**Yaşar University**

**Spring, 2023 - 2024**

**SE 2224 - Software System Analysis**

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

|  |  |
| --- | --- |
| **Student Name:** | **Ahmet Kemal Biliciler** |
| **Student No:** | **21070006001** |
| **Department Name:** | **Software Engineering** |
| **Course Section No:** | **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**

This section defines the reason for creating the Software Requirements Specification (SRS) document. Its purpose is to clearly outline the goals and functionalities of the "FavoriteSites" software application. It serves as a guide for developers, stakeholders, and users, ensuring a shared understanding of what the system aims to achieve and how it should operate.

## **Scope**

a) The software product to be produced is named "FavoriteSites."

b) The software product will function as an application allowing users to maintain a record of visited locations and share their favorite sites with others.

c) The goal of the application is to provide users with a user-friendly platform for managing their travel memories, fostering social interaction through sharing experiences, and enhancing overall user satisfaction in accessing and organizing visited locations.

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

• SRS: Software Requirements Specifications

• GUI: Graphical User Interface

• IDE: Integrated Development Environment

• JDBC: Java Database Connectivity

• UML: Unified Modeling Language

## **References**

1. Lecture Notes and Labs.
2. ChatGPT
3. <https://www.geeksforgeeks.org>

## **Overview**

1. Introduction: Introduces the FavoriteSites Application, providing an overview of its purpose and scope.
2. Design and Implementation Constraints: Describes the factors and limitations affecting the development process, including hardware and software requirements, interfaces, technologies, and tools.
3. Specific Requirements: Details the functional and non-functional requirements, performance expectations, software system attributes, and includes use case analysis along with behavioral and structural models.
4. Behavioral Models: Demonstrates the application's behavior using Sequence diagrams.
5. Structural Models: Illustrates the foundational structure of the application through Class diagrams.
6. Process Modeling: Explains how the system processes information and data, utilizing Data Flow diagrams.
7. Conclusion and Future Work: Summarizes the document and suggests potential enhancements for the application's future development.

# **Design and Implementation Constraints**

Hardware limitations: Standard desktop requirements.

Interfaces: Must integrate with MySQL database.

Technologies: Java for programming, Apache NetBeans IDE 12.4, MySQL 8.0 for database management, Visual Paradigm 17.1 for UML modeling.

Language: The application must be developed using the Java programming language.

Tools: Visual Paradigm for modeling, Java Swing for GUI development, and JDBC for database connectivity.

# **Specific Requirements**

## **Functional Requirements**

1. The system shall provide a login interface for users to enter their username and password.
2. The system shall authenticate users based on the provided username and password.
3. The system shall allow authenticated users to add a new location to their list of visited sites.
4. The system shall require users to input the Country name, City name, Year visited, Season visited, Best feature, Comments, Rating when adding a new visit.
5. The system shall automatically assign a unique visit ID to each newly added visit.
6. The system shall allow users to delete a visited location by providing its visit ID.
7. The system shall allow users to view and edit information about a visited location.
8. The system shall display a list of visited locations sorted by country name.
9. The system shall allow users to search for visited locations based on the year of visit.
10. The system shall provide a feature to share a favorite visit ID with a friend.
11. The system shall display shared visit information from friends.
12. The system shall display images of visited locations when requested by the user.
13. The system shall display list of visited locations based on user preferences, such as by best country for food or most visited country etc.
14. The system shall handle errors gracefully and provide informative error messages to users when necessary.

## **Performance Requirements**

a) The system should support a minimum of 100 terminals concurrently accessing the application.

b) The system should be capable of supporting up to 500 simultaneous users accessing the application at the same time.

c) The system should be able to handle and process up to 10,000 records of visited locations without any significant impact on performance.

## **Software System Attributes**

1. The system should operate reliably with an average of fewer than two failures per week.
2. The system should be accessible to users at least 99% of the time.
3. The system should encrypt all user data during transmission and storage.
4. The system should be designed with modular components to facilitate easy updates and modifications.
5. The system should ensure user-friendly interactions, allowing users to perform common tasks with minimal effort and providing clear instructions when needed.

