Bytewise Limited

Fellowship

Documentation

(DOCUMENTATION DOCUMENT)

for

Android App (Digital Yazman)

Version 1.0

By

Ali Hamza

To:

Kashif Mehmood

Table of Content

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by:

Kashif Mahmood

Signature_____

1. Introduction

The City Android App is designed to meet the goals and objectives of enhancing the overall experience of the residents in the city. It is aimed at providing a user-friendly interface with easy-to-navigate features that enable users to access information, news, and services with ease.

The scope of the app includes providing information related to the emergency services, public transportation, local news, opportunities, and local businesses. The key stakeholders of the app include the residents, local businesses, and emergency services.

1.1. Purpose

The purpose of City Android app is to provide residents with a comprehensive guide to the city, including information on local news, businesses, transportation, and services. The app is designed to enhance the user experience by offering an intuitive and user-friendly interface that enables easy navigation and access to relevant information. Through this app, users will be able to explore and discover all that the city has to offer.

2. Design methodology and software process model

Design methodology refers to the systematic approach of designing the software application. For the City Android App, the following design methodology and software process model can be used:

1. **Agile software development:** This model can be used as it allows for flexibility in the design process and promotes collaboration among team members.
2. **Object-oriented design:** This methodology can be used to design the application as it allows for modular and scalable design, making it easier to maintain and modify the codebase.
3. **User-centered design:** This methodology can be used to ensure that the app meets the needs of its users, taking into consideration the feedback and preferences of the target audience.
4. **Model-View-Controller (MVC) architecture:** This architecture can be used to design the application as it separates the presentation layer from the business logic, making it easier to modify and test the application.

3. System overview

The city app is designed to provide a platform for the residents of a small city to access information and services related to their city through their mobile devices. The app aims to provide a user-friendly interface that allows users to quickly and easily access information related to local news, events, public services, and local businesses.

The app will be built for the Android platform and will be developed using the Kotlin programming language. The app will also incorporate various third-party dependencies. The app will be designed

with a modern and intuitive user interface, with the goal of making it easy for users to navigate and find the information they need.

The system will be hosted on a cloud-based server, which will allow for scalable and flexible hosting and maintenance of the app. The app will also incorporate various security features to protect user data and ensure the safety and privacy of users.

