Software Requirements Specification Template

Indian Student Association - NMWSU

Software Requirements Specification

Version – 1.2

Wednesday, June 10, 2020

Team Members:

Bharat Reddy Male

Bhavya Deepthi Gorrepati

Mahalakshmi Kongari

Sai Jyothsna Mathi

Jeevan Reddy Mure

Dheeraj Edupuganti

Submitted in partial fulfilment

Of the requirements of

CSIS 44-691 Graduate Directed Project 1

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| June’09,2020 | Version – 1.0 | Team | Initial Document |
| June’10,2020 | Version – 1.1 | Team | Revised Requirements |
| June’10,2020 | Version – 1.2 | Team | Updated document with ER-Diagram |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**Table of Contents Page Number**

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. References
   5. Overview
2. General Description
   1. Product Perspective
   2. Product Functions
   3. User Characteristics
   4. General Constraints
   5. Assumptions and Dependencies
3. Specific Requirements
   1. External Interface Requirements
      1. User Interfaces
      2. Hardware Interfaces
      3. Software Interfaces
      4. Communications Interface
   2. Functional Requirements
   3. Use Cases
   4. Class/Objects
   5. Non-Functional Requirements

3.5.1. Performance

3.5.2. Reliability

* + 1. Availability
    2. Security
    3. Portability
  1. Inverse Requirements
  2. Design Constraints
  3. Logical Database Requirements
  4. Other Requirements
  5. Prototypes (for complete project)
  6. Use Case Diagrams

1. Design

4.1. ER diagram

4.2. GUI

1. Analysis Models

5.1. Data Flow Diagram

5.2. Sequence Diagram

1. **Introduction**
   1. **Purpose:** The purpose of this document is to present a detailed description oftheIndian Student Association – NWMSU, a website that is intended to help the Indian community studying at Northwest Missouri State University. This website will be used to inform the community about the events being held by ISA, the team will be able to better manage the requests of the students and have a better interaction with them during a pandemic situation.
   2. **Scope:** This application will be a web-based application for the Indian community studying at Northwest Missouri State University. The system will be designed to help the user to:
      1. Register with ISA
      2. Request Airport rides
      3. View upcoming events hosted by ISA
      4. Contact ISA for any issues
      5. Give feedback to ISA, etc.
   3. **Definitions, Acronyms, and Abbreviations:**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| ISA | Indian Students Association (ISA) is a recognized organization in the university. It is headed by Sai Narneni |
| IDE | An integrated development environment (IDE) is a [software application](https://en.wikipedia.org/wiki/Application_software) that provides comprehensive facilities to [computer programmers](https://en.wikipedia.org/wiki/Computer_programmer) for [software development](https://en.wikipedia.org/wiki/Software_development). |
| Database | Collection of all the information monitored by this system. |
| GUI | A GUI (graphical user interface) is a system of interactive visual components for computer software |
| User | A person who accesses the application |
| Data breach | A data breach is an incident that exposes confidential or protected information |
| Malware | Malware is the collective name for several malicious software variants, including viruses, ransomware, and spyware |
| CRUD | In computer programming, create, read, update, and delete are the four basic functions of persistent storage. |
| HTML | Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser |
| Captcha | A CAPTCHA is a type of challenge-response test used in computing to determine whether or not the user is human. |

* 1. **References:**
* [ISA - Northwest Missouri State University](https://nwmissouri.campuslabs.com/engage/organization/nwmissouri_isa)
  1. **Overview:** In the next section, the general description you will find an overview of the functionality of the product. It describes the general factors that affect the product and its requirements and is used to establish a context for the technical requirements specification.

The third section, specific requirements, of this document, is written primarily for the developers and describes technical terms, the details of the functionality of the product. This subsection of the SRS will list each of the factors that affect the requirements stated in the SRS.

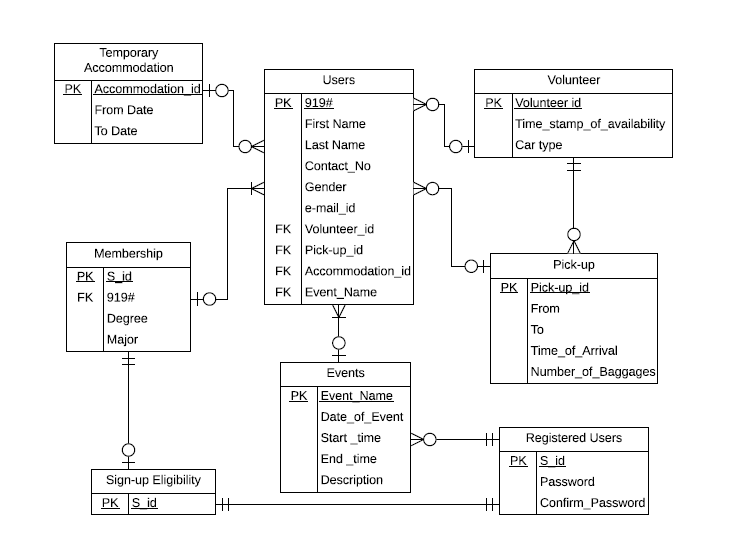
The fourth and fifth section of this document provides an overview of the application’s design and analysis model.

