Software Requirements Specifications

Student Activity Website
Version V2

Safa Zainab
Akshita Addagatla
Hesham Mahmoud
Wasay Mohammed Abdul
Irbaz Abdul Rauf Sayyed

March 23, 20

Table of Contents

1 Introduction

1.1 Purpose

This project seeks to enhance the student experience by developing a centralized online portal. This comprehensive website will empower students to manage their accounts, locate classmates and faculty, and search for essential resources such as textbooks and potential roommates. The platform will further promote convenience by facilitating online purchases of meal plans and bus tickets. Additionally, the website will serve as a central hub for campus activities and elections, keeping students informed and engaged. By consolidating these functionalities, this project aims to streamline everyday tasks for students and foster a more connected campus community.

1.2 Scope

This project is focused on the development of a dynamic online platform specifically designed to cater to the needs and activities of students. The website will offer a comprehensive suite of functionalities aimed at streamlining various aspects of student life.

Inclusions:

- Account Management: Students can establish accounts, log in, and update their personal information.
- People Search: The website facilitates searching for other students and faculty members based on department, name, or a combination of these criteria.
- Textbook Management: Students can search for textbooks by title, author, or ISBN. The system will display availability in the school library or recommend bookstores for purchase.
- Roommate Finder: This feature assists students in finding potential roommates by matching preferences such as move-in date, gender, and budget.
- Online Purchases: The platform facilitates convenient online purchases of meal plans and bus tickets using credit cards. Students can also manage their purchase history.
- Event Exploration: The website serves as a central hub for students to discover upcoming sports activities, parties, and student union elections. They can view schedules, select events of interest, and potentially register for participation.
- Student Union Election (Mock): The website will include a mock student union election poll to demonstrate its functionality in handling voting and displaying results through a bar chart.

Exclusions:

- Integration with external systems for textbook purchase beyond bookstore recommendations.
- Real financial transactions for meal plans and bus tickets.

- Functionality for registering or managing actual participation in events (beyond displaying information and potentially selecting events).

1.3 Definitions, Acronyms, and Abbreviations

1. req - requirement
2. sec - section
3. ref - reference

1.4 References

1.
2. "IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984, vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205. keywords: {Software engineering;System analysis and design;software;requirements;specifications},
3. Alshazly, A. A., Elfatatry, A. M., & Abougabal, M. S. (2014). Detecting defects in the software requirements specification. *Alexandria Engineering Journal*, *53*(3), 513-527.
4. Fabbrini, F., Fusani, M., Gnesi, S., & Lami, G. (2000, May). Quality evaluation of software requirement specifications. In *Proceedings of the Software and Internet Quality Week 2000 conference* (pp. 1-18).
5. European Union. (n.d.). GDPR information portal. GDPR.eu: https://gdpr.eu/
6. International Association of Privacy Professionals (IAPP). (n.d.). IAPP GDPR information portal. https://iapp.org/
7. Federal Trade Commission (FTC). (2023, January 19). Complying with COPPA https://www.ftc.gov/business-guidance/resources/complying-coppa-frequently-asked-questions

1.5 Overview

This SRS details a Student Activity Website. Students can manage accounts, search for people, find textbooks, and purchase items like meal plans and bus tickets. It also helps find roommates and explore events.

2 Overall Description

2.1 Product Perspective

This Student Activity Website is a web application designed for students at a university or college. It aims to serve as a central hub for various student needs and activities.

2.2 Product Functions

2.2.1 Student Management:

- Account Management: This includes student registration, login, and information updates.
- People Search: This function allows searching for other students and faculty by department or name, facilitating communication and collaboration.

2.2.2 Resource Management:

- Textbook Management: Students can search for textbooks by title, author, or ISBN. The system displays library availability or suggests bookstores for purchase.
- Roommate Finder: This functionality helps students find compatible roommates based on preferences like move-in date, gender, and budget.

2.2.3 Event Management:

- Online Purchases: Students can purchase meal plans and bus tickets conveniently using credit cards.
- Event Exploration: This section acts as a central hub for discovering upcoming campus events like sports activities, parties, and even a mock student union election (poll).

2.3 User Characteristics

The users of the Student Activity Website encompass a diverse group primarily consisting of students within an educational institution. Below are the general characteristics of the intended users:

Educational Level: The primary users are expected to be students enrolled in the educational institution, ranging from undergraduate to graduate level. Therefore, the educational level of the users typically spans from high school graduates to those pursuing advanced degrees.

Experience: Users are assumed to possess basic experience in navigating web-based platforms and interacting with online services. As students within an educational environment, they are likely to have previous experience with digital tools and online platforms, albeit with varying levels of proficiency.

Technical Expertise: While a basic understanding of internet usage and computer literacy is assumed, the technical expertise of the users may vary widely. Some users may be proficient in utilizing various online services, while others may require more guidance and intuitive interfaces. Therefore, the website should aim to cater to users with varying levels of technical expertise, ensuring accessibility and ease of use for all.

2.4 Constraints

This section outlines various constraints and limitations that will impact the development process of the Student Activity Website:

2.4.1 Regulatory Policies

The development of the website must adhere to relevant regulatory policies governing data privacy, security, and accessibility, such as GDPR (General Data Protection Regulation) or COPPA (Children's Online Privacy Protection Act), ensuring compliance with legal requirements and protecting user data.

2.4.2 Hardware Limitations

Scalability: The website might not initially handle a large user base, potentially requiring performance optimization or server upgrades as the user population grows.

Secure online payment processing is crucial. The website should integrate with a reputable payment gateway that adheres to industry security standards.

2.4.3 Interfaces to Other Applications

The website may require integration with existing applications or systems within the educational institution, such as student information systems or library databases. Compatibility and seamless interaction with these interfaces must be ensured during development.

2.4.4 Parallel Operation

The website should support parallel operation to accommodate multiple users accessing the platform simultaneously without performance degradation. This includes efficient handling of database transactions and concurrent user sessions.

2.4.5 Audit Functions

The incorporation of audit functions is essential for tracking user activities, system changes, and data modifications. This ensures accountability and facilitates troubleshooting and analysis of security incidents or compliance breaches.

2.4.6 Control Functions:

Control functions must be implemented to manage user access levels, permissions, and privileges within the system. This includes authentication mechanisms, role-based access control, and administrative controls for managing content and user accounts.

2.4.7 Higher-Order Language Requirements:

The development must be conducted using appropriate programming languages and frameworks capable of supporting the required functionality and scalability of the website. This may involve utilizing higher-order languages such as JavaScript, Python, PHP, and associated libraries and frameworks.

2.4.8 Signal Handshake Protocols:

Signal handshake protocols, such as XON-XOFF or ACK-NACK, need to be implemented for reliable communication between client and server components, particularly in scenarios involving data transmission or real-time interactions.

2.4.9 Reliability Requirements:

For a trustworthy user experience, the Student Activity Website must prioritize reliability. This includes minimal downtime, accurate data management, smooth performance, user-friendly error handling, and a disaster recovery plan to ensure website and data stability.

2.4.10 Criticality of the Application:

Considering the criticality of the application in facilitating student activities and services, robustness and stability are paramount. Any system failures or disruptions could significantly impact student engagement and campus operations.

2.4.11 Safety and Security Considerations:

The Student Activity Website prioritizes user privacy by securing sensitive information. This includes login credentials, personal details (names, addresses, email), financial information (credit cards for purchases), and potential activity preferences (roommate search, meal plans, events attended). Robust security measures are crucial to protect this data and ensure user trust.

2.5 Assumptions and Dependencies

The following factors can influence the requirements of the Student Activity Website:

- **External System Integration:** The website relies on integration with existing systems like the library database and university student database. Delays or limitations in access or functionality of these external systems can impact website functionalities. For example, if the library database integration encounters issues, the ability to display library availability for textbooks might be hindered.

