

Index

Sr.no	Title	Page no.
1	Introduction	
	1.1. Introduction of the project Weather App	
	1.2. Scope of work	
	1.3.Operating environment-Hardware and software	
	1.4.Module Description	
	1.5.Detail Description of technology used	
2	Proposed System	
	2.1. Proposed System	
	2.2. Objectives of System	
	2.3. User Requirement	
3	Analysis and Design	
	3.1. Class Diagram	
	3.2. Use Case Diagram	
	3.3. Activity Diagram	
	3.4 Sequence Diagram	
	3.5. Data Dictionary	
	3.6 Table Sequence	
	3.7.Test procedure and implementation	
4	User Manual	
5	Drawback and Limitations	
6	Proposed Enhancements	
7	Test Cases	
8	User Interface Screens	
9	Conclusion	
10	Bibliography	

1. INTRODUCTION

1.1 Introduction to the project Weather App

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere at a given place and using meteorology to project how the atmosphere will change. The role of Technology has been remarkable in the field of weather forecasting. Weather data is not only necessary for researchers or scientists, ordinary people can be benefitted from it as well. People nowadays are feeling the necessity of weather data as well. There are a variety of weather mobile apps in Google Play and the App store.

- 1. Real-Time Weather Data:** WeatherNow integrates with a powerful weather API to fetch real-time weather information for any location around the world. Get access to current weather conditions, forecasts, temperature, humidity, wind speed, and more.
- 2. Registration and Login with Firebase:** We prioritize the security of your data. WeatherNow employs Firebase for user authentication and management. You can register a new account, log in securely.

1.2 Scope Of Work

1. Basic Features:

- Current weather conditions (temperature, humidity, wind speed).
- 3 to 7-day forecasts.
- Sunrise and sunset times.

2. Location Services:

- Geolocation for current weather.
- Save multiple locations.

3. User Personalization:

- User accounts for preferences.
- Customizable notifications.

4. Additional Features:

- Weather maps (radar, satellite).
- Air Quality Index (AQI) and pollen count.
- Widgets for home screens.

5. Social Integration:

- Share weather updates.
- Community features.

6. Information :

- Weather-related news and articles.

1.3 Operating Environment – Hardware and Software

▪ **Hardware Requirements :**

- Processor → Intel Core i3
- Hard Disk → 512GB
- RAM → 4GB

▪ **Software Requirements :**

- Operating System → Windows 7 or later, macOS, Linux
- Frontend → Android Studio(XML,JAVA)
- Backend → Weather API and Firebase
- Tools/ IDE → Android Studio

1.4 Module Description

1. **Registration** ➔ Email, Password
2. **Login** ➔ Email, Password
3. **Weather API** ➔
 - **Location** ➔ Name , Latitude, Longitude
 - **Current** ➔ Temperature
 - Condition** ➔ Text, Icon
 - **Forecast** ➔
 - Forecastday** ➔ Date
 - **Hour** ➔ Time, Temperature , is_day
 - Condition** ➔ Text, Icon, Wind

1.5 Detail Description of technology Used

Frontend :

1. **Java Language:** Android app development primarily relies on Java, a versatile, platform-independent programming language known for its portability and ease of use.
2. **Android Studio:** Developers use Android Studio as the official integrated development environment (IDE) for creating Android apps. It offers a suite of tools, including code editors and layout designers, to simplify the development process.
3. **XML Layouts and Views:** The app's user interface is defined using XML layout files, which include various views (e.g., buttons, text fields, images). XML files are linked to Java code to manage user interactions.

Backend :

API (Application Programming Interface): An API, or Application Programming Interface, is a set of rules and protocols that allows different software applications to communicate and interact with each other. APIs are crucial for enabling the integration of different systems, services, and applications. They serve as intermediaries that allow developers to harness the functionality of other software without needing to understand the intricacies of how that software works internally.

Firebase: Firebase is a comprehensive mobile and web application development platform provided by Google. It offers a set of tools and services that help developers build high-quality apps more efficiently.

Firebase simplifies many aspects of app development by offering a scalable, fully managed infrastructure for backend services. Developers can focus on building features and user experiences while Firebase handles the underlying infrastructure and server-side logic. It's a popular choice for startups and developers looking to accelerate app development without sacrificing quality and security.

2. PROPOSED SYSTEM

2.1 Proposed System

1. Frontend Development (Android Studio with XML and Java):

User Interface (UI) Design:

- **XML Layouts:** Design various screens using XML layout files, defining the arrangement and appearance of UI elements.
- **Widgets and Views:** Implement interactive elements such as buttons, text views, and image views.
- **Resource Management:** Efficiently manage resources like images and strings.

Networking and API Integration:

- **HTTP Requests:** Use Java libraries or Android's built-in libraries for making HTTP requests to fetch weather data from the API.
- **AsyncTask/Thread Management:** Implement asynchronous tasks or threads to avoid blocking the UI thread during network calls.

2. Backend Development (Firebase):

Firebase Setup:

- **Firebase Console:** Set up a Firebase project through the Firebase console.
- **Authentication:** Implement Firebase Authentication for user registration and login.

Realtime Database:

- **Database Structure:** Design the Firebase Realtime Database structure to store user-specific data and preferences.
- **Integration:** Integrate Firebase SDK in the Android app for seamless data communication.

Integration and Deployment:

Weather API Integration:

- **API Key Handling:** Safely handle API keys using secure methods.
- **Retrofit or HTTP Library:** Use Retrofit or another HTTP library to interact with the weather API.

