11.09.2024

Mobile Information Systems Project

Flutter App

Faculty of Computer Science and Engineering

Students
Iva Dodikj 201506
Beyza Ismail 201511

11.09.2024

Contents

11.09.2024

Project Overview

NeighborNet is a mobile application designed to promote community engagement within interconnected building complexes. The app offers a centralized platform where residents can interact, share information, and stay updated on local events. With a focus on user-friendly features, NeighborNet aims to enhance communication and involvement within the community by allowing users to report issues, participate in events, and connect with neighbors.

Key Features

1. **Community Forum:** Users can engage in discussions, share updates, and exchange information with fellow residents.
2. **Event Management:** Users can create and view upcoming events in their area, fostering a sense of community by enabling participation in local activities.
3. **Issue Reporting:** The app enables users to report maintenance issues or other concerns directly to building management, complete with image uploads for enhanced clarity.
4. **User Authentication:** Firebase Authentication ensures secure login and registration processes, providing users with a personalized experience.
5. **Real-time Database:** Firebase Firestore is used to store community events, user profiles, and issue reports, enabling real-time updates and seamless data retrieval.
6. **Location Services:** The app integrates Google Maps to offer users a visual overview of their neighborhood, including community points of interest.

The application provides a smooth and intuitive interface, combining essential functionalities into a single app, aimed at boosting residents' involvement in community life.

Architecture

NeighborNet is developed using Flutter, ensuring a cross-platform experience for both Android and iOS users.

Frontend

The frontend is designed to be user-friendly, with a focus on clear navigation and seamless user interactions.
Key UI components include:

1. **Registration and Login:** Custom forms for user sign-up and authentication.
2. **Dashboard:** Displays events and recent updates from the neighborhood.
3. **Issue Reporting:** Allows users to submit issues through a clean and simple form, with an option to upload pictures.

4. **Events Page:** Lists upcoming events and enables users to create and share new events.
5. **Map Integration:** Uses Google Maps API to display a user's location and important landmarks in the vicinity.

Backend

NeighborNet uses Firebase as its backend, which includes:

1. **Firebase Authentication:** Manages user sign-up, login, and authentication sessions securely.
2. **Firebase Firestore:** Serves as the app's database, storing information about users, events, and issues reported.
3. **Firebase Storage:** Handles image uploads securely, associating images with specific issue reports.
4. **Google Maps API:** Powers the neighborhood map, offering users location-based insights and navigation.

The integration of Firebase ensures real-time synchronization of data across all users, providing seamless interaction with the app.

Technologies Used

NeighborNet utilizes a wide range of technologies to ensure a smooth, scalable, and secure user experience:

- **Flutter:** For building the cross-platform mobile interface.
- **Firebase Authentication:** To manage user login and registration securely.
- **Firebase Firestore:** A real-time NoSQL database that stores user data, events, and reported issues.
- **Firebase Storage:** To securely handle uploaded images.
- **Google Maps API:** For rendering interactive maps and showing the user's location relative to nearby landmarks.

Data Flow

NeighborNet's data flow revolves around Firebase services:

- **User Data:** Registration information and user profiles are stored securely in Firebase Authentication and Firestore.
- **Events and Issues:** Events and issues reported by users are stored in Firestore, enabling real-time updates for all users.
- **Image Storage:** Uploaded images are stored in Firebase Storage and linked with relevant documents in Firestore.

All interactions with the Firebase backend are real-time, ensuring that any changes are reflected immediately across the app.

Security Considerations

NeighborNet incorporates several security features to ensure data protection and user privacy:

1. **Firebase Authentication:** Securely handles user authentication and authorization, utilizing best practices for password management and data encryption.
2. **Firestore Security Rules:** These rules are set up to ensure only authorized users can view or modify sensitive data, such as personal information or issue reports.
3. **Secure Image Storage:** Images uploaded by users are stored securely using Firebase Storage, ensuring that only the relevant users or building management have access to the files.
4. **Data Encryption:** Both user data and uploaded images are encrypted both in transit and at rest.

These security measures guarantee a safe environment for users to share information without compromising privacy.