SmartBike – Cycling Companion App

Name:Zheheng Li

I. Introduction

SmartBike is a comprehensive cycling companion application designed to enhance the cycling experience for both casual and serious cyclists. The app provides real-time tracking, performance metrics, and safety features to help users monitor and improve their cycling activities. Built using modern Android development practices and technologies, SmartBike aims to be a reliable and user-friendly cycling companion.

II. Problem Statement

Cyclists often face several challenges:

1. Difficulty in tracking and analyzing their rides
2. Lack of real-time performance metrics
3. Safety concerns during rides
4. Need for integration with external sensors
5. Desire for historical data analysis

SmartBike addresses these challenges by providing:

- Real-time GPS tracking and route recording
- Comprehensive performance metrics
- Crash detection and emergency features
- Bluetooth LE integration for external sensors
- Detailed ride history and statistics

III. Design and Implementation

Architecture

The app follows the MVVM (Model-View-ViewModel) architecture pattern with the following components:

1. **Data Layer**
- Room Database for persistent storage
- Repository pattern for data access
- Type converters for complex data types
2. **Business Logic Layer**
- ViewModel for UI state management
- Foreground Service for ride tracking
- Use Cases for business logic
3. **Presentation Layer**
- MVVM pattern with data binding
- Navigation Component for navigation
- Material Design 3 for modern UI

Key Features Implementation

1. **Real-time Ride Tracking**

- GPS location tracking
- Route visualization
- Performance metrics calculation

2. **Bluetooth Integration**
- Heart rate monitor support
- Cycling sensor connectivity
- Device management

3. **Crash Detection**
- Accelerometer-based monitoring
- Impact force calculation
- Emergency contact integration

4. **Data Management**
- Local database storage
- Ride history management
- Statistics calculation

External API Integration

1. **OpenStreetMap API**
- Map visualization
- Route tracking
- Location services

2. **OpenStreetMap Services**
- Location updates
- Activity recognition
- Geofencing

3. **Android Bluetooth API**
- Device scanning
- Connection management
- Data transfer

IV. Minimum UI Requirements

The app meets all minimum UI requirements through:

1. **Clear and Intuitive Layout**
- Bottom navigation for main sections
- Material Design components
- Consistent spacing and typography

2. **Visually Appealing Design**
- Material Design 3 components
- Custom color scheme
- Responsive layouts

3. **Informative Feedback**

- Real-time metrics display
- Toast messages for actions
- Loading indicators

4. **Responsive Design**

- Adaptive layouts
- Support for different screen sizes
- Orientation changes handling

V. Additional Features

1. **Advanced Safety Features**

- Crash detection
- Emergency contact integration
- High-impact alerts

2. **Enhanced Statistics**

- Detailed ride analysis
- Performance trends
- Achievement tracking

3. **Device Integration**

- Heart rate monitoring
- Cycling sensor support
- Custom device profiles

VI. Testing and Evaluation

Testing Process

1. **Unit Testing**

- ViewModel logic
- Repository operations
- Utility functions

2. **Integration Testing**

- Database operations
- Service interactions
- API integrations

3. **UI Testing**

- Navigation flows
- User interactions
- State management

VII. Conclusion

SmartBike successfully implements a comprehensive cycling companion app that addresses the needs of modern cyclists. The project demonstrates effective use of Android development practices and integration of various APIs and services. The modular architecture ensures maintainability and scalability for future enhancements.

VIII. Figma

https://www.figma.com/proto/WFjqmF4uoAtCrQpvYPtsIu/Final-Project?node-id=1-1215&t=HlcyDSpwynvzEmvY-1

IX. Demo Video

It's already in the zip.

X. References

1. **Documentation**
- Android Developer Documentation
- Kotlin Documentation
- Material Design Guidelines
- Google Maps API Documentation
2. **Libraries and Tools**
- Android Jetpack
- Room Database
- Google Play Services
- Material Components for Android