Firebase and Weather API Integration:

- **Data Synchronization:** Sync data between the Firebase Realtime Database and weather API responses.
- **Asynchronous Operations:** Manage asynchronous operations effectively to avoid blocking the main thread.

2.2 Objectives Of The System

1. **Accurate Weather Information:** Provide up-to-date and accurate weather data to users by integrating with a reliable weather data source or API.
2. **Weather Forecast:** Offer current weather conditions and forecasts for the next few days, including temperature, precipitation, wind speed, and humidity.
3. **Multiple Locations:** Allow users to switch between multiple locations.
4. **Performance Optimization:** Ensure the app runs smoothly and responds quickly, even on older or less powerful devices.

2.3 User Requirement

1. Current Weather:

- Display real-time temperature, humidity, wind speed, and conditions.

2. Weather Forecast:

- Provide a 3 to 7-day forecast with daily and hourly details.

3. Location-Based Services:

- manually add multiple locations.

4. User Accounts and Preferences:

- Allow account creation with preferences like default location and units.

5. Air Quality:

- Display Air Quality.

6. Widgets:

- Support home screen widgets with customization options.

7. Performance and Security:

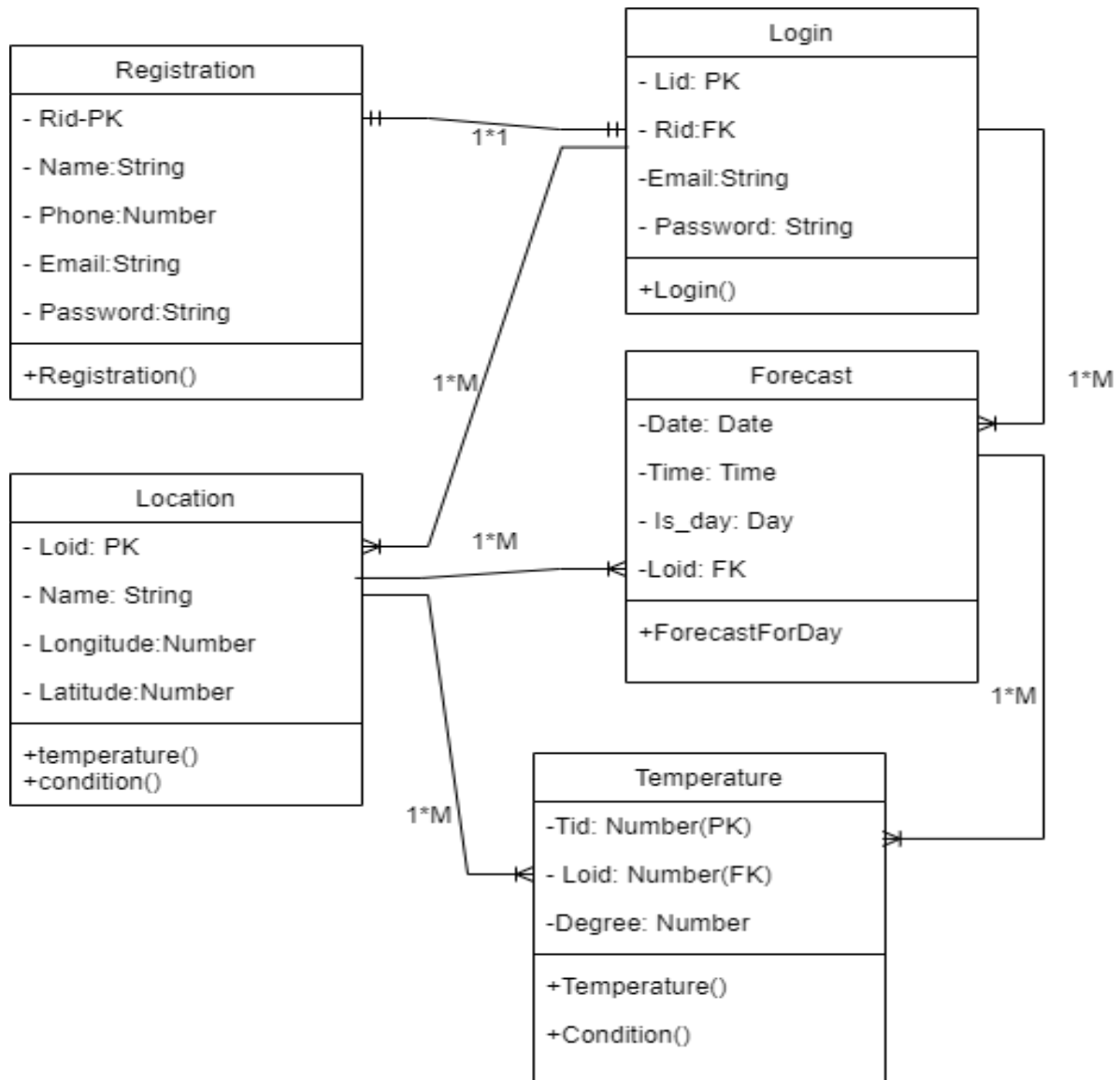
- Ensure fast, smooth performance and secure user data storage.

8. Scalability and Compatibility:

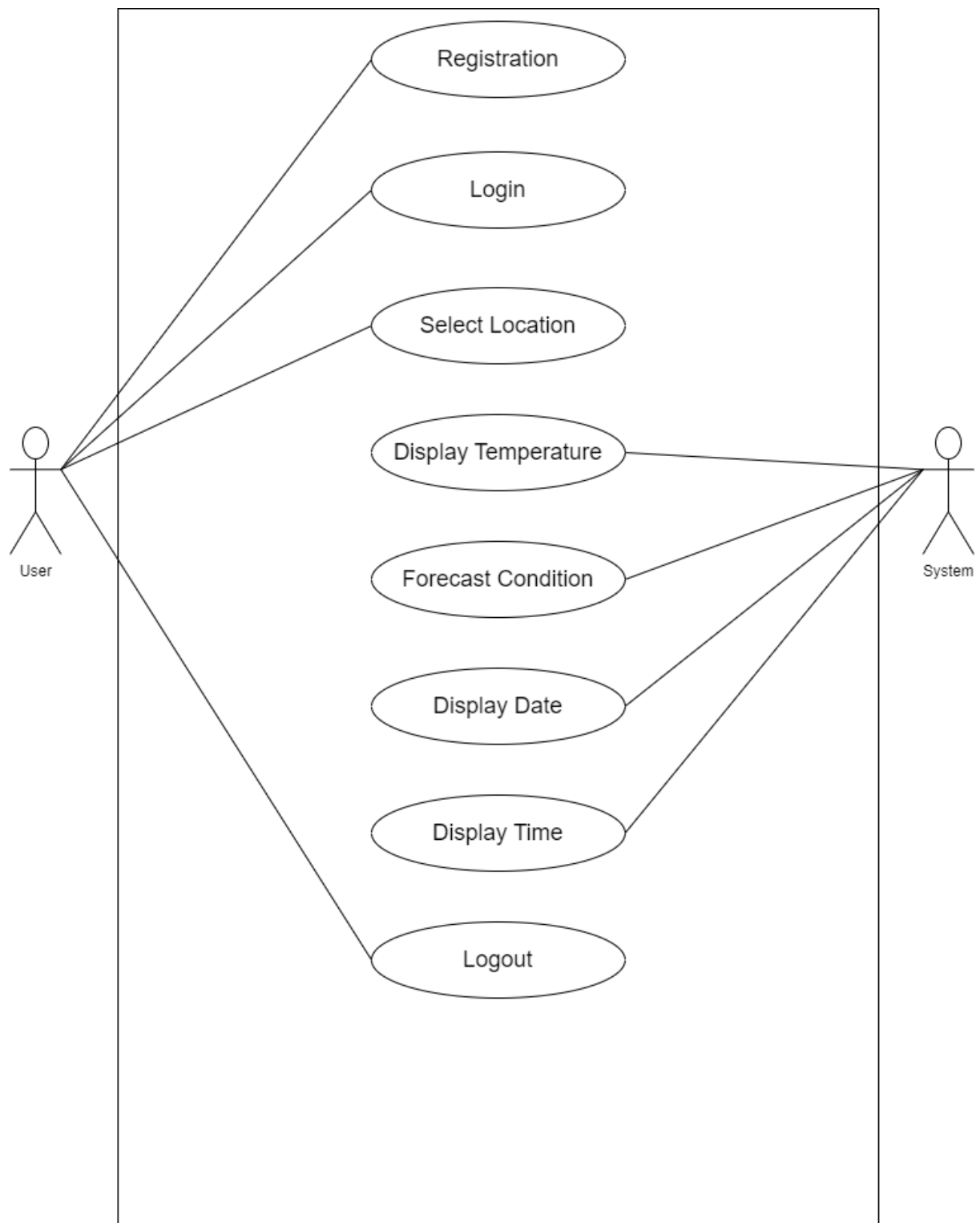
- Handle a scalable number of users and devices.
- Compatibility with various Android versions and devices.

3. ANALYSIS AND DESIGN

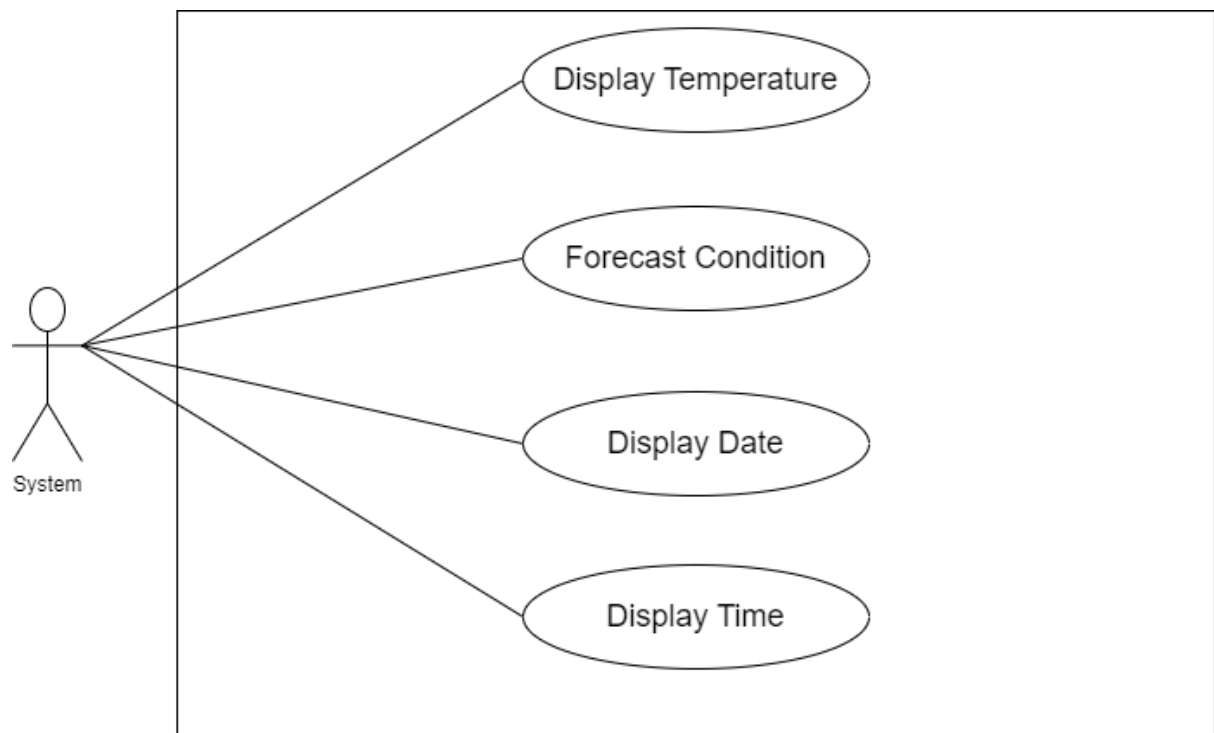
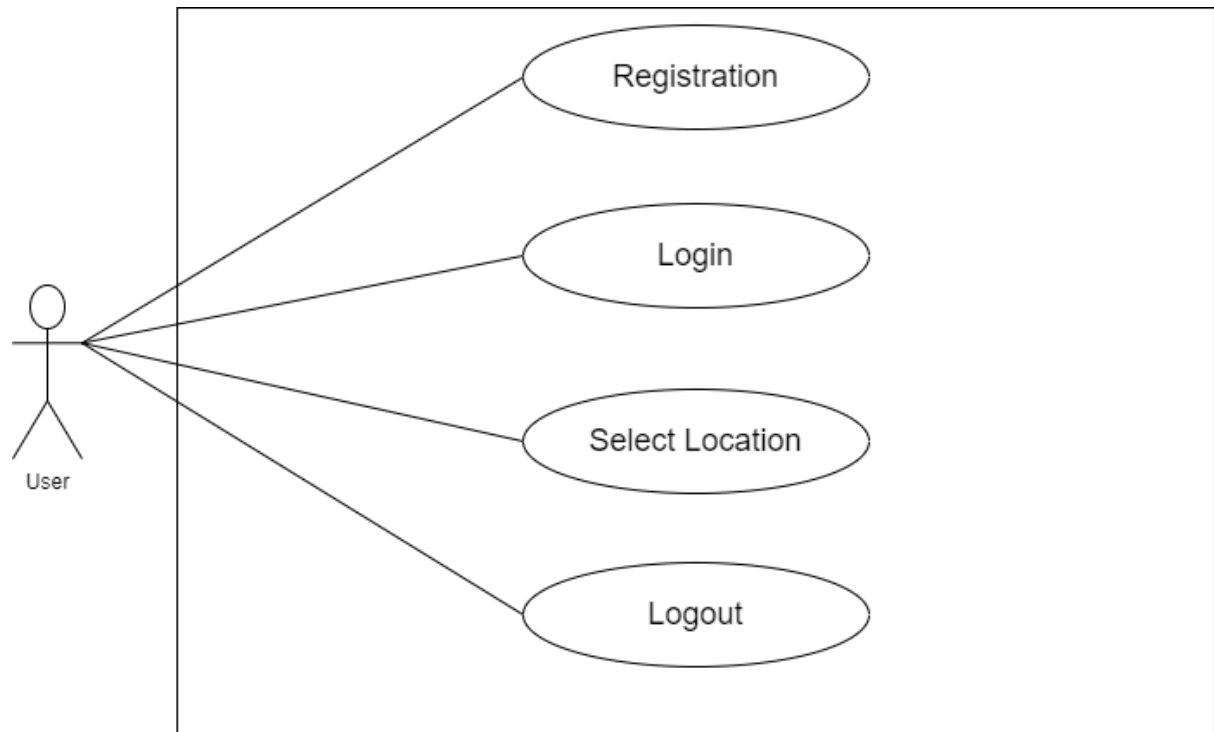
3.1 Class Diagram



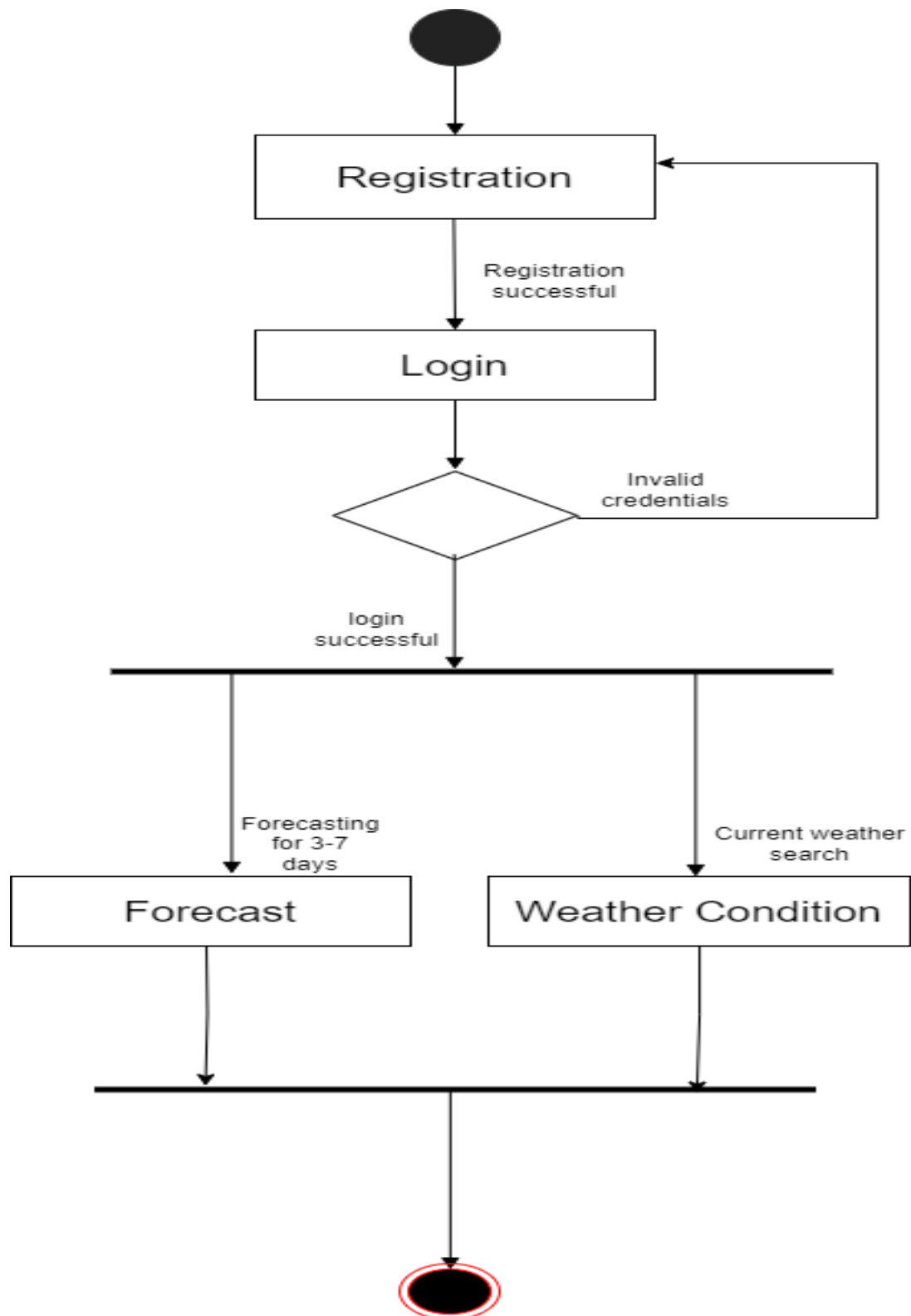
3.2 Global Use Case Diagram



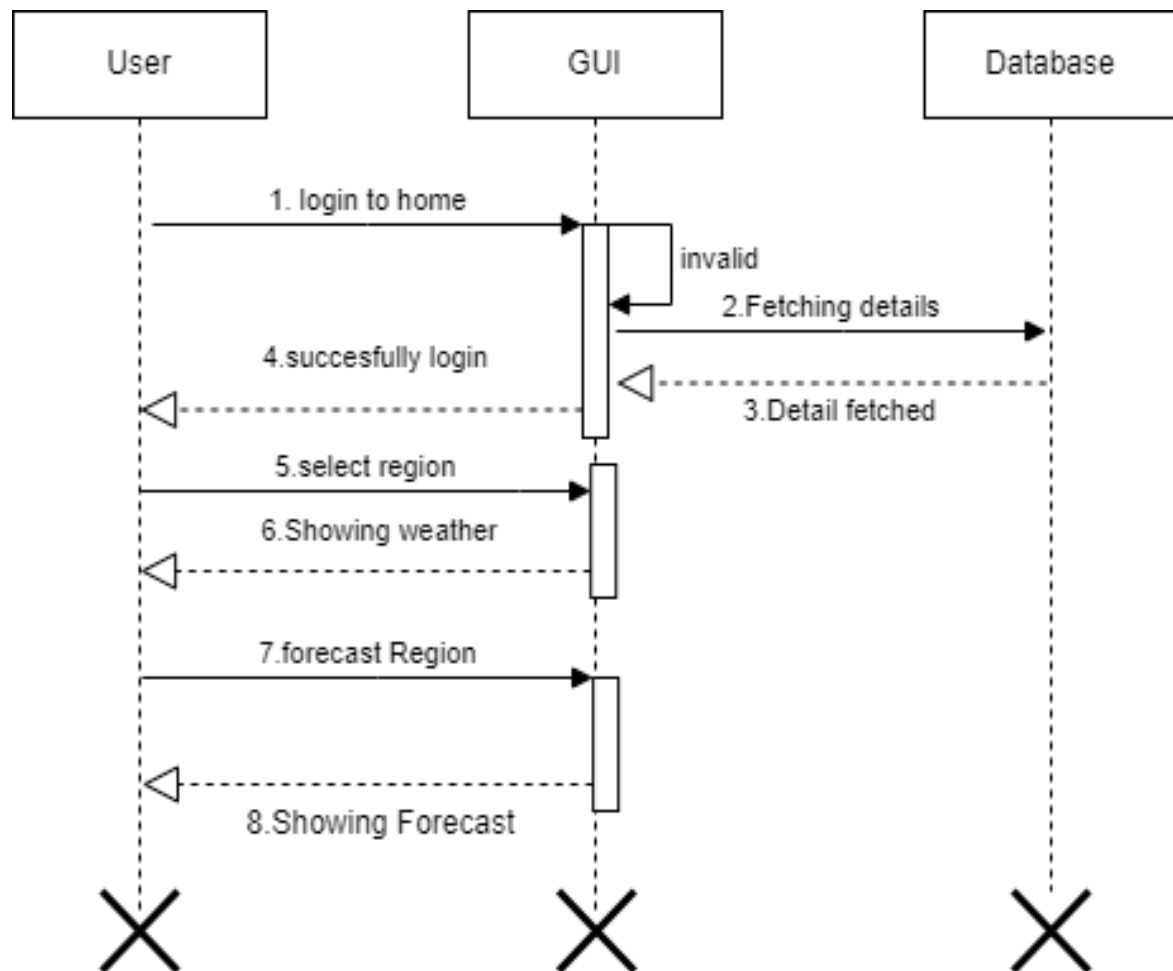
3.2 Individual Use Case Diagram



3.3Activity Diagram



3.4 Sequence Diagram



3.5 Data Dictionary

No	Data Element	Description	Data Type	Size	List of Specific	Data Store	Remarks
1	Email	User Email	Text	50	It store user email	Registration Muster	It contains user email
2	Password	User Password	Text	20	It store user pwd.	Registration Muster	It contains user pwd
3	Login	User Email	Text	50	It store user email	Login Muster	It contain user login
4	Password	User Pwd	Text	20	It store user pwd	Login Muster	It contains user pwd
5	Location	Location name	Text	25	It store location	Location Muster	It contains location
6	Latitude	Latitude	Number	10	It store latitude	Location Muster	It contains location latitude
7	Longitude	Longitude	Number	10	It store Longitude	Location Muster	It contains location longitude
8	Temperature	Temperature	Number	50	It store Current temperature	Location Muster	It contains current temperature
9	Text	Condition text	Text	20	It store text	Location Muster	It contain condition of location
10	Icon	Condition icon	Image	-	It store condition icon	Location Muster	It contains icon
11	Date	Date	Number	50	It store date	Forecast Muster	It contains date
12	Time	Time	Number	50	It store time	Forecast Muster	It contains time
13	Temperature	Temperature	Number	50	It store temperature	Forecast Muster	It contains temperature

14	Is day	day	Date	20	It store days	Forecast Muster	It contains week days
15	Text	Forecast text	text	50	It store text	Forecast Muster	It contains forecast
16	Icon	Forecast icon	image	-	It store image	Forecast Muster	It contains forecast icon
17	wind	Speed of the wind	Number	50	It store wind speed	Forecast Muster	It contain speed of wind

3.6 Sequence Table

User Table

Sr.no	Field name	Field type	Size	Constant	Description
1.	Uid	Number	20	Not Null-PK	User id
2.	Name	Text	50	Not Null	User Name
3.	Phone	Number	50	Not Null	User Phone
4.	Email	Text	20	Not Null	User Email
5.	Password	Text	20	Not Null	User Pwd

Location Table

Sr.no	Field name	Field type	Size	Constant	Description
1.	Loid	Number	20	Not Null-PK	Location id
2.	Name	Text	50	Not Null	Location name
3.	Longitude	Number	10	Not Null	Longitude
4.	Lattitude	Number	10	Not Null	Lattitude

Forecast Table

Sr.no	Field name	Field type	Size	Constant	Description
1.	Date	Date	-	Not Null	Forecast Date
2.	Time	Time	-	Not Null	Forecast Time
3.	Is_day	Day	-	Not Null	Forecast Day
4.	Loid	Number	20	Not Null-FK	Location id

Temperature Table

Sr.no	Field name	Field type	Size	Constant	Description
1.	Tid	Number	20	Not Null-PK	Temperature id
2.	Loid	Number	20	Not Null-FK	Location id
3.	Degree	Number	10	Not Null	Temperature degree
4.	Wind_speed	Text	10	Not Null	Wind speed

3.7 Test Procedure And Implementation

Unit Testing:

- **Objective:** Verify the correctness of individual units or components.
- **Implementation:**
 - Test each function or method independently.
 - Use testing frameworks like JUnit for Java.

Integration Testing:

- **Objective:** Validate the interactions between different components.
- **Implementation:**
 - Test the integration of frontend components (XML layouts and Java code).
 - Ensure proper communication with backend services (APIs and Firebase).

User Interface (UI) Testing:

- **Objective:** Confirm that the user interface is intuitive and visually appealing.
- **Implementation:**
 - Check UI elements for proper alignment and responsiveness.
 - Verify that widgets and maps display correctly.

Performance Testing:

- **Objective:** Evaluate the app's responsiveness, speed, and resource usage.
- **Implementation :**
 - Measure app launch time and response to user interactions.
 - Test under different network conditions for data retrieval performance.

4 USER MANUAL

1.Introduction

About:

- Welcome to the Weather Application for accurate and timely weather updates.

Key Highlights:

- Real-time updates
- 3 to 7-day forecasts
- Personalized user preferences

2. Getting Started

Installation:

- Download from Google Play Store.
- Create an account for personalized features.
- Enable location services for accurate forecasts.

3. UI Tour

Home Screen:

- Current weather and forecasts.

4. Privacy and Security

Privacy Policy:

- Read our policy on data handling.

Account Security:

- Use a strong password and keep it secure.

Location Data:

- Used solely for accurate weather information.

5. Contact and Support

Customer Support:

- Contact support@weatherapp.com for assistance.

Feedback:

- Share your thoughts through in-app feedback.

5. DRAWBACK AND LIMITATION

1. Data Accuracy:

- **Drawback:** Weather predictions are subject to change, and occasional inaccuracies may occur.
- **Limitation:** The app relies on external weather APIs, and real-world conditions may vary.

2. Limited Location Coverage:

- **Drawback:** Some remote or less-populated areas might have limited or less accurate weather data.
- **Limitation:** The app's accuracy may vary depending on the availability of weather station data in certain regions.

3. Dependency on Internet Connectivity:

- **Drawback:** Real-time weather updates and certain features require an active internet connection.
- **Limitation:** Users may experience limitations in offline mode, and some features may be restricted.

4. Device Compatibility:

- **Drawback:** Older devices or those with limited resources might experience performance issues.
- **Limitation:** High-end features such as complex maps may not work optimally on low-spec devices.

5. Battery Consumption:

- **Drawback:** Continuous use of location services and data updates may contribute to increased battery consumption.
- **Limitation:** Optimizing for low battery usage while providing real-time updates can be challenging.

6. PROPOSED ENHANCEMENTS

1. Android Wear Integration:

Proposal: Develop a companion app for Android Wear devices.

Enhancement: Users can receive weather updates directly on their smartwatches, with features like quick glances at current conditions and upcoming forecasts.

2. Offline Maps and Forecast:

Proposal: Improve offline functionality.

Enhancement: Allow users to download maps and weather forecasts for offline use, useful in areas with limited connectivity.

3. Location-Based Widgets:

Proposal: Introduce location-based widgets.

Enhancement: Widgets automatically update based on the user's current location, providing relevant weather information at a glance.

4. Dark Mode Scheduler:

Proposal: Introduce a dark mode scheduler.

Enhancement: Users can set specific times for the app to switch between light and dark modes, reducing eye strain during different parts of the day.

5. Improved Battery Optimization:

Proposal: Optimize battery usage.

Enhancement: Implement efficient background processes and location updates to minimize the impact on device battery life.

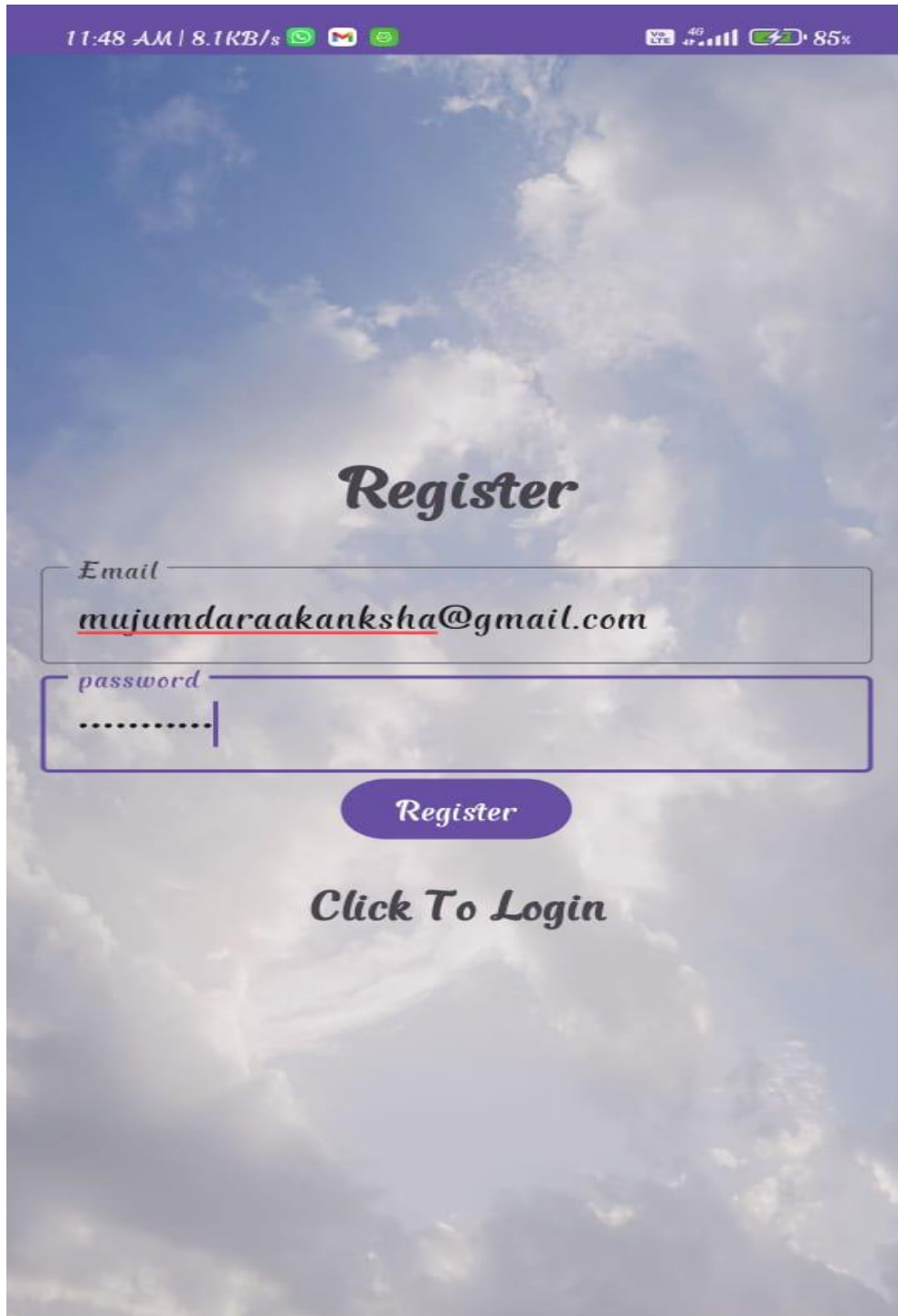
6. TEST CASES

Test Case Id	Test Case Description	Test Steps	Expected Result
TC001	Verify the presence of registration screen	Open the app and navigate to the registration screen.	The registration screen is displayed with input fields for username/email, password, and other necessary details.
TC002	Verify successful registration	Enter valid registration details. Click the registration/submit button.	The app should display a success message and, if applicable, navigate to the login screen or home/dashboard.
TC003	Verify error messages with invalid registration data	Enter incomplete or invalid registration details. Click the registration/submit button.	Error messages should be displayed next to the relevant input fields, indicating the issues with the provided information.
TC004	Verify the presence of login screen	Open the app and navigate to the login screen.	The login screen is displayed with input fields for username/email and password.
TC005	Verify successful login with valid credentials	Enter valid username/email and password. Click the login button.	The app should navigate to the home/dashboard screen, indicating a successful login.
TC006	Verify error message with invalid credentials	Enter an invalid username/email or password. Click the login button.	An error message should be displayed, indicating that the login failed due to incorrect credentials.
TC007	Verify app launch and home screen	Open the weather app. Observe the home screen displaying current weather information for the user's location or a default location.	The app should launch successfully, and the home screen should show current weather details, such as temperature, condition, and location.
TC007	Verify location change functionality	Change the app's location manually or	The app should update the weather information to

		allow it to detect the user's location.	reflect the chosen or detected location, including temperature, condition, and forecasts.
TC008	Verify weather details accuracy	Compare the app's displayed temperature, condition, and other details with a reliable external source (e.g., a reputable weather website).	The app's weather information should be accurate and align with the data from the external source, within an acceptable margin of error
TC009	Verify hourly and daily forecast accuracy	Navigate to the hourly and daily forecast sections. Compare the forecasted details with a reliable external source.	The app's forecasts should be accurate and align with the data from the external source, within an acceptable margin of error.
TC0010	Verify wind speed and direction accuracy	Observe the app's displayed wind speed and direction. Compare it with a reliable external source (e.g., a weather website or another app).	The app's wind speed and direction information should align with the data from the external source, within an acceptable margin of error.
TC0011	Verify time zone adjustments for different locations	Change the app's location to a different time zone. Observe the accuracy of time-related information (e.g., sunrise, sunset, hourly forecasts).	The app should adjust time-related information correctly based on the chosen or detected time zone, ensuring accurate timing details.
TC0012	Verify battery usage and resource efficiency	Monitor the app's impact on device battery and resources (CPU, memory) during regular usage and background updates.	The app should have minimal impact on battery life and system resources, providing an efficient and user-friendly experience.

TC0013	Verify location permission handling	Disable location permissions for the app. Open the app and attempt to view weather information	The app should request permission to access the device's location. If denied, it should provide a user-friendly message and offer a way to enable location permissions.
TC0014	Verify app responsiveness during network changes	Change the network connectivity status (e.g., switch between Wi-Fi and mobile data) while using the app.	The app should adapt to changes in network connectivity seamlessly, maintaining functionality and updating data accordingly.
TC0015	Verify app update functionality	Check for the availability of app updates. If an update is available, follow the update process (e.g., through the app store).	The app should successfully update to the latest version without data loss or disruptions in functionality.

8. USER INTERFACE SCREENS



11:48 AM | 8.1KB/s

4G 85%

Register

Email

mujumdaraakanksha@gmail.com

password

.....

Register

Click To Login

Login

Email

mujumdaraakanksha@gmail.com

password

.....

Login

[Click To Register](#)



Mumbai

Enter your city

Mumbai



28.0°C



Mist

Today's Weather Forecast

12:00 AM

24.2°C



25.6 Km/h

01:00 AM

25.6°C



24.8 Km/h

02:00 AM

26.9°C



24.5 Km/h

03:00 AM

26.9°C



23.5 Km/h

9. CONCLUSION

In conclusion, the proposed enhancements for the weather application on Android strive to elevate the user experience, leveraging the capabilities of the Android platform and incorporating modern design principles. These enhancements are designed to make the app more feature-rich, visually appealing, and user-friendly. By addressing key areas such as offline functionality, integration with wearable devices, and improved accessibility, the goal is to meet the diverse needs of Android users and provide a comprehensive weather experience.

The implementation of these enhancements would require close collaboration with Android developers, user interface designers, and weather experts. Regular user feedback, iterative development cycles, and thorough testing would be essential to ensure the successful integration of these features. Additionally, staying abreast of the latest Android technologies and design guidelines will contribute to keeping the app current and competitive in the ever-evolving mobile landscape.

Ultimately, the conclusion emphasizes the commitment to continuous improvement, user satisfaction, and providing a robust and innovative weather application tailored to the Android ecosystem.

10.BIBLIOGRAPHY

- <https://www.weatherapi.com/>

Weather APIs (Application Programming Interfaces) are services that provide developers with access to weather data, allowing them to integrate real-time or forecasted weather information into their applications, websites, or services.

- <https://www.postman.com/>

Postman is a popular collaboration platform for API development. Postman provides tools for designing, testing, and managing APIs .We can also fire some HTTP request like GET,PUT,SET Request fetch the data from various sources.

- <https://developer.android.com/docs>

Official documentation website for Android developers. The information provided on this website is essential for developers working on Android app development.

- <https://www.w3schools>

"W3Schools" is a popular online learning platform that provides tutorials and references on web development technologies. The website offers a wide range of content, including HTML, CSS, JavaScript, XML, SQL, Python, PHP, android and more.