Software Requirements Specification Template

Students Information Exchange Web

Software Requirements Specification

<Version 5.0>

<10/15/2023>

Spurthi Ravula

Vinay Meenkeri

Supriya Kutikanti

Vinay Chary Nangunoori

Satya Deepu Mandapati

Surya Mohan Jagani

Submitted in partial fulfilment

Of the requirements of

CSIS 44-691 Graduate Directed Project 1

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| <date> | <Version 3> | <Your Name> | <First Revision> |
| 10/15/2023 | 5.0 | Vinay Meenkeri | GUI |
| 10/15/2023 | 5.0 | Spurthi Ravula | GUI |
| 10/15/2023 | 5.0 | Surya Mohan Jagani | GUI |
| 10/15/2023 | 5.0 | Vinaychary Nangunoori | GUI |
| 10/15/2023 | 5.0 | Supriya Kutikanti | GUI |
| 10/15/2023 | 5.0 | SatyaDeepu Mandapati | GUI |

# Document Approval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
| Spurthi Ravula | Spurthi Ravula | Organizer module | 10/15/2023 |
| Vinay Meenkeri | Vinay Meenkeri | User Module | 10/15/2023 |
| Vinaychary Nangunoori | Vinaychary Nangunoori | User Module | 10/15/2023 |
| Supriya Kutikanti | Supriya Kutikanti | Organizer module | 10/15/2023 |
| SatyaDeepu Mandapati | SatyaDeepu Mandapati | Organizer module | 10/15/2023 |
| Surya Mohan Jagani | Surya Mohan Jagani | User Module | 10/15/2023 |

**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

4.1. Data Flow Diagram

4.2. Sequence Diagram

**1.Introduction**

The Northwest Missouri State University Information Exchange Website (NWMSU-IEW) is a vital platform designed to facilitate seamless information sharing and collaboration among the students of Northwest Missouri State University (NWMSU). This research paper aims to provide a comprehensive understanding of the NWMSU-IEW, including its purpose, scope, key definitions, acronyms, and abbreviations, as well as references and an overview of its significance

**1.1.Purpose**

The primary purpose of the NWMSU-IEW is to serve as an online hub for students to exchange information, resources, and engage in meaningful interactions. It is a central platform designed to enhance communication and collaboration among NWMSU students. By fostering a digital environment conducive to information sharing, the website aims to enrich the academic, social, and extracurricular experiences of NWMSU students.

**1.2.Scope**

The NWMSU-IEW encompasses a wide range of features and functionalities, including but not limited to:

Academic resources: Providing access to course materials, lecture notes, and study resources shared by students.

Event and club promotion: Facilitating the promotion of student clubs, events, and extracurricular activities.

Discussion forums: Creating spaces for students to engage in academic and non-academic discussions.

Classifieds: Allowing students to buy, sell, or trade items within the university community.

Information dissemination: Serving as a reliable source of communication for rides to airport, Walmart and other places of needs as per requirements.

**1.3.Definitions, Acronyms, and Abbreviations**

**NWMSU:** Northwest Missouri State University

**IEW:** Information Exchange Website

**Academic Resources:** Materials and information related to coursework and academic pursuits.

**Discussion Forums:** Online platforms for open dialogue and conversation.

**Classifieds:** Sections of the website for buying, selling, and trading items.

**Collaboration Tools:** Features and utilities that support group work and collaboration.

**University Community:** All students, faculty, and staff affiliated with NWMSU.

**1.4.References**

This research paper draws upon a variety of sources, including academic literature, official NWMSU documents, and online resources related to information exchange websites and their role in higher education institutions.

**1.5.Overview**

In summary, the NWMSU-IEW serves as a valuable digital ecosystem for NWMSU students, fostering communication, collaboration, and resource sharing. This paper will delve deeper into the website's functionalities, user experiences, benefits, and potential challenges in the subsequent sections, shedding light on its role in enhancing the educational journey and campus life of NWMSU students.

**2.General Description**

**2.1.Product Perspective**   
  
The "Student Info Exchange" is a web-based platform designed to facilitate communication and resource sharing among university students. It aims to provide a central hub for students to connect, exchange information, and access various services. Here's a breakdown of the product perspective :  
**System Interfaces:** The system will interface with a database to store user information, accommodation listings, textbook/material/furniture listings, event information, and other relevant data. Additionally, it may utilize external APIs for mapping and live chat functionality.   
**User Interfaces:** The system will have user-friendly web interfaces for students to register, log in, search for accommodations, view events, chat with other users, and utilize maps for location-based services.   
**Hardware Interfaces:** The system will be accessible via standard web browsers on various devices such as desktops, laptops, tablets, and smartphones.  
**Software Interfaces:** The application may integrate with authentication providers for user registration and login. It may also utilize mapping APIs for location-based services and incorporate a chat application for live chat functionality.

**2.2.Product Functions**   
  
The primary functions of the "Student Info Exchange" project include:  
**Student Registration and Login:** Students can register for an account by providing necessary personal details. Registered users can log in securely.   
**Accommodation Search:** Users can search for available accommodations, filtering by location and price range.  
**Resource Sharing:** Students can post listings to offer or request textbooks, study materials, and furniture from other students.  
**Ride Sharing:** Students can offer or request rides to and from the airport, as well as to shopping destinations like Walmart.  
**Event Information:** Users can access information about upcoming events, parties, and gatherings within the university community.  
**Certification Assistance:** The platform provides information and assistance related to university processes, certifications, transcripts, and other administrative tasks.  
**Live Chat:** Users can engage in real-time chat with other students for instant communication and assistance.  
**Search Functionality:** Comprehensive search capabilities are provided to help users find specific content within the platform.  
**Map Integration:** The system integrates mapping services to assist users in locating accommodations, events, and other services on a map.

**2.3.User Characteristics**   
  
The system is designed to serve the following user categories:  
**Students:** The primary users who register, log in, and utilize the platform for various purposes including resource sharing, accommodation search, event information, certification assistance, and chat.   
**Admins:** Administrators who manage and maintain the platform, oversee user accounts, monitoring content, and addressing any issues.

**2.4.General Constraints**   
  
**Data Privacy and Security:** The system will adhere to data privacy regulations, ensuring the confidentiality and security of user data.  
**Scalability:** The platform will be designed to handle a potentially large number of users.   
**Compatibility:** It will be compatible with popular web browsers and accessible on various devices.  
**Reliability:** The system will be reliable, with minimal downtime and robust error handling.   
**Resource Availability:** Availability of resources such as map data and external APIs may impact system functionality.

**2.5.Assumptions and Dependencies**  
  
**Assumptions:**  
Users have internet access and web browsing capabilities.   
Users will provide accurate and honest information during registration.  
The platform complies with relevant legal and ethical standards.  
  
**Dependencies:**   
The system will depend on third-party services for features like mapping and live chat.  
The availability of these external services may affect the overall functionality of the system.

**3.Specific Requirements**

**3.1.External Interface Requirements**

**3.1.1.User Interfaces**

The user interfaces should be intuitive, responsive, and user-friendly.

Supported web browsers include Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

The system should have a responsive design to adapt to different screen sizes and devices.

Users shall be able to register, log in, post listings, search for accommodations, view events, chat, and use mapping services via the user interface

**3.1.2Hardware Interfaces**

The system shall be accessible on various devices, including desktop computers, laptops, tablets, and smartphones.

Minimum hardware requirements for clients:

Processor: Dual-core 1.8 GHz or higher

RAM: 4 GB

Display: 1024x768 resolution or higher

**3.1.3.Software Interfaces**

The system shall integrate with an authentication provider for user registration and login.

Mapping services shall be integrated using mapping APIs for location-based services.

Live chat functionality may be incorporated using a third-party chat application

**3.1.4.Communications Interface**

The system shall utilize HTTPS (SSL/TLS) for secure data transmission over the internet.

Real-time chat communications shall be established using standard protocols (e.g., WebSocket) for instant messaging.

**3.2. Functional Requirements**

Functional requirements define the specific functionalities of your system. Here are some functional requirements for the "Student Information Exchange Web" project:

**User Registration and Login**

Users shall be able to register for an account by providing personal information.

Registered users shall log in securely using their email and password.

**Accommodation Search**

Users shall search for available accommodation, filtering by location and price range.

**Resource Sharing**

Students shall post listings to offer or request textbooks, study materials, and furniture.

Users can browse and respond to listings.

**Ride Sharing**

Students shall offer or request rides to and from the airport and shopping destinations.

Users can browse and respond to ride-sharing listings.

**Event Information**

Users can access information about upcoming events, parties, and gatherings within the university community.

Event details shall include date, time, location, and description.

**Certification Assistance**

The platform shall provide information and assistance related to university processes, certifications, transcripts, and other administrative tasks.

**Live Chat**

Users shall engage in real-time chat with other students for instant communication and assistance.

Chat history may be stored for reference.

**Search Functionality**

Comprehensive search capabilities shall be provided to help users find specific content within the platform.

Users can search for accommodations, listings, events, and chat conversations.

**Map Integration**

The system shall integrate mapping services to assist users in locating accommodations, events, and other services on a map.

Users can view maps with location markers.

**3.3. Use Cases**

1. User Registration and Authentication:

Actors: User, System

The user wants to sign up for an account and log in securely.

