EECS 581 Project 3

Team Number
Team 12

Team Members
- Liam Aga
- Aiden Barnard
- Reem Fatima
- Jorge Trujillo
- Alejandro Sandoval
- Pashia Vang

Project Name
Side Quest

Project Synopsis
A website that gives fun campus challenges and tracks your progress.

Architecture

Overview
Side Quest is a website / web app that makes exploring the University of Kansas campus more fun.
Instead of focusing only on halls, the website covers *any location on campus* - such as buildings, landmarks, and secret spots.
At each location, users can see short, fun challenges like:
- “Ride the sketchy elevator in Summerfield.”
- “Find the Jayhawk hidden near Strong Hall.”
- “Get a coffee from The Underground.”

When a user completes a challenge, they can check it off. The app keeps track of everything through a personal account, so progress stays saved across devices. This also allows progress to be compared between all users with statistics shown in the website.

Core Features
1. **Campus Locations**
    - Includes all types of spots: halls, libraries, and outdoor areas.
    - Each location has its own quests and short descriptions.
2. **Quest List**

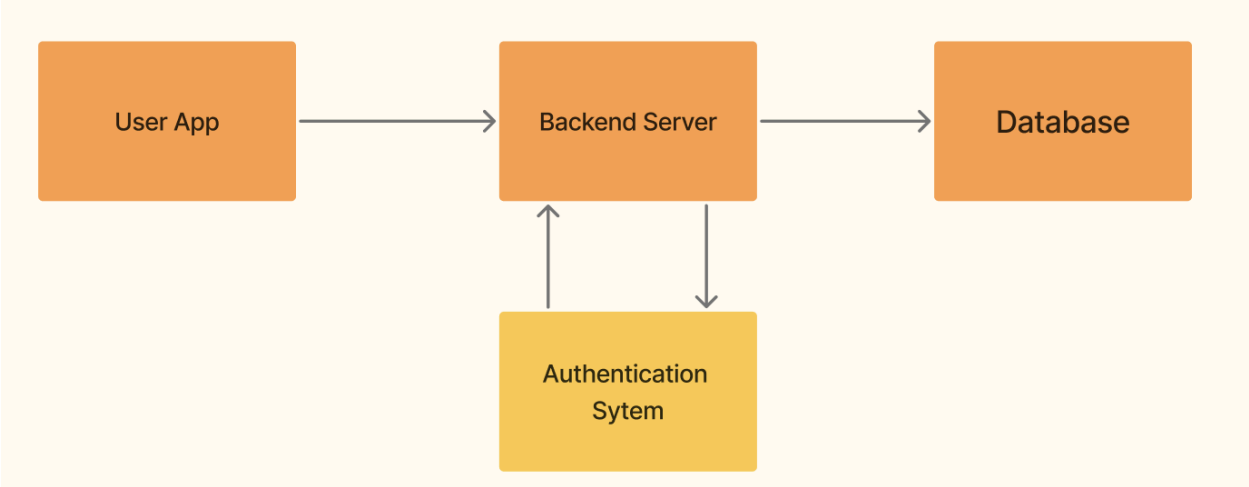
    - Shows fun tasks or dares you can do at each place.
3. **User Accounts**
    - Each user can sign up or log in.
    - Tracks completed quests and total progress.
4. **Progress Tracking**
    - Displays how many quests have been completed and which locations are still open.
    - Saves progress automatically under each user's account.
5. **Checkmark System**
    - Users can mark a quest as completed with a tap.
6. **Community Interaction**
    - Users can see their progress in leaderboards and compete for quest completion.
7. **Location Additions (Future Addition)**
    - Allow administrators to provide location and quest updates without modifying the code.

User Flow

1. The user opens the web app and logs into their account.
2. They see a map or list of campus locations.
3. They pick a location (for example, *Watson Library*).
4. The web app shows quests for that spot.
5. The user completes a quest and taps the checkmark.
6. Progress updates in their account automatically.

