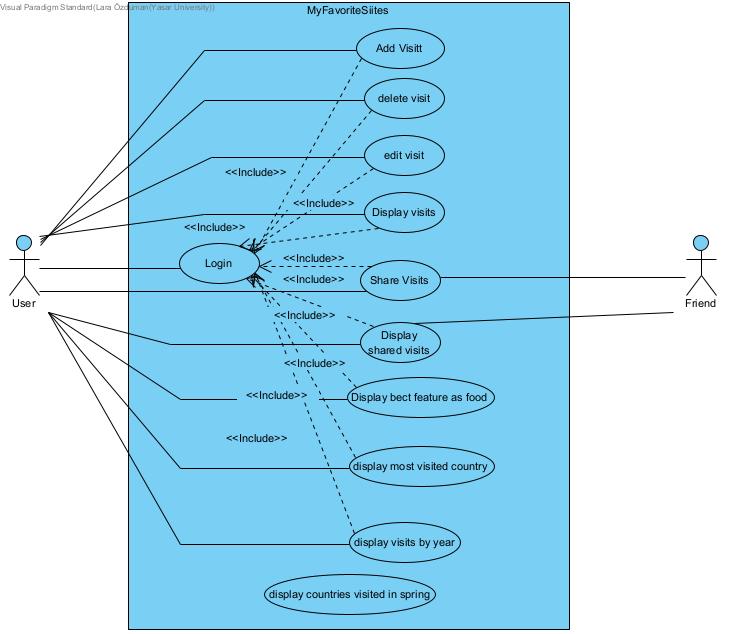
1. Friend with Share Favorite Destinations

Relationship: The Friend is a recipient in the Share Favorite Destinations use case.

Description: The Friend receives the shared list of favorite destinations from the User

### **Use Case Diagram**

This use case diagram shows how the application use cases work with and how they depend on each other.



# **Behavioral Models**

## **Sequence Diagram**

This Sequence diagram shows how adding a new visit works in the system

A diagram of a project

Description automatically generated

This Sequence Diagram show how sharing a visit with a friend works in the system

A diagram of a diagram

Description automatically generated

This sequence diagram shows how display the shred visit of your friend works in the system

A diagram of a diagram

Description automatically generated

# **Structural Models**

## **Class Diagram**

This class diagram shows all the classes in the system with their attributes and operands, their connection and how they interact to each other in the system

A diagram of a computer program

Description automatically generated with medium confidence

# **Process Modeling**

## **Data Flow Diagram (DFD)**

This context data flow diagram shows how the system interacts with its actors and their interactions in between.

A diagram of a system

Description automatically generated

This level 0 data flow diagram show the processes and databases that interact with the actors, what interactions they are having and how the data flows through the system.

A diagram of a computer

Description automatically generated

# **Graphical User Interface(s) (GUIs)**

This is the login frame which is the first frame in the system, it has username and password fields to login.

A screenshot of a login screen

Description automatically generated

This is the login frame with information entered, when login button is clicked this frame closes.

A screenshot of a login screen

Description automatically generated

This is the main frame which includes add,delete,display,edit,display most visited country, display best feature as food, display visit by year, display spring visits, share, display shared visits and display image functions.

A screenshot of a computer

Description automatically generated

This is the add function , this frame up when the add button in the main frame is selected

A screenshot of a computer

Description automatically generated

This is the add frame with information entered

A screenshot of a computer

Description automatically generated

When the add button is selected after entering information, this success message comes up

A screenshot of a computer

Description automatically generated

This is the delete function and this frame opens up after selecting the delete button on the main frame. With a given visit ID it deletes an added visit.

A screenshot of a computer

Description automatically generated

This is the delete with visit ID entered

A screenshot of a computer

Description automatically generated

When selecting the delete button on the delete frame after entering visit ID, this message pops up

A screenshot of a computer

Description automatically generated

This is the display function, when the display button is selected in the main frame ,it displays all the visited countries in a message.

A screenshot of a computer screen

Description automatically generated

This is the edit frame, it has text fields to enter information

A screenshot of a computer

Description automatically generated

To edit it is necessary to choose which previous visit that is to be edited, This part displays the previous visits to choose from

A screenshot of a computer

Description automatically generated

After selecting a visit, this question comes up to make sure that user is sure that they want to edit this visit.

A screenshot of a computer

Description automatically generated

After selecting yes and editing , when the update button on the edit frame is pressed, this success message comes up.

A screenshot of a computer

Description automatically generated

When the display button in the main frame is selected, it displays the country that the user hab been the most to in a message.

A screenshot of a computer

Description automatically generated

When display best feature as food button is selected in the main frame, it displays the visit in which the best feature is food in a text field with their name and ratings.

A screenshot of a computer

Description automatically generated

With the display visit by year, there needs to be a year input in the enter year field.

A screenshot of a computer

Description automatically generated

After entering the input and selecting the display visits by year, a new frame opens up with the information of visits which were done in the selected year

A screenshot of a computer

Description automatically generated

When the display spring visits button is selected, a new text field open up which contains the information of visits that were done in spring.

A screenshot of a computer

Description automatically generated

When share button is selected to share a visit with a friend, the user enters friends username and the visit ID of the visit they want to share.

A screenshot of a computer

Description automatically generated

After entering the information in the share frame and selecting the share visit button, this success message comes up.

A screenshot of a computer screen

Description automatically generated

When the display shared visits button is selected, a new frame displays the shared information of visits which are from the users friend.

A screenshot of a computer

Description automatically generated

To display the image of the country visited, firstly it is necessary to obligatory to enter a visit ID in the enter Visit ID area to find the location to be displayed.

A screenshot of a computer

Description automatically generated

After entering the visit ID and selecting the display image button, the image is displayed in another frame with the country name written on it.

A screenshot of a computer

Description automatically generated

# **Conclusion and Future Work**

In this project, I developed an travel information storage application that allows users to store the information of and manage their travelling experiences with the function of the application like adding a new visit, deleting visits, editing and updating a visit, displaying all visits with some of them including criteria ,sharing visits with friends and seeing the shared visit of their friend.

The system is built with Java and MySQL database to store and manage data and the tools used to build the system and diagrams are IntelliJ IDEA and Visual Paradigm which helped the process and ensured a maintainable codebase.

The application successfully meets the specified requirements by providing the following key features:

Adding Locations: Users can add new travel locations with details such as country, city, year, season, best feature, comments, and rating.

Deleting Visits: Users can delete a visit from the database using the visit ID.

Updating Visits: Users can display, edit, and update visit information.

Viewing Information: Users can view country names based on best feature, display location images, find visits by year, and identify countries visited the most or only in spring.

Sharing Visits: Users can share their favorite visits with friends, and view visits shared with them.

Future Work:   
While the current program is functional and working there is still room for improvement.  
There are few enhancements that can be added like:

* Delete function displaying all the options and letting the user select instead of takin in visit ID input
* Adding in a notification system to let the user know about a new shared visit
* Providing better search and filter options to cover all the criteria the user needs

By implementing these enhancements , this application would become a better tool to keep track of the users traveling history and would provide a better experience for the user.