

Android Development
Project Report

Moters: The Material Version of toters



Université Saint-Joseph de Beyrouth
Faculté d'ingénierie et d'architecture
Institut national des télécommunications et de l'informatique

January 6, 2025
Submitted by

Ghady Youssef
ghady.youssef@net.usj.edu.lb

Antoine Karam
antoine.karam3@net.usj.edu.lb

Repository Link

Application Overview

The application is designed to streamline food ordering and delivery, offering users a seamless experience to discover restaurants, order and track deliveries. Its primary features include:

- Secure user authentication using JWT.
- Browse restaurants, add food items to the cart, and place orders.
- Mark your favorite restaurants for quick access in the future.
- Submit and view restaurant reviews.
- Track order history, including past and pending orders.
- Access detailed order information, such as the current status of an order.
- Order tracking based on your location and the restaurant's location.
- Rate delivery drivers to provide feedback.
- View and update personal details through the user profile.

Technical Details

The application leverages the following tools, libraries, and design principles:

- **Pattern:** MVVM (Model-View-ViewModel).
- Separation of concerns for scalability and testability.

Technologies:

- **Programming Language:** Kotlin.
- **Libraries:**
 - Retrofit for API calls.
 - Room for local database.
 - Jetpack Compose for a declarative UI.
 - Google Play Services (GM) to access location data.
 - LiveData and ViewModel for UI-related data.
 - Osmroid for the map rendering.
 - JWTDecode for token authentication.
- **Design Patterns:**
 - Singleton for shared resources.
 - Repository for data management.

Challenges and Solutions

During development, several challenges were encountered:

- **Token persistence across app sessions**
Solution: The JWT token was stored securely in SharedPreferences and validated using the JWTDecode library.
- **Efficient state management across activities**
Solution: Shared ViewModel instances were used to maintain consistency between activities.
- **Integrating Compose with View-based UIs**
Solution: A clear project structure allowed us to easily collaborate and integrate both UI technologies into the project.

Setup Instructions

To set up and run the application, follow these steps:

1. Clone the repository: `git clone https://github.com/Ghaady/delivery-app`.
2. Open the API in Visual Studio Code (Make sure you have .NET 8 installed).
3. Install PostgreSQL.
4. Resolve all dependencies and set up the database.
5. Run the server.
6. Open the project in Android Studio.
7. Ensure that you have the required dependencies installed:
 - Android SDK 34 is a working version.
 - Gradle 8.7 is a working version.
8. Build the project and resolve dependencies.
9. Run the app on an emulator or physical device.