

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**ARCHITECTURAL DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SPRING 2020**



**TEAM INFRARED
INFRARED ARENA**

**EDGAR ACEVEDO
JUSTINE BATONGMALAKI
LINDA PHANVILAY
DANNY VU
JEAN-MARCEL YACHO**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.21.2019	LP	document creation
1.0	11.11.2019	LP, JB, DV, JMY, EA	complete draft
2.0	12.10.2019	LP, JB, DV, JMY, EA	complete draft ver 2
3.0	05.02.2020	LP, JB, DV, JMY, EA	complete draft ver 3
4.0	05.12.2020	EA, JB, JMY, DV, LP	final draft

CONTENTS

- 1 Introduction** **5**
 - 1.1 Product Concept 5
 - 1.2 Scope 5
 - 1.3 Key Requirements 5

- 2 System Overview** **6**
 - 2.1 Client Layer Description 6
 - 2.2 Server Layer Description 6
 - 2.3 Tagging Layer Description 6
 - 2.4 Network Layer Description 6

- 3 Subsystem Definitions & Data Flow** **7**

- 4 Client Subsystems** **8**
 - 4.1 User Interface 8

- 5 Server Subsystems** **10**
 - 5.1 Cloud Subsystem 10
 - 5.2 Database Subsystem 11

- 6 Tagging Subsystems** **13**
 - 6.1 Emitter Subsystem 13
 - 6.2 Sensor subsystem 13

- 7 Network Subsystems** **15**
 - 7.1 Bluetooth Subsystem 15
 - 7.2 Internet Connection Subsystem 15
 - 7.3 GPS Subsystem 16

LIST OF FIGURES

1	A simple architectural layer diagram	6
2	A simple data flow diagram	7
3	UI subsystem description diagram	8
4	Cloud subsystem description diagram	10
5	Database subsystem description diagram	11
6	Emitter subsystem description diagram	13
7	Sensor subsystem description diagram	14
8	Bluetooth subsystem description diagram	15
9	Internet connection subsystem description diagram	16
10	GPS subsystem description diagram	17

LIST OF TABLES

2	Dataflow Descriptions	7
3	UI subsystem interfaces	9
4	Cloud interfaces	10
5	Database interfaces	12
6	Emitter Subsystem interfaces	13
7	Sensor Subsystem interfaces	14
8	Bluetooth interfaces	15
9	Internet interfaces	16
10	GPS interfaces	17

1 INTRODUCTION

1.1 PRODUCT CONCEPT

Infrared Arena will allow users to have their own private laser tag matches with their friends. The system will consist of a laser tag gun, with an attachment for holding the users smartphone, clip-on sensors, and an app user will download on their smartphones. The app will allow users to create and join multiplayer matches which will be hosted on a cloud server. The intended audience for this system is users 12 and up. The app and hardware will be available commercially to the public.

1.2 SCOPE

- The intended audience for this system is users 12 and up.
- The app and hardware will be available commercially to the public.
- The max diameter of the playing field is 1 mile
- Players will connect to the same server using WiFi, data connection, or Bluetooth connection.
- Total development costs will not exceed \$800.
- Application is only available using Android.

1.3 KEY REQUIREMENTS

- Team will take an existing system (RECOIL) and alter it to their needs to create a better gaming experience. System will consist of phone (mobile application) and hardware (physical laser gun).
- Mobile application will be able to sense changes in hardware and update real time, such as ammo and health of player
- App will be connected to a map API in order to create a reflect the real world terrain onto the mini map in the game.
- User's will be able to draw a game boundary through the mobile application.
- App will be cloud based to allow multiple users to access the game servers to play.
- App will display a smaller version of the map of the arena drawn by players.
- Mini map will display the location and direction of friendlies using GPS of user.
- Teammates will be able to communicate using pings such as an enemy being spotted, missing, rendezvous, and back up. Teammate will be notified by display on their mini map, by sound, or by text on screen. Physical laser gun requires an orange tip on the outer rim of the gun
- Physical laser gun requires an orange tip on the outer rim of the gun.

