জাতীয় বিশ্ববিদ্যালয়
National University Bangladesh

Project Report on

Inframote (IR Remote Application)

Submitted to:
The Controller of the Examinations
National University Bangladesh
Gazipur-1704, Bangladesh

Supervised By:
Mr. Mohammad Ariful Hyder
Professor
Department of Computer Science and Engineering
Dhaka City College

Prepared By:

Ahesanul Karim Abid
Registration Number: 16502000521
Session: 2016-17
Bachelor of Computer Science and Engineering
Dhaka City College (NU-6408)

Dhaka City College
(ESTD. 1957)
Road No. 2, Dhanmondi Residential Area, Dhaka-1205, Bangladesh
www.dhakacitycollege.edu.bd
Date of submission: 15/12/2021

Letter of Testimonial

5th December 2021
To The Controller of the Examinations
National University Gazipur-1704, Bangladesh

Subject: Submission of project report on “Inframote (IR Remote Application)”

Dear Sir,
This is to inform you that my report on the project "Inframote (IR Remote Application)" has been completed.

I followed the instructions of my institution's supervisor while also attempting to follow the guidelines provided by you in preparing this project. However, this has equipped me with a lot of experience that will help me perform similar projects in the future. I'd like to thank you, as well as my supervisor and department, for their assistance and advice in preparing the report, presentation and project.

If you require any extra information regarding this report, I will be happy to supply it. Thank you very much.

Sincerely yours,

(Ahesanul Karim Abid)
Registration Number: 16502000521
Session: 2016-17
Computer Science and Engineering
Dhaka City College (NU-6408)

Declaration

I solemnly declare that the work described in this project report, titled "Inframote (IR Remote application)," was completed under the direction of Md. Ariful Hyder, Department of Computer Science and Engineering, Dhaka City College, Bangladesh. It has never been submitted to another university, college, or organization for any academic qualification, certificate, diploma, or degree. I have not violated on any existing copyright, and no part of this report has been plagiarized from previous work done for a degree or other purposes.

(Ahesanul Karim Abid)
Registration Number: 16502000521
Session: 2016-17
Computer Science and Engineering
Dhaka City College (NU-6408)

Acknowledgment

To begin, I would want to offer my heartfelt gratitude to Almighty God for providing me with the opportunity to finish my project report in a very convenient manner.

Mr. Mohammad Ariful Hyder, Professor of Computer Science and Engineering Department at Dhaka City College, is my esteemed supervisor, and I owe him my gratitude and thanks. He worked with me throughout the process of finishing my project. His guidance improved my confidence and enabled me to complete my studies and project on schedule. I'm also grateful to my department head, as well as all of the professors and seniors who helped me along the way by mentoring us and reviewing my work.

Finally, I would like to express my gratitude to all who have read this report and will benefit from it in the future.

(Ahesanul Karim Abid)
Registration Number: 16502000521
Session: 2016-17
Computer Science and Engineering
Dhaka City College (NU-6408)

Table of Contents

Chapter 1
Introduction

Chapter 1 | Introduction

1.1 Introduction

Infrared signals are being used by most remote-controlled gadgets (such as TVs, DVD players, and air conditioners) to connect with the host device because they consume minimum power and work over a short distance. Some recent smartphones include an infrared blaster that can act as a remote control. However, without this feature, infrared signals can be made utilizing the headphone jack/USB port and an IR transmitter LED that can interface with the smartphone to generate required IR signals.

Some apps (MI Remote, Zaza Remote, and so on) have been already developed to control this IR blaster, but only a few of them support the workaround I have mentioned. Sadly, I was unable to locate any apps that allow users to contribute or create custom remotes. I also discovered that none of them support the local branded devices (such as Minister, Vision, Defender, Walton).

To resolve these issues, I propose to develop a new Android app called InfraMote. InfraMote will be a free app for Android users who would like to utilize their phone as a IR remote controller to manage their TV, fan, light, or air conditioner, or any other electronic equipment. This app will address all of the flaws mentioned above.

1.1.1 Motivation

The project will enable me to gain knowledge and experience in a variety of computer science and engineering fields, including server management, database management, backend technology, REST API, android application development, microectronics, embedded systems, digital signal processing, SDLC (Software development life cycle), project management, documentation writing, and presentation skills.

I want to add this project to my portfolio, use the gathered knowledge and experience to land my ideal job, and start my own business with this knowledge and expertise. However, with the help of Google Ads and the Google Play store, the end product of the project may be able to generate a considerable amount of revenue; if this is the case, I would like to improve the features of the service and offer a more acceptable and upgraded product under this domain.

Last but not least, everyone, including myself, will be able to enjoy a new Android application with local remote support, which excludes the drawbacks described in the previous section.

1.2 limitation of existing system

I've reviewed and tested a number of apps that allow you to control an IR transmitter or IR blaster that are already available in the Android app store.

1.2.1 Available apps

Here is a list of apps that I have reviewed:

MI Remote Controller (Xiaomi Inc)

Play store Rating: 4.1 (Recorded in 9 December 2021)

Notable Feature:

i. Friendly UI

Limitations

i. Does not have a lots of remote
ii. Does not have local branded remotes (Vision, Minister, etc.)
iii. No Customization
iv. Supports only few android devices
v. Requires internet for adding new remote configuration
vi. Drains a lot of power
vii. Does not support user contribution

ZaZa Remote (Tiqiaa Co. Ltd)

Play store Rating: 4.2 (Recorded in 9 December 2021)

Notable Feature

i. Supports a large number of remote
ii. Supports DIY remote control using headphone jack and IR Transmitter LED
iii. Usable with Alexa

Limitations

i. Does not have local remotes
ii. No Customization
iii. Requires internet for adding new remote configuration
iv. Does not support user contribution

More similar remote apps:

Lean Remote (Lean remote) - Rating 4.3, Broadlink Universal TV Remote (Broadlink) - Ratting 4.5

Summary of limitation in existing system vs Inframote

Features	MI Remote Controller	Zaza Remote	Others	Inframote
Allows contribution	✖	✖	✖	✔
Supports external receiver	✖	✖	✖	✔
Ability to create custom remote	✖	✖	⚠	✔
Supports local device	✖	✖	⚠	✔
Friendly UI	✔	✖	✖	✔

1.3 Proposed System

The proposed system is an Android app that solves all the issues discussed in 1.2. I have pointed out the goals here:

i. Creating an IR remote android application that solves all the limitations below:
    1. Allows contribution from users
    2. Has supports for all Local Brands
    3. Has supports for External/DIY IR Transmitter device
    4. Has ability to create custom remote using the app
    5. Has ability to record IR signals
ii. Creating a central API and database that will be responsible for managing remote and users data
iii. Prototyping an IR receiver model to collect and decode IR signal from existing remotes

1.4 Business Model

I'll utilize Google Ads to monetize this Android app and cover the costs of the server and database. The app will be proprietary. Ad-free access is available to remote contributors and premium users.

- ✔ Ad-free experience for premium users
- ✔ Ad-free experience for contributors
- ✔ Regular users who are neither premium users nor contributors will have to watch Google ads to use the application

1.5 Conclusion

This chapter provides a comprehensive outline of the planning as well as an overview of the project. The goal and motivation for developing the proposed system have been detailed here. I'll go over My previous knowledge and the background research I'll need to complete this assignment successfully in the upcoming chapter.

Chapter 2
Background Study

Chapter 2 | Background Study

2.1 Introduction

I've provided notes regarding the background studies, which are essential to working on this project, in this chapter. I attempted to explain how the Infrared Remote Controller System works, as well as how the web server, database, and Android application are coupled and can be used in this project. The majority of the information was acquired and analyzed from several websites and a few textbooks. In the reference area, you'll find all those website URLs and books' titles with their authors' names.

2.1.1 Infrared Radiation (IR)

According to Wikipedia,

> *"Infrared, sometimes called infrared light, is electromagnetic radiation with wavelengths longer than those of visible light. It is therefore invisible to the human eye. IR is generally understood to encompass wavelengths from around 1 millimeter to the nominal red edge of the visible spectrum, around 700 nanometers."*

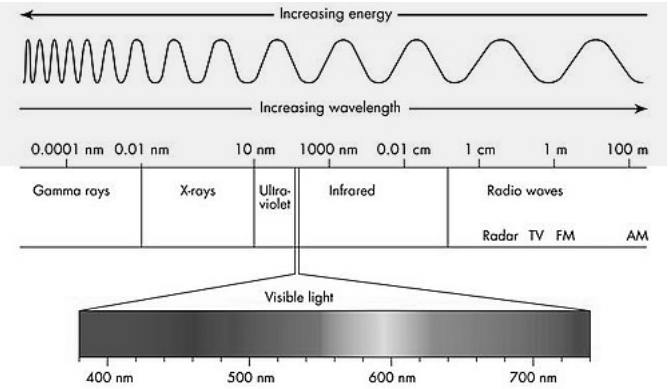


Figure 2.1: Wavelengths of electromagnetic spectrum.

2.1.2 What is IR Remote controller system?

The IR remote converts digital signals into infrared light signals, which it then sends by flashing a specific LED (IR Transmitter) capable of emitting infrared light. Using an IR receiver, the controlled device receives the signal and decodes it into digital signals that the internal microprocessor can understand and execute the instructions from the signal.

2.1.3 Why IR Remotes are so popular?

IR remotes are found in almost every home appliances, including televisions, air conditioners, lights, fans, and speakers. Since the day it was discovered, it has become increasingly popular. This technology has a number of advantages, which are listed below:

i. Power efficient
ii. Wireless communication system
iii. Inexpensive to manufacture
iv. Because of the short range, it's safe
v. The signal is undetectable to the naked eyes
vi. Remote controls are physically smaller in size

2.1.4 Limitation of IR Remote Control System

Despite the fact that this communication system has so many advantages. The following are some of the limitations of the IR Signal:

i. Only works within a certain range of distance (about 30 feet)
ii. Can’t pass through walls
iii. It's possible that it'll interfere with other gadgets in the area
iv. It may be affected by direct sunlight or other light sources that emit infrared radiation
v. Transmission rate is slow.

2.1.5 Working principal of IR Remote

When we push a button on the IR remote, it makes contact with the printed wires beneath it, completing a circuit that the integrated chip on the remote recognizes, and the remote then sends a binary/digital instruction through the IR LED. According to the voltage provided by the integrated chip, the LED transmits a series of light pulses.

Modulation

The IR LED emits infrared light when it receives a modulated signal from the transmitter micro controller unit. Modulation assist in the isolation of the actual signal from the noise.

Demodulation

The IR light is converted to an electric current by a photo diode in the receiver. A transimpedance amplifier is used to convert the current to a voltage. The voltage is amplified, filtered, and demodulated before being used as an input by the receiver micro-controller unit.

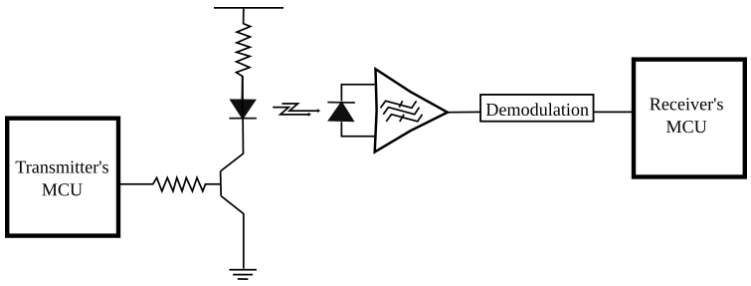


Figure 2.2: Infrared signal modulation and demodulation

2.1.6 Modulation and encoding techniques

Amplitude Shift Keying (ASK) Modulation

The Amplitude Shift Key is abbreviated as ASK. Because of its high performance, simplicity, and low cost, it is a widely common modulation technique across industries. The receiver and transmitter's carrier are tuned to the same frequency in this method, which is commonly 30, 33, 36, 38, 40, or 58 kHz.

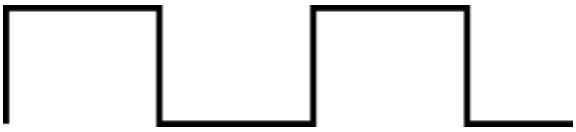


Figure 2.4: Actual Signal

Figure 2.3: ASK Modulated Signal

There are different types of encoding in ASK modulation

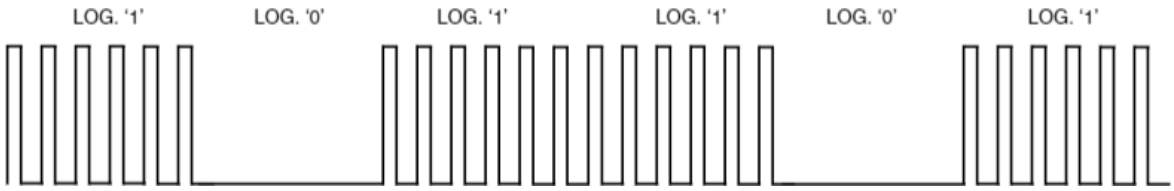


Figure 2.5: Pulse Position Encoding

Position encoding

i. Constant width for each bit
ii. Modulated pulse represents logic 1
iii. Space represents 0

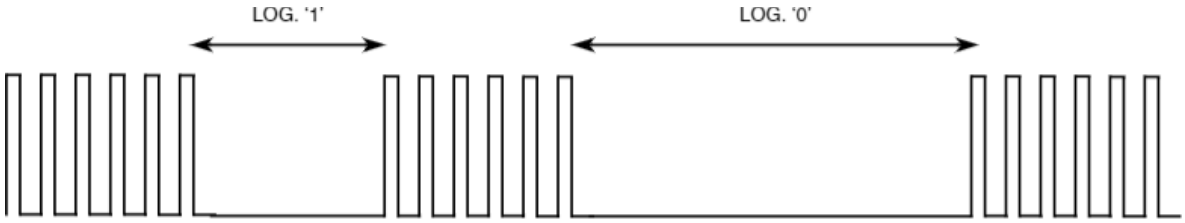


Figure 2.6: Pulse Distance Encoding

Distance encoding

i. Space width distinguish between logic 0 and 1
ii. composed of spaces and pulse

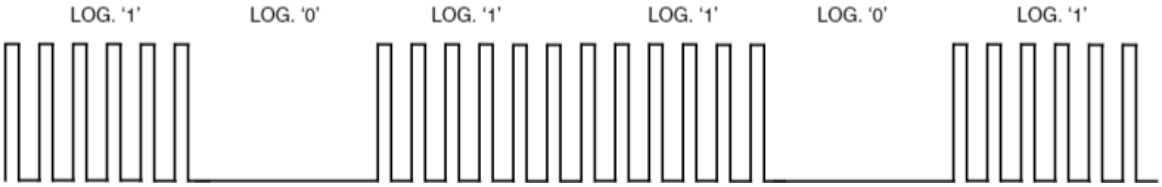


Figure 2.7: Manchester Encoding

Manchester Encoding

i. modulated pulse to space means logic 1
ii. spaces to modulated pulse means logic 0
iii. constant width for each bit

Frequency Shift Keying (FSK) Modulation

Different frequencies are used for different logic levels in FSK modulation. In FSK modulation, there are no intervals between the modulated pulses.

Figure 2.8: Frequency Shift Keying (FSK)

2.1.7 Macro command

It's a set of commands that we can pre-program to run sequentially and deliver with a single button press.

2.2 Software Development

2.2.1 Software Requirement Specification

Different types of Users

End User	The one who uses the product or service.	
Stakeholder	The one who is affected by the product by any means	
	Primary User	End user of the product
	Secondary User	The one who will use the app occasionally or use it through intermediary
Tertiary User	The one who will affected by the use of product or make discussion about the product.	

Types of Requirements

Business Requirement	Defines the purpose of the project
Business Rule	Defines the budget guidelines and regulation. (eg. Legal Regulation, Privacy policies)
User Requirement	Task that user can accomplish with the product: Generally defined with: Use cases, User stories, Story Boards, Scenarios
Functional Requirement	Generally defined using information flow diagram to define the behavior of the product.
Non-Functional Requirement	Defines the following characteristics of the each task: • Accuracy • Dependability • Usability • Security • Efficiency • Performance • Maintainability
External Interfaces	External Parameters that will affect the development of the product (eg. External API, Database etc). Data flow diagram can be used to show the relation of external interface
Physical product settings	Physical parameters that will influence the product
Development Constraints	Production Environment (Bandwidth, Processing Power, Memory, Platform, Environment)

Steps for collecting and documenting Software requirement

(STEP 1) Eliciting Requirement

- Meeting, discussion and collaboration with clients
- Distinguish wants and needs
- Check if the requirements are realistic and feasible

Tools & diagrams used here:

- Use case diagram (Preferred software: Draw.io Desktop)
- Wire-frame (Preferred software: Figma Desktop)
- Story Board (Draw.io/Figma)

(STEP 2) Expressing Requirements

- Framing the requirement
- Writing user stories

Tools & diagrams used here:

- Backlog
- Story Map

(STEP 3) Prioritizing Requirements

Prioritizing the user stories in the backlog by their importance in the following criteria:

- Must be done
- Should be done
- Could be done
- Won't be done

Tools & diagrams used here:

- Backlog

(STEP 4) Analyzing Requirements

Check the documented requirements to make sure the requirements have the following characteristics

- Correct (Accurate)
- Complete (No Missing Information)
- Clear (No Ambiguity)
- Consistent (No Conflict)
- Feasible (Possible To Implement)
- Traceable (Trackable)

(STEP 4) Managing Requirements

- Tracking and managing changes
- Update requirements

2.3 Conclusion

In this chapter, I've attempted to summarize all of the background research that I conducted prior to preparing this project. This chapter details all of the materials and courses I used to prepare for this adventure. In the following chapter, I'll go over the requirements for the Inframote IR Application.

Chapter 3
Software Requirement Specification (SRS)

Chapter 3 | Software Requirement specification (SRS)

3.1 Introduction

The description of what the system should accomplish, the service or services it delivers, and the constraints are all included in the system requirement. This chapter covers the requirements for the system I'll be working on over the next several days. All of the activities that the program will be able to do, as well as their input and output, are specified here. I've listed all of the hardware and software interfaces, as well as the expected reaction times for each activity, how error monitoring will be handled, the system's expected maintainability, what critical information need to protect, the security level, and any other constraints.

3.2 Use Case Diagram

In a system, a use case diagram is used to show distinct actors and their tasks. Use case diagrams assist in recognizing how various types of users will interact with and benefit from the program. It creates a link between each task and its user. A use case diagram helps in the identification of software requirements. Figure 3.1 shows the use case diagram for this application.

3.2.1 Actors

The various categories of app users are referred to as actors. The sign 웃 (Stick Man) denotes an actor in use case diagram. Below is a quick overview of the actors involved in this system:

Consumer

The term "Consumer" refers to an Inframote Android application user who uses the app on a regular basis. In fact, it refers to a user who will get this software from the app store, install it on their phone, and utilize it to get the most out of it by using various remotes.

Account holder

The word "account holder" refers to the app's users who will register and create a user account in order to have an impact on our system (such as contributing and reporting) and store their data in the cloud.

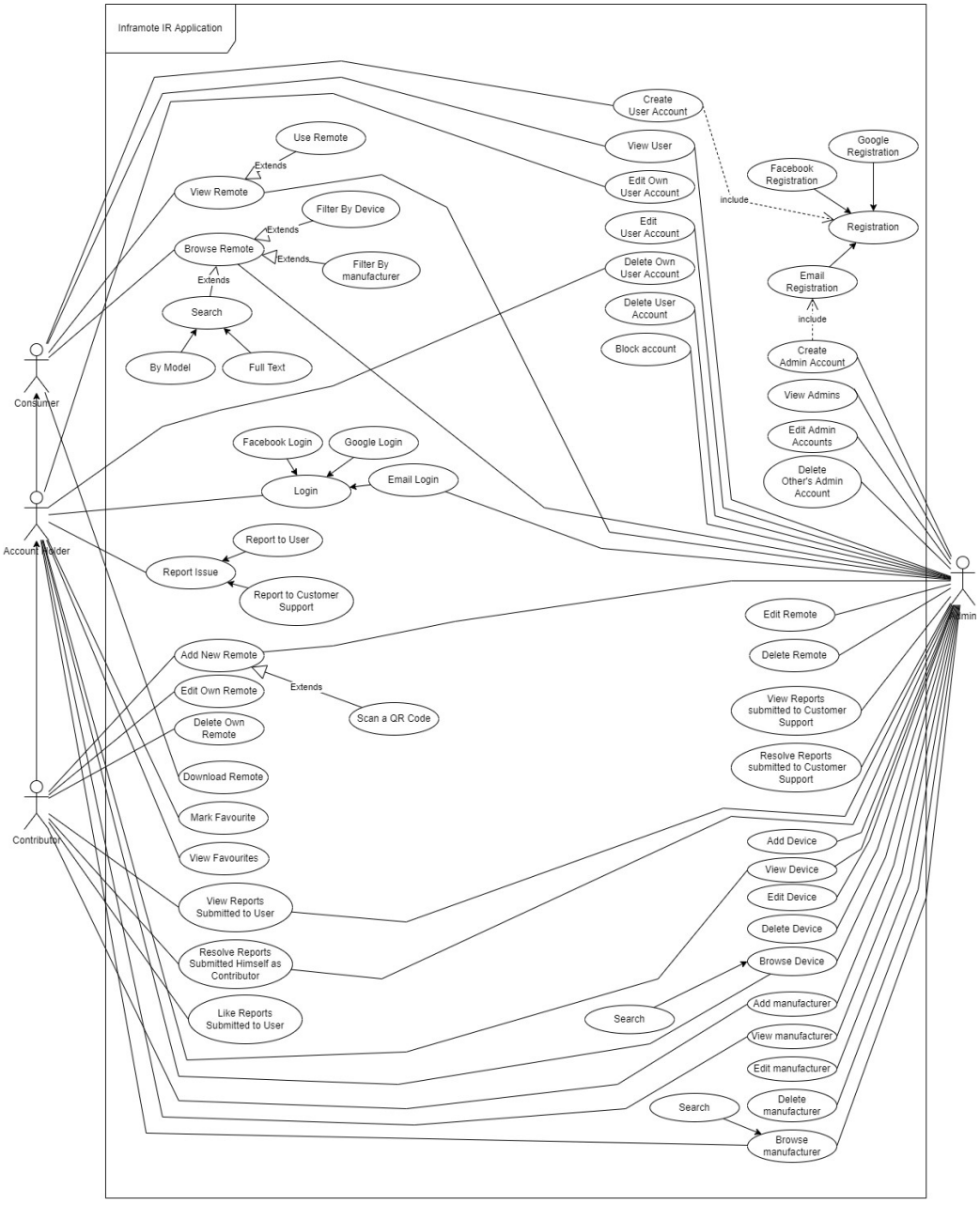
Contributor

By making a contribution to this app, an "account holder" can become a "contributor." A contributor can create, manage, and share their own remotes.

3.2.2 System & Goals

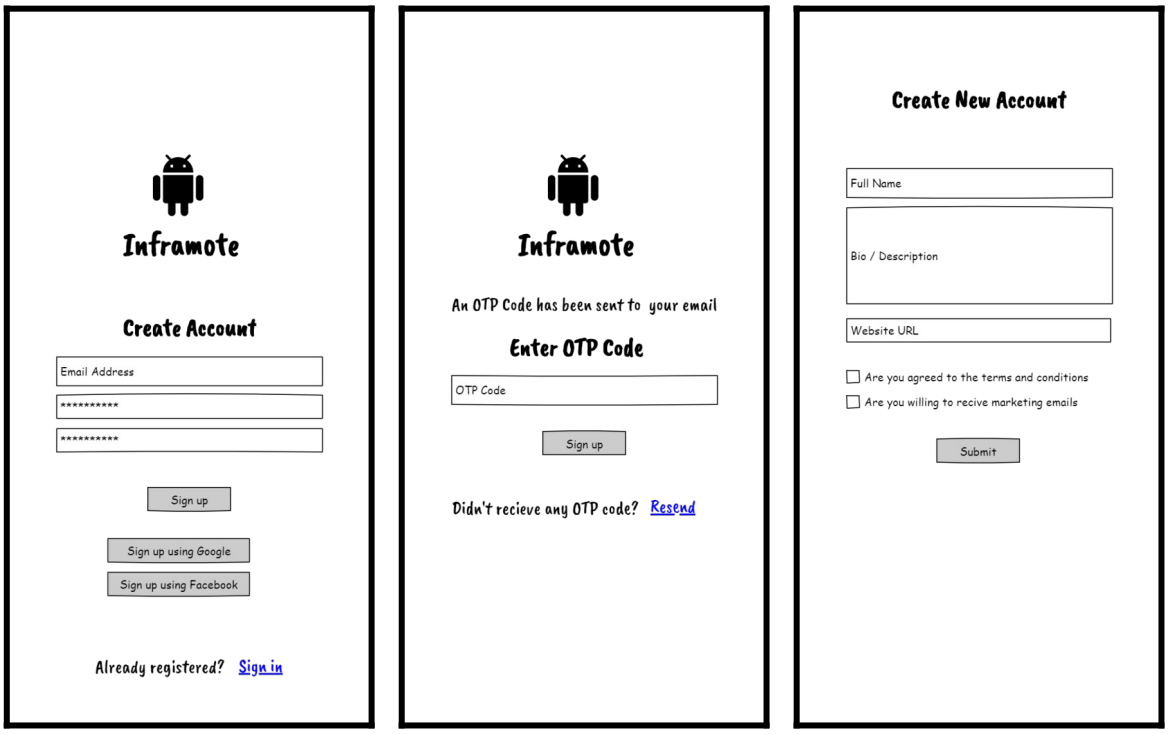
The Inframote Remote Application is the system in this scenario. A rectangle and a title are used to identify the system in use case diagram.

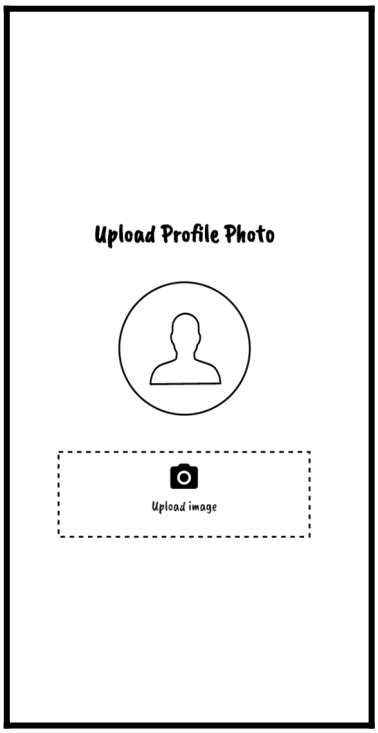
Goals are several tasks that a user can perform by using the application. The goals are represented by a circular box with the goal written inside.

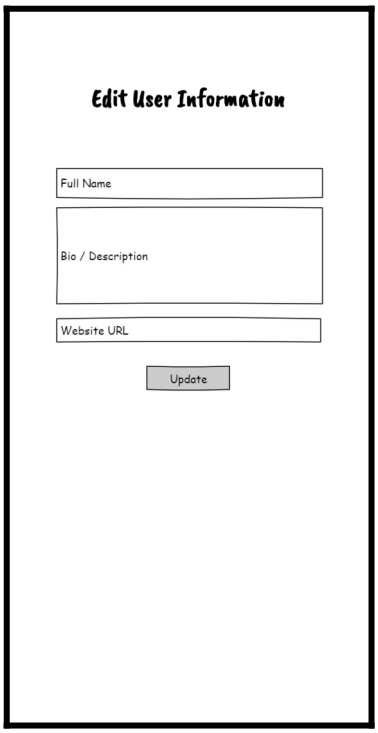


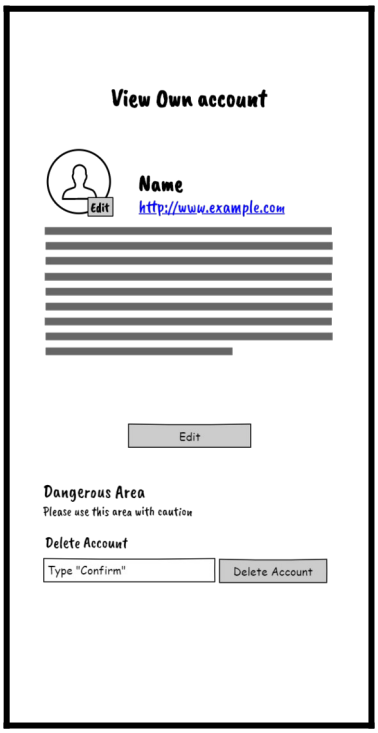
3.3 Wireframe

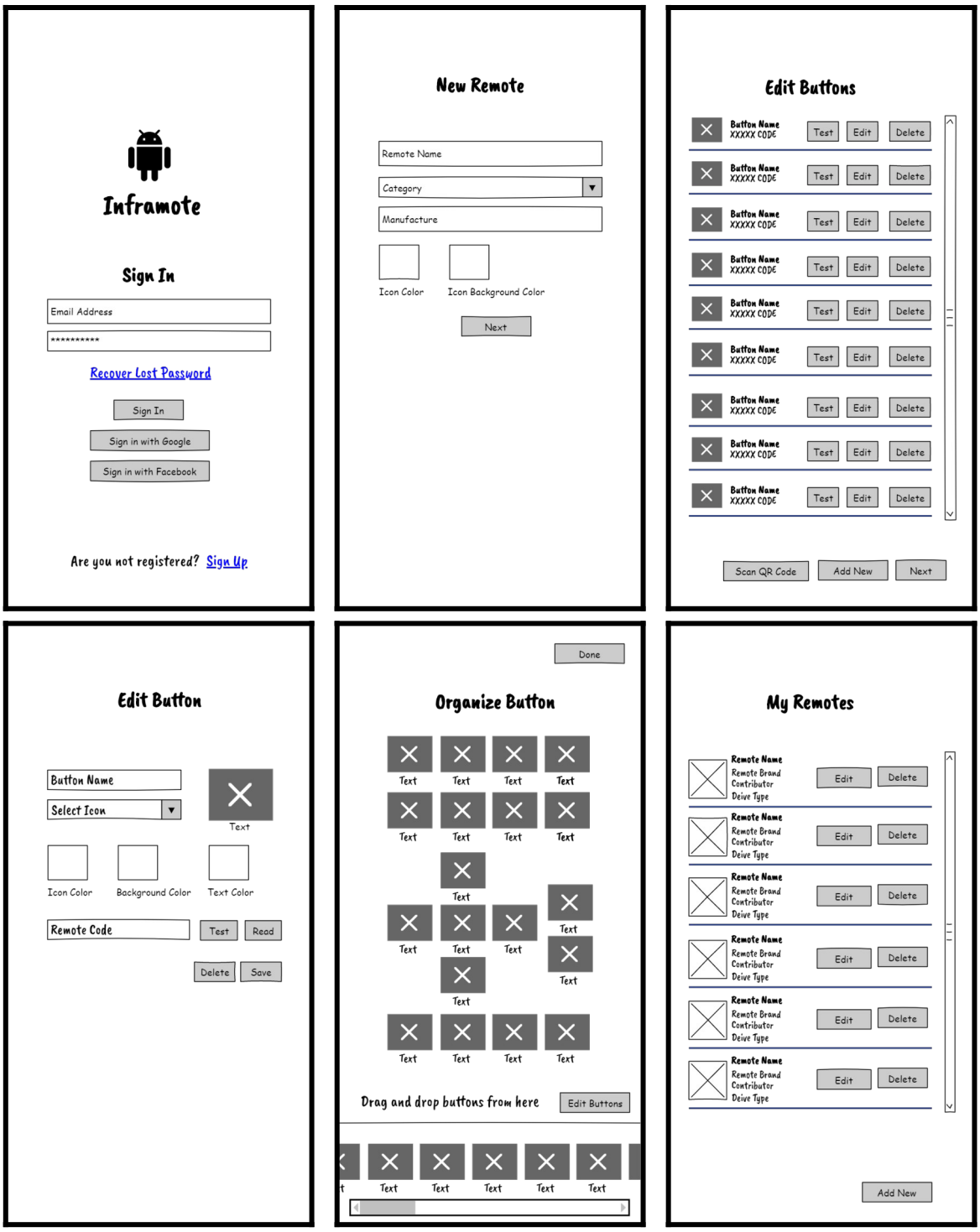
3.3.1 Inframote Android Application Wireframe

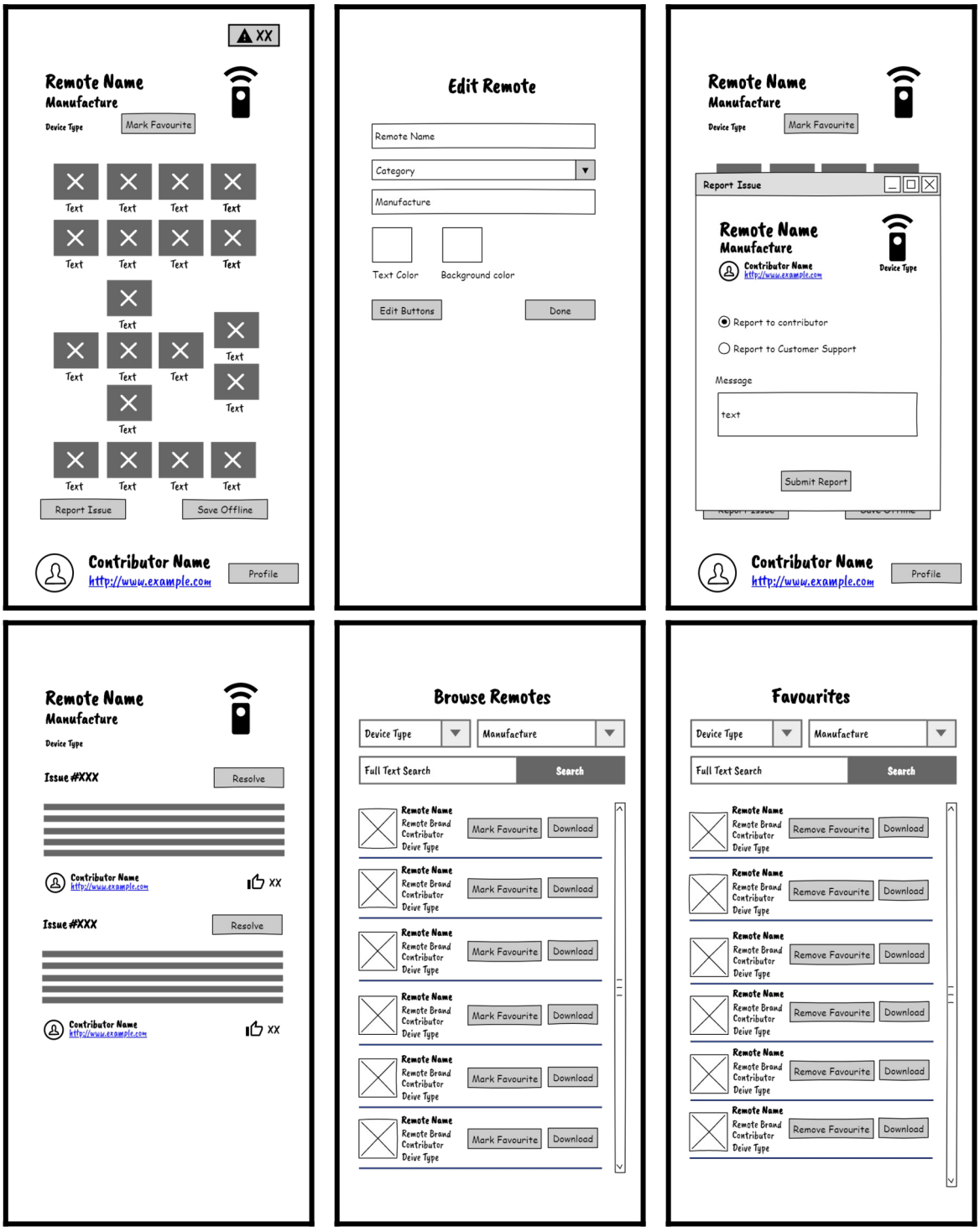


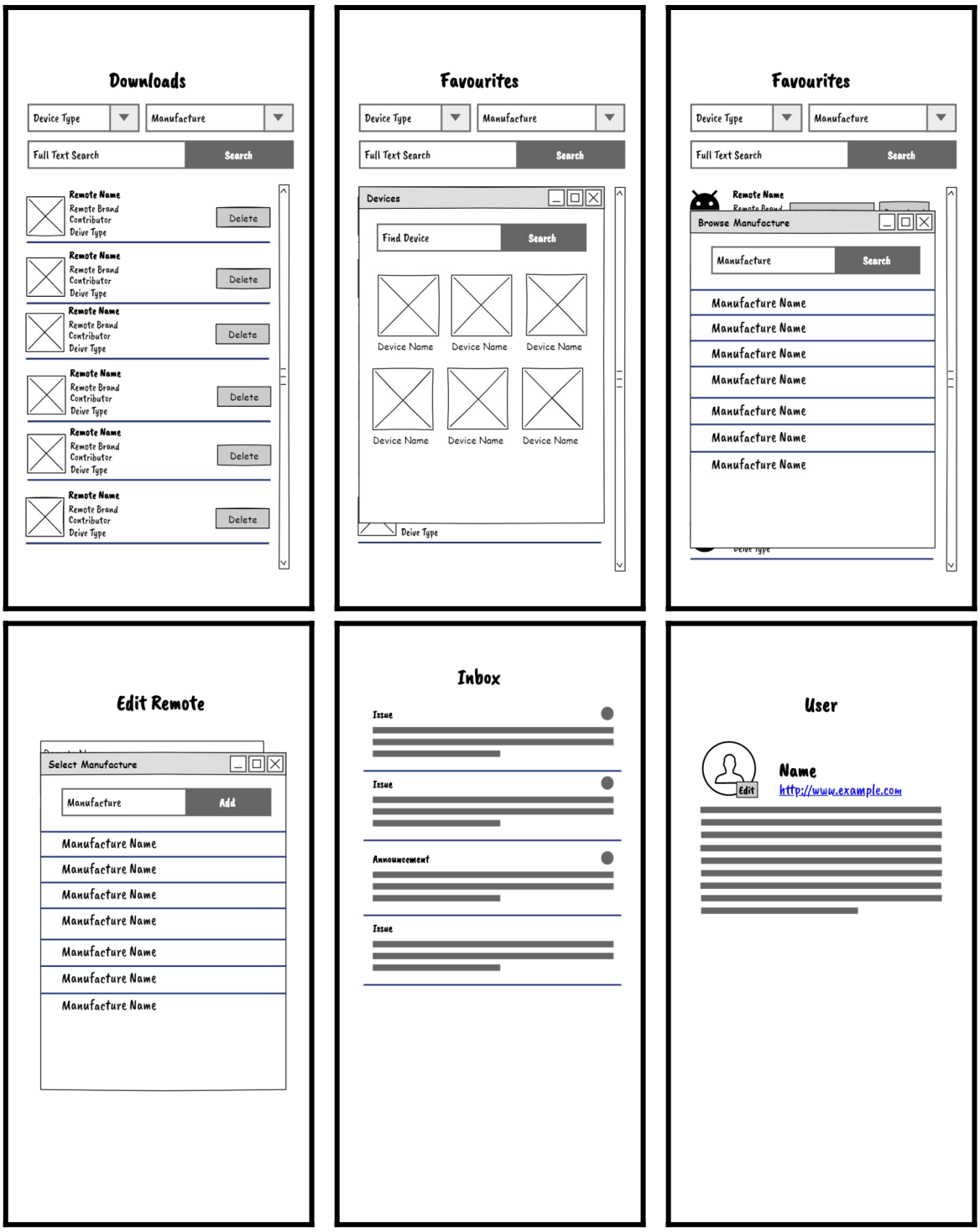

Upload Profile Photo
Upload image


Edit User Information
Full Name
Bio / Description
Website URL
Update


View Own account
Edit
Name
http://www.example.com
Edit
Dangerous Area
Please use this area with caution
Delete Account
Type "Confirm"
Delete Account


Inframote
Sign In
Email Address
****
Recover Lost Password
Sign In
Sign in with Google
Sign in with Facebook
Are you not registered? Sign Up
New Remote
Remote Name
Category
Manufacture
Icon Color
Icon Background Color
Next
Edit Buttons
Button Name
XXXXX CODE
Test
Edit
Delete
Scan QR Code
Add New
Next
Edit Button
Button Name
Select Icon
Text
Icon Color
Background Color
Text Color
Remote Code
Test
Read
Delete
Save
Done
Organize Button
Text
Drag and drop buttons from here
Edit Buttons
My Remotes
Remote Name
Remote Brand
Contributor
Deive Type
Edit
Delete
Add New


XX
Remote Name
Manufacture
Device Type
Mark Favourite
Text
Report Issue
Save Offline
Contributor Name
http://www.example.com
Profile
Edit Remote
Remote Name
Category
Manufacture
Text Color
Background color
Edit Buttons
Done
Report Issue
Report to contributor
Report to Customer Support
Message
text
Submit Report
Issue #XXX
Resolve
Browse Remotes
Device Type
Manufacture
Full Text Search
Search
Remote Brand
Contributor
Deive Type
Download
Favourites
Remove Favourite


Downloads
Device Type
Manufacture
Full Text Search
Search
Remote Name
Remote Brand
Contributor
Deive Type
Delete
Favourites
Devices
Find Device
Device Name
Favourites
Browse Manufacture
Manufacture Name
Edit Remote
Select Manufacture
Add
Inbox
Issue
Announcement
User
Name
http://www.example.com
Edit

3.4 Conclusion

I've documented the Software Requirement Specification for the Inframote IR Remote Application in this chapter. To illustrate the requirements for this application, I used diagrams and wire frames. In the next chapter, I'll go over the system's architecture and design in greater detail.

Chapter 4
Architectural Details and Design

Chapter 4 | Architectural Details and Design

4.1 Introduction

The core architecture and design of the Inframote system will be discussed in this chapter. Later chapters will go over the system in greater depth. This chapter simply covers the major difficulties and challenges, such as database design, information flow, circuit diagrams, and UML class diagrams (with only major classes). Because this chapter is intended to cover only the most important topics, I may not go into great detail on each one.

4.2 Project Layout

This project is divided into 3 major segments.
Database & API: This is the heart of the software, and it's in charge of managing user and remote data. The Serverless RESTFUL API (using AWS lambda) will interface with the database, the Android app, and any third-party services (e.g., Oauth), and will include documentation for developers to create applications that use the API.
Android application (APP): This is the end-user product that allows regular people to access and use the service. For the time being, I will exclusively work on the Android platform due to its popularity, availability, and low development cost.
Electronics & Hardware:
In this phase, I'll develop a hardware prototype for decoding remote IR signals and using the decoded signals as input to our application.

4.3 Database

For this application, the database is in charge of storing all user and dynamic resource data. Because nearly every sort of data in our application is linked to each other in some way, I opted to use a relational database management system (RDBMS) for this project.

Preferred Technology

PostgreSQL
The PostgreSQL database management system is a relational database management system. Aside from being an open-source program with a large community, It has a number of benefits over other RDBMS,

- Supporting additional and more advanced datatypes,
- Supports more features
- Supports complicated querying

4.3.1 Entity Relationship diagram (ERD)

ERD is used to understand and document the relationships between different database entities and their properties. The Entity Relationship Diagram for Inframote Application is given below:

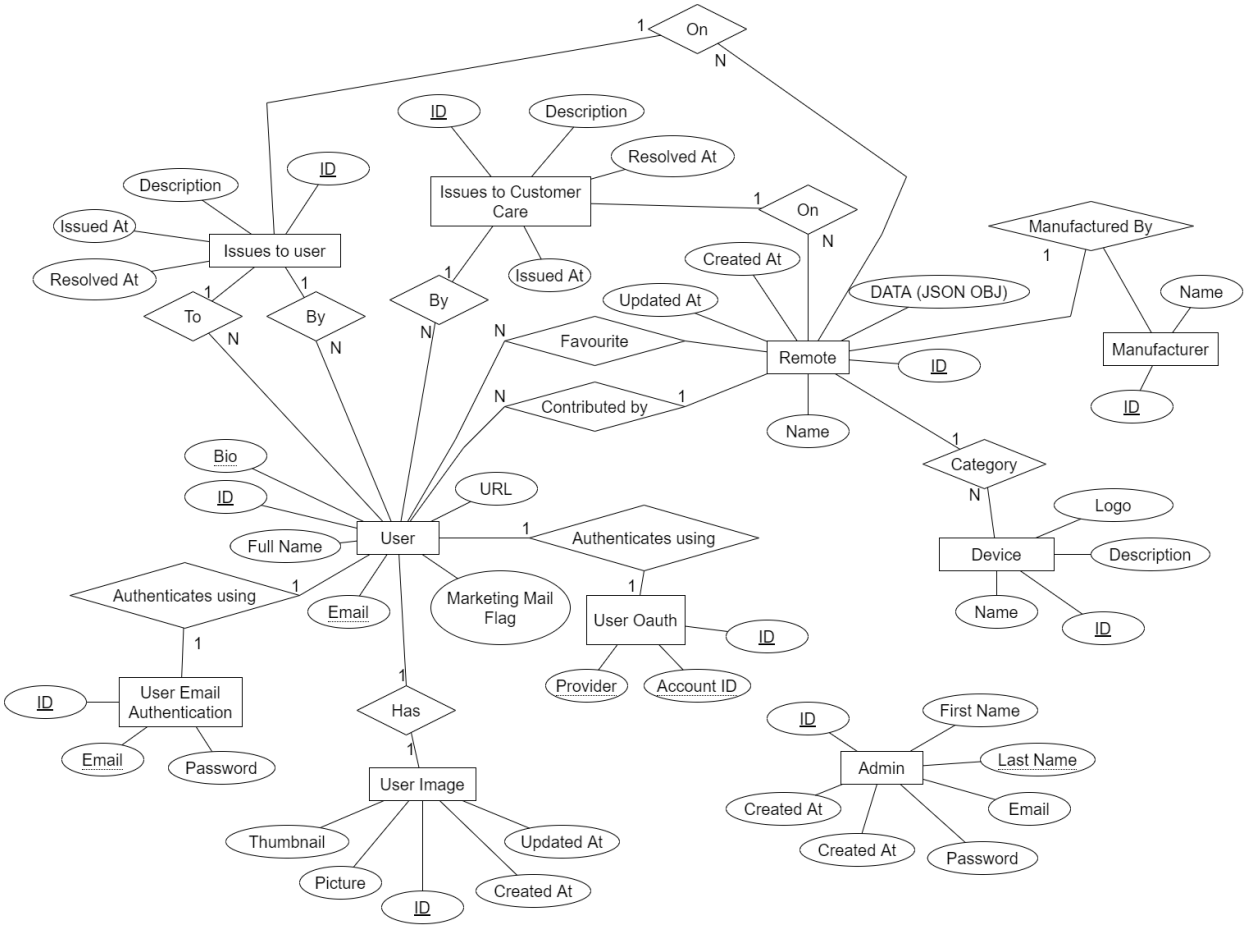


Figure 4.1: Entity Relationship Diagram of Inframote

4.4 Serverless REST API

The API is in charge of facilitating communication between services and end-user devices. The serverless REST API will be developed by taking advantages of AWS Lambda Function. It will work as normal REST API but doesn't require any specific server. AWS Lambda Function uses AWS API Gateway to recive and response request. For this system I am going to use this feature for the following reasons:

- No need to manage dedicated server
- It is Cheap
- It is scalable

The following diagram depicts how the API will work in AWS ecosystem.

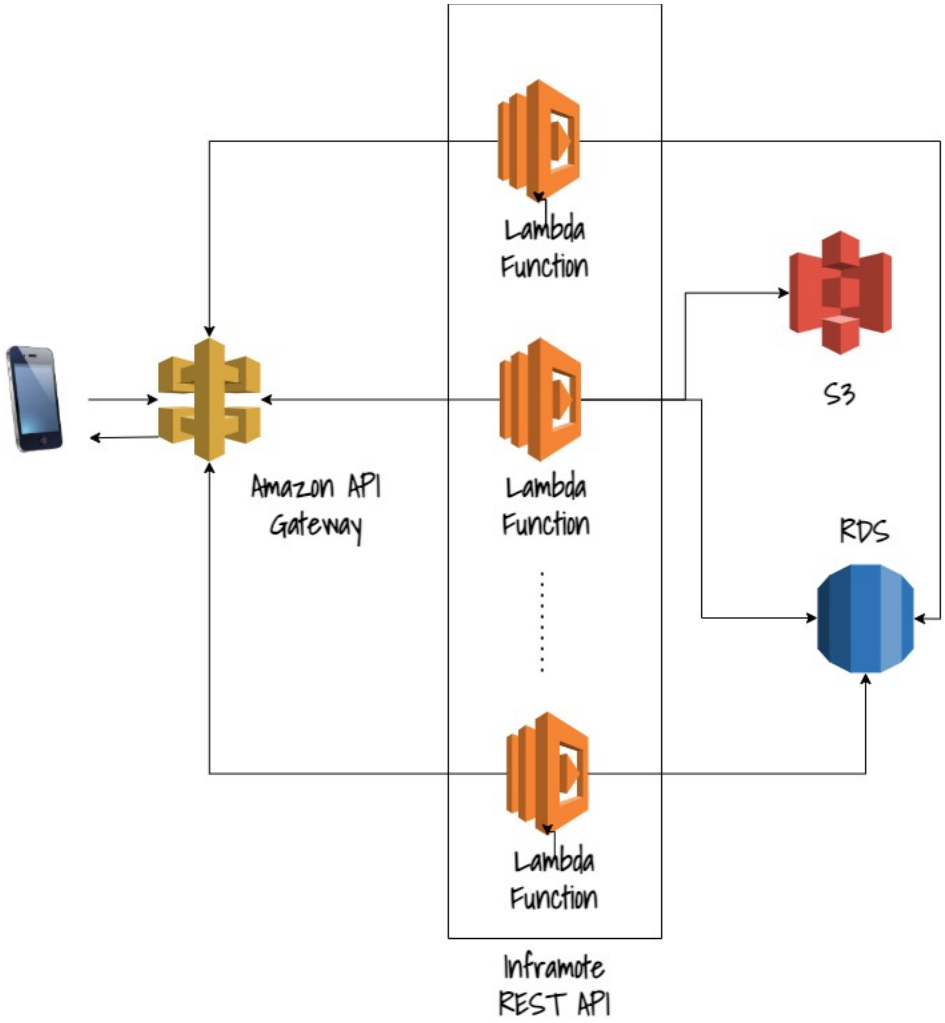


Figure 4.2: Inframote API AWS Diagram

4.5 Android Application

It is the application's front end. This is what the end user will utilize to get the most out of the system. The Android app will be available on the Google Play Store.

The Android application will provide a graphical user interface to the end user. It can work with an external sender and receiver or the Android's built-in IR receiver and sender. It will use the serverless API to communicate with the database.

The following data flow diagram depicts the data flow in the Inframote Application.

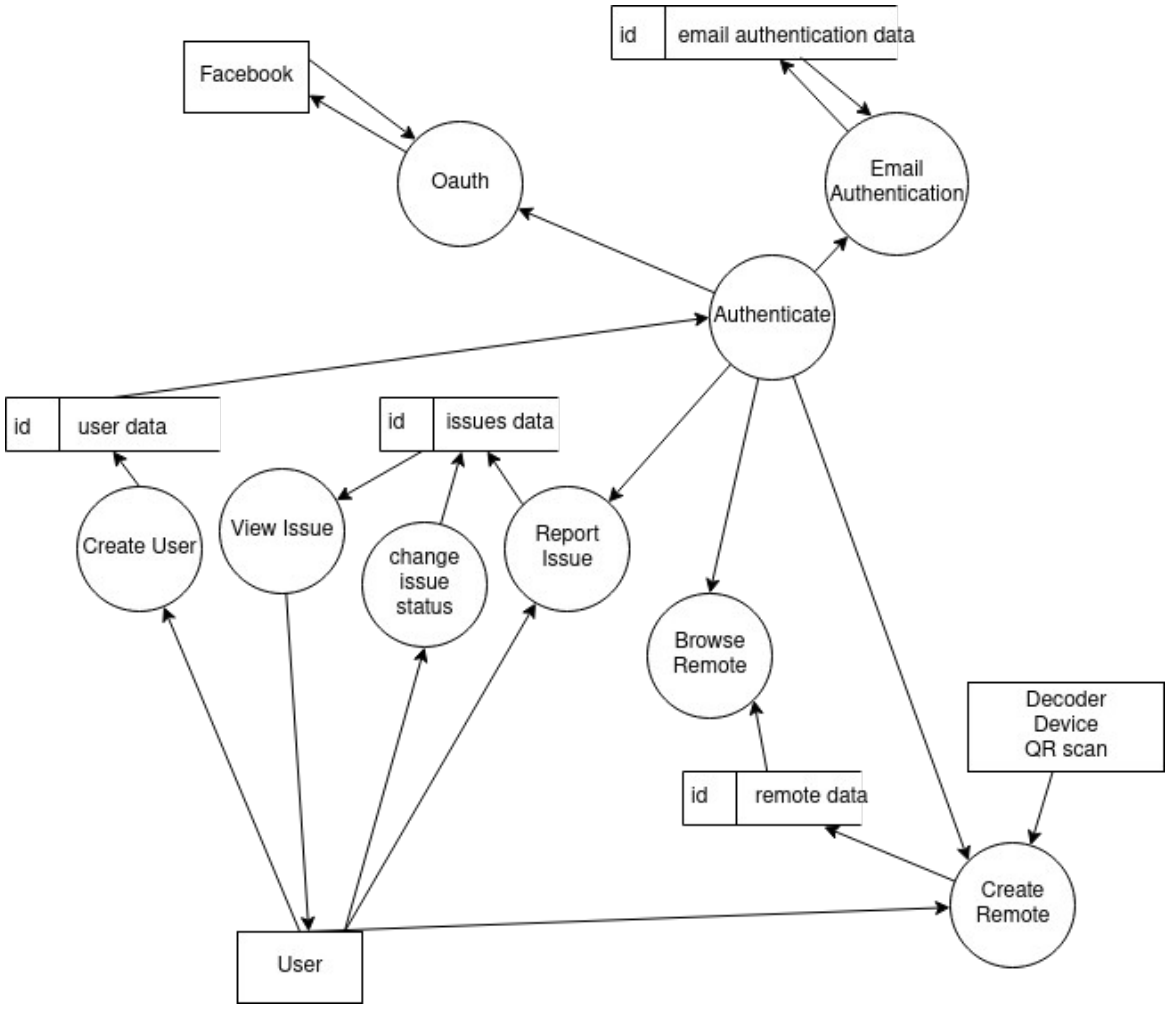
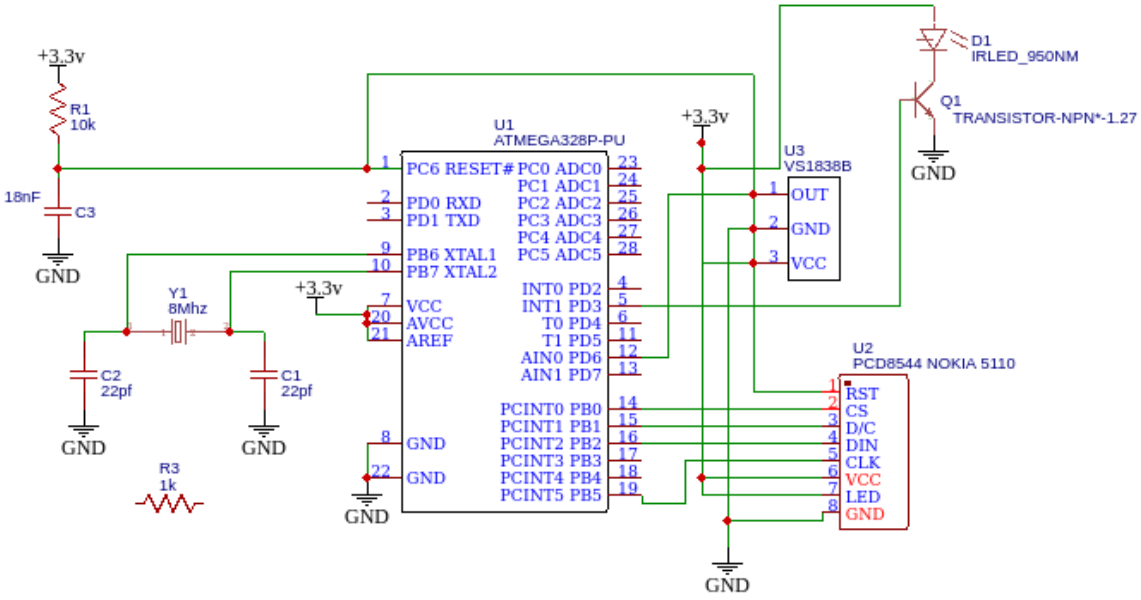


Figure 4.3: Data Flow Diagram of Inframote Application

4.6 Hardware Prototype

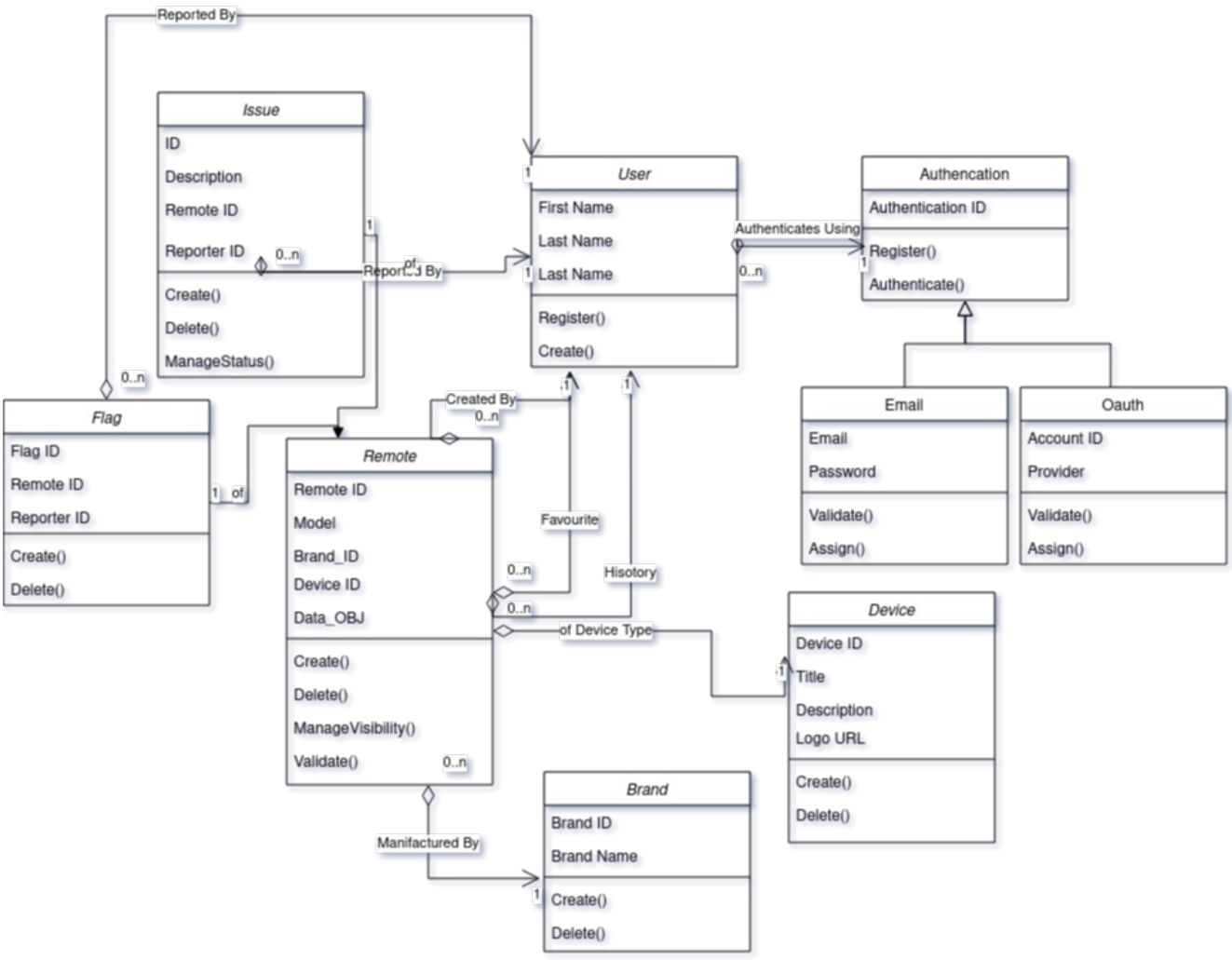
Because most Android phones lack an IR receiver, I'll need to create a hardware prototype that can decode IR signals from a physical remote and utilize the decoded data as input for this application. After that, the decoded data is used to encode and generate IR signals using the Android's built-in IR blaster or external IR blasters.

The circuit design below shows how I'll build the prototype:



4.7 UML Class Diagram

The UML Class diagram is used to visualize an application's object-oriented structure. This application's UML Class diagram is as follows:



4.8 Conclusion

Only the essential design principles of each product segment are covered in this chapter. You will be able to grasp the overall concept and foundation of our project after reading this chapter. In the next chapter, we'll go into the detail of the construction.

References

1. Code Complete 2 by Steve McConnell
2. Clients need and Software Requirement Note, University of Alberta
3. Wikipedia