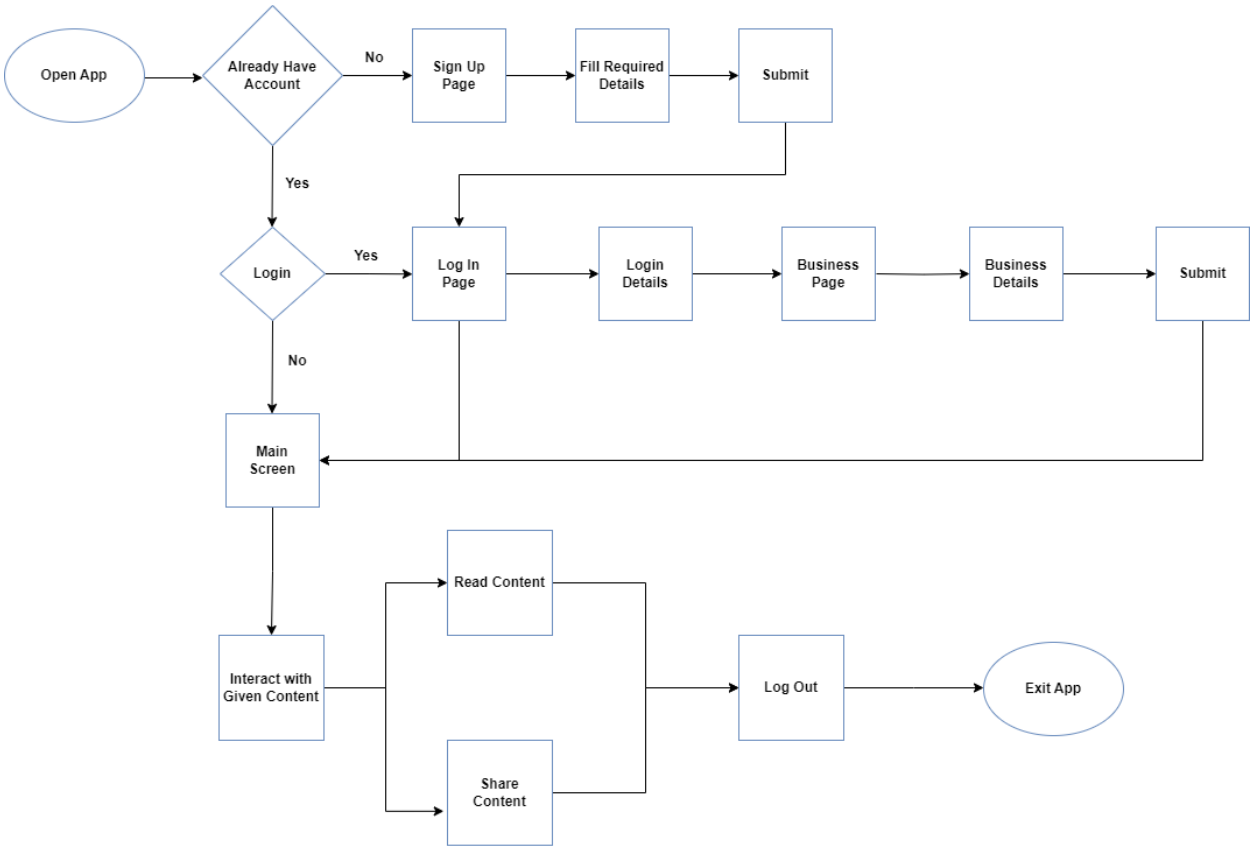
3.1. Architectural design

The architecture of the City Android App will follow the Model-View-Controller (MVC) design pattern. This pattern separates the application logic into three interconnected components: the Model, the View, and the Controller.

1. **Model:** This component will handle the data and business logic of the application. It will store all the data related to the app's functionalities, such as businesses, news, and city information.
2. **View:** This component will be responsible for displaying the app's user interface to the end-users. It will include screens for displaying the city's news, events, city information, and other features.
3. **Controller:** This component will act as an intermediary between the Model and the View components. It will receive user input from the View component, process it using the Model component, and update the View component with the results.

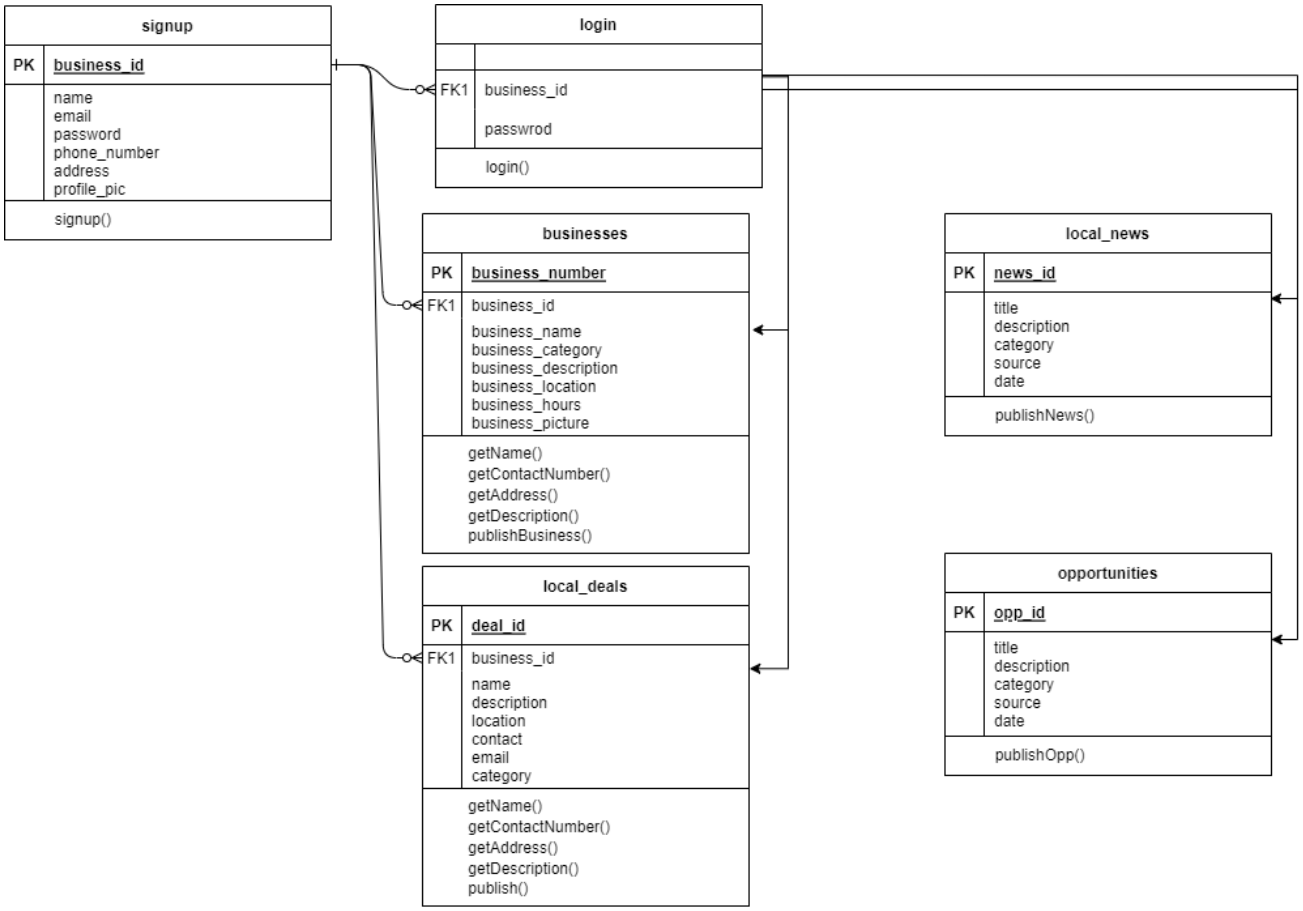
Overall, the architectural design of the app will be scalable, modular, and easy to maintain.

