Fpt University

TRƯỜNG ĐẠI HỌC FPT

Capstone project document

GREEN - HOA LAC

HÒA LẠC

Version 1.2 approved

Prepared by Green Hoa Lac group

Date created: 05-08-2017

Last updated: 08-22-2017

HÒA LẠC

Green – Hoa Lac		
Group members	Pham Quang Khang	SE03624
	Le Anh Tuyen	SE03863
	Nguyen Thi Ly Linh	SE04216
	Bui Manh Tri	SE03590
	Lam Duc Thang	SE03529
	Ho Quang Hao	SE03856
Supervisor	Phan Duy Hung	
Extra supervisor	N/A	
Capstone project code	GHL	
Capstone project name	Green – Hoa Lac	

Table of contents

Chapter 1: Introduction

I. Introduction and motivation

Nowadays, technological development follows no physical or invisible boundaries. Remote monitoring, data collection and processing systems are gaining more and more popularity in many different areas such as agriculture, medical or education.

Our group expect to apply our knowledge which we have learned to develop a remote monitoring system in own FPT university in Hoa Lac. As a result, we decide to research and develop a remote watering system.

II. Existing products

An excellent remote watering system requires the productivity and save cost. It cannot stop with only actions turn on or off pumps through text message or phone call from long distance. User's requirements are increasing rapidly. With the technology development especially in embedded systems, people expect their remote watering system as automatically as possible, which can reduce need in human resources.

1. Schneider Sprinkler Timer and Controller

Figure 1.1: Schneider Sprinkler Timer and Controller

Mechanism: Controlling by telephone's text messages

- Turn on or off device with long distance by telephone
- Send text message to telephone to announce or warn status of devices such as: be on, be off, power loss, be cut off, SIM banned
- Activate on or off devices by text messages or called miss
- Set up time for turning on or off devices by one message (30 minutes – 23.5 hours)

Advantages:

- Thanks to external antenna, these devices can receive high signal, high interference resistance
- High touch and stability

2. *HT-01 Mobile phone Controller*

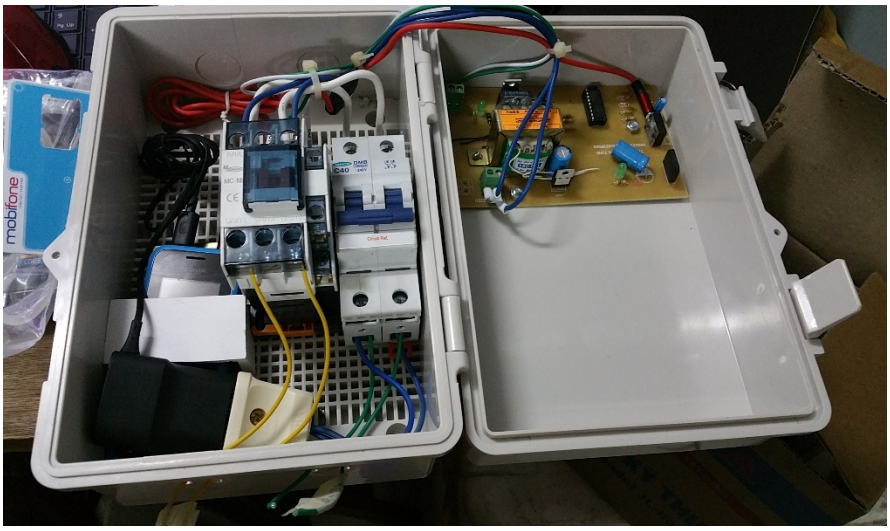


Figure 1.2: HT-01 Mobile phone Controller

Mechanism: Controlling by telephone's calls

- Turn on or off device with long distance by telephone
- Call to a telephone number installed in HT -01 and valid code to turn on devices. The same for turn off devices but using other valid code.

Advantages:

- Low price for farmers and gardeners

III. Scope

The group develops a new remote watering system called Green – Hoa Lac system with two main parts: android application to control and web management to set up, update users and areas applied the system.

On the one hand with the web management (called Green - Hoa Lac web management system), we manage nodes and users of whole system. First of all, Google map is applied to manage node's locations because it provides accurate and portable locations, which is usually updated in case of changing. This way help us reduce time and efforts to design and build a new atlas served for node's position and other information management. In addition, one area can divide into many subareas and that can be controlled by many other people, which lead to a need in management user and activity history of each node. As a result, we can easily shrink or extend scope of areas expected.

On the other hand with an android application (called Green - Hoa Lac android application), we use it remote pumps in each node, set scheduler for both short and long time duration for each node, and also can view weather forecast, activity history. The application use the same database and Google map with the above web. We control and set schedule for pumps through Subscriber Identity Module and the need for internet connection is limited.

Table 1.1: Scope and boundaries of whole system

Feature	Sub Feature	Description
Green- Hoa Lac web management system	Node's management	- Create, update, delete node in the map - Manage information, description, history and position of node
	User's management	- Create, update, delete, lock or unlock users - View activity history

Green – Hoa Lac android application	Turn on/ off node	Control turn on or off node with long distance
	Set/ get/ cancel schedule	Set or cancel schedule for node with detailed time, duration with frequency like one day, daily, weekly. Get all list of schedules of nodes
	View weather forecast	With support of the Internet connection, user can view weather forecast directly in the application to control and set scheduler.
	Synchronize and view node history	The application enable users control the system according to two ways which are online and offline. Therefore, synchronize and view node's history are very important to control users and nodes.

IV. Definitions, Acronyms and Abbreviations

Table 1.2: Scope and boundaries of whole system

#	Acronyms	Definitions
1	GHLS	Green - Hoa Lac system
2	GHLWMS	Green - Hoa Lac web management system
3	GHLA	Green - Hoa Lac android application
4	PCM	Pump control module

Chapter 2: Project management Plan

I. Project organization

1. System process model

Our project applies both plan-driven and agile processes. In plan-driven process, process activities are planned in advance and progress is measured against this plan while in agile process, planning is incremental and it is easier to change the process to reflect changing requirements.

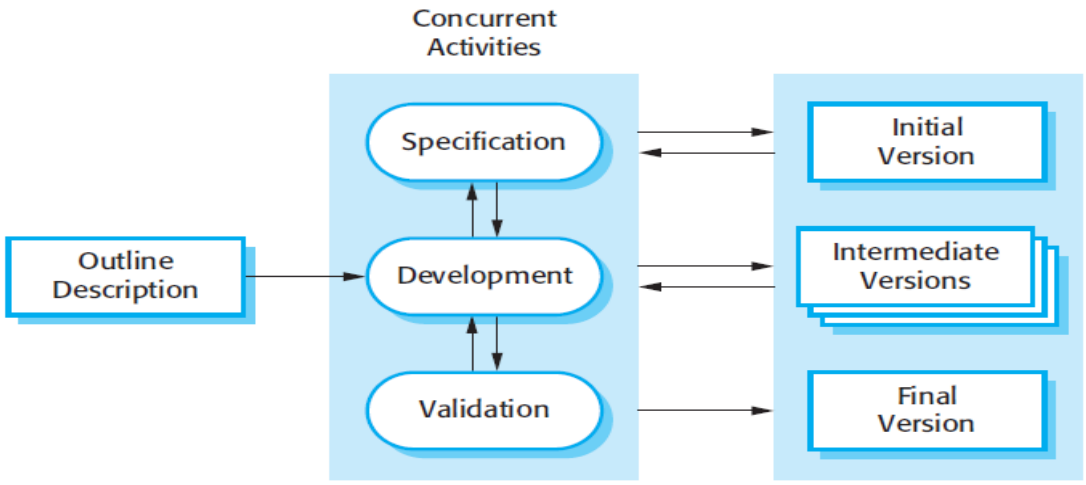


Figure 2.1: The Incremental Development Model

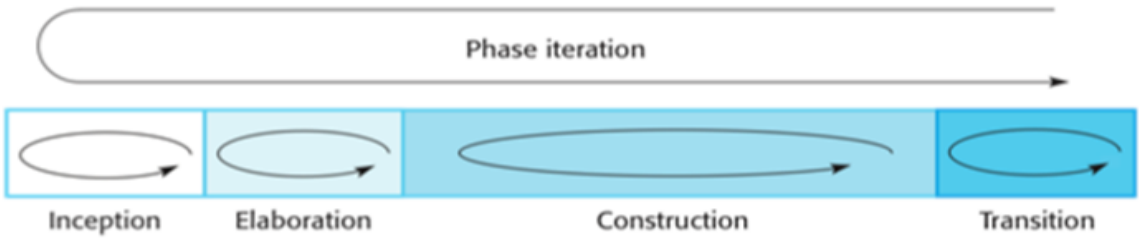


Figure 2.2: The Iterative Development Model

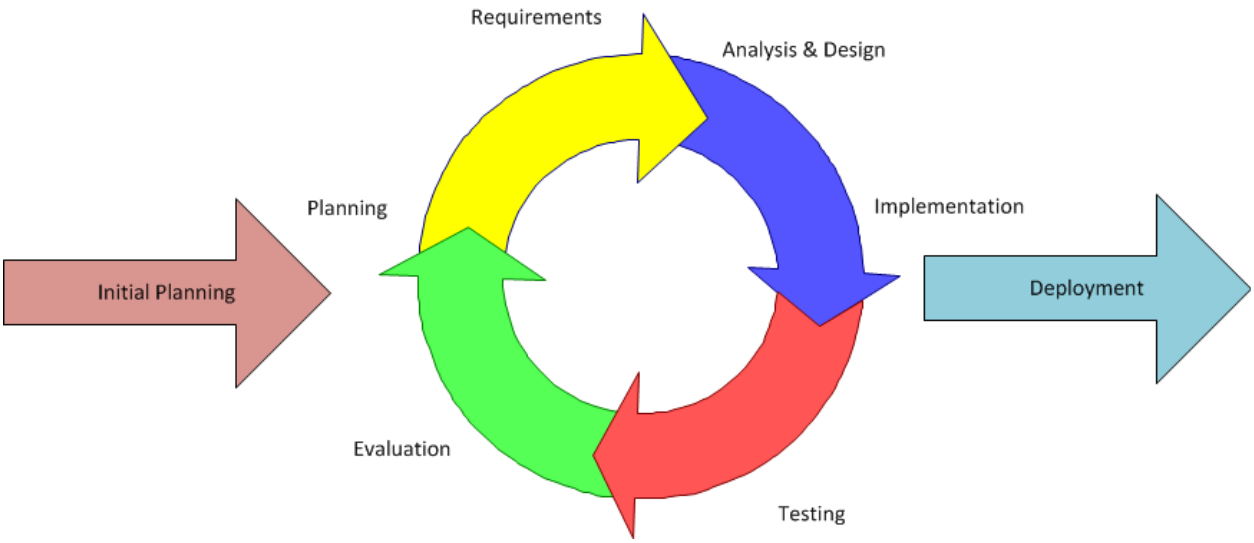


Figure 2.3: System Process Model

On the other hand, there is a combination between the Iterative and Incremental Software Process Model in the project. Iterative and incremental software development begins by the way plan and continue through iterative development cycles involving continuous user's feedback and the incremental addition of features concluding with the deployment of completed software at the end of each cycle, which is appropriate for a gradual increase in feature additions and a cyclical release and upgrade pattern. These models are usually chosen in some situation. Firstly, the cost of accommodating changing requirements is reduced. The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model. Secondly, it is easier to get customer feedback on the development work that has been done. Customers can comment on demonstrations of the software and see how much has been implemented. In addition, more rapid delivery and deployment of useful software to the customer is possible. Finally, these model do not require knowledge, skill of team's members in advance.

2. *Roles and responsibilities*

Table 2.1: Roles and responsibilities

No.	Full name	Role	Responsibilities
1	Ph.D. Phan Duy Hung	Supervisor	- Supporting in raise idea and giving advices for design the system. - Approving and supporting process to run project. - Suggesting solutions when the project has issues.
2	Pham Quang Khang	Project manager	- Managing member's tasks, set scheduler and risk - Set common rules for all members in project - Organizing meeting, communication plan to keep track project's process. - Approve solutions to resolve issues
		Developer	- Follow process of project and common rules. - Develop Green Hoa Lac web management system
3	Le Anh Tuyen	Technical leader	- Design system architecture
		Developer	- Follow process of project and common rules. - Develop interaction with hardware

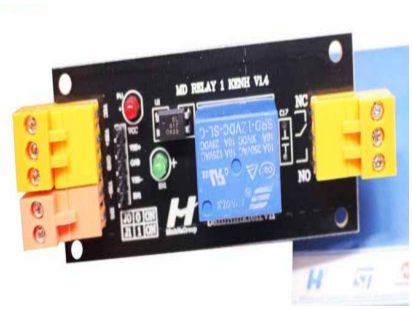
4	Nguyen Thi Ly Linh	Developer	- Follow process of project and common rules. - Develop Green Hoa Lac web management system
		QA	- Keeping all member on process and follow common rule - Controlling quality of the projects: time, function, risk.
5	Bui Manh Tri	Developer	- Follow process of project and common rules. - Develop Green Hoa Lac android application
6	Lam Duc Thang	Developer	- Follow process of project and common rules. - Develop Green Hoa Lac android application
7	Ho Quang Hao	Tester	- Follow process of project and common rules. - Test all cases following requirement.
		QA	- Controlling quality of the projects: time, function, risk.

II. Tools and infrastructures

1. Hardware

Table 2.2: List hardware devices

Name	Image	Information
Module SIM800A MH		Designed for global market, SIM800 is a quad-band GSM/GPRS module that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM800 features GPRS multi-slot class 12/ class 10 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4 Digital information: - Voltage: 9-12V DC_2A - SIM800A - Size: 75x75mm - Module developed from PCB of Module Sim 900A V1 - Be like Module Sim 900A V1, Module use sets of commands AT message, call phone - Because Sim800A does not have a mechanism selecting manually or automatically

Adapter AC-DC Mini 5V700mA		Digital information: - Input voltage: AC85-265V - Frequency: 50/60Hz - Size: 30x20x17.5MM - Output: DC5V 700mA
Module DS1307+AT24C32 [1]		Supporting to communicate with DS18B20 Digital information: - Using DS1307 - Using EEPROM AT24C32 - Standard I2C
Module Relay 1 channel 5V-220V/10A		Control devices through Relay Digital information: - Signal into control: DC5V - Default control: + Turn off - 0, turn on – 1 - Changing J1, J0 to change control level - Output: + Contact point relay 220V 10A + NC : close + NO : open - Symbol power: + VCC, GND are common power + VSS+ , VSS- are power of Relay

2. *Tools and software*

Table 2.3: Tools and software

Tools	Image	Information
Window 10		• Computer operating system developed and released by Microsoft • Used for: programming, hosting • Version: Windows 10
GitHub		• Used for software development and other version control tasks, control version, source code
Google drive		• Storing documents like: software requirement specification, communication plan or risk management
Balsamiq Mockups		• Draw screen mockup for web and android application. • Version: 3.5.14
Astah		• Draw diagrams for the whole system • Version: 7.1.0

Robo 3T		• Manage system's database server • Version: 1.1.1
Fritzing		• Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. • Version: 0.9.3b
IDE Arduino		• Write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open source software • Version: 1.8.4
Android studio		• Build android application intelligent code • Version: 2.2.3
Web storm		• Code web • Version: 2017.2

Chrome		• Browser support to browsing internet
Office		• Project 2013 • Excel 2013 • Word 2013 • Power point 2013

III. Risk management plan

1. Risks register

No	Rank	Risk	Description	Category	Root cause	Triggers	Potential Responses	Risk Owner	Probability	Impact	Status
R1	1	Out of budget	Out of estimated cost before finishing project	Financial risk	We usually loose control and update financial reports	Increase time and budget for project	Make sure PM is sensitive and focus on any change of finance in project	LinhNTL	Medium	High	PM will have report of finance in every month
R2	2	Communication	Team member doesn't have enough time to meet together	People risk	Capstone project is started during members are studying other subjects in semester	Reduce performance of project	Member must destroy their schedule to meeting with team	TuyenLA, development team	High	High	Members should arrange their work and time to meet together more
R3	3	Team member	Team member doesn't have enough experience	People risk	Don't training before begin developing	Reduce performance of project	Plan for training knowledge and skill for developer	TuyenLA, development team	Medium	Medium	PM and techinical leader will tranfer and training knowledge in project
R4	4	Bugs	Still having bugs in time delevering	Market risk	Don't intergate testing before releasing	Reduce quality of product	Plan cycling for testing	HaoHQ	Medium	High	Test leader have to intergate testing before release
R5	5	Change requirement in developing process	The requirements change for new fucntion or service	Technical risk	We won't analyze detail each function and technical in system	The SRS is not stable may cause be late for delivering time	Usually up to date SRS and tranfer for all member about CR in project	TuyenLA, development team	Medium	Medium	Spend more time for analyze requirements and research technology to do
R6	6	Device, Hardware	Device for developing and testing is limited. Hardware must change constantly	Technical risk	Device and hardware are not stability	Increase time and budget for project	Members must use more time to buy device and hardware	TuyenLA, development team	High	High	Technical team will make a check list plan for hardware, devices
R7	7	Network connection	Team members get difficult during exchange information and unified opinion	People risk	Members must go home and can not meet together directly	Reduce performance of project	Members maybe misunderstanding and make project plan is changed	TuyenLA, development team	High	High	Members must prepare some applications to improve quaility of meeting through video call

Figure 2.4: Risk register

2. *Risk probability and impact*

PROBABILITY	HIGH			R2 R6 R7
	MEDIUM		R3 R5	R1 R4
	LOW			
		LOW	MEDIUM	HIGH
	IMPACT			

Figure 2.5: Risk probability and impact

IV. Communication plan

1. *Project structure*

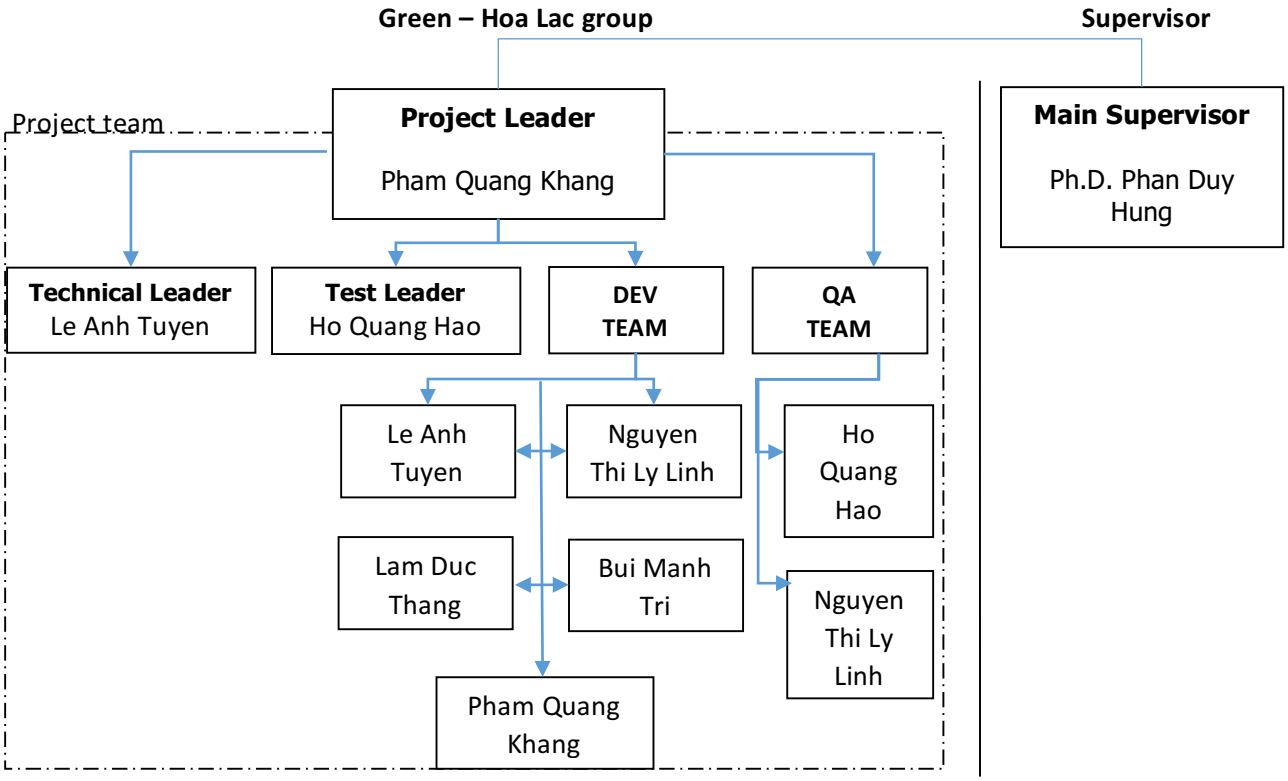


Figure 2.6: Project structure

2. *Project communication*

2.1. Format, content, and level of detail of key project information

Table 2.4: Format, content, and level of detail

Information	Author	Receiver	Method/Technology
Schedule updates	Project manager	Project team Supervisor	Email Group meeting
Project status	Project manager	Project team Supervisor	Email Group meeting

Agenda/Meeting Minutes	Project manager	Project team	Email
Issues	Test Leader	Project manager	Email Group meeting
Status report	Project team	Project manager Supervisor	Email
Project announcement	Project manager	Project team Supervisor	Email Instant message
User requirement	Project manager Supervisor	Project team	Email Hard copy

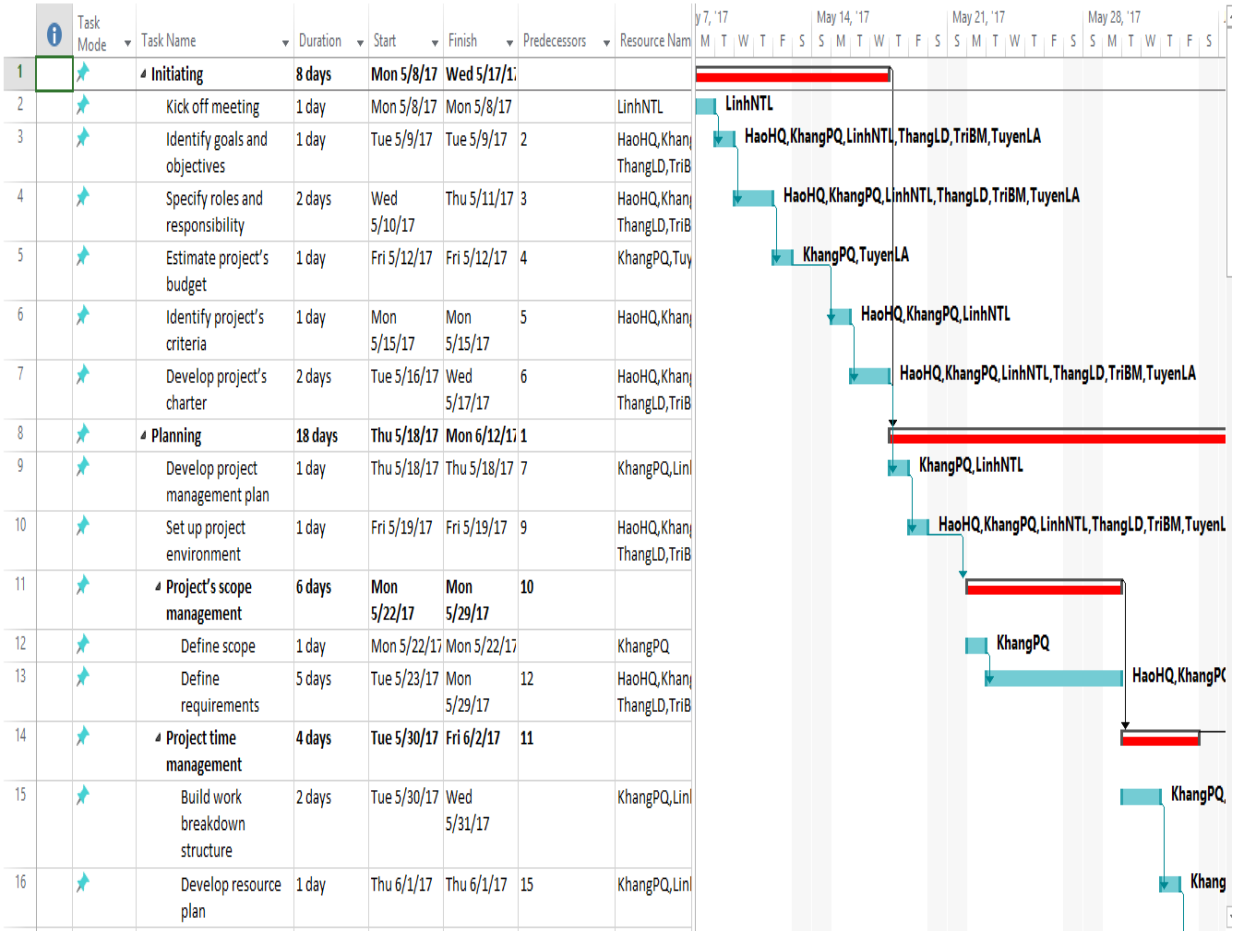
2.2. Project report

Table 2.5: Project report

Activity	Participants	Frequency/When	Method
Daily Meeting Report	Author: Team leader (developer and tester) Participants: all members in develop and test team	17h30 each day (Monday- Friday)	Short Meeting
Weekly Status Report	Author: Pham Quang Khang	Friday	E-mail

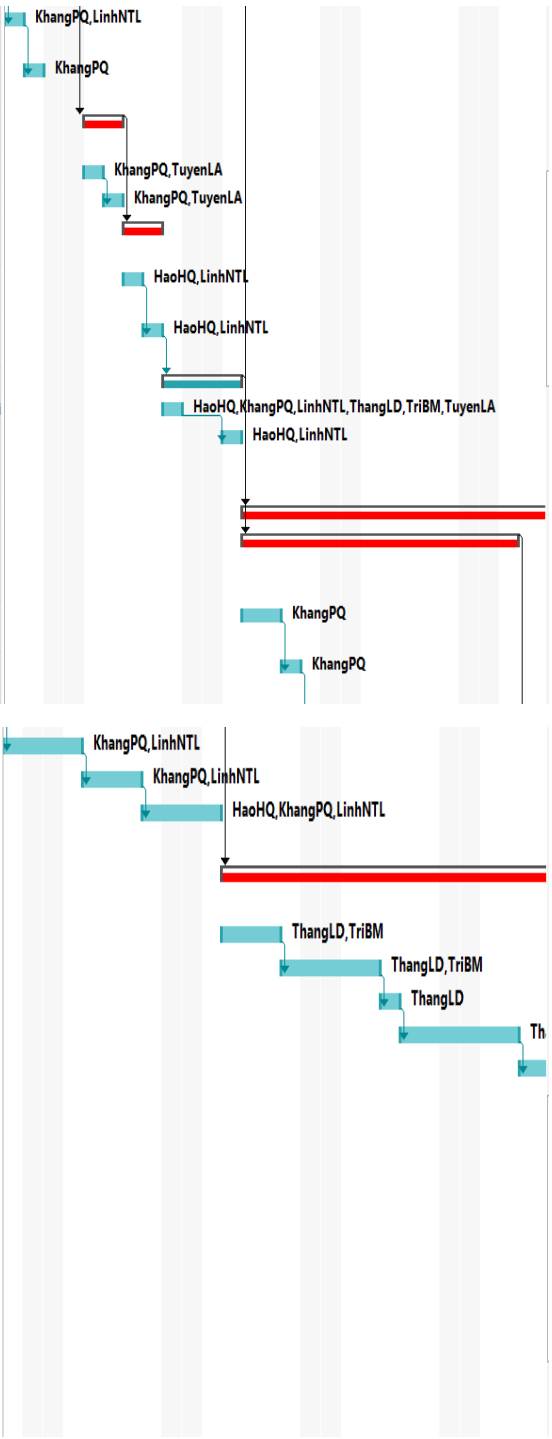
	Distribution list: Nguyen Thi Ly Linh, Le Anh Tuyen, Ho Quang Hao		
Ad-hoc Discussion	Initiator: Ph.D. Phan Duy Hung or Le Anh Tuyen or Pham Quang Khang Participants: all members	Event-based	Email Skype Meeting

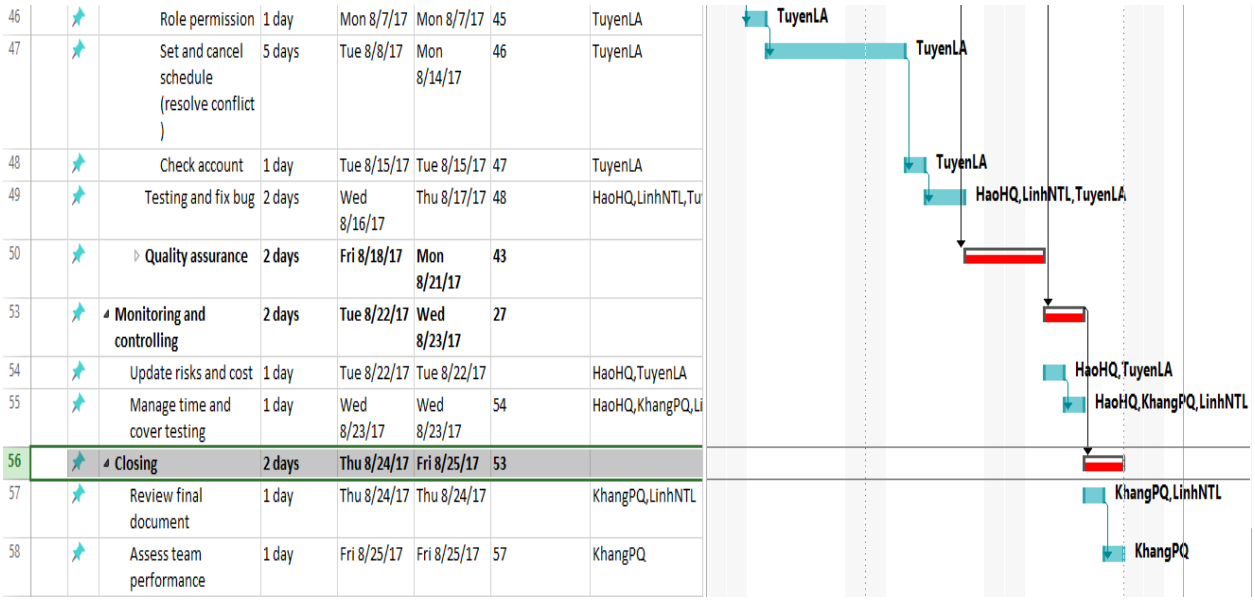
V. Project schedule



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Nam
1		**Initiating**	**8 days**	**Mon 5/8/17**	**Wed 5/17/17**		
2		Kick off meeting	1 day	Mon 5/8/17	Mon 5/8/17		LinhNTL
3		Identify goals and objectives	1 day	Tue 5/9/17	Tue 5/9/17	2	HaoHQ,Khan ThangLD,TriB
4		Specify roles and responsibility	2 days	Wed 5/10/17	Thu 5/11/17	3	HaoHQ,Khan ThangLD,TriB
5		Estimate project's budget	1 day	Fri 5/12/17	Fri 5/12/17	4	KhangPQ,Tuy
6		Identify project's criteria	1 day	Mon 5/15/17	Mon 5/15/17	5	HaoHQ,Khan
7		Develop project's charter	2 days	Tue 5/16/17	Wed 5/17/17	6	HaoHQ,Khan ThangLD,TriB
8		**Planning**	**18 days**	**Thu 5/18/17**	**Mon 6/12/17**	**1**	
9		Develop project management plan	1 day	Thu 5/18/17	Thu 5/18/17	7	KhangPQ,Linl
10		Set up project environment	1 day	Fri 5/19/17	Fri 5/19/17	9	HaoHQ,Khan ThangLD,TriB
11		**Project's scope management**	**6 days**	**Mon 5/22/17**	**Mon 5/29/17**	**10**	
12		Define scope	1 day	Mon 5/22/17	Mon 5/22/17		KhangPQ
13		Define requirements	5 days	Tue 5/23/17	Mon 5/29/17	12	HaoHQ,Khan ThangLD,TriB
14		**Project time management**	**4 days**	**Tue 5/30/17**	**Fri 6/2/17**	**11**	
15		Build work breakdown structure	2 days	Tue 5/30/17	Wed 5/31/17		KhangPQ,Linl
16		Develop resource plan	1 day	Thu 6/1/17	Thu 6/1/17	15	KhangPQ,Linl

16		Develop resource plan	1 day	Thu 6/1/17	Thu 6/1/17	15	KhangPQ,LinhNTL
17		Specify deliverables	1 day	Fri 6/2/17	Fri 6/2/17	16	KhangPQ
18		**Project cost management**	**2 days**	**Mon 6/5/17**	**Tue 6/6/17**	**14**	
19		Estimate cost	1 day	Mon 6/5/17	Mon 6/5/17		KhangPQ,TuyenLA
20		Define budget	1 day	Tue 6/6/17	Tue 6/6/17	19	KhangPQ,TuyenLA
21		**Project quality management**	**2 days**	**Wed 6/7/17**	**Thu 6/8/17**	**18**	
22		Define quality criteria	1 day	Wed 6/7/17	Wed 6/7/17		HaoHQ,LinhNTL
23		Quality assurance plan	1 day	Thu 6/8/17	Thu 6/8/17	22	HaoHQ,LinhNTL
24		**Risk management**	**2 days**	**Fri 6/9/17**	**Mon 6/12/17**	**23**	
25		Identify risks	1 day	Fri 6/9/17	Fri 6/9/17		HaoHQ,KhangPQ,Li
26		Risk management plan	1 day	Mon 6/12/17	Mon 6/12/17	25	HaoHQ,LinhNTL
27		**Executing**	**50 days**	**Tue 6/13/17**	**Mon 8/21/17**	**8**	
28		**Develop web management system**	**10 days**	**Tue 6/13/17**	**Mon 6/26/17**	**24**	
29		Design architecture	2 days	Tue 6/13/17	Wed 6/14/17		KhangPQ
30		Design GUI + database	1 day	Thu 6/15/17	Thu 6/15/17	29	KhangPQ
31		Coding front-end	2 days	Fri 6/16/17	Mon 6/19/17	30	KhangPQ,LinhNTL
32		Coding back-end	3 days	Tue 6/20/17	Thu 6/22/17	31	KhangPQ,LinhNTL
33		Testing and fix bugs	2 days	Fri 6/23/17	Mon 6/26/17	32	HaoHQ,KhangPQ,Li
34		**Develop android application**	**20 days**	**Tue 6/27/17**	**Mon 7/24/17**	**28**	
35		Design architecture	3 days	Tue 6/27/17	Thu 6/29/17		ThangLD,TriBM
36		Design GUI	3 days	Fri 6/30/17	Tue 7/4/17	35	ThangLD,TriBM
37		Design database	1 day	Wed 7/5/17	Wed 7/5/17	36	ThangLD
38		Coding front end	4 days	Thu 7/6/17	Tue 7/11/17	37	ThangLD,TriBM
39		Coding back-end	6 days	Wed 7/12/17	Wed 7/19/17	38	ThangLD,TriBM
40		Testing and fix bugs	3 days	Thu 7/20/17	Mon 7/24/17	39	HaoHQ,LinhNTL,Th TriBM
41		**Develop hardware**	**20 days**	**Tue 7/25/17**	**Mon 8/21/17**	**34**	
42		Make circuit design	2 days	Tue 7/25/17	Wed 7/26/17		TuyenLA
43		**Coding and run module**	**14 days**	**Thu 7/27/17**	**Tue 8/15/17**	**42**	
44		Main program	2 days	Thu 7/27/17	Fri 7/28/17		TuyenLA
45		Application controller	5 days	Mon 7/31/17	Fri 8/4/17	44	TuyenLA





46		Role permission	1 day	Mon 8/7/17	Mon 8/7/17	45	TuyenLA
47		Set and cancel schedule (resolve conflict)	5 days	Tue 8/8/17	Mon 8/14/17	46	TuyenLA
48		Check account	1 day	Tue 8/15/17	Tue 8/15/17	47	TuyenLA
49		Testing and fix bug	2 days	Wed 8/16/17	Thu 8/17/17	48	HaoHQ,LinhNTL,Tu
50		**Quality assurance**	**2 days**	**Fri 8/18/17**	**Mon 8/21/17**	**43**	
53		**Monitoring and controlling**	**2 days**	**Tue 8/22/17**	**Wed 8/23/17**	**27**	
54		Update risks and cost	1 day	Tue 8/22/17	Tue 8/22/17		HaoHQ,TuyenLA
55		Manage time and cover testing	1 day	Wed 8/23/17	Wed 8/23/17	54	HaoHQ,KhangPQ,Li
56		**Closing**	**2 days**	**Thu 8/24/17**	**Fri 8/25/17**	**53**	
57		Review final document	1 day	Thu 8/24/17	Thu 8/24/17		KhangPQ,LinhNTL
58		Assess team performance	1 day	Fri 8/25/17	Fri 8/25/17	57	KhangPQ

Figure 2.7: Project schedule

Chapter 3: System Requirement Specification

I. Green Hoa Lac Android Application and PCM

1. Functional Requirement Specification

1.1. UC01 – Login

1.1.1. Screen Design

Figure 3.1: Login application

Figure 3.2: Google sign in screen

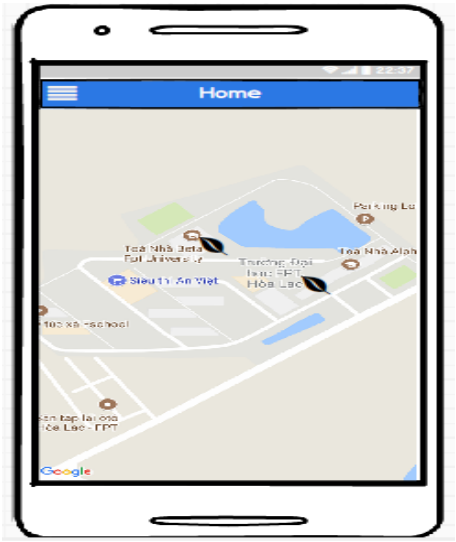


Figure 3.3: Home screen

Figure 3.4: Navigation drawer screen

Table 3.1: Login application Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Login	Button			Navigate to Google Sign In Page.