2 SYSTEM OVERVIEW

This section describes the overall structure of our software system as represented by Figure 1 and is comprised of four layers, client, network, tagging, and server layer.

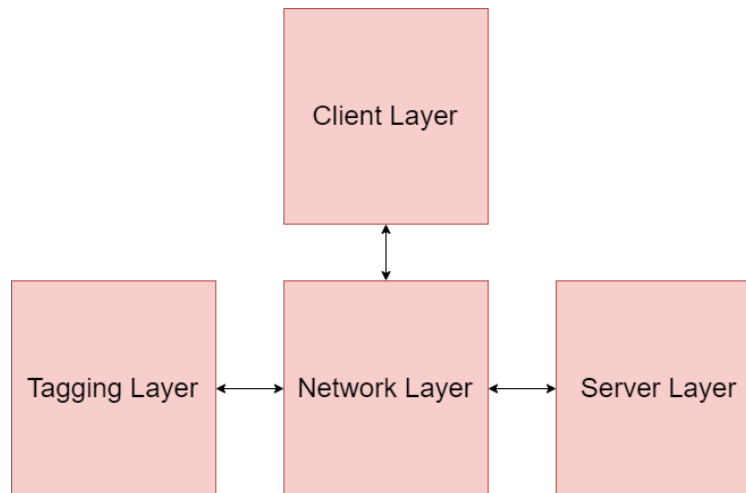


Figure 1: A simple architectural layer diagram

2.1 CLIENT LAYER DESCRIPTION

The client layer represents the smartphone being used to play Infrared Arena. The only subsystem of this layer is the mobile application. The app allows users to create or join multiplayer matches. The UI of the app will be constantly updated based on interactions with the other layers of the Infrared Arena systems. The client layer will receive data from the tagging layer through the Bluetooth subsystem of the network layer. This data, along with user data collected by the app, will be sent to the server layer through the internet subsystem of the network layer. In the server layer, data collected from each player will be processed. After processing the game state will be updated, and these updates will be sent back to the client layer. These updates to the game state will be reflected in the UI of the app.

2.2 SERVER LAYER DESCRIPTION

The server layer contains 2 subsystems that work together to store data and provide the user with information as needed. The Cloud subsystem provides our computing environment and management of data and resources. It continuously stores, manages, and updates the data which may be received through the user's GPS and online connectivity. The Database subsystem receives its data through the cloud and also sends information back to the user.

2.3 TAGGING LAYER DESCRIPTION

The tagging layer consists of the emitter and sensor subsystem. The emitter subsystem is used to target and tag players, while the sensor subsystem is used to receive the tag.

2.4 NETWORK LAYER DESCRIPTION

The Network Layer subsystem is in charge of data connection between the Client, Tagging and Server subsystems. Input into the Network layer is through Bluetooth connectivity from the Tagging subsystem to the Client subsystem. The Tagging subsystem determines if user is tagged. The Client subsystem sends data through a internet connection to the Network subsystem which goes to the Server subsystem. The Server subsystem connects users to the laser tag game.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

This section breaks down our system's layer of abstraction to another level of detail. As described below, Figure 2 provides more insight into the logical subsystems that compose each layer and how they interact/interface between one another. These corresponding subsystems will be further detailed in the sections to follow.

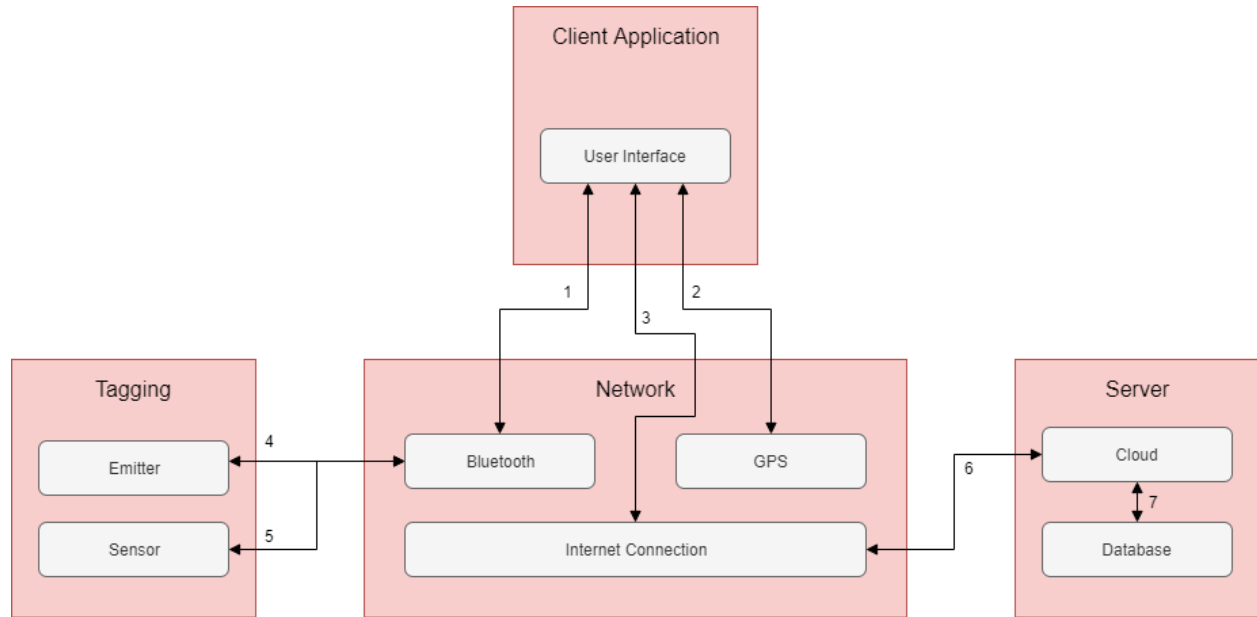


Figure 2: A simple data flow diagram

ID	Description
1	User device connects application to hardware by connecting to Bluetooth
2	User device receiving location data from GPS service
3	Data sent to cloud for processing
4	Emitter sends back information
5	Sensors sends back information
6	Data sent to cloud for storage
7	Information in database managed by cloud

Table 2: Dataflow Descriptions

4 CLIENT SUBSYSTEMS

This layer is the main focus of our project. We are developing this application which will be the users tool to connect to the toy guns, connect with friends, and play the game. Any input will come through this layer and will output to other layers.

4.1 USER INTERFACE

The mobile application is one of the main sources of user interaction in Infrared Arena. The app has 2 main sources of input: the user, and the tagging layer. Data collected from the user and tagging layer will be sent to the cloud layer over the internet. This data will be used to update the current game state. Updates to the game state will be presented to the user through the UI in the app.

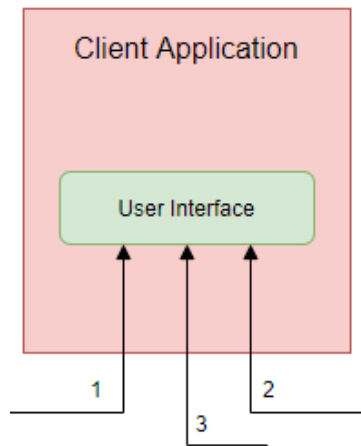


Figure 3: UI subsystem description diagram

4.1.1 ASSUMPTIONS

It is assumed that users of Infrared Arena own a smartphone that has access to mobile data, wifi, bluetooth, and GPS services. The smartphone will be running Android 8.0 or newer.

4.1.2 RESPONSIBILITIES

The UI of the app will feature an interactive map of the users surroundings and have clearly marked boundaries for the play area. The player will be able to see the locations of their teammates and the direction they are facing. User can ping different points on the map with built in messages such enemy here or rendezvous here. Ping points will be visible to all the members of the user's team. Scoreboard will be displayed during matches and will be updated accordingly. This scoreboard will display individual scores of all players, as well as total team scores. Users will be able to choose a nickname that will be displayed to other users during matches. The app will allow users to choose between joining or hosting a match. When hosting a match, users will be able to adjust match settings such as time limit and game mode.

4.1.3 SUBSYSTEM INTERFACES

Table 3: UI subsystem interfaces

ID	Description	Inputs	Outputs
#01	Map	User selects ping type User selects point on map	Ping point is displayed to users teammates
#02	Scoreboard	User eliminates opponent	Scoreboard is updated
#03	UI buttons	User taps button	Changes to new screen Input is sent to the cloud

5 SERVER SUBSYSTEMS

The purpose of the Server layer is to serve as a connection point for players. The server will provide a common connection point for players to exchange their network information, allowing them to connect in game. In addition, the server will act as a processing unit for the application, processing all inputs from players to update the game state. This will be achieved through the cloud and database subsystems. Further details are described below.

5.1 CLOUD SUBSYSTEM

The cloud subsystem will utilize Firebase services such as Realtime Database and Firestore. The Realtime Database follows a NoSQL format and automatically synchronizes the information when connected to an Internet connection. Firebase allows for the user to temporarily go offline without losing data and quickly syncing once a connection is formed again. In the Fall, the team was set to use DigitalOcean services to create DigitalOcean Droplets that are Linux-based virtual machines that run on top of virtualized hardware but were unable to use due to budget constraints and issues corresponding with COVID-19.

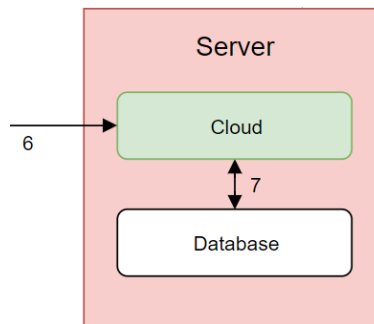


Figure 4: Cloud subsystem description diagram

5.1.1 ASSUMPTIONS

It is assumed that users of Infrared Arena own a smartphone that have access to data or Wifi, Bluetooth, and GPS services, in order to access the cloud and therefore, the server.

5.1.2 RESPONSIBILITIES

The cloud will interact with the client by receiving data transferred through HTTP Requests in the network layer. The data received will be stored in the database subsystem that will be managed through the cloud.

5.1.3 SUBSYSTEM INTERFACES

Table 4: Cloud interfaces

ID	Description	Inputs	Outputs
#01	Transfer data between layers	Data received from other layers/database	Direct data to intended layers
#02	Store data in database	Data received from other layers	Transfer data to database

5.2 DATABASE SUBSYSTEM

The Database Subsystem will receive input from users and send information back to their devices. One key feature of multiplayer games is the ability to sync the state between the client and server. The Database will continuously receive, store, and update the information needed for the game's program to simulate the changes in the environment with respect of time and players input.

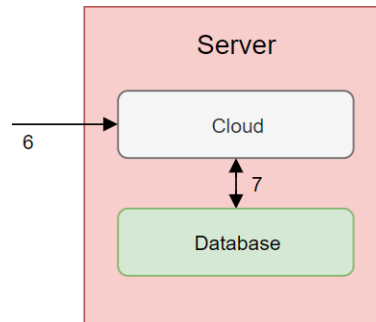


Figure 5: Database subsystem description diagram

5.2.1 ASSUMPTIONS

There are multiple approaches to syncing the state of a multiplayer game. This database is run on the assumption that the server applies Server Reconciliation, for client-server synchronization. This database is also expected to be utilizing Firebase Realtime Database service for this application.

5.2.2 RESPONSIBILITIES

The Database will interact with the Cloud subsystem to send and receive information to the user. This subsystem will update the state of the game and the frame, encompassing player interactions, elements, and movement (GPS) in the application. To achieve this, the database needs to maintain at least these 3 main responsibilities:

- **GameStateHistory:** The Database is expected to save the history of GameState within a time frame P. The GameState is the state of the game on the client at the time of applied user input, and should be tagged with a timestamp to be used to tell the server when to process a user input.
- **ProcessedUserInput -** History of UserInput processed within a time frame P to be used in updating the current GameState.
- **UnprocessedUserInput -** When the server receives an input from the user, it should be inserted into the UnprocessedUserInput. This is UserInput received, but not processed yet, also within time frame P.

5.2.3 SUBSYSTEM INTERFACES

Table 5: Database interfaces

ID	Description	Inputs	Outputs
#01	Save Game State	data automatically received from client	stored information to database
#02	Save game server	data received from other layers	stored data to database
#03	Extract game server data	data received from database	data sent to client
#04	Save Processed User Input	data on processed user input	stored data to database
#05	Save Unprocessed User Input	data on received user input	stored data to database

6 TAGGING SUBSYSTEMS

The Tagging subsystem utilizes the emitter subsystem to tag the targeted player and initiate a hit which sends the information to the client subsystem. The sensor subsystem receives the tag and sends the information towards the client subsystem.

6.1 EMITTER SUBSYSTEM

The emitter subsystem is the laser gun. The targeting of the laser is to shoot towards the enemy and tag them for points.

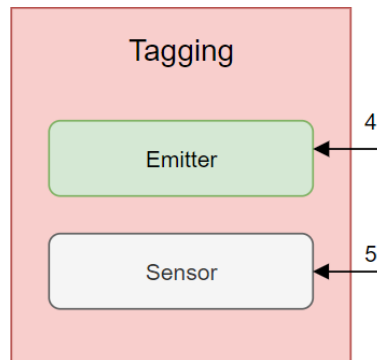


Figure 6: Emitter subsystem description diagram

6.1.1 ASSUMPTIONS

Laser gun expected to be connected through Bluetooth between the emitter and phone. Sensors are connected and setup.

6.1.2 RESPONSIBILITIES

The laser gun will interact with the client layer by connecting to the phone through the network layer which utilizes the Bluetooth connectivity.

6.1.3 SUBSYSTEM INTERFACES

Table 6: Emitter Subsystem interfaces

ID	Description	Inputs	Outputs
#01	User shoots	Trigger pulled	App Ammo Decrease

6.2 SENSOR SUBSYSTEM

The sensor is the hit markers for the laser gun. These sensors receive the tagging made by the player and sends that information to score points for that team/player.

6.2.1 ASSUMPTIONS

Laser gun is connected with sensors.

6.2.2 RESPONSIBILITIES

The sensor subsystem is responsible for receiving the tag. It will interact with the emitter subsystem.

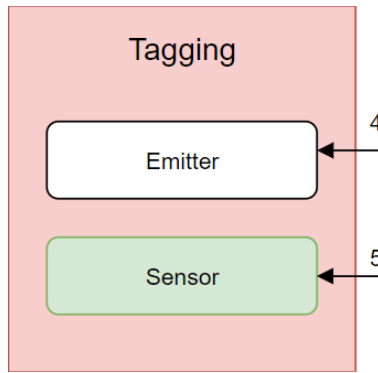


Figure 7: Sensor subsystem description diagram

6.2.3 SUBSYSTEM INTERFACES

Table 7: Sensor Subsystem interfaces

ID	Description	Inputs	Outputs
#01	User is hit	receive dmg	health decrease

7 NETWORK SUBSYSTEMS

7.1 BLUETOOTH SUBSYSTEM

The Bluetooth subsystem will serve as the connection between the phone and hardware system to indicate the changes from the hardware system into the application.

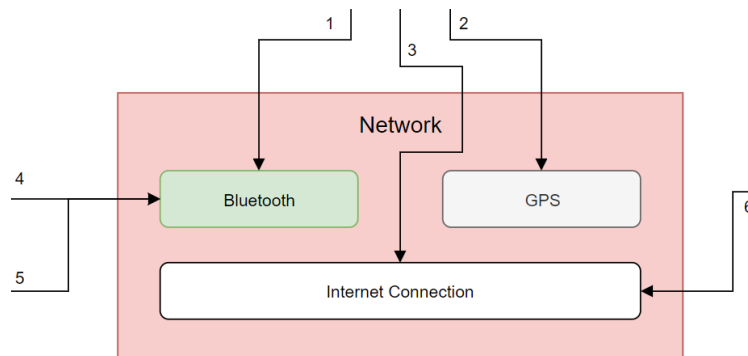


Figure 8: Bluetooth subsystem description diagram

7.1.1 ASSUMPTIONS

- Device is able to connect to bluetooth.
- Phone has bluetooth enabled.
- There is negligible latency between bluetooth connection from laser gun to phone.
- Internet subsystem is working.

7.1.2 RESPONSIBILITIES

Bluetooth is responsible for the connection between the network layer, client layer and tagging layer subsystems. It serves as a point of communication with the phone and laser gun.

7.1.3 BLUETOOTH SUBSYSTEM INTERFACES

Table 8: Bluetooth interfaces

ID	Description	Inputs	Outputs
#01	Bluetooth connection	sync device	device connected
#02	User has no ammo	N/A	UI notifies to reload

7.2 INTERNET CONNECTION SUBSYSTEM

The Internet Connection subsystem of the Network Layer's purpose to send the data from the Client Layer to the Network Layer which then goes to the Server Layer from which the client will use to play the application.

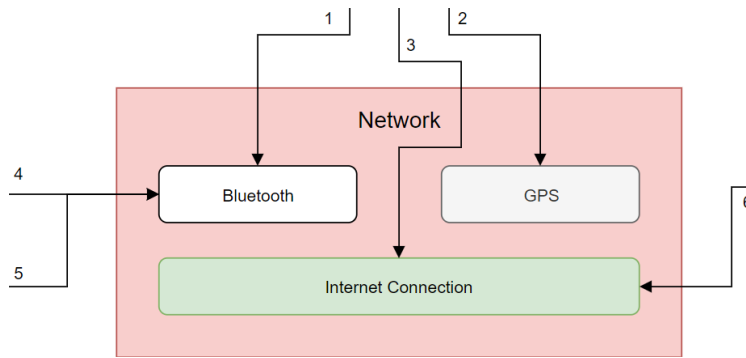


Figure 9: Internet connection subsystem description diagram

7.2.1 ASSUMPTIONS

- Bluetooth subsystem is working.
- Internet connection is stable.
- Internet connection is secure.
- All user's are able to connect to the same server.
- The server is available and working.
- The GPS connection is working.

7.2.2 RESPONSIBILITIES

- Every time the bluetooth connection between the hardware system receives a change, the UI will update.

7.2.3 SUBSYSTEM INTERFACES

Table 9: Internet interfaces

ID	Description	Inputs	Outputs
#01	Players Interaction	Health Decr Health Incr No Health Ammo Decr Ammo Incr No Ammo No health	UI Updates
#02	Pinging enemies	Gather ping placed Enemy spotted ping placed	Respective ping icon is appears on teammates screens

7.3 GPS SUBSYSTEM

The GPS subsystem of the Network Layer's purpose is to update the location, direction, and position of the players in the game.

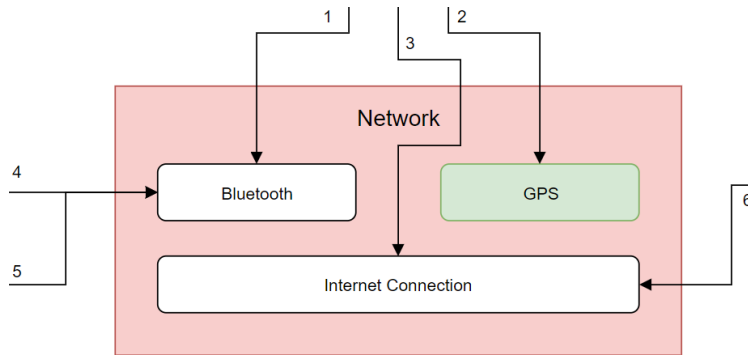


Figure 10: GPS subsystem description diagram

7.3.1 ASSUMPTIONS

- Users have internet connection obtained in some way (wifi or data connection).
- The GPS location of the device is accurate.
- The Bluetooth connection of the device is working.
- The user has given location access to the application.
- MAP API is correctly implemented.

7.3.2 RESPONSIBILITIES

GPS interacts with the client layer as it gives the view of the map. The map gets updated with player locations through using the internet subsystem.

7.3.3 SUBSYSTEM INTERFACES

Table 10: GPS interfaces

ID	Description	Inputs	Outputs
#01	Direction changes	GPS Player Movement	Player Icon Direction Updates Map screen updates
#02	Player moves	GPS Player movement	Player location updates Map screen updates