3.2. Process flow/ Representation



Process Flow Diagram

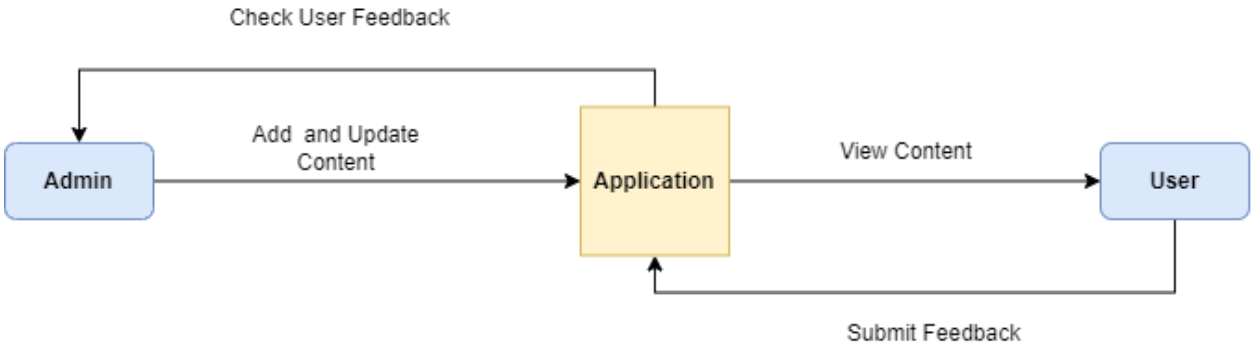
4. Class Diagram



Class Diagram

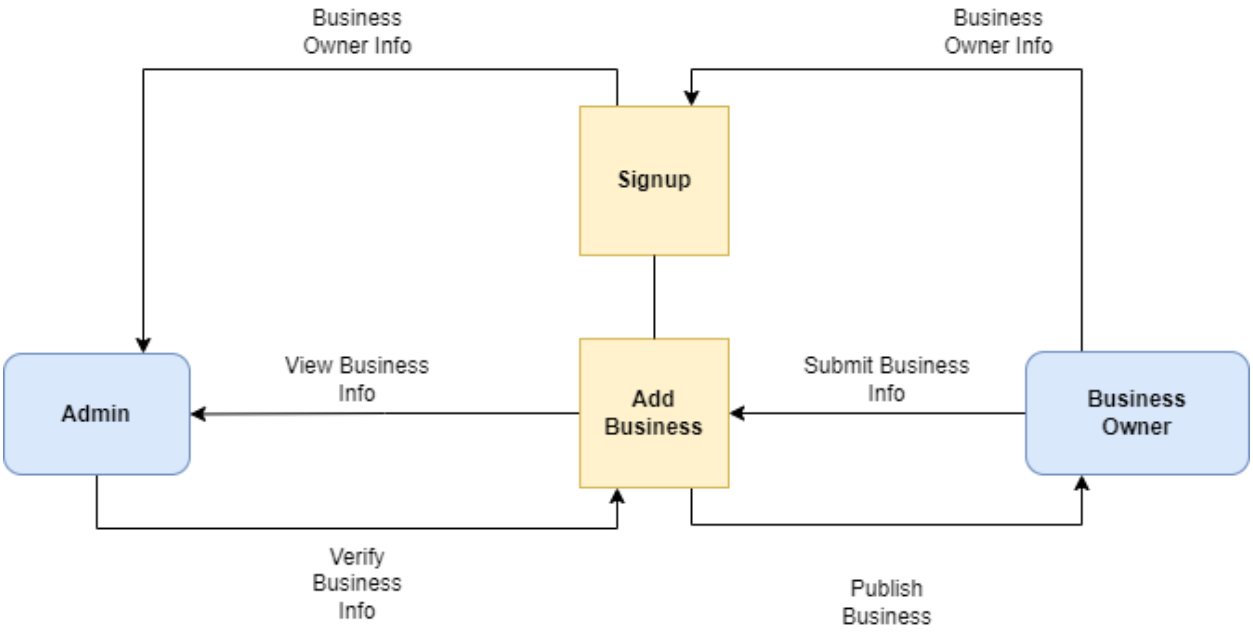
5. Data Flow Diagram

5.1. Level-0 DFD



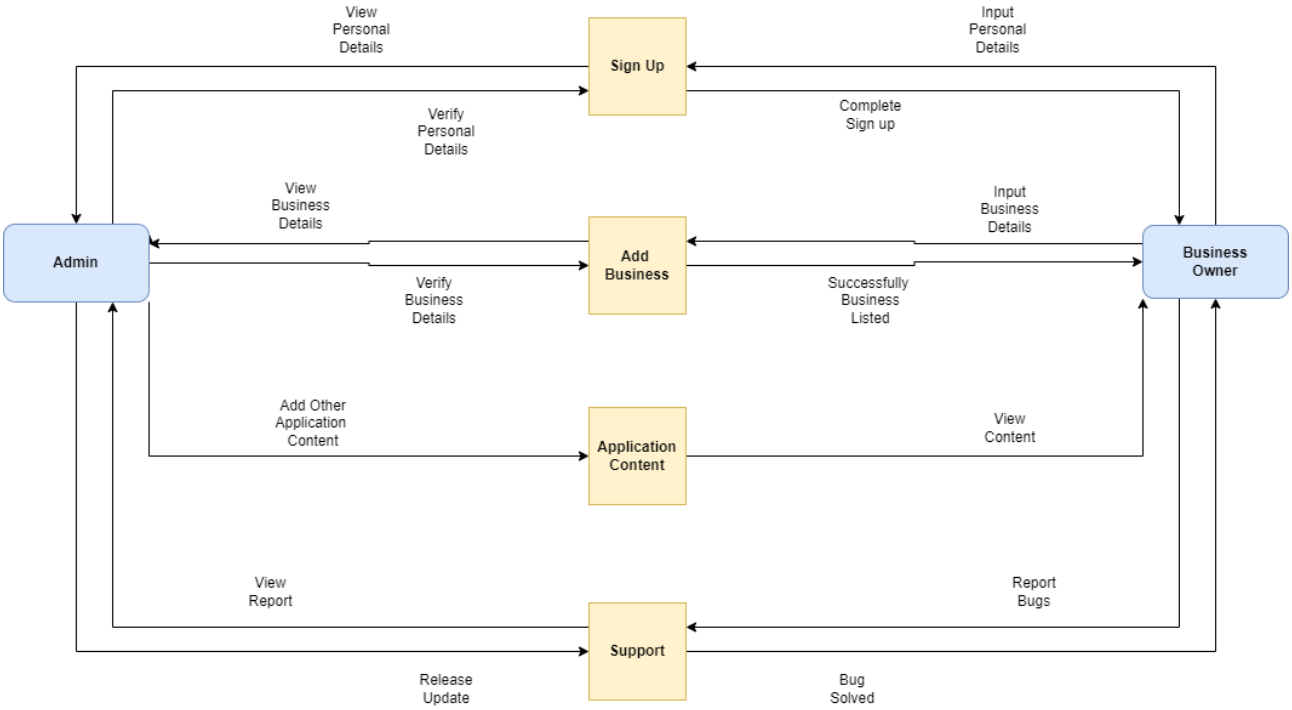
Data Flow Diagram 1

5.2. Level-1-DFD



Data Flow Diagram 2

5.3. Level-3-DFD



Data Flow Diagram 3

6. Data Design:

6.1. Data Directory

The data dictionary can include information such as data element name, data type, length, allowed values, and usage notes. Data directory for Business, local deals and opportunity entities are following:

Business Entity

Attributes	**Type**	**Description**
business_id	Integer	Unique identifier for the business
name	String	Name of the business
address	String	Address of the business
phone_number	String	Phone number of the business
email	String	Email of the business
category	String	Category of the business
Description	String	Description of the business
image	Image	Image of the business

Local Deals Entity

Attributes	**Type**	**Description**
deal_id	Integer	Unique identifier for the deal
Business_id	Integer	ID of the business offering the deal
title	String	Title of the deal
description	String	Description of the deal
start_date	Date	Start date of the deal
end_date	Date	End date of the deal
image	Image	Image for the deal

Opportunities Entity

Attributes	Type	Description
opprotunity_id	Integer	Unique identifier for the opportunity
title	String	Title of the opportunity
description	String	Description of the opportunity
category	String	Category of the opportunity
start_date	Date	Start date of the opportunity
end_date	Date	End date of the opportunity
location	String	Location of the opportunity
phone_number	String	Phone number of the opportunity
email	String	Email of the opportunity
image	Image	Image of the opportunity

7. Algorithm & Implementation

The app will be developed using the Jetpack Compose framework for Android, which provides a modern and efficient way of building user interfaces. The app will be divided into various modules, each responsible for a specific feature or set of related features.

The app will also implement various security measures to protect user data and ensure the privacy of its users. These measures include encryption of sensitive data and adherence to best practices for secure coding.

In terms of implementation, the app will be developed using a model-view-view model (MVVM) architecture, which separates the presentation logic from the business logic. This approach will enable easier testing, maintenance, and scalability of the app. Additionally, the app will be developed using version control software, such as Git, to enable collaboration and tracking of changes during the development process.

8. Software requirements traceability matrix

Requirement ID	Requirement Description	Design Element
RE01	The app shall allow users to search for local business.	Search Bar
RE02	The app shall allow users to view business details.	Business detail page
RE03	The app shall provide a dashboard for business owners to manage their listing.	Business dashboard page
RE04	The app shall provide a way for user to report inaccurate information.	Report Page
RE05	The app shall be responsive.	User feedbacks

