Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2020

xXPew^{3}Xx
Lasers: Combat Evolved

DIPTIN DAHAL
JASON AUTRY
KATERINA GOMEZ
PHU LY
TAUSIF ZAMAN

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	1.01.2016	GH	document creation
0.2	02.23.2020	DD, JA, KG, PL, TZ	first draft

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

Lasers Combat Evolved is a simple to use android application that works with a gun and vest pair to allow the user to play a game of laser tag in a geographical area of their choosing by making use of the phone's GPS tracking ability. The phone will act as a relay and validate for the vest/gun to communicate with a remote database server that will store information about the game. The phone will also work as a map for the player by retrieving the player locations and displaying them on a map of the user defined game area.

2 SYSTEM OVERVIEW

The overall system will be similar to a Real-Time System where information will be updates after a certain amount of time. The system will be consist of 3 main layers Database Layer , Android Application Layer and Hardware Layer. The basic strategy is letting them handle difference data processes as such Hardware Layer will be handle only signal data from any hardware devices and pass them on to the Android Application Layer. Using a certain implementation, these data will commute through a Bluetooth Module sub-layer within the Hardware layer that allow the Android Application Layer to request and receive data via Bluetooth signal. Lastly, Database Layer will receive information from Android Application Layer to control and process it for placement in the database. The processed data will then sent to the database for storage until it is needed by the Android Application Layer. The relevant processed data will then be retrieved by Android Application Layer for display.

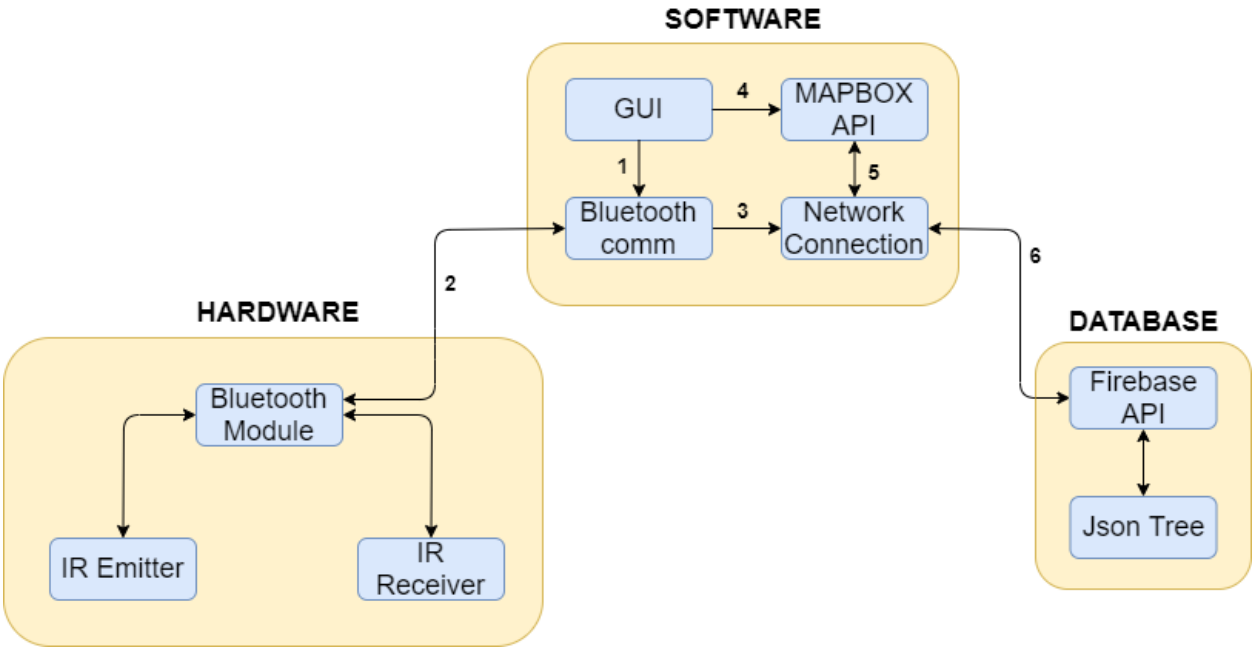

Figure 1: System architecture

3 Database Layer Subsystems

3.1 Layer Hardware

This layer mainly involved with the Fire-base system; therefore, there is not a specific hardware components will be involved in the application itself.

3.2 Layer Operating System

This layer required a Network OS system since Fire-base is required in this layer and it is also a network OS System at core.

