Software Requirements Specification (SRS)

Project: LSUS Connect
Version: 1.0
Authors: Showrab Bin Habib, Robert Lovelace, Abid Talukder, Wei Chen

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for **LSUS Connect**, a centralized campus engagement platform for Louisiana State University Shreveport (LSUS). The document is intended for stakeholders including project sponsors, developers, testers, student government representatives, and end-users (students, faculty, and staff).

1.2 Scope

LSUS Connect will provide a secure, web-based and mobile-accessible platform to enable:

- Student marketplace (buy, sell, exchange items)
- Housing and roommate matching
- Club/organization promotion
- Peer-to-peer messaging
- Campus event announcements
- Job and internship postings
- Incoming students query

System Objectives:

- Enhance student engagement and networking.
- Provide a safe, LSUS-only community platform.
- Consolidate multiple student needs into one digital hub.

System Boundaries:

- In-scope: Student-driven interactions (listings, events, housing, communication).
- Out-of-scope: Academic course management (already covered by Moodle).

2. Overall Description

2.1 Product Perspective

LSUS Connect is a new platform that will integrate with the university's email system for authentication. It will be hosted in the cloud (AWS) and accessible via browsers and mobile devices.

2.2 Product Functions

Major functions include:

- User authentication via LSUS email
- User profile management
- Marketplace listings with images and descriptions
- Housing/roommate finder
- Event creation, browsing, and RSVP
- Messaging/chat system
- Notifications (email + in-app)
- Admin moderation and reporting tools

2.3 User Classes and Characteristics

- **Students**: Primary users; create/join events, list items, find roommates.
- **Clubs/Organizations**: Create and promote events.
- **Admins/Moderators (IT/Student Affairs/SGA)**: Oversee reported content, enforce rules.
- **Incoming students**: can post queries with guest mode.

2.4 Operating Environment

- Web browsers: Chrome, Edge, Firefox, Safari
- Mobile devices: iOS, Android (responsive web)
- Frontend: React JS, Tailwind, Javascript
- Backend: Node.js, Express.js
- Database: MongoDB Atlas

2.5 Design and Implementation Constraints

- Authentication limited to LSUS email domain
- Compliance with FERPA data privacy rules
- Limited budget and student developer availability

2.6 User Documentation

- Online help center and FAQs
- Quick-start video tutorials
- Step-by-step onboarding guide

2.7 Assumptions and Dependencies

- Students will have stable internet access.
- University IT will support SSO or email verification.
- Funding for cloud services will be maintained.

---

3. System Features (Functional Requirements)

Each feature is described with purpose, input, and output.

3.1 User Authentication

- **Description**: Students log in using LSUS email.
- **Input**: Email, password.
- **Output**: Successful login or error message.

3.2 Marketplace

- **Description**: Students post and browse items for sale.
- **Input**: Item details, images, price.
- **Output**: Visible listing to peers.

3.3 Housing/Roommate Finder

- **Description**: Students search for roommates or housing ads.
- **Input**: Preferences (budget, gender, location).
- **Output**: Suggested matches.

3.4 Event Management

- **Description**: Clubs/organizations post events, students RSVP.
- **Input**: Event details, RSVP action.
- **Output**: Event listing, RSVP confirmation.

3.5 Messaging/Chat

- **Description**: Real-time communication between students.

- **Input**: Text messages.
- **Output**: Message thread.

3.6 Admin Moderation

- **Description**: Admins review and act on reported content.
- **Input**: Reported post details.
- **Output**: Approved, removed, or flagged content.

---

4. External Interface Requirements

4.1 User Interfaces

- Responsive UI for mobile and web.
- Simple navigation (dashboard, categories, notifications).

4.2 Software Interfaces

- Integration with LSUS email authentication system.
- MongoDB database access.
- RESTful APIs for communication between front-end and back-end.

4.3 Communication Interfaces

- HTTPS for secure data transfer.
- SMTP for email notifications.

---

5. Non-Functional Requirements

5.1 Performance

- 95% of requests should complete in <2 seconds.
- Support 3,000 concurrent users.

5.2 Security

- Enforce LSUS email login.
- Encrypt passwords (bcrypt).
- Role-based access control for admins.

5.5 Scalability

- System expandable to 10,000+ users.

5.6 Maintainability

- Code follows ESLint & Prettier standards.
- Automated unit + integration testing.

---

6. Other Requirements

- Must comply with **FERPA** regulations for student data privacy.
- Must provide an admin audit log for accountability.