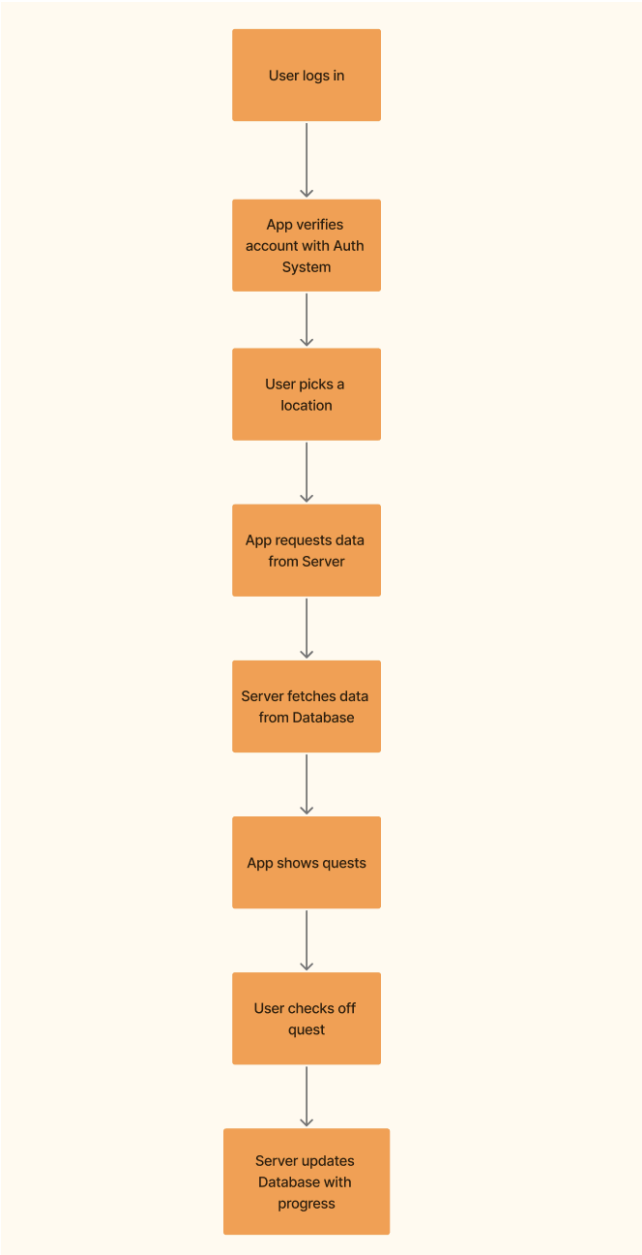
System Design

1. Diagram – High-Level Overview



- The **User App** displays data and lets users interact with quests.
- The **Backend Server** handles requests, sends data, and updates progress.
- The **Authentication System** manages user accounts and login sessions.
- The **Database** stores users, locations, and quest progress.

2. Diagram - Data Flow

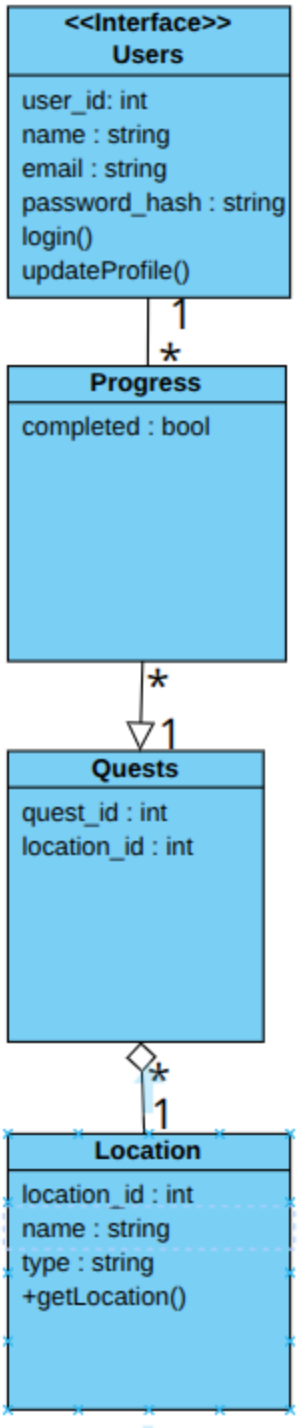


3. Diagram - Example Data Model

A database will be used to store user account information.

Table	Fields	Description
Users	user_id, name, email, password_hash	Stores login and profile info

Locations	location_id, name, type, description	Info about any KU location
Quests	quest_id, location_id, text	List of challenges for each spot
Progress	progress_id, user_id, quest_id, completed	Tracks which quests a user finished



Data Flow Example

1. The user first logs in with their Side Quest account.
2. When they view a location, the app fetches that location's quests.
3. After completing one, the user marks it done.
4. The app updates the *Progress* table under their account.

5. Their personal dashboard shows total quests completed and unlocked spots.

Possible Technologies

- **Frontend & Backend:**
  The app will use **Next.js**, which handles both the user interface and server-side logic in one framework. This makes it easier to manage pages, routes, and data requests all in one place.

- **Database:**
  Either **Firebase Firestore** or **PostgreSQL** will be used to store data like user accounts, locations, and quest progress. Firebase offers real-time updates, while PostgreSQL provides a strong relational structure for organized data.

- **Authentication:**
  Firebase Authentication or **Auth0** will handle user sign-up and login. These services will be used to make it simple to manage accounts securely, track user sessions, and connect with other accounts if needed.

User Experience Goals

- Clean and colorful design that feels fun and welcoming.
- Checkmarks and progress bars to show completion clearly.
- Easy navigation - users should find new locations and quests fast.
- Lighthearted tone with short, funny quests.
- Easily accessible, the web app should be compatible on various devices.

Future Plans

- Add a campus map that highlights completed and uncompleted locations.
- Introduce a **leaderboard** for most active users.
- Allow users to share their achievements with friends.