3.3 Layer Software Dependencies

The most important dependency on this layer is Fire-base API due to its necessary to access the database from the any clients to the Fire-base server.

3.4 Fire-base Network

Fire-base Network is the key subsystem for this layer. This subsystem provide a strong connection between any players in the game to the database. This is a concrete base to transfer data among the players by establishing the connections and acting as a gate way for the passages to go through.

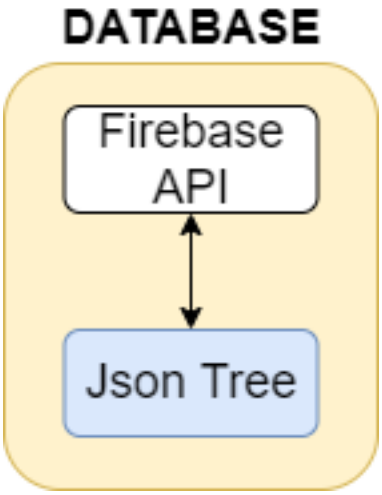

Figure 2: Database subsystem description diagram

3.4.1 Subsystem Programming Languages

Fire-base Network required C script to run due to C being the main language used for Unity App Development. This sub-system also used JavaScript as its back-end structure from the Fire-base software.

3.4.2 Subsystem Data Structures

This subsystem will recognize users information input from the network as player.

3.4.3 Subsystem Data Processing

This system will receive players' data from the network API by Android Application Layer and compares it to the data information as a method of authenticating.

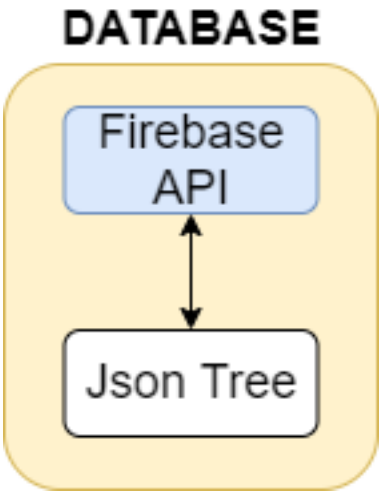

Figure 3: Database subsystem description diagram

3.5 JSON TREE

3.5.1 SUBSYSTEM PROGRAMMING LANGUAGES

Fire-base Network required C script to run due to C being the main language used for Unity App Development. This sub-system also used JavaScript as its back-end structure from the Fire-base software.

3.5.2 SUBSYSTEM DATA STRUCTURES

the Data will be collected and divided into two team, each team with the same number of player, each player have proper player ID, team ID, location with longitudes and latitudes.

3.5.3 SUBSYSTEM DATA PROCESSING

After the authentication, the data will be put in the correct order based on where it should be in the Json Tree.

4 Android Application Layer Subsystems

4.1 Layer Hardware

This layer required the actual android smart phone as the main hardware to have the application to be running and it must have network connection such as wifi and mobile connection.

4.2 Layer Operating System

This layer will be mainly on Android OS system since it is our main objective.

4.3 Layer Software Dependencies

This layer required the minimum of software as Android version 8 .

4.4 Mapbox API

Mapbox Api is an SDK tool that provides developer to mapping the location of users on the actual map. This subsystem was named base on its function which it will use the Mapbox tool to record the location of users and put it onto the map.

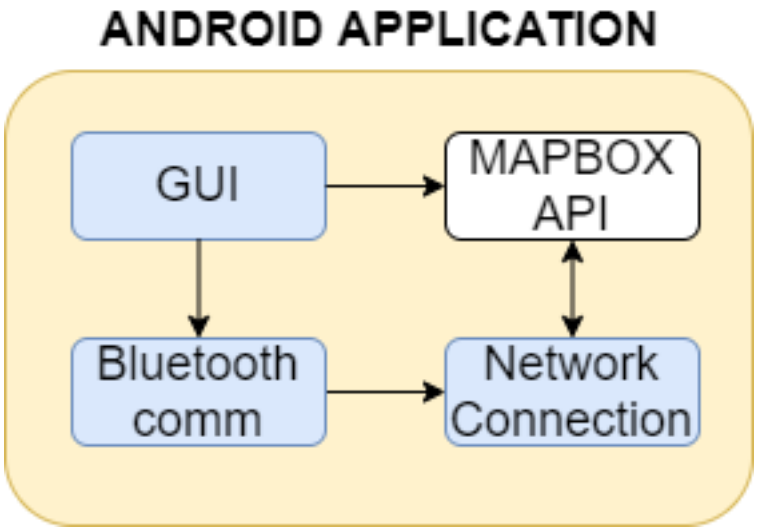

Figure 4: MAPBOX API in Android Application

4.4.1 Subsystem Hardware

The hardware must capability of GPS locating due since Mapbox required the status of the phone based on the global rid as its input.