1. **General Description**
   1. **Product Perspective:** This website is mainly targeted on the Indian community that is willing to join Northwest Missouri State University. The ISA team will use this website as a commonplace for responding to queries and requests from students. It would also be used to manage the events that the team hosts.
   2. **Product Functions:** The website will have the following functionalities:
      * Login
      * Registration
      * Event creation
      * Pickup monitoring
      * Chatbox
      * Help(Contact Us, Alumni, FAQ)
      * Temporary accommodation
   3. **User Characteristics:** There would be two types of users that would be using this application. The ISA team and normal users. The ISA team would be able to respond to user requests like pick-up, temporary accommodation, and add upcoming events, etc. The normal users would be able to request for pickup, view event details, and request for temporary accommodation, etc. everyone would be able to access the chat feature and forum, however.
   4. **General Constraints:** Since this is a website and will mostly be accessed on a variety of devices with many kinds of browsers that are of user’s choice, the website’s performance might not be the same for everyone. Also, this being a website, users need an active internet connection for accessing it and a slower internet connection would result in performance reduction.
   5. **Assumptions and Dependencies:** The application would not perform the same for everyone since it depends on the device capable of handling the website. The application has a dependency on an active internet connection for using it.
2. **Specific Requirements**
   1. **External Interface Requirements:**
      * User Interfaces: A website will be required for providing the user with an interface to perform actions on the applications.
      * Hardware Interfaces: The user can access the application on a browser either on a mobile or a computer. The data will be handled by a server hosted physically or virtually over a database.
      * Software Interfaces: An interface for the user and the system is required, the database handling software is required to manage the data of the application. An IDE is required for the development and management of the application.
      * Communications Interface: The project will be using an email-based communication system for managing user’s login, registration, and subscribed notifications.
   2. **Functional Requirements:**

The application when accessed should return basic information about the University and the Organization as its home page. The page should contain a navigation bar that has an ISA logo and Northwest Missouri State University logo to the left and links to:

* Home
* Events
  + Upcoming events
  + Past events
* Services
  + Membership
  + Pickup
  + Temporary Accommodation
  + Post Arrival checklist
  + Pre Arrival checklist
  + Volunteer for pickup
* Help
  + Contact us
  + Alumni
  + FAQ
* Admin Login
* 24/7 Online Chat
* **Home:** This should be the main page that should be returned when the application is accessed. This page should contain a carousel of Indian culture as its background. The carousel should be followed by information about the college and the organization. The footer should contain links to social media profiles of the organization and the university, etc.
* **Events:** This page displays past and upcoming events held by team ISA. The navigation bar for “events” contains a sub-menu of links to past and upcoming events.
  + The past events page returns the details, sponsors, and images captured during a past event.
  + The upcoming events page has details about the upcoming events that are being hosted by ISA.
* **Services:** The user will be provided forms to fill in their details for pickup, membership, temporary accommodation, and volunteer pickup. The data inputted in the forms will be stored at a central database for CRUD operations.
  + Pickup form: Users need to fill in their details like the first name, last name, contact no, from, to, time of arrival, number of baggage.
  + Membership form: Users need to fill in their details like the first name, last name, contact no, address, email id, degree, gender.
  + Temporary Accommodation: Users need to fill in their details like first name, last name, contact no, number of days required, gender.
  + Volunteer pickup: Users need to fill in their details like the first name, last name, contact no, number of baggage allowed.
  + Pre Arrival checklist provides necessary information and tasks to be completed by a student before they arrive at the university.
  + Post Arrival checklist provides necessary information and tasks to be completed by a student after they arrive at the university.
* **Help:** The user will be provided with links to pages for contact us, alumni and FAQ’s
  + Contact us: This page provides contact information about the current ISA team and a short description of them.
  + Alumni: This page provides details about the ISA alumni. This would help students get to know about their seniors and their experience at the university.
  + FAQ: This page contains frequently asked questions and answers to them.
* **24/7 Online Chat Box**: This feature is an online chat 24/7 which helps users to raise their queries and get them clarified instantly.
* **Admin Login:** This functionality is only intended for the ISA team for managing the application. Post login, the team will be provided with functionalities like:
  + View pending pickup requests.
  + Add a new upcoming event.
  + Update the past events.
  + View pending temporary accommodation requests.
  + Display a list of volunteers available for pickup.
  + Update the help menu.
  1. **Use cases:**
  2. **Class/Objects:**
  3. **Non-Functional Requirements:** 
     + Performance: The application should be able to respond to multiple users without having a delay and the UI should be rendered without any delay in displaying content.
     + Reliability:The application should always be available without any failures to the users for a given period.
     + Availability:The application should be available to a user at any given point i.e., even if there is a lot of load on the application due to multiple requests being handled.
     + Security:Data of the users is crucial and needs to be kept safe, the application should be safe from data breach and malware attacks and the application needs to have a way for the user to authenticate before they access the content.
     + Portability: The application should be able to adhere to different environments like mobile, desktop, tablet, etc. and should function in the same way in any browser.
  4. **Inverse Requirements:** The application has the following inverse requirements
     + - The application should not allow a user to login to it when the user-provided blank, incorrect credentials.
       - The user should not be able to access the profile of another user when he logs in using their credentials.
       - The registration functionality should not allow a user to register into the application when provided improper data.
       - The user should not complete any forms with improper/missing data.
  5. **Design Constraints:** The application should allow non-humans to operate. A captcha mechanism should be included to avoid improper access. The GUI should be implemented in HTML.
  6. **Logical Database Requirements:** The application should be able to perform CRUD operation across multiple tables without any data corruption.
  7. **Other Requirements:** Computers for the team, required software licenses to be acquired, the workplace needs to be setup.
  8. **Prototypes (for the complete project):**
  9. **Use Case Diagrams:**

1. **Design:**
   1. **ER diagram:**

****

In our project, we have entities such as Volunteer, Pick-up, Temporary Accommodation, Membership, Sign-up Eligibility, Registered-users, and Events. There is a central table called Users.

**Attributes of the Entities:**

In the **Pick-up** entity, we have attributes such as Pick-up id, From, To, Time of arrival, Number of Baggage’s where the information related to the pick-up request is stored such as the user's Pick-up location(From), Destination(To), their time of arrival and number of pieces of baggage they have.

In the **Temporary Accommodation** entity, we have attributes such as Accommodation id, From-date, and To-date. From-date and To-date stores the information about the number of days a user requires temporary accommodation.

In the **Volunteer** entity, we have attributes such as, Volunteer id, Time stamp of availability which stores the information about the available time of the volunteer for pickup and car type which stores the information about the type of the car the volunteer has/uses for the pick-up.

In the **Membership** entity, we have attributes such as, S-id, 919#, Degree which stores the type of degree the user is pursuing and major which stores the data about the major the user is pursuing.

In the **Events** entity, we have attributes such as, Event Name which stores the name of the event, Date of Event, Start time, End time which stores the date, start and end time of that particular event and Description stores the data about the brief introduction on that event.

In the **Sign-up Eligibility** entity, we have an attribute S-id that stores the S-id’s of the current ISA team, so that it can restrict the sign-up/registration process to that particular team whose S-id’s are stored in prior on this table based on verification/validation.

In the **Registered Users** entity, we have attributes such as S-id, Password, and Confirm Password which stores the information to get the users to log into the portal to make any additions or modifications to the portal such as adding/modifying events.

In **Users** entity, we have attributes such as 919# to store the 919# of the user, First name, Last name to store the first and last names of the user, contact no, email id to store the contact number and email id of the user respectively, Gender to store the user’s gender, while the volunteer id, accommodation id, pick-up id, event name are the Primary keys of the other entities acting as foreign keys in this entity.

**Relationship between Entities**

In the ER diagram, considering the pickup request first, the relation from users and pick-up is optional-one and the relation from pick-up to users is optional-many. This is because the pick-up request page must be accessed by many users while each user will be allotted with the single pick-up request if needed. Same case with Temporary Accommodation and Volunteer pages. The relation between users to temporary accommodation is optional-one and the relation from temporary accommodation to users is optional many. This is because the temporary accommodation page must be accessible to all the users while a user is allotted to single accommodation if registered for.

The relation between users to volunteer is optional-one and the relation from volunteer to users is optional-many. This is because the volunteer page must be accessed by all the users while each user can sign up for single volunteer service at a time.

The relation between pick-up to volunteer is mandatory-one and the relation from volunteer to pick-up is optional-many. This is because each pick-up is assigned with a single volunteer while each volunteer can have multiple pickups.

The relation between users to membership is optional-one and the relation from membership to users is mandatory-many. This is because all the data that the user table has need not mandatorily have the membership but the membership is open to multiple users.

The relation between Sign-up eligibility to membership is mandatory-one and the relation from membership to sign-up eligibility is optional-one. This is because all the users who have membership are not eligible for sign-up however, the people who have sign-up eligibility must have the membership.

The relation between Sign-up eligibility to registered users is mandatory-one and the relation from registered users to sign-up eligibility is mandatory-one. This is because all the people who have sign-up eligibility can be registered users and all the registered users have sign-up eligibility.

The relation between Events to registered users is mandatory-one and the relation from registered users to Events is optional-many. This is because a registered user can add/modify any number of events while each event can be added/modified by a single registered user.

The relation between Events to Users is mandatory-many and the relation from Users to Events is optional-one. This is because an event is open to any number of users at a time but a user may attend a single event at a time.

* 1. **GUI:**

1. **Analysis Models:** 
   1. **Data Flow Diagram:**
   2. **Sequence Diagram:**

## 