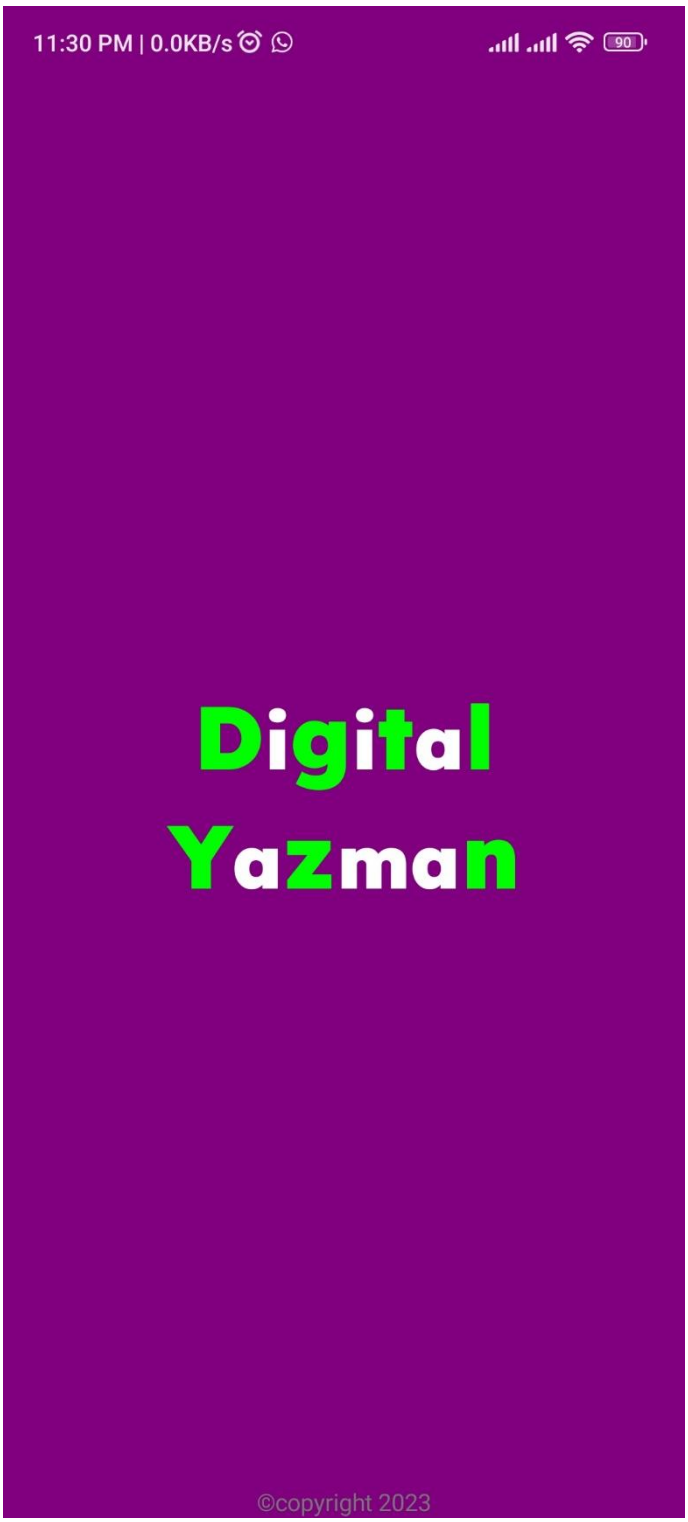
9. Human interface design

The user will see a splash screen at the start of the application. After the splash screen, the user will move to the main menu. In the main menu the user will have many options:

- Businesses & Services
- Emergency Services
- City History
- Local News & Deals
- Opportunities
- Support
- Transport