- **Security Protocols:** The website's security measures depend heavily on the chosen payment gateway and its adherence to industry security standards. A reliable payment gateway ensures secure online transactions for purchases like meal plans and bus tickets.
- **User Device Compatibility:** While designed for accessibility across various devices (desktops, laptops, tablets, smartphones), limitations in browser capabilities or screen sizes on certain devices might necessitate adjustments to the user interface. The website might require testing and optimization for different screen sizes to ensure a consistent user experience across platforms.
- **User Adoption:** The success of the website relies on effective marketing and awareness campaigns within the university/college community. Without proper communication efforts, students might not be aware of the website's existence and its benefits, hindering user adoption.

2.6 Apportioning of Requirements

The below list of requirements may be delayed until future versions of the system.

- **Advanced People Search Options:** This could include searching by major or student year, allowing for more specific searches within the university community.
- **Textbook Price Comparison Tool:** An integrated tool could help students compare prices from different bookstores before purchasing textbooks.
- **Messaging System:** This would facilitate communication between users, allowing students to connect directly through the platform.
- **Club Management Features:** Dedicated functionalities could be implemented for student clubs to manage their activities and connect with members.
- **Forum Discussions:** A forum section could be introduced for students to interact on various topics, fostering a sense of community online.
- **Enhanced Event Management System:** Future versions could include functionalities like RSVP functionality for events and calendar integration for better event organization.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 **Login using username/password.**

Description	User authenticates the system with existing credentials.
Inputs	Username, password
Source	User input from login form
Outputs	Authentication status (success/failure)
Destination	User interface (feedback to user)
Requires	User must have a previously registered account with a valid username/password combination
Precondition	User is on the login page
Postcondition	Upon successful login, the user is granted access to authorized features. Upon failed login, the user is presented with an error message.
Side Effects	May trigger security actions (lockout after multiple failed attempts).

3.1.2 Search students/faculty by department, first and/or last name

Description	User searches for people within the university system
Inputs	Search terms (department, first name, last name - may be partial inputs)
Source	User input from the search form
Outputs	List of matching results, including name, department, phone, and email
Destination	User interface (displayed on the search results page)
Requires	An existing database of students and faculty
Precondition	User is on the people search page
Postcondition	Search results are displayed. If no results, an appropriate message is displayed.
Side Effects	None

3.1.3 Search textbooks by title, author, ISBN

Description	User completes the purchase of a selected meal plan.
Inputs	Selected plan type (monthly/semester), payment information (credit card details, billing address)
Source	User input from the meal plan purchase form
Outputs	Payment confirmation, updated user account balance (if applicable)
Destination	User interface (confirmation screen), payment gateway, back-end user account database
Requires	Integration with a secure payment gateway, user to have a registered account
Precondition	User has selected a meal plan and is on the payment page.
Postcondition	Payment processed; meal plan added to the user account. May trigger email confirmation to the user.
Side Effects	May affect inventory systems if meal plans have usage limits.

3.1.4 Online meal plan purchase (payment integration)

Description	User authenticates the system with existing credentials.
Inputs	Username, password
Source	User input from login form
Outputs	Authentication status (success/failure)
Destination	User interface (feedback to user)
Requires	User must have a previously registered account with a valid username/password combination
Precondition	User is on the login page
Postcondition	Upon successful login, the user is granted access to authorized features. Upon failed login, the user is presented with an error message.
Side Effects	May trigger security actions (lockout after multiple failed attempts).

3.1.5 Cast vote in student union election poll

Description	User participates in the election poll.
Inputs	User selection of a candidate
Source	User input from the election poll form
Outputs	Vote recorded.
Destination	Database storing poll results
Requires	The election poll to be active during a specified timeframe.
Precondition	User is logged in (possibly restrict one vote per user).
Postcondition	Vote is registered and counted towards the poll results.
Side Effects	May need mechanisms to prevent duplicate votes.

3.2 Qualitative Requirements (Non-Functional)

3.2.1 Usability

The website's ease of use is paramount, requiring an interface that is both intuitive and accessible to users with varying levels of technical proficiency. This necessitates a design ethos that emphasizes clarity and simplicity, ensuring that functions, menus, and elements are logically arranged and presented in an uncluttered manner. Furthermore, the system must be equipped to deliver clear error messages, promptly notifying users of encountered issues while offering precise guidance to expedite issue resolution, thereby bolstering the overall user experience and usability of the platform.

3.2.2 Accessibility

The website's dedication to inclusivity spans its compatibility with assistive technologies, guaranteeing effortless access for users with disabilities and fostering a diverse user base. Additionally, the system should offer alternative content formats, such as text transcripts for multimedia elements, catering to various user preferences and accessibility requirements, thereby enhancing the platform's inclusiveness and usability.

3.2.3 Performance

Fast search results and responsive page loading are imperative for optimal performance. The system should swiftly generate accurate search outcomes across activities listings, people directories, and textbooks, facilitating an efficient user experience with minimal query processing times. Furthermore, pages must load promptly on all devices and network conditions, with special attention given to critical features such as login and search functionalities. This ensures a seamless user journey, mitigating frustration and enhancing overall engagement with the platform.

3.2.4 Security

Protecting student data is paramount, requiring robust measures such as state-of-the-art encryption and access controls to safeguard sensitive information. Additionally, stringent password policies and reliable encryption mechanisms should be enforced to fortify user authentication processes, ensuring the security of user accounts against potential threats. These measures are crucial for maintaining data integrity and user trust in the system.3.2.5 Product Requirements

3.2.5 Reliability

Ensuring minimal downtime and robust error handling are vital for user satisfaction. The system must maintain high reliability and availability, minimizing service disruptions. Additionally, it should handle errors effectively to prevent interruptions and ensure a seamless user experience. Moreover, the system must be scalable to accommodate fluctuations in user traffic during peak periods, ensuring optimal performance.

4. Behavioral Description

4.1 System States

4.1.1 Login State

- Description: The system is in this state when a user attempts to log in.
- Events: User provides login credentials.
- Actions:
  - The system verifies user credentials.
  - If authentication is successful, transitions to the home state.
  - If authentication fails, displays an error message, and remains in the Login state.

4.1.2 Home State

- Description: The main state of the system after successful login.
- Events: The user selects a module or feature from the home screen.

- Actions:
    - Provides access to various modules such as Customer Management, Interaction Tracking, Sales Pipeline, and Reports.
    - Navigate to the selected module/state based on user input.

4.1.3 Customer Management State

- Description: State for managing customer records.
- Events: The user selects the Customer Management module from the Home screen.
- Actions:
    - Users can view, add, edit, or delete customer records.
    - The system updates the database based on user actions.

4.1.4 Interaction Tracking State

- Description: State for logging interactions with customers.
- Events: The user selects the Interaction Tracking module from the Home screen.
- Actions:
    - Users can log interactions such as calls, emails, and meetings.
    - The system records interaction details and associates them with respective customer records.

4.1.5 Sales Pipeline State

- Description: State for managing leads, opportunities, and deals.
- Events: The user selects the Sales Pipeline module from the Home screen.
- Actions:
- Users can manage leads, update opportunity stages, and close deals.
- The system updates the sales pipeline visualization and database based on user actions.

4.1.6 Reports State

- Description: State for generating and viewing reports.
- Events: The user selects the Reports module from the home screen.
- Actions:
    - Users can generate predefined or custom reports.
    - The system processes report requests, retrieves relevant data, and generates reports for user viewing or download.

4.2 Events and Actions

4.2.1 Login Event

- Description: The event is triggered when a user attempts to log in.

- Actions:
    - The user provides login credentials.
    - The system verifies credentials and performs authentication.

4.2.2 Navigation Events

- Description: Events are triggered when a user selects a module or feature from the home screen.
- Actions:
    - The user selects a module or feature.
    - The system navigates to the selected module/state.

4.2.3 Add/Edit/Delete Customer Event

- Description: Events triggered when a user adds, edits, or deletes a customer record.
- Actions:
    - The user initiates an add, edit, or delete action.
    - The system updates the customer database accordingly.

4.2.4 Log Interaction Event

- Description: The event is triggered when a user logs an interaction with a customer.
- Actions:
    - The user logs interaction details.
    - The system associate’s interaction with respective customer records.

4.2.5 Schedule Task/Reminder Event

- Description: Event triggered when a user sets a follow-up task or reminder.
- Actions:
    - The user sets tasks or reminders.
    - The system adds tasks or reminders to the user’s schedule.

4.2.6 Manage Sales Pipeline Event

- Description: Event triggered when a user updates the status of leads, opportunities, or deals.
- Actions:
    - User updates status.
    - The system reflects changes in the sales pipeline visualization and database.

4.2.7 Generate Report Event

- Description: Event triggered when a user generates a report.
- Actions:

- The user selects report parameters.
- The system processes report requests and generates reports for user viewing or download.

5. Validation and Criteria

5.1. Performance Bounds

Performance bounds specify the acceptable limits for various performance metrics of the Student Activity Website. These bounds ensure that the system meets the required performance standards under normal operating conditions. Key performance metrics include:

5.1.1. Response Time:

The maximum time allowed for the system to respond to user interactions, such as loading pages or executing queries. For example, the response time for loading the home page should be within 2 seconds under typical server load conditions.

5.1.2. Throughput:

The maximum number of transactions or requests the system can handle per unit time. This metric ensures that the system can efficiently serve multiple users simultaneously without degradation in performance. The system should support a throughput of at least 100 transactions per minute during peak usage hours.

5.1.3. Scalability:

The ability of the system to handle increasing user loads by adding resources such as servers or processing power. The system should demonstrate scalability by maintaining consistent performance even as the user base grows, with minimal impact on response time or throughput.

5.1.4. Resource Utilization:

Monitoring and optimizing resource utilization, including CPU, memory, and disk space, to ensure efficient operation and prevent resource exhaustion. The system should aim for optimal resource utilization, with CPU usage below 70% and memory usage below 80% under normal operating conditions.

5.2. Classes of Tests

To validate the functionality and performance of the Student Activity Website, the following classes of tests will be conducted:

5.2.1. Unit Tests:

Individual components and modules of the system will undergo unit testing to ensure they function correctly in isolation. This includes testing functions, methods, and classes to verify their behavior and functionality according to specifications.

5.2.2. Integration Tests:

Integration tests will verify the interaction and compatibility between different modules and components of the system. This includes testing data flow, communication protocols, and interface compatibility to ensure seamless integration.

5.2.3. System Tests:

System-level tests will evaluate the overall functionality and performance of the Student Activity Website as a complete system. This includes end-to-end testing of user scenarios, functional requirements, and performance benchmarks to validate the system's behavior under various conditions.

5.2.4. Performance Tests:

Performance testing will assess the system's response time, throughput, and scalability under different load conditions. This includes stress testing, load testing, and scalability testing to identify performance bottlenecks and ensure the system meets performance bounds.

5.2.5. Acceptance Tests:

Acceptance testing will involve user acceptance testing (UAT) to validate that the system meets user requirements and expectations. This includes conducting tests with real users to gather feedback and verify usability, functionality, and overall satisfaction.

5.3. Expected Software Response

The expected software response outlines the desired behavior and response of the Student Activity Website under different user interactions and system conditions. Key expectations include:

5.3.1. Fast and responsive user interface:

The website should provide a smooth and interactive user experience with minimal latency in loading pages and executing actions.

5.3.2. Accurate and relevant search results:

Search functionalities such as people search and textbook search should return accurate and relevant results based on user queries.

5.3.3. Secure online transactions:

The system should ensure the security and integrity of online transactions for purchasing meal plans and bus tickets, with encryption and authentication mechanisms in place.

5.3.4. Real-time event updates:

The website should display timely and up-to-date information on on-campus events, with notifications or alerts for newly added events or changes in schedules.

5.3.5. Intuitive user interactions:

The system should feature user-friendly interfaces and intuitive navigation paths, making it easy for users to access functionalities and perform tasks without confusion or errors.

5.4. Special Considerations

Special considerations for the validation and criteria of the Student Activity Website include:

5.4.1. Accessibility:

Ensuring that the website is accessible to users with disabilities, complying with accessibility standards such as WCAG (Web Content Accessibility Guidelines) to provide an inclusive user experience.

5.4.2. Security:

Implementing robust security measures to protect user data, including encryption, authentication, and authorization mechanisms to prevent unauthorized access or data breaches.

5.4.3. Compliance:

Ensuring compliance with regulatory requirements such as GDPR (General Data Protection Regulation) and COPPA (Children's Online Privacy Protection Act) to safeguard user privacy and data protection.

5.4.4. Usability:

Conduct usability testing to evaluate the ease of use and effectiveness of the website's interfaces and features, incorporating user feedback to improve usability and user satisfaction.

5.4.5. Performance Optimization:

Continuously monitoring and optimizing system performance to maintain responsiveness and scalability, addressing performance issues and bottlenecks proactively to enhance user experience.

6. Bibliography

- IEEE. (1984). IEEE Guide for Software Requirements Specifications. IEEE Std 830-1984.
- Alshazly, A. A., Elfatatry, A. M., & Abougabal, M. S. (2014). "Detecting defects in the software requirements specification." *Alexandria Engineering Journal, 53*(3), 513-527.
- Fabbrini, F., Fusani, M., Gnesi, S., & Lami, G. (2000, May). Quality evaluation of software requirement specifications. In *Proceedings of the Software and Internet Quality Week 2000 conference* (pp. 1-18).
- European Union. (n.d.). GDPR information portal. Retrieved from https://gdpr.eu/
- International Association of Privacy Professionals (IAPP). (n.d.). IAPP GDPR information portal. Retrieved from https://iapp.org/
- Federal Trade Commission (FTC). (2023, January 19). Complying with COPPA. Retrieved from [https://www.ftc.gov/business-guidance/resources/complying-coppa-frequently-asked-questions]

7. Appendix

7.1. Glossary of Terms

- SRS: Software Requirements Specification
- GDPR: General Data Protection Regulation
- COPPA: Children's Online Privacy Protection Act
- API: Application Programming Interface
- UI: User Interface
- UX: User Experience
- HTML: Hypertext Markup Language
- CSS: Cascading Style Sheets
- JS: JavaScript
- PHP: Hypertext Preprocessor
- SQL: Structured Query Language

7.2. References

7.2.1. Appendix A: Sample Input/Output Formats

- Input formats for user registration, search queries, and event registration, including data fields and validation criteria.
- Output formats for search results, event listings, and user profiles, detailing the information presented to users and the format in which it is displayed.

7.2.2. Appendix B: Background Information

- Detailed overview of the educational institution, including its size, student demographics, and technological infrastructure.
- Comprehensive analysis of existing systems and platforms that the Student Activity Website will integrate with, highlighting compatibility requirements and data exchange protocols.

7.2.3. Appendix C: Description of Problems

- In-depth exploration of the challenges faced by students, such as difficulty in managing accounts, locating resources, and participating in campus activities.
- A detailed explanation of how the Student Activity Website addresses these challenges through its features and functionalities, including user feedback and usability studies.

7.2.4. Appendix D: Packaging Instructions

- Detailed instructions for packaging the software code to meet security, export, and deployment requirements specified by the educational institution and regulatory authorities.
- Comprehensive overview of encryption and security measures implemented to ensure compliance with GDPR and COPPA regulations, including data protection policies and secure data transmission protocols.