

KHAN INSTITUTE OF COMPUTER
SCIENCE AND INFORMATION
TECHNOLOGY

COMPUTER SCIENCE DEPARTMENT

FoodApp: A Dual-Role Mobile Food Ordering System

FINAL PROJECT REPORT

Subject: Mobile Application Development (BSCS-5B)

Instructor: Sir Uzair Hassan

Author:

Muhammad Harib

Registration: [232201069]

Department:

Computer Science

January 6, 2026

Contents

1	Project Abstract	2
2	System Architecture and Design	2
2.1	Database Schema Design	2
3	Technical Implementation	2
3.1	Core Technologies	2
3.2	Implementation of Content Management	3
4	Challenges and Debugging	3
5	Conclusion and Future Scope	3

1 Project Abstract

The **FoodApp** is a native Android application designed to bridge the gap between local food vendors and customers. Developed using Java and integrated with Firebase, the system provides a dual-interface experience. Customers can browse dynamic menus, manage a shopping cart, and track orders in real-time. Simultaneously, administrators can manage the digital storefront, update categories, and process incoming orders. This project addresses modern UI/UX challenges and implements a robust backend-as-a-service (BaaS) architecture.

2 System Architecture and Design

The application follows the **Model-View-Controller (MVC)** pattern, ensuring high maintainability and scalability.

2.1 Database Schema Design

The data is stored in a NoSQL format within the Firebase Realtime Database. Key nodes include:

- **Category:** Stores category metadata including `Name`, `Id`, and `ImagePath`.
- **Foods:** Contains food details, linked to categories via `CategoryId`.
- **Orders:** Tracks user transactions, including address and status (Pending/Delivered).

3 Technical Implementation

3.1 Core Technologies

Component	Technology Used
Programming Language	Java (JDK 11+)
Development Environment	Android Studio (Ladybug/Flamingo)
Backend Services	Firebase (Auth, Database, Storage)
Image Processing	Glide Library
Navigation	Chip Navigation Bar

Table 1: Project Technology Stack

3.2 Implementation of Content Management

The Administrative side uses specific activities like `AddFoodActivity` to populate the database. A critical design choice was implemented to use numeric IDs to prevent application crashes during user-side data parsing.

```
1 // Generating a numeric ID to match legacy DB structures
2 int numericId = (int) (System.currentTimeMillis() % 100000);
3 Map<String, Object> map = new HashMap<>();
4 map.put("Id", numericId);
5 map.put("Title", foodTitle);
6 ref.child(String.valueOf(numericId)).setValue(map);
```

Listing 1: Example of Numeric ID Generation

4 Challenges and Debugging

A major technical hurdle involved **Dependency Hell**. The navigation library used conflicted with the modern `androidx.activity` versions.

Solution: By implementing a Gradle constraint, we unified the fragment library version across the project:

```
1 constraints {
2     implementation("androidx.fragment:fragment:1.7.1") {
3         because("Force version to fix FragmentManager stability")
4     }
5 }
```

5 Conclusion and Future Scope

The FoodApp successfully demonstrates the power of real-time cloud integration in mobile apps. Future development will focus on:

- **Payment Integration:** Deployment of Stripe or Braintree for secure transactions.
- **Real-time Tracking:** Using Google Maps API for delivery driver localization.
- **Push Notifications:** Alerting users when an admin changes order status to "Delivered".