4.4.2 Subsystem Programming Languages

Mapbox required C since this subsystem will be programs and modifies in Unity Script.

4.4.3 Subsystem Data Structures

Mapbox required inputs location of Longitudes and Latitude on the current device. The structure must have two double variable represent them.

4.4.4 Subsystem Data Processing

Mapbox took the data record and map its on to the database.

4.5 Bluetooth Communication

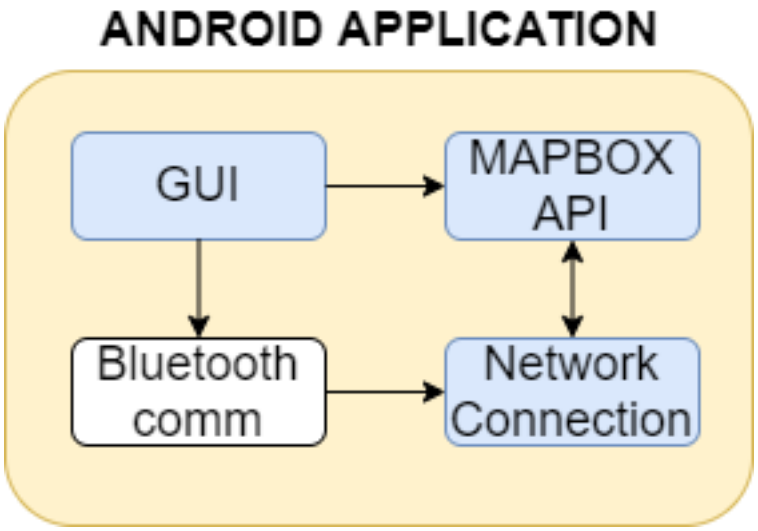

Figure 5: Bluetooth communication in Android Application

4.5.1 Subsystem Hardware

This subsystem required the smart phone device to have a Bluetooth options in order to communicate with the hardware device.

4.5.2 Subsystem Programming Languages

4.5.3 Subsystem Data Structures

This issue is in development process.

4.5.4 Subsystem Data Processing

This issue is in development process.

4.6 Network Connection

This sub layer designed to connect the player to the database, hence, it will connect one players to the other players. Basically, the sub-layer will allow players connect to the database and store their information based on player's data.

4.6.1 Subsystem Programming Languages

This subsystem will used Unity Script which it will be in C.

4.6.2 Subsystem Data Structures

This issue is in development process.

4.6.3 Subsystem Data Processing

This issue is in development process.

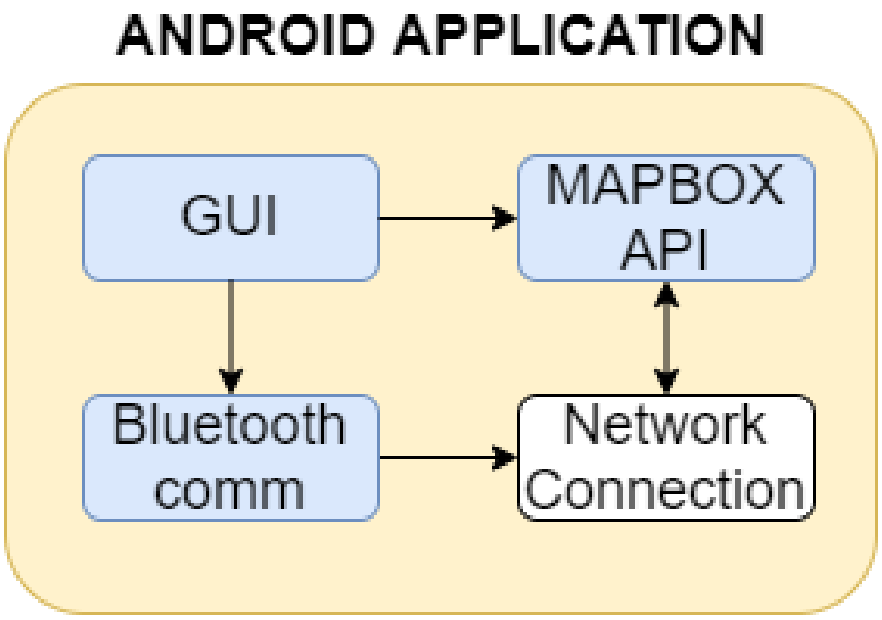

Figure 6: Network Connection in Android Application

4.7 Graphical User Interface

This sub system mainly is the displacement of the app. Its mainly shows what is going on in the app and showing inputs that the player's putting it in. This sub system supposed to have a good look and feels.