9.1. Screen Images

Splash Screen

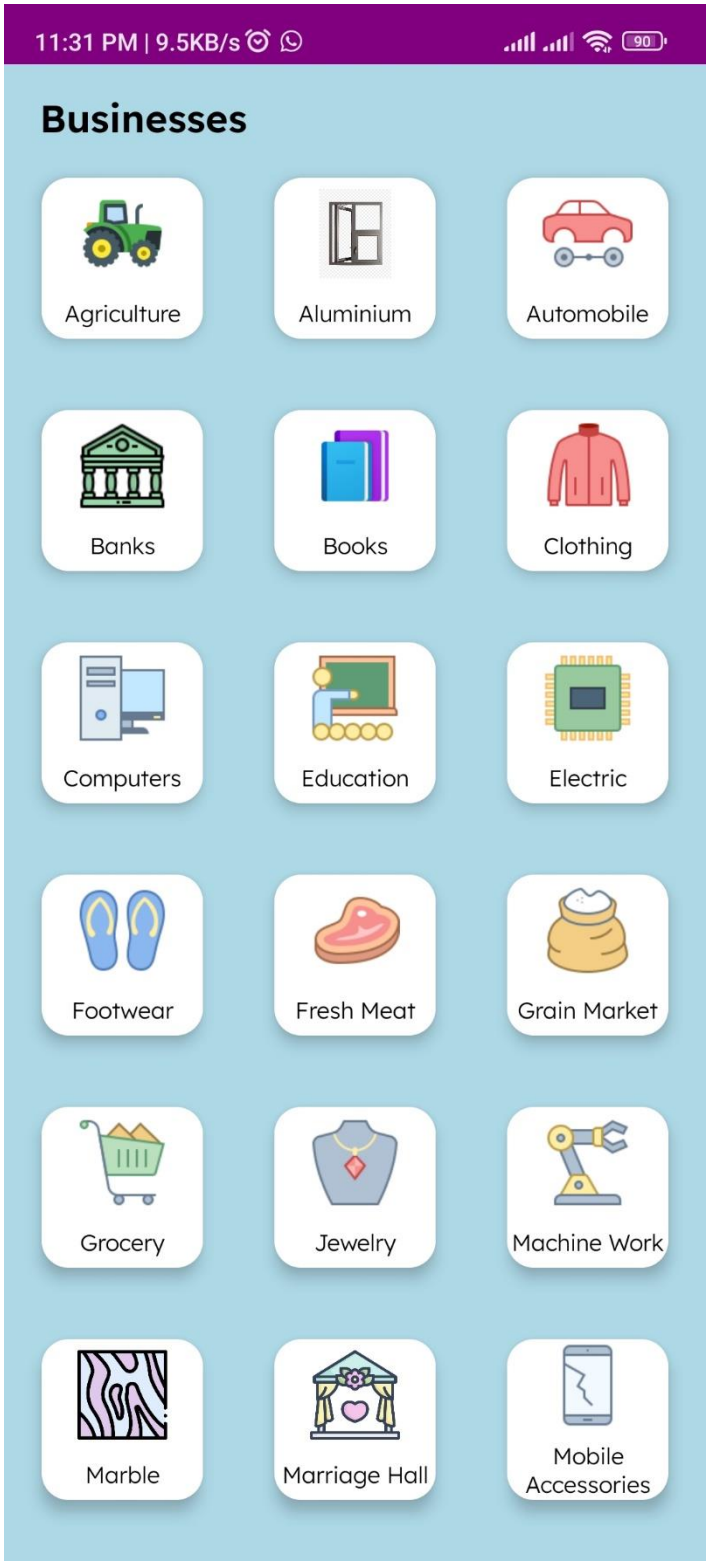


User Interface 1

Main Screen

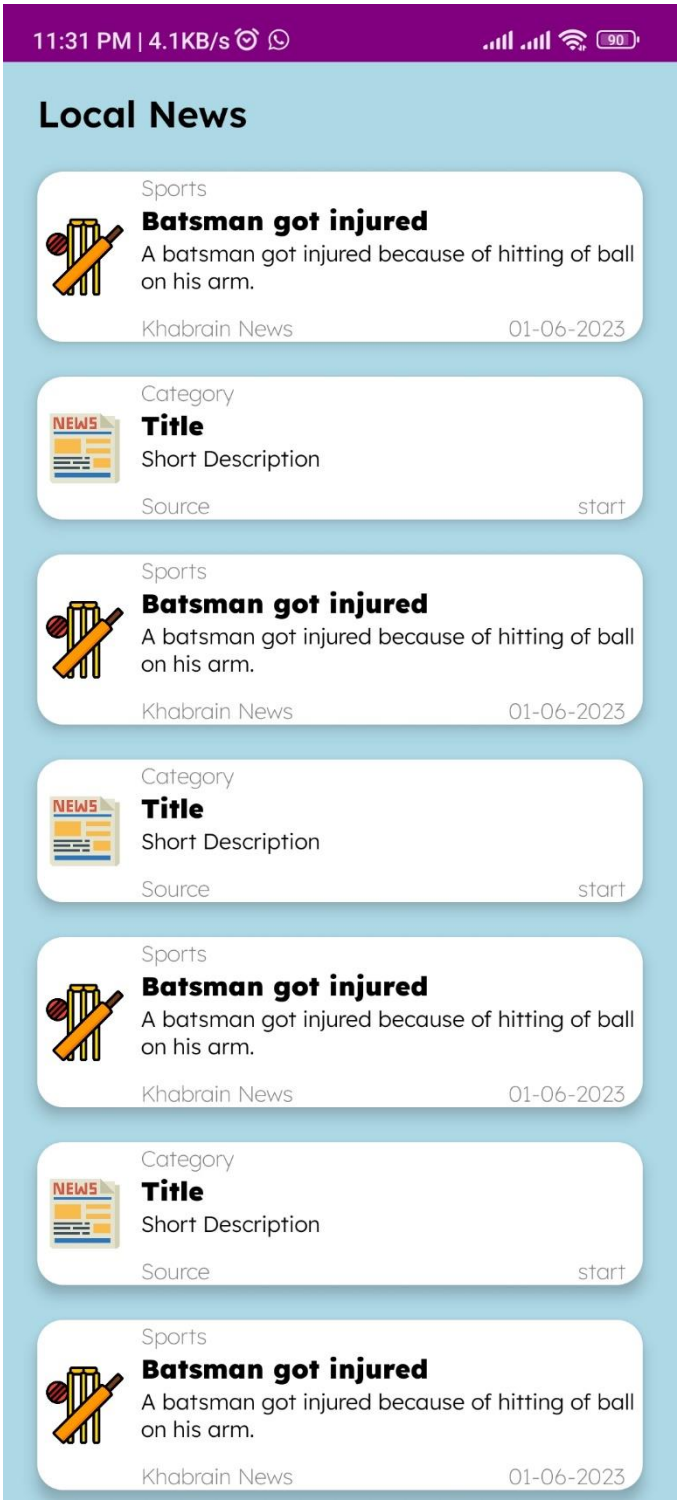
User Interface 2

Business Screen



User Interface 3

Local News Screen



User Interface 4

9.2. Screen objects and actions

Many actions can be performed by the user to achieve the required functionalities of the application. This action can be performed on different screens such as:

- Home screen
- Businesses screen
- Emergency screen
- History screen
- Local News screen
- Opportunities screen
- Service screen
- Support screen
- Transport screen

Few of them are shown in following tables.