1.1.2. Use case specification

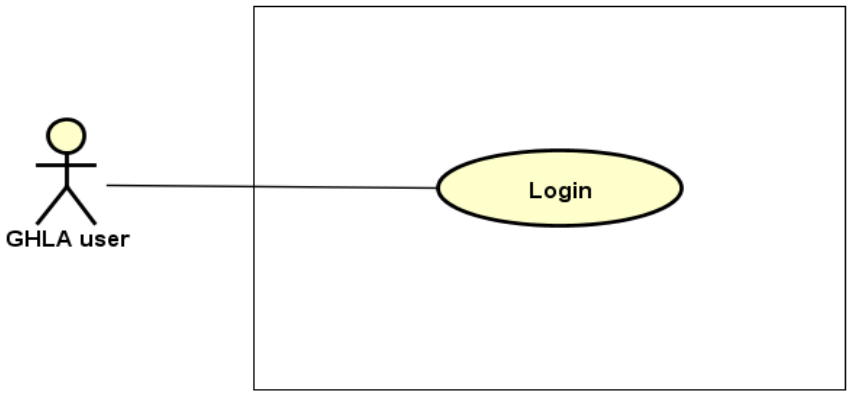


Figure 3.5: Login application Use case diagram

Use Case ID	UC01	Use Case Name		Login	
Author	ThangLD	Version	1.0	Date	22/06/2017
Actor	GHLA User				
Description	The function allows an user to be able to login in the android application when he/she have had an account and his/her account is still active (or not blocked)				
Precondition					
Trigger					

Post-Condition	
Main flows	
Step	***Actor***	***Action***
1	User	Open GHLA
2	GHLA	Display Login screen with the following field: - Login button
3	User	Click on Login button.
4	GHLA	Navigate to Google Sign In Page
5	User	Choose an account in list Google accounts or Enter other email and password for other account.
6	GHLA	Validate the account and then display Home screen

Alternative flows

AT1	At step 4 in the main flows, if users had logged for the first time and there is no internet connection at the present time,	
Sub step	**Actor**	**Action**
4.1	GHLA	Display Home Screen (Map) with data getting from the last login

AT2	At step 5 in the main flows**,** if users enter wrong email account or email's password,	
Sub step	**Actor**	**Action**
5.1	GHLA	Display an error with message "Couldn't find your Google Account" or "Wrong password. Try again", try again step 5.

AT3	At step 5 in the main flows**,** although email account exists, it is blocked or in role “member” or does not exist in database.	
Sub step	**Actor**	**Action**
5.1	GHLA	Display a popup with message “Check your account or Internet connection!”
5.2	User	Click “Ok”
5.3	GHLA	Return step 2

Business Rules

#	***Rule Description***
BR01	GHLA user includes 3 objects: Admin, Manager, Member
BR02	- For the first time logging the GHLA, user must have Internet connection for their android device - For the next time logging the GHLA, + If having Internet connection, GHLA automatically load data (map, node, activity history) from the server’s database and send log to the server. + If not having Internet connection, GHLA will use data in local device (realism) from the previous time logged.

1.2. UC02 – View weather's information

1.2.1. Screen Design

Figure 3.6: View weather's information screen

Table 3.2: View weather's information Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Location	Text			Name of province
2	Temperature	Text			Temperature at the present time. unit : ^{o}C
3	Speed of wind	Text			Speed of wind at the present time. unit: m/s
4	Amount of rain	Text			Amount of rain. unit: %
5	Weather forecast	List view			Weather forecast for the present day and the next 4 day

1.2.2. Use case specification

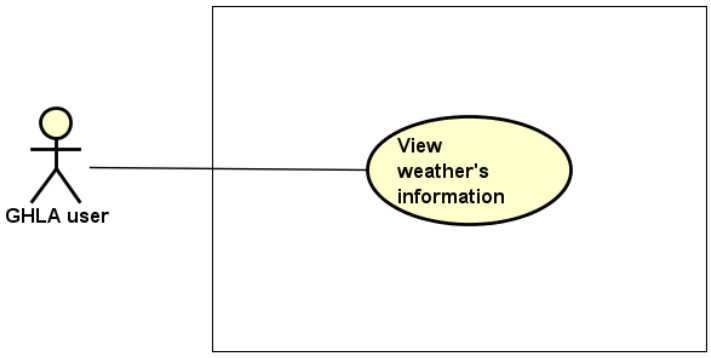


Figure 3.7: View weather's information Use case diagram

Use Case ID	**UC02**	**Use Case Name**		**View weather's information**	
Author	**TriBM**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to view information of weather forecast				
Precondition	There is Internet connection				
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Touch button on top left of the Home Screen			
2	GHLA	Display a navigation drawer on the left of the screen			
3	User	Select and touch Weather's information item on the drawer			
4	GHLA	Display View weather's information screen			

Alternative flows

AT1	At step 4 in the main flows, if there is no internet connection at the present time,	
Sub step	**Actor**	**Action**
4.1	GHLA	Display message “Check your account or internet connection” on popup at center
4.2	User	Touch “Ok” button on the popup
4.3	GHLA	Return Home Screen

1.3. UC03 – View application’s information

1.3.1. Screen Design

Figure 3.8: View application's information screen

Table 3.3: View application's information Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Information of project and instructors	Text			
2	Team's members	Text			

1.3.2. Use case specification

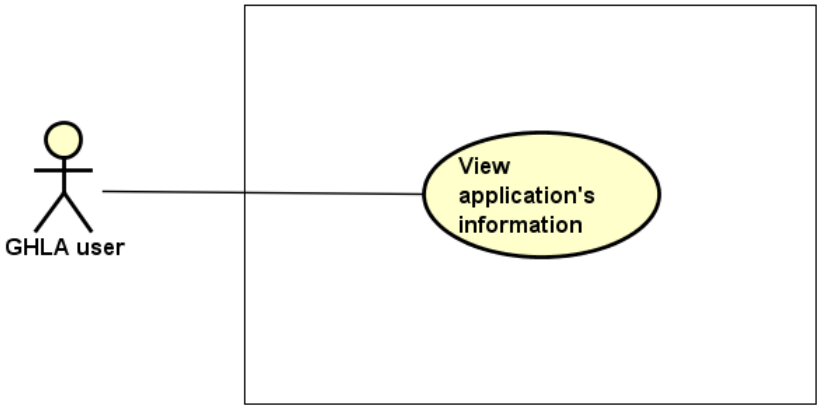


Figure 3.9: View application's information Use case diagram

Use Case ID	UC03	Use Case Name		View application's information	
Author	TriBM	Version	1.0	Date	22/06/2017
Actor	GHLA User				
Description	The function allows an user to be able to view application information				
Precondition	User logged GHLA				

Trigger		
Post-Condition		
Main flows		
Step	***Actor***	***Action***
1	User	Touch button on top left of the Home Screen
2	GHLA	Display a navigation drawer on the left of the screen
3	User	Select and touch Application information item on the drawer
4	GHLA	Display View application information" screen

1.4. UC04 – Turn on node

1.4.1. Screen Design

Figure 3.10: Turn on node screen

Table 3.4: Turn on node Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Mode	Radio group			There are 3 types of mode: - Remote control - Schedule - Check and cancel schedule
2	Status	Drop down list			Status depends on type of mode. If mode is in Remote control, status contains: - Turn on node - Turn off node
3	Duration	Drop down list			There are 4 items to select: - Unlimited - 15 minutes - 20 minutes - 30 minutes
4	Time	Spinner			Time is disable in this case
5	Days in week	Check box			Days in week is disable in this case
6	Day	Button			Day is disable in this case
7	Send	Button			Send a request to PCM

1.4.2. Use case specification

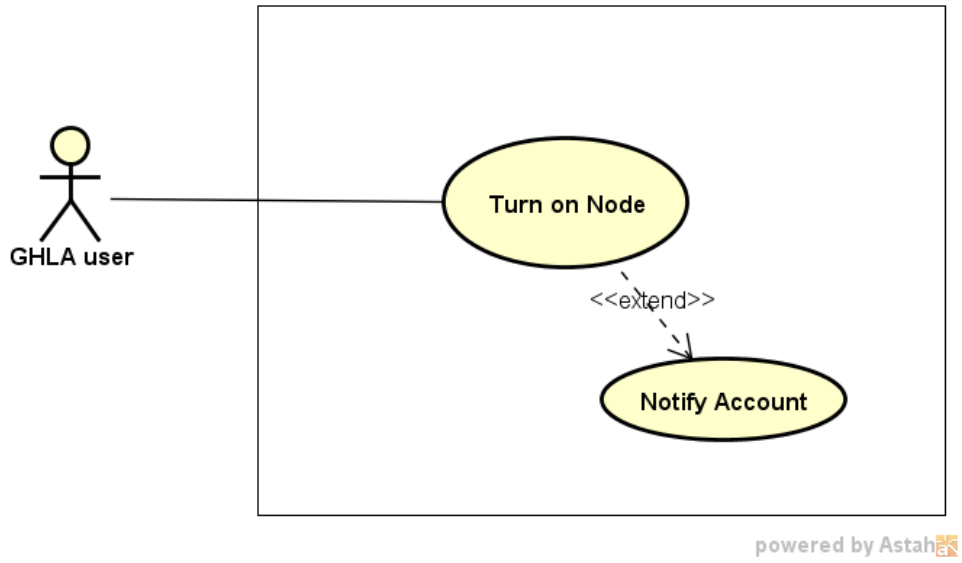


Figure 3.11: Turn on node Use case diagram

Use Case ID	**UC04**	**Use Case Name**		**Turn on node**	
Author	**TriBM, TuyenLA**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to turn on node				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Select and touch a node on Home Screen			
2	GHLA	Display Control node screen as default			
3	User	- Select Remote control mode,			

		- Select Turn on status - Touch Send button
4	GHLA	Display a confirm dialog with message “Are you sure perform this action!”
5	User	Touch “Ok” or “Cancel” button on the dialog
6	GHLA	If user select “Ok”, GHLA validate the phone number and send a bit string with the content user select in step 3 to PCM. Else, return step 2.
7	Module GHL	- Turn on node immediately during the duration time and send a message to user “Turn on successfully!” - Check account and send message to user if account is lower than 10.000 VND
8	GHLA	In case, user receive message “Turn on successfully!” from Module GHL, GHLA send this activity history to server

1.5. UC05 – Turn off node

1.5.1. Screen Design

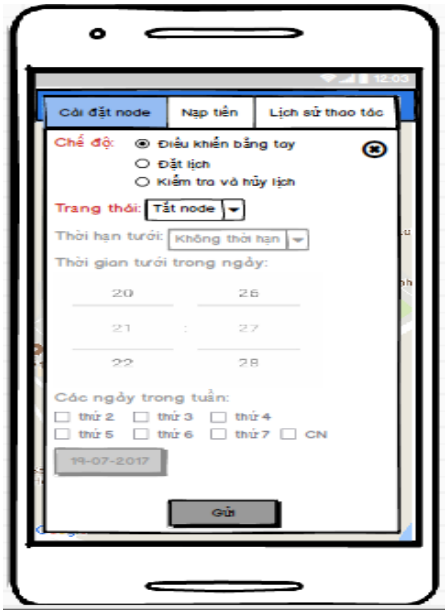


Figure 3.12: Turn off node screen

Table 3.5: Turn off node Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Mode	Radio group			There are 2 types of mode: - Remote control - Schedule - Check and cancel schedule
2	Status	Drop down list			Status depends on type of mode. If mode is in Remote control, status contains: - Turn on node - Turn off node
3	Duration	Drop down list			Duration is disable in this case
4	Time	Spinner			Time is disable in this case
5	Days in week	Check box			Days in week is disable in this case
6	Day	Button			Day is disable in this case
7	Send	Button			Send a request to PCM

1.5.2. Use case specification

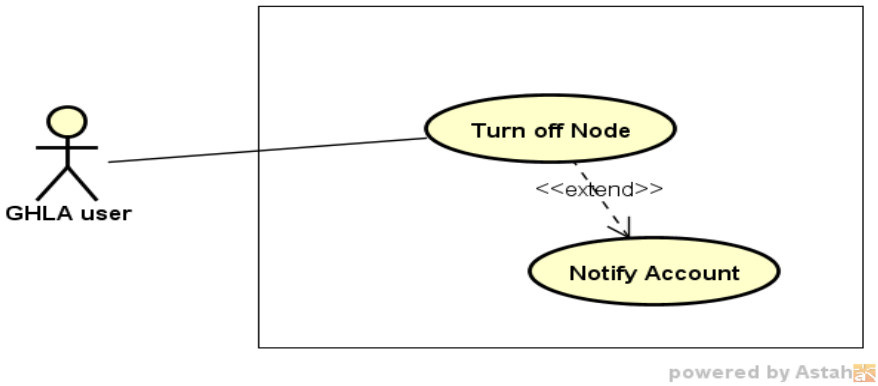


Figure 3.13: Turn off node Use case diagram

Use Case ID	**UC05**	**Use Case Name**		**Turn off node**	
Author	**TriBM, TuyenLA**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to turn off node				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Select and touch a node on Home Screen			
2	GHLA	Display Control node screen as default			
3	User	- Select Remote control mode, - Select Turn off status - Touch Send button			
4	GHLA	Display a confirm dialog with message “Are you sure perform this action!”			

5	User	Touch “Ok” or “Cancel” button on the dialog
6	GHLA	If user select “Ok”, GHLA validate the phone number and send a bit string with the content user select in step 3 to PCM. Else, return step 2.
7	Module GHL	- Check user and turn off node immediately and send a message to user with content “Turn off successfully!” - Check account and send message to user if account is lower than 10.000 VND
8	GHLA	In case, user receive message “Turn off successfully!” from Module GHL, GHLA send this activity history to server

Business Rules

#	***Rule Description***
BR03	GHLA users can turn off pump in a node immediately in accurate cases: - They are people who turn on the pump - They are managers or admin

1.6. UC06 – Set schedule for node

1.6.1. Screen Design

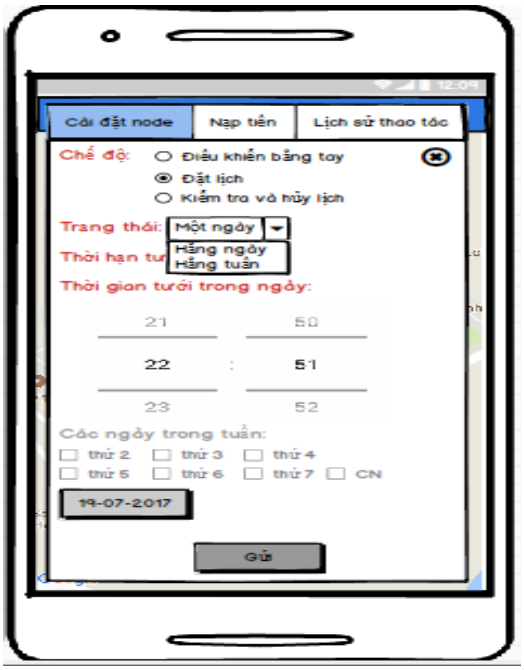


Figure 3.14: Set schedule for node

Table 3.6: Set schedule for node Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Mode	Radio group			There are 3 types of mode: - Remote control - Schedule - Check and cancel schedule
2	Status	Drop down list			Status depends on type of mode. If mode is in Schedule, status contains: - One day - Daily

					- Weekly
3	Duration time	Drop down list			Duration time contains: - Unlimited - 15 minutes - 20 minutes - 30 minutes
4	Time	Spinner			Time depends on type of mode. If mode is in schedule, time get default at present time Else, time is disable
5	Days in week	Check box			Day in week depends on type of mode and status. If mode is in schedule and status is one day or daily, days in week is disable Else, if mode is in schedule and status is weekly, days in week contains a group of check box with content from Monday to Sunday
6	Day	Button			If mode is in schedule and status is one day, Day is enable.
7	Send	Button			Send a request to PCM

1.6.2. Use case specification

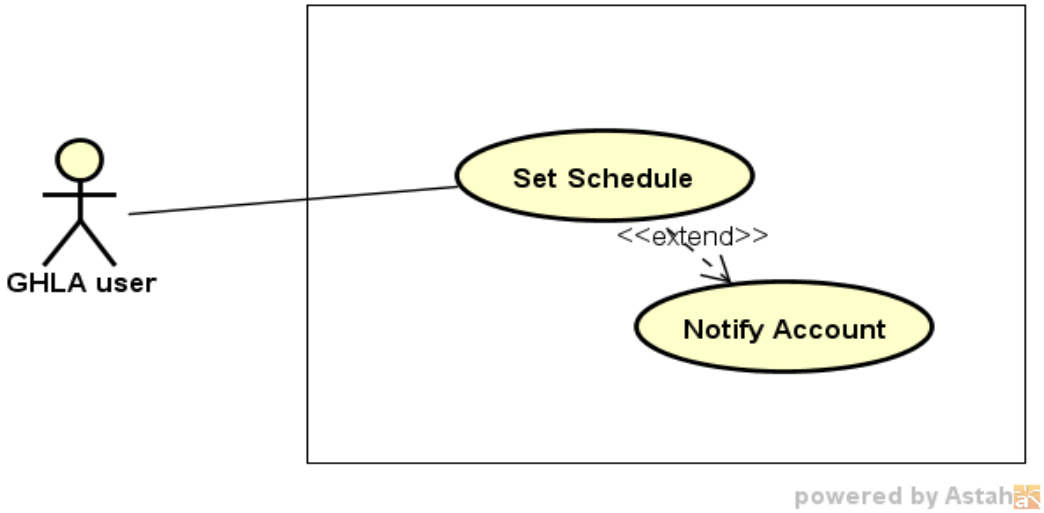


Figure 3.15: Set schedule for node Use case diagram

Use Case ID	**UC06**	**Use Case Name**		**Set schedule for node**	
Author	**TriBM, TuyenLA**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to set schedule for node				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Select and touch a node on Home Screen			
2	GHLA	Display Control and Schedule node screen as default			
3	User	- Select Schedule mode, status, duration time, time, days in week, select a day.			

		- Touch Send button
4	GHLA	Display a confirm dialog with message “Are you sure perform this action!”
5	User	Touch “Ok” or “Cancel” button on the dialog
6	GHLA	If user select “Ok”, GHLA validate the phone number and send a bit string with the content user select in step 3 to PCM. Else, return step 2.
7	Module GHL	- Set up schedule according received message, validate it and then send a message with content “Set up schedule successfully!” to user - Check account and send message if account is lower than 10.000 VND
8	GHLA	In case, user receive message “Set up schedule successfully!” from Module GHL, GHLA send this activity history to server

Alternative flows

AT	At step 7 in the main flows, if user set up a schedule for the one day and time for this day passed,	
Sub step	**Actor**	**Action**
7.1	PCM	Send a message with content “The day or time passed. This schedule is not performed !” to user

Alternative flows

AT2	At step 7 in the main flows, if user set up a schedule that conflicts with other existed schedules	
Sub step	**Actor**	**Action**
7.2	PCM	Send a message with content “There are conflicts between existed schedules and your schedule! Try other set schedule!” to user

1.7. UC07 – View schedules

1.7.1. Screen Design

Figure 3.16: View Schedules

Table 3.7: View schedules Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	List Schedules	Dialog			Show all schedules in a list view with 2 buttons: - Delete all button - Cancel button

1.7.2. Use case specification

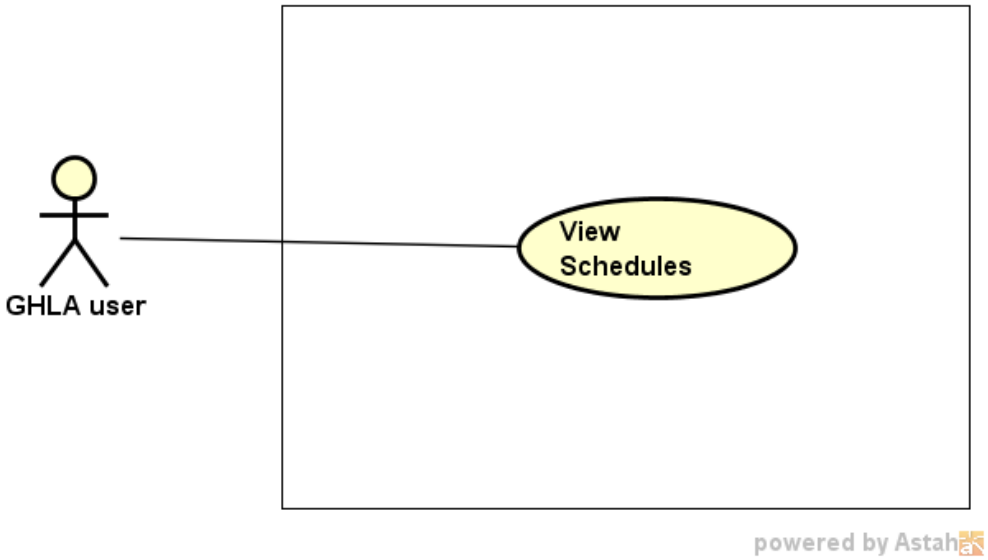


Figure 3.17: View schedules of node Use case diagram

Use Case ID	**UC07**	**Use Case Name**		**View schedules of node**	
Author	**TriBM, TuyenLA**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to view all schedules of nodes				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Select and touch a node on Home Screen			
2	GHLA	Display Control and Schedule node screen as default			

3	User	Select Check and cancel schedule mode
4	GHLA	Send request that is view schedules of node to PCM
5	Module GHL	Send all schedules according to request of GHLA
6	GHLA	Display a list view with all schedules.

1.8. UC08 – Cancel schedules of node

1.8.1. Screen Design

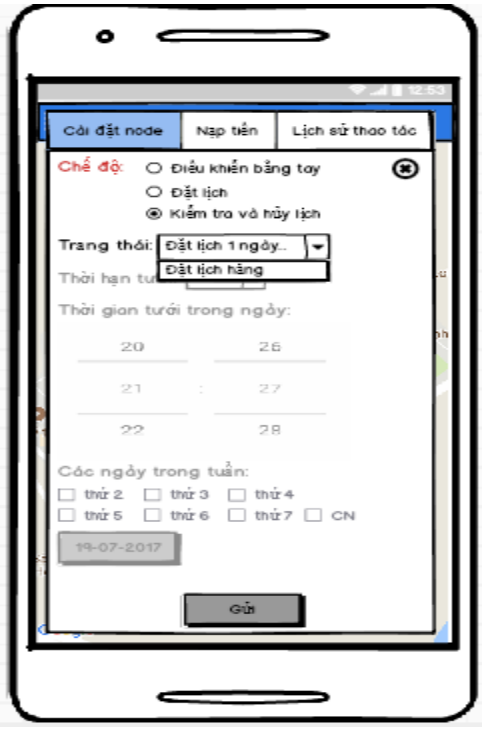


Figure 3.18: Cancel schedule for node screen

Table 3.8: Cancel schedule for node Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Mode	Radio group			There are 3 types of mode: - Remote control - Schedule

					- Check and cancel schedule
2	Status	Drop down list			Status depends on type of mode. If mode is in Check and cancel schedule, Status contain list of all schedules that users want to cancel.
3	Send	Button			Send a request to PCM

1.8.2. Use case specification

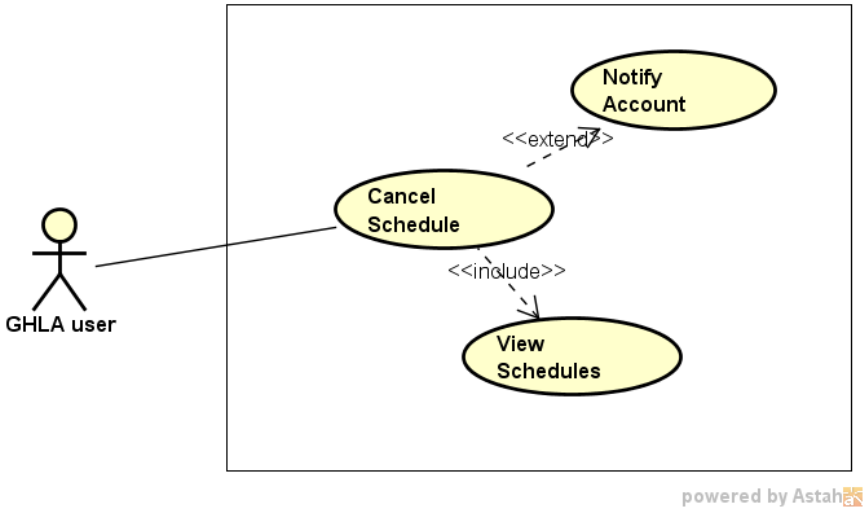


Figure 3.19: Cancel schedule for node Use case diagram

Use Case ID	**UC08**	**Use Case Name**		**Cancel schedule for node**	
Author	**TriBM, TuyenLA**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to cancel one or all schedules of node				

Precondition		
Trigger		
Post-Condition		
Main flows		
Step	***Actor***	***Action***
1	User	Select and touch a node on Home Screen
2	GHLA	Display Control and Schedule node screen as default
3	User	Select Check and cancel schedule mode
4	GHLA	Send request that is view schedules of node to PCM
5	Module GHL	Send all schedules according to request of GHLA
6	GHLA	Display a list view with all schedules.
7	User	Select one schedule on List view or touch Delete button on the screen
8	GHLA	Display Check and cancel schedule screen with status includes all schedules you have selected to cancel
9	User	Touch Send button on the screen
10	GHLA	Display a confirm dialog with message contain all schedules user want to cancel
11	User	Select “Ok” Button
12	GHLA	Send a request cancel schedule to PCM
13	Module GHL	- Cancel all schedule according to the request and send a message to android device to notify for user with content “Cancel the schedule successfully!” - Check account and send message to user if account is lower than 10.000 VND
14	GHLA	In case, user receive message “Cancel the schedule successfully!” from Module GHL, GHLA send this activity history to server

1.9. UC09 – Recharge

1.9.1. Screen Design

Figure 3.20: Recharge screen

Table 3.9: Recharge Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Card number	Text box			It is a place used to fill card number
2	Recharge	Button			

1.9.2. Use case specification

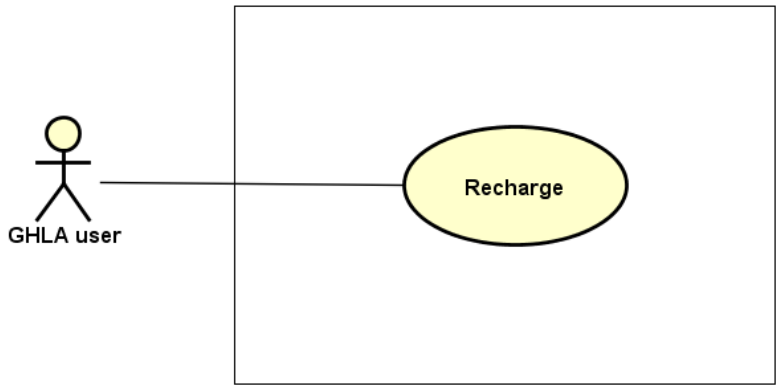


Figure 3.21: Recharge Use case diagram

Use Case ID	**UC08**	**Use Case Name**		**Recharge**	
Author	**TriBM, TuyenLA**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to recharge money				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Select and touch a node on Home Screen			
2	GHLA	Display Control and Schedule node screen as default			
3	User	Select Recharge and Check account tab			
4	GHLA	Display Recharge and Check account screen			
5	User	- Fill card number on text box on the screen - Touch Recharge button			
6	GHLA	Recharge money for the above node through Switchboard			

1.10. UC10 – View node's history

1.10.1. Screen Design

Figure 3.22: View node's history screen

Table 3.10: View note's history Screen definition:

#	Field Name	Type	Mandatory	Max Length	Description
1	Node's history	List view			The list view contains items including actor, description and time.
2	View history	Button			
3	Synchronize node's history	Button			

1.10.2. Use case specification

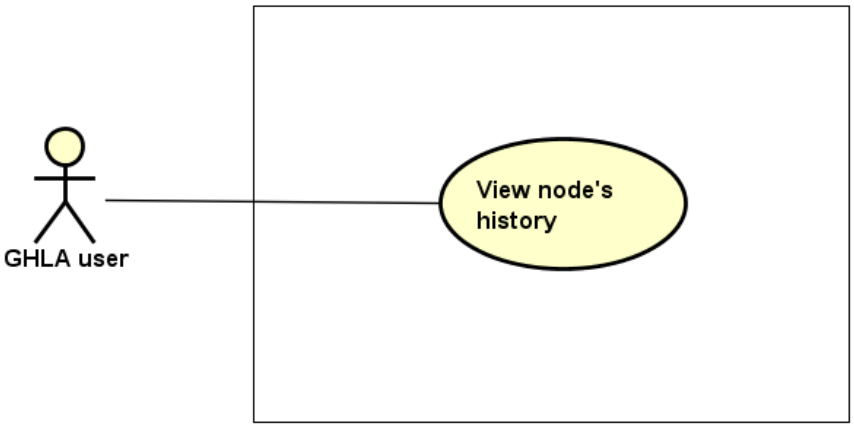


Figure 3.23: View node's history Use case diagram

Use Case ID	**UC10**	**Use Case Name**		**View node's history**	
Author	**ThangLD**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to view history of node they select				
Precondition	There is an Internet connection				
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Select and touch a node on Home Screen			
2	GHLA	Display Control node screen as default			
3	User	Select History tab			
4	GHLA	Display History screen			
5	User	Select and touch View Log button on the screen			
6	GHLA	Load activity history from the database server and display it			

1.11. UC11 – Synchronize nodes

1.11.1. Screen Design

Figure 3.24: Synchronize nodes Screen

Table 3.11: Synchronize nodes Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Synchronize	Popup			Confirm dialog to synchronize nodes on the map

1.11.2. Use case specification

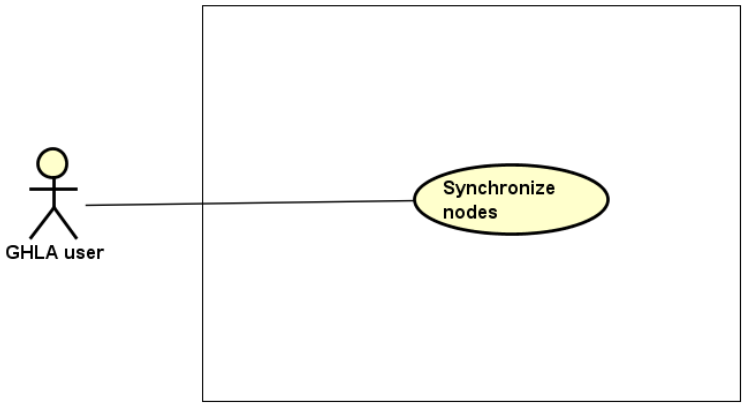


Figure 3.25: Synchronize nodes Use case diagram

Use Case ID	**UC11**	**Use Case Name**		**Synchronize nodes**	
Author	**ThangLD**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows a user to be able to synchronize nodes with the database server.				
Precondition	There is an internet connection				
Trigger					
Post-Condition					

Main flows		
Step	***Actor***	***Action***
1	User	Touch button on top left of the Home Screen
2	GHLA	Display a navigation drawer on the left of the screen
3	User	Select and touch Synchronize nodes item on the drawer
4	GHLA	Display a confirm dialog on the screen
5	User	Select and touch one of the two above buttons
6	GHLA	If user select “Agree”, GHLA will synchronize nodes with database and reload Home page with map. Else, go to step 2.

