Open Brief Proposal

Michaela Kemp 231001

Open Window, School of Fundamentals

DV200

Lecturer(s): Tsungai Kutsuro

12 September 2024

Guardian Angel – A Women's Safety Companion App

Problem Statement:

Women in South Africa, as in many parts of the world, often face safety concerns when traveling alone, especially at night or through potentially unsafe areas. Fear of harassment or violence restricts their freedom of movement, putting them at risk. Many women feel anxious about traveling alone and don't have access to reliable solutions for finding companionship when they need it.

Significance:

The Guardian Angel app is designed to address this problem by connecting women who are traveling alone with nearby companions (other women) in real time. This will create a network of support, allowing women to travel more safely by finding trustworthy companions. The app will also feature emergency alerts and live location sharing, adding extra layers of security for users who may find themselves in unsafe situations.

How the Problem is Addressed:

Guardian Angel allows women to post and respond to travel requests, pairing users with others who are heading in the same direction. It provides users with the opportunity to feel safer when traveling by giving them a companion. The app will also include safety features like an emergency alert button and live tracking, helping users quickly get help if needed.

Target Audience:

Primary Users:

The main audience for this app is women aged 18 and above who frequently travel alone, particularly in urban and suburban areas. This includes university students, working professionals, and women who commute or travel during late hours.

Benefits:

- **Increased Safety**: By finding trusted companions, users will feel more confident and safe when traveling alone.
- **Supportive Community**: The app creates a community of women who can support each other by offering companionship when needed.
- **Emergency Assistance**: The app's emergency alert feature ensures users can quickly get help in dangerous situations.

Technology Stack:

Choice:

For this project, I have chosen the MERN stack (MongoDB, Express, React, Node.js) because it allows for the development of dynamic, interactive web applications using a unified language—JavaScript. MERN's real-time capabilities make it ideal for an app that needs to connect users quickly and reliably.

Components:

- **MongoDB**: Will be used to store user profiles, travel requests, and reviews.
- **Express**: Will manage routing and handle the API requests between the frontend and backend.
- **React**: Will power the user interface, allowing users to interact with the app smoothly.
- **Node.js**: Will handle the server-side processes, managing authentication, requests, and real-time data updates.

Application Features:

1. **User Registration & Login**:
    - **CRUD**: Users will be able to create an account (Create), log in to their profile (Read), update their personal details and preferences (Update), and delete their account (Delete).
2. **Post a Travel Companion Request**:
    - **CRUD**: Users can post requests for companions (Create), view available travel companions (Read), update their posted requests (Update), and cancel requests when they're no longer needed (Delete).
3. **Accept and View Companion Requests**:
    - **CRUD**: Users can view posted travel requests (Read), accept or decline companion offers (Update), and remove themselves from a trip (Delete).
4. **Emergency Alerts**:
    - **Feature**: Users can send an SOS message to their emergency contacts if they find themselves in danger.
5. **Review System**:
    - **CRUD**: After completing a trip, users can leave a review (Create), view ratings and reviews (Read), update their reviews (Update), and delete reviews if necessary (Delete).

Database Design:

Key Collections:

1. **Users**: Stores data like usernames, email addresses, emergency contacts, and personal preferences.
    - Fields: user_id, name, email, password, emergency_contacts, reviews.
2. **Travel Requests**: Contains the details of travel requests posted by users.

- Fields: request_id, user_id, destination, time, status, created_at.

3. **Reviews**: Stores ratings and feedback for travel companions.
    - Fields: review_id, user_id, companion_id, rating, comment.

User Interface and Experience:

UI/UX Design:

The app's user interface will be simple, clean, and easy to use, designed for users who may be on the move and need quick access to features. It will include a map view showing nearby companions, a feed of posted travel requests, and an easily accessible emergency button.

User Experience:

The user experience will focus on ensuring users can quickly and safely find travel companions and send emergency alerts if needed. The interface will minimize the number of clicks needed to perform key actions like posting requests or contacting companions.

Security Considerations:

Potential Risks:

1. **Data Breaches**: Sensitive information such as location and contact details could be exposed.
2. **Account Impersonation:** Unauthorized users may try to access other users' accounts.
3. **Unsafe Pairings**: Users may end up matched with untrustworthy companions.

Mitigation Strategies:

- **Input Validation**: All user inputs will be carefully validated to prevent any malicious data from being submitted.
- **Authentication**: Token-based authentication (JWT) will be used to ensure secure logins.
- **Authorization**: Proper access control will ensure that only verified users can access travel requests and profiles.
- **Encryption**: All data will be transmitted securely over SSL using HTTPS.

Project Timeline:

Week 1-2:

- Finalize app concept, design wireframes, and plan the database structure.

Week 3:

- Set up the project repo, configure MongoDB, and implement the basic authentication system.

Week 4-5:

- Build the main features like posting and accepting companion requests.
- Implement CRUD operations for user profiles and travel requests.

Week 6:

- Add emergency features like live tracking and push notifications.

Week 7-8:

- Complete testing, refine the user interface, and prepare for deployment on cloud platforms.

Challenges and Risks:

Potential Challenges:

1. **Real-Time Tracking**: Ensuring accurate live location sharing may be complex.

Strategies:

- **Tracking**: Use reliable services like Google Maps API for real-time location tracking.

Conclusion:

The **Guardian Angel** app offers a practical solution to a significant real-world problem: women's safety while traveling alone. By creating a community of support and offering essential safety features like real-time companion finding and emergency alerts, this app has the potential to make a meaningful difference. With its focus on core CRUD operations, user security, and a simple user experience, Guardian Angel will be a valuable tool for ensuring safer travel for women.