SE 4485: Software Engineering Projects

Fall 2024

Architecture Documentation

|  |  |
| --- | --- |
| Group Number | 8 |
| Project Title | Design, Develop, and Test a Location Verification App to be Installed on the Two Smartphone Alternatives |
| Sponsoring Company | The Fellows Consulting Group |
| Sponsor(s) | Tom Hill |
| Students | 1. Abdullah Chaudhry 2. Omar Hussain 3. Reya Dawlah 4. Danya Almintakh 5. Alvin Mathew 6. Mageto Nyakoni |

TITLE PAGE

ABSTRACT

* brief summary of the entire document

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

* introduction to the entire document ● purpose and scope of the document
* description of the structure of the document

ARCHITECTURAL STYLE(S) USED (Alvin)

* how does the architecture support various features of your application

1. Client-Server Architecture: The devices communicate with a central server that manages data and authentication services
   * + Real-Time Data Exchange: Client-server allows for seamless real-time communication between homeowner and service personnel’s mobile devices and the server
     + Scalability: Client-server supports adding new features or increasing the number of users without worsening performance
2. Microservices Architecture: The app uses different functions (location tracking, identity verification, user management) and they are handled by separate, loosely coupled services
   * + Modular Development: Each feature is built as an independent service so updating each feature is easy without affecting the entire system
     + Platform Independence: Microservices can be deployed independently on both iOS and Android
3. Event-Driven Architecture: The app uses event-driven architecture to respond to triggers (location updates, verification requests) in real-time
   * Real-Time Tracking: Service personnel’s location is constantly updated and streamed to homeowner through event-driven mechanisms
   * Immediate Notifications: When the app detects significant events (ex. the service personnel arriving at location), notifications are pushed to homeowner and supervisor

ARCHITECTURAL MODEL (Mageto, Abdullah)

* + use packages stereotyped as subsystems <<subsystem>>
  + no classes in the architectural model

TECHNOLOGY, SOFTWARE, AND HARDWARE USED (Danya)

* + describe the technology used for implementing this project
  + list all software and hardware required to support the technology
  + explain the communication between application server and database server

**Technology Description:** This project is implemented using Flutter, a cross-platform UI toolkit that allows for building natively compiled applications for mobile, web, and desktop from a single codebase. The core functionality of this application involves tracking the real-time location of service personnel, achieved using the geolocator package for accessing GPS services and obtaining location data. The app uses asynchronous programming (async/await) to handle real-time location updates and user interactions efficiently.

**Software Requirements:**

1. **Flutter SDK**: For cross-platform development, supporting both iOS and Android.
2. **Dart Programming Language**: Flutter applications are written in Dart.
3. **Geolocator Package**: A Dart package for accessing GPS services, allowing the app to get the current location and listen to location updates.
4. **Firebase (Optional)**: For real-time database functionalities and managing user authentication.
5. **Visual Studio Code or Android Studio**: Integrated Development Environments (IDEs) for writing, testing, and debugging the code.

**Hardware Requirements:**

1. **Smartphones/Tablets**: Devices running either iOS or Android, with GPS capabilities to support location tracking.
2. **Server**: A cloud server (e.g., AWS, Google Cloud) to host the backend and database for managing user data, location history, and communication between clients.
3. **Computers/Laptops**: For developing and testing the application, equipped with enough memory and processing power to run emulators and compile the application.

**Communication Between Application Server and Database Server:** The communication between the application server and the database server typically follows a client-server model:

* The **client app** (Flutter app) sends HTTP requests to the **application server** for operations like user authentication, location updates, and fetching location history.
* The **application server** processes the request and interacts with the **database server** to retrieve or store data.
* The **database server** (e.g., a real-time database like Firebase or an SQL/NoSQL database) stores information such as user profiles, authentication tokens, and real-time location data.
* The **application server** sends the requested data or a response back to the **client app**.
* The app uses this data to update the UI, such as displaying the current location of service personnel or generating service reports for supervisors.

This setup ensures that data exchange is secure, synchronized, and consistent across all clients while allowing the app to scale and support real-time updates.

RATIONALE FOR YOUR ARCHITECTURAL STYLE AND MODEL (Omar)

TRACEABILITY FROM REQUIREMENTS TO ARCHITECTURE (Reya)

* + provide a mapping between requirements and architecture
  + clearly describe how each requirement in the *Requirements Documentation* is captured in the architecture

EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

* + students should work with their project sponsor(s) to identify all the standards and constraints that should be applied for preparing this document

ADDITIONAL REFERENCES

* + include other related references that are not included the section above