Alternative flows

AT1	At step 3 in the main flows, if there is no internet connection at the present time,	
Sub step	**Actor**	**Action**
4.1	GHLA	Display Home Screen (Map) with a dialog message “You need have Internet connection to perform this action!” following a button “Ok”
4.2	User	Select and touch “Ok” button
4.3	GHLA	Display Home Screen (Map)

1.12. UC12 – Synchronize node’s history

1.12.1. Screen Design

Figure 3.26: Synchronize node's history Screen

Table 3.12: Synchronize node's history Screen Definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Node's history	List view			The list view contains items including actor, description and time.
2	View history	Button			
3	Synchronize node's history	Button			

1.12.2. Use case specification

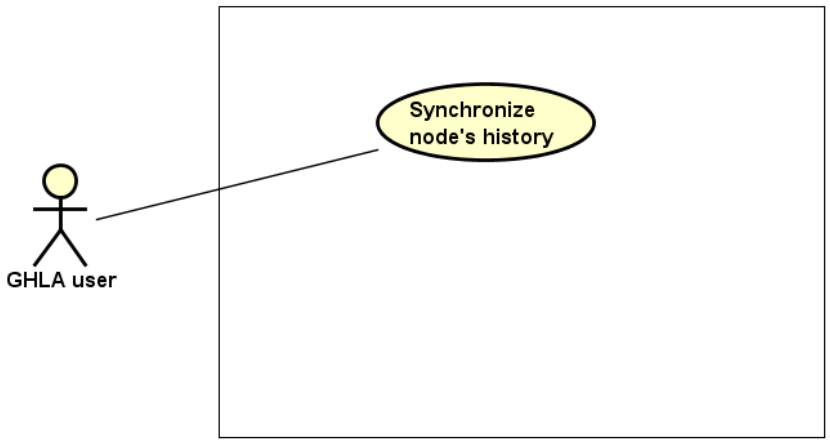


Figure 3.27: Synchronize node's history Use case diagram

Use Case ID	**UC12**	**Use Case Name**		**Synchronize node's history**	
Author	**ThangLD**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows an user to be able to synchronize node's history				
Precondition	There is an internet connection				
Trigger					
Post-Condition					

Main flows		
Step	***Actor***	***Action***
1	User	Select and touch a node on Home Screen
2	GHLA	Display Control node screen as default
3	User	Select History tab
4	GHLA	Display History screen
5	User	Select and touch Synchronize node's history button on the screen
6	GHLA	Synchronize all node's histories that are dependent on the local device and are not dependent on the database server to the database server. Then, display a confirm dialog with message "Synchronize successful!" following a button "Ok"
7	User	Select and touch "Ok" button
8	GHLA	Display History screen

1.13. UC13 – Logout

1.13.1. Screen Design

Figure 3.28: Logout Screen

Table 3.13: Logout Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Confirm	Dialog			Confirm dialog

1.13.2. Use case specification

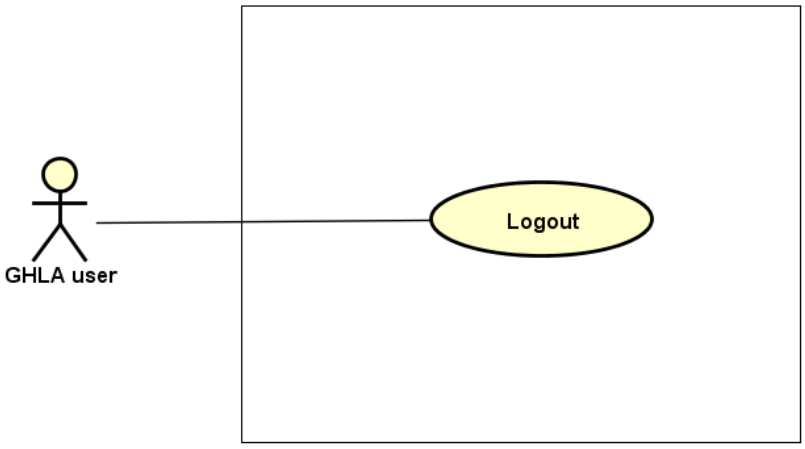


Figure 3.29: Logout Use case diagram

Use Case ID	**UC13**	**Use Case Name**		**Logout**	
Author	**ThangLD**	**Version**	**1.0**	**Date**	**22/06/2017**
Actor	GHLA User				
Description	The function allows user to be able to logout the application				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Touch button on top left of the Home Screen			
2	GHLA	Display a navigation drawer on the left of the screen			
3	User	Select and touch Logout item on the drawer			
4	GHLA	Display confirm dialog in the screen			
5	User	Select “Agree” or “Disagree” button and touch it			
6	GHLA	If user select “Agree” button, logout user from the application Else, go to step 2			

2. *Non- Functional Requirement Specification*

Table 3.14: Nonfunctional requirements for GHLA

#	Nonfunctional requirements	Description
1	Security	Users have to sign in with Google's accounts
2	Availability	The user session timed out is dependent on Google
3	Usability	Language: Vietnamese

II. Green Hoa Lac Web Management System

1. Functional Requirement Specification

1.1. UC01 – Login

1.1.1. Screen Design

Figure 3.30: Login Screen

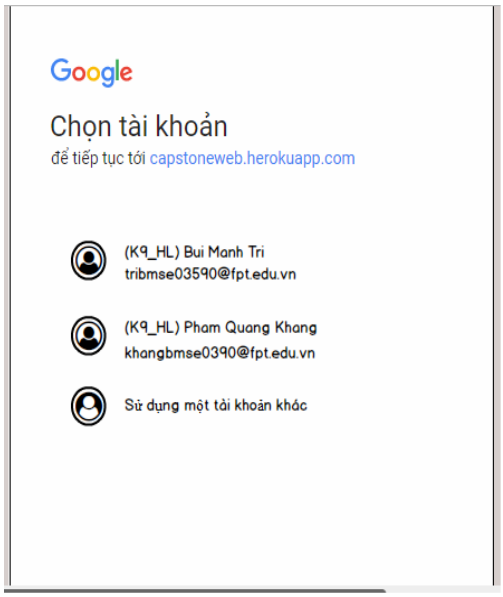


Figure 3.31: Google Sign in Screen

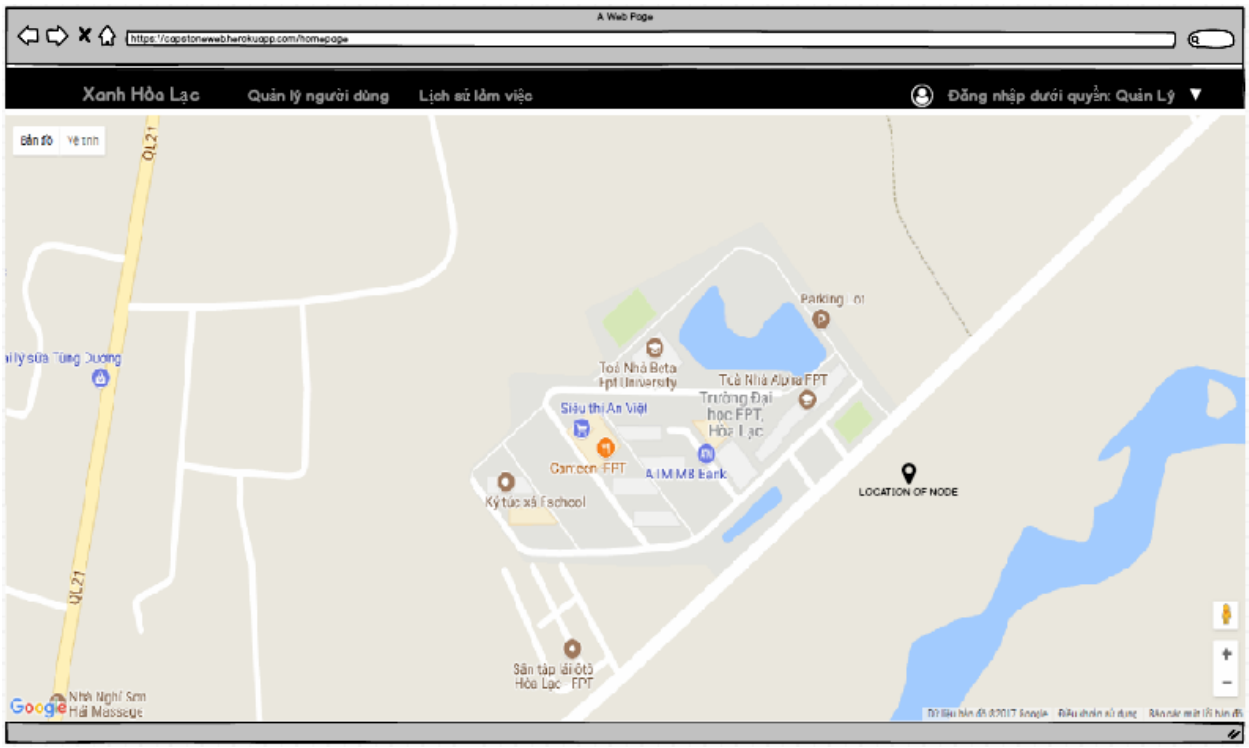


Figure 3.32: Home Page

Table 3.15: Login Screen Definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Login	Button			Navigate to Google Sign In Page.

1.1.2. Use case specification

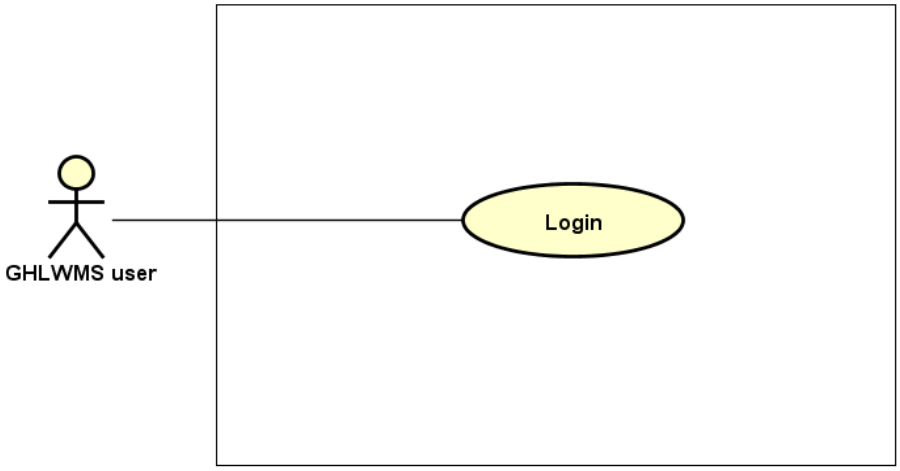


Figure 3.33: Login Use Case Diagram

Use Case ID	UC01	Use Case Name		Login	
Author	KhangPQ	Version	1.0	Date	16/06/2017
Actor	GHLMS User				
Description	The function allows users to be able to login in the website when he/she have had an account and his/her account is still active (or not blocked).				
Precondition					
Trigger					

Post-Condition		
Main flows		
Step	***Actor***	***Action***
1	User	Type URL: https://capstoneweb.herokuapp.com into location field of internet browser and then press enter
2	GHLWMS	Display Login screen with the following field: - Login button
3	User	Click on Login button.
4	GHLWMS	Navigate to Google Sign In Page
5	User	Choose an account in list Google accounts or Enter other email and password for other account.
6	GHLWMS	Validate the account and then display Home screen

Alternative flows

AT1	At step 5 in the main flows**,** if users enter wrong email account or email's password,	
Sub step	**Actor**	**Action**
5.1	GHLWMS	Display an error with message "Couldn't find your Google Account" or "Wrong password. Try again", try again step 5.

AT2	At step 5 in the main flows**,** although email account exists, it is blocked or in role "member" or does not exist in database.	
Sub step	**Actor**	**Action**
5.1	GHLWMS	Return step 2.

Business Rules

#	Rule Description
BR04	GHLWMS user includes 2 types of role: Admin and Manager.
BR05	There are 3 types of users which are managed in GHLWMS including Admin, Manager and Member

1.2. UC02 – View user's information

1.2.1. Screen Design

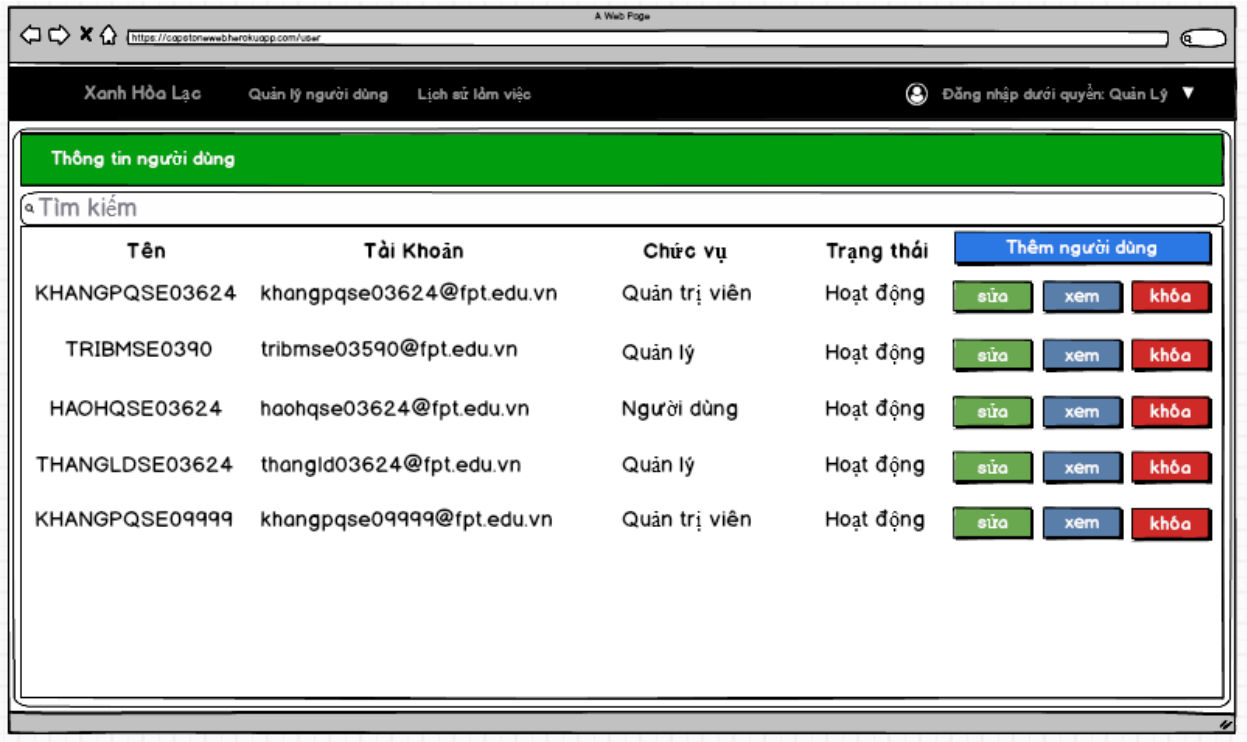


Figure 3.34: View user's information Screen

Table 3.16: View user information screen information

#	Field Name	Type	Mandatory	Max Length	Description
1	List all users	Table			Containing 5 columns: username, account, role, status, action.

1.2.2. Use case specification

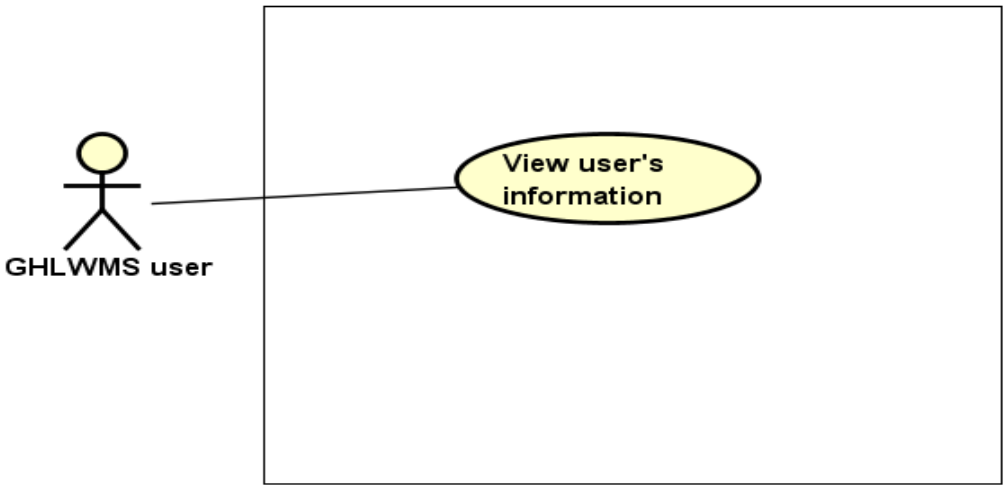


Figure 3.35: View user's information Use case diagram

Use Case ID	**UC02**	**Use Case Name**		**View user's information**	
Author	**LinhNTL**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to view user's information in GHLWMS with the following fields: username, account, role or status				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click User management on the navigation bar			
2	GHLWMS	Display list all accounts in User management page in a table with the following columns groups: - (Username, account, role, status) - (Update role, view log, block)			

1.3. UC03 – Add user

1.3.1. Screen Design

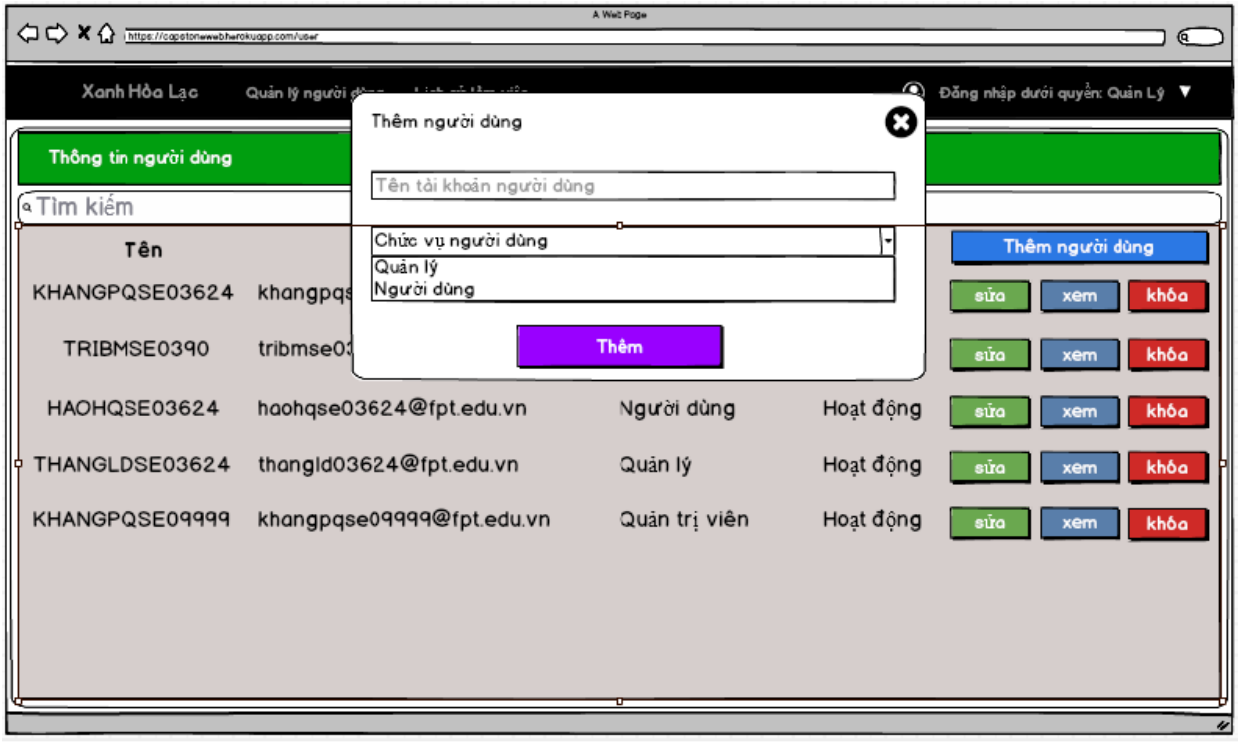


Figure 3.36: Add user screen

Table 3.17: Add user Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	User name	Text box	Yes		mail
2	Role of user	Drop down list	Yes		There 2 types of role: - Member - Manager
3	Add	Button			

1.3.2. Use case specification

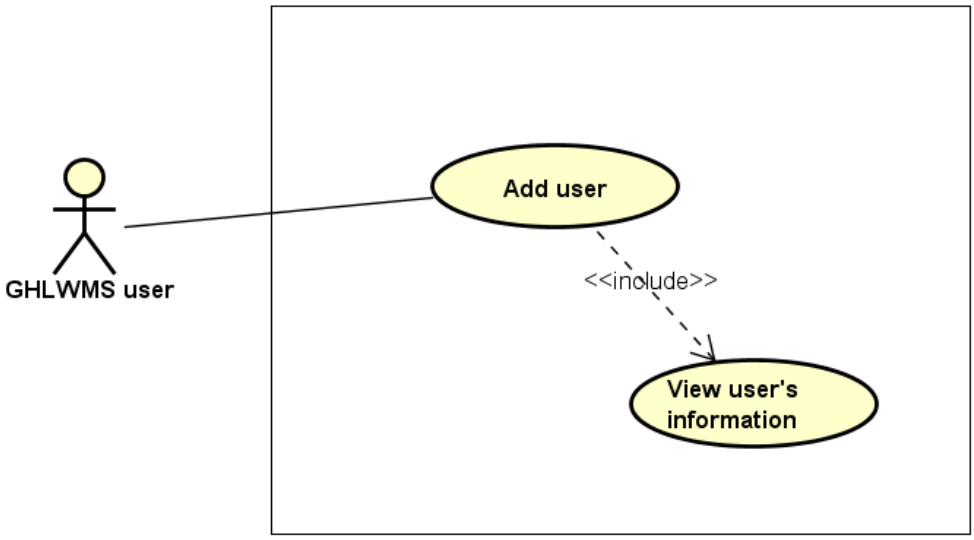


Figure 3.37: Add user Use case diagram

Use Case ID	**UC03**	**Use Case Name**		**Add user**	
Author	**LinhNTL**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows users to be able to add accounts to GHLWMS.				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click User management on the navigation bar			
2	GHLWMS	Display User management page			
3	User	Click Add button on right top of the screen.			
4	GHLWMS	Display Add user popup with the following fields:			

		- User name (text box) - Type of role (drop down list) - Add (button)
5	User	- Enter User name - Select type of role - Click on Add button
6	GHLWMS	- Validate User name, role and then add to the database - Display message “Add new user successfully” - Hide the popup and reload list users in User management page

Alternative flows

AT1	At step 4 in the main flows**,** if users do not enter email account or select type of role	
Sub step	**Actor**	**Action**
4.1	GHLWMS	Display an error with message "User name or role must be filled”, try again step 4

AT2	At step 6 in the main flows**,** if users enter wrong syntax of email account	
Sub step	**Actor**	**Action**
6.1	GHLWMS	Display an error with message "Wrong username”, try again step 4.

Business Rules

#	Rule Description
BR06	Domain of mail is one of the following: - Gmail.com - Fpt.edu.vn - Fe.edu.vn - Gmail.com.vn

1.4. UC04 – Search user's information

1.4.1. Screen Design

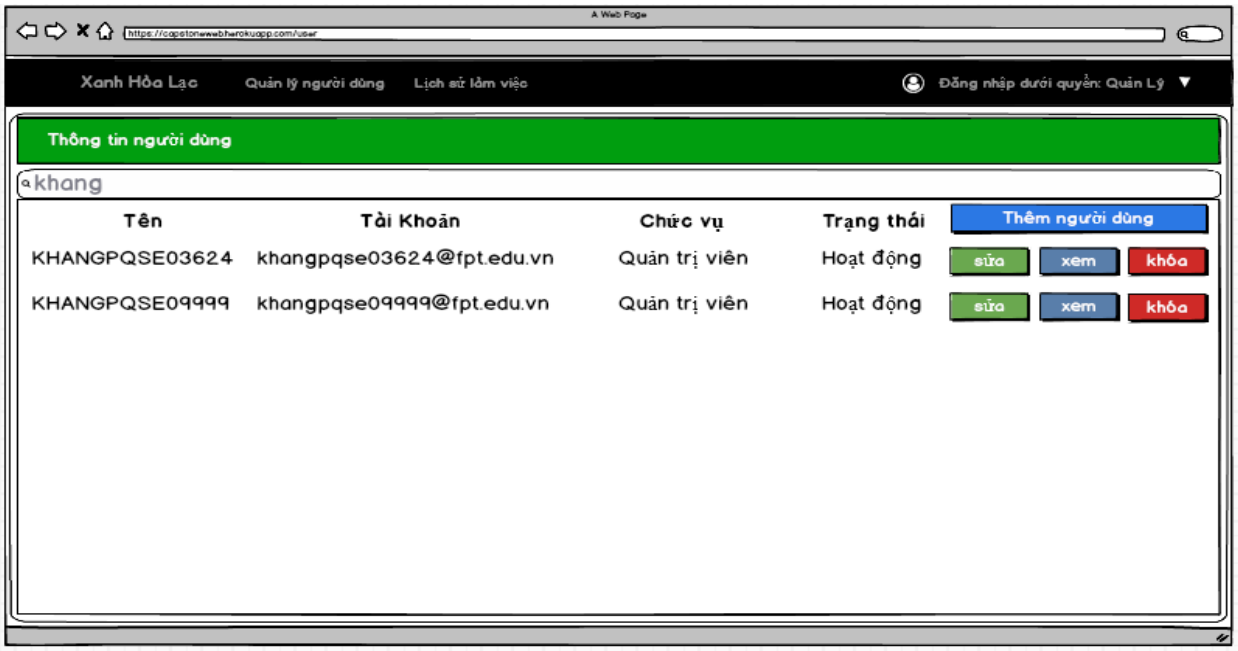


Figure 3.38: Search user's information screen

Table 3.18: Search user's information Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Filter	Text box			Filter by username account, role or status.
2	List user's information	Table			The table contains 5 columns: username, account, role, status and action (update role, view log, block)

1.4.2. Use case specification

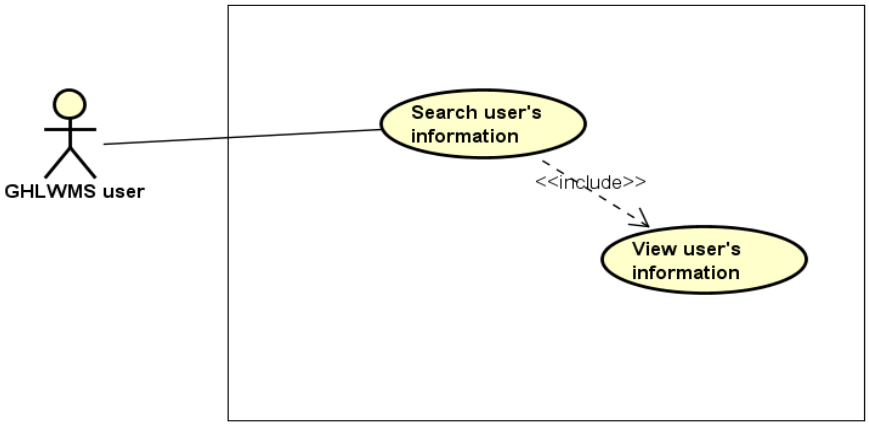


Figure 3.39: Search user's information Use case diagram

Use Case ID	UC04	Use Case Name		Search user's information	
Author	LinhNTL	Version	1.0	Date	16/06/2017
Actor	GHLWMS User				

Description	The function allows an user to be able to search user ’s information in GHLWMS by username, account, role or status
Precondition	
Trigger	
Post-Condition	

Main flows		
Step	***Actor***	***Action***
1	User	Click User management on the navigation bar
2	GHLWMS	Display User management page
3	User	Enter information in Filter text box.
4	GHLWMS	Display list accounts that satisfy the above information in the table with the following field groups: - (Username, account, role, status) - (Update role, view log, block)

Alternative flows

AT1	At step 3 in the main flows**,** if users enter characters that do not match any information	
Sub step	**Actor**	**Action**
3.1	GHLWMS	Display “No result is found”.

1.5. UC05 – Update user's role

1.5.1. Screen Design

Figure 3.40: Update user's role screen

Table 3.19: Update user's role Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Type of role	Drop down list			There are 2 type of roles: - Manager - Member
2	Update	Button			

1.5.2. Use case specification

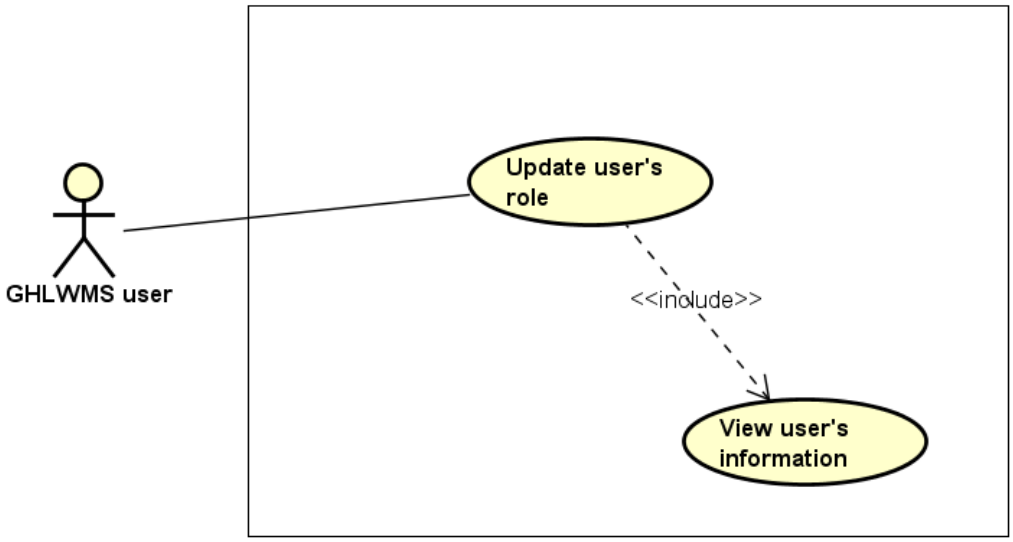


Figure 3.41: Update user's role Use case diagram

Use Case ID	**UC05**	**Use Case Name**		**Update user's role**	
Author	**LinhNTL**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to update other user's role				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click User management on the navigation bar			
2	GHLWMS	Display User management page			
3	User	Select an account and click Update role button			

4	GHLWMS	Display Update role popup
5	User	- Select a role from the drop down list role - Click Update button
6	GHLWMS	- Validate and update role in the database - Display message “Update successfully” in popup - Hide popup, reload list of users in User management page

Alternative flows

AT1	At step 6 in the main flows**,** if users selected account of their own or account of others who are in the same role or Admin	
Sub step	**Actor**	**Action**
6.1	GHLWMS	Display error with message “Access denied”

AT2	At step 6 in the main flows**,** if users did not select a role in the drop down list and then click “Update” button	
Sub step	**Actor**	**Action**
6.1	GHLWMS	Display error with message “Role must be chosen”

Business Rules

#	***Rule Description***
BR07	GHLWMS users cannot update user’s role in some cases: - Update their own role - Update other user’s role who has the same role with user working now. - Update role of Admin

1.6. UC06 – View user's history

1.6.1. Screen Design

Figure 3.42: View user's history screen

Table 3.20: View user's history Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	List 5 previous activities of the account	Popup			Content of list is divided into 4 main columns: target node, activity, description, time

1.6.2. Use case specification

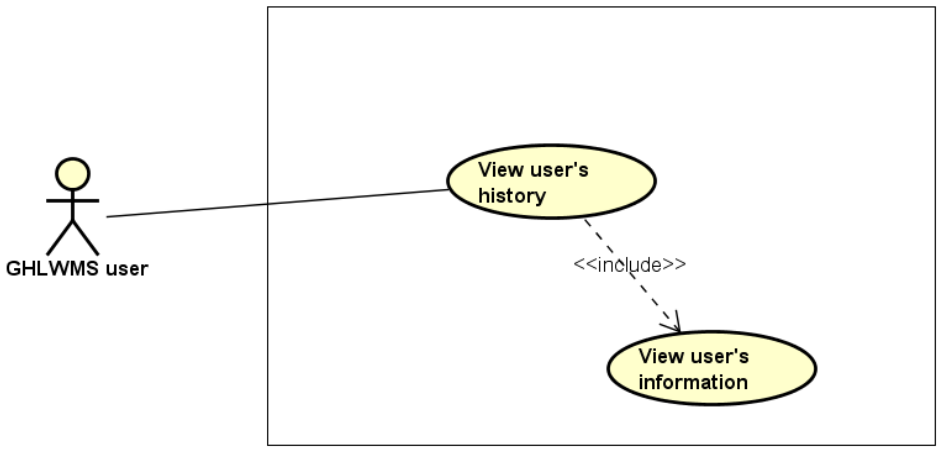


Figure 3.43: View user's history Use case diagram

Use Case ID	**UC06**	**Use Case Name**		**View user's history**	
Author	**LinhNTL**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows a user to be able to view activities history of all users.				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click User management on the navigation bar			
2	GHLWMS	Display User management page			
3	User	Select an account and click View history button			
2	GHLWMS	Display list 5 previous activities with information about: (target node, action, description, time) of the account in a popup			

1.7. UC07 – Lock user

1.7.1. Screen Design

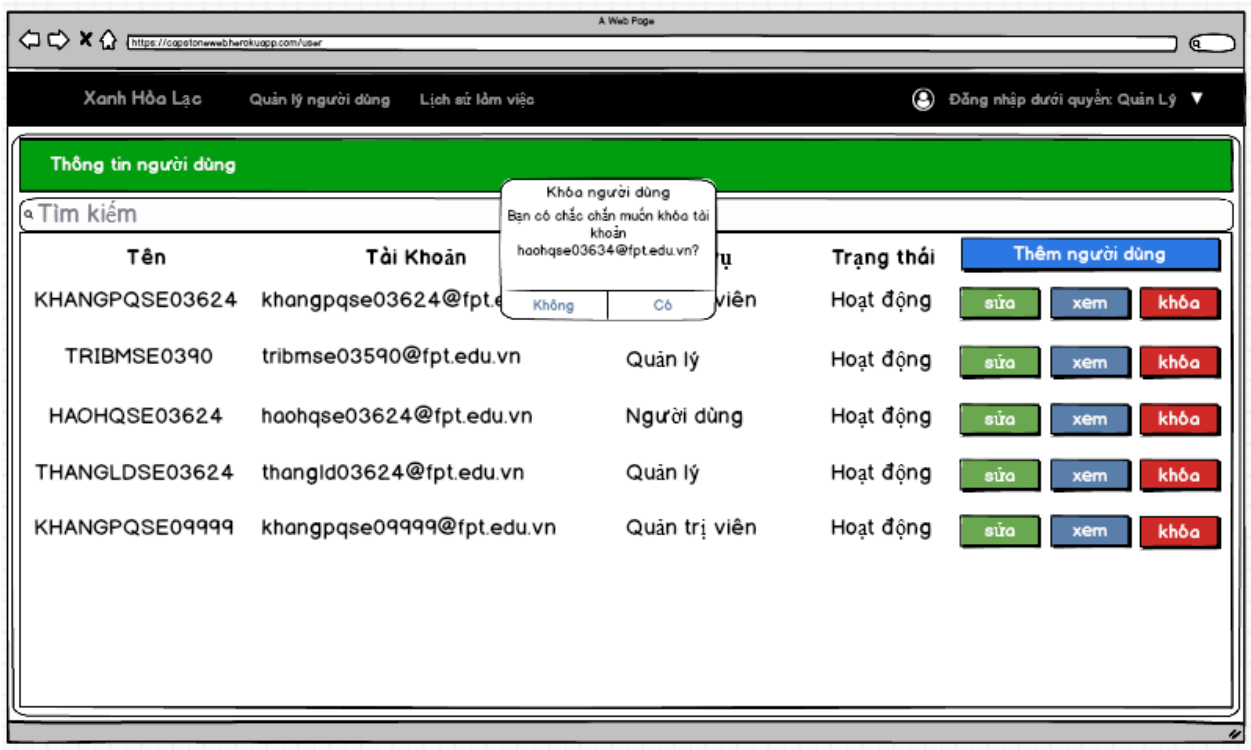


Figure 3.44: Lock user screen

Table 3.21: Log user Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Lock user	Dialog			Confirm dialog

1.7.2. Use case specification

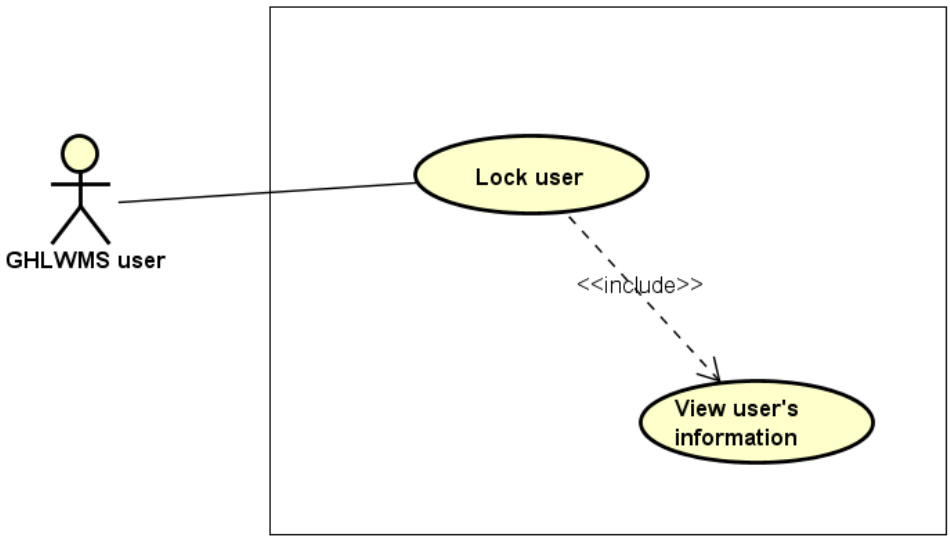


Figure 3.45: Lock user Use case diagram

Use Case ID	**UC07**	**Use Case Name**		**Lock user**	
Author	**LinhNTL**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to lock other user				
Precondition	Login GHLWMS successfully				
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click User management on the navigation bar			
2	GHLWMS	Display User management page			
3	User	Select an account and click Lock button			
4	GHLWMS	Display Confirm to lock user popup			
5	User	Click button Agree/ Disagree on the popup			

6	GHLWMS	If click “Disagree”, stay at User management page Else: - Validate and update status in the database - Display message “Lock the user successfully” in popup - Hide popup, reload list of users in User management page

Alternative flows

AT1	At step 6 in the main flows**,** if users selected account of their own or account of others who are in the same role or Admin	
Sub step	**Actor**	**Action**
6.1	GHLWMS	Display error with message “Access denied”

Business Rules

#	***Rule Description***
BR08	GHLWMS users cannot lock an account in some cases: - Lock their own accounts - Lock other user’s role who has the same role with user working now. - Lock role of Admin

1.8. UC08 – Unlock user

1.8.1. Screen Design

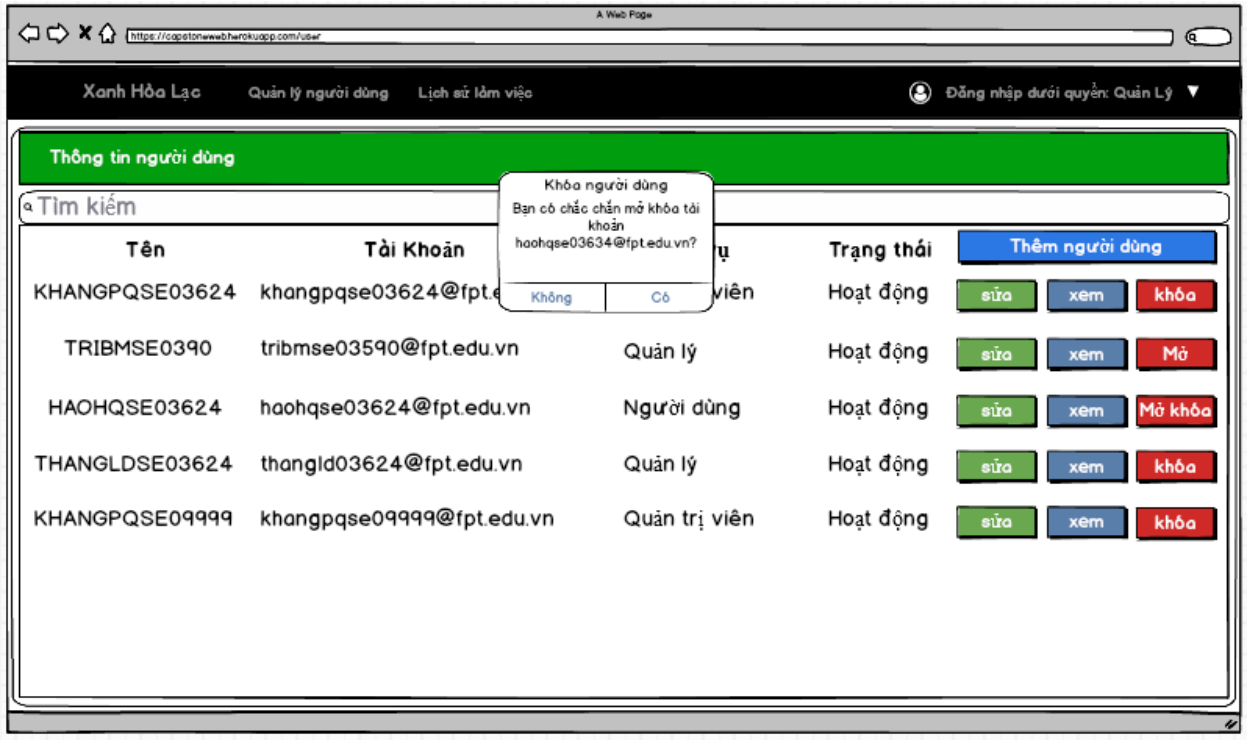


Figure 3.46: Unlock user screen

Table 3.22: Unlock user Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Unlock user	Dialog			Confirm dialog

1.8.2. Use case specification

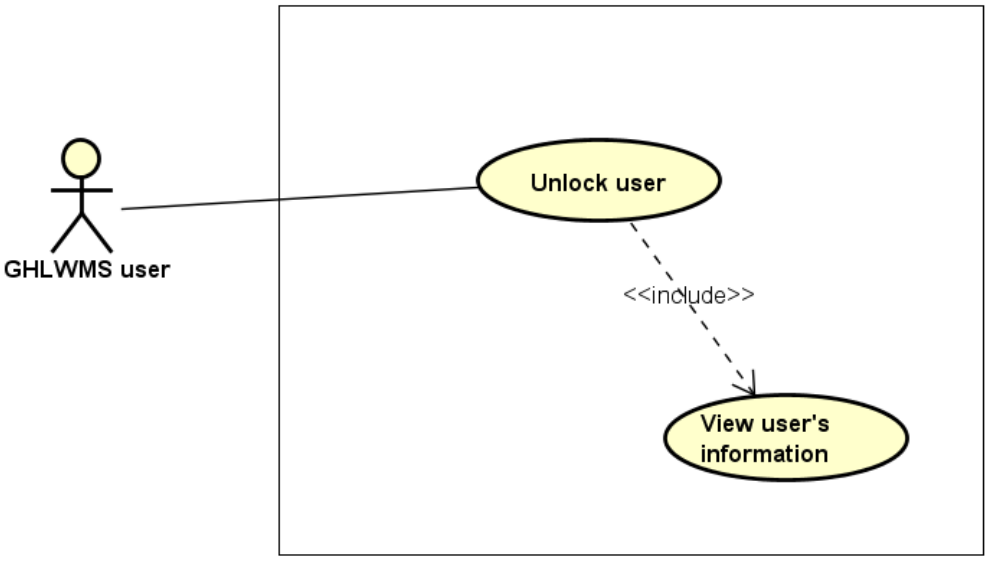


Figure 3.47: Unlock user Use case diagram

Use Case ID	**UC08**	**Use Case Name**		**Unlock user**	
Author	**LinhNTL**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to unlock other user				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click User management on the navigation bar			
2	GHLWMS	Display User management page			
3	User	Select an account and click Unlock button			
4	GHLWMS	Display Confirm to unlock user popup			
5	User	Click button Agree/ Disagree on the popup			

6	GHLWMS	If click “Disagree”, stay at User management page Else: - Validate and update status in the database - Display message “Unlock the user successfully” in popup - Hide popup, reload list of accounts in User management page

Business Rules

#	*Rule Description*
BR09	GHLWMS users can unlock an account in some cases: - Admin can unlock all type of account. - Manager can unlock other manager’s accounts and other members

1.9. UC09 – View activity history

1.9.1. Screen Design

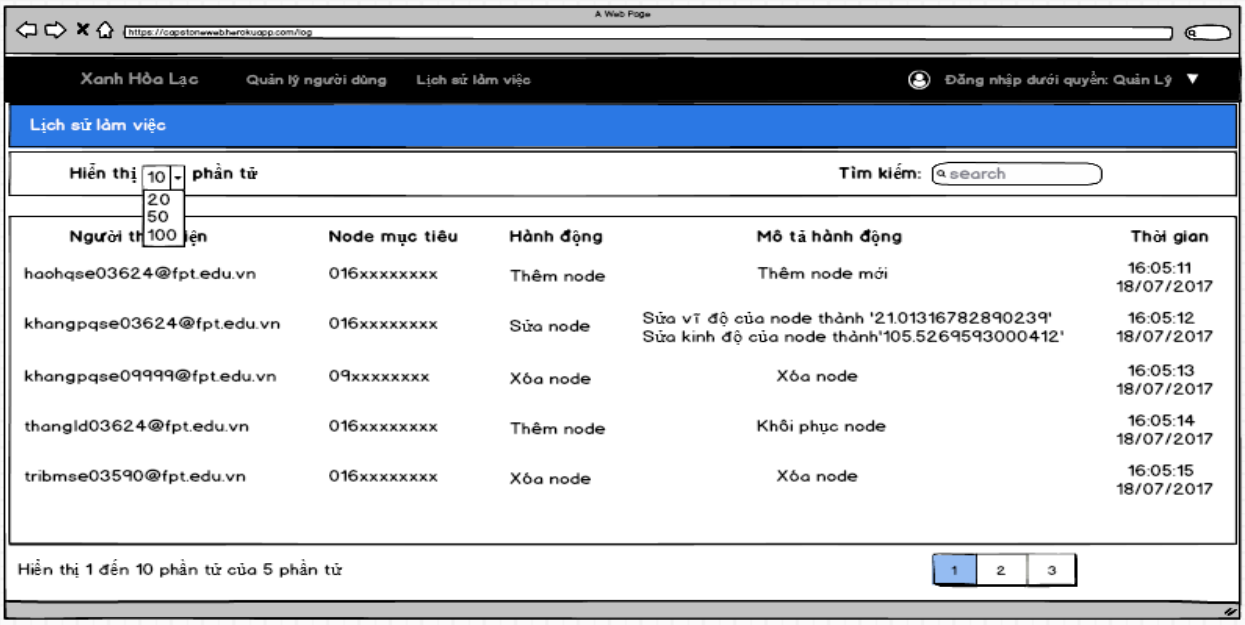


Figure 3.48: View activity history screen

Table 3.23: View activity history Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	List activity history	Table			The table contains 5 columns: user, target node, activity, description, time

1.9.2. Use case specification

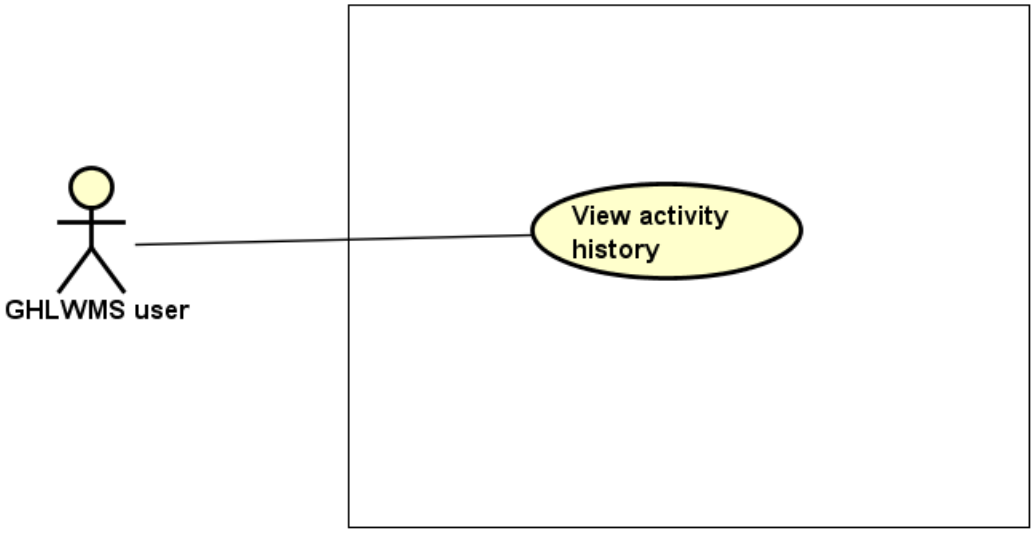


Figure 3.49: View activity history Use case diagram

Use Case ID	UC09	Use Case Name		View activity history	
Author	LinhNTL	Version	1.0	Date	16/06/2017
Actor	GHLWMS User				
Description	The function allows an user to be able to view activity history in GHLWMS with the following fields: user, target node, action, description, time				

Precondition		
Trigger		
Post-Condition		
Main flows		
Step	***Actor***	***Action***
1	User	Click Activity history on the navigation bar
2	GHLWMS	Display Activity history page with a table which has default top 10 previous actions.

Business Rules

#	***Rule Description***
BR10	GHLWMS will automatically delete each activity history of node after 30 days created.

1.10. UC10 – Search activity history

1.10.1. Screen Design

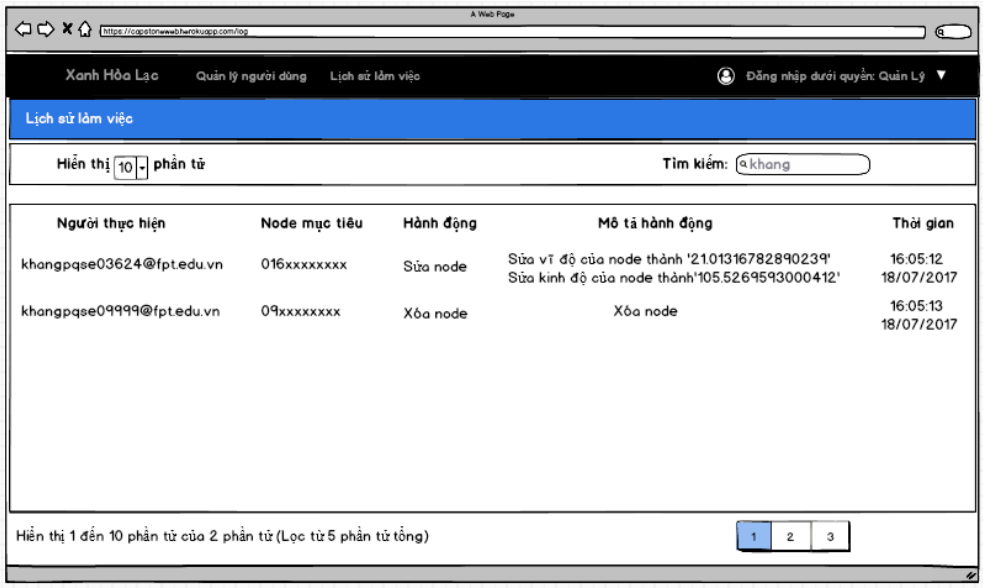


Figure 3.50: Search activity history screen

Table 3.24: Search activity history Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Search	Text field			Filter by user, target node, action, description, time.
2	List activity history	Table			The table contains 5 columns: user, target node, activity, description, time

1.10.2. Use case specification

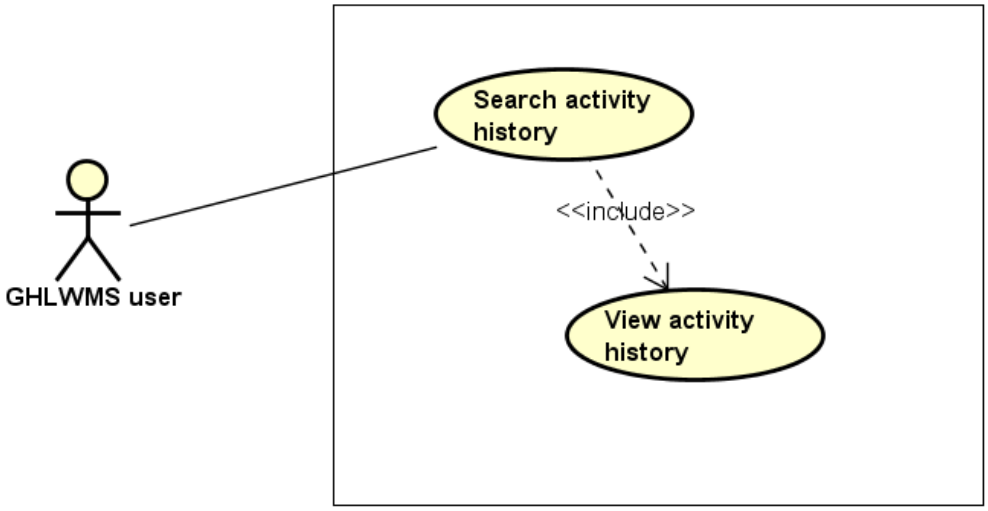


Figure 3.51: Search activity history Use case diagram

Use Case ID	**UC10**	**Use Case Name**		**Search activity history**	
Author	**LinhNTL**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to search activity history in GHLWMS by user, target node, action, description, time				
Precondition					
Trigger					
Post-Condition					

Main flows		
Step	***Actor***	***Action***
1	User	Click Activity history on the navigation bar
2	GHLWMS	Display Activity history page
3	User	Enter information in Search field
4	GHLWMS	Display list histories that satisfy the above information in the table with the following fields: user, target node, action, description, time

Alternative flows

AT1	At step 3 in the main flows**,** if users enter characters that do not match any information	
Sub step	**Actor**	**Action**
3.1	GHLWMS	Display “No result is found”.

1.11. UC11 – Add new node

1.11.1. Screen Design

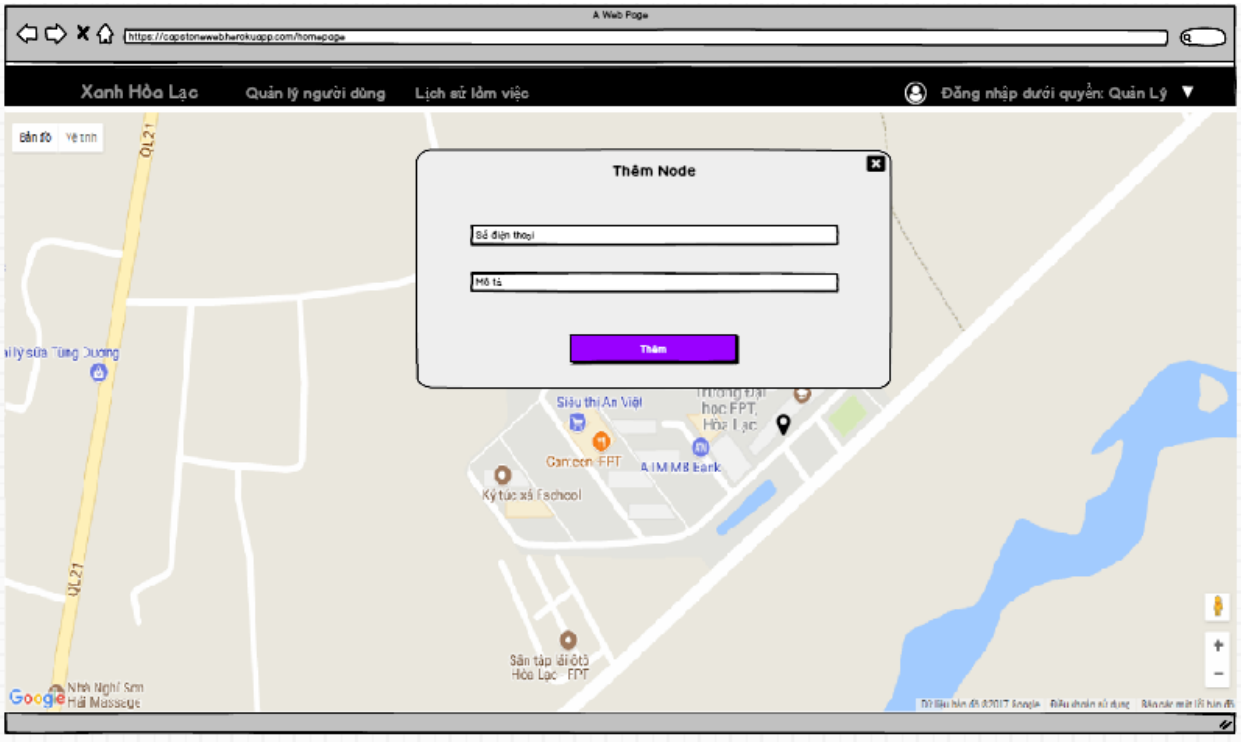


Figure 3.52: Search activity history Use case diagram

Table 3.25: Add new node Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Telephone number	Text field	Yes	15	Must be a string of number
2	Description	Text field	No		
3	Add	Button			

1.11.2. Use case specification

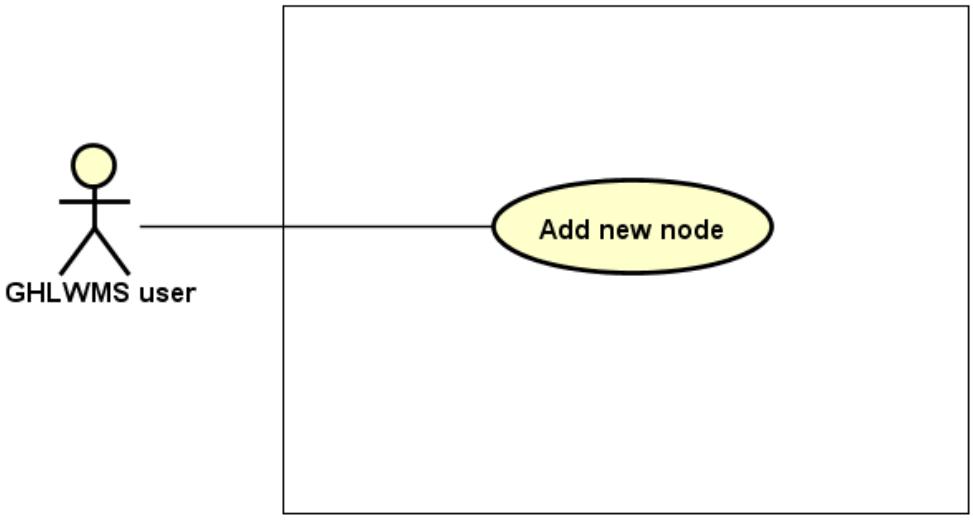


Figure 3.53: Add new node Use case diagram

Use Case ID	**UC11**	**Use Case Name**		**Add new node**	
Author	**KhangPQ**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows a user to be able to add new node to GHLWMS.				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click Green Ho La in the navigation bar			
2	GHLWMS	Display Green Ho La page by a map with nodes (also is home page by default when login)			
3	User	Select a position for new node and click there			
4	GHLWMS	Display Add node popup with the following fields:			

		- Telephone number (text field) - Description (text field) - Add (button)
5	User	- Enter telephone number - Enter description - Click on Add button
6	GHLWMS	- Validate telephone number and then add new node to the database - Display message “Add new node successfully” - Hide the popup and reload “Green Ho La” page (or home page)

Alternative flows

AT1	At step 6 in the main flows**,** if telephone number is invalid syntax,	
Sub step	**Actor**	**Action**
6.1	GHLWMS	Display an error with message "wrong phone number”, try again step 5

Business Rules

#	***Rule Description***
BR11	Telephone number must be in the following syntax: - First number is 0 or the two first numbers is 84 - If first number is 0: there must have minimum 10 numbers after the first - If the two first numbers is 84: there must have minimum 11 numbers after the two first
BR12	Does not exist 2 nodes that have the same location.

1.12. UC12 – Update node's position

1.12.1. Screen Design

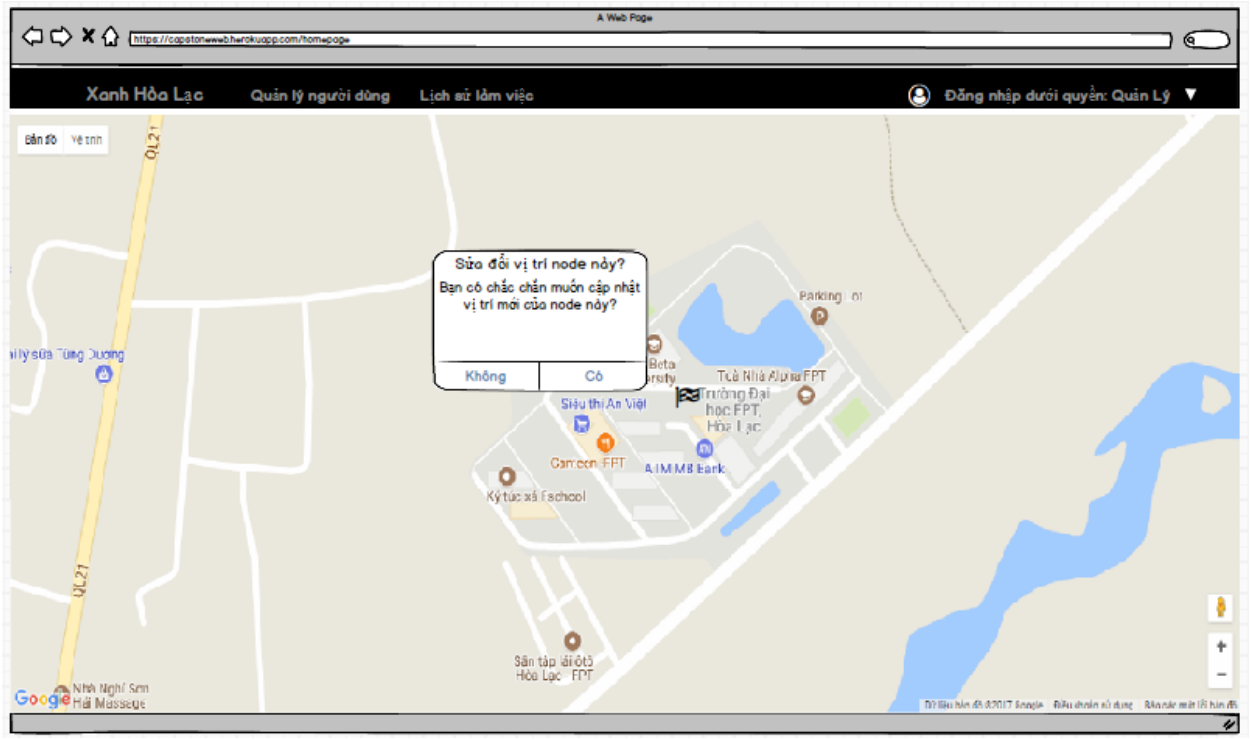


Figure 3.54: Update node's position screen

Table 3.26: Update node position Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Update node's position	Dialog			Confirm dialog

1.12.2. Use case specification

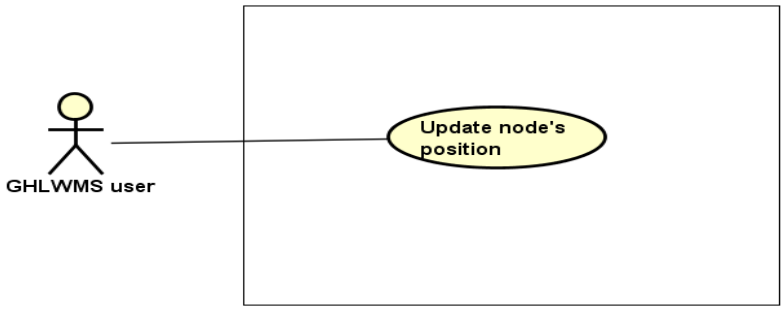


Figure 3.55: Update node's position Use case diagram

Use Case ID	**UC12**	**Use Case Name**		**Update node's position**	
Author	**KhangPQ**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows a user to be able to update node's position in GHLWMS.				
Precondition	Exist at least one node in GHLWMS				
Trigger					
Post-Condition					

Main flows		
Step	***Actor***	***Action***
1	User	Click Green Ho La in the navigation bar
2	GHLWMS	Display Green Ho La page by a map with nodes (also is home page by default when login)
3	User	Select a node, move the node to new position in the map.
4	GHLWMS	Display "Confirm to update position of node" with the following fields: - Agree button - Disagree button
5	User	Click button Agree/ Disagree on the popup
6	GHLWMS	- If users enter "Disagree" button, hide the popup - Else : + Update new position of the node in the database + Display "Update node's position successfully" + Stay at "Green Ho La" page (or home page)

1.13. UC13 – View node's information

1.13.1. Screen Design

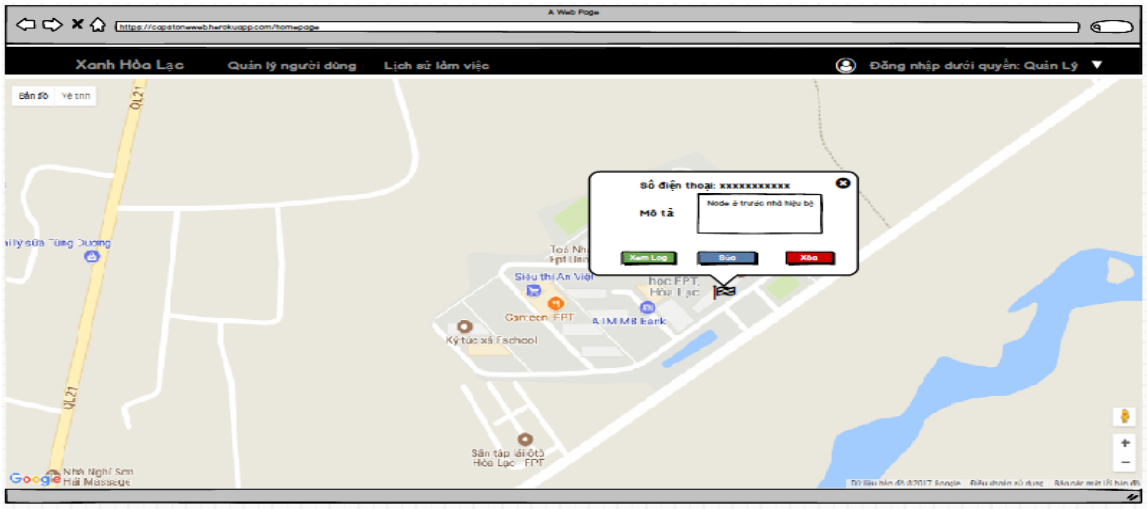


Figure 3.56: View node's information screen

Table 3.27: View node's information Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Telephone number	Text			
2	Description	Text box			
3	View history	Button			
4	Update	Button			
5	Delete	Button			

1.13.2. Use case specification

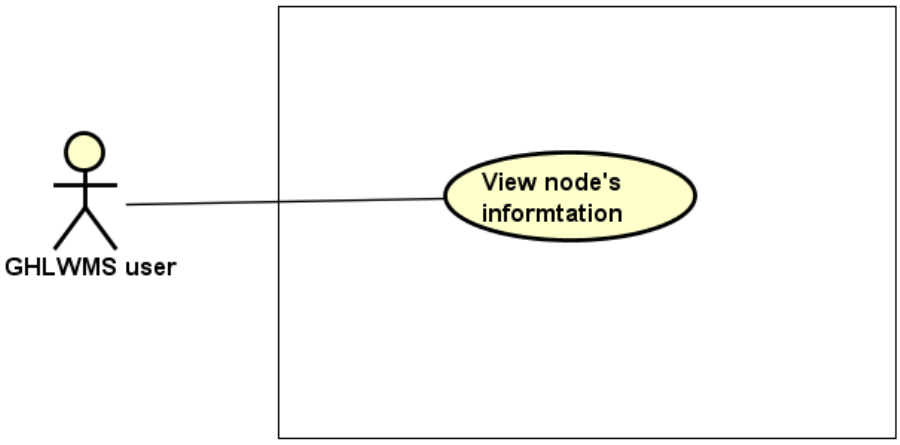


Figure 3.57: View node's information Use case diagram

Use Case ID	**UC13**	**Use Case Name**		**View node's information**	
Author	**KhangPQ**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows a user to be able to view information of a node including telephone number and description.				
Precondition	Exist at least one node in GHLWMS				
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click Green Ho La in the navigation bar			
2	GHLWMS	Display Green Ho La page by a map with nodes (also is home page by default when login)			
3	User	Select a node and click the node.			
4	GHLWMS	Display Node's information window			

1.14. UC14 – View node's history

1.14.1. Screen Design

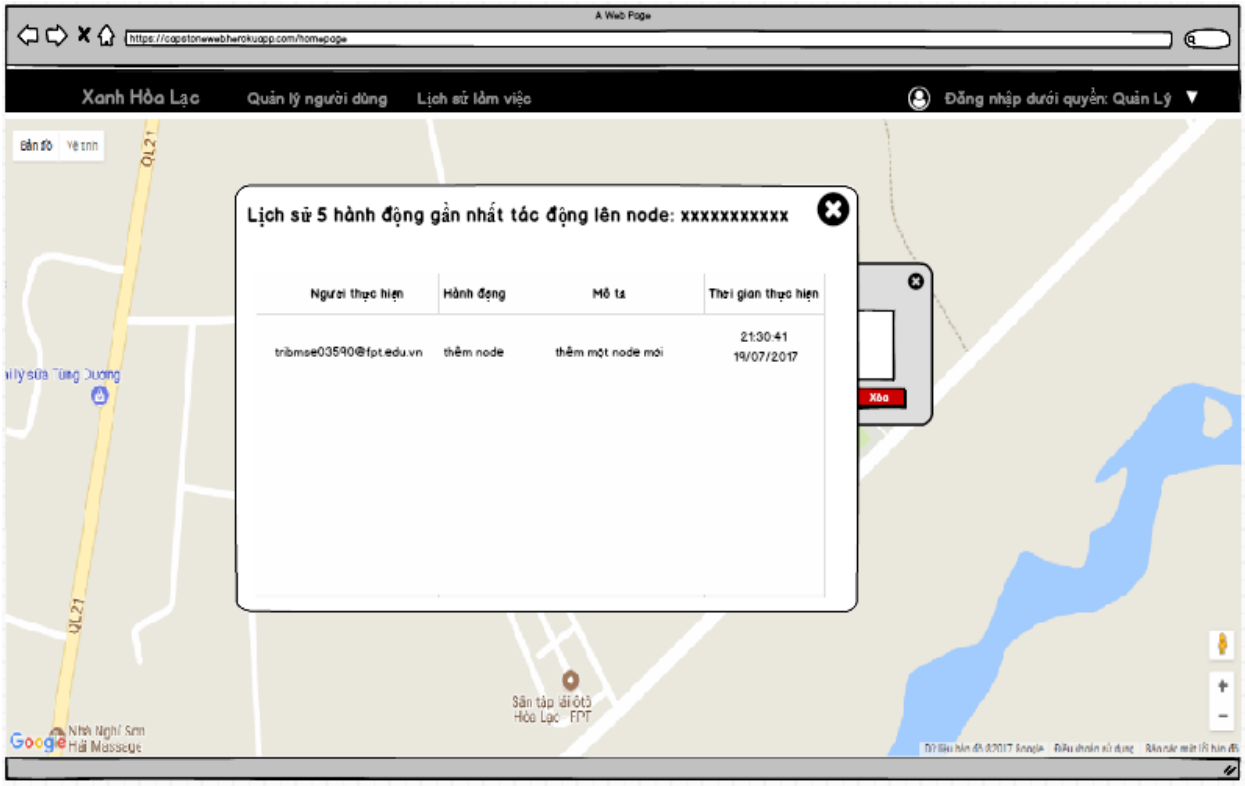


Figure 3.58: View node's history screen

Table 3.28: View node's history screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Top 5 previous action in the node	Popup			It contains 4 columns: users, action, description, and time

1.14.2. Use case specification

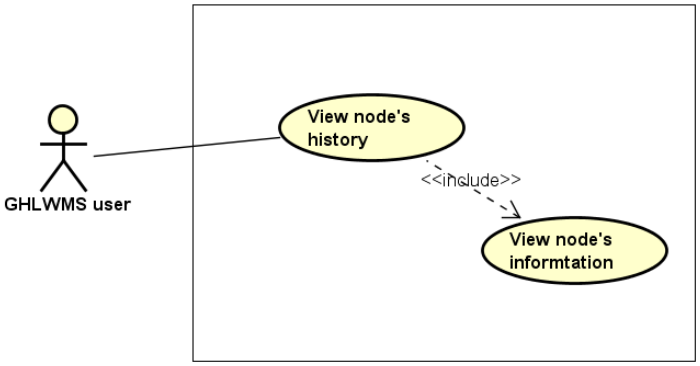


Figure 3.59: View node's history Use case diagram

Use Case ID	**UC14**	**Use Case Name**		**View node's history**	
Author	**KhangPQ**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows a user to be able to view list 5 previous the previous node's activities in GHLWMS.				
Precondition	Exist at least one node in GHLWMS				
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click Green Ho La in the navigation bar			
2	GHLWMS	Display Green Ho La page by a map with nodes (also is home page by default when login)			
3	User	Select a node and click the node.			
4	GHLWMS	Display Node's information window			
5	User	Click View log button in the window			
6	GHLWMS	Display List node's activities popup with a table which contains 4 columns: user, action, description, time			

1.15. UC15 – Update node's information

1.15.1. Screen Design

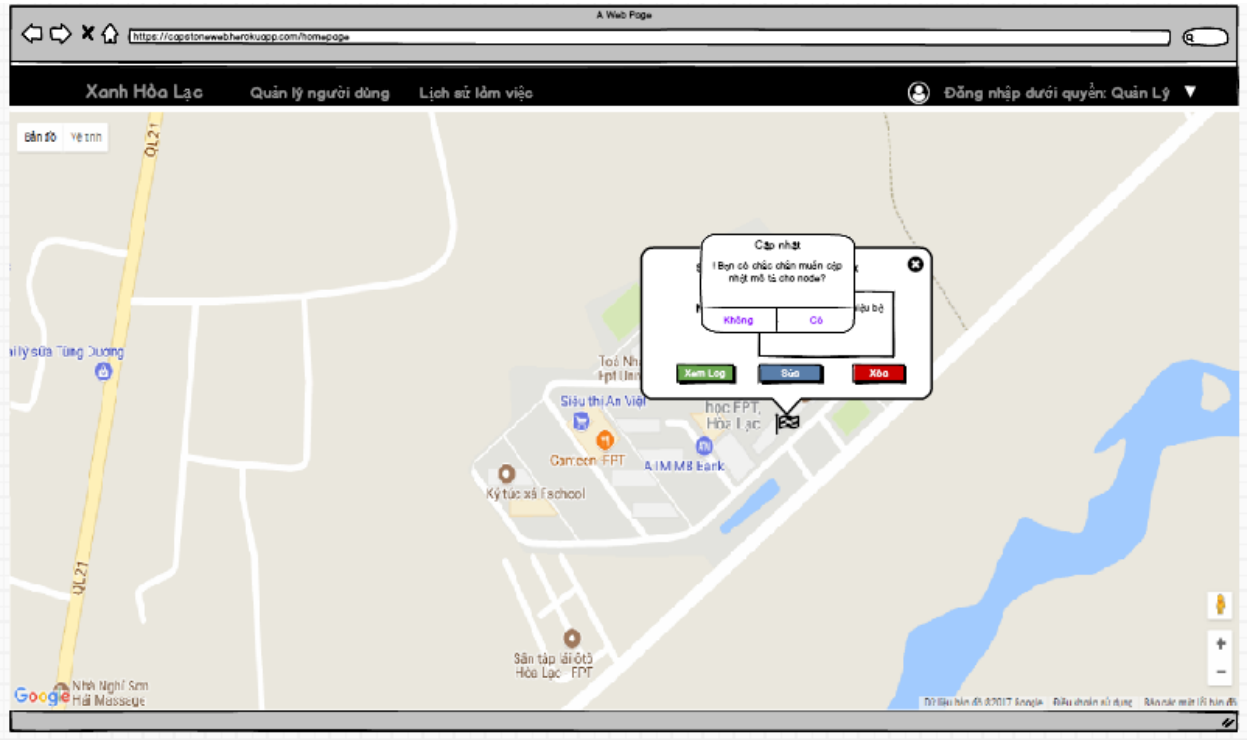


Figure 3.60: Update node's information screen

Table 3.29: Update node's information Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Update node's information	Dialog			Confirm dialog

1.15.2. Use case specification

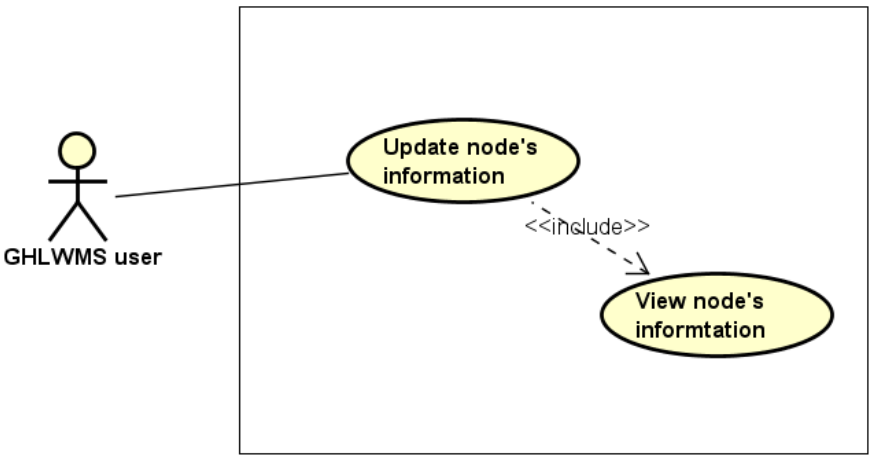


Figure 3.61: Update node's information Use case diagram

Use Case ID	**UC15**	**Use Case Name**		**Update node's information**	
Author	**KhangPQ**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to update node's information				
Precondition	Exist at least one node in GHLWMS				
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click Green Ho La in the navigation bar			
2	GHLWMS	Display Green Ho La page by a map with nodes (also is home page by default when login)			
3	User	Select a node and click the node.			
4	GHLWMS	Display Node's information window			

5	User	- Enter new description in Description text box in the window - Click Update button in the window
6	GHLWMS	Display Confirm to update node's information popup
7	User	Click "Agree" or "Disagree" button the confirm popup
8	GHLWMS	- If click "Agree" button, the new node's information is updated in the database - Else, return step 4

Alternative flows

AT1	At step 5 in the main flows**,** if click "Update" button without enter new description	
Sub step	**Actor**	**Action**
5.1	GHLWMS	Return step 4

1.16. UC16 – Delete node

1.16.1. Screen Design

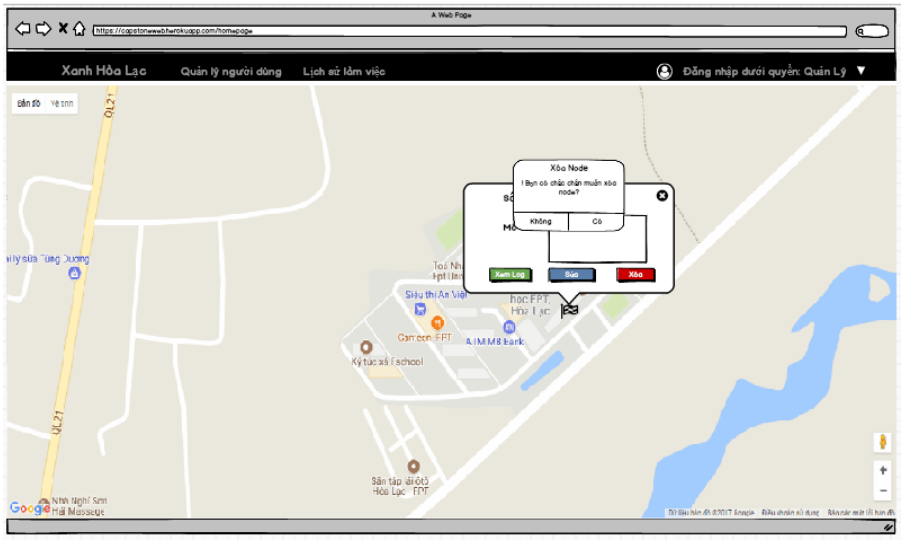


Figure 3.62: Delete node screen

Table 3.30: Delete node Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Delete node	Dialog			Confirm dialog

1.16.2. Use case specification

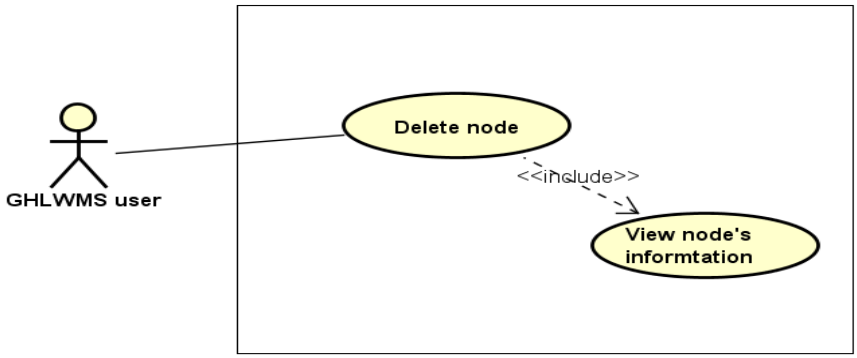


Figure 3.63: Delete node Use case diagram

Use Case ID	UC16	Use Case Name		Delete node	
Author	KhangPQ	Version	1.0	Date	16/06/2017
Actor	GHLWMS User				
Description	The function allows an user to be able to delete node				
Precondition	Exist at least one node in GHLWMS				
Trigger					
Post-Condition					
Main flows					
Step	*Actor*	*Action*			
1	User	Click Green Ho La in the navigation bar			
2	GHLWMS	Display Green Ho La page by a map with nodes (also is home page by default when login)			

3	User	Select a node and click the node.
4	GHLWMS	Display Node's information window
5	User	- Enter new description in Description text box in the window - Click Delete button in the window
6	GHLWMS	Display Confirm to delete the node popup
7	User	Click "Agree" or "Disagree" button the confirm popup
8	GHLWMS	- If click "Agree" button, update status of the node (IsDeleted = true) in the Database - Hide the node in the map - Else, return step 4

Business Rules

#	***Rule Description***
BR13	In case, users delete a node and then add a new node with the same telephone number, GHLWMS consider they are the same node and save an activity history for the node as "Restore a node which was deleted before"

1.17. UC17 – Sort activity history

1.17.1. Screen Design

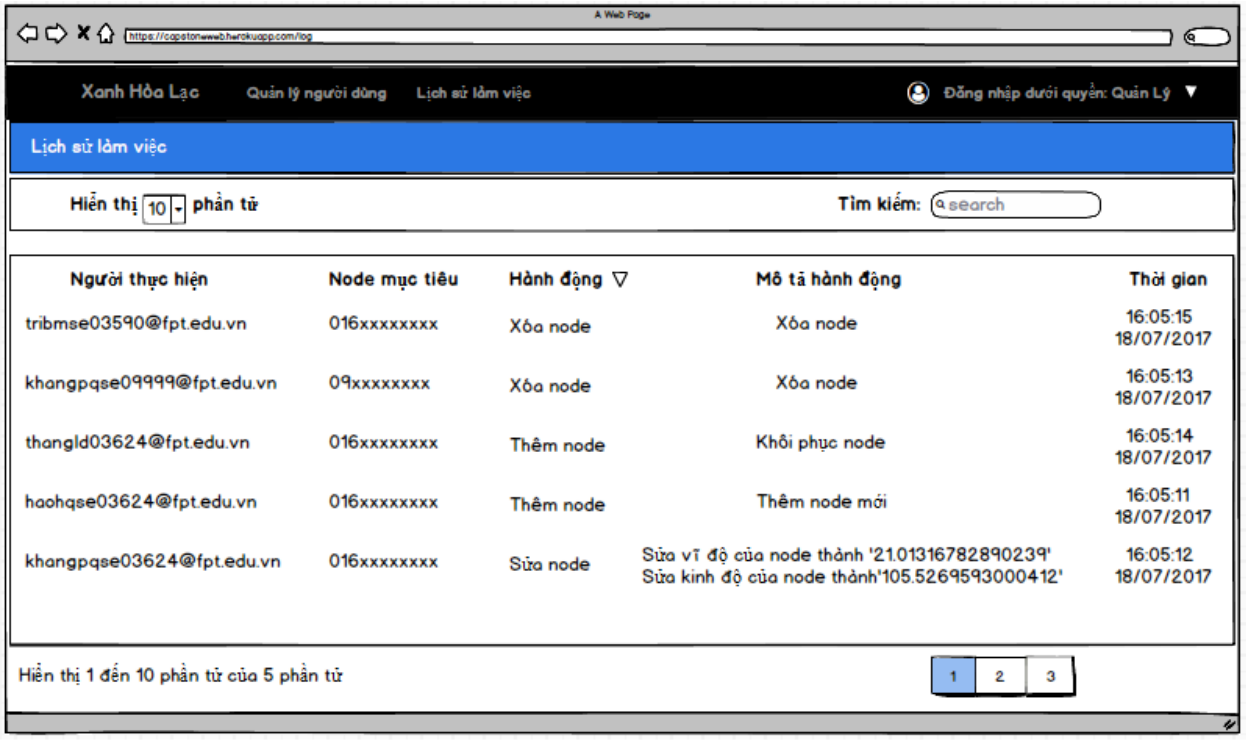


Figure 3.64: Sort activity history screen

Table 3.31: Sort activity history Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	List activity history	Table			The table contains 5 columns: user, target node, activity, description, time

1.17.2. Use case specification

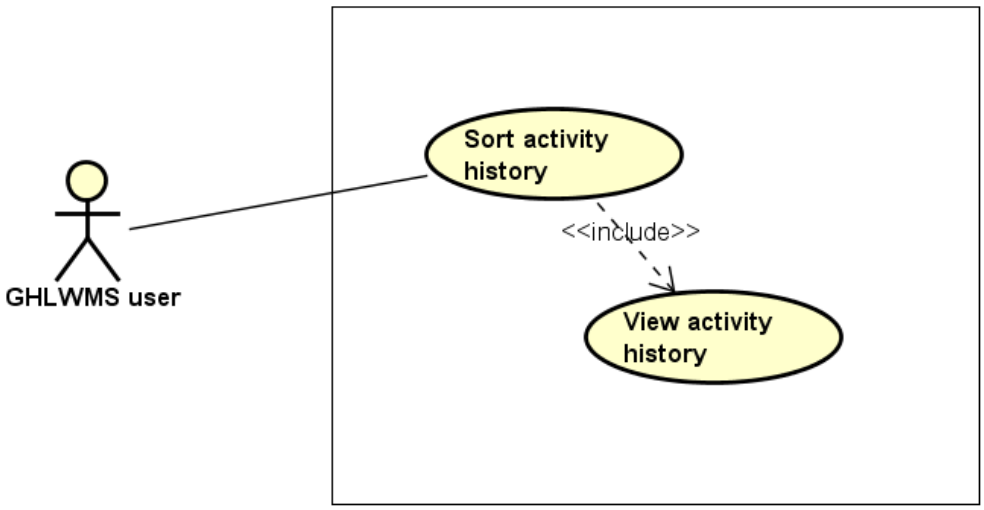


Figure 3.65: Sort activity history Use case diagram

Use Case ID	**UC17**	**Use Case Name**		**Sort activity history**	
Author	**KhangPQ**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to sort activity history by user, target , node, activity, description or time				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click Activity history on the navigation bar			
2	GHLWMS	Display Activity history page with a table which has default top 10 previous actions.			
3	User	Click on one of column's title in the table			
4	GHLWMS	Sort list activity history by the title selected by users.			

1.18. UC18 – Logout

1.18.1. Screen Design

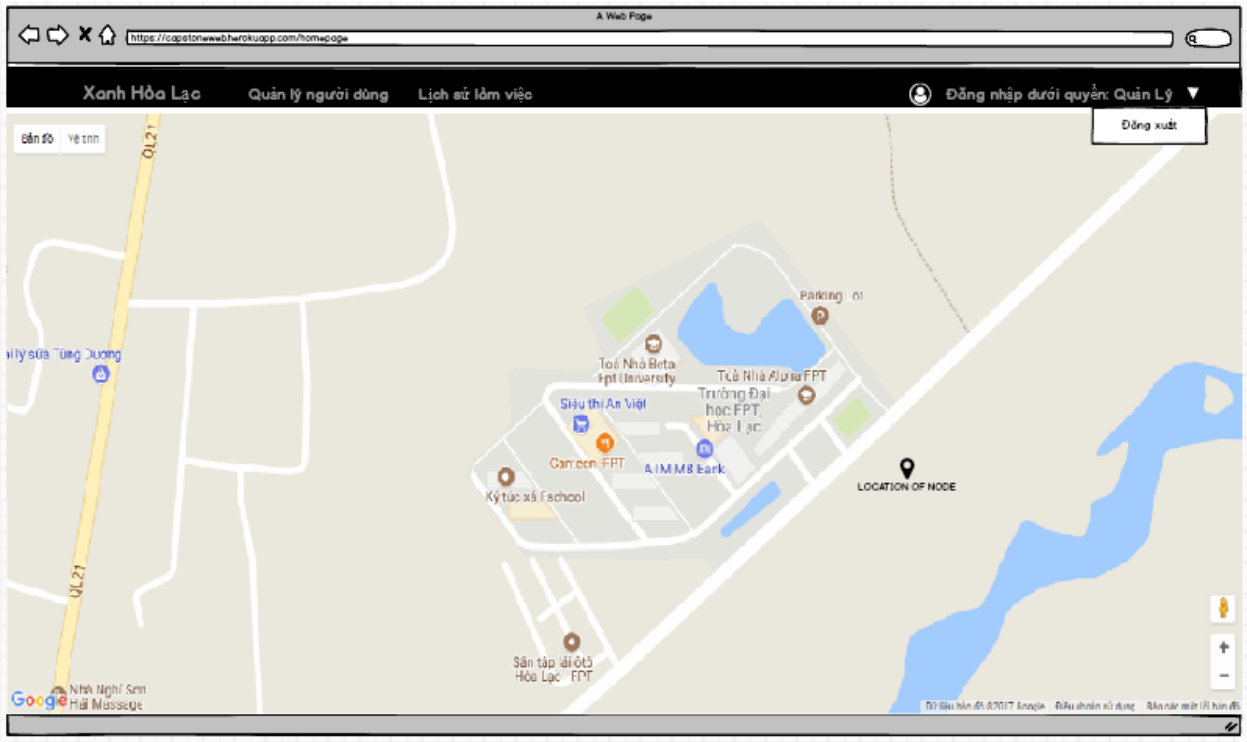


Figure 3.66: Logout screen

Table 3.32: Logout Screen definition

#	Field Name	Type	Mandatory	Max Length	Description
1	Logout	An Item of drop down list			

1.18.2. Use case specification

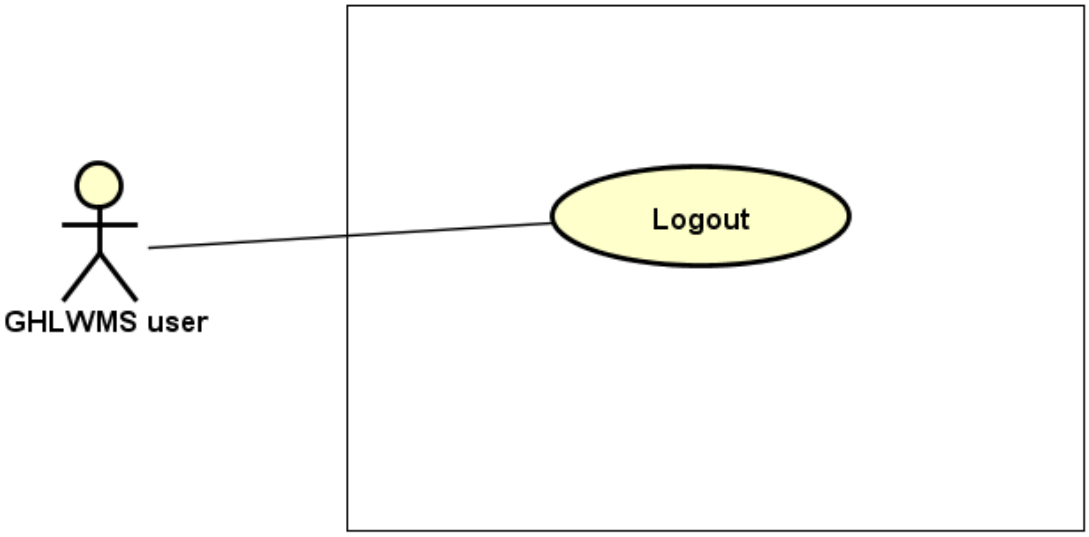


Figure 3.67: Logout Use case diagram

Use Case ID	**UC18**	**Use Case Name**		**Log out**	
Author	**KhangPQ**	**Version**	**1.0**	**Date**	**16/06/2017**
Actor	GHLWMS User				
Description	The function allows an user to be able to log out off GHLWMS				
Precondition					
Trigger					
Post-Condition					
Main flows					
Step	***Actor***	***Action***			
1	User	Click a drop down list on the right top of the navigation bar			
2	GHLWMS	Display list item in the drop down list			
3	User	Select Logout			
4	GHLWMS	Logout user and navigate to Login page			

2. *Non-Functional Requirement Specification*

Table 3.33: Nonfunctional requirements for GHLWMS

#	Nonfunctional requirements	Description
1	Security	Users have to sign in with Google's accounts
2	Availability	The user session must time out after 24 hours
3	Usability	Language: Vietnamese

Chapter 4: System design

I. Architecture design

1. System architecture design

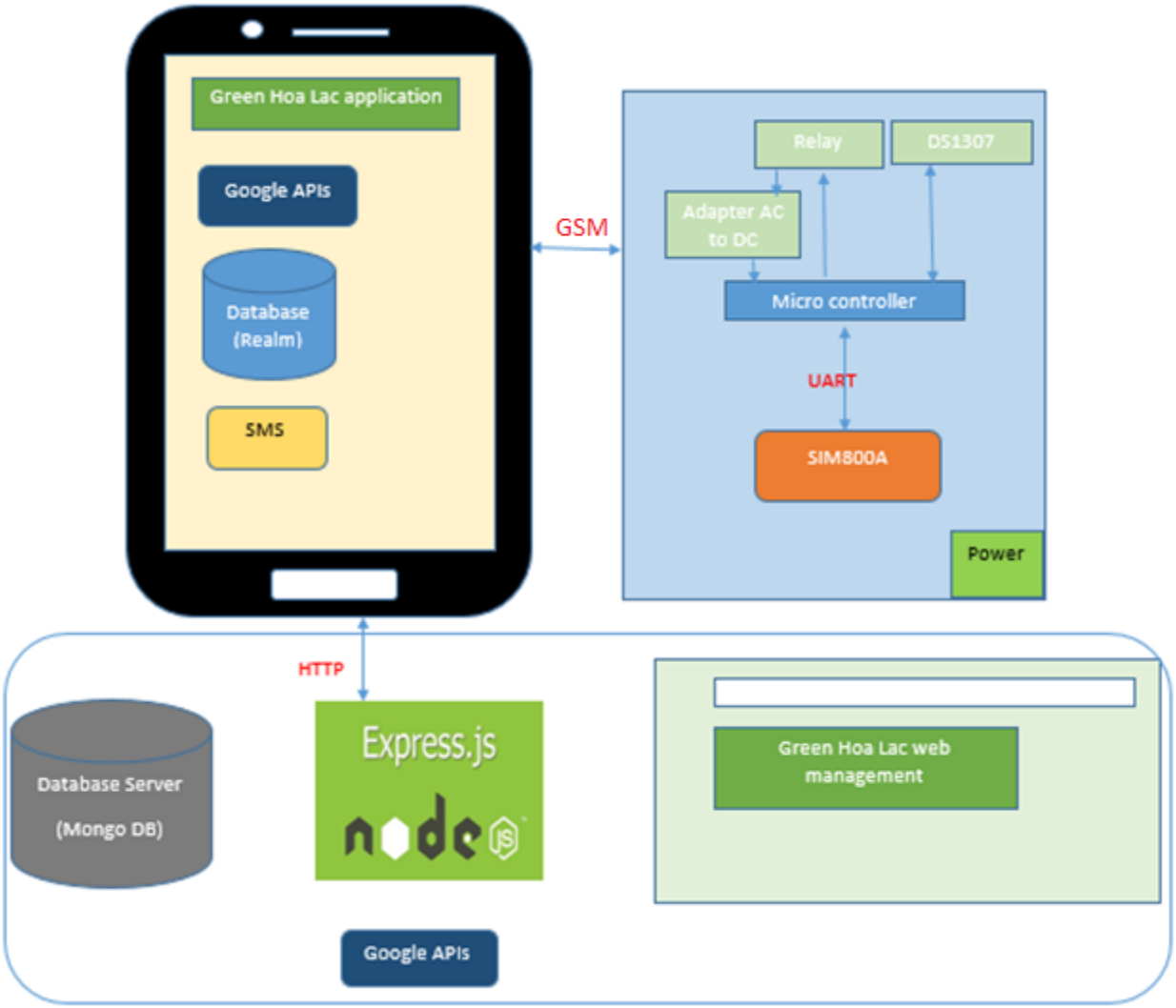


Figure 4.1: Architecture design

Green Hoa Lac system has 5 parts:

❖ *Green Hoa Lac mobile application:*

Android platform is chosen to develop Green Hoa Lac mobile application. Users can control this application through Android containers.

The application uses local database (Realm) to store map information in case of losing Internet connection.

The application sends SMS messages to control parts of Micro controller, and receive notification and messages by SMS messages.

We use GSM [2] for the reason that is the application and Module Node interact each other through SMS.

❖ *Module Node*

Arduino Mega is chosen as the micro controller process unit that connected five modules: Module SIM800A MH, Adapter, Relay, Module DS1307+AT24C32

Module SIM800A MH has features of sending and receiving SMS from mobile device, then move the SMS string data to micro controller. In this time, Arduino is receiving string data and optimizing this string, sending signal to each part of GHLS.

Controller and Module SIM800A MH interact each other using UART protocol

❖ *Server database*

We use mongo DB as a server database of GHLS.

All information about user, node, session and history are store in this database

❖ *Green Hoa Lac management web*

We design a web to manage whole GHLS. Any actions related users are performed and shown on this web. In addition, to extend or shrink real scope of GHLS, we also do through the web.

❖ *Server*

NodeJS Express Application is chosen to develop APIs such as: Weather, Node, Log, User and Authentication.

2. *Serial port (UART) communication protocol*

The SIM800 GPRS/GSM+GPS Shield is used UART protocol to communicate with an Arduino/Arduino clone.

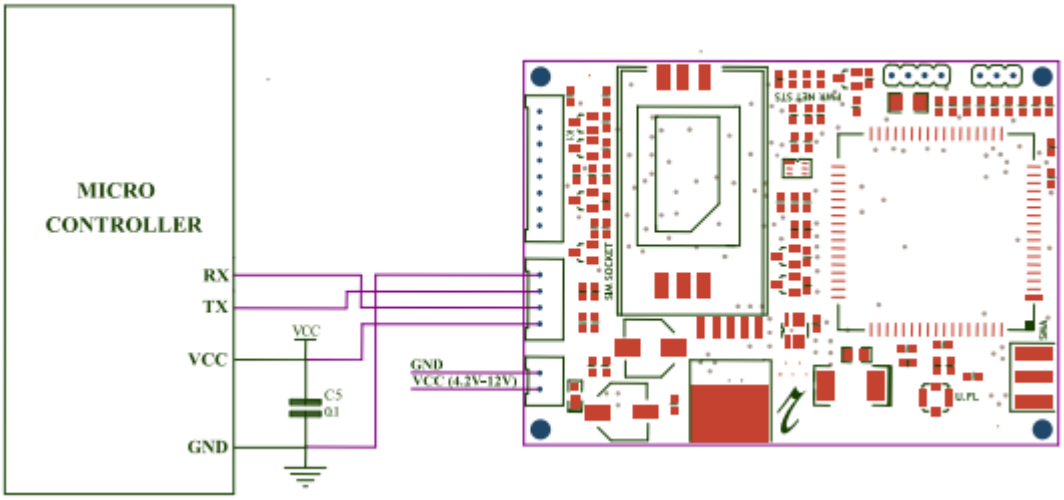


Figure 4.2: Serial port (UART) communication protocol

3. *GSM*

The system requires stable devices and a few of data. Therefore, GSM [3] is the best choice, which is the most stable device. In addition, this module is available in the market, we can easily buy and research it.

We only use SMS in this system. SMS uses standardized communications protocols to allow fixed line or mobile phone devices to exchange short text messages. SMS messages can be used to transport almost any kind of data (within the character limit).

II. Mechanical design

1. Design box

1.1. Overall

Figure 4.3: Real box

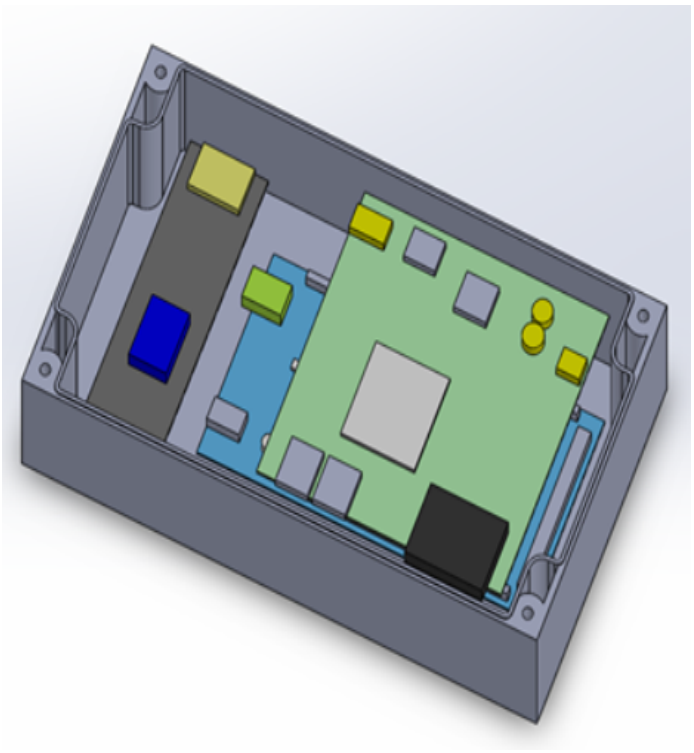


Figure 4.4: Box 3D

1.2. Box with all parts

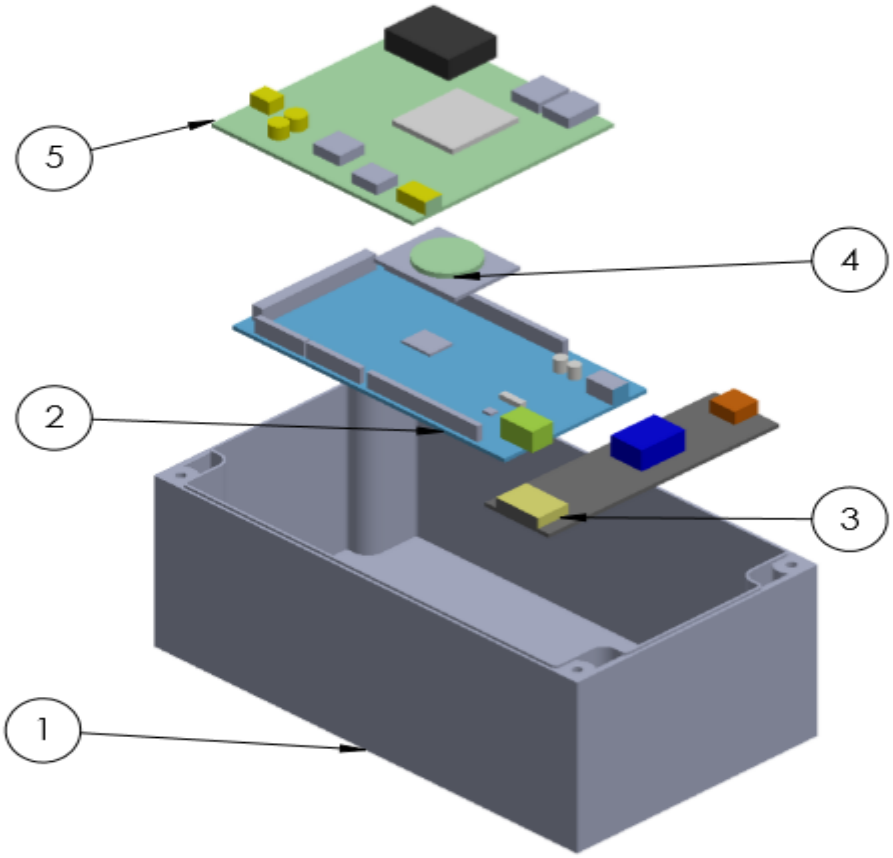


Figure 4.5: Box 3D with all parts

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Box		1
2	Arduino board		1
3	Relay		1
4	RTC		1
5	Sim Module		1

Figure 4.6: Note

1.3. Design box

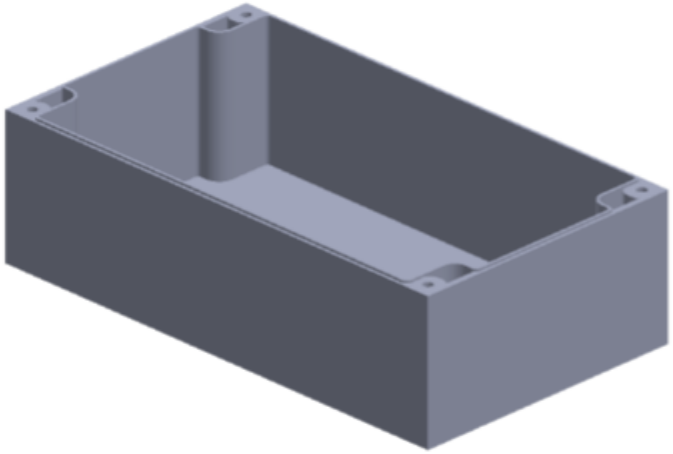


Figure 4.7: Box

Figure 4.8: The vertical projection

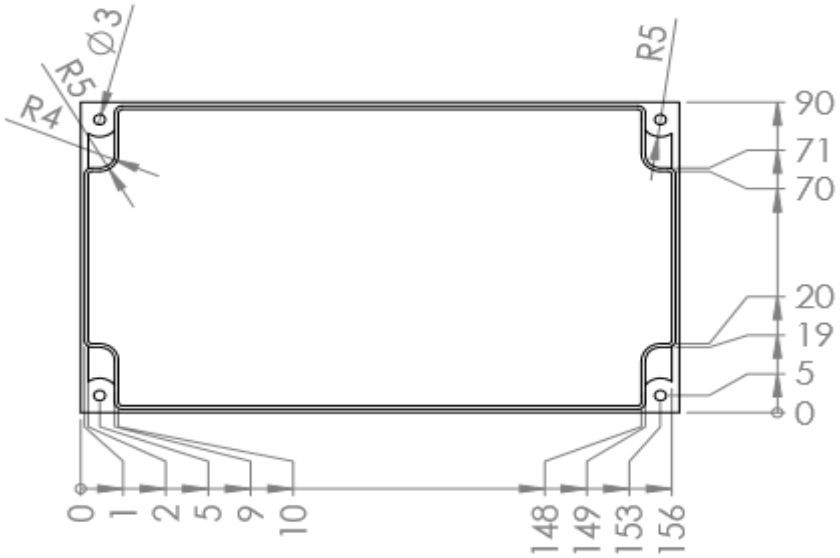


Figure 4.9: Top view

2. Schematic

2.1. All parts

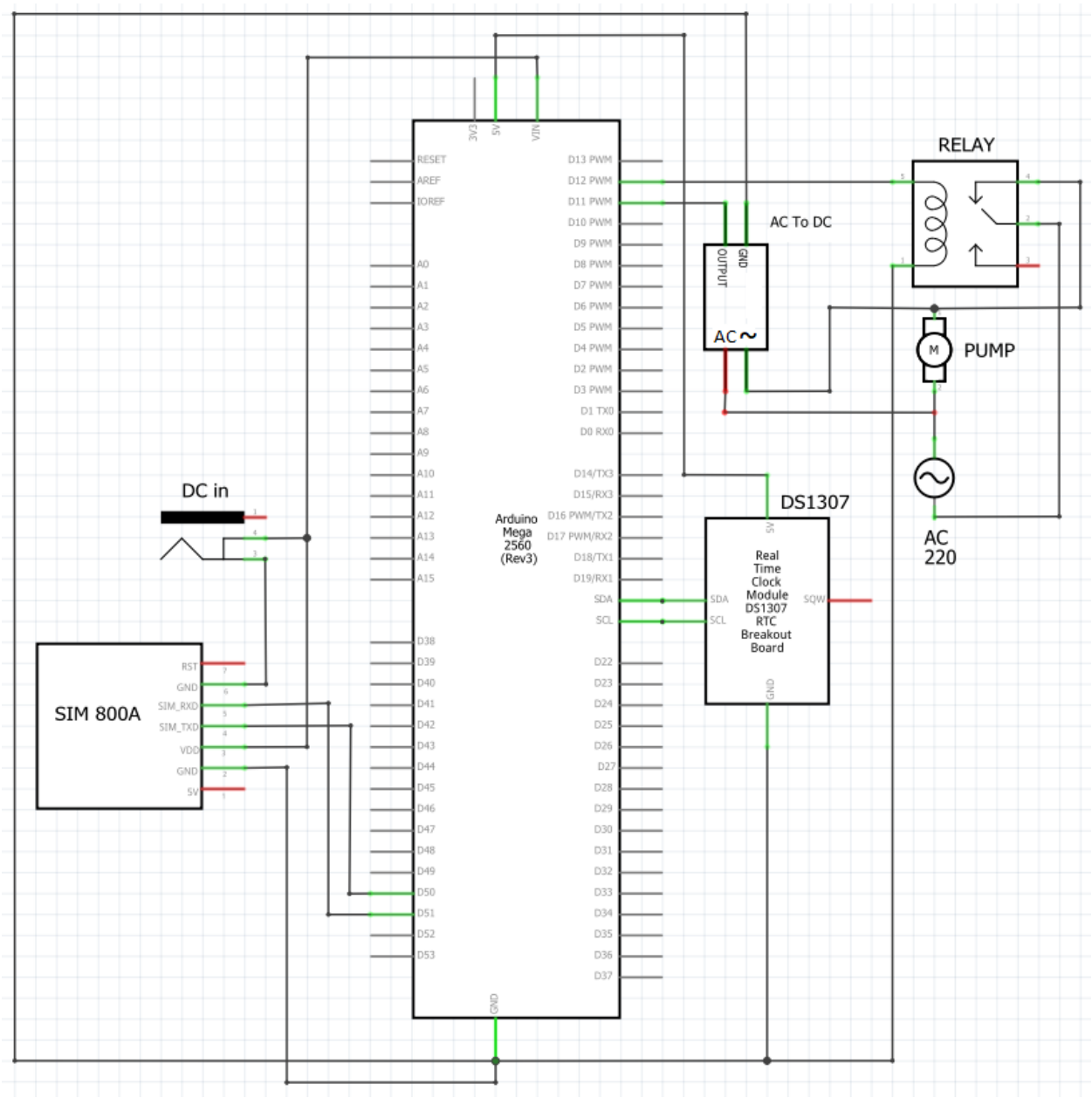


Figure 4.10: Schematic

2.2. Detail

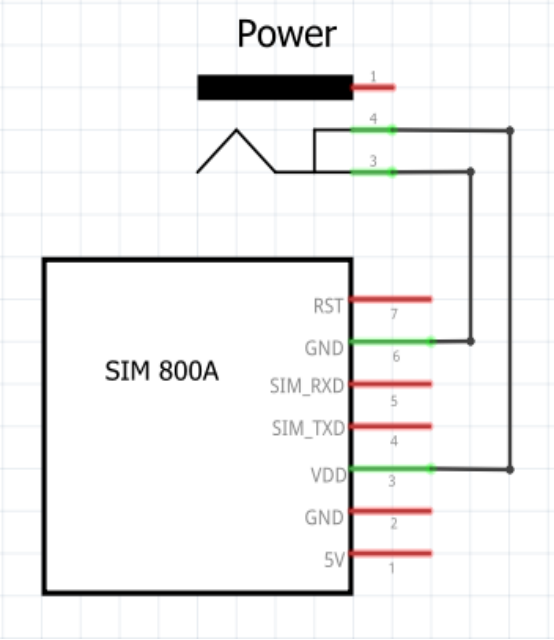


Figure 4.11: SIM and Power

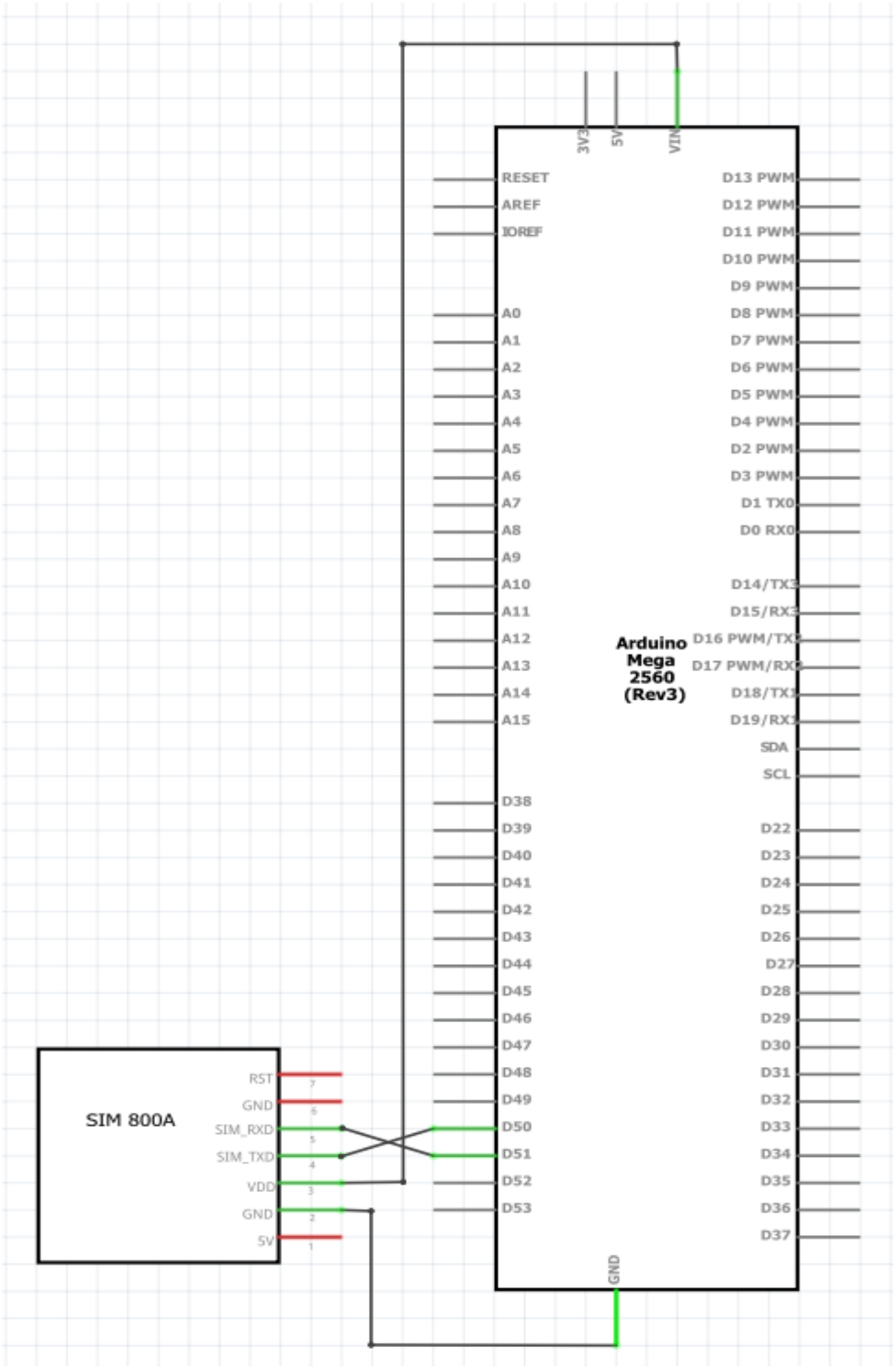


Figure 4.12: SIM and Arduino

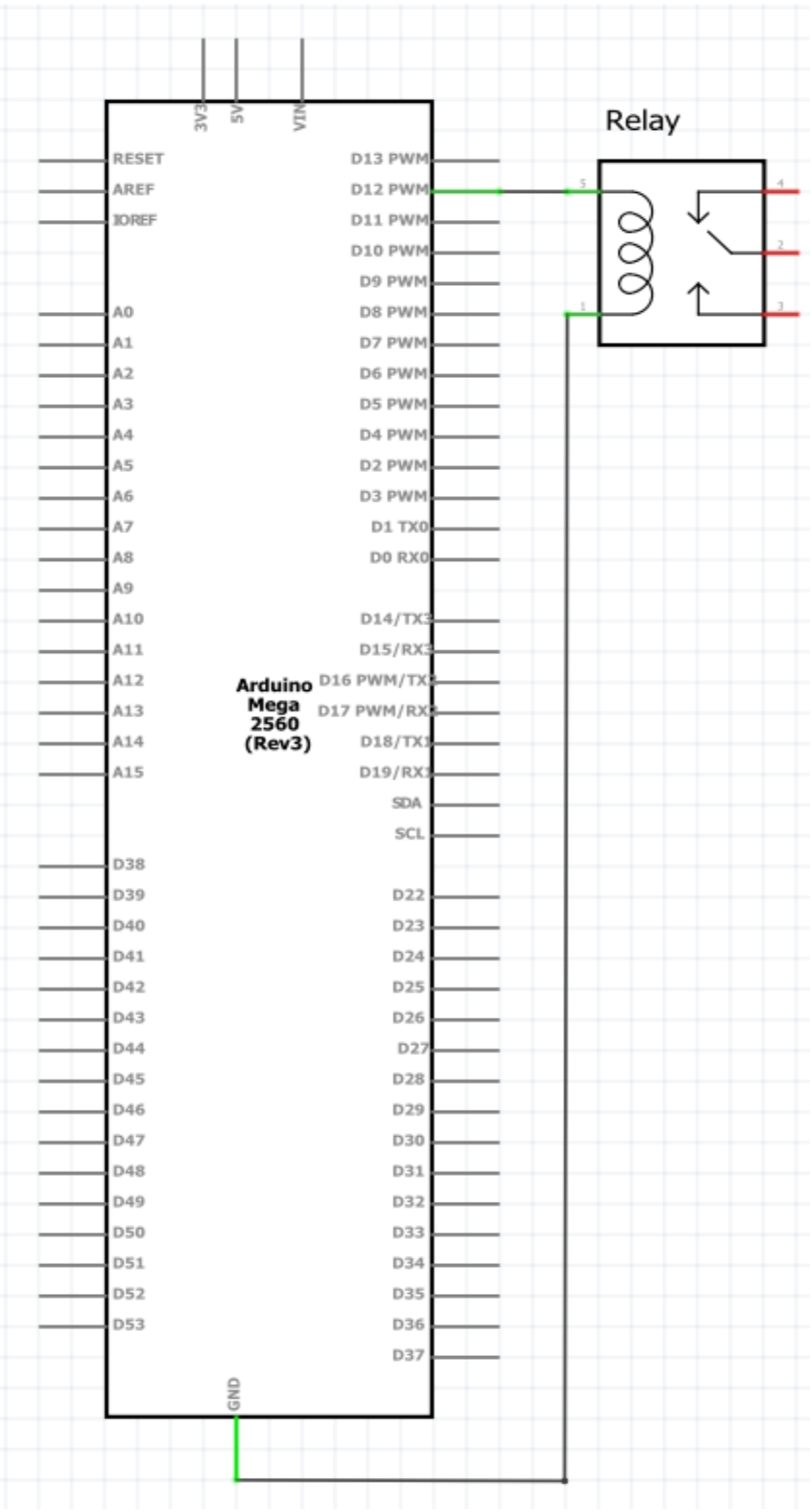


Figure 4.13: Arduino and Relay

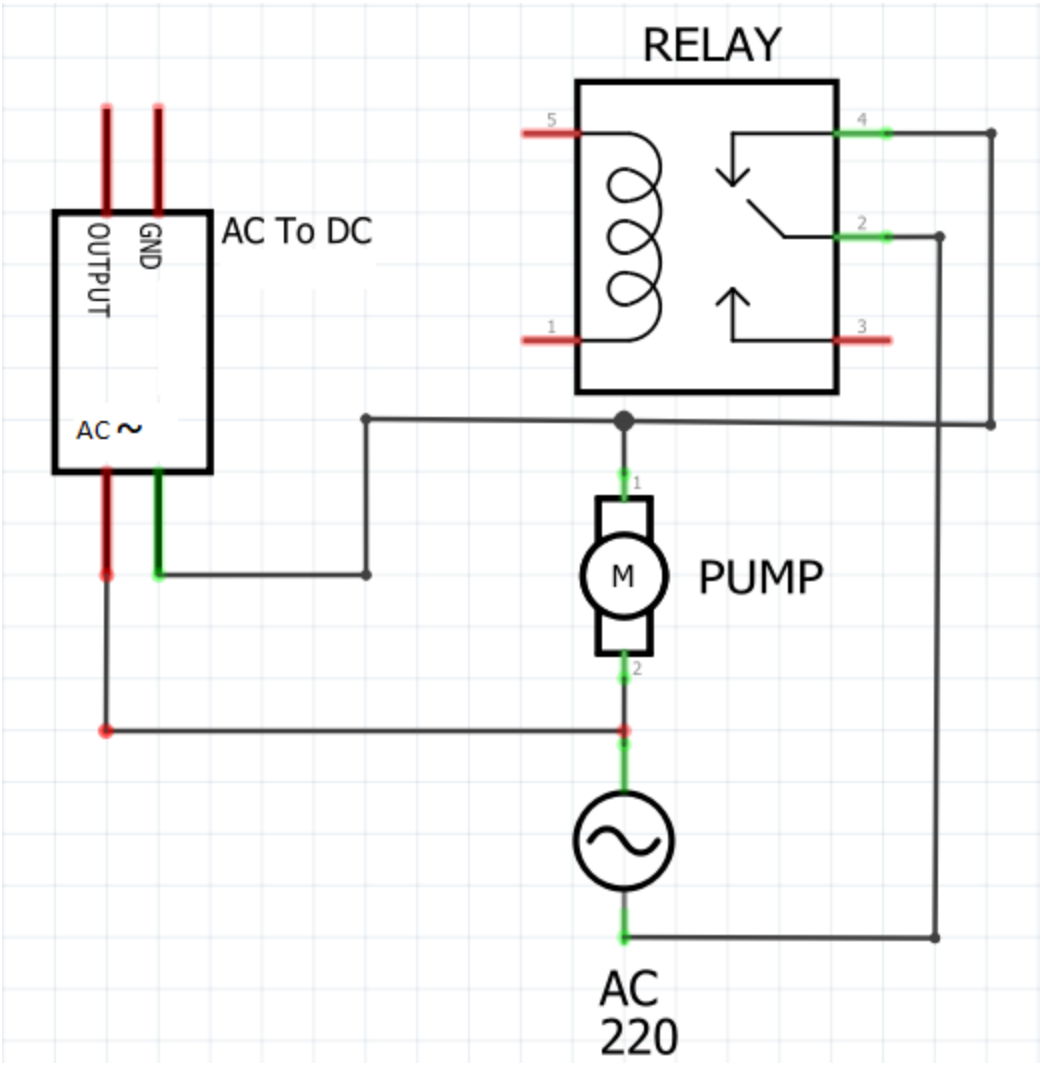


Figure 4.14: Relay, pump and adapter

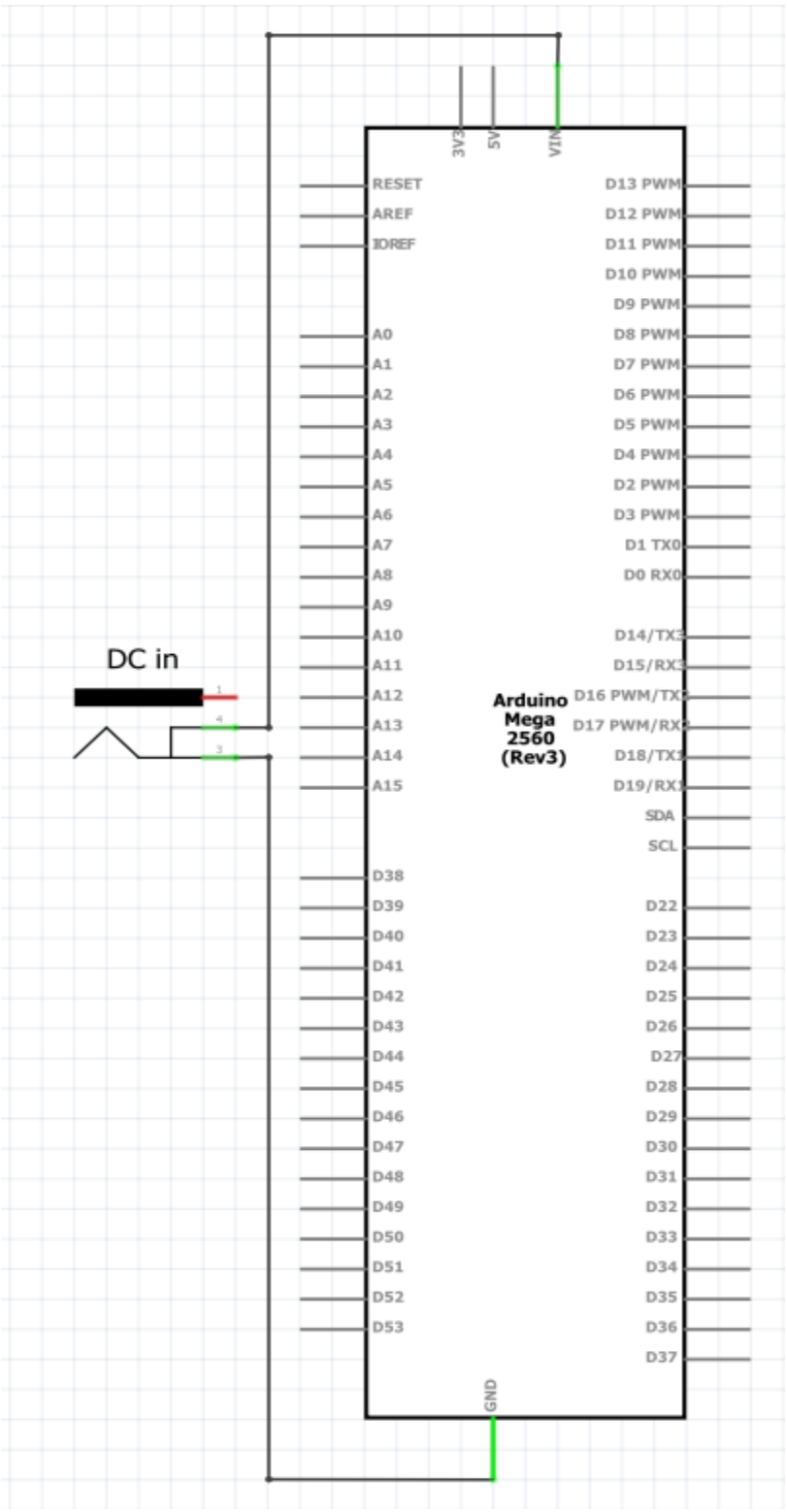


Figure 4.15: Power and Arduino

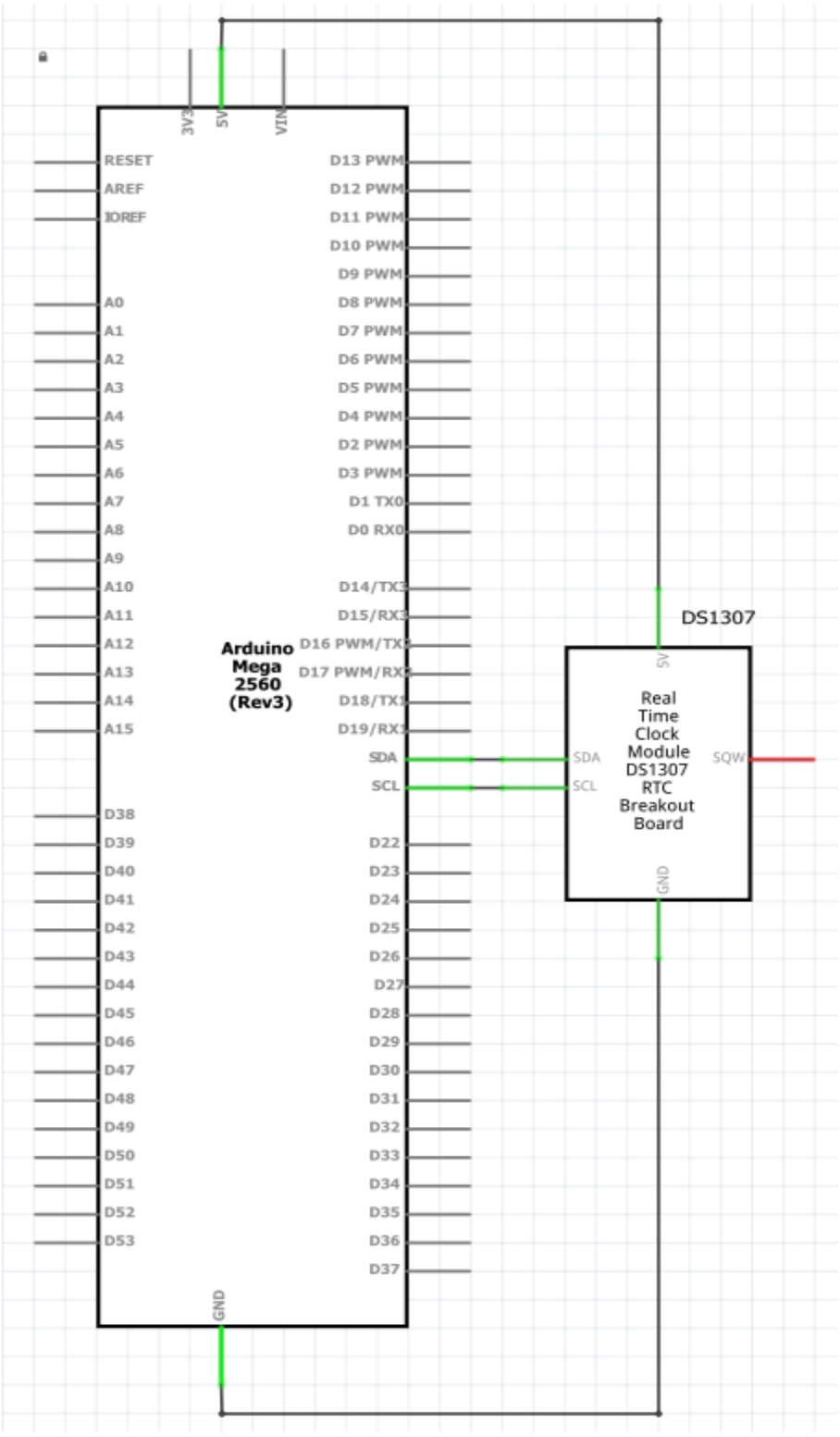


Figure 4.16: Arduino and DS1307

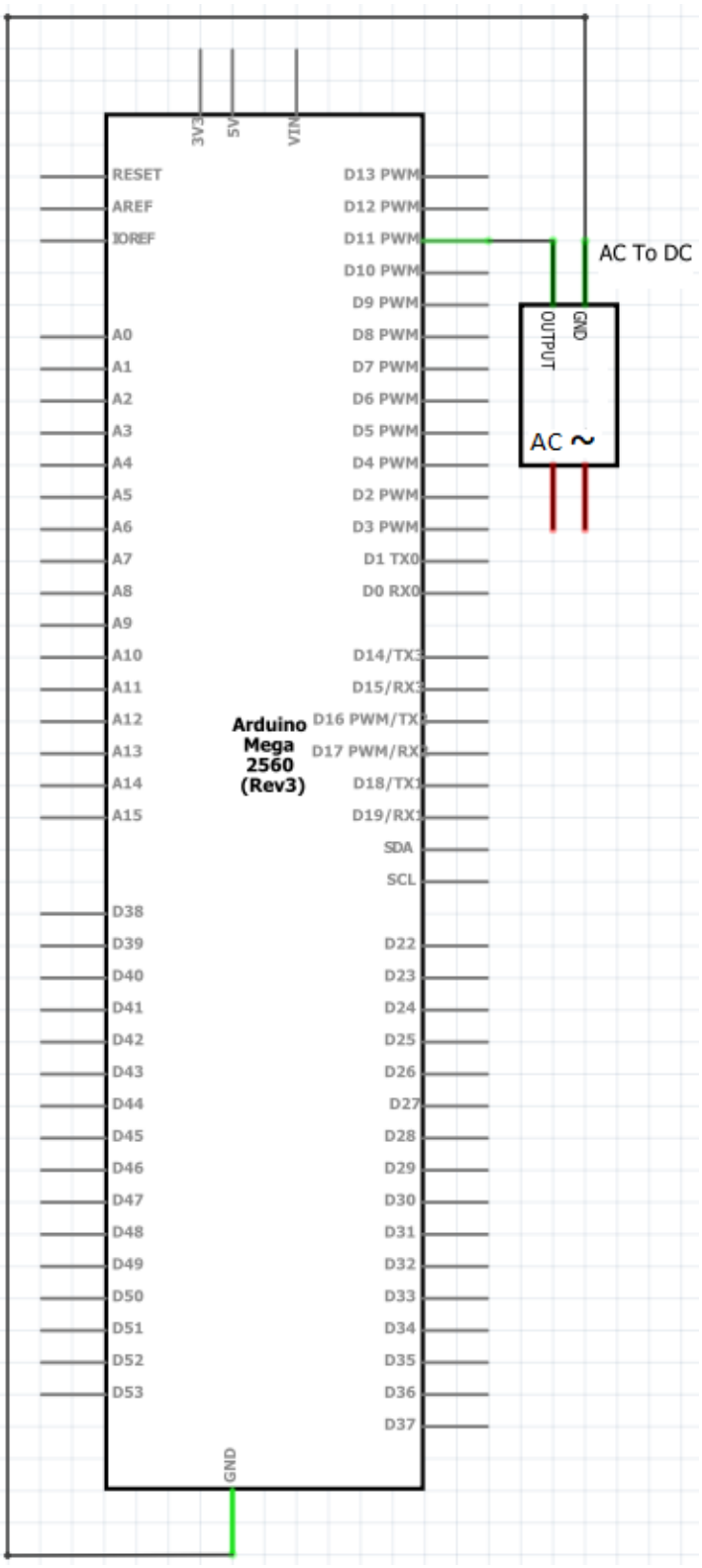


Figure 4.17: Arduino and Adapter

III. Firmware design

1. SMS syntax

We define syntax to interact between android application and the hardware module though SMS message. Each request from the application to hardware module contains 16 characters following rules

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Length	3 bits			2 bits		2 bits		7 bits							1 bit	1 bit
Value	DURATION(mins)			MINUTE		HOUR		DAYS							STATUS	MODE

Data Type	bit	Name	Value		Description	Notes
	0	MODE		1	Control manual	
				2	Schedule	
				3	Admin (control manual)	
	1	STATUS	MODE = 1	0	Turn on node	bit [2-12] = 0
				1	Turn off node	If do not set time turning off: bit [13-15] = 0
			MODE = 2	1	Set schedule once	bit [2-8] including value of day month year
				2	Set schedule daily	bit [2-8] = 0
				3	Set schedule weekly	bit [2-8] include value of date in week
				0	Cancel schedule	
					bit [2] = 0	review Schedule
					bit [2] = 1	Cancel schedule once
					bit [2] = 2 bit [3] = index [0-3] of selected schedule in multi-time list	Cancel schedule repeatly
					bit [2] = 3	Cancel all schedule
			MODE = 3	1	Admin	
	2	DAYS	Sa	DAY	when having specify values which are day month year :	
	3		Fr		* bit [2-7] including values of date month year according format yy/MM/dd	
	4		Th	MONTH	when having values which are date in week:	
	5		We		* Date which is set schedule is set = 1, date which is not set schedule is set = 0 into bit respectivel	
	6		Tu	YEAR	* Example: set schedule from Mon-Fri, Sat and Sun do not set then bit [2-8] = [0111110]	
	7		Mo			
	8		Su	WDAY		
	9 10	HOUR	24h	[00-23]		
	11 12	MINUTE		[00-59]		
	13 14 15	DURATION	phút	[0-999]		

Figure 4.18: SMS syntax

2. *Some main flows*

2.1. Get on

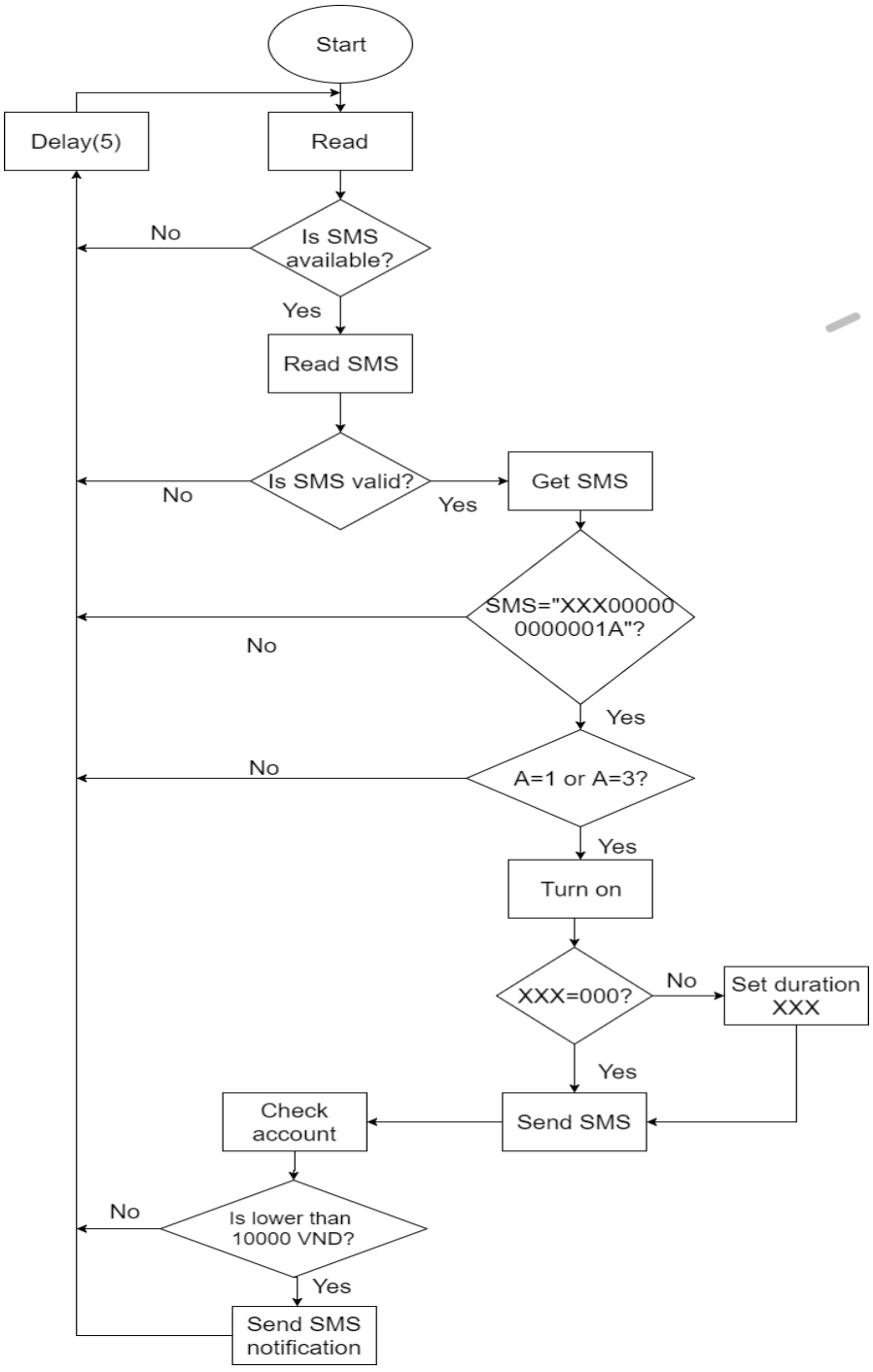


Figure 4.19: Get on

2.2. Get off

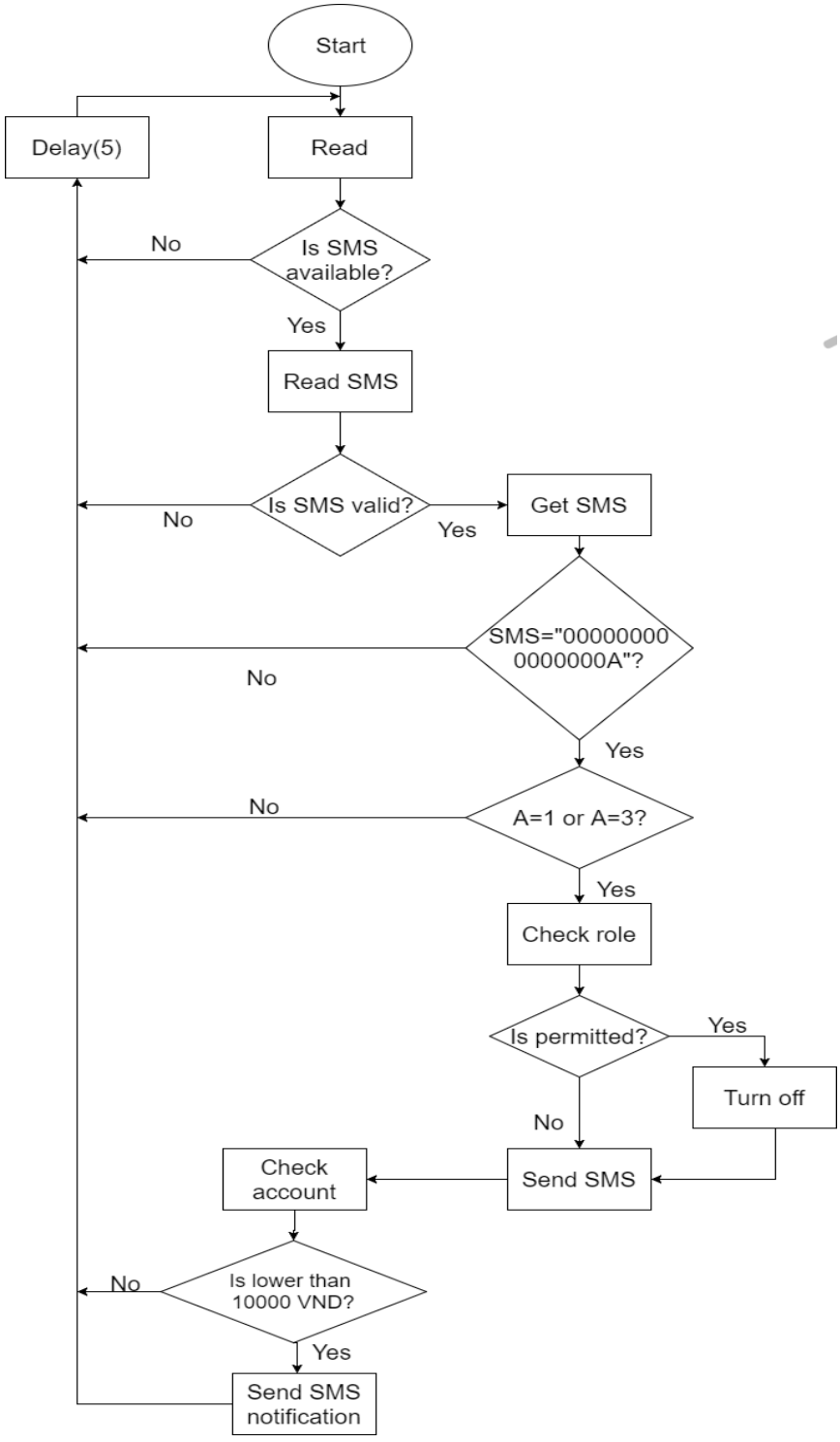


Figure 4.20: Get off

2.3. Set schedule

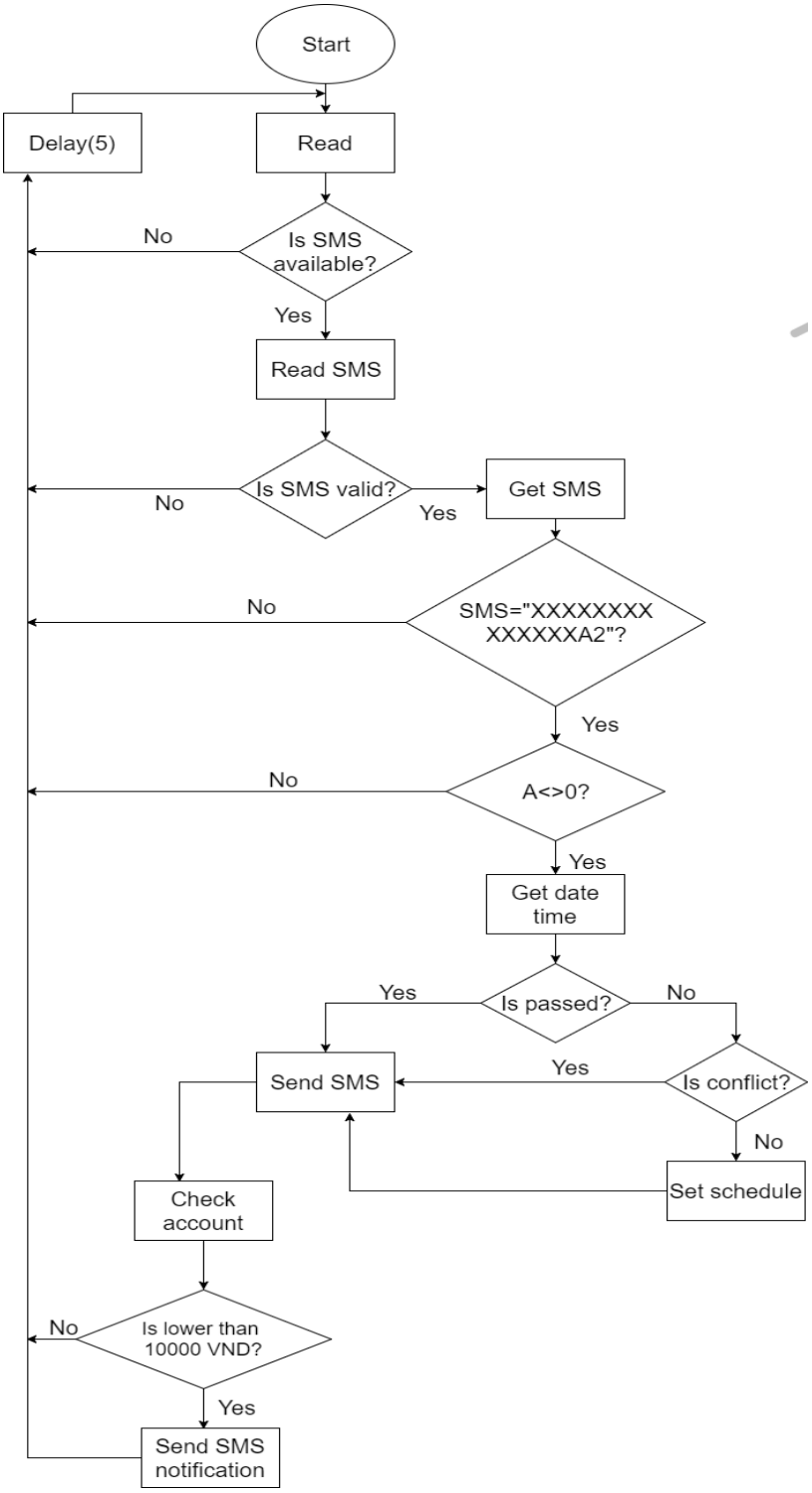


Figure 4.21: Set schedule

2.4. Get schedule

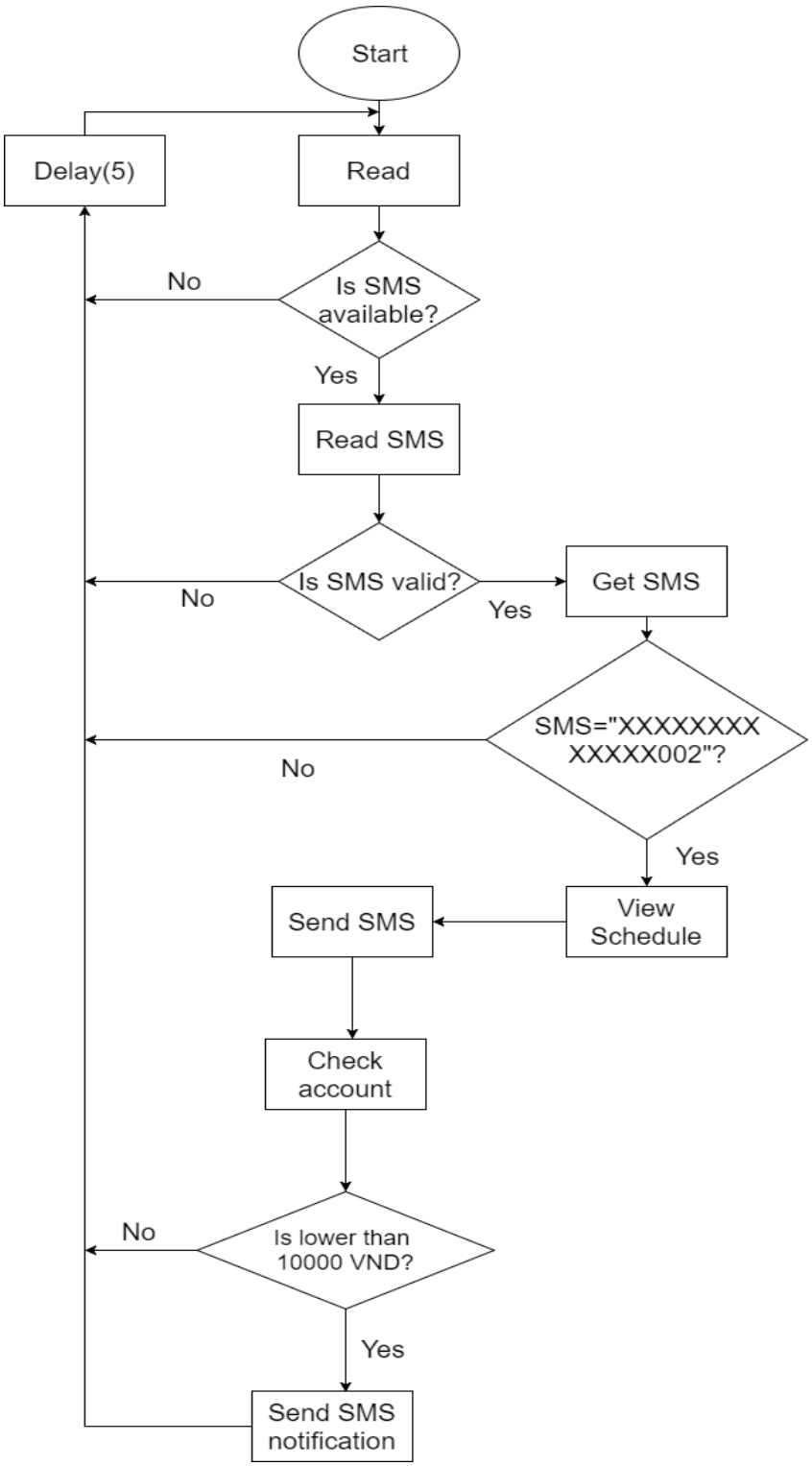


Figure 4.22: Get schedule

2.5. Cancel schedule

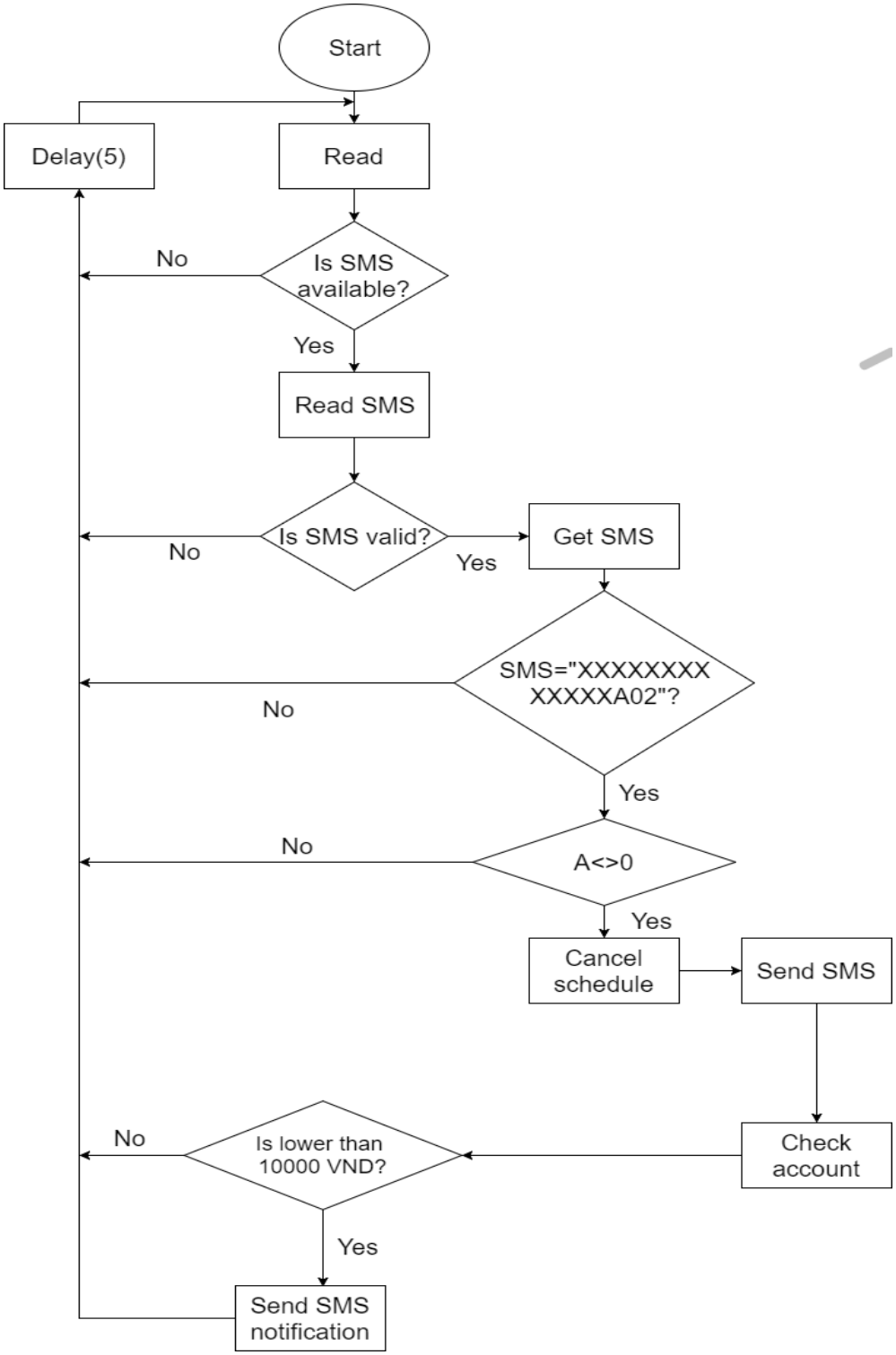


Figure 4.23: Cancel schedule

IV. Software design

For the application, we have developed many function. However, there are six main functions related to authentication, node control, synchronize.

1. Login

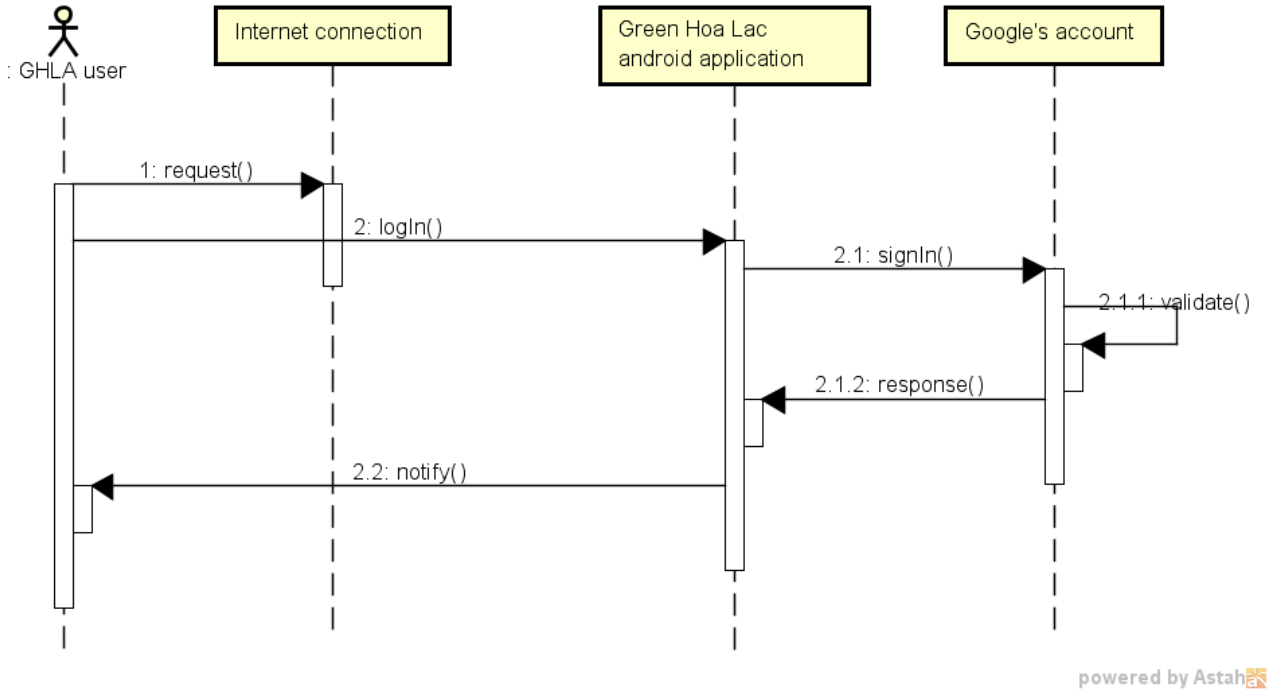


Figure 4.24: Login sequence diagram

2. *Turn on*

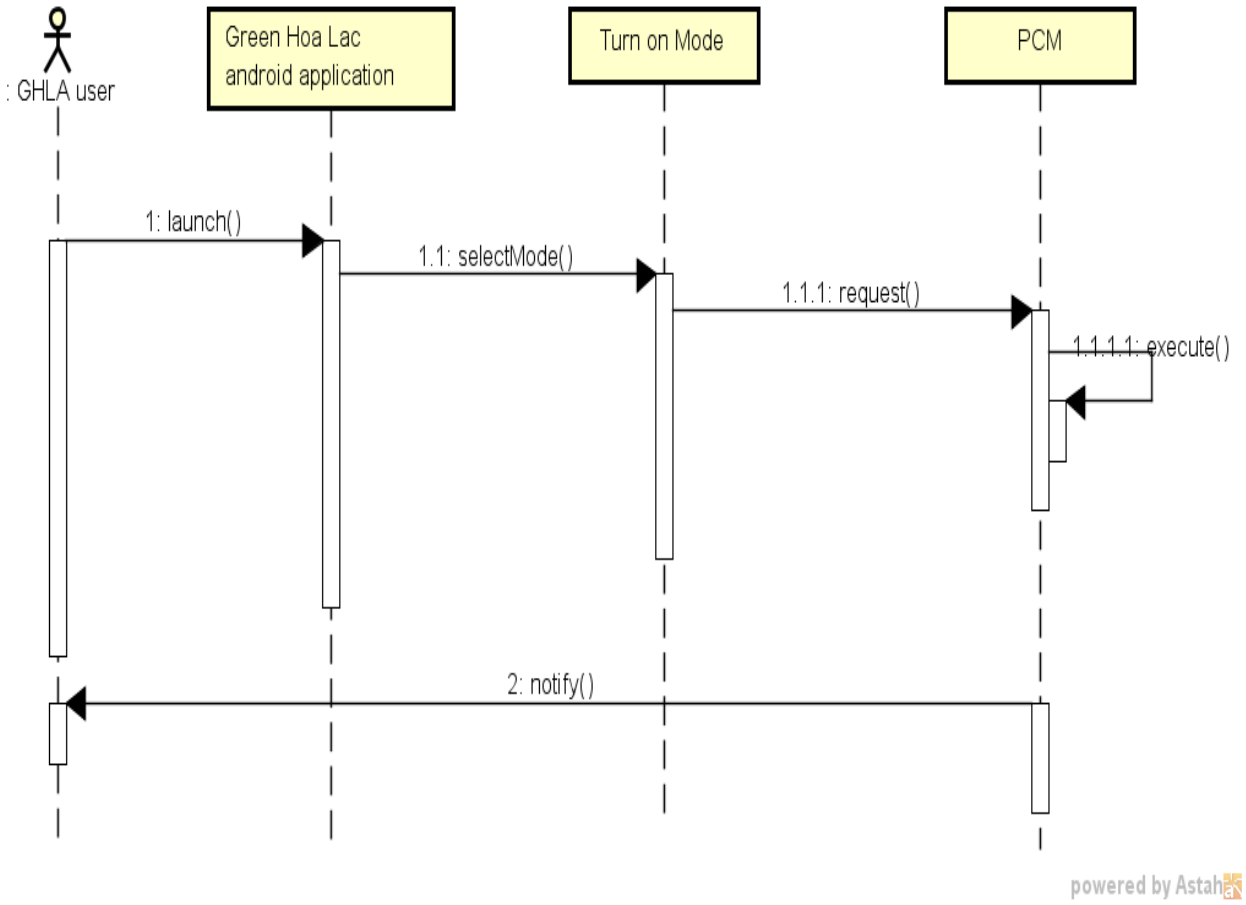


Figure 4.25: Turn on node sequence diagram

3. *Turn off*

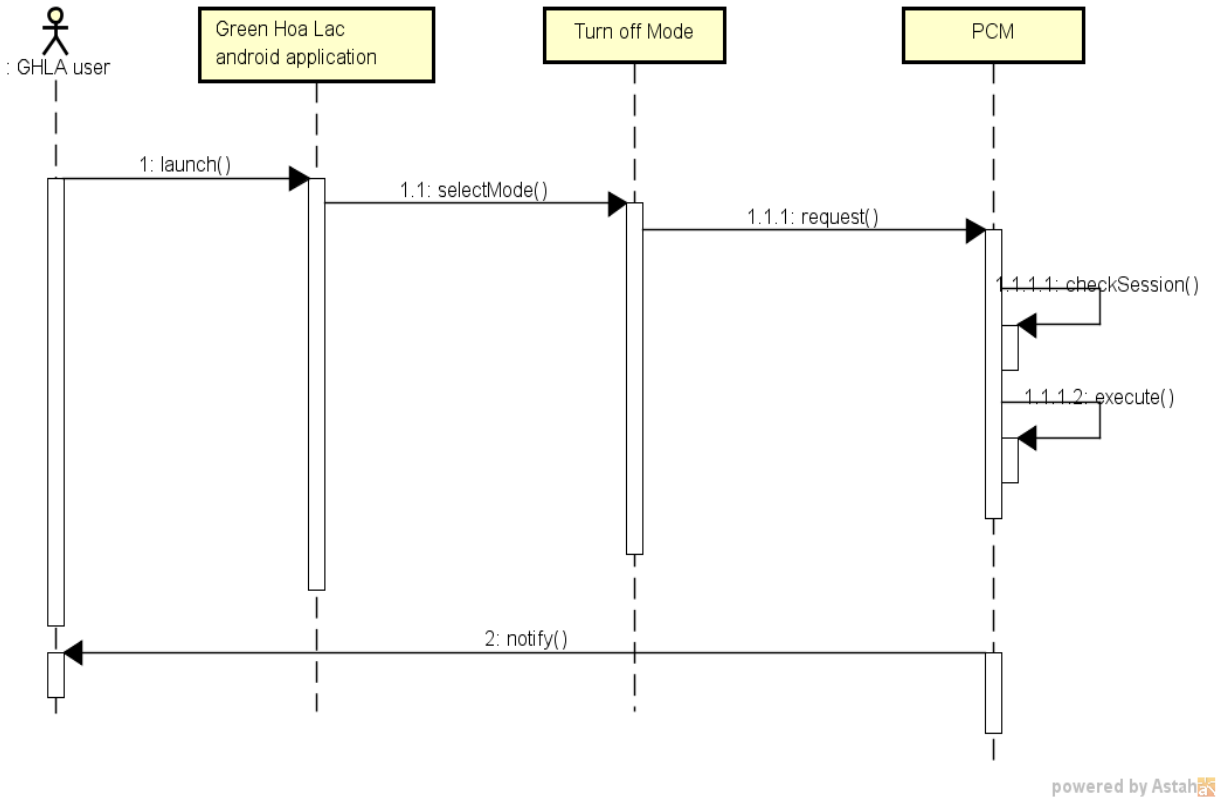


Figure 4.26: Turn off node sequence diagram

4. Set schedule

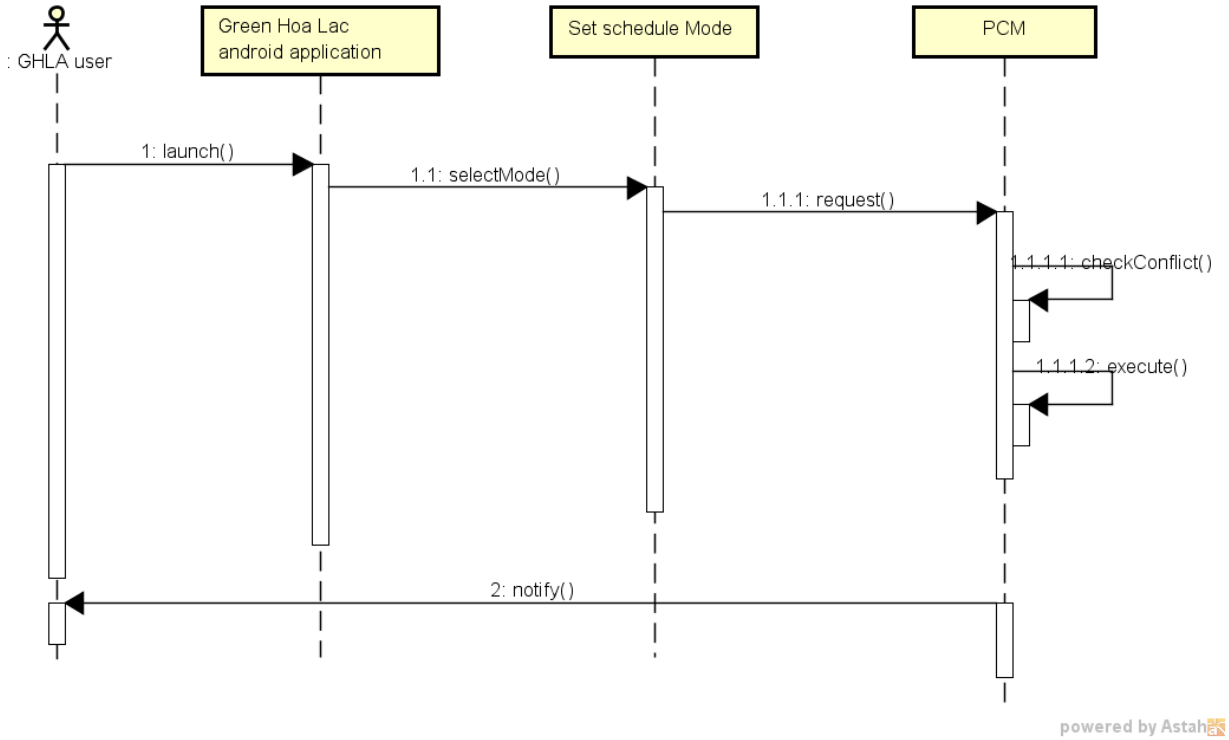


Figure 4.27: Set schedule sequence diagram

5. Cancel Schedule

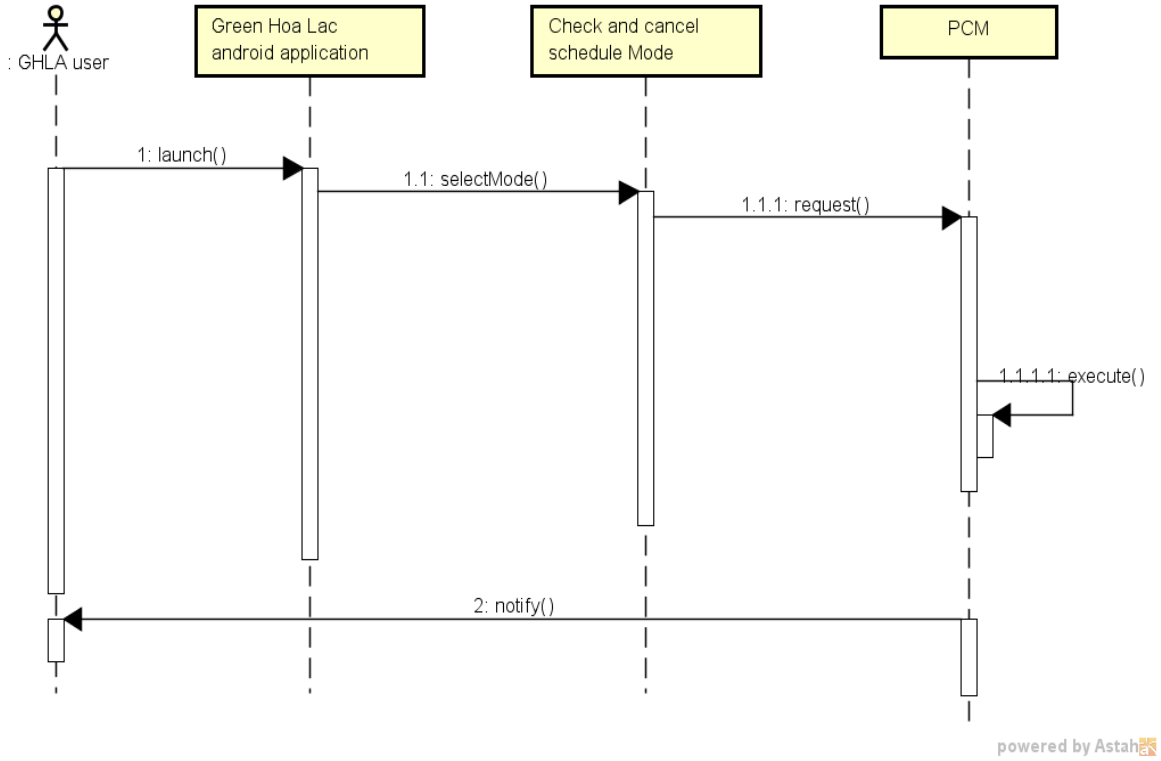


Figure 4.28: Cancel Schedule

6. *Synchronize node*

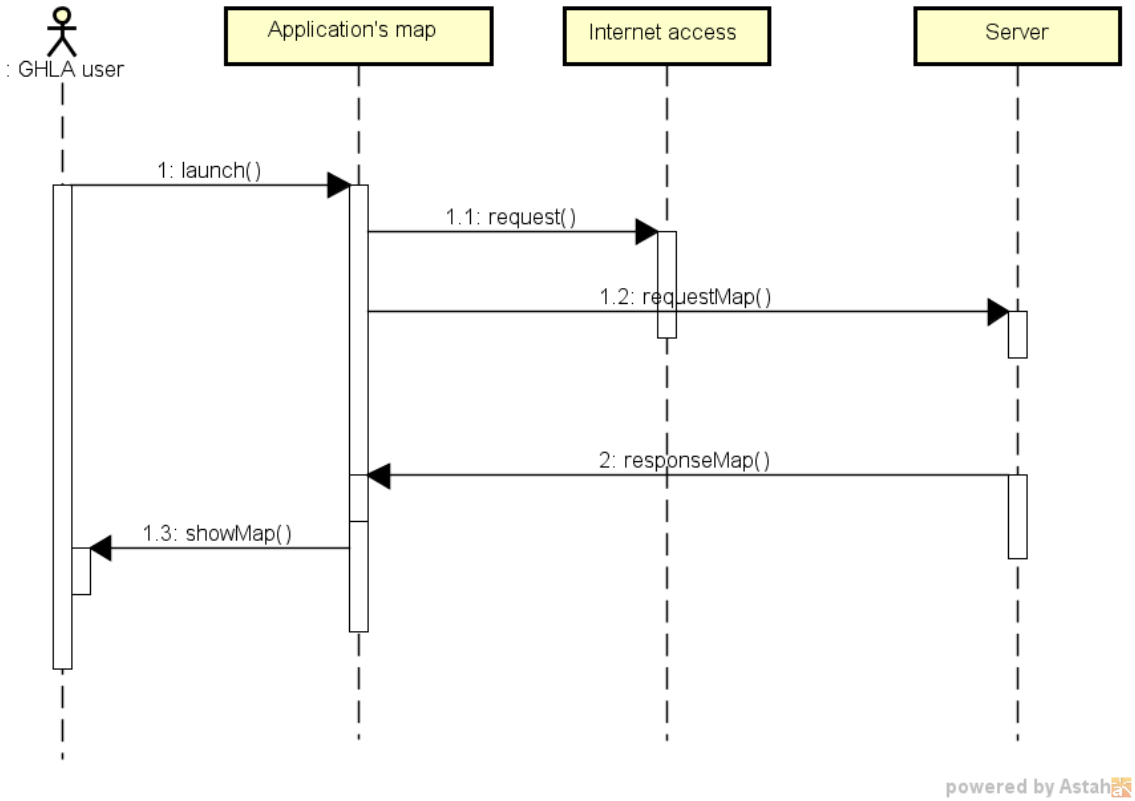


Figure 4.29: Synchronize node sequence diagram

V. Database design

1. *Server database*

We choose MongoDB [4] to develop system's database, which is NoSQL database.

MongoDB is best suited to dynamic query needs, requiring fast speed for a large database. MongoDB is fast, fast, and fast. MongoDB supports field searching, search results, and syntax searches. Queries can return the specified fields in the text, and may also include user-defined javascript functions. Just like relational databases, any field in MongoDB is indexed.

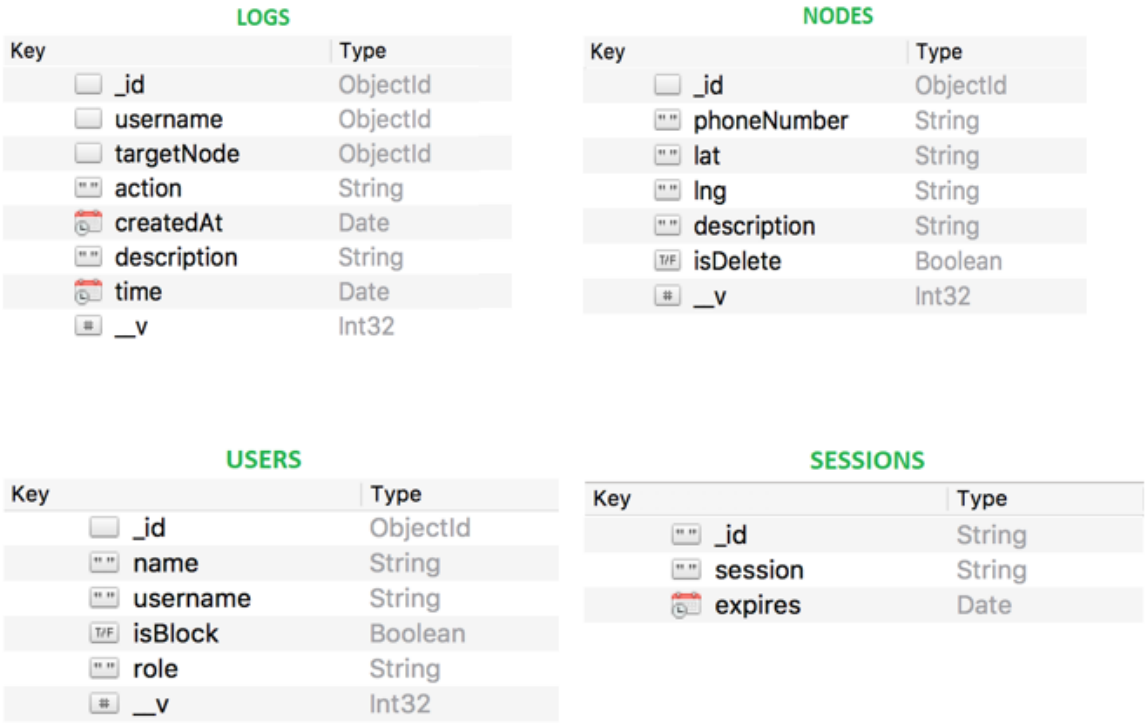


Figure 4.30: Server database (MongoDB)

In above design, we have:

logs.username=users._id
logs.targetNode = nodes._id

2. *Android application database*

Realm [5] mobile database is an open source, embedded database library optimized for mobile use. Realm uses a "data container" model. Our data objects are stored in a Realm *as objects.* We decided to choose Realm for some reasons*:* offline-first, store native objects, zero-copy

UserInformationObject		
String	emailAddress	@PrimaryKey
String	urilmage	
String	role	
String	idToken	
String	name	

NodeObject		
String	id	@PrimaryKey
String	lat	
String	lng	
String	description	
String	phone	
boolean	isDelete	
String	version	

LogNode		
int	idLogNode	@PrimaryKey
String	isSync	
String	userName	
String	targetNode	
long	time	
String	action	
String	description	

Figure 4.31: Realm database design

Chapter 5: Implement and testing

I. Implement

1. Version 1.0

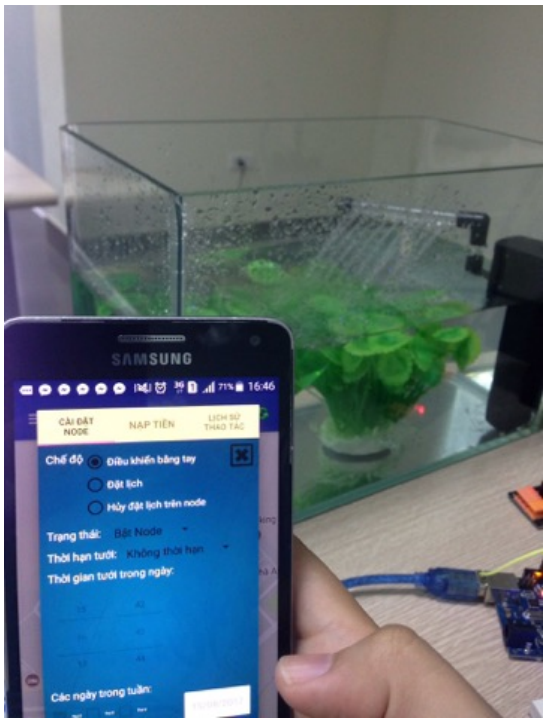
We have the first version of GHLS with features like:

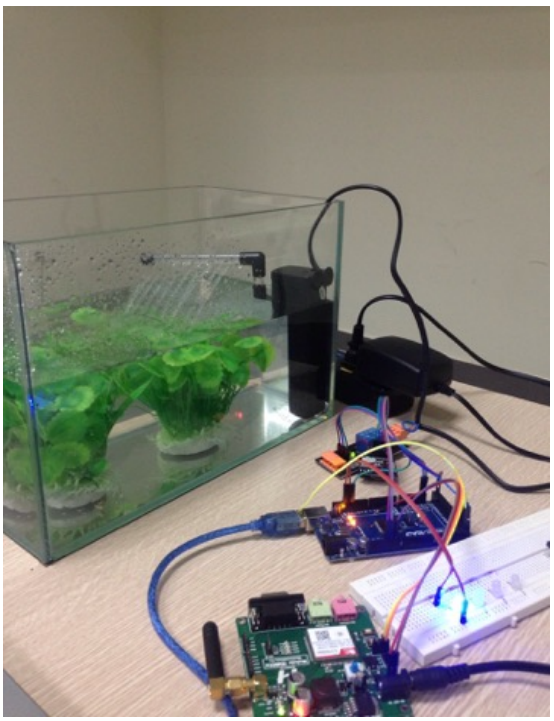
- Turn On/Of Node
- Set schedules
- Cancel schedules
- Notify account

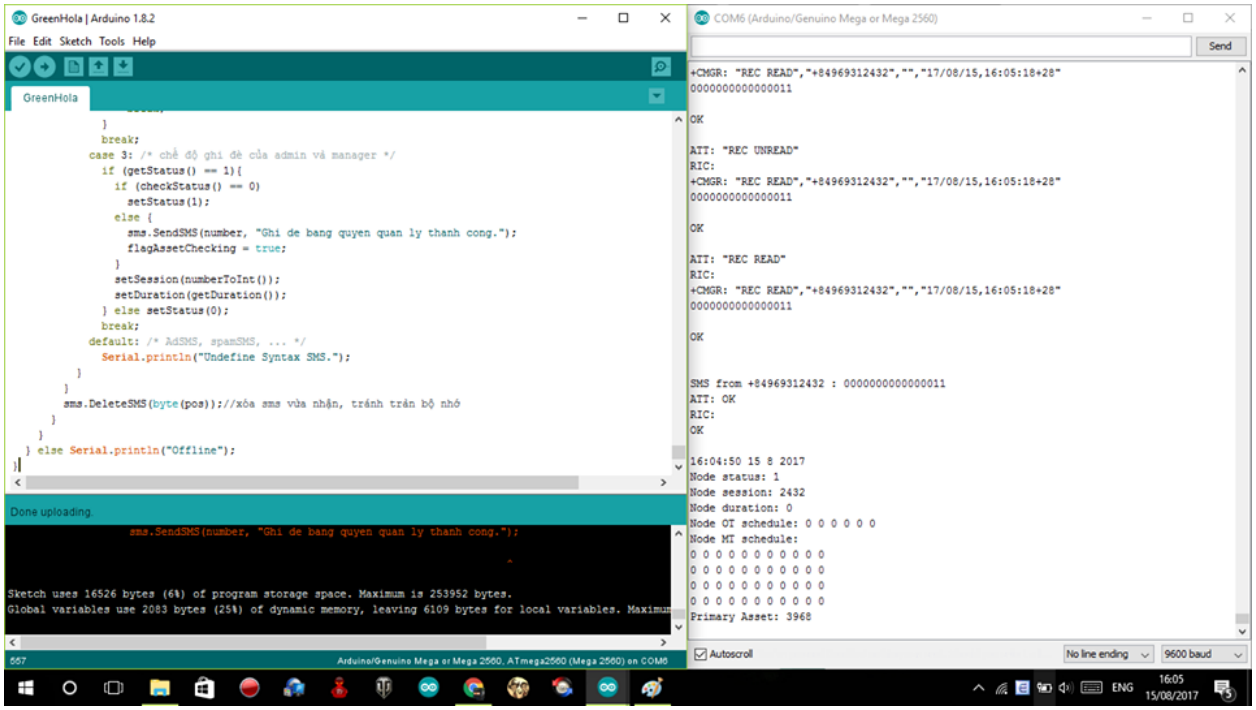
With this prototype, we expected it can cover all main functions of GHLS, and be used Q&A technical in LAB room environment.

Besides the hardware module, we gave a software application in mobile device, which is helpful to control GHLS.

- Send On/Off request mode
- Send set schedule request mode
- Get map, update nodes
- Cancel schedule







The version is kind of stable and most of responses from module SIM800 are corrected as expected.

2. *Version 1.1*

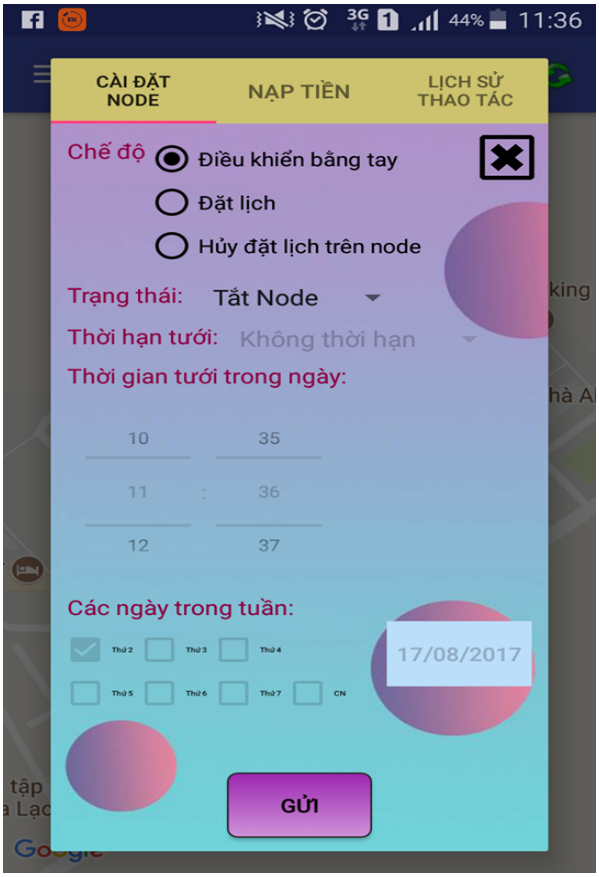
This version is based on version 1.0, which update features like:

- Turn On/Of Node
- Set schedules
- Cancel schedules
- Notify account

In addition, the software had been updated more functions for User as below:

- Send On/Off request mode
- Send set schedule request mode
- Get map, update nodes
- Schedule information
- Cancel schedule
- Weather, user information

The version is stable, easy to set up. Especially, we have customized graphic user interface and design box for hardware module to be useful for users.

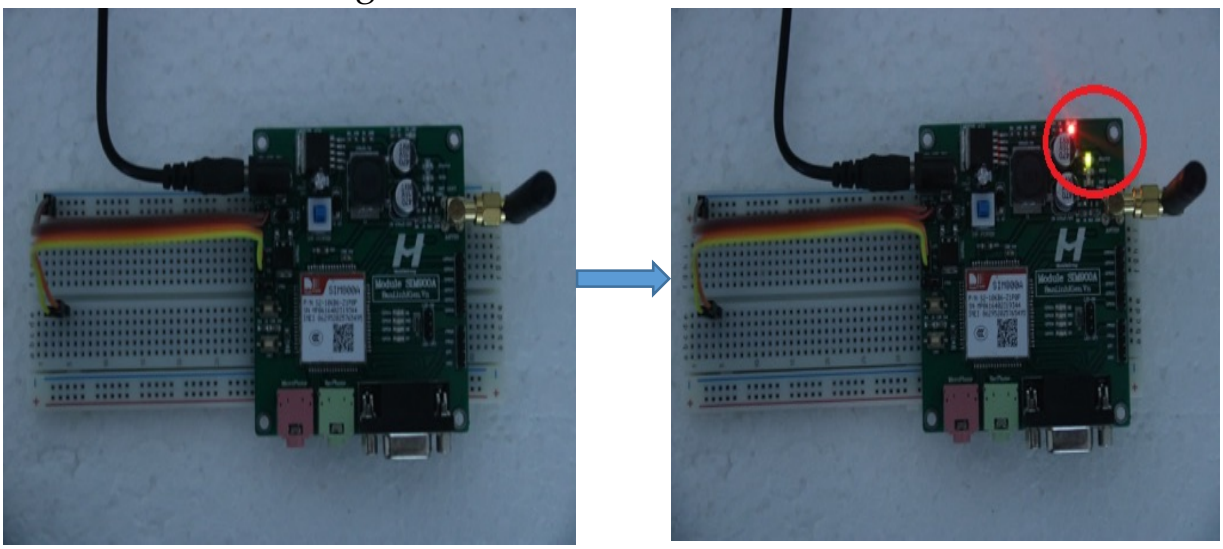

11:36
CÀI ĐẶT NODE
NẠP TIỀN
LỊCH SỬ THAO TÁC
Chế độ
Điều khiển bằng tay
Đặt lịch
Hủy đặt lịch trên node
Trạng thái:
Tắt Node
Thời hạn tưới:
Không thời hạn
Thời gian tưới trong ngày:
10
35
11
36
12
37
Các ngày trong tuần:
17/08/2017
GỬI


11:41
CÀI ĐẶT NODE
NẠP TIỀN
LỊCH SỬ THAO TÁC
Tên
Hành Động
Thời gian
khangpqse03624@fpt.edu.vn
Đặt lịch hằng tuần
Wed Aug 16 18:08:36 GMT+07:00 2017
khangpqse03624@fpt.edu.vn
Bật node trong 000 phút
Wed Aug 16 18:07:42 GMT+07:00 2017
khangpqse03624@fpt.edu.vn
Đặt lịch 1 ngày trong 015 phút , lúc 21:06- ngày: 16//08//17
Wed Aug 16 18:06:59 GMT+07:00 2017
khangpqse03624@fpt.edu.vn
Đặt lịch 1 ngày trong 015 phút , lúc 18:06- ngày: 16//08//17
Wed Aug 16 18:06:56 GMT+07:00 2017
khangpqse03624@fpt.edu.vn
Tắt node
Wed Aug 16 18:06:40 GMT+07:00
ĐỒNG BỘ LOG LÊN SERVER
XEM LOG

II. Evidences

1. *Test Module SIM800A power*

1.1. Image



1.2. Expected results

Module SIM does not work

1.3. Actual results

Module SIM lacked power

1.4. Cause

Power supply cord is loose

1.5. Solution

Plugged in cord again

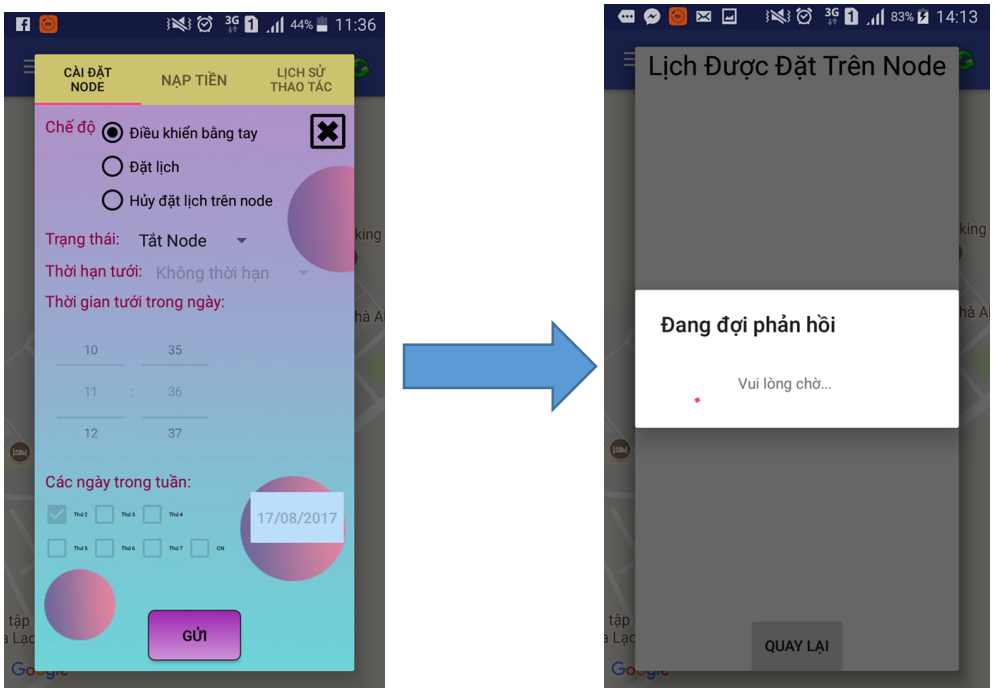
2. *Can't turn OFF by admin account*

2.1. Step details

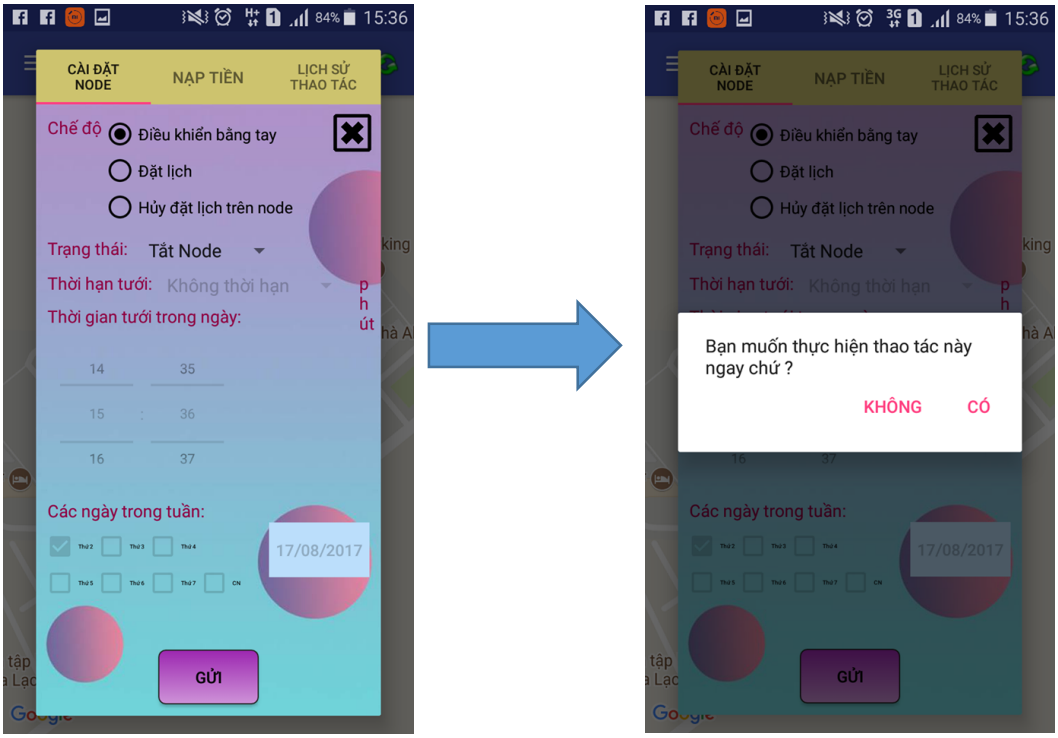
- Choose mode for Node.(Manual control)
- Choose status mode is ON.
- Set time working of Node.
- Click button Send to turn ON Node.
- After about 2-3 minutes, choose status mode is OFF.
- Click button Send to turn OFF Node.

2.2. Images

Get bug:



Fix bug:



2.3. Expected Results

"Cancel Schedule" doesn't appear.

2.4. Actual Results

"Cancel Schedule" appeared.

2.5. Cause

Conflict code: Mode of admin account is same mode of cancel schedule (mode is 3). So when click button Send, system can't get permission of admin account to perform schedule.

2.6. Solution

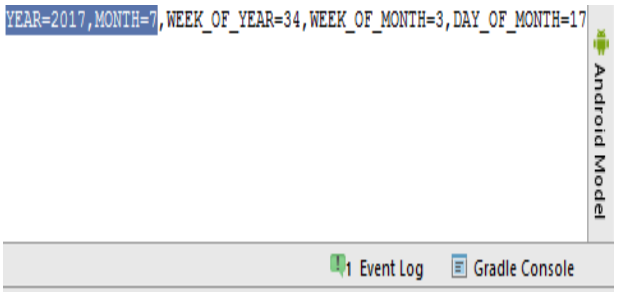
Change mode of admin account different with mode of cancel schedule.

3. Set timer in future is not working.

3.1. Step details

- Choose mode for Node.(Set schedule)
- Choose status mode is one day.
- Set time for working.(about 15 minutes)
- Set start time.(in future)
- Pick date. (in future)
- Click button Send.

3.2. Images



Fix Bug:

```
calendar = Calendar.getInstance();
year = calendar.get(Calendar.YEAR);
month = calendar.get(Calendar.MONTH)+1;
day = calendar.get(Calendar.DAY_OF_MONTH);
```

3.3. Expected Results

Node must turn ON at start time moment.

3.4. Actual Results

Message response is: “Can’t schedule at moment in the past”. System is not working.

3.5. Cause

Conflict code: When use getDate() of DatePicker in code, system get a calendar lacked one month.

3.6. Solution

Add one month to the calendar in the code.

Chapter 6: Conclusion and perspective

In Green- Hoa Lac system, the group have built a small system successfully with some sub-modules:

- A web managing users, nodes information.
- A hardware module controlling the pump.
- An android application interacting with hardware module GHL.

In addition, during a period of four months, all team members have practiced and learned a huge amount of knowledge in many aspects such as: embedded hardware, software, development process. In addition, we also have improved other skills:

- Communication and teamwork
- Time management
- Studying new technologies
- Researching and applying software development process
- Leadership and Q&A skill

Moreover, we will refine the GHLS better for business purposes in the future. We want to develop this system that satisfies customer's requirements as many as possible. We believe that this system can be successful soon by our efforts.

Chapter 7: Reference

[1] DallasSemiConductor, "DS1307," [Online]. Available: http://k2.arduino.vn/img/2015/02/09/0/892_882450-1423489456-0-ds1307.pdf. [Accessed 11 06 2017].

[2] Wikipedia, “GSM,” [Online]. Available: https://en.wikipedia.org/wiki/GSM. [Accessed 22 06 2017].

[3] Arduino, "GSM- library," [Online]. Available: https://www.arduino.cc/en/Reference/GSM. [Accessed 22 06 2017].

[4] MongoDB, "MongoDB," [Online]. Available: https://docs.mongodb.com/manual/introduction/. [Accessed 30 06 2017].

[5] RealmTeam, "Realm," [Online]. Available: https://realm.io/docs/data-model/. [Accessed 22 06 2017].