## **Use Case Analysis**

### **Actors**

There are two primary actors that will interact with the system:

User: Represents any individual who uses the FavoriteSites application to manage their visited locations, add visit, delete visit, share visit, and perform other related actions.

Friend: Represents a user's friend who receives shared visit IDs from the user and can view the shared visit information within the application.

### **Scenarios**

Add Visit:

1. User inputs the details of a new visited location, including country name, city name, year visited, season visited, best feature, comments, and rating.
2. User submits the information, triggering the system to add the new visit to the database.

Preconditions: User must be logged into the system.

Post conditions: The new visit is successfully added to the visits table.

Share Visit :

1. User selects a visited location and chooses to share its visit ID with a friend.
2. User specifies the friend's username and confirms the sharing action.

Preconditions: User must be logged into the system and have a visited location selected.

Post conditions: The visit ID is successfully shared with the specified friend.

.

### **Use Case Forms**

Use Case Name: Add Visit

Participating Actors: User

Description: This use case involves the user adding a new visited location to the system providing details such as country name, city name, year visited, season visited, best feature, comments, and rating.

Trigger: User initiates the action to add a new visit.

Preconditions: User must be logged into the system.

Normal Course (Flow of Events):

1. User selects the option to add a new visit.
2. User inputs the details of the new visit.
3. User submits the information.

Post Conditions: The new location is successfully added to visits table.

Exceptions: If required fields are not filled, the system prompts the user to complete all necessary information.

Use Case Name: Share Visit

Participating Actors: User, Friend

Description: This use case involves the user sharing the visit ID of a visited location with a friend, allowing the friend to view the shared visit information.

Trigger: User selects a visited location and chooses to share its visit ID.

Preconditions: User must be logged into the system and have a visited location selected.

Normal Course (Flow of Events):

1. User selects a visited location .
2. User selects share visit feature.
3. User enters the id of the visited location.
4. User enters the friend's username.
5. User confirms the sharing action.

Post Conditions: The visit ID is successfully shared with the specified friend. Friend receives the shared ID.

Exceptions: If the friend's username is invalid or the visit ID cannot be shared for any reason, the system notifies the user accordingly.

Use Case Name: Update Visit

Participating Actors: User

Description: This use case involves the user modifying the details of a visited location in visits table.

Trigger: User selects the option to update a visit.

Preconditions: User must be logged into the system and have at least one visited location saved.

Normal Course (Flow of Events):

1. User selects the visited location they want to update
2. User enters the id of the visit they want to update then selects update visit feature.
3. User modifies the desired fields (e.g., country name, city name, rating, comments).
4. User submits the updated information.

Post Conditions: The visited location details are successfully updated in the visits table.

Exceptions: If user encounters an error while submitting the changes, the system maintains the previous state of the visited location.

Use Case Name: Delete Visit

Participating Actors: User

Description: This use case involves the user deleting a visited location from the visits table.

Trigger: User selects the option to delete a visited location.

Preconditions: User must be logged into the system and have at least one visit in the table.

Use Case Name: View Shared Visits

Participating Actors: User

Description: This use case involves viewing the visit information shared to the user.

Trigger: User selects the view shared visits.

Preconditions: User must be logged into the system and have a shared visits to them.

Use Case Name: Search Visits by Year

Participating Actors: User

Description: This use case involves the user searching for visited locations based on the year of visit.

Trigger: User initiates a search by specifying the year of visit.

Preconditions: User must be logged into the system.

Use Case Name: Search Best Countries for Food

Participating Actors: User

Description: This use case involves the user searching for countries known for their best food features.

Trigger: User initiates a search for countries with the best food.

Preconditions: User must be logged into the system.

Use Case Name: Visits Only for Spring

Participating Actors: User

Description: This use case involves the user searching for locations visited only during the spring season.

Trigger: User initiates a search for visits in the spring season.

Preconditions: User must be logged into the system.

Use Case Name: Most Visited Countries

Participating Actors: User

Description: This use case involves the user viewing the list of countries they have visited the most.

Trigger: User requests a list of their most visited countries.

Preconditions: User must be logged into the system.

Use Case Name: Display Visit

Participating Actors: User

Description: This use case involves the user viewing the details of a specific visited location.

Trigger: User selects a location to view its details.

Preconditions: User must be logged into the system.

Use Case Name: Display Image

Participating Actors: User

Description: This use case involves the user viewing the image of a specific visited location.

Trigger: User selects a location to view its image.

Preconditions: User must be logged into the system.

Use Case Name: Refresh

Participating Actors: User

Description: This use case involves the user refreshing the table to see the latest updates.

Trigger: User initiates a refresh action.

Preconditions: User must be logged into the system.

Use Case Name: Login

Participating Actors: User

Description: This use case involves the user logging into the application using their credentials.

Trigger: User enters their username and password to log in.

Preconditions: User must have valid login credentials.

Use Case Name: Open Menu

Participating Actors: User

Description: This use case involves the user opening the menu to access different features of the application.

Trigger: User selects the menu option.

Preconditions: User must be logged into the system.

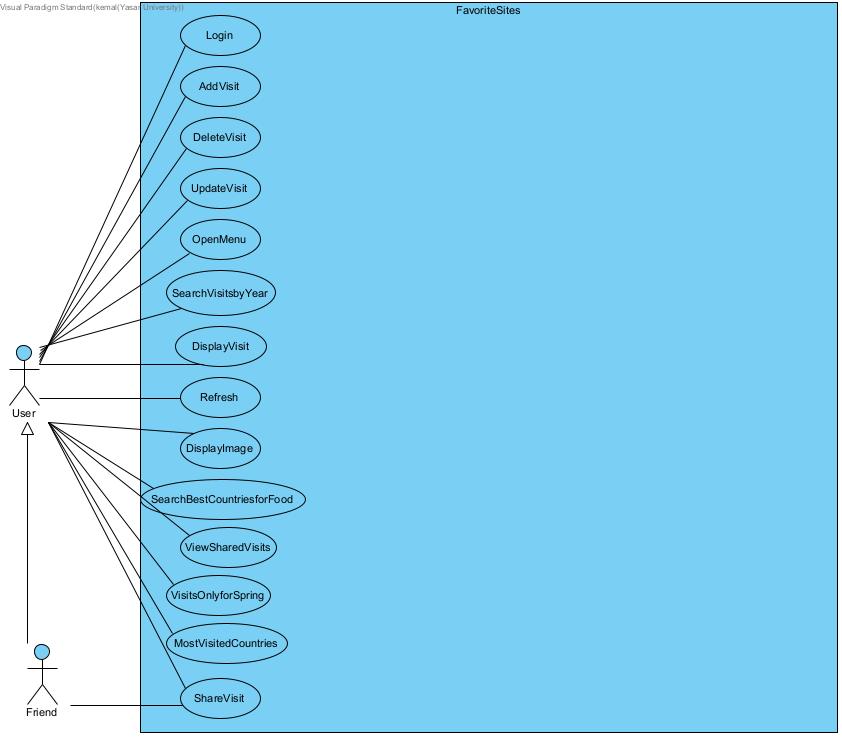
### **Relationships among Actors and Use Cases**

User: Interacts with all use cases including Add Visit, Update Visit, Delete Visit, Best Countries for Food, Visits Only for Spring, Most Visited Countries, Display Location, Display Image, Refresh, Login, Share Visit, Open Menu ,and View Shared Visits.

Friend: Interacts with the Share Visit use case.

This setup highlights that the User has broad access to all functionalities within the FavoriteSites application, while the Friend's interaction is limited to receiving information that has been shared with them.

### **Use Case Diagram**



The use case diagram illustrates the interactions between the actors and the use cases in the FavoriteSites application.

User: The primary actor who interacts with all the use cases.

Friend: This actor interacts specifically with the "Share Visit" use case.

The diagram demonstrates the comprehensive range of functionalities available to the User and the specific interaction point for the Friend within the application.

# **Behavioral Models**

## **Sequence Diagram**

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

This is a sequence diagram for Add Visit. It explains how the process of Adding a new visit to the visits table works. FavoriteSites takes details of the visit and sends it to the database for storing it.

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

This diagram outlines a sequence of steps for refreshing data in Mainframe table with database system in response to a user initiated action. Mainframe sends an query to database to see all the visit data in that moment and displays it.

metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

This diagram outlines a sequence of steps for login process with using inputted username in database to see the real password and compares it with password input.

# **Structural Models**

## **Class Diagram**

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturulduThis class diagram provides a clear overview of all the classes within the application, including their attributes and methods. The LoginForm class facilitates user login, and upon successful authentication, it initiates the MainFrame class. The MainFrame class grants access to all the application's functionalities.

# **Process Modeling**

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

Context Diagram:

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

This context diagram simplifies the data flow within the system. It transforms user queries into database queries, retrieves the results, and returns them to the user. Also using the visit ID that user entered, it retrieves the image of the location.

Level 0 DFD:

metin, diyagram, plan, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

This diagram is a more complicated version of the data flow illustrated in the above context diagram. It explains how system process with additional functions that were not shown previously .

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

**Forms:**

LoginForm ShareVisit

metin, ekran görüntüsü, ufuk çizgisi, bina içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Add Visit Update Visit

metin, ekran görüntüsü, sayı, numara, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

MainFrame MenuFrame

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Functions:**

Login

metin, ekran görüntüsü, ufuk çizgisi, gökdelen içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of the login GUI is to take user information and validate the information with the data comes from database if it matches then user directed to the Mainframe.

Shared Visits With Me

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI is to show visit information that shared with us by other users

Shared Visits with your friend

metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI is sharing visit with your friends by entering their name and the visit ID of the visit user wants to share.

Display Image

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI show image of the location that user desire.

Open Menu

metin, ekran görüntüsü, ekran, görüntüleme, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI to show functions that user can execute.

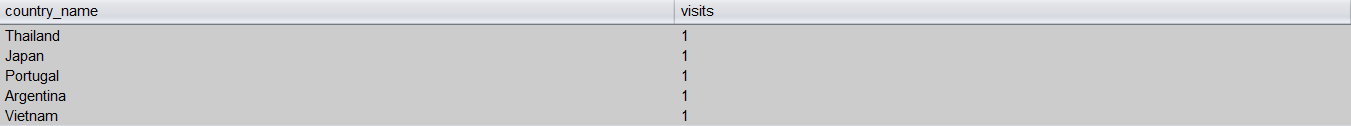
Search by year

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI Displaying visit information of the visits happened in year that user enters.

Best Countries for Food



Purpose of this GUI Displaying countries which best feature is food.

Visits Only for Spring

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu Purpose of this GUI Displaying countries which users visited in spring.

Most Visited Countries

çizgi, metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI Displaying countries which user visited the most.

Add Visit

metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI is adding new visits into the database.

Update Visit

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Purpose of this GUI is to update a visit that already exists.

Delete Visit and Refresh

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu



metin, sayı, numara, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

The purpose of the Delete Visit is deleting information of the visit user wants to delete with entering visit ID.

The purpose of the Refresh is to refresh the table with the current visits data in database.

User need to refresh the table when they deleted a visit if they want to see the update on the table. That is why these two GUI explained under one title.

Display Visit

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

The purpose of this GUI is displaying the visit that user entered.

# **Conclusion and Future Work**

The "FavoriteSites" project successfully delivers a user-friendly application for managing and sharing travel locations. Key features include login, detailed visit records, easy data manipulation, and social sharing. Also the project effectively uses a structured database for displaying tables. Overall, "FavoriteSites" offers a robust platform for users to document and share their travel experiences efficiently.

Following features can be added to the project as a future work:

Adding a map feature for users to visually see the visited locations on the map.

Offer customizable themes and layouts to enhance user experience and personalization.

By adding multi-language support in the future, we could enable users to easily interact in different languages.