Home Screen

Screen Object	Action
Menu with icons representing the various services available in the app businesses, emergency, history, etc.	Tap on a menu icon to go to the corresponding section of the app.
User login/registration button	Tap on the login/registration button to create an account or log in to an existing account.

Business Screen

Screen Object	Action
List of local businesses	Tap on a business item to view its details
Each business item includes a name, address, phone number, and other relevant information	Tap on the call button to call the business directly from the app

Local News Screen

Screen Object	Action
List of local news articles	Tap on an article item to view its full details

Share button for each article item	Tap on the share button to share the article

10. Overall description

10.1. Product perspective

This android application is entirely a new product, which is the solution for the people of Yazman city. The main perspective of this app is, to make the life easier of people in regarding to find something. Simply open app finds your desire item or worker or service and ask provider for it. People of the city can also find opportunities in the city.

10.2. Operating environment

The app is only for android phones. About 99% of android devices can run this app. The phones form API level 21 and android version 5.0 to API level 33 and android version 13 can run this app.

10.3. Design and implementation constraints

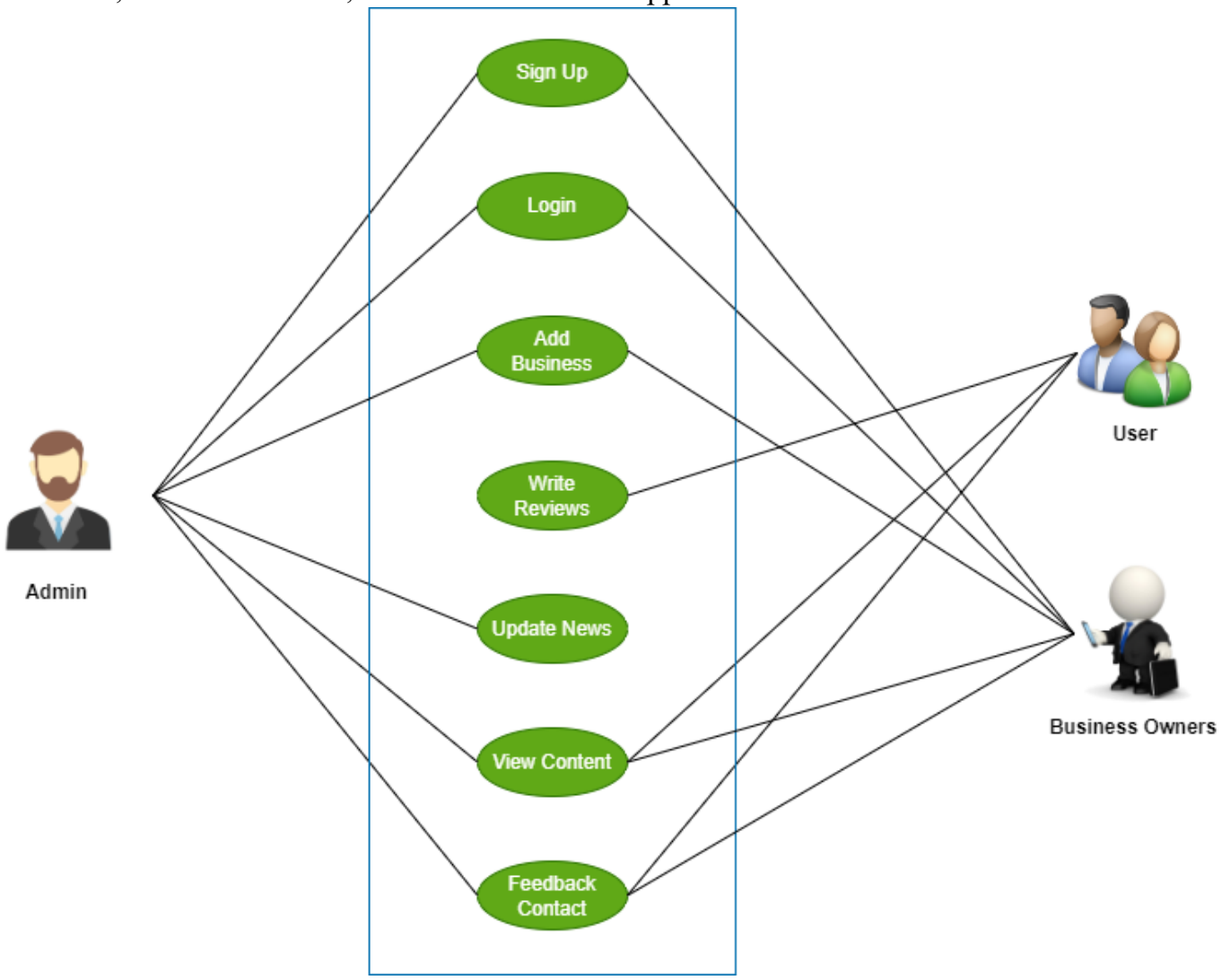
Android Studio software is used to develop this app with official android language of google kotlin.

