

WEEK 2

Software Requirements Specification (SRS) for CampusConnect

1. Functional Requirements

Functional requirements specify what the app must do. They are detailed through use cases, which describe system behavior, and user stories, which capture user needs and interactions.

Use Cases

- **User Registration and Authentication**
 - o **UC-1: Register with a university email (e.g., .edu) to ensure only students join.**
 - o **UC-2: Log in using email and password for secure access.**
 - o **UC-3: Reset forgotten password via email for account recovery.**
- **Profile Management**
 - o **UC-4: Edit profile information (e.g., name, bio, profile picture) to personalize accounts.**
 - o **UC-5: View other users' profiles to connect with peers.**
- **Home Feed**

- o **UC-6: View a feed of posts from friends, groups, or university announcements for updates.**
- o **UC-7: Post text, images, or links to the feed to share content.**
- o **UC-8: Like, comment on, or share posts to engage with others.**
- **Group Chats and Study Groups**
- o **UC-9: Create or join study groups for specific courses to collaborate academically.**
- o **UC-10: Send and receive real-time messages in group chats for instant communication.**
- **Event Calendar**
- o **UC-11: View a calendar of campus events, deadlines, and club activities for awareness.**
- o **UC-12: Add personal events or deadlines to manage schedules.**
- o **UC-13: Receive notifications for upcoming events to stay informed.**
- **Coursework Forums**
- o **UC-14: Post questions or resources in course-specific forums to support learning.**
- o **UC-15: Reply to forum threads to foster discussion.**
- **Marketplace**

- o **UC-16: List items for sale (e.g., textbooks, furniture) to trade within the community.**
- o **UC-17: Browse and search for items in the marketplace to find needed goods.**
- o **UC-18: Contact sellers or buyers through in-app messaging for transactions.**

User Stories

- **As a student, I want to join study groups for my courses so I can collaborate with classmates.**
 - **As a student, I want to see all campus events in one place so I don't miss important activities.**
 - **As a student, I want to buy and sell items within a trusted university community.**
 - **As a club leader, I want to post events and updates to reach more students.**
 - **As a user, I want real-time chat features to communicate instantly with my peers.**
-

2. Non-Functional Requirements

Non-functional requirements define the app's performance and quality attributes, ensuring it meets user expectations beyond core functionality.

- **Performance**

- o The home feed must load in under 2 seconds with a stable internet connection.
- o Real-time chat messages should appear within 1 second of being sent.
- Scalability
 - o The app must support at least 10,000 concurrent users without performance issues.
- Security
 - o All user data must be encrypted during transmission and storage for privacy.
 - o Only verified university students (via email) can access the app for exclusivity.
- Usability
 - o The app must be intuitive, with a learning curve of no more than 10 minutes for new users.
 - o The UI must be responsive and consistent across iOS and Android devices.
- Reliability
 - o The app must maintain 99.9% uptime, with no more than 1 hour of downtime per month.
- Compatibility
 - o The app must support the latest two versions of iOS and Android for broad reach.

- **Accessibility**
 - o The app should comply with WCAG 2.1 guidelines to ensure accessibility for users with disabilities.
-

3. System Constraints

System constraints establish the boundaries and limitations for the app's development and operation.

- **Technology Stack**
 - o The frontend must be built using Flutter for cross-platform compatibility.
 - o Firebase must be used for authentication, database, and real-time features.
 - **Device Compatibility**
 - o The app must run on devices with Android 8.0+ and iOS 12.0+.
 - **Internet Connectivity**
 - o The app requires an active internet connection, with no offline mode initially.
 - **Timeframe**
 - o The app must be launched within 6 months from the project start date.
-

4. Assumptions and Dependencies

Assumptions and dependencies clarify expectations and external factors critical to the project's success.

Assumptions

- **University students have access to smartphones running iOS or Android.**
- **Students are willing to adopt a university-specific social app over general platforms.**
- **Universities will provide email domains (e.g., .edu) for verification.**
- **Firebase's free tier will be sufficient for initial development and testing.**

Dependencies

- **External Services**
 - o **Firebase for authentication, database, and cloud functions.**
 - o **Third-party APIs for push notifications (e.g., Firebase Cloud Messaging).**
- **Development Tools**
 - o **Flutter SDK and Dart for frontend development.**
 - o **Android Studio and Xcode for testing on emulators and devices.**
- **Team Skills**
 - o **Developers must be skilled in Flutter, Dart, and Firebase.**

- o **Designers must understand Flutter's widget system for consistent UI/UX.**