Technical Project Report - Android Module

GoGetIt!

Subject:	Introdução à Computação Móvel
Date:	Aveiro, 19/06/2024
Students:	108615: Tiago Fonseca Cruz 107572: Gonçalo Rafael Correia Moreira Lopes
Project abstract:	The "GoGetIt" application is designed to streamline the food ordering and delivery process, catering to both customers and delivery personnel. Key achievements include real-time order tracking, integration with Firebase for backend support, and a user-friendly interface for both customers and delivery agents.

Report contents:

1 Application concept

Purpose: The "GoGetIt" app is designed to simplify the food ordering and delivery process. It targets customers looking for convenient meal delivery from their favorite restaurants and delivery personnel seeking an efficient system for managing and delivering orders.

Target Users:

- **Customers:** Can browse restaurants, place orders, and check order history.
- **Delivery Personnel:** Manage order pickups, navigate to customer locations, and confirm deliveries.

Benefits:

- **For Customers:** Enjoy a hassle-free ordering experience, ensuring timely and reliable meal deliveries.
- **For Delivery Personnel:** Benefit from an organized workflow that reduces delivery times and enhances service quality, leading to increased job satisfaction and customer trust.

2 Implemented solution

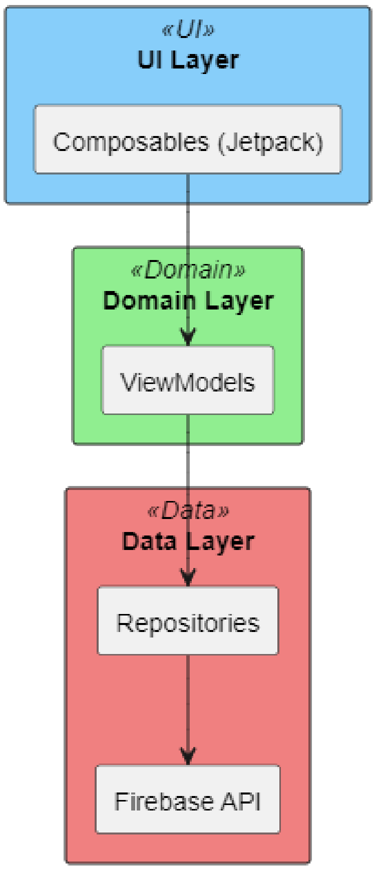
Architecture overview (technical design)

- **Architecture:** The app follows the MVVM (Model-View-ViewModel) architecture, ensuring a clear separation of concerns.

- **Backend Integration:** Firebase is used for database management, authentication, and real-time updates.

- **Components:**
  - **UI Layer:** Developed using Jetpack Compose, leveraging composable functions to create a dynamic and responsive user interface.
  - **Data Layer:** Consists of repositories that interact directly with Firebase, handling data storage, retrieval, and updates efficiently.
  - **Domain Layer:** Comprises ViewModels that manage business logic, ensuring smooth interaction between the UI and Data layers, and providing a responsive and interactive user experience.

Diagram:



Data Models and Persistence:

- **Firebase:** Utilized for real-time database, storing order details, user information, and restaurant data.

- **Data Structures:** Kotlin data classes are employed to represent key entities within the app, such as Order, Restaurant, and User.

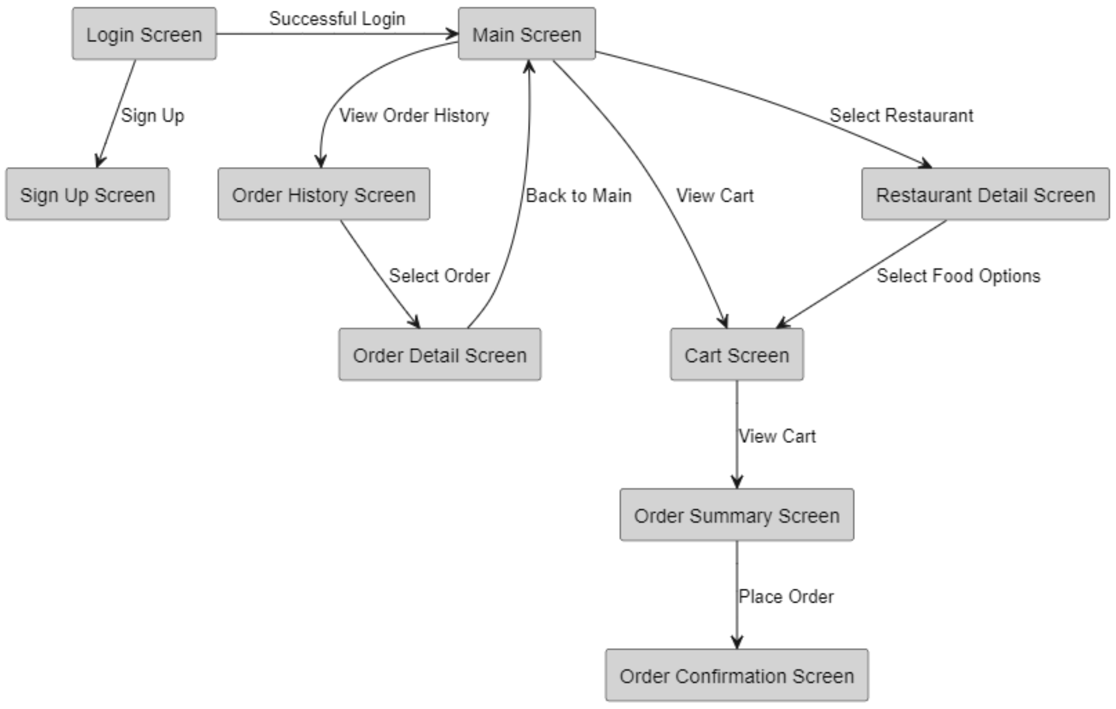
- **Synchronization:** Firebase listeners are used to enable real-time updates, ensuring data consistency between the app and the backend. This approach ensures that any changes made in the backend are immediately reflected in the app.

Implemented interactions

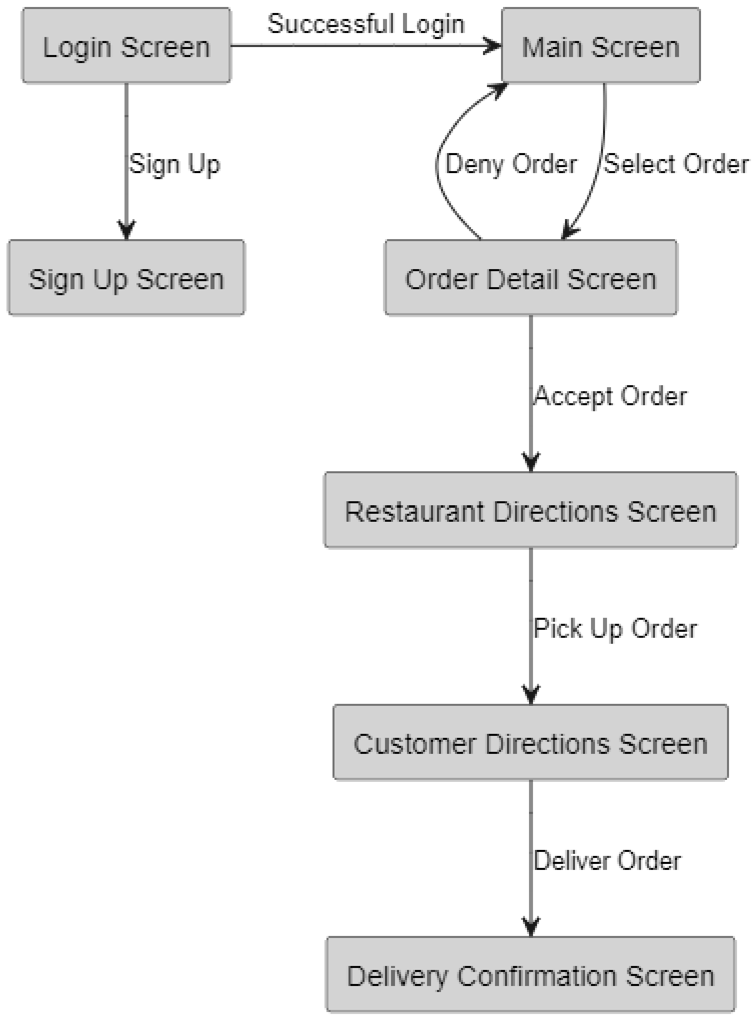
- **Main Interactions:**
  - **Customer Journey:** Login -> Browse Restaurants -> Place Order -> Confirm Delivery.
  - **Delivery Personnel Journey:** Login -> View Assigned Orders -> Navigate to Pickup -> Deliver Order -> Confirm Delivery.

Visual Navigation Map:

Main Costumer App:



Delivery Personnel App:



Project Limitations

- **Incomplete Features:**
  - Advanced filtering of restaurants by cuisine type.
  - In-app chat between customers and delivery personnel.
- **Known Bugs:**
  - Occasional delays in real-time updates.
  - GPS location inaccuracies in some devices.

3 Conclusions and supporting resources

Lessons learned

- **Technical Challenges:**
    - **Firebase Integration:** Required a deep understanding of asynchronous data handling, which was more complex than initially expected.
    - **Real-time Location Tracking:** Proved to be more challenging than initially anticipated.

- **Surprising Elements:** Jetpack Compose greatly simplified the UI development process, providing a reactive and declarative approach to building interfaces.

- **Course Suggestions:** More focus on advanced topics like real-time data synchronization and state management would be beneficial for future students.

Work distribution within the team

In the project the work distribution was 60% for Tiago Cruz and 40% for Gonçalo Lopes.

Project resources

Resource:	Available at:
Code repository:	https://github.com/TiagoC18/GoGetIt
Ready-to-deploy APK:	Inside of code repo

Reference materials

- **Firebase Documentation:** Essential for integrating real-time database and authentication.
- **Jetpack Compose:** Official Documentation for building the UI.
- **Android Developer Blog:** Various posts on best practices and architectural patterns.