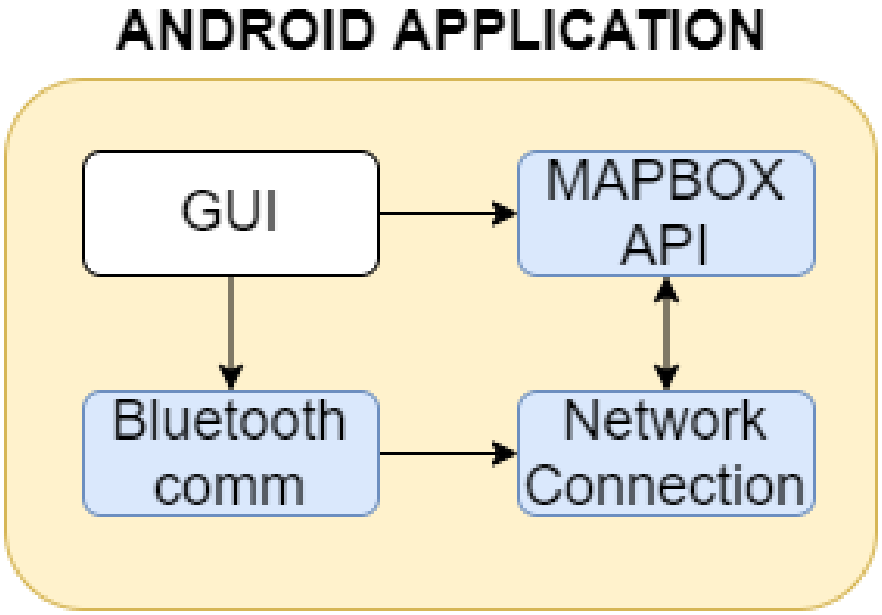

Figure 7: GUI in Android Application

4.7.1 Subsystem Hardware

The sub system prefer a clean screen as displacement however, as long as the hardware comes with a screen and it works.

4.7.2 Subsystem Programming Languages

This subsystem will used Unity Script which it will be in C.

4.7.3 Subsystem Data Structures

Data mostly be the scene and how it looks and feel in each scene and this is still in development process.

4.7.4 Subsystem Data Processing

Data will be process by moving players' data from one scenes to another scene and it will be exclusively prevent the data from being breaks during the transformation.

5 Hardware Layer Subsystems

This subsystem comprises of all the hardware aspects of the gun device. The system comprises of a IR emitter, which will be used to send the necessary packets of information to the opponent's vests.

5.1 Layer Hardware

The Micro Controller - Arduino Uno is the main component of the hardware layer. This layer connects with Grove Base Shield V2, which consists of all the ports for adding devices like IR Emitter - Seeed Studio Grove and IR Receiver - Seeed Studio Grove IR Receiver.

5.2 Layer Operating System

5.3 Layer Software Dependencies

This layer is programmed using C. The Genuino IDE from arduino is used for editing all the codes. And the code is dependent on the Infrared Library.

5.4 Bluetooth Module

This subsystem stays in between hardware layer (IR emitter and IR receiver) and the android layer. The HiLetgo HC-05 blue-tooth module is used in this layer, and this helps transmit data between different systems.

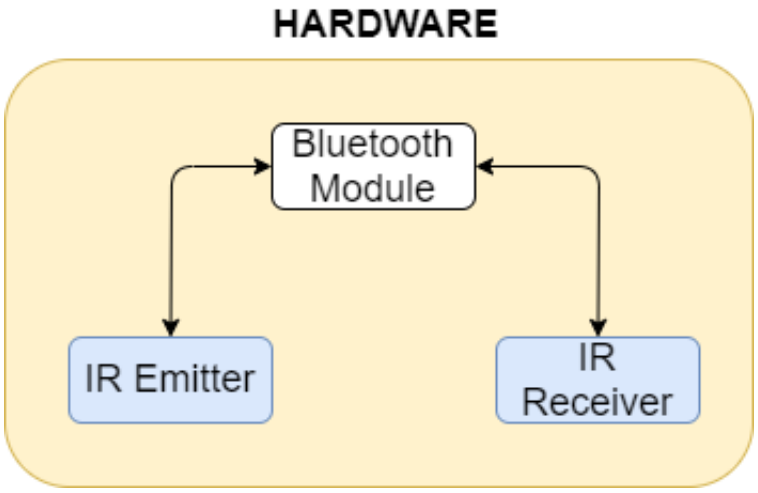

Figure 8: Bluetooth Module in Hardware Subsystem

5.4.1 Subsystem Hardware

Bluetooth Modeule - HiLetgo HC-05 Wireless Bluetooth RF Transceiver 6 Pin Wireless Serial Port Communication

5.4.2 Subsystem Operating System

5.4.3 Subsystem Software Dependencies

The Genuino IDE from arduino is used for editing all the codes. And the code is dependent on the Software Serial library.

5.4.4 Subsystem Programming Languages

This layer is programmed using C.

5.4.5 Subsystem Data Structures

Default Bluetooth data structure

5.4.6 Subsystem Data Processing

No data processing in this layer.

5.5 IR Emitter

This layer is responsible for transmitting digital signals into the real world as analog signals to transmit information. The data received from the blue-tooth module is transmitted.

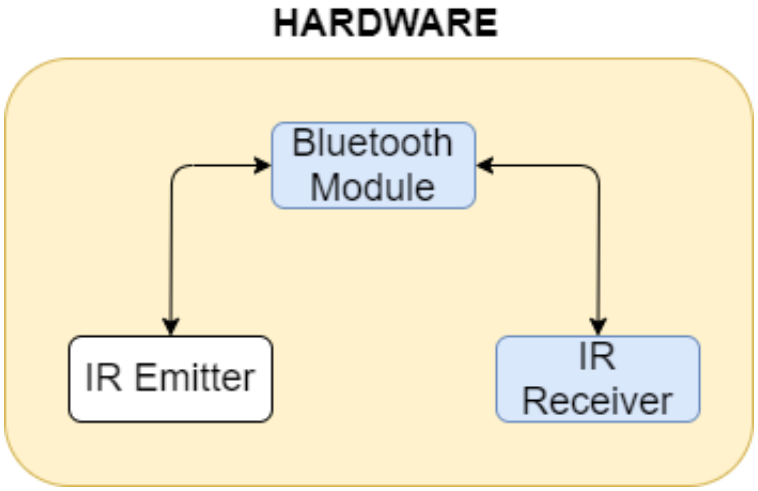

Figure 9: IR Emitter in Hardware Subsystem

5.5.1 Subsystem Hardware

IR Emitter - Seeed Studio Grove IR Emitter Push Button - Seeed Studio Grove Push Button

5.5.2 Subsystem Operating System

5.5.3 Subsystem Software Dependencies

5.5.4 Subsystem Programming Languages

5.5.5 Subsystem Data Structures

5.5.6 Subsystem Data Processing

5.6 IR Receiver

This layer is responsible for receiving analog signals from the IR transmitter. The data received is transmitted to the blue-tooth module and is next updated in database layer.

5.6.1 Subsystem Hardware

IR Receiver - Seeed Studio Grove IR Receiver

5.6.2 Subsystem Operating System

5.6.3 Subsystem Software Dependencies

5.6.4 Subsystem Programming Languages

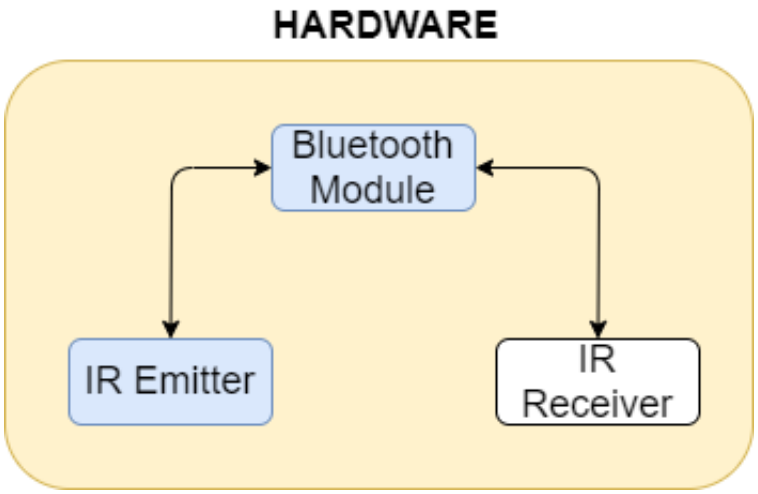

Figure 10: IR Receiver in Hardware Subsystem

5.6.5 Subsystem Data Structures

5.6.6 Subsystem Data Processing

6 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES