SmartGlass Detailed Technical Documentation

1. Introduction

SmartGlass is a real-time sports analytics tool designed for use with Brilliant Labs Frame smart glasses. The application connects an Elo M51 Android tablet to the Frame glasses via Bluetooth 5.3 and displays live basketball statistics pulled from the Synergy Sports API. The goal of this capstone project is to provide coaches and analysts with heads-up, hands-free access to actionable game insights.

2. System Overview

SmartGlass consists of three main components:

- Python Backend: Responsible for interfacing with Synergy Sports API and outputting relevant game stats.
- Android Kotlin Application: Runs on the Elo tablet and serves as both the user interface and BLE communication bridge.
- Brilliant Labs Frame Glasses: Display real-time data sent via Lua scripts.

Technologies used:

- Kotlin, Python 3.10
- Bluetooth Low Energy (BLE)
- Android SDK (API 33+)
- Gradle Build System

3. Installation and Deployment

Prerequisites:

- Android Studio

- Python 3.10+

- Git

- USB Debugging enabled on Elo tablet

Steps:

1. Clone Repo:

git clone https://github.com/SmartGlass-Analytics/smartglass

2. Android App:

- Open smartglass/app in Android Studio

- Build & deploy to Elo M51 tablet

3. Python Backend:

- Navigate to repo root

- Run apipractice.py to fetch stats

4. BLE Connection:

- Ensure Frame glasses are powered on

- Tap "Connect" on the app to pair via Bluetooth

4. Feature Descriptions

- Stat Display: Stats such as FG%, 3PT%, and player highlights are displayed.

- Real-Time Sync: Stats fetched every few seconds and pushed to the glasses.

- Modular UI: Android UI allows user interaction and manual refresh.

5. Architecture and Data Flow

[Synergy API] --(Python)--> [Tablet File/API] --(BLE GATT + Lua)--> [Frame Glasses]

6. Modifying and Extending the System

- Kotlin files: MainActivity.kt, layout XMLs

- Python files: apipractice.py

- New stats can be added by:

- Updating Python parsing logic

- Creating new UI buttons in Android

- Formatting new Lua strings to send

7. Integration with External Resources

- Synergy API: Requires credentials in SynergyAuth.txt

- Brilliant Labs Frame: BLE-based interaction, use nRF Connect for UUID discovery

8. Testing and Validation

- BLE connection tested with Frame emulator and nRF Connect

- Python tested with mock API returns

- Manual inspection and screenshots confirm proper stat rendering

9. Known Issues and Workarounds

- UUID discovery required via third-party tool (e.g., nRF Connect)

- Frame display space is limited

- Ensure Bluetooth permissions are enabled at runtime

10. FAQs

What if the glasses don't connect?

Make sure they are powered on and close to the tablet.

How do I change what's displayed?

Update the Lua string in the Kotlin BLE callback.

What Android version is required?

Android 10 or later.