The system verifies the user's credentials and provides access.

2.Create and Share Information:

Actors: User

User can create and share various types of information (text, files, links) with other users or groups.

The user may customize the privacy settings for shared information.

3.Search and Discover Information:

Actors: User

The users can use usernames, keywords, tags, or categories to search for and find information provided by other users.

User can view and access shared information.

1. Messaging and Notifications: -

Actors: User

The user has a choice to send and receive messages from other users.

User receive notifications for new messages or updates to shared information.

1. Profile Management: -

Actors: User

User can edit their profile information, including a profile picture and bio.

Users can customize their privacy settings.

1. Reporting and Moderation: -

Actors: User, Moderator

The user has the option to report false or misleading information.

Moderator can review reports and take appropriate actions.

**3.4. Class/Objects**

**1. User:**

Attributes: username, email, password, profile picture, bio, privacy settings.

Methods: register, login, edit profile, manage privacy settings

**2. Information Item:**

Attributes: title, content, type (text, file, link), owner, timestamp, privacy settings.

Methods: create, edit, delete, share, set privacy settings

**3. Search Engine: -**

Methods: searchByKeyword, searchByTag, searchByCategory.

**4. Message:**

Attributes: sender, receiver, content, timestamp.

Methods: send, receive

**5. Notification: -**

Attributes: user, content, timestamp

**6. Moderator: -**

Attributes: username, email, password.

Methods: reviewReports, takeAction

**3.5. Non-Functional Requirements**

3.5.1. Performance

* The servers / webpage must be capable of handling a large volume of student data and user requests efficiently.
* Response time for user queries should not exceed 2 seconds.
* The system should support concurrent access by at least 500 users.

3.5.2. Reliability

* The system should have a minimum uptime of 99.9%.
* Data backups should occur daily and be stored securely.
* Automatic error detection and recovery mechanisms should be in place.

3.5.3 Availability

* The platform should be accessible to authorized users at all times.
* Redundancy and failover mechanisms should be in place to minimize downtime.
  + 1. Security
* User authentication and authorization mechanisms should be robust.
* Data encryption should be implemented for data transmission and storage.

3.5.5Portability

* The system should be compatible with major web browsers (Chrome, Firefox, Safari, Edge).
* Mobile responsiveness is required for access from smartphones and tablets.

**3.6.Inverse Requirements**

* **No Third-Party Advertising:** The platform should not display third-party  
  advertisements to users, ensuring a distraction-free experience and preserving user privacy.
* **No Personal Information Sharing:** Users should not be allowed to share sensitive personal information, such as Social Security numbers or financial details, within the platform's chat or listings.
* **No Intrusive User Tracking**: The platform should not engage in intrusive user tracking or data collection practices without explicit user consent. It should adhere to privacy regulations.

**3.7.Design Constraints**

* Compliance with data protection regulations (e.g., GDPR, FERPA) is mandatory.
* Budget and resource constraints should be considered during development.

**3.8.Logical Database Requirements**

* The database must support efficient querying and indexing of student information.
* Data normalization principles should be followed to minimize redundancy.
* Data archiving and purging strategies should be implemented for historical data.

**3.9.Other Requirements**

* **User Reporting and Moderation System**

Implement a reporting system that allows users to flag inappropriate content or behavior.

**3.10 . Prototypes (for complete project)**

3.10.1 Login Screen and Registration Screen

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

3.10.2 Home Page

A screenshot of a website

Description automatically generated

A group of images of different colored objects

Description automatically generated

3.10.3 Housing Screen

A screenshot of a computer

Description automatically generated

3.10.4 Ride Booking Screen

A screenshot of a website

Description automatically generated

3.10.5 Events Screen

A screenshot of a website

Description automatically generated

3.10.6 Certification Assistance

A screenshot of a login form

Description automatically generated

3.10.7 Students Maps

A screenshot of a map

Description automatically generated

3.10.8 Student Marketplace

A screenshot of a computer

Description automatically generated

**4.Design**

**4.1. ER diagram**

A diagram of a student activity

Description automatically generated

* Users can have multiple roles (student, owner).
* Students can be both drivers and riders in rides.
* Users can be owners or renters of accommodations.
* Students can attend multiple events, and events can have multiple attendees.
* Users can list multiple products and buy multiple products.
* Students can submit multiple certification assistance requests.

1. GUI;

Home:

A screenshot of a computer

Description automatically generated

A group of images of different colors

Description automatically generated

Student Housing / Accommodation

A screenshot of a web page

Description automatically generated

Events:  
A screenshot of a website

Description automatically generated

## Ride

A screenshot of a website

Description automatically generated