11. Requirement identifying technique

- **Interviews:** Conducting interviews with city officials, residents, and business owners to identify their requirements for the app.
- **Surveys:** Conducting surveys to gather feedback and requirements from a large number of city residents and businesses.
- **Focus Groups:** Conducting focus groups with representatives from different segments of the city's population, such as residents, business owners, and city officials, to identify their requirements.
- **Workshops:** Holding workshops with representatives from different segments of the city's population to gather and refine requirements.
- **Observations:** Observing city residents and businesses in their natural settings to identify their pain points and requirements.
- **Prototyping:** Creating a working prototype of the app and gathering feedback from stakeholders to refine requirements.

12. Use case diagram

12.1. Admin, Business Owners, User Interaction with app:



Use Case Diagram 1

13. Use case description

The table below indicate a comprehensive use case for Android City App.

Use Case ID:	UC-1
Use Case Name:	City News
Actors:	User
Description:	This use case describes the scenario where a user wants to view the latest news in the city.
Trigger:	User selects the "Local News" option from the app menu.
Preconditions:	User has an active internet connection
Postconditions	User can view the latest news in the city.
Normal Flow:	1. User opens the app and selects the "Local News" option from the menu. 2. App displays the latest news and events in the city.
Alternative Flows: [Alternative Flow 1 – Not in Network]	None
Exceptions:	None
Business Rules	News is updated regularly by the City App Handers.
Assumptions:	None

Use Case ID:	UC-2
Use Case Name:	Local Businesses and Services
Actors:	User, Local Business Owner
Description:	This use case describes the scenario where a user wants to search for local businesses and services in the city.
Trigger:	User selects the "Businesses" option from the app menu.
Preconditions:	User has an active internet connection
Postconditions	User can view a list of local businesses and services based on search criteria.
Normal Flow:	1. User opens the app and selects the "Businesses" option from the menu. 2. App prompts the user to enter search criteria (e.g., business name). 3. User enters search criteria and submits the search. 4. App displays a list of local businesses and services that match the search criteria.
Alternative Flows: [Alternative Flow 1 – Not in Network]	None
Exceptions:	None
Business Rules	Admin can create and manage their business in the app.
Assumptions:	The accuracy of business information is the responsibility of the business owners.

14. Functional Requirements

- **User Authentication:** The app must have a login system that allows users to create and manage their accounts.
- **Local News:** The app should have a local news section that displays the latest news and events happening in the city.
- **City Guide:** The app should provide information about the city, such as tourist attractions, local businesses, and emergency services.
- **City Services:** The app should provide access to city services, such as paying bills, reporting issues, and requesting permits.
- **Social Media Integration:** The app should integrate with social media platforms to allow users to share news and events with their friends and followers.
- **Notifications:** The app should have a notification system that alerts users about important news, events, and updates.
- **Transportation:** The app could provide information on public transportation options in the city, including bus and train schedules and fare information.

Identifier	FR1
Title	City news
Requirement	The app shall provide a news section that displays the latest updates and happenings in the city.
Source	User requirement
Rationale	Users want to stay informed about important news in the city.
Business Rule (if required)	News must be updated regularly.
Dependencies	None
Priority	High

Identifier	FR2
Title	Report issues
Requirement	The app shall allow users to report issues such as unrelated news, bugs to the admin.
Source	User requirement
Rationale	Users want an easy way to report issues and have them addressed by admin.
Business Rule (if required)	None
Dependencies	Integration with admin
Priority	High

Identifier	FR3
Title	Local Business
Requirement	The app shall provide a directory of local businesses in the city with information such as contact details, operation hours, and reviews.
Source	User requirement
Rationale	User want to be able to find and support local businesses in their city.
Business Rule (if required)	Business owners can manage their business profiles in the app. The accuracy of business information is the responsibility of the business owners.
Dependencies	None
Priority	Medium

15. Non-Functional Requirements

15.1. Performance

- Response time for search functionality should be less than 3 seconds.
- App should be able to handle at least 1000 simultaneous users.

15.2. Security

- User authentication should be implemented to prevent unauthorized access.
- User data should be encrypted during transmission and storage.
- App should comply with industry-standard security practices and guidelines.

15.3. Usability

- App should have a simple and intuitive user interface.
- App should be compatible with different screen sizes and resolutions.
- Text and icons should be easily readable and distinguishable.

7.4. Compatibility

- App should be compatible with the latest Android operating system version.
- App should be compatible with a variety of devices from different manufacturers.
- App should be compatible with popular third-party libraries and frameworks.

15.5. Reliability

- App should be able to handle errors and exceptions gracefully.
- App should have a backup and recovery plan in case of system failure or data loss.
- App should have a minimum uptime of 99.9%.

15.6. Supportability

- App should be maintained, updated, and supported over its lifetime.

16. Appendix I

1. https://www.uplabs.com/ui-kits/android
2. https://chat.openai.com/
3. https://www.uplabs.com/ui-kits/android
4. https://developer.android.com/jetpack/compose
5. http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm