SOFTWARE DEVELOPMENT PROJECT

WIRELESS RELAY

Oleh :

1. Kevin Christian Mulia (217011667)
2. Patrick Soebiantoro (217011668)

Jl. Ngagel Jaya Tengah No.73-77, Baratajaya, Kec. Gubeng, Kota SBY, Jawa Timur 60284

DAFTAR ISI

Halaman

CHAPTER I
PENDAHULUAN

1.1 Background

An estimation by Energy.gov shows that average households has 5% of their electricity bills made up of light usage. A lot of the time, the lights was not even turned on for a useful purpose, it was just used to light up an empty room. Some may be busy to turn that light switch off, or maybe some did turn it off after a measly 20 minutes after they are done doing what they previously did. However that adds up in a ridiculous pace.

If a household still uses the old 60-watt light bulbs, they could easily waste $900 a year for every 8 hours at night they forget to turn the lights off. The more efficient LED lights may be a lifesaver in that case, but it will still cost them $180. That is all still in the context of light bulbs, and not other humongous electricity bill producer in average households like computers, Wi-Fi, and many more. Turning those other devices off can yield you an extra $200 or even a double of that figure.

WiRe looks toward the case and hope to help everyone by providing a service that helps households to save their money. In addition to economical benefits, the more noble goal of environmental benefits is also achieved. WiRe knew that its adoption will costs the users, but it hopes to provide a return of investment in the near future, and to promote a healthier lifestyle and a healthier mindset about energy usage in the current world.

1.2 Goals

WiRe, on the technical aspects aims for a service that covers:

-Replacing traditional switches with a newer Arduino-based electrical switches and showcase more control, flexibility and further development potential from the IT perspective for the switches.

-Allowing the owner of the said electrical switches to operate the switches as easy as possible from either the palm of their hands or even using other methods like voice commands.

-Providing a great, intuitive interface, using Android phones for the users, both amateur and experts to rely on and operate the said switches.

-Maintaining or even improving the portability and security aspects of traditional switches when implementing the IoT switches.

1.3 Boundaries

A perfect service will be costly, time consuming, and nearly impossible. WiRe understood that and aimed at the fulfillment of a much smaller objective, where the program may not be as good as some people's expectation, however still showing results of the concept introduced, where later, the service can be improved to better match those expectations. To not hinder the current development, in this sections will be defined some of the current limitations of the project, the problems which were in the scope of the development and what are not. Following are some of the problems that the current project will not be accounted for:

1. Devices can be owned only by a single account, even though groups can be shared and operated by multiple accounts.

2. For performance reasons, shared groups may not show their real current state. A user may still operate the said groups, but the results they will be getting will be the same with what is shown in their devices.

3. The main goals of the security aspects are to costs an attacker as much resource as possible in hope to prevent attacks and in the event of a breach, no other services except that of WiRe's will be vulnerable, for WiRe is prone to attacks where an attacker intercepted the data transmission between the client and the server then reverse engineered the service.

4. The quality of some functions like QR code scanner, speech recognition features, and encryption methods, depend on the services provided by their respective external sources; the development of this project only focuses on creating the application for the goals that have been written and not perfecting each of the features used by the developers in creating this project.

1.4 Methodologies

The project's development is divided to 6 steps that needs to be completed in order for a successful development of the service:

1. Analyze the feasibility of the service: economically, technically, and operationally.

2. Designing a service that covers most of the aspects needed for the development that is possible to be made during the time frame.

3. Create web services and prepare the server with necessary server setups and task scheduler for the timer function needed for the project.

4. Prepare the IoT switches, and ensure that they are able to operate with just the server and themselves.

5. Create an interface to allow users to manipulate the database to an extent.

6. Testing the integration and results of the whole project.

BAB II

Base Theory

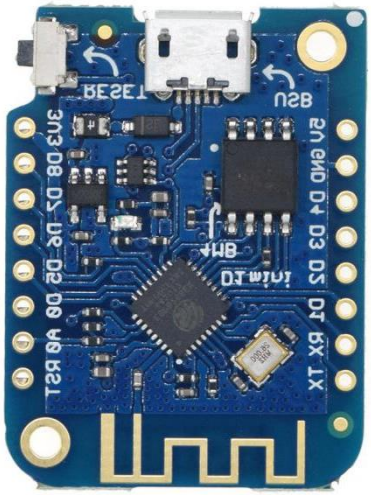
2.1 IoT Device - About Arduino, C++

Arduino is a collection of C or C++ functions that controls a supporting hardware. The Arduino project started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $50, a considerable expense for many students. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas. Casey Reas is known for co-creating, with Ben Fry, the Processing development platform. The project goal was to create simple, low cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forked the project and renamed it Arduino. Those supporting hardware are typically Arduino and similar boards.

Picture 2.1
Arduino Logo

2.2. Wemos D1 Mini

The WeMos D1 Mini is a WiFi-based development board module from the ESP8266 family which can be programmed using the Arduino IDE software as is the case with NodeMCU. The board has a small form factor, with only 34.2mm x 25.66mm size. Wemos D1 Mini operates on 3.3V, and a clock speed of 80 MHz. This board includes a built-in WiFi Module ESP8266 to support WiFi-related operations.



Picture 2.2
Wemos D1 Mini

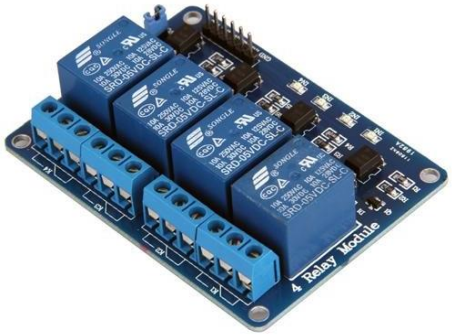
2.3 WiFi Module ESP8266

WiFi **Module ESP8266** is a Wi-Fi module created by Espressif Systems that functions as an addition to a microcontroller like Arduino, allowing the said microcontroller to create a connection through Wi-Fi. ESP8266 is packed with full TCP/IP stack capability. This board uses L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106 Micro running at 80 MHz as its processor. This chip operates with low-power usage and at a pretty extreme temperature range, -40°C to +125°C.

Picture 2.3
WiFi Module ESP8266 Chip

2.4 Relay Switch

A relay is an electrically operated switch. Commonly, relays control one electrical circuit by opening and closing contacts in another circuit. Even though many transfers utilize an electromagnet to precisely operate a switch, however other working standards are likewise utilized, for example, solid-state relays. The advantage of relays is that it takes a relatively small amount of power to operate the relay coil, but the relay itself can be used to control motors, heaters, lamps or AC circuits which themselves can draw a lot more electrical power.



Picture 2.4
Relay Switch

2.5 LED

LED or Light Emitting Diode, is a semiconductor light source that uses electricity, which is then converted to light energy. LED uses a lot less electricity when compared to other light bulbs, therefore making LED very easy to use on Arduino boards. LEDs uses DC, direct current, instead of AC (Alternating current). The LEDs will receive the required electricity from their anode pin and complete the circuit through their cathode pin.

Picture 2.5
LED Light

2.6 Mobile Device - About android and codes used

Android is software for mobile devices that includes operating systems, middleware and key applications. Application development on the Android platform uses the Java programming language. A series of core Android applications include email clients, SMS programs, calendars, maps, browsers, contacts, and others. By providing an open development platform, Android developers offer the ability to build very rich and innovative applications. Developers are free to take advantage of hardware, access location information, run background services, set alarms, add notifications to the status bar, and more. Android relies on the Linux 2.6 version for core system services such as security, memory management, process management, network stack, and driver models. The kernel also acts as an abstraction layer between the hardware and the entire software stack.

Picture 2.6
Android Logo

2.7 Java

Java is one of the programming languages, besides Kotlin that is used to develop Android applications. Java is a multi-platform programming language based on C and C ++ with the principle of "write once, run anywhere". The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets). The latest version is Java 11,

released on September 25, 2018, which follows Java 10 after only six months in line with the new release schedule. Java 8 is still supported but there will be no more security updates for Java 9.

Picture 2.7
Java Programming Language Logo

2.8 JavaScript

JavaScript is a straightforward programming language with which web designers can compose a content and make sites that are intelligent. It was created by Netscape yet is presently utilized by most internet browsers. Since it can communicate with HTML, it permits web engineers to include dynamic components inside the sites. It is an open source programming language, and anybody can utilize it without purchasing a permit for use.

Picture 2.8
JavaScript Programming Language Logo

2.9 PHP

PHP: Hypertext Preprocessor is a server-side scripting language, where the code will be executed on the server. This language is usually used in web development. PHP development began in 1994 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI. PHP will be used to provide services to access some databases or certain functions that can be accessed through a web network. PHP will be run on Apache HTTP Server, to be able to provide a restful web service.

There are two different ways the client can send data to the web server. The GET method sends the encoded user information appended to the uniform resource locator as a part of the page request. On the other hand, the POST method pass on information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING. This allows the information to not be visible in the uniform resource locator.

Picture 2.9
PHP Programming Language Logo

2.10 MySQL

MySQL is one of the most used open source database, enabling the efficient delivery of reliable, powerful and scalable applications. MySQL was created by a Swedish company, MySQL AB, founded by David Axmark, Allan Larsson and Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. MySQL delivers the ease of use, scalability, and high performance in data management.

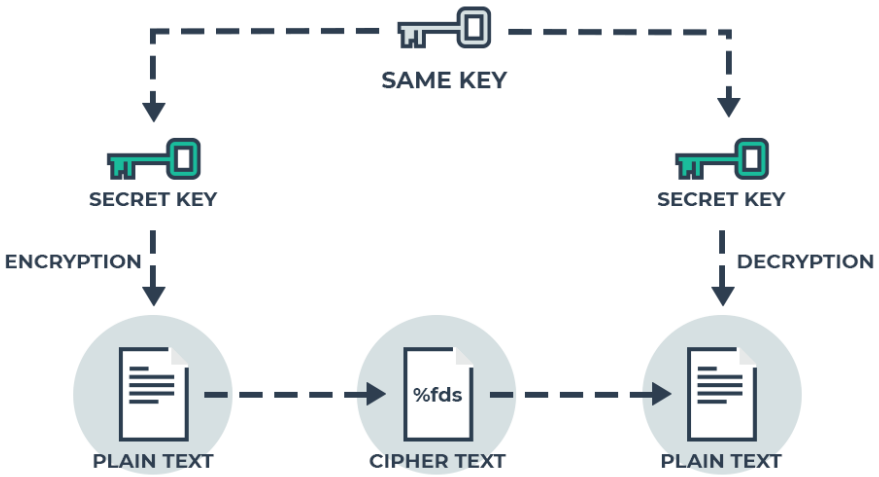
Picture 2.10
MySQL Logo

2.11 Encryption

Encryption in security or cryptography hopes to provide a secure communication between two or more parties. The main methods used in this practice is encryption and hash functions. Encryption in cryptography is the process of encoding or scrambling a message that only the authorized receiver, that knows the key, can decode and know the meaning of the message. However, the security level of an encryption depends on the method of encryption used.

2.11.1 Symmetric Cipher

Symmetric Cipher mainly is an encryption-decryption method that uses a single key for both processes. As the key of encryption and decryption is the same, key used is a shared secret between both parties. There are 2 cipher method in symmetric cipher: the stream ciphers which encrypts the value one by one, and block cipher that encrypts a block or a range of values at a time. The mainly used symmetric cipher is DES and AES, both being block ciphers.



Picture 2.11
Symmetric Encryption Workflow

2.11.2 Simple XOR Cipher

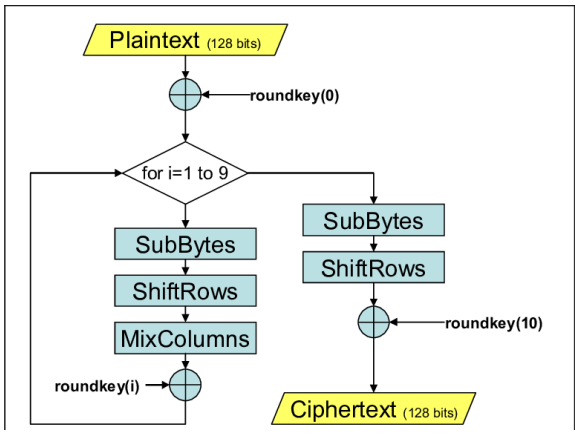
Simple XOR Cipher is an encryption method that uses the fact that a bit can be XORed with another bit of the key and later can be changed back to its real value. This method of encryption is pretty hard to crack, and this cipher as a stream cipher is very fast and light-weight. However, this cipher is susceptible to alteration attacks, which means that a form of verification, such as HMAC, Hash-based Message Authentication Code which utilize hash functions, to be used to prevent Man-In-The-Middle attack.

p	q	$p+q$
0	0	0
0	1	1
1	0	1
1	1	0

Picture 2.12
XOR Table

2.11.3 AES

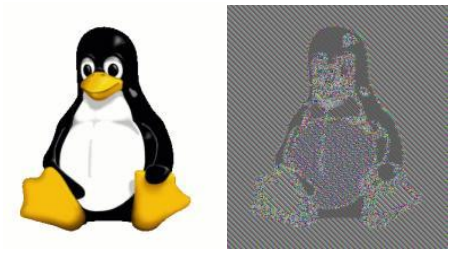
Advanced Encryption Standard, which is originally called Rijndael, is a block cipher that was first published in 1998 by their designers, Vincent Rijmen and Joan Daemen. It is derived from Square, another block cipher invented by the same persons. AES commonly supports the use of 128-bit key with 10 rounds of steps, 192-bit keys with 12 rounds of steps, and 256-bit key with 14 rounds of steps. The steps repeated are SubBytes step, ShiftRows step, mixColumns step and AddRoundKey.



Picture 2.13
AES Algorithm

2.11.4 AES ECB

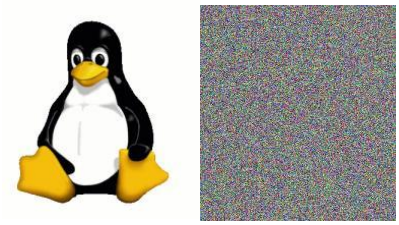
AES ECB, or AES Electronic Codebook mode which is named after phyisical codebooks, is the simplest encryption mode of Rijndael. It simply uses a key and must not be used too much, especially in a very long string of data, because it doesn't hide patterns that well as other modes. Other modes such as CBC creates a more unique pattern. That being said, the implementation of AES ECB is a lot easier because it allows parallel encryption and decryption.



Picture 2.14
Original Data Compared to AES ECB Encrypted Data

2.11.5 AES CBC

AES CBC, or AES Cipher Block Chaining implements the operation of CBC mode that is invented by Ehrsam, Meyer, Smith, and Tuchman in 1976. This mode utilizes initialization vector alongside with the key, which is XORed later and modified with each of XOR process. This processes sure is resulting a more unique patterned message when compared to their ECB counterpart.

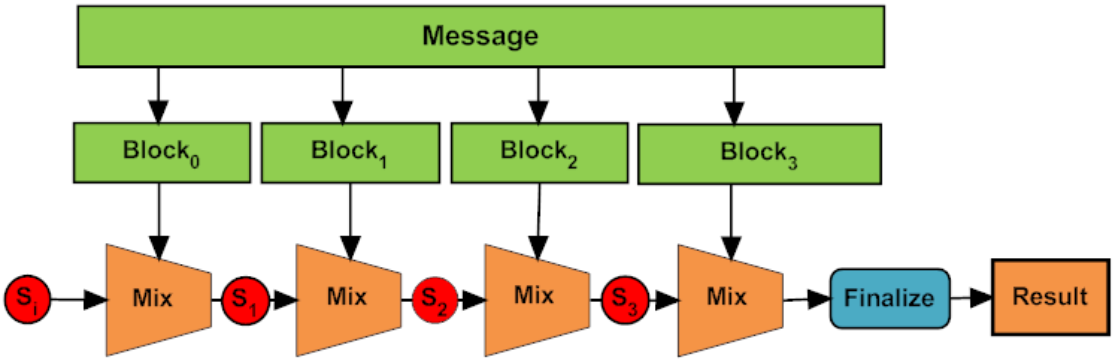


Picture 2.15
Original Data Compared to AES CBC Encrypted Data

2.11.6 Hash Function

Hash function is a function that can be used to scramble data of any size to a data of a fixed size. The values returned often is called hash values, hash codes,

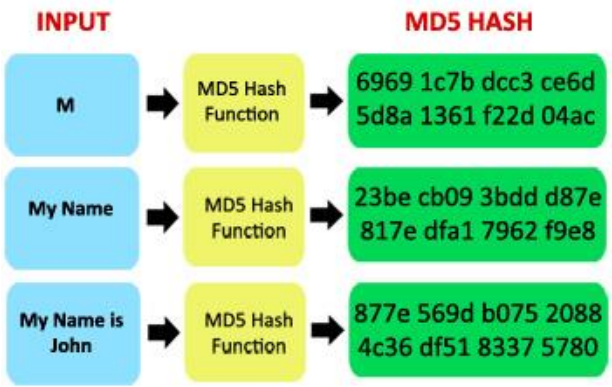
digests, or hashes. The results are irreversible. Hash functions relies simply on its algorithm, and not a key, therefore the results of a hash of the same values are always the same. Hash functions is usually used as a one-way function for a sensitive data where only verification is needed or to check data integrity of a message or file.



Picture 2.17
Hash Function Workflow

2.11.7 MD5

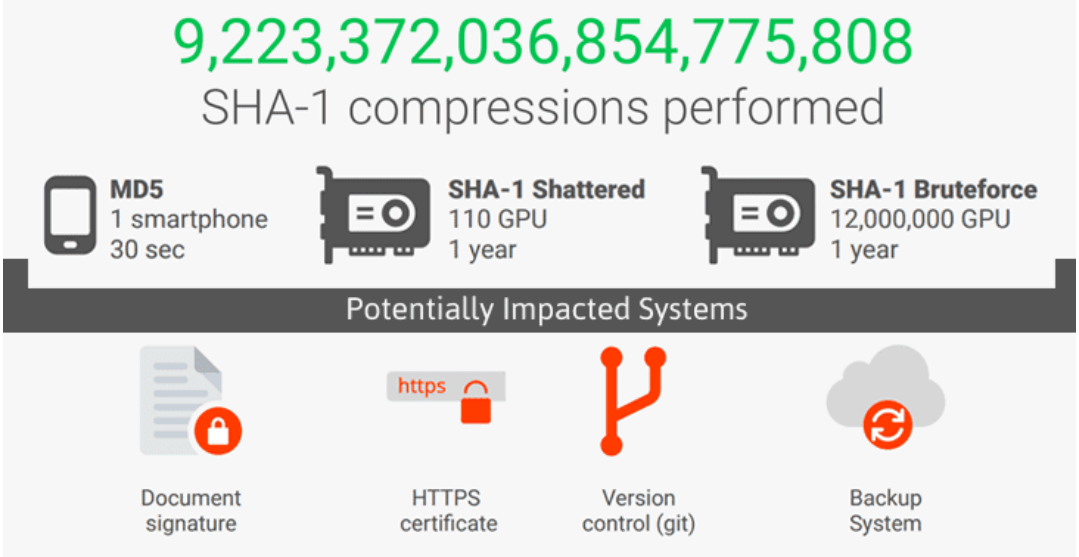
MD5, widely known as message digest 5, is a hash function which creates a 128-bit hash value. It is designed by Ronald Rivest of MIT. It was created in 1991 to replace the likely to be insecure MD4 which its weaknesses was found by Hans Dobbertin later after MD5's invention. MD5 weaknesses is later found by the same Hans Dobbertin, despite that, MD5 is still widely used. MD5 uses 4 rounds of 16 similar operations based on a non-linear function.



Picture 2.18
Few Examples of MD5 Hashes

2.11.8 SHA-2

SHA-2, Secure Hash Algorithm 2, is one of the newer Secure Hash Algorithms standards, which consist of SHA-0, SHA-1, SHA-2 and SHA-3. SHA-2 was designed by United States National Security Agency. SHA-2 supports hash values of six varieties: 224 bits, 256 bits, 384 bits, or 512 bits. SHA uses 64 or 80 rounds of operations. Recently, Google is able to conduct a collision attack, or more widely known as shattered attack which is using SHA-1's flaw, leaving SHA-2 to be a better alternative to use.



Picture 2.19
The Collision Attack that 'SHAttered' the Internet

2.12 Git

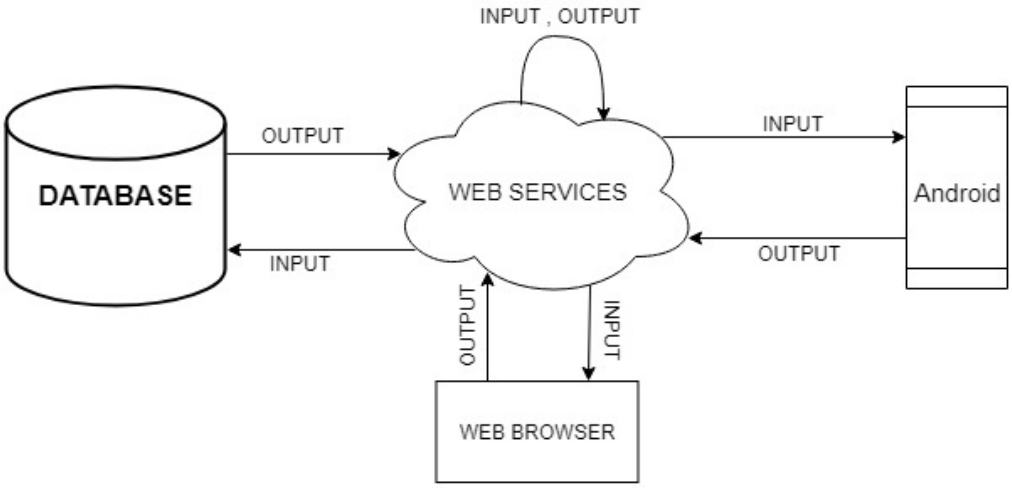
Git is an open source version control system to monitor and integrate software codes. Git monitors changes in a source code during development stage of the software, allowing a coordinated work among the developers, ensuring no code conflicts happens. It supports a faster speed in development, data integrity of the software, distribution, cheap branching, and non-linear, multiple workflows, especially in large-scaled projects projects and to be more precise, complex ones which requires rapid changes to be done. Git outclasses previous SCM tools, for example Subversion and ClearCase because of the results given by Git's features. Git also stores every version of a software pushed to it, meaning Git has a complete collection of a software and every change committed in the range of the entirety of a software's life. Last but not least, Git easy to learn and use.

Picture 2.20
Common Git Commands

CHAPETER III
DESIGN

3.1 Architectural Design

This sub-chapter will explain about the architectural design of this system.



Picture 3.1
Architectural Design System

3.1.1 Web Services

3.1.1.1 From Server

1. admin_checkTimer

This web service work as a timer to the system to check whether a group is needed to be turned on or turned off. Its input are all the group ID that has a timer schedule and the output is a command to on or off.
Input: -
Output: -

2. admin_registerDevice

This web service is used to register new device that have been bought by the user. By scanning the QR Code of the product, it will extract the string of its ID.
Input: device's id.
Output: -

3.1.1.2 From Web Browser

1. confirmAccountPasswordChange

This web service is used to update a new password of a user.
Input: account email and account password.
Output: success of failure of command.

2. confirmAccountRegistration

This web service is used to confirm a new a user's account that has been registered.
Input: confirmation email.
Output: success or failure of command.

3.1.1.3 From IoT Device

1. getDeviceState

This web service is used to know a state of a certain device, whether it's on or off. It also use the XOR and HMAC encryption to secure the device's id.
Input: device's id.
Output: new state of the device.

2. getDeviceWiFiPassword

This web service is used to get the device's WiFi password. It also use the XOR and HMAC encryption to secure the device's id.
Input: device's id.
Output: device owner's WiFi password.

3. getDeviceWiFiSSID

This web service is used to get the device's WiFi password. It also use the XOR and HMAC encryption to secure the device's id.
Input: device's id.
Output: user's WiFi SSID.

3.1.1.4 From Mobile Device

1. deleteDevice

This web service is used to delete the device from all the database.
Input: account's id, account's password, and device's id.
Output: success or failure of command.

2. deleteGrouping

This web service is used to delete a group from the database.
Input: account's id, account's password, and grouping's id.
Output: success or failure of command.

3. deleteGroupingDevice

This web service is used to delete a specific device from a particular group from the database.
Input: account's id, account's password, and grouping's id.
Output: success or failure of command.

4. deleteSharedGrouping

This web service is use to delete the authority of another user that had been shared it's authority to control the group.

Input: account’s id, account’s password, and grouping’s id.
Output: success or failure of command.

5. deleteTimer

This web service is used to delete the timer schedule.
Input: account’s id, account’s password, and timer’s id.
Output: success or failure of command.

6. getAccountId

This web service is used to get the user’s authority and status, whether the account is being used or not. By selecting the user’s email and password as an input for this web service.
Input: account’s email, account’s password.
Output: account’s id, account’s block, account’s use.

7. getAccountWiFiPassword

This web service is used to get the user’s wifi password, by selecting the user’s email and password as an input for this web service.
Input: account’s id, account’s password.
Output: account’s wifi password.

8. getAccountWiFiSSID

This web service is used to get the user’s wifi SSID, by selecting the user’s email and password as an input for this web service.
Input: account’s id, account’s password.
Output: account’s wifi SSID

9. getDeviceOwner

This web service is used to know the owner of the device. This web service has an input which is device_id to obtain the owner information.
Input: device's id.
Output: account’s id.

10. getGroupDevice

This web service is used to get all the devices that are registered in a specific group, that’s why grouping_id is the only input in this service.
Input: grouping’s id.
Output: device’s id, device’s name.

11. getGroupingDevices

This web service is used to get all the devices that are registered in a specific group, that’s why grouping_id is the only input in this service.
Input: grouping’s id.
Output: device’s id, device’s name.

12. getGroupingDeviceChoice

This web service is used to select all device that is available to be added to a new group.
Input : account's id and grouping's id
Output : an array of device's id, name and state

13. getGroupingState

This web service is used to get the state of the group, is it in "ON" state or "OFF" state.
Input: grouping's id.
Output: group's states.

14. getOnDevicesPercentage

This web service is used to know how many percent of the device owned by the user is on right now.
Input : account's id
Output : a percentage number

15. getOwnedDevices

This web service is used to get all the devices that owned by the user. By selecting all the devices in the database where the account's id work as the parameter.
Input: account's id.
Output: device's id, device's name, device's state.

16. getOwnedDevicesState

This web service is used to get all the device's states. Account's id is used to select all the device from the device table that is owned by the user.
Input: account's id.
Output: device's state.

17. getOwnedGroupings

This web service is used to get all the group that is owned by user. Account's id is used to select all the group that had been created by the user.
Input: account_id, account_password, and grouping_id.
Output: 1.

18. getOwnedTimers

This web service is used to get all the timer schedule that is set by user. Account's id is used to select all the timer that had been created by the user.
Input: account's id.
Output: timer's id, timer's name.

19. getSharedGroupings

This web service is used to get all the group that had been shared to the user.
Input: account's id.

Output: grouping’s id, grouping’s name, authority.

20. registerAccount

This web service is used to register a new user. It’s uniqueness of the username is checked. This web service also equipped with encryption. The confidentiality and integrity of the username and password is handled seriously.
Input: account’s email, account’s password.
Output: success or failure of command.

21. registerDeviceOwnership

This web service is used to update the device ownership, to whom they belong.
Input: account’s id, account’s password, and device’s id.
Output: success or failure of command.

22. registerGrouping

This web service is used to make a new group under the user’s authority.
Input: account’s id, account’s password, and grouping’s name.
Output: success or failure of command.

23. registerGroupingDevice

This web service is used to register all the devices that selected to become a particular group’s member.
Input: account’s id, account’s password, grouping’s id, device’s id.
Output: success or failure of command.

24. registerSharedGrouping

This web service is used to share the authority to control the group by the group’s owner to other user.
Input: account’s id, account’s password, and grouping’s id.
Output: success or failure of command.

25. registerTimer

This web service is used to create a new timer schedule.
Input: account’s id, account’s password, and grouping’s id, timer’s name, timer’s start, timer’s action, timer’s state, timer’s d0, timer’s d1, timer’s d2, timer’s d3, timer’s d4, timer’s d5, timer’s d6.
Output: success or failure of command.

26. removeItemGroup

This web service is used to delete a device from a group, it can only be done by the owner of the group.
Input : grouping’s id and device’s id
Output : success or failure string

27. resetAccountPassword

This web service is used to email the user web page link that contains a form to change the user's password. But before calling the page, first this web service will check whether the user is existed or not.
Input : user's account email
Output : an email

28. setAccountKey

This web service is used to generate new encryption key for user, which is done every time the user logins. The encryption key then will be used for transactions between the web service and the user.
Input: account's id, account's password, and account's key.
Output: generated server's key.

29. setAccountUse

This web service is used to update the status of a user. If it's logged in, then the status will be updated into "1" and when a user log out of the account, the value will be updated into "0".
Input: account's id.
Output: success or failure of command.

30. setDeviceState

This web service is used to update the state of the device. To turn it on or turn off.
Input: account's id, account's password, device's id and device's state.
Output: success or failure of command.

31. setGroupingState

This web service is used to update the state of the device. To turn it on or turn off.
Input: account's id, account's password, grouping's id and device's state.
Output: success or failure of command.

32. setOwnedDevicesState

This web service is used to update the state of the device. To turn it on or turn off.
Input: account's id, account's password, and device's state.
Output: success or failure of command.

33. setTimerState

This web service is used to set the timer's state, timers can only be set by the owner of the timer itself.
Input : account's id and password, timer's id and state
Output : success or failure of command

34. updateAccountPassword

This web service is used to update a new password of a user account.
Input: account's id and account's password.
Output: success or failure of command.

35. updateAccountWiFi

This web service is used to update the new WiFi SSID and new WiFi Password of a user.
Input: account's id, account's password, account's wifi SSID and account's wifi password.
Output: success or failure of command.

36. updateGrouping

This web service is used to update the name of a group.
Input: account's id, account's password, grouping's id and group's name.
Output: success or failure of command.

37. updateTimer

This web service is used to update the timer, including the day, time and name of the schedule.
Input: account's id, account's password, and grouping's id, timer's name, timer's start, timer's action, timer's state, timer's d0, timer's d1, timer's d2, timer's d3, timer's d4, timer's d5, timer's d6.
Output: success or failure of command.

38. loadTimer

This web service will select all device that is available to be registered in a new group.

Input : timer's id

Output : an array of timer

39. onoffdevice

This web service is used to turning the device on or off, referring to it's mac address.

Input : device's mac address and status

Output : success or failure statement

40. logout

This android function is used to logged the user out from the system.

Input : account's id and password

Output : success or failure statement

41. setKey

This android function is used to generate a key for user once they are logged into the system. It's used to communicate with the server.

Input : account's id and password, random string

Output : a generated key

42. setGroupingState

This function is used to change the state of a group that owned by the user or maybe by the other user who's authorized to use or control this group.

Input : account's id and password, grouping's id, device's id and state

Output : success or failure statement

43. sendrecover

This function is used to reset the user's password if they forget their password by inserting user's email.

Input : account's email

Output : success or failure statement

44. login

This function is used to logging in the system. Works the same way like the other login function, this click event is connected to a web service called 'getAccountId.php'.

Input : account's email and password

Output : success or failure statement

45. updateAccountPass

This function will be used for updating user's account password. This function is connected to a web service called 'updateAccountPassword.php.

Input : account's id and password, account's new password

Output : success or failure statement

46. updateAccWifi

This function will be used to update the user's WiFi information that have been written in the database since they registered it. This function is connected to a web service called 'updateAccountWifi.php'.

Input : account's id and password, account's wifi SSID and password

Output : success or failure statement

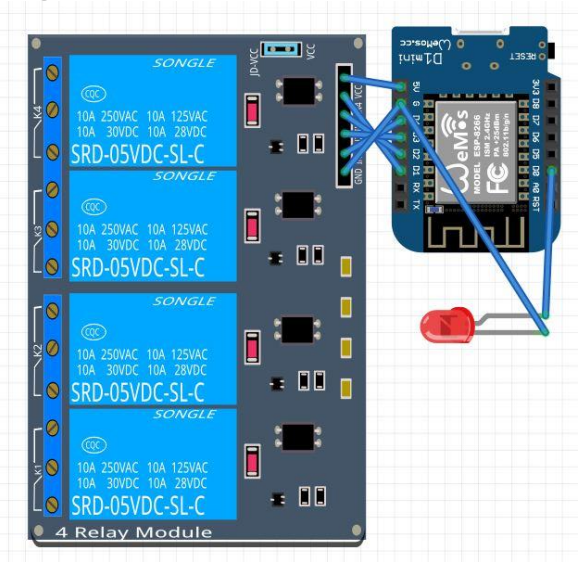
47. setDevice

This function is used to register new device that the user has just bought. By scanning QR code from the device with a device id and name on it.

Input : account's id and password, device's id and password

Output : success or failure statement

3.1.2 IoT Design



Materials :

1. Wemos D1 Mini
   - Pin 5V is connected to Relay's VCC
   - Pin D1 is connected to Relay's IN1
   - Pin D2 is connected to Relay's IN2
   - Pin D3 is connected to Relay's IN3
   - Pin D4 is connected to Relay's IN4
   - Pin G is connected to Relay's GND and LED's cathode
   - Pin D0 is connected to LED's anode
2. Relay Switch 5V
3. LED

3.2 Database Design

Database's Design of Wireless Relay System (WiRe) will be explained in this section. It will define the Entity Relational Diagram, which will show the relation between entities in database, and also explanations for each relation that had been made.

3.2.1 Database Entity

By using ER-Diagram to illustrate the database, it will explain the correlation between each entity in Wireless Relay System (WiRe). All of these entities are contained in the database.

1. ACCOUNT

This entity contain user's information such as e-mail to login into the app. In this entity have 9 fields, there are:

- account_id: Auto-Generated ID given by the system
- account_password: SHA-256 result of salted password.
- account_block: To verify the account whether it is blocked or not
- account_wifi_ssid: User's WiFi name
- account_wifi_password: AES CBC encrypted user's WiFi password.
- account_key: A 16 bit key to be used in some encryptions.
- account_email: AES CBC encrypted user's email to login.
- account_time: Date and time of a member officially registered.
- account_use: Verify whether the account is already logged in or not.

2. DEVICE

This entity keep all the information of all devices that has been registered by the user right before it's installed. This entity has 4 fields, there are:

- device_id : ID of a device that has been registered by users.
- device_state: Showing the device is in "on" or "off".
- device_name: Name of a device.
- account_id: Showing whom is the owner of this device. It's a foreign key from table ACCOUNT.

3. GROUPING

Devices can be gathered and be put inside a group. This table will hold the information of the group itself. It has 3 fields, there are:

- grouping_id: Contain the group's ID.
- grouping_name: Name of a group.
- account_id: Showing whom is the owner of this device. It's a foreign key from table ACCOUNT.

4. AUTHORITY

This entity is used to verify the account user whether he or she is allowed to have control of the group or not. Only 2 fields inside this table:

- grouping_id: Contain the group's ID. It's a foreign key from table GROUPING.
- account_id: Showing whom is the owner of the device. It's a foreign key from table ACCOUNT.

5. MEMBER

Instead of saving the details of a user, it contains information about the members of a group. There are 2 fields in this table:

- grouping_id: Contain the group's ID. It's a foreign key from table GROUPING.
- device_id: ID of a device that has been registered by users and had been assigned to a group.

6. TIMER

It keeps the information of a group where the time has already set. Using a boolean to make the system aware, whether the timer needs is on or off. It has 13 fields, there are:

- timer_id : Contain the timer's ID
- timer_start : Show when the timer is start
- timer_state : Boolean whether the group is on or off
- timer_d0: A boolean that informs the system, does it need to be turned on Sundays.
- timer_d1: A boolean that informs the system, does it need to be turned on Mondays.
- timer_d2: A boolean that informs the system, does it need to be turned on Tuesdays.
- timer_d3: A boolean that informs the system, does it need to be turned on Wednesdays.
- timer_d4: A boolean that informs the system, does it need to be turned on Thursdays.
- timer_d5: A boolean that informs the system, does it need to be turned on Fridays.
- timer_d6: A boolean that informs the system, does it need to be turned on Saturdays.
- grouping_id: Contain the group's ID. It's a foreign key from table GROUPING.
- timer_name: The timer's name.
- timer_action: A command to turned on or to turned off.

7. CONFIRMATION

It is used when a new user is just registered their email. It works temporarily hold the user's email and password until it is confirmed by the user. Only consist of 2 fields, there are:

- confirmation_email: E-mail that had been registered.
- account_pass: Password that had been inputted.

8. TRAFFIC

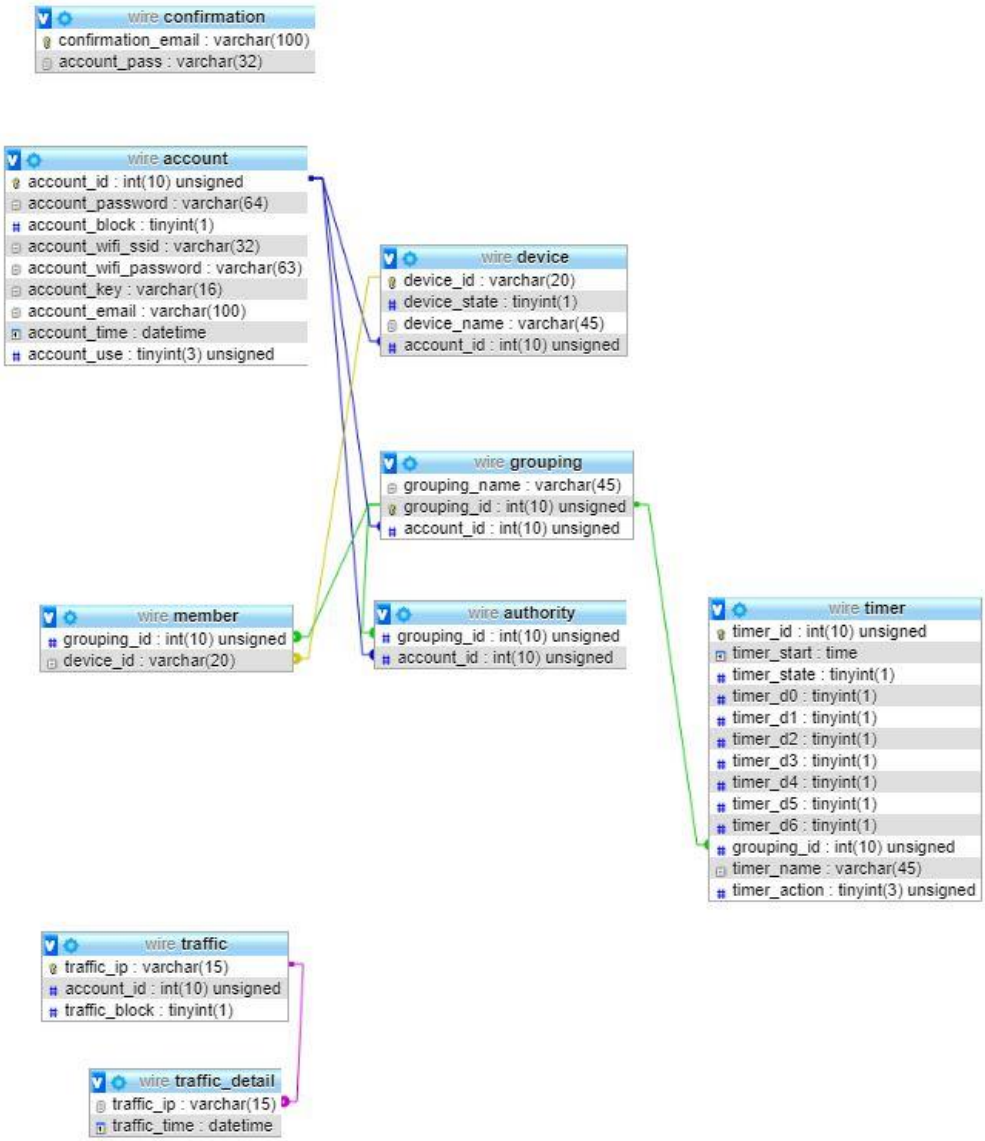
Contains information of user's traffic. It has 3 fields, there are:

- traffic_ip: User's IP Address.
- account_id: The ID that logged in last time.

- traffic_block: Showing that the IP Address is blocked or not.

9. TRAFFIC DETAIL
   Contains the details of the traffic. It has 2 fields, there are:
   - traffic_ip: User's IP Address. It's a foreign key from table TRAFFIC.
   - traffic_time: Date and time of a user's access to database.

3.2.2 ER-Diagram Design



Picture 3.2
Illustration of ER-Diagram

The diagram above illustrates all the database in Wireless Relay Server using PHPMYADMIN. As you can see, there are 3 parts in this database. The first part is a confirmation table. The second part consist of account, member, device, grouping, authority, and timer. All the tables are connected to each other. The third part is traffic and traffic detail.

3.2.3 Table Design

In this sub-chapter contain the description of these tables that is used in Wireless Relay System.
Those tables are:

1. **Confirmation Table**

   Containing the user's email and password that has just being registered. This is the structure of confirmation table:

Table 3.1
Confirmation Table

Field	Type	Length	Constraint	Detail
confirmation_email	varchar	100	primary key	
account_pass	varchar	32		

   This table has 2 fields with confirmation_email as a primary key. Account_pass is the user's password that is registered.

2. **Account Table**

   This table is containing the registered user's data in Wireless Relay System. This is the structure of Account table:

Table 3.2
Account Table

Field	Type	Length	Constraint	Detail
Account_id	Int	10	Primary key	
Account_password	Varchar	64		
Account_block	Tiny	1		
Account_wifi_ssid	Varchar	32		
Account_wifi_password	Varchar	63		
Account_key	Varchar	16		

Table 3.2
Lanjutan

Field	Type	Length	Constraint	Detail
Account_email	Varchar	100		
Account_time	Datetime			
Account_use	Tinyint	3	unsigned	

The account table has 9 fields, with account_id as the primary key. account_password, account_block, account_wifi_ssid, account_wifi_password, account_key, account_email, account_time, and account_use are the information needed to be filled and stored in WiRe system's database.

3. **Device Table**

This table will hold each device's details that has been registered by the users. This is the structure of device table:

Table 3.3
Device Table

Field	Type	Length	Constraint	Detail
Device_id	Varchar	20	Primary key	
Device_state	Tinyint	1		
Device_name	Varchar	45		
Account_id	Int	10	Foreign key	Showing the owner of the device

This table has device_id field as a primary key. The other fields such as device_state, device_name are the details off the device. Account_id is a foreign key from account table is used to the owner of the said device.

4. **Grouping Table**

This table will keep the group's information, groups are formed by several devices. This is the structure of grouping table:

Table 3.4
Grouping Table

Field	Type	Length	Constraint	Detail
Grouping_id	Int	10	Primary key	
Grouping_name	Varchar	45		
Account_id	Int	10	Foreign key	Showing the owner of the group

Each group has an assigned group id, which is used in this table as grouping_id, to be used as the primary key of the table. Grouping_name is used to describe the name of the group. Account_id is a foreign key from account table is used to show who is the owner of the group.

5. **Member Table**

This table is keeping the information of the group's member. This is the structure of member table:

Table 3.5
Member Table

Field	Type	Length	Constraint	Detail
Grouping_id	Int	10	Foreign key	Showing where this device is grouped
Device_id	Varchar	20	Foreign key	Showing the members of the group

No primary key had been set in this table. It has 2 fields, grouping_id and device_id. Both are foreign keys from device table and grouping table.

6. **Authority Table**

This table is used to check the authority of a user to control a group. This is the structure of authority table:

Table 3.6
Authority Table

Field	Type	Length	Constrain	Detail
Grouping_id	int	10	Foreign key	Showing which group is allowed to be controlled
Account_id	Int	10	Foreign key	Showing which user is allowed to control

This table has no primary key, it only have 2 foreign keys. grouping_id and account_id are foreign keys from other table grouping table and account table.

7. **Timer Table**

This table contains data about a group's timer, when it should turned on or turned off. This is the structure of timer table:

Table 3.7
Timer Table

Field	Type	Length	Constraint	Detail
Timer_id	Int	10	Primary key	
Timer_start	Time			When the timer is started

Timer_state	Tinyint	1		Whether this group is on or off
Timer_d0	Tinyint	1		
Timer_d1	Tinyint	1		
Timer_d2	Tinyint	1		
Timer_d3	Tinyint	1		
Timer_d4	Tinyint	1		
Timer_d5	Tinyint	1		
Timer_d6	Tinyint	1		
Grouping_id	Int	10	Foreign key	Showing which group has these timers
Timer_name	Varchar	45		Timer's name
Timer_action	Tinyint	3		Action to execute, turn on or off

This table has the most fields among the other tables. Its primary key is timer_id. It also has a foreign key which is grouping_id to show which group that this timer is applied. The other 11 fields are information that told when to execute and what action that needed to be done.

8. Traffic Table

This table shows data about the user's IP address whether it's been blocked or not. The table also records the last account_id that uses said IP address for additional countermeasures. This is the structure of the traffic table:

Table 3.8
Traffic Table

Field	Type	Length	Constraint	Detail
Traffic_ip	Varchar	15	Primary key	
Account_id	Int	10	Foreign key	
Traffic_block	Tinyint	1		

Traffic_ip is the primary key of this table. Account_id is a foreign key from account table, showing whose IP is this. Traffic_block will indicate wheter the IP is blocked or not.

9. Traffic_detail table

This table will record when the IP is login to use the Wireless Relay System. This is the structure of traffic_detail table:

Table 3.9

Traffic_detail Table

Field	Type	Length	Constrain	Detail
Traffic_ip	Varchar	15	Foreign key	
Traffic_time	Datetime			

It will record the time when the user is accessing server. The user's IP Address and the time both are stored in this table.

3.3 Interface Design

This sub-chapter will describe and explain all the user interface that will appear in the application. Pictures are also included to make the visualization become clear.

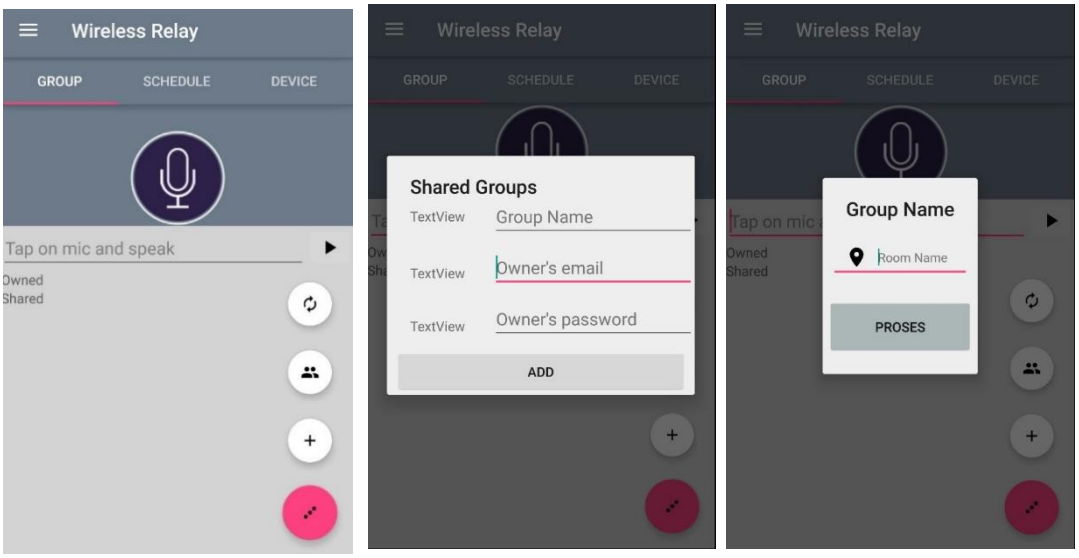
(a) (b)

Picture 3.4
(a) Splash Screen (b) Login Page

When Wireless Relay Apps is opened, this page will be the first one to reveal. Splash screen, after that it will directly headed to login page. Every user has their own user ID and Pasword, to identify that this person is registered or not.

If they forget their password, users must recover their password by click the 'Forget Password?' button and insert their email. WiRe server will directly send a

link to revocer or change user's password. If the user didn't have any account yet, click 'Register' button to register.



(a) (b) (c)

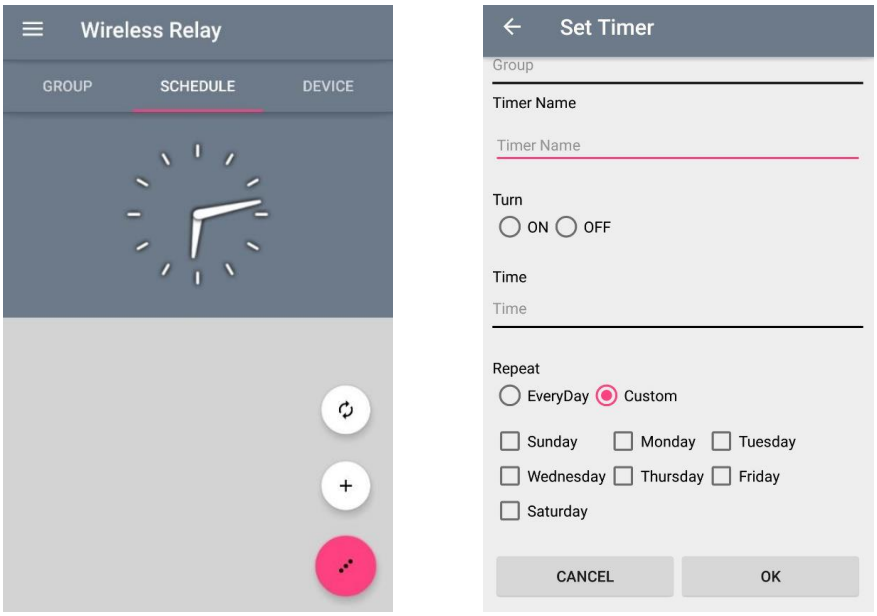
Picture 3.5

(a)Group Page (b)Shared Group Box (c)Add new Group

After the user logged in, the user will be forwarded to Group Page. On this page you can see how many groups do you have. Also groups that users shared to another user.

At the middle of the page, there is an image button with a 'mic' as an icon. When it is pressed, it will recognize your voice and be able to convert it into speech as a command. Press the play button to input the command, the system will try to understand the command and execute what the user's want or desire.

There will be 3 extended buttons, the first one is used to refresh the page. The second one with 'people' icon, this button will show you a box (appear on picture 3.5b) to share a group that owned by the user. The group owner need to select which group that wanted to be shared. User's email that wanted to be shared is need to be input. The last one is the owner's password, this password is used to authorized the process. The third button with '+' icon, is used to add a new group.



(a) (b)

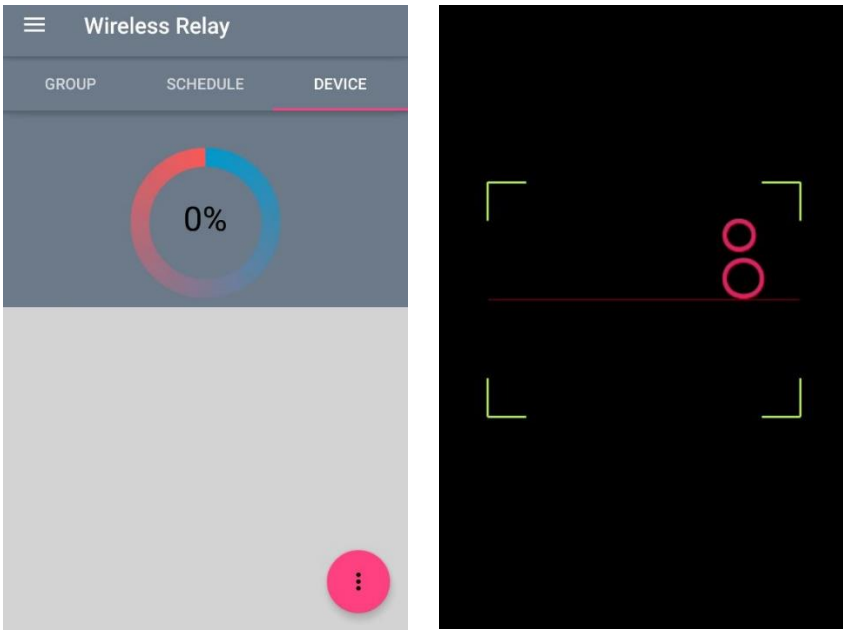
Picture 3.6

(a)Schedule Page (b)Add new Timer

This page contains a timer of a group, when it should be turned on or off. The switch button beside the time, is to trigger the timer on or off. At the middle top of the page, there will be a clock. The time is based on the user location. No matter which region, all the timer will adapt to the time zone where the user is.

A floating button is provided at the bottom right corner. As usual, the first button is used to refresh the whole page. The second button with '+' icon, is used to add a new timer setting of a group.

Referring to picture 3.6b, user need to choose which group that this timer want to be applied. Next, give a name for the timer. After that, choose which command that need to be executed. Input the time when it should be executed. Lastly, user need to select which days they want this timer to be executed.



(a) (b)

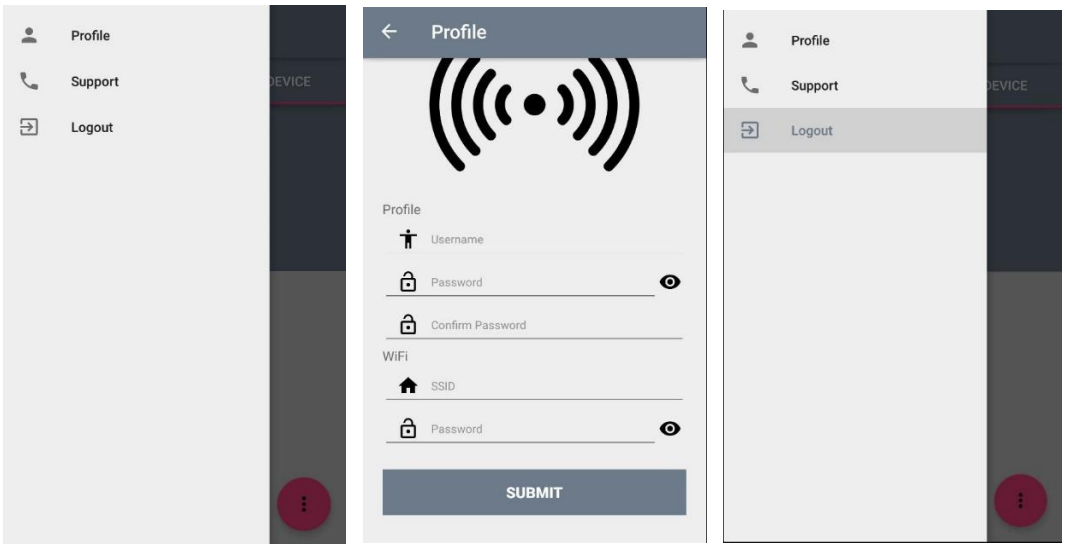
Picture 3.7

(a)Device Page (b)QR Code Scanner

This page will display all the devices that owned by the user. At the middle of the page, there is a percentage of how many devices are on or in use right now.

At the bottom right of this page, a button can be clicked and display two more button. The first one is used to refresh the page and the other one is used to add a new device.

Refers to picture 3.7b, a scanner will appear on the user's mobile phone. It is meant to scan a QR code that stamped on the device. Those QR code carries device's ID and name. Once it is scanned, it will automatically register it to the system, on this user.



(a) (b) (c)

Picture 3.8

(a)Side Bar Menu (b)Profile Page (c)Logout Button

On the top left page, there is a three stripe button, it will display a side bar menu with three buttons which are Profile, Support, and Logout button.

First button which is Profile button will direct the user to a profile page. This page will get all the information of the user including the username, password, WiFi's SSID and password. This profile can be edited by directly change some contents and click the Submit button to save all the changes that have been made.

Second button is Support button, which held all the information that the user need to improve the user's ability to use this application.

The last button will be Logout button, used to logged the users out of the system. After they are logged out, the page will directed to the login page.

BAB IV
IMPLEMENTATION

4.1 WEB SERVICE and FORM

4.1.1 Web Service confirmAccountPasswordChange

This web service is made to confirm when user wanted to change a password. Before the password is changed, it's need to be confirmed first. The program's codes are written below.

Program Code 4.1.1 Web Service confirmAccountPasswordChange

```
1: <?php
2: include '_header.php';
3: $account_email = mysqli_real_escape_string($conn,
   $_POST['account_email']);
4: $account_password = mysqli_real_escape_string($conn,
   $_POST['account_password']);
5: $query = mysqli_query($conn, "SELECT account_email as email
   FROM account;");
6: $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
7: while($exec = mysqli_fetch_array($query)){
8: $real_email = $ecb->decrypt($exec['email']);
9: if(hash("sha256",$GLOBALS['crypto']['header_salt'].$real_email.
   $GLOBALS['crypto']['end_salt'])==$account_email){
10:   $unique_salt = substr($GLOBALS['crypto']['more_salt'],
   strlen($real_email)%strlen($GLOBALS['crypto']['more_salt']));
11:   $password =
   hash("sha256",$GLOBALS['crypto']['header_salt'].$unique_salt.$a
   ccount_password.$real_email.$GLOBALS['crypto']['end_salt']);
12:   $query = mysqli_query($conn, "update account set
   account_password = '".$password."' where account_email =
   '".$exec['email']."';");
13:   echo "Your password has successfully been changed";
14:   include '_footer.php';
15:   }
16:   }
17:   echo $GLOBALS['error']["email_fail"];
18:   include '_footer.php';
```

Line 2 is an include method of another page PHP called header. Line 3 and 4 of the code is about storing the parameter from the Web browser to variables in PHP. At line 5, a query is used to select all email from account table. Line 6 contains a variable to declare the encryption function which is called AES_128_ECB. While line 7 until 16 is used for looping, line 8 is a variable with a user's real email in it. Line 9 to line 15, it has if commands that if the data that fetched from the server is

correspondening with the account_email variable, the password will be encrypted on line 10 until 11, and line 12 is to update the database. Line 13 is to indicate that the changes is successfully overwrite. Line 14 and 18 is an include method of another page of PHP called footer.

4.1.2 Web Service confirmAccountRegistration

This web service is made to confirm a new account that is just registered. By sending email to the user's email, a link is sent as the body of the email. The program's codes are written below.

Program Code 4.1.2 Web Service confirmAccountRegistration

```
1:  <?php
2:  include '_header.php';
3:  $confirmation_email = mysqli_real_escape_string($conn,
    $_GET['confirmation_email']);
4:  $query = mysqli_query($conn, "SELECT confirmation_email as
    email, account_pass as pass FROM confirmation;");
5:  $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
6:  while($exec = mysqli_fetch_array($query)){
7:  $real_email = $ecb->decrypt($exec['email']);
8:  if(hash("sha256",$GLOBALS['crypto']['header_salt'].$real_email
    .$GLOBALS['crypto']['end_salt'])==$confirmation_email){
9:  $moving = mysqli_query($conn, "insert into account
    (account_email, account_pass, account_block, account_key,
    account_time) values
    ('".$exec["email"]."','".$exec["pass"]."', 0, '',now());");
10: $delete = mysqli_query($conn, "delete from confirmation where
    confirmation_email = '".$exec["email"]."';");
11: echo 1;
12: include '_footer.php';
13: }
14: }
15: echo $GLOBALS['error']['registration_fail'];
16: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. Line 4 of the code is about storing the parameter from the Web browser to variables in PHP. At line 5, a query is used to select all email from table confirmation. Line 6 contains a variable to declare the encryption function which is called AES_128_ECB. While line 6 until 14 is used for looping, line 8 is a variable with a user's real email in it. Line 8 to line 13, it has if commands that if the data that fetched from the server is corresponds with the confirmation_email variable or not. At line 9, the account is inserted to the account table and at line 10, the confirmation is deleted from the confirmation table. Line 12 and 16 is an include method of another page of PHP called footer.

4.1.3 Web Service deleteDevice

This web service is made to delete a device that owned by the user. Before directly delete it, a user validation checking will be done first. The program's codes are written below.

Program Code 4.1.3 Web Service deleteDevice

```
1: <?php
2: include '_header.php';
3: $device_id = $_POST['kode'];
4: $result = array();
5: $hapus = mysqli_query($conn, "update device set account_id =
   NULL where device_id = '".$device_id."';");
6: if($hapus == 1){
a. array_push($result, array('status'=> "Item deleted"));
7: }else{
a. array_push($result, array('status'=> "Item delete failed"));
8: }
9: echo json_encode(array("result"=>$result));
10: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 4 is about storing the parameter from the Android device to variables in PHP. Line 4 is creating a new array variable named 'result'. Line 5 is a query to update the account's id from the device. Line 6 to line 8 will push a string showing us whether it's success or not. Line number 9 will send the result back to the android using json_encode. Line 10 is an include method of another page of PHP called footer.

4.1.4 Web Service deleteGrouping

This web service is made to delete a group that owned by the user. Before directly delete it, a user validation checking will be done first. Deleting a group will also delete the group from table timer, authority, member and grouping itself. The program's codes are written below.

Program Code 4.1.4 Web Service deleteGrouping

```
1:<?php
2:include '_header.php';
3:$account_id = mysqli_real_escape_string($conn,
  $_POST['account_id']);
4:$account_password = mysqli_real_escape_string($conn,
  $_POST['account_password']);
5:$grouping_id = mysqli_real_escape_string($conn,
  $_POST['grouping_id']);
6:$iv = mysqli_real_escape_string($conn, $_POST['iv']);
7:$chat_key = ValidateUser($conn, $account_id,
  $account_password);
```

```
8: $cbc = new AES_128_CBC($chat_key, $iv);
9: $grouping_id = $cbc->decrypt($grouping_id);
10:   $query = mysqli_query($conn, "delete from timer where
  grouping_id = ".$grouping_id.";");
11:   $query2 = mysqli_query($conn, "delete from authority where
  grouping_id = ".$grouping_id.";");
12:   $query3 = mysqli_query($conn, "delete from member where
  grouping_id = ".$grouping_id.";");
13:   $query4 = mysqli_query($conn, "delete from grouping where
  grouping_id = ".$grouping_id.";");
14:   echo 1;
15:   include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3until 6 is about storing the parameter from the Android device to variables in PHP. Line 7 is used to check whether the user is valid or not. Line 8 contains a variable to declare the encryption function which is called AES_128_ECB. Line 9 is used to decrypt the grouping's id. Line 10 is used to delete the group from the timer table with group's id as a parameter. Line 11 is used to remove all the authority that can control this group. Line 12 is used to delete all the members inside the group. Line 13 is used to remove the group from the grouping's table. Line 15 is an include method of another page of PHP called footer.

4.1.5 Web Service deleteGroupingDevice

This web service will delete a device from a group. By deleting a device's id and grouping's id from member table. The program's codes are written below.

Program Code 4.1.5 Web Service deleteGroupingDevice

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
6:  $device_id = mysqli_real_escape_string($conn,
    $_POST['device_id']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $device_id = $cbc->decrypt($device_id);
11: $grouping_id = $cbc->decrypt($grouping_id);
12: $query = mysqli_query($conn, "delete from member where
    grouping_id = ".$grouping_id." and device_id =
    '".$device_id."');");
```

```
13:  echo 1;
14:  include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8 is used to check whether the user is valid or not. Line 9 contains a variable to declare the encryption function which is called AES_128_ECB. Line 10 is used to decrypt the device's id. Line 11 is used to decrypt the grouping's id. Line 12 is used to remove a member inside the group. Line 14 is an include method of another page of PHP called footer.

4.1.6 Web Service deleteSharedGroup

This web service removes the authority of another user to control the group. The program's codes are written below.

Program Code 4.1.6 Web Service deleteSharedGroup

```
1: <?php
2: include '_header.php';
3: $account_id = mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: $account_password = mysqli_real_escape_string($conn,
   $_POST['account_password']);
5: $grouping_id = mysqli_real_escape_string($conn,
   $_POST['grouping_id']);
6: $iv = mysqli_real_escape_string($conn, $_POST['iv']);
7: $chat_key = ValidateUser($conn, $account_id,
   $account_password);
8: $cbc = new AES_128_CBC($chat_key, $iv);
9: $grouping_id = $cbc->decrypt($grouping_id);
10:   $query = mysqli_query($conn, "delete from authority where
   account_id = ".$account_id." and grouping_id =
   ".$grouping_id.";");
11:   echo 1;
12:   include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 until 6 is about storing the parameter from the Android device to variables in PHP. Line 7 is used to check whether the user is valid or not. Line 8 contains a variable to declare the encryption function which is called AES_128_ECB. Line 9 is used to decrypt the device's id. Line 10 is used to remove the authority of a user to use or control this group. Line 12 is used to remove a member inside the group. Line 14 is an include method of another page of PHP called footer.

4.1.7 Web Service deleteTimer

This web service will delete the timer.

Program Code 4.1.7 Web Service deleteTimer

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
      $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
      $_POST['account_password']);
5:  $timer_id = mysqli_real_escape_string($conn,
      $_POST['timer_id']);
6:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
7:  $chat_key = ValidateUser($conn, $account_id,
      $account_password);
8:  $cbc = new AES_128_CBC($chat_key, $iv);
9:  $timer_id = $cbc->decrypt($timer_id);
10: $query = mysqli_query($conn, "delete from timer where timer_id
      = ".$timer_id.";");
11: echo 1;
12: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 until 6 is about storing the parameter from the Android device to variables in PHP. Line 7 is used to check whether the user is valid or not. Line 8 contains a variable to declare the encryption function which is called AES_128_ECB. Line 9 is used to decrypt the timer's id. Line 10 is used delete the timer from the database. Line 12 is used to remove a member inside the group. Line 14 is an include method of another page of PHP called footer.

4.1.8 Web Form FormAccountPasswordChange

This web form is made to allow any user to overwrite their own password by filling the new password they've desired.

Based on the Attachment A, it's a web page that contains a form to allow user change their password by entering user's email that is used on this account also the new password that wanted to be set.

4.1.9 Web Service getAccountId

This web service is to get the account's id, also the account's use and block statement. If the account is blocked, then it'll return an error message saying that

this account is blocked. If the account is used already, then it'll return an error message saying that this account is used.

Program Code 4.1.9 Web Service getAccountId

```
1:  <?php
2:  include '_header.php';
3:  $account_email = mysqli_real_escape_string($conn,
      $_POST['account_email']);
4:    $account_password = mysqli_real_escape_string($conn,
      $_POST['account_password']);
5:  $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
6:  $query = mysqli_query($conn, "SELECT account_id as id,
      account_block as block, account_use as used FROM account
      where account_email = '".$ecb->encrypt($account_email)."'
      and account_password = '".$account_password."' limit 1;");
7:  $exec = mysqli_fetch_array($query);
8:  if(strlen($exec['block']) > 0){
9:  if($exec['block']<1){
10: if($exec['used']<1){
11: $query = mysqli_query($conn, "update set account_use = 1 where
      account_id = ".$exec['id'].";");
12: $base_key = substr(md5($account_email),0,16);
13: $iv = substr(md5($account_password),0,16);
14: $cbc = new AES_128_CBC($base_key, $iv);
15: $eid = $cbc->encrypt($exec['id']);
16: echo $eid;
17: }
18: else{
19: echo $GLOBALS['error']['id_use'];
20: }
21: }
22: else{
23: echo $GLOBALS['error']['id_block'];
24: }
25: }
26: else{
27:   echo $GLOBALS['error']['wrong_format'];
28: }
29: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 and 4 is about storing the parameter from the Android device to variables in PHP. Line 5 contains a variable to declare the encryption function which is called AES_128_ECB. Line 6 is used to select the account's id, account's blocked status and account's used status from account table. The parameters that are used to select these 3 columns are the account's email that had been encrypted and the account's password. Line 7 is a variable to store the result of the selection. Line 8 will check whether the account is blocked or not, if it's blocked it will show an error message. Line 9 will check whether the account is blocked or not, if it's blocked it will show an error message. Line 10 will check whether the account is being used or not, to

make it simple is it online or not. If the account is online then it will show an error message also. If it passes all the requirements, on line 11, it will update the account's used status. Line 12 and 13 is used to encrypt the account's email and password. Line 14 contains a variable to declare the encryption function which is called AES_128_ECB. Line 15 is to encrypt the ID. Line 29 is an include method of another page of PHP called footer.

4.1.10 Web Service getAccountWiFiPassword

This web service will get the wifi's password on a certain account.

Program Code 4.1.10 getAccountWiFi Password

```
1: <?php
2: include '_header.php';
3: $account_id = mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: $account_password = mysqli_real_escape_string($conn,
   $_POST['account_password']);
5: $iv = mysqli_real_escape_string($conn, $_POST['iv']);
6: $chat_key = ValidateUser($conn, $account_id,
   $account_password);
7: $cbc = new AES_128_CBC($chat_key, $iv);
8: $query = mysqli_query($conn, "select account_wifi_password as
   password from account where account_id = ".$account_id." limit
   1;");
9: $exec = mysqli_fetch_array($query);
10: $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
11: echo $cbc->encrypt($ecb->decrypt($exec['password']));
12: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 until 5 is about storing the parameter from the Android device to variables in PHP. Line 6 is used to check whether the user is valid or not. Line 7 contains a variable to declare the encryption function which is called AES_128_ECB. Line 8 is used to select the account's WiFi password. Line 9 is a variable that store the result from the selection. Line 10 is used to declare the encryption function which is called AES_128_ECB. Line 11 is used to to encrypt the password after it's decrypted. Line 12 is an include method of another page of PHP called footer.

4.1.11 Web Service getAccountWiFiSSID

This web service will get the wifi's SSID on a certain account.

Program Code 4.1.11 Web Service getAccountWiFiSSID

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
      $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
      $_POST['account_password']);
5:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
6:  $chat_key = ValidateUser($conn, $account_id,
      $account_password);
7:  $cbc = new AES_128_CBC($chat_key, $iv);
8:  $query = mysqli_query($conn, "select account_wifi_ssid as ssid
      from account where account_id = ".$account_id." limit 1;");
9:  $exec = mysqli_fetch_array($query);
10: $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
11: echo $cbc->encrypt($ecb->decrypt($exec['ssid']));
12: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 until 5 is about storing the parameter from the Android device to variables in PHP. Line 6 is used to check whether the user is valid or not. Line 7 contains a variable to declare the encryption function which is called AES_128_ECB. Line 8 is used to select the account's WiFi SSID. Line 9 is a variable that store the result from the selection. Line 10 is used to declare the encryption function which is called AES_128_ECB. Line 11 is used to to encrypt the password after it's decrypted. Line 12 is an include method of another page of PHP called footer.

4.1.12 Web Service getDeviceOwner

This web service is used to know who's the owner of this device by selecting the account's ID from the device that is tagged with this ID.

Program Code 4.1.12 Web Service getDeviceOwner

```
1: <?php
2: include '_header.php';
3: $device_id = mysqli_real_escape_string($conn,
   $_POST['device_id']);
4: $query = mysqli_query($conn, "SELECT account_id as id from
   device where device_id = '".$device_id."';");
5: $exec = mysqli_fetch_array($query);
6: if(!is_null($exec['user'])){
7: echo $exec['user'];
8: }
9: else{
10: echo -1;
11: }
12: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is used to select the account's id from the device. Line 5 is a variable that store the result from the selection. If the user isn't null then it will return the owner's id. If its null, then it will return -1. Line 12 is an include method of another page of PHP called footer.

4.1.13 Web Service getDeviceState

This web service is functioned to get a state of a device.

Program Code 4.1.13 Web Service getDeviceState

```
1:  <?php
2:  include '_header.php';
3:  $device_id = mysqli_real_escape_string($conn,
    XOR_Encrypt(base64_decode($_GET['device_id']),$GLOBALS['crypto
    ']['xor_mac_key']));
4:  $device_pin = mysqli_real_escape_string($conn,
    $_GET['device_pin']);
5:  $hmac_id = substr($device_id, 0, 32);
6:  $device_id = substr($device_id, 32);
7:  if(MD5_HMAC($device_id, $GLOBALS['crypto']['xor_mac_key'],
    $GLOBALS['crypto']['xor_mac_key'])!==$hmac_id){
8:  echo $hmac_id;
9:  include '_footer.php';
10: }
11: $query = mysqli_query($conn, "SELECT device_state as state
    FROM device where device_id = '".$device_id."-".$device_pin."'
    limit 1;");
12: $exec = mysqli_fetch_array($query);
13: if(!is_null($exec['state'])){
14: $state = $exec['state']. RandomString(5);
15: $state = base64_encode(XOR_Encrypt(MD5_HMAC($state,
    $GLOBALS['crypto']['xor_mac_key'],
    $device_id).$state,$device_id));
16: echo $state;
17: }
18: else{
19: echo -1;
20: }
21: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 until line 6 is about storing the parameter from the Android device to variables in PHP. Line 11 is used to get the device's state. Line 12 is a variable that store the

result from the selection. If it's not null then the state will be encrypted otherwise it will echo '-1'. Line 21 is an include method of another page of PHP called footer.

4.1.14 Web Service getDeviceWiFiPassword

This web service will be used to get the device's wifi password.

Program Code 4.1.14 Web Service getDeviceWiFiPassword

```
1:  <?php
2:  include '_header.php';
3:  $device_id = mysqli_real_escape_string($conn,
    XOR_Encrypt(base64_decode($_GET['device_id']),$GLOBALS['crypto
    ']['xor_mac_key']));
4:  $hmac_id = substr($device_id, 0, 32);
5:  $device_id = substr($device_id, 32);
6:  if(MD5_HMAC($device_id, $GLOBALS['crypto']['xor_mac_key'],
    $GLOBALS['crypto']['xor_mac_key'])!=$hmac_id){
7:  echo -1;
8:  include '_footer.php';
9:  }
10: $query = mysqli_query($conn, "select a.account_wifi_ssid as
    ssid, a.account_wifi_password as password from account a,
    device d where a.account_id = d.account_id and d.device_id =
    '".$device_id."-0' limit 1;");
11: $exec = mysqli_fetch_array($query);
12: if(isset($exec['password'])){
13: $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
14: $ssid = $ecb->decrypt($exec['ssid']);
15: $password = $ecb->decrypt($exec['password']);
16: $epassword = base64_encode(XOR_Encrypt(MD5_HMAC($password,
    $GLOBALS['crypto']['xor_mac_key'], $ssid).$password, $ssid));
17:   echo $epassword;
18: }
19: else{
20:   echo -1;
21: }
22: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 until line 6 is about storing the parameter from the Android device to variables in PHP. Line 11 is used to get the device's state. Line 12 is a variable that store the result from the selection. If it's not null then the state will be encrypted otherwise it will echo '-1'. Line 22 is an include method of another page of PHP called footer.

4.1.15 Web Service getDeviceWiFiSSID

This web service will be used to get the device's wifi SSID.

Program Code 4.1.15 Web Service gerDeviceWiFiSSID

```
1:  <?php
2:  include '_header.php';
3:  $device_id = mysqli_real_escape_string($conn,
    XOR_Encrypt(base64_decode($_GET['device_id']),$GLOBALS['crypto
    ']['xor_mac_key']));
4:  $hmac_id = substr($device_id, 0, 32);
5:  $device_id = substr($device_id, 32);
6:  if(MD5_HMAC($device_id, $GLOBALS['crypto']['xor_mac_key'],
    $GLOBALS['crypto']['xor_mac_key'])!=$hmac_id){
7:  echo -1;
8:  include '_footer.php';
9:  }
10: $query = mysqli_query($conn, "select a.account_wifi_ssid as
    ssid from account a, device d where a.account_id =
    d.account_id and d.device_id = '".$device_id."-5' limit 1;");
11: $exec = mysqli_fetch_array($query);
12: if(isset($exec['ssid'])){
13: $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
14: $ssid = $ecb->decrypt($exec['ssid']);
15: $mac = substr($device_id, 0, 17);
16: $ssid_key = generateSSIDKey($mac);
17: $essid = base64_encode(XOR_Encrypt(MD5_HMAC($ssid,
      $GLOBALS['crypto']['xor_mac_key'], $ssid_key).$ssid,
      $ssid_key));
18: echo $essid;
19: }
20: else{
21: echo 0;
22: }
23: include '_footer.php';
24: function generateSSIDKey($mac){
25:     $key = "0";
26:     if(!is_null($mac) && strlen($mac)>16){
27:         $seed = $mac{15};
28:         $key = "";
29:         if(!is_numeric($seed)){
30:             $seed = 0;
31:         }
32:         $key .= substr($mac,$seed+2);
33:         $key .= $mac;
34:         $key = str_replace(":", "", $key);
35:     }
36:     return $key;
37: }
```

Line 2 is an include method of another page PHP called header. At line 3 until line 6 is about storing the parameter from the Android device to variables in PHP. Line 10 is used to get the account's wifi ssid. Line 11 is a variable that store the result from the selection. If it's not null then the state will be encrypted

otherwise it will echo '-1'. Line 22 is an include method of another page of PHP called footer.

4.1.16 Web Service getGroupDevices

This web service is used to get all the member of a group by selecting all the device's id that is included in that group.

Program Code 4.1.16 Web Service getGroupDevices

```
1: <?php
2: include '_header.php';
3: $grouping_id = mysqli_real_escape_string($conn,
   $_POST['grouping_id']);
4: $result = array();
5: $query = mysqli_query($conn, "select a.grouping_id,
   b.device_id ,b.device_name from member a, device b where
   a.device_id = b.device_id and a.grouping_id =
   '".$grouping_id."';");
6: while($row = mysqli_fetch_array($query)){
a. array_push($result, array('grouping_id'=> $row['grouping_id'],
   'device_id'=>$row['device_id'],
   'device_name'=>$row['device_name']));
7: }
8: echo json_encode(array("result"=>$result));
9: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is used to select the group's member. Line 5 is a variable that store the result from the selection. Line 6 is to pass the result using json. Line 7 is an include method of another page of PHP called footer.

4.1.17 Web Service getGroupingDevices

This web service is used to get all the member of a group by selecting all the device's id that is included in that group.

Program Code 4.1.17 Web Service getGroupingDevices

```
1: <?php
2: include '_header.php';
3: $grouping_id = mysqli_real_escape_string($conn,
   $_POST['grouping_id']);
4: $query = mysqli_query($conn, "select d.device_id as id,
   d.device_name as name from device d, member m where d.device_id
   = m.device_id and m.grouping_id = '".$grouping_id."';");
5: $exec = mysqli_fetch_array($query);
```

```
6: echo json_encode($exec);
7: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is used to select the group's member. Line 5 is a variable that store the result from the selection. Line 6 is to pass the result using json. Line 7 is an include method of another page of PHP called footer.

4.1.18 Web Service getGroupingState

Ths web service is made to get the state of a group, whether the group is on or off. The code is given below.

Program Code 4.1.18 Web Service getGroupingState

```
1:  <?php
2:  include '_header.php';
3:  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
4:  $query = mysqli_query($conn, "SELECT count(*)-
    sum(device_state) as off_device FROM device where device_id
    in( select device_id from grouping where grouping_id =
    '".$grouping_id."');");
5:  $exec = mysqli_fetch_array($query);
6:  if(!is_null($exec['off_device'])){
7:  if($exec['off_device']==0){
8:  echo 1;
9:  }
10: else{
11: echo 0;
12: }
13: }
14: else{
15: echo 0;
16: }
17: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is used to select all the device that's on. Line 5 is a variable that store the result from the selection. Line 6 is to check the variable is null or not. Line 7, if the device that's off equals to 0 then return 1, if it's not return 0. Line 17 is an include method of another page of PHP called footer.

4.1.19 Web Service getOnDevicesPercentage

This web service is used to know how many devices are on. Because of the service's name is 'percentage' it will return percentage as the result. So the device that's on divided by all the device owned by this account.

Program Code 4.1.19 Web Service getOnDevicesPercentage

```
1: <?php
2: include '_header.php';
3: $account_id = $_POST['account_id'];
4: $qjml_nyala = mysqli_query($conn, "SELECT count(*) as jml FROM
   device where account_id = '".$account_id."' and device_state =
   '1';");
5: $jmlnyala = mysqli_fetch_array($qjml_nyala)['jml'];
6: $qjml = mysqli_query($conn, "SELECT count(*) as jml FROM device
   where account_id = '".$account_id."';");
7:  $jml = mysqli_fetch_array($qjml)['jml'];
8:  $prosentase = ($jmlnyala / $jml)*100;
9:  $result = array();
10: array_push($result, array('prosentase'=>$prosentase));
11: echo json_encode(array("result"=>$result));
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 and 5 is used to select all the device that it's state is '1' means the device is on. Line 6 and 7 is a query to count every devices owned by the user. Line 8 is a variable declared as '$prosentase' containing the result of the division. Line 9 and 10 will put the result into an array called '$result'. Line 11 is an include method of another page of PHP called footer.

4.1.20 Web Service getOwnedDevices

This web service is made to know every single device that is owned by the user by selecting all device that contains a certain user id.

Program Code 4.1.20 Web Service getOwnedDevices

```
1: <?php
2: include '_header.php';
3: $account_id = mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: result = array();
5: $query = mysqli_query($conn, "select device_id as id,
   device_name as name, device_state as state from device where
   account_id = ".$account_id.";");
6: while($row = mysqli_fetch_array($query)){
7: array_push($result, array('id'=> $row['id'],
   'name'=>$row['name'], 'state'=>$row['state']));
```

```
 8: }
 9: echo json_encodearray("result"=>$result));
10:  include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is used to select all the device that contains the user's id. Line 5 is a variable that store the result from the selection. Line 6 is to pass the result in an array. Line 7 is an include method of another page of PHP called footer.

4.1.21 Web Service getOwnedDevicesState

This web service is made to know the states of every devices that is owned by the user.

Program Code 4.1.21 Web Service getOwnedDevicesState

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $query = mysqli_query($conn, "SELECT count(*)-
    sum(device_state) as off_device FROM device where device_id in
    (select device_id from device where account_id =
    ".$account_id.");");
5:  $exec = mysqli_fetch_array($query);
6:  if(!is_null($exec['state'])){
7:  if($exec['off_device']==0){
8:  echo 1;
9:  }
10: else{
11: echo 0;
12: }
13: }
14: else{
15: echo 0;
16: }
17: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is used to select all the device that contains the user's id. Line 5 is a variable that store the result from the selection. Line 6 is an 'if' command to check if the state is null then it'll return 0. If the state is not null and the off_device is not 0 then it'll return 1, on the other hand it'll return 0 if the off_device is 0. Line 17 is an include method of another page of PHP called footer.

4.1.22 Web Service getOwnedGrouping

This web service will return all the groups that is owned by the user, by selecting all the group's ID that is tagged with this user's ID.

Program Code 4.1.22 getOwnedGrouping

```
1: <?php
2: include '_header.php';
3: $account_id = mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: $result = array();
5: $query = mysqli_query($conn, "select grouping_id as id,
   grouping_name as name from grouping where account_id =
   ".$account_id.";");
6: while($row = mysqli_fetch_array($query)){
7: $query_state = mysqli_query($conn, "select count(*) as jml from
   device where device_state = '1' and device_id in(select
   device_id from member where grouping_id = '".$row['id']."');");
8: $exec_state = mysqli_fetch_array($query_state);
9: if($exec_state['jml'] > 0){
10:         $status = TRUE;
11:   }else{
12:         $status = FALSE;
13:   }
14:   array_push($result, array('id'=> $row['id'],
   'name'=>$row['name'], 'state'=>$status));
15:   }
16:   echo json_encode(array("result"=>$result));
17:   include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is declaring a new array named 'result'. Line 5 is a used to select all grouping's ID and name where the account's ID equals to the user's account ID. Line 6 until line 15 is to count every device that is on in every group. It will return the group's ID, name and the status of the group. Line 16 encode it with json_encode and sent them back to android device. Line 17 is an include method of another page of PHP called footer.

4.1.23 Web Service getOwnedTimers

This web service is made to know all the timer or schedule that had been made before.

Program Code 4.1.23 Web Service getOwnedTimers

```
1: <?php
2: include '_header.php';
3: $account_id = mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: $result = array();
5: $query = mysqli_query($conn, "select t.timer_id as id,
   t.timer_name as name, t.timer_state as state from timer t,
   grouping g where t.grouping_id = g.grouping_id and g.account_id
   = ".$account_id.";");
6: while($row = mysqli_fetch_array($query)){
7:    array_push($result, array('id'=> $row['id'],
   'name'=>$row['name'], 'state'=>$row['state']));
8: }
9: echo json_encode(array("result"=>$result));
10:   include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is declaring a new array named 'result'. Line 5 is a used to select a timer of every group. Line 6 until line 8 is to push the result into an array, contains it's ID, name and state. Line 10 encode it with json_encode and sent them back to android device. Line 17 is an include method of another page of PHP called footer.

4.1.24 Web Service getSharedGroupings

This web service is used to select which group that is shared to this user.

Program Code 4.1.24 Web Service gerSharedGroupings

```
1:<?php
2:include '_header.php';
3:$account_id = mysqli_real_escape_string($conn,
  $_POST['account_id']);
4:$query = mysqli_query($conn, "select g.grouping_id as id,
  g.grouping_name as name, a.account_email as email from grouping
  g, authority a where a.account_id = ".$account_id.";");
5:$ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
6:while($row = mysqli_fetch_array($query)){
7:$query_state = mysqli_query($conn, "SELECT ifnull(count(*)-
  sum(device_state),0) as state FROM device where device_id
  in( select device_id from grouping where grouping_id =
  '".$row['id']."');");
8:  $exec_state = mysqli_fetch_array($query_state);
9:  if($exec_state['state'] > 0){
10:   $status = FALSE;
11: }else{
12:   $status = TRUE;
13: }
```

```
14: array_push($result, array('id'=> $row['id'], 'name'=>
    $row['name'], 'email'=> $ecb->decrypt($row['email']),
    'state'=> $status);
15: }
16: echo json_encode($exec);
17: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. At line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is used to select all the group that is shared to the user by other user. Line 5 a variable to declare the encryption function which is called AES_128_ECB. Line 6 until line 15 is to get the state of the group that had been shared to the user. Also push it into the result array, containing it's ID, name, email, and it's state. Line 16 will encode the result. Line 17 is an include method of another page of PHP called footer.

4.1.25 Web Service registerAccount

This web service is used to register a new user into WiRe's database. A requirement is also implemented such as the user must be unique. After it's successfully registered, the user need to confirm their email.

Program Code 4.1.25 Web Service registerAccount

```
1:  <?php
2:  include '_header.php';
3:  include 'modules/EmailUtils.php';
4:  include 'modules/FormatUtils.php';
5:  $account_email = mysqli_real_escape_string($conn,
    $_POST['account_email']);
6:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
7:  if(checkEmailFormat($account_email)){
8:  $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
9:  $eemail = $ecb->encrypt($account_email);
10: if(isUniqueUser($conn, $eemail)){
11: $query = mysqli_query($conn, "insert into confirmation
    (confirmation_email, account_password) values
    ('".$eemail."','".$account_password."');");
12: $msg = "<p align='center'>Please click this link to confirm
    your registration at WiRe: </p><br>";
13: $url =
    "http://".$ip_wire."/WiRe/confirmAccountRegistration.php?confi
    rmation_email=".hash("sha256",$GLOBALS['crypto']['header_salt'
    ].$account_email.$GLOBALS['crypto']['end_salt']);
14: $img = "<img src =
    'https://i.ytimg.com/vi/V015SjjbYXE/maxresdefault.jpg'>";
15: sendeMail("wire@noreply.com", $account_email, "Confirming WiRe
    account registration", $msg."<a href
    ='".$url."'>".$img."</a>");
```

```
16: echo 1;
17: }
18: else{
19: echo $GLOBALS['error']['email_not_unique'];
20: }
21: }
22: else{
23: echo $GLOBALS['error']['wrong_format'];
24: }
25: include '_footer.php';
26: function isUniqueUser($conn, $user) {
27:     $query = mysqli_query($conn, "SELECT count(*) as users
    FROM account where account_email = '".$user."';");
28:     $exec = mysqli_fetch_array($query);
29:     $total = $exec['users'];
30:     $query = mysqli_query($conn, "SELECT count(*) as users
    FROM confirmation where confirmation_email = '".$user."';");
31:     $exec = mysqli_fetch_array($query);
32:     $total += $exec['users'];
33:     if($total<1){
34:         return true;
35:     }
36:     else{
37:         return false;
38:     }
39: }
```

Line 2 until 4 is an include method of another page PHP called header, EmailUtils, and FormatUtils. Line 10 until line 24 is about checking the user's email format, if the format is wrong then it will return an error message 'wrong format'. After that checking the uniqueness of the email, if it's not unique then it will return an error message 'email not unique'. After all the checking requirements is passed, the email will be inserted into confirmation table and send an email to the new user. Line 25 is about storing the parameter from the Android device to variables in PHP. From line 26 until 39 is a function called 'isUniqueUser', to check whether the email's is already used or not.

4.1.26 Web Service registerDeviceOwnership

This web service will tagged the device with the user's ID, telling that this device is owned by the user

Program Code 4.1.26 Web Service registerDeviceOwnership

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
```

```
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $device_id = mysqli_real_escape_string($conn,
    $_POST['device_id']);
6: $device_name = mysqli_real_escape_string($conn,
   $_POST['device_name']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $device_id = $cbc->decrypt($device_id);
11: $device_name = $cbc->decrypt($device_name);
12: $ecb = new AES_128_ECB($GLOBALS['crypto']['device_aes']);
13: $device_id = $ecb->decrypt($device_id);
14: $query = mysqli_query($conn, "update into device set
    account_id = ".$account_id." where device_id like
    '".$device_id."-%';");
15: echo $query;
16: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. From line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8 is used to validate a user, whether the user is valid or not. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 and 11 is used to decrypt the device's id and name. Line 12 contains a variable to declare the encryption function which is called AES_128_EBC. Line 13 is used to decrypt the device's id. Line 15 is a query to tagged the device with the user's ID. Line 16 is an include method of another page of PHP called footer.

4.1.27 Web Service registerGrouping

This web service is used to register a new group by the user. The group's name must be unique, so there won't be any group that has the same name.

Program Code 4.1.27 Web Service registerGrouping

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $grouping_name = mysqli_real_escape_string($conn,
    $_POST['grouping_name']);
6:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
7:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
8:  $cbc = new AES_128_CBC($chat_key, $iv);
9:  $grouping_name = $cbc->decrypt(grouping_name);
```

```
10: $check =  mysqli_query($conn, "select count(*) as total from
    grouping where account_id=".$account_id." and
    grouping_name='".$grouping_name."');");
11: $result = mysqli_fetch_array($check);
12: if($result['total']<1){
13: $query = mysqli_query($conn, "insert into grouping
    (account_id,grouping_name) values
    (".$account_id.",'".$grouping_name."');");
14: $check =  mysqli_query($conn, "select grouping_id as id from
    grouping where account_id=".$account_id." and
    grouping_name='".$grouping_name."');");
15: $result = mysqli_fetch_array($check);
16: echo $result['id'];
17: }
18: else{
19: echo $GLOBALS['error']["group_duplicate"];
20: }
21: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. From line 3 until 6 is about storing the parameter from the Android device to variables in PHP. Line 7 is used to validate a user, whether the user is valid or not. Line contains a variable to declare the encryption function which is called AES_128_CBC. Line 9 is used to decrypt the grouping's id. Line 10 is to count whether the group's name is used or not. Line 11 is to store the result from the query. From line 12 to 17 if the group's name is not used or not registered, it will be inserted into grouping table and check the group's id. After that return the group's id otherwise return an error message if the name is used or duplicated. Line 21 is an include method of another page of PHP called footer.

4.1.28 Web Service registerGroupingDevice

This web service is used to register all device that wanted to be added in a certain group.

Program Code 4.1.28 Web Service registerGroupingDevice

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
6:  $device_id = mysqli_real_escape_string($conn,
    $_POST['device_id']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
```

```
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $device_id = $cbc->decrypt($device_id);
11: $grouping_id = $cbc->decrypt($grouping_id);
12: $query = mysqli_query($conn, "insert into member (grouping_id,
    device_id) values (".$grouping_id.",'".$device_id."');");
13: echo 1;
14: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. From line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8 is used to validate a user, whether the user is valid or not. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 is used to decrypt the device's id. Line 11 is used to decrypt the grouping's id. Line 12 is to insert into table member with grouping's id and device's id as values. Line 13 will echo 1 which means success to insert the member. Line 14 is an include method of another page of PHP called footer.

4.1.29Web Service registerMember

This web service is used to register all device that wanted to be added in a certain group.

Program Code 4.1.29 Web Service registerMember

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
6:  $device_id = mysqli_real_escape_string($conn,
    $_POST['device_id']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $device_id = $cbc->decrypt($device_id);
11: $grouping_id = $cbc->decrypt($grouping_id);
12: $query = mysqli_query($conn, "insert into member
    (grouping_id, device_id) values
    ('".$grouping_id."','".$device_id."');");
13: echo 1;
14: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. From line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8 is used to validate a user, whether the user is valid or not. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 is used to decrypt the device's id. Line 11 is used to decrypt the grouping's id. Line 12 is to insert into table member with grouping's id and device's id as values. Line 13 will echo 1 which means success to insert the member. Line 14 is an include method of another page of PHP called footer.

4.1.30 Web Service registerSharedGrouping

This web service will allow the owner of a group to share the group to other user. It will add a new authority on the authority table by inserting the other user's ID and the group's ID that wanted to be shared.

Program Code 4.1.30 Web Service registerSharedGrouping

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $grouping_name = mysqli_real_escape_string($conn,
    $_POST['grouping_name']);
6:  $owner_email = mysqli_real_escape_string($conn,
    $_POST['owner_email']);
7:  $owner_password = mysqli_real_escape_string($conn,
    $_POST['owner_password']);
8:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
9:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
10: $cbc = new AES_128_CBC($chat_key, $iv);
11: $grouping_name = $cbc->decrypt($grouping_name);
12: $owner_email = $cbc->decrypt($owner_email);
13: $owner_password = $cbc->decrypt($owner_password);
14: $ecb = new AES_128_CBC($GLOBALS['crypto']['server_aes']);
15: $check = mysqli_query($conn, "select count(*) as count from
    account a, grouping g where a.account_id = g.account_id and
    a.account_email = ".$ecb->encrypt($owner_email)." and
    a.account_password = '".$owner_password."' and g.grouping_name
    = '".$grouping_name."';");
16: $result = mysqli_fetch_array($check);
17: if($result['count']<1){
18: $query_id = mysqli_query($conn, "select grouping_id as id from
    account a, grouping g where g.account_id = ".$account_id." and
    g.grouping_name = '".$grouping_name."';");
19: $result_id =  mysqli_fetch_array($query_id);
20: if(isset($result_id['id'])){
```

```
21: $query = mysqli_query($conn, "insert into authority
    (account_id, grouping_id) values
    (".$account_id.",".$result_id['id'].");");
22: echo 1;
23: }
24: else{
25: echo $GLOBALS['error']["group_fail"];
26: }
27: }
28: else{
29: echo $GLOBALS['error']["group_duplicate"];
30: }
31: include '_footer.php';
```

Line 2 is an include method of another page PHP called header. From line 4 until 8 is about storing the parameter from the Android device to variables in PHP. Line 9 is used to validate a user, whether the user is valid or not. Line 10 contains a variable to declare the encryption function which is called AES_128_CBC. Line 11 is used to decrypt the grouping's id. Line 12 is used to decrypt the owner's email. Line 13 is used to decrypt the owner's password. Line 14 contains a variable to declare the encryption function which is called AES_128_CBC. Line 15 to 17 is to know whether the group is shared or not. Line 18 is used to select the group's id. Line 19 will store the result in a variable. Line 20 and 21, if the id isn't null then it will be inserted into authority table and will echo 1. If it fails, it will return an error message. Line 31 is an include method of another page of PHP called footer.

4.1.31Web Service registerTimer

This web service is used to register a new timer of a group made by the owner of the group.

Based on Attachment B, line 2 is an include method of another page PHP called header. From line 3 until 18 is about storing the parameter from the Android device to variables in PHP. Line 19 is used to validate a user, whether the user is valid or not. Line 20 contains a variable to declare the encryption function which is called AES_128_CBC. Line 21 to 26 is used to decrypt the timer's name, timer's start, timer's action, timer's state. Line 27 to 33 is used to decrypt the timer from day 0 to day 6. Line 34 is a query to insert new timer in timer table. Line 36 is an include method of another page of PHP called footer.

4.1.32 Web Service setAccountKey

This web service is to set the account's key.

Program Code 4.1.32 setAccountKey

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $account_key = mysqli_real_escape_string($conn,
    $_POST['account_key']);
6:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
7:  $dummy_id = mysqli_real_escape_string($conn,
    $_POST['dummy_id']);
8:  $account_key = $account_key; //RSA
9:  $base_key = substr(md5($account_key), 0, 16);
10: $cbc = new AES_128_CBC($base_key, $iv);
11: $dummy_id = $cbc->decrypt($dummy_id);
12: if($dummy_id!=$account_id){
13: echo $GLOBALS['error']["id_fail"];
14: include '_footer.php';
15: }
16: ValidateUser($conn, $account_id, $account_password);
17: echo DiffieHellman_Count($conn, $account_key, $account_id);
18: include '_footer.php';
19: function DiffieHellman_Count($conn, $pub_client, $account_id){
20: $pri_server =
    RandomInt($GLOBALS['crypto']['diffiehellman_length']);
21: $pub_server =
    bcpowmod($GLOBALS['crypto']['diffiehellman_base'],$pri_server,
    $GLOBALS['crypto']['diffiehellman_limit']);
22: $shared =
    bcpowmod($pub_client,$pri_server,$GLOBALS['crypto']['diffiehel
    lman_limit']);
23: while(strlen($shared)<16){
24: $shared = "0".$shared;
25: }
26: Save_Shared($conn, $shared, $account_id);
27: return $pub_server;
28: }
29: function Save_Shared($conn, $key, $account_id){
30: $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
31: $key = $ecb->encrypt($key);
32: $query = mysqli_query($conn, "update account set account_key =
    '".$key."' where account_id = ".$account_id.";");
33: }
```

Line 2 until 4 are included method of another page PHP called header, EmailUtils, and FormatUtils. Line 3 until 7 is about storing the parameter from the Android device to variables in PHP. On line 8, account's id will be encrypted with md5 and the result will be substring from character 0 until 16. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 is used to decrypt dummy id. If dummy id contains the same value with account id then it will give error message and include the footer page. Otherwise it will validate user and then line 17 is an include method of another page of PHP called footer.

4.1.33 Web Service setAccountUse

This web service is to set that this account is already in use right now.

Program Code 4.1.33 setAccountUse

```
1: <?php
2: include '_header.php';
3: $account_id=mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: $query = mysqli_query($conn, "update account set account_use =
   0 where account_id = ".$account_id.";");
5: echo 1;
6: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is a query to update the status of an account is being used or not. Line 5 will return value 1, to indicate if it's success. Line 6 is an include method of another page of PHP called footer.

4.1.34 Web Service setDeviceState

This web service is to set the device's state, whether the user want to turn it on or off.

Program Code 4.1.34 setDeviceState

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $device_id = mysqli_real_escape_string($conn,
    $_POST['device_id']);
6:  $device_state = mysqli_real_escape_string($conn,
    $_POST['device_state']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $device_id = $cbc->decrypt($device_id);
11: $device_state = $cbc->decrypt($device_state);
12: $query = mysqli_query($conn, "update device set device_state =
    ".$device_state." where device_id = '".$device_id."';");
```

```
13: echo 1;
14: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8, a validation of a user will be checked. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 and 11 will decrypt the device's id and state. Line 12 is a query to update the device's state. Line 13 will return value 1, to indicate that the process is done and success. Line 14 is an include method of another page of PHP called footer.

4.1.35 Web Service setGroupingState

This web service is similar to the previous web service, it's used to set the state of the group whether the user want to turn it on or off.

Program Code 4.1.35 setGroupingState

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
6:  $device_state = mysqli_real_escape_string($conn,
    $_POST['device_state']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $grouping_id = $cbc->decrypt(grouping_id);
11: $device_state = $cbc->decrypt($device_state);
12: $query = mysqli_query($conn, "SELECT device_id as id FROM
    device where device_id in (select device_id from member where
    grouping_id = ".$grouping_id.");");
13: $query = mysqli_query($conn, "update device set device_state =
    ".$device_state." where device_id in (select device_id from
    member where grouping_id = ".$grouping_id.");");echo 1;
14: echo $query
15: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8, a validation of a user will be checked. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 and 11 will decrypt

the grouping's name and state. Line 12 is a query to select every device's id that is tagged as the member of this group. Line 13 to 15 will update all the device's state from the id that had been taken from line 12. Line 16 will return value 1, to tell the Android device that the process is done and success. Line 17 is an include method of another page of PHP called footer.

4.1.36 Web Service setOwnedDeviceState

This web service is to set the device's state that is owned by the user only.

Program Code 4.1.36 setOwnedDeviceState

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $device_state = mysqli_real_escape_string($conn,
    $_POST['device_state']);
6:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
7:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
8:  $cbc = new AES_128_CBC($chat_key, $iv);
9:  $device_state = $cbc->decrypt($device_state);
10: $query = mysqli_query($conn, "update device set device_state =
    ".$device_state." where account_id = ".$account_id.";");
11: echo 1;
12: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 until 6 is about storing the parameter from the Android device to variables in PHP. Line 7, a validation of a user will be checked. Line 8 contains a variable to declare the encryption function which is called AES_128_CBC. Line 9 will decrypt the device's state. Line 10 is a query to update the device owned by the user. Line 11 will return value 1, to tell the Android device that the process is done and success. Line 12 is an include method of another page of PHP called footer.

4.1.37 Web Service setTimerState

This web service will set the timer state, whether the user want to use it to turn them on or off.

Program Code 4.1.37 Web Service setTimerState

```
1: <?php
2: include '_header.php';
3: $account_id = mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: $account_password = mysqli_real_escape_string($conn,
   $_POST['account_password']);
5: $timer_id = mysqli_real_escape_string($conn,
   $_POST['timer_id']);
6: $timer_state = mysqli_real_escape_string($conn,
   $_POST['timer_state']);
7: $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8: $chat_key = ValidateUser($conn, $account_id,
   $account_password);
9: $cbc = new AES_128_CBC($chat_key, $iv);
10:   $timer_id = $cbc->decrypt($timer_id);
11:   $timer_state = $cbc->decrypt($timer_state);
12:   $query = mysqli_query($conn, "update timer set timer_state =
   ".$timer_state." where timer_id = '".$timer_id."';");
13:   echo $query;
14:   include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8, a validation of a user will be checked. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 and 11 will decrypt the timer's ID and state. Line 12 is a query to set the timer based on the state given at first. Line 13 will return value 1, to tell the Android device that the process is done and success. Line 14 is an include method of another page of PHP called footer.

4.1.38 Web Service updateAccountPassword

This web service will update the password of a user, as the user asking to change the password.

Program Code 4.1.38 updateAccountPassword

```
1: <?php
2: include '_header.php';
3: $account_id = mysqli_real_escape_string($conn,
   $_POST['account_id']);
4: $account_password = mysqli_real_escape_string($conn,
   $_POST['account_password']);
5: ValidateUser($conn, $account_id, $account_password);
6: $query = mysqli_query($conn, "update into account set
   account_password = '".$account_password."' where account_id =
   ".$account_id.";");
7: $exec = mysqli_fetch_array($query);
8: echo 1;
9: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 and 4 is about storing the parameter from the Android device to variables in PHP. Line 5, a validation of a user will be checked. Line 6 is a query to update the account password with new password that is given by the user. Line 7 and 8 will start updating the password and return value 1, to tell the Android device that the process is done and success. Line 9 is an include method of another page of PHP called footer.

4.1.39 Web Service updateAccountWiFi

This web service used to update the WiFi that is used by this user.

Program Code 4.1.39 updateAccountWiFi

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $account_wifi_ssid = mysqli_real_escape_string($conn,
    $_POST['account_wifi_ssid']);
6:  $account_wifi_password = mysqli_real_escape_string($conn,
    $_POST['account_wifi_password']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $account_wifi_ssid = $cbc->decrypt($account_wifi_ssid);
11: $account_wifi_password =
    $cbc->decrypt($account_wifi_password);
12: $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
13: $query = mysqli_query($conn, "update into account set
    account_wifi_ssid = '".$ecb->encrypt($account_wifi_ssid)."',
    account_wifi_password =
    '".$ecb->encrypt($account_wifi_password)."' where account_id
    = ".$account_id.";");
14: $exec = mysqli_fetch_array($query);
15: echo 1;
16: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8, a validation of a user will be checked. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 and 11 will decrypt

the account wifi's ssid and password. Line 12 contains a variable to declare the encryption function which is called AES_128_EBC. Line 13 to 15 will update all the device's state from the id that had been taken from line 12. Line 16 will return value 1, to tell the Android device that the process is done and success. Line 17 is an include method of another page of PHP called footer.

4.1.40 Web Service updateGrouping

This web service is used to update the group's name, it can only be done by the owner.

Program Code 4.1.40 updateGrouping

```
1:  <?php
2:  include '_header.php';
3:  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4:  $account_password = mysqli_real_escape_string($conn,
    $_POST['account_password']);
5:  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
6:  $grouping_name = mysqli_real_escape_string($conn,
    $_POST['grouping_name']);
7:  $iv = mysqli_real_escape_string($conn, $_POST['iv']);
8:  $chat_key = ValidateUser($conn, $account_id,
    $account_password);
9:  $cbc = new AES_128_CBC($chat_key, $iv);
10: $grouping_id = $cbc->decrypt($grouping_id);
11: $grouping_name = $cbc->decrypt($grouping_name);
12: $query = mysqli_query($conn, "update grouping set
    grouping_name = '".$grouping_name."' where grouping_id =
    ".$grouping_id.";");
13: echo 1;
14: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8, a validation of a user will be checked. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 and 11 will decrypt the grouping's id and name. Line 12 will update the group's name base on the group's id. Line 13 will return value 1, to tell the Android device that the process is done and success. Line 14 is an include method of another page of PHP called footer.

4.1.41 Web Service updateTimer

This web service is used to update the timer's name, time to start, the action, it's state and what days.

Line 2 included method of another page PHP called header. Line 3 until 7 is about storing the parameter from the Android device to variables in PHP. Line 8, a validation of a user will be checked. Line 9 contains a variable to declare the encryption function which is called AES_128_CBC. Line 10 and 11 will decrypt the grouping's id and name. Line 12 will update the group's name base on the group's id. Line 13 will return value 1, to tell the Android device that the process is done and success. Line 14 is an include method of another page of PHP called footer.

4.1.42 Web Service removeItemGroup

This web service is used to delete a device from a group, it can only be done by the owner of the group.

Program Code 4.1.42Web Service removeItemGroup

```
1:  <?php
2:  include '_header.php';
3:  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
4:  $device_id = mysqli_real_escape_string($conn,
    $_POST['device_id']);
5:  $hapus = mysqli_query($conn, "delete from member where
    grouping_id = '".$grouping_id."' and device_id =
    '".$device_id."';");
6:  if($hapus == 1){
7:    echo "Item deleted";
8:  }else{
9:    echo "Item not deleted";
10: }
11: include '_footer.php';
```

Line 2 included method of another page PHP called header. Line 3 and 4 is about storing the parameter from the Android device to variables in PHP. Line 5, is a query to delete a member of a group from the member table. Line 6 to line 10 will conduct whether the query is done successfully or not. Line 11 is an include method of another page of PHP called footer.

4.1.43 Web Service resetAccountPassword

This web service will call the 'FormAccountPasswordChange' which has mentioned above.

Program Code 4.1.43 Web Service resetAccountPassword

```
1:  <?php
2:  include '_header.php';
3:  include 'modules/EmailUtils.php';
4:  include 'modules/FormatUtils.php';
5:  $account_email = mysqli_real_escape_string($conn,
    $_POST['account_email']);
6:  $ecb = new AES_128_ECB($GLOBALS['crypto']['server_aes']);
7:  $query = mysqli_query($conn, "select count(*) as users from
    account where account_email =
    '".$ecb->encrypt($account_email)."';");
8:  $exec = mysqli_fetch_array($query);
9:  if($exec['users']>0){
10: $msg = "<p align='center'>Please click this link to reset your
    password at WiRe: </p><br>";
11: $url =
    "http://".$ip_wire."/WiRe/FormAccountPasswordChange.php?accoun
    t_email=".hash("sha256",$GLOBALS['crypto']['header_salt'].$acc
    ount_email.$GLOBALS['crypto']['end_salt']);
12: $img = "<img src =
    'https://i.ytimg.com/vi/V015SjjbYXE/maxresdefault.jpg'>";
13: $closure = "If you didn't remember applying for a password
    reset, you can leave this email alone.";
14: sendeMail("wire@noreply.com", $account_email, "Reset WiRe
    Password", $msg."<a href ='".$url."'>".$img."</a>".$closure);
15: echo 1;
16: }
17: else{
18: echo $GLOBALS['error']['email_fail'];
19: }
20: include '_footer.php';
```

Line 2 until line 4 is an include method of another page PHP which are header, EmailUtils, FormatUtils. Line 5 is about storing the parameter from the Android device to variables in PHP. Line 6 contains a variable to declare the encryption function which is called AES_128_CBC. Line 7 is a query to check whether the account with this email is exist or not. Line 9 until line 19, if the user is exist, it will call the 'ForAccountPasswordChange' form to change the password otherwise it will return an error message 'error email'. And line 20 is an include method of another page of PHP called footer.

4.1.44 Web Service getGroupingDeviceChoice

This web service will select all device that is available to be registered in a new group.

Program Code 4.1.44 Web Service getGroupingDeviceChoice

```
1.  <?php
2.  include '_header.php';
3.  $account_id = mysqli_real_escape_string($conn,
    $_POST['account_id']);
4.  $grouping_id = mysqli_real_escape_string($conn,
    $_POST['grouping_id']);
5.  $result = array();
6.  $query = mysqli_query($conn, "select device_id as id,
    device_name as name, device_state as state from device where
    account_id = ".$account_id." and device_id not in(select
    device_id from member where grouping_id =
    '.$grouping_id.');");
7.  while($row = mysqli_fetch_array($query)){
8. array_push($result, array('id'=> $row['id'],
   'name'=>$row['name'], 'state'=>$row['state']));
9.  }
10. echo json_encode(array("result"=>$result));
11. include '_footer.php';
```

Line 2 is an include method of another page PHP which are header. Line 3 and 4 is about storing the parameter from the Android device to variables in PHP. Line 5 is declaring a new array variable called ‘result’. Line 6 is a query to select every device that is still available or not a member of another group. Line 7 until 9, all the device will be put inside the result array, contains it’s id, name and state. Line 10 will encode the result to send it back to android device. And line 11 is an include method of another page of PHP called footer.

4.1.45 Web Service loadTimer

This web service will select all device that is available to be registered in a new group.

Program Code 4.1.45 Web Service loadTimer

```
1.  <?php
2.  $conn = mysqli_connect("localhost", "root", "", "wire");
3.  $id = $_POST['kode'];
4.  $result = array();
```

```
5.  $query = mysqli_query($conn, "SELECT a.*, b.grouping_name FROM
    timer a, grouping b where a.grouping_id = b.grouping_id and
    a.timer_id = '".$id."';");
6.  $data = mysqli_fetch_array($query);
7.  array_push($result, array('id'=> $data['timer_id'],
8.  'id_group'=> $data['grouping_id'],
9.  'nm_group'=> $data['grouping_name'],
10. 'nm_timer'=> $data['timer_name'],
11. 'timer_state'=> $data['timer_state'],
12. 'timer_start'=> $data['timer_start'],
13. 'timer_d0'=> $data['timer_d0'],
14. 'timer_d1'=> $data['timer_d1'],
15. 'timer_d2'=> $data['timer_d2'],
16. 'timer_d3'=> $data['timer_d3'],
17. 'timer_d4'=> $data['timer_d4'],
18. 'timer_d5'=> $data['timer_d5'],
19. 'timer_d6'=> $data['timer_d6']
20. ));
21. echo json_encode(array("result"=>$result));
```

Line 2 is an include method of another page PHP which are header. Line 3 is about storing the parameter from the Android device to variables in PHP. Line 4 is declaring a new array variable called ‘result’. Line 5 is a query to select every timer owned by the user. Line 6 until 20, is about storing all the selected timer into result array. Line 21 will encode the result to send it back to android device.

4.1.46 Web Service onoffdevice

This web service is used to turning the device on or off, referring to it’s mac address.

Program Code 4.1.46 Web Service onoffdevice

```
1:  <?php
2:  include '_header.php';
3:  $mac = $_POST['mac'];
4:  $status = $_POST['status'];
5:  $result = array();
6:  $update = mysqli_query($conn, "update device set device_state =
    '".$status."' where device_id = '".$mac."';");
7:  if($update == 1){
8:  array_push($result, array('status'=>"Device changed"));
9:  }else{
10: array_push($result, array('status'=>"Device not changed"));
11: }
12: echo json_encode(array("result"=>$result));
```

Line 2 is an include method of another page PHP which are header. Line 3 and 4 is about storing the parameter from the Android device to variables in PHP. Line 5 is declaring a new array variable called ‘result’. Line 7 until 11, if the update is successfully done ‘Device Changed’ statement will be pushed, otherwise ‘Device not changed’ statement will be pushed into result array. Line 12 will encode the result to send it back to android device.

4.2 ANDROID FUNCTION

4.2.1 Function logout()

This function is made to logging out from the system. Works the same way like the other logout function, this click event is connected to a web service and will logged the account out of the system.

Referring to attachment D, at first a text ‘Loading..’ is shown, telling that the process is in working. After that a it connects to the web service using ‘StringRequest’ Method, if the response equals to ‘1’ then login page will be shown otherwise an error message will be shown for the user.

4.2.2 Function setKey()

This function is used to generate a key for a user once they login. It’s used to communicating between server and users. This function connected to a web service called ‘setAccountKey.php’.

Referring to Attachment E, at first a text ‘Loading key..’ is shown, telling that the process is in working. There are some parameters that will be posted they are account_id, account_password, account_key, iv, and dummy_id. After that it connects to the web service using ‘StringRequest’ Method. If the result is below 16 characters then will be added with “0” at the front of the key until the length is 16 characters. After that the key is saved using preferences. If the process to generate the key is error or fail, it will show a text “The server is unreachable” or an error message.

4.2.3 Function setGroupingState()

This function is used to change the state of a group that owned by the user or maybe by the other user who’s authorized to use or control this group. This function connected to a web service called ‘setGroupingState.php’.

Referring to Attachment F, at first a text ‘Loading..’ is shown, telling that the process is in working. There are some parameters that will be posted they are account_id, account_password, grouping_id, device_id, device_state, and iv. After that it connects to the web service using ‘StringRequest’ Method. If the response equals to ‘1’, then the system will show ‘Item saved’, if it’s not then the system will show ‘Item not saved’ or an error message will reveal.

4.2.4 Function sendrecover()

This function is used to reset the user’s password if they forget their password by inserting user’s email. This function connected to a web service called ‘resetAccountPassword.php’.

Referring to Attachment G, at first a text ‘Loading ID..’ is shown, telling that the process is in working. There’s a parameter that will be posted it is account_email. After that it connects to the web service using ‘StringRequest’ Method. If the response equals to ‘1’, then the system will show ‘Send password recovery success’ and the page is directed to login page. If the response isn’t ‘1’, then the system will give an error message to notify the user that the password recovery can’t be sent.

4.2.5 Function login()

This function is used to logging in the system. Works the same way like the other login function, this click event is connected to a web service called ‘getAccountId.php’.

Referring to Attachment H, at first a text ‘Loading ID..’ is shown, telling that the process is in working. There’s a parameter that will be posted it is account_email and account_password. After that it connects to the web service using ‘StringRequest’ Method. The response will be encrypted with SHA and needed to be decrypted. Once it’s decrypted, the system will make a session of this account and go to the ‘MainActivity’. But if the response is error, the server will give an error message whether it’s server failure or others.

4.2.6 Function updateAccountPass()

This function will be used for updating user’s account password. This function is connected to a web service called ‘updateAccountPassword.php.

Referring to Attachment I, at first a text ‘Loading..’ is shown, telling that the process is in working. There are some parameters that will be posted they are account_id, account_password, and account_new_password. After that it connects to the web service using ‘StringRequest’ Method. If the response equals to ‘1’ then the password is already updated, but if it’s not ‘1’ it will give an error message.

4.2.7 Function updateAccWifi()

This function will be used to update the user's WiFi information that have been written in the database since they registered it. This function is connected to a web service called 'updateAccountWifi.php'.

Referring to Attachment J, at first a text 'Loading..' is shown, telling that the process is in working. There are some parameters that will be posted they are account_id, account_password, account_wifi_ssid, and account_wifi_password and iv. Before the account_wifi_ssid and password were sent, they were encrypted first to make sure that the data is unreadable. After that it connects to the web service using 'StringRequest' Method. If the response equals to '1' then the WiFi information is already updated, but if it's not '1' it will give an error message.

4.2.8 Function saveDevice()

This function is used to register new device that the user has just bought. By scanning QR code from the device with a device id and name on it. There are also some parameters that will be posted they are account_id, account_password, device_id, device_password, and iv. . After that it connects to the web service using 'StringRequest' Method. If the response equals to '1' then the registration is complete, but if it's not '1' it will give an error message.

CHAPTER V
DOCUMENTATION

5.1 Technician

5.1.1 Manufacturing of device

Technician first must prepare a relay module, a led, an Arduino board that sports ESP8266 and 5 volt output, which WeMos D1 mini board is guaranteed to have, and the necessary cables and equipment to craft the device as stated in chapter 3 of this book. Technician can use almost any relay module, however the D0 pin of the WeMos D1 mini board, more commonly known as pin number 5 of the Arduino board must be used before other pins. After assembly, the technician must access the register device web service, and add the device id, which is the combination MAC Address and each of the board's used pins, one by one. An example of a registration of a device with the MAC address AA:AA:AA:AA:AA:AA and with the pin D1 (5), will be by accessing this link, http://1.1.1.1/system/admin_registerDevice.php?device_id=AA:AA:AA:AA:AA:AA-5 , in the web browser. This action will generate a QR code with their special text like the example below.

Gqs2b0ETmivCHEfOwTSq20bMILbHdbbcPqRrqibQoj4=

There may be multiple QR codes for a device along with the normal text version of each of the QR codes, which depends on the number of relay pins used, which then are stored with the device as well. This codes will be user for the users to register the devices later.

5.2 User

5.2.1 Registration

A new user first must download the application on their Android phone. When prompted, the user can simple press the register button, this will cause another page to appear. User then will fill the asked information, and finish the registration process. The user then will open their email, and press the link that is included with the email. This action will confirm the user's registration, and allow them to access the application the next time they log in. It must be noted that an account can be used by one phone at a time, not allowing a double login scenario.

5.2.2 Ownership Authentication

Users must have their QR codes of the device before they are able to use the devices. They then can access the device tab of the Android program, and press the floating button in the bottom right corner. A new page shall rises, with QR code scanner with a text box in the bottom part of the page. User then can either scan the QR codes or input the texts of the paper included with the device. After the process is done, the devices that have the same device id will appear on user's device tab. Users then can edit each and every of their device by pressing a specific device tab, or even delete them one by one if they wish to by holding the device tab for a period of time and confirming it.

5.2.3 Groupings of Devices

First, the user must have registered at least a device, then the users if they wish, can go to the group tab of the Android program. There will be 2 floating buttons, in the bottom right corner of the device. After the bottom one has been pressed, the button will prompts another page to appear, where users can specify about the group's information and what devices are included in the said group. Then the group will be shown at the user's group tab of the application. When a user wanted to, the groups can also have their details accessed by pressing the specific group tabs.

Another way to add a group is by asking other registered people to share their group with the user. By using the upper floating button, other person can type their to-be-shared group and their log in information. After finalizing the process, the user will have access to use the other person's shared group. However, the user won't have any access in editing the said group.

5.2.4 Device Usage

In every tab of a group or a device, there will be a switch in their right. Pressing this button when the button is not colored blue, will color the switch blue, and turns on the device or group. The other way, when a pressed button was colored blue, and now is turning gray, will turn off the said device or group. This method will also work for activating or deactivating a timer, which will be explained after this.

5.2.5 Timers

Automated turning on or off a group is possible because of this feature. The user can access the menu by going to the schedule tab, and accessing the usual floating button, which is still and always located in the bottom right corner of the

device. The button will show a page which will ask about what the timer should do and when to do it. To edit a registered timer, the user can just press a timer's tab. To delete however, the user need to hold the timer's tab and confirm its deletion.

5.2.6 Speech Commands

Speech button in the group tab, can only be used for simple commands for the time being. The commands are "turn on " with the group's name, and "turn off" added with the group's name. This will allow a quick turning on or off a group. When speech is not possible, the user can simply type the queries in the text box just below the speech button, this will have the same effect as the speech button.

5.2.7 Additional Menus

More menus can be found by accessing the side menu in the left of the program. Users can access their profile, and edit their password and WiFi information. The support tab will prompt the user to the email application, allowing users to communicate with the developers of the WiRe application. The sign out menu will allow user to log out of the application, allowing the user to access the account from different phones or other user to access their own account in that phone.

Chapter VI
Conclusion

This chapter will end the development report of WiRe's, there may be some flaws and wrongdoings because of the developers' inexperience causing the application to be far from perfect. Some of the more interesting and needed features were discovered too late to be implemented, however the developers have tried their best to implement them and do apologize for such drawbacks. The developers also welcome the readers to give ideas and constructive criticism for them to help create a better project in the future.That being said, it doesn't undermine the fact that the developers are fully confident about the product of their hard work that they committed, which is embodied in the form of a working WiRe application and service that has gone beyond their previous expectations. The developers wish the best for both the reader and the users.

6.1 Summary

WiRe's results has been spectacular, even better than expected on some components:

-Functionality given by WiRe is more than what traditional switches can offer, from geographical advantage where users can access the switch through the internet, to chronological advantage where timers can be set in advance to turn a device on or off at a later time.

-WiRe's intuitive IoT switch is easy to set and to operate even for those who doesn't understand computers or electrical engineering.

-WiRe has an adequate user interface to allow access to database from different inputs such as voice, and also some unique ways to represent data to allow users to digest a lot of information without feeling overwhelmed.

-Secure end-to-end encryption and database's encryption has been implemented is to provide a reliable service.

6.2 Recommendations

The developer concludes that the application WiRe has achieved satisfactory results and even more than its current objectives, yet there were some flaws which the developers failed to implement for a more ideal WiRe for them. Aside from the

developer’s commentaries, there were also advices and suggestions given by more experienced developers which can be implemented to polish future developments, and here are some points to be considered during the next development of WiRe by both the current developers and experts:

-More customizations such as day theme and night theme, languages for the mobile device’s interface.

-Implementation of chat head, allowing users more freedom to access WiRe while accessing other applications.

-Real time checking thread’s implementation for the groups, timers and devices.

-More detailed and streamlined objectives at a shorter time frames, allowing a development with higher clarity and productivity.

-Implementation of notifications for shared groups and devices’ usage.

-Search of more advanced and reputable libraries to be used at future developments.

-More time and resources should be invested into testing the project to ensure the completion of the goals.

-Operation of WiRe with technologies like bluetooth which doesn’t require a central server and an internet connection.