

Capstone project document

GREEN - HOA LAC



Version 1.2 approved

Prepared by Green Hoa Lac group

Date created: 05-08-2017

Last updated: 08-22-2017

|  |  |  |
| --- | --- | --- |
| 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

[1. Chapter 1: Introduction 11](#_Toc491266559)

[I. Introduction and motivation 11](#_Toc491266560)

[II. Existing products 11](#_Toc491266561)

[1. Schneider Sprinkler Timer and Controller 11](#_Toc491266562)

[2. HT-01 Mobile phone Controller 12](#_Toc491266563)

[III. Scope 13](#_Toc491266564)

[IV. Definitions, Acronyms and Abbreviations 14](#_Toc491266565)

[1. Chapter 2: Project management Plan 15](#_Toc491266566)

[I. Project organization 15](#_Toc491266567)

[1. System process model 15](#_Toc491266568)

[2. Roles and responsibilities 17](#_Toc491266569)

[II. Tools and infrastructures 19](#_Toc491266570)

[1. Hardware 19](#_Toc491266571)

[2. Tools and software 21](#_Toc491266572)

[III. Risk management plan 24](#_Toc491266573)

[1. Risks register 24](#_Toc491266574)

[2. Risk probability and impact 25](#_Toc491266575)

[IV. Communication plan 26](#_Toc491266576)

[1. Project structure 26](#_Toc491266577)

[2. Project communication 26](#_Toc491266578)

[2.1. Format, content, and level of detail of key project information 26](#_Toc491266579)

[2.2. Project report 27](#_Toc491266580)

[V. Project schedule 28](#_Toc491266581)

[3. Chapter 3: System Requirement Specification 31](#_Toc491266582)

[I. Green Hoa Lac Android Application and PCM 31](#_Toc491266583)

[1. Functional Requirement Specification 31](#_Toc491266584)

[1.1. UC01 – Login 31](#_Toc491266585)

[1.2. UC02 – View weather’s information 35](#_Toc491266586)

[1.3. UC03 – View application’s information 37](#_Toc491266587)

[1.4. UC04 – Turn on node 39](#_Toc491266588)

[1.5. UC05 – Turn off node 42](#_Toc491266589)

[1.6. UC06 – Set schedule for node 46](#_Toc491266590)

[1.7. UC07 – View schedules 50](#_Toc491266591)

[1.8. UC08 – Cancel schedules of node 52](#_Toc491266592)

[1.9. UC09 – Recharge 55](#_Toc491266593)

[1.10. UC10 – View node’s history 57](#_Toc491266594)

[1.11. UC11 – Synchronize nodes 59](#_Toc491266595)

[1.12. UC12 – Synchronize node’s history 61](#_Toc491266596)

[1.13. UC13 – Logout 64](#_Toc491266597)

[2. Non- Functional Requirement Specification 66](#_Toc491266598)

[II. Green Hoa Lac Web Management System 67](#_Toc491266599)

[1. Functional Requirement Specification 67](#_Toc491266600)

[1.1. UC01 – Login 67](#_Toc491266601)

[1.2. UC02 – View user’s information 71](#_Toc491266602)

[1.3. UC03 – Add user 73](#_Toc491266603)

[1.4. UC04 – Search user’s information 76](#_Toc491266604)

[1.5. UC05 – Update user’s role 79](#_Toc491266605)

[1.6. UC06 – View user’s history 82](#_Toc491266606)

[1.7. UC07 – Lock user 84](#_Toc491266607)

[1.8. UC08 – Unlock user 87](#_Toc491266608)

[1.9. UC09 – View activity history 89](#_Toc491266609)

[1.10. UC10 – Search activity history 91](#_Toc491266610)

[1.11. UC11 – Add new node 94](#_Toc491266611)

[1.12. UC12 – Update node’s position 97](#_Toc491266612)

[1.13. UC13 – View node’s information 99](#_Toc491266613)

[1.14. UC14 – View node’s history 101](#_Toc491266614)

[1.15. UC15 – Update node’s information 103](#_Toc491266615)

[1.16. UC16 – Delete node 105](#_Toc491266616)

[1.17. UC17 – Sort activity history 108](#_Toc491266617)

[1.18. UC18 – Logout 110](#_Toc491266618)

[2. Non-Functional Requirement Specification 112](#_Toc491266619)

[4. Chapter 4: System design 113](#_Toc491266620)

[I. Architecture design 113](#_Toc491266621)

[1. System architecture design 113](#_Toc491266622)

[2. Serial port (UART) communication protocol 115](#_Toc491266623)

[3. GSM 115](#_Toc491266624)

[II. Mechanical design 116](#_Toc491266625)

[1. Design box 116](#_Toc491266626)

[1.1. Overall 116](#_Toc491266627)

[1.2. Box with all parts 117](#_Toc491266628)

[1.3. Design box 118](#_Toc491266629)

[2. Schematic 119](#_Toc491266630)

[2.1. All parts 119](#_Toc491266631)

[2.2. Detail 120](#_Toc491266632)

[III. Firmware design 127](#_Toc491266633)

[1. SMS syntax 127](#_Toc491266634)

[2. Some main flows 128](#_Toc491266635)

[2.1. Get on 128](#_Toc491266636)

[2.2. Get off 129](#_Toc491266637)

[2.3. Set schedule 130](#_Toc491266638)

[2.4. Get schedule 131](#_Toc491266639)

[2.5. Cancel schedule 132](#_Toc491266640)

[IV. Software design 133](#_Toc491266641)

[1. Login 133](#_Toc491266642)

[2. Turn on 134](#_Toc491266643)

[3. Turn off 135](#_Toc491266644)

[4. Set schedule 136](#_Toc491266645)

[5. Cancel Schedule 137](#_Toc491266646)

[6. Synchronize node 138](#_Toc491266647)

[1. Server database 139](#_Toc491266648)

[2. Android application database 140](#_Toc491266649)

[5. Chapter 5: Implement and testing 141](#_Toc491266650)

[I. Implement 141](#_Toc491266651)

[1. Version 1.0 141](#_Toc491266652)

[2. Version 1.1 143](#_Toc491266653)

[II. Evidences 145](#_Toc491266654)

[1. Test Module SIM800A power 145](#_Toc491266655)

[1.1. Image 145](#_Toc491266656)

[1.2. Expected results 145](#_Toc491266657)

[1.3. Actual results 145](#_Toc491266658)

[1.4. Cause 145](#_Toc491266659)

[1.5. Solution 145](#_Toc491266660)

[2. Can’t turn OFF by admin account 145](#_Toc491266661)

[2.1. Step details 145](#_Toc491266662)

[2.2. Images 146](#_Toc491266663)

[2.3. Expected Results 147](#_Toc491266664)

[2.4. Actual Results 147](#_Toc491266665)

[2.5. Cause 147](#_Toc491266666)

[2.6. Solution 147](#_Toc491266667)

[3. Set timer in future is not working. 147](#_Toc491266668)

[3.1. Step details 147](#_Toc491266669)

[3.2. Images 147](#_Toc491266670)

[3.3. Expected Results 148](#_Toc491266671)

[3.4. Actual Results 148](#_Toc491266672)

[3.5. Cause 148](#_Toc491266673)

[3.6. Solution 148](#_Toc491266674)

[6. Chapter 6: Conclusion and perspective 149](#_Toc491266675)

[7. Chapter 7: Reference 150](#_Toc491266676)

[Figure 1.1: Schneider Sprinkler Timer and Controller 11](#_Toc491266677)

[Figure 1.2: HT-01 Mobile phone Controller 12](#_Toc491266678)

[Figure 2.1: The Incremental Development Model 15](#_Toc491266679)

[Figure 2.2: The Iterative Development Model 15](#_Toc491266680)

[Figure 2.3: System Process Model 16](#_Toc491266681)

[Figure 2.4: Risk register 24](#_Toc491266682)

[Figure 2.5: Risk probability and impact 25](#_Toc491266683)

[Figure 2.6: Project structure 26](#_Toc491266684)

[Figure 2.7: Project schedule 30](#_Toc491266685)

[Figure 3.1: Login application screen 31](file:///C:\Users\khang\Downloads\GreenHoaLacDocument_%20Version_1.2%20(1).docx#_Toc491266686)

[Figure 3.2: Google sign in screen 31](#_Toc491266687)

[Figure 3.3: Home screen 31](#_Toc491266688)

[Figure 3.4: Navigation drawer screen 31](#_Toc491266689)

[Figure 3.5: Login application Use case diagram 32](#_Toc491266690)

[Figure 3.6: View weather's information screen 35](#_Toc491266691)

[Figure 3.7: View weather's information Use case diagram 36](#_Toc491266692)

[Figure 3.8: View application's information screen 37](#_Toc491266693)

[Figure 3.9: View application's information Use case diagram 38](#_Toc491266694)

[Figure 3.10: Turn on node screen 39](#_Toc491266695)

[Figure 3.11: Turn on node Use case diagram 41](#_Toc491266696)

[Figure 3.12: Turn off node screen 42](#_Toc491266697)

[Figure 3.13: Turn off node Use case diagram 44](file:///C:\Users\khang\Downloads\GreenHoaLacDocument_%20Version_1.2%20(1).docx#_Toc491266698)

[Figure 3.14: Set schedule for node 46](#_Toc491266699)

[Figure 3.15: Set schedule for node Use case diagram 48](#_Toc491266700)

[Figure 3.16: View Schedules 50](#_Toc491266701)

[Figure 3.17: View schedules of node Use case diagram 51](#_Toc491266702)

[Figure 3.18: Cancel schedule for node screen 52](#_Toc491266703)

[Figure 3.19: Cancel schedule for node Use case diagram 53](#_Toc491266704)

[Figure 3.20: Recharge screen 55](#_Toc491266705)

[Figure 3.21: Recharge Use case diagram 56](#_Toc491266706)

[Figure 3.22: View node's history screen 57](#_Toc491266707)

[Figure 3.23: View node's history Use case diagram 58](#_Toc491266708)

[Figure 3.24: Synchronize nodes Screen 59](#_Toc491266709)

[Figure 3.25: Synchronize nodes Use case diagram 60](#_Toc491266710)

[Figure 3.26: Synchronize node's history Screen 61](#_Toc491266711)

[Figure 3.27: Synchronize node's history Use case diagram 62](#_Toc491266712)

[Figure 3.28: Logout Screen 64](#_Toc491266713)

[Figure 3.29: Logout Use case diagram 65](#_Toc491266714)

[Figure 3.30: Login Screen 67](#_Toc491266715)

[Figure 3.31: Google Sign in Screen 68](file:///C:\Users\khang\Downloads\GreenHoaLacDocument_%20Version_1.2%20(1).docx#_Toc491266716)

[Figure 3.32: Home Page 68](#_Toc491266717)

[Figure 3.33: Login Use Case Diagram 69](#_Toc491266718)

[Figure 3.34: View user's information Screen 71](#_Toc491266719)

[Figure 3.35: View user's information Use case diagram 72](#_Toc491266720)

[Figure 3.36: Add user screen 73](#_Toc491266721)

[Figure 3.37: Add user Use case diagram 74](#_Toc491266722)

[Figure 3.38: Search user's information screen 76](#_Toc491266723)

[Figure 3.39: Search user's information Use case diagram 77](#_Toc491266724)

[Figure 3.40: Update user's role screen 79](#_Toc491266725)

[Figure 3.41: Update user's role Use case diagram 80](#_Toc491266726)

[Figure 3.42: View user's history screen 82](#_Toc491266727)

[Figure 3.43: View user's history Use case diagram 83](#_Toc491266728)

[Figure 3.44: Lock user screen 84](#_Toc491266729)

[Figure 3.45: Lock user Use case diagram 85](#_Toc491266730)

[Figure 3.46: Unlock user screen 87](#_Toc491266731)

[Figure 3.47: Unlock user Use case diagram 88](#_Toc491266732)

[Figure 3.48: View activity history screen 89](#_Toc491266733)

[Figure 3.49: View activity history Use case diagram 90](#_Toc491266734)

[Figure 3.50: Search activity history screen 91](#_Toc491266735)

[Figure 3.51: Search activity history Use case diagram 92](#_Toc491266736)

[Figure 3.52: Search activity history Use case diagram 94](#_Toc491266737)

[Figure 3.53: Add new node Use case diagram 95](#_Toc491266738)

[Figure 3.54: Update node's position screen 97](#_Toc491266739)

[Figure 3.55: Update node's position Use case diagram 97](#_Toc491266740)

[Figure 3.56: View node's information screen 99](#_Toc491266741)

[Figure 3.57: View node's information Use case diagram 100](#_Toc491266742)

[Figure 3.58: View node's history screen 101](#_Toc491266743)

[Figure 3.59: View node's history Use case diagram 102](#_Toc491266744)

[Figure 3.60: Update node's information screen 103](#_Toc491266745)

[Figure 3.61: Update node's information Use case diagram 104](#_Toc491266746)

[Figure 3.62: Delete node screen 105](#_Toc491266747)

[Figure 3.63: Delete node Use case diagram 106](#_Toc491266748)

[Figure 3.64: Sort activity history screen 108](#_Toc491266749)

[Figure 3.65: Sort activity history Use case diagram 109](#_Toc491266750)

[Figure 3.66: Logout screen 110](#_Toc491266751)

[Figure 3.67: Logout Use case diagram 111](#_Toc491266752)

[Figure 4.1: Architecture design 113](#_Toc491266753)

[Figure 4.2: Serial port (UART) communication protocol 115](#_Toc491266754)

[Figure 4.3: Real box 116](#_Toc491266755)

[Figure 4.4: Box 3D 116](#_Toc491266756)

[Figure 4.5: Box 3D with all parts 117](#_Toc491266757)

[Figure 4.6: Note 117](#_Toc491266758)

[Figure 4.7: Box 118](#_Toc491266759)

[Figure 4.8: The vertical projection 118](#_Toc491266760)

[Figure 4.9: Top view 118](#_Toc491266761)

[Figure 4.10: Schematic 119](#_Toc491266762)

[Figure 4.11: SIM and Power 120](#_Toc491266763)

[Figure 4.12: SIM and Arduino 121](#_Toc491266764)

[Figure 4.13: Arduino and Relay 122](#_Toc491266765)

[Figure 4.14: Relay, pump and adapter 123](#_Toc491266766)

[Figure 4.15: Power and Arduino 124](#_Toc491266767)

[Figure 4.16: Arduino and DS1307 125](#_Toc491266768)

[Figure 4.17: Arduino and Adapter 126](#_Toc491266769)

[Figure 4.18: SMS syntax 127](#_Toc491266770)

[Figure 4.19: Get on 128](#_Toc491266771)

[Figure 4.20: Get off 129](#_Toc491266772)

[Figure 4.21: Set schedule 130](#_Toc491266773)

[Figure 4.22: Get schedule 131](#_Toc491266774)

[Figure 4.23: Cancel schedule 132](#_Toc491266775)

[Figure 4.24: Login sequence diagram 133](#_Toc491266776)

[Figure 4.25: Turn on node sequence diagram 134](#_Toc491266777)

[Figure 4.26: Turn off node sequence diagram 135](#_Toc491266778)

[Figure 4.27: Set schedule sequence diagram 136](#_Toc491266779)

[Figure 4.28: Cancel Schedule 137](#_Toc491266780)

[Figure 4.29: Synchronize node sequence diagram 138](#_Toc491266781)

[Figure 4.30: Server database (MongoDB) 139](#_Toc491266782)

[Figure 4.31: Realm database design 140](#_Toc491266783)

[Table 1.1: Scope and boundaries of whole system 13](#_Toc491266784)

[Table 1.2: Scope and boundaries of whole system 14](#_Toc491266785)

[Table 2.1: Roles and responsibilities 17](#_Toc491266786)

[Table 2.2: List hardware devices 19](#_Toc491266787)

[Table 2.3: Tools and software 21](#_Toc491266788)

[Table 2.4: Format, content, and level of detail 26](#_Toc491266789)

[Table 2.5: Project report 27](#_Toc491266790)

[Table 3.1: Login application Screen definition 32](#_Toc491266791)

[Table 3.2: View weather's information Screen definition 35](#_Toc491266792)

[Table 3.3: View application's information Screen definition 38](#_Toc491266793)

[Table 3.4: Turn on node Screen definition 40](#_Toc491266794)

[Table 3.5: Turn off node Screen definition 43](#_Toc491266795)

[Table 3.6: Set schedule for node Screen definition 46](#_Toc491266796)

[Table 3.7: View schedules Screen definition 50](#_Toc491266797)

[Table 3.8: Cancel schedule for node Screen definition 52](#_Toc491266798)

[Table 3.9: Recharge Screen definition 55](#_Toc491266799)

[Table 3.10: View note's history Screen definition: 57](#_Toc491266800)

[Table 3.11: Synchronize nodes Screen definition 59](#_Toc491266801)

[Table 3.12: Synchronize node's history Screen Definition 62](#_Toc491266802)

[Table 3.13: Logout Screen definition 64](#_Toc491266803)

[Table 3.14: Nonfunctional requirements for GHLA 66](#_Toc491266804)

[Table 3.15: Login Screen Definition 69](#_Toc491266805)

[Table 3.16: View user information screen information 71](#_Toc491266806)

[Table 3.17: Add user Screen definition 73](#_Toc491266807)

[Table 3.18: Search user's information Screen definition 77](#_Toc491266808)

[Table 3.19: Update user's role Screen definition 79](#_Toc491266809)

[Table 3.20: View user's history Screen definition 82](#_Toc491266810)

[Table 3.21: Log user Screen definition 84](#_Toc491266811)

[Table 3.22: Unlock user Screen definition 87](#_Toc491266812)

[Table 3.23: View activity history Screen definition 90](#_Toc491266813)

[Table 3.24: Search activity history Screen definition 92](#_Toc491266814)

[Table 3.25: Add new node Screen definition 94](#_Toc491266815)

[Table 3.26: Update node position Screen definition 97](#_Toc491266816)

[Table 3.27: View node's information Screen definition 99](#_Toc491266817)

[Table 3.28: View node's history screen definition 101](#_Toc491266818)

[Table 3.29: Update node's information Screen definition 103](#_Toc491266819)

[Table 3.30: Delete node Screen definition 106](#_Toc491266820)

[Table 3.31: Sort activity history Screen definition 108](#_Toc491266821)

[Table 3.32: Logout Screen definition 110](#_Toc491266822)

[Table 3.33: Nonfunctional requirements for GHLWMS 112](#_Toc491266823)

## Chapter 1: Introduction

### 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.

### 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.

#### 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

#### HT-01 Mobile phone Controller

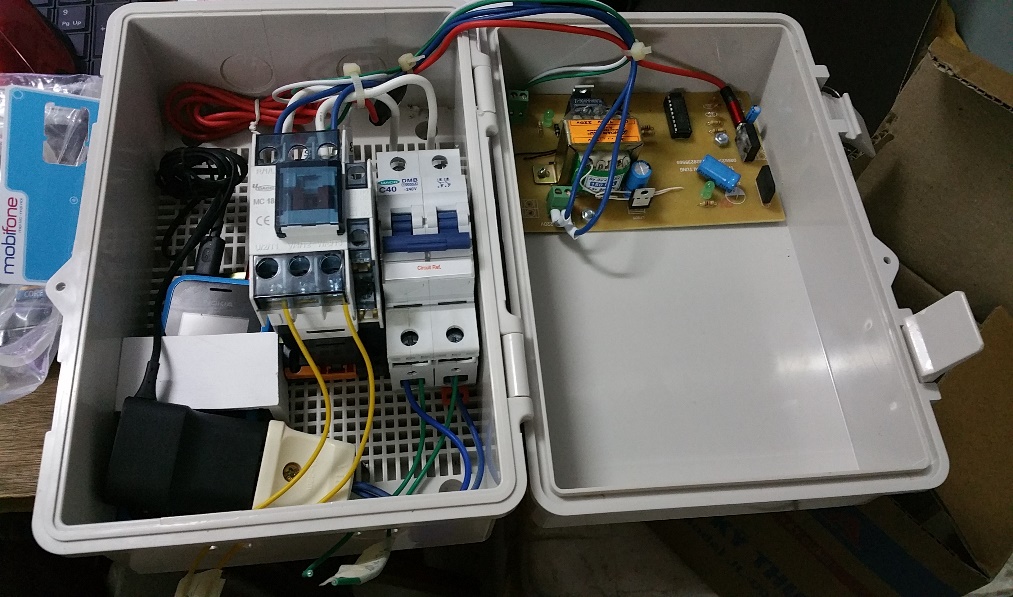


Figure 1.2: HT-01 Mobile phone Controller

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

### 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. |

### 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

### Project organization

#### 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.

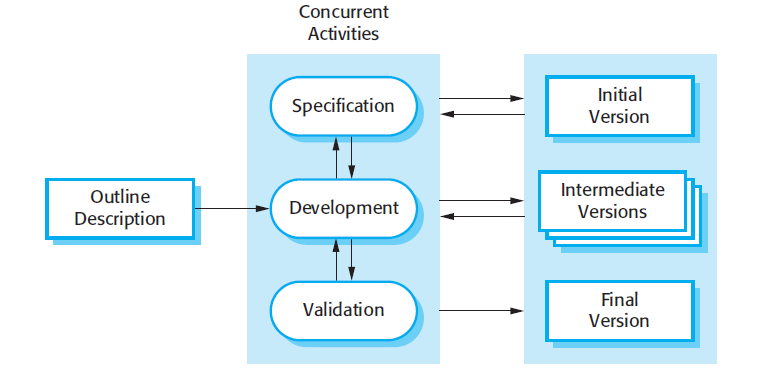


Figure 2.1: The Incremental Development Model

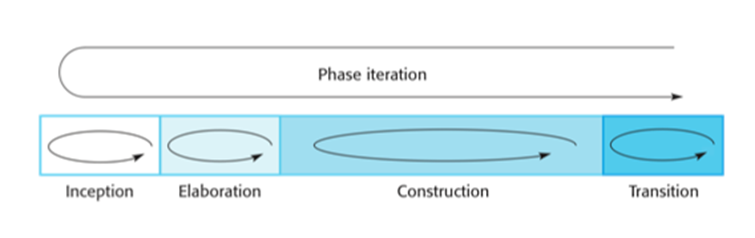


Figure 2.2: The Iterative Development Model

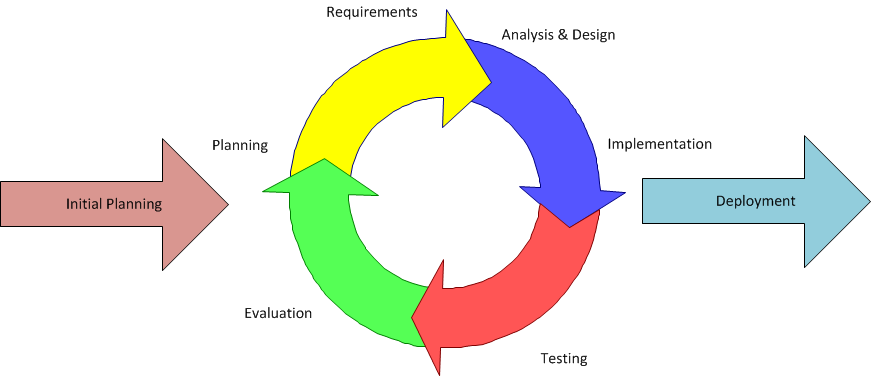


Figure 2.3: System Process Model

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.

#### 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. |

### Tools and infrastructures

#### 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 |

#### 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 |

### Risk management plan

#### Risks register

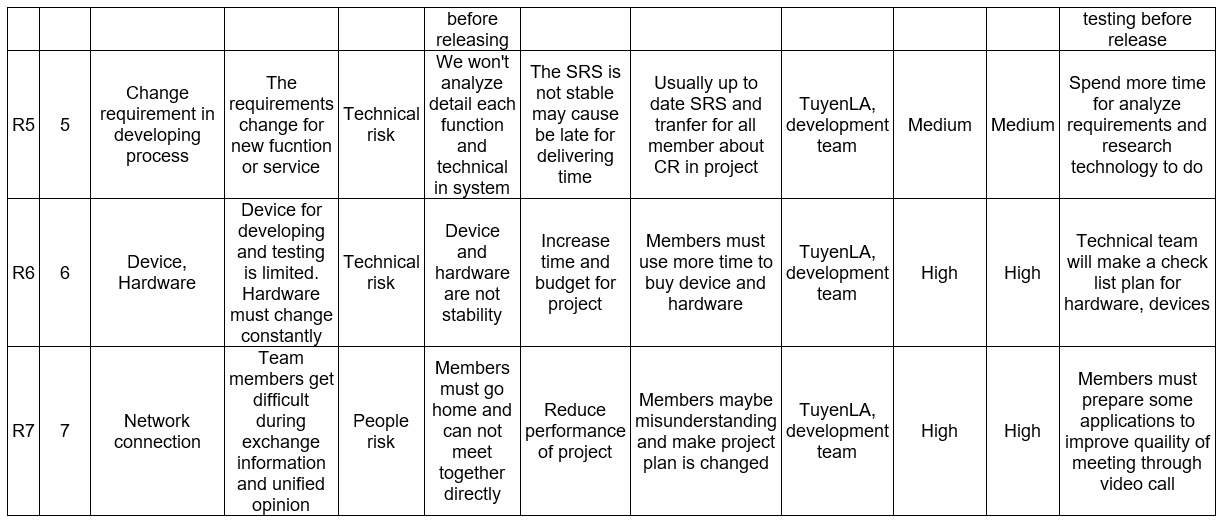
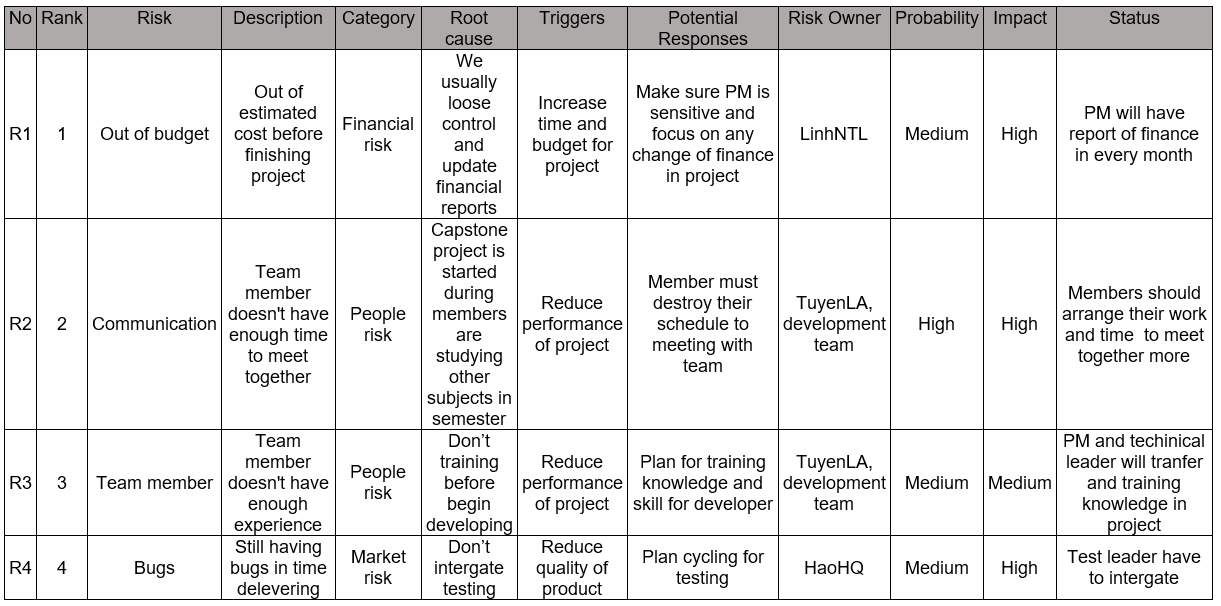


Figure 2.4: Risk register

#### 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

### Communication plan

#### Project structure

Project team

**Project Leader**

Pham Quang Khang

**Supervisor**

**Technical Leader**

Le Anh Tuyen

**Main Supervisor**

Ph.D. Phan Duy Hung

**Green – Hoa Lac group**

**Test Leader**

Ho Quang Hao

Ho Quang   
Hao

**QA  
TEAM**

**DEV  
TEAM**

Nguyen Thi Ly Linh

Le Anh Tuyen

Nguyen Thi Ly Linh

Lam Duc Thang

Bui Manh Tri

Pham Quang Khang

Figure 2.6: Project structure

#### Project communication

##### Format, content, and level of detail of key project information

Table 2.4: Format, content, and level of detail

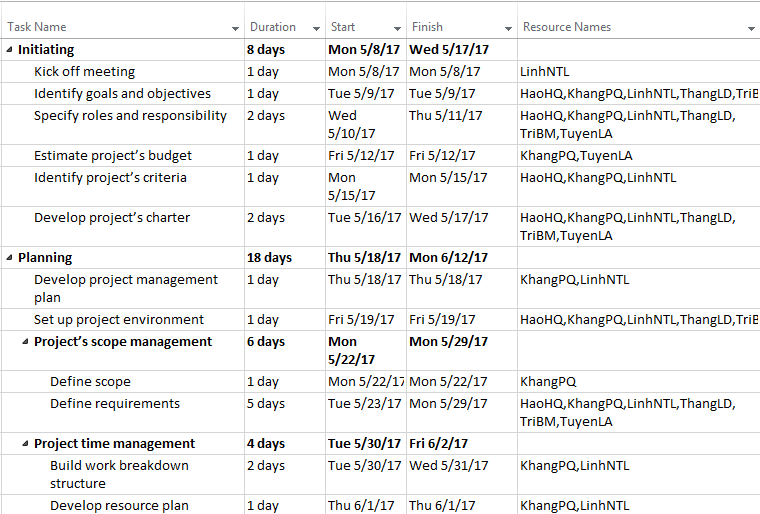
|  |  |  |  |
| --- | --- | --- | --- |
| Information | Author | Receiver | Method/Technology |
| Schedule updates | Project manager | Project team  Supervisor | Email  Group meeting |
| Project status | Project manager | Project team  Supervisor | Email  Group meeting |
| Agenda/Meeting Minutes | Project 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 |

##### Project report

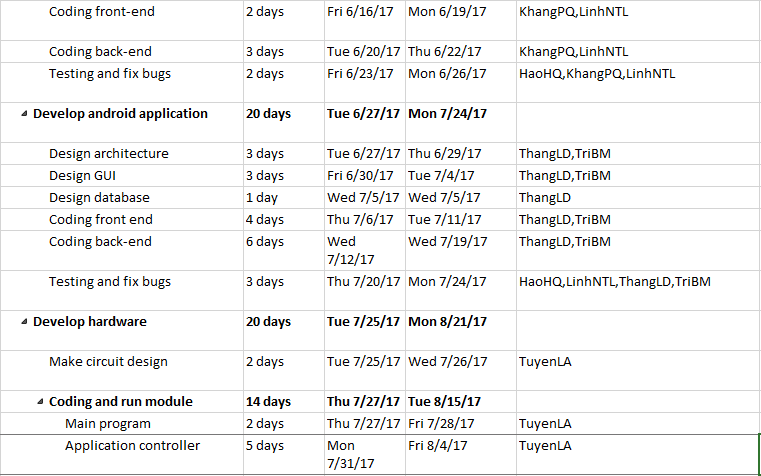
Table 2.5: Project report

|  |  |  |  |
| --- | --- | --- | --- |
| 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  Distribution list: Nguyen Thi Ly Linh, Le Anh Tuyen,  Ho Quang Hao | Friday | E-mail |
| 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 |

### Project schedule







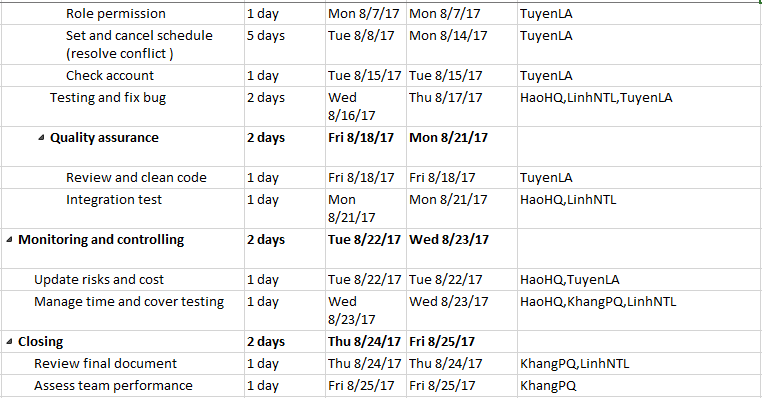


Figure 2.7: Project schedule

## Chapter 3: System Requirement Specification

### Green Hoa Lac Android Application and PCM

#### Functional Requirement Specification

##### UC01 – Login

###### Screen Design

|  |  |
| --- | --- |
| Figure 3.1: Login application screen | 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. |

###### Use case specification

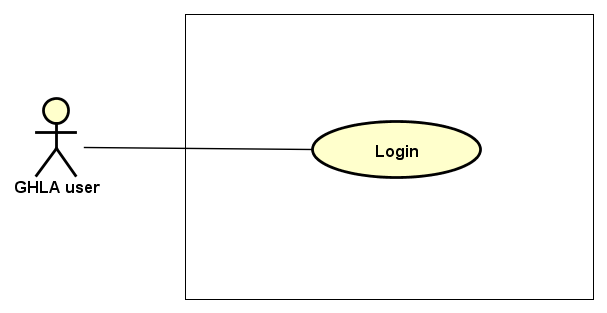


Figure 3.5: Login application Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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. |

##### UC02 – View weather’s information

###### Screen Design

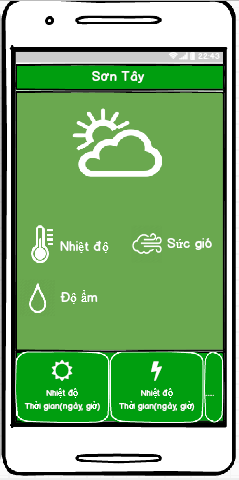


Figure 3.6: View weather's information screen

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 : oC |
| 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 |

###### Use case specification

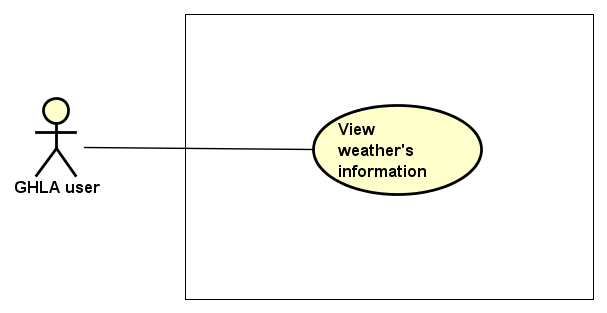


Figure 3.7: View weather's information Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC03 – View application’s information

###### Screen Design

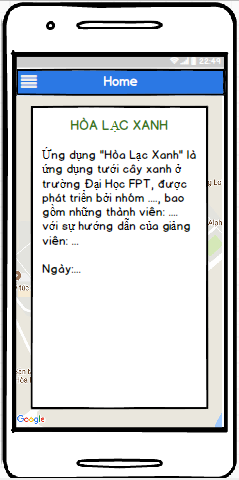


Figure 3.8: View application's information screen

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 |  |  |  |

###### Use case specification

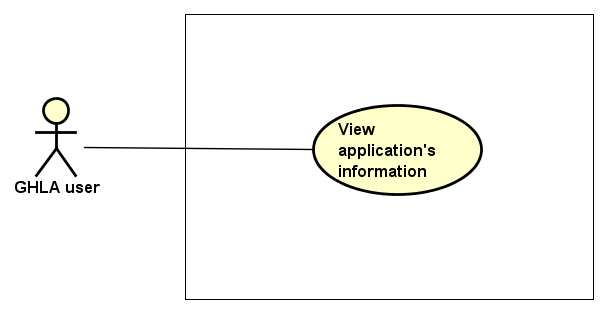


Figure 3.9: View application's information Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC04 – Turn on node

###### 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 |

###### Use case specification

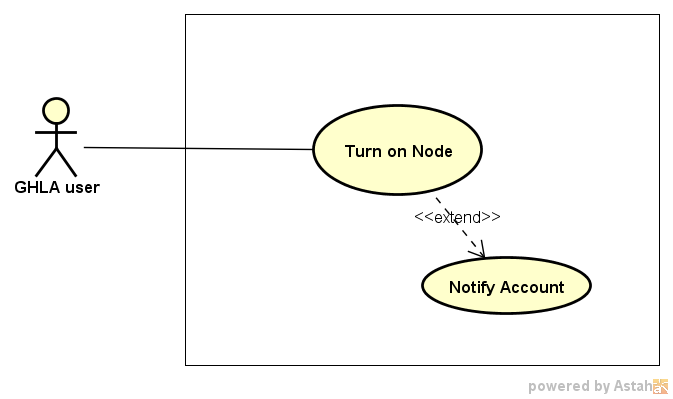


Figure 3.11: Turn on node Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |
|  |  |  | | | | |

##### UC05 – Turn off node

###### 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 |

###### Use case specification

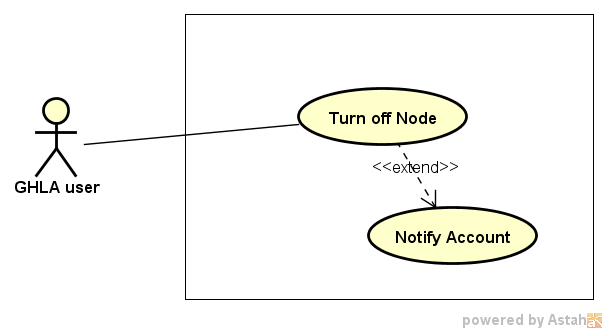


Figure 3.13: Turn off node Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC06 – Set schedule for node

###### Screen Design

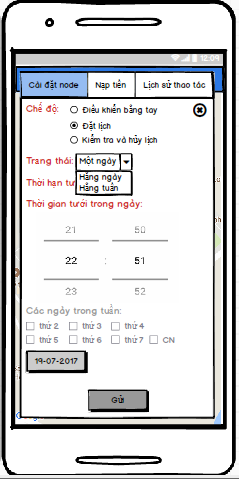


Figure 3.14: Set schedule for node

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 |

###### Use case specification

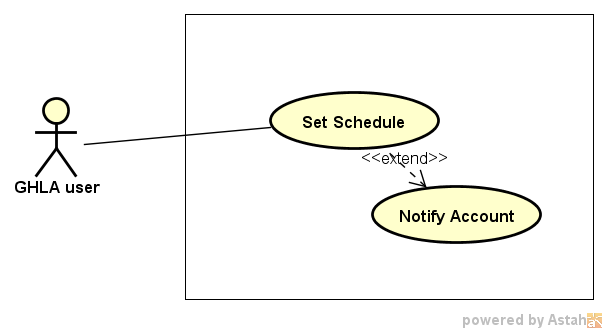


Figure 3.15: Set schedule for node Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC07 – View schedules

###### 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 |

###### Use case specification

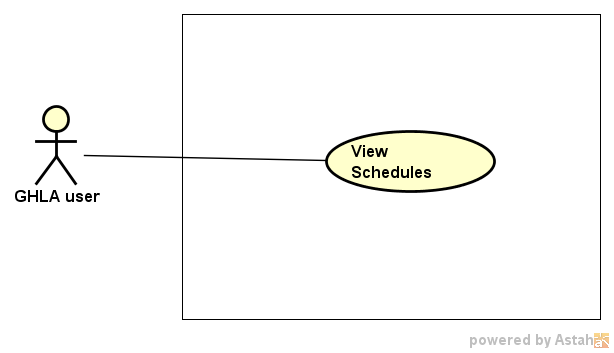


Figure 3.17: View schedules of node Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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. | | | | |

##### UC08 – Cancel schedules of node

###### Screen Design

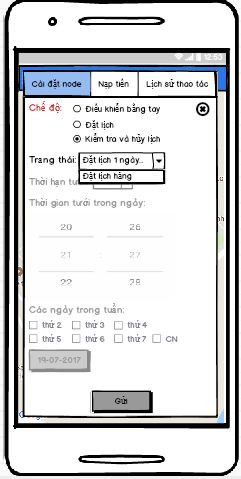


Figure 3.18: Cancel schedule for node screen

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 |

###### Use case specification

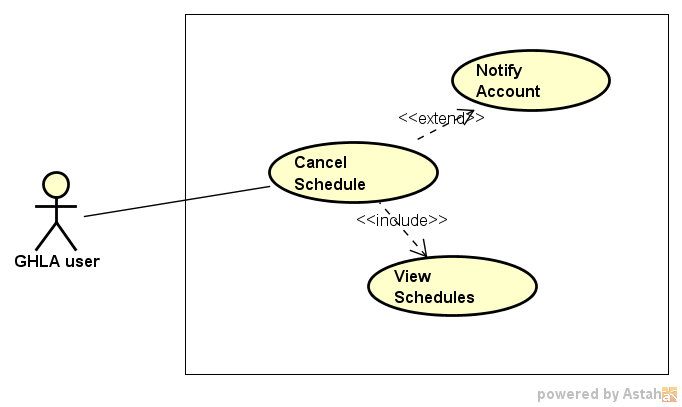


Figure 3.19: Cancel schedule for node Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC09 – Recharge

###### 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 |  |  |  |

###### Use case specification

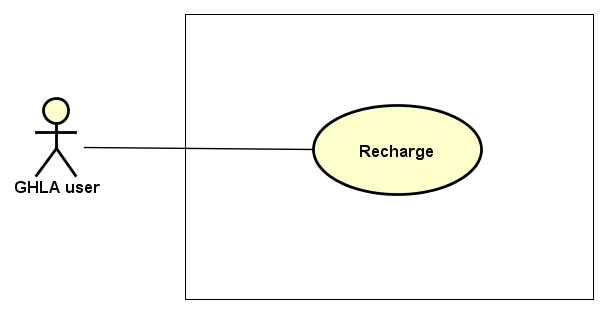


Figure 3.21: Recharge Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC10 – View node’s history

###### 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 |  |  |  |

###### Use case specification

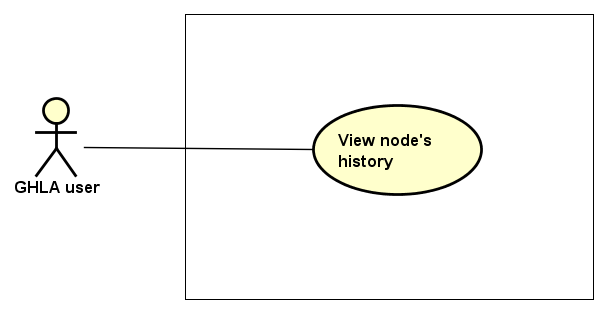


Figure 3.23: View node's history Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC11 – Synchronize nodes

###### 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 |

###### Use case specification

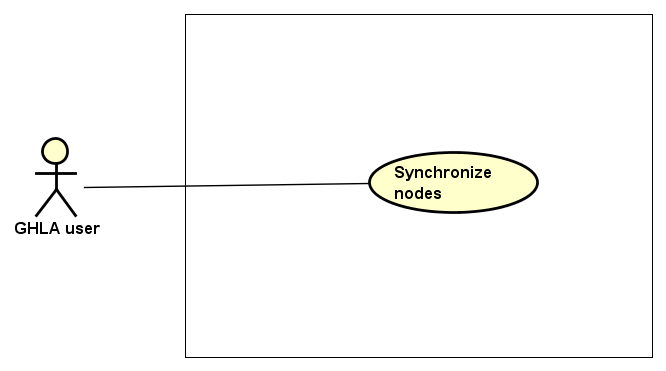


Figure 3.25: Synchronize nodes Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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) |

##### UC12 – Synchronize node’s history

###### Screen Design

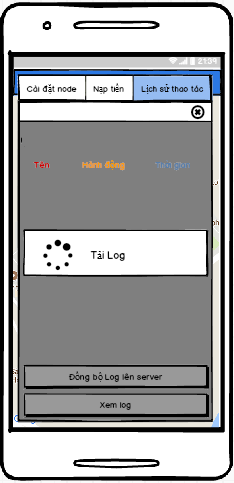


Figure 3.26: Synchronize node's history Screen

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 |  |  |  |

###### Use case specification

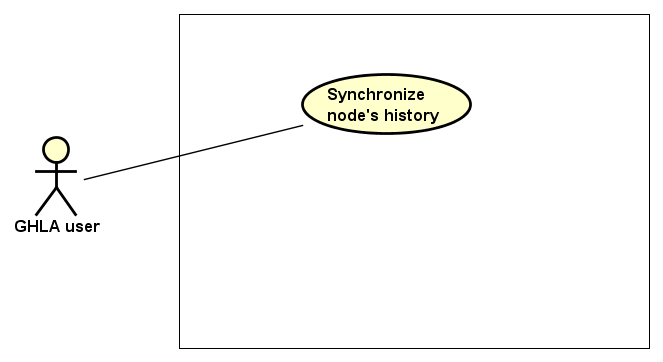


Figure 3.27: Synchronize node's history Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC13 – Logout

###### Screen Design

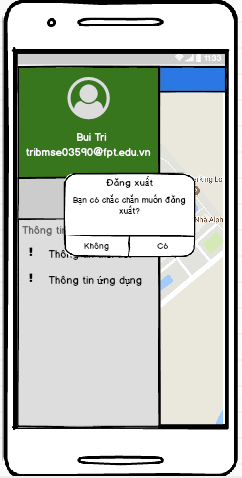


Figure 3.28: Logout Screen

Table 3.13: Logout Screen definition

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

###### Use case specification

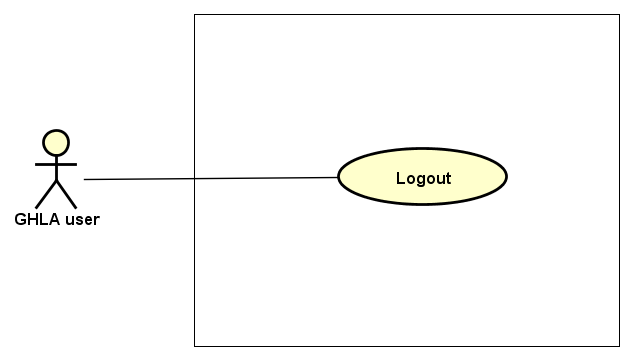


Figure 3.29: Logout Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

#### 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 |

### Green Hoa Lac Web Management System

#### Functional Requirement Specification

##### UC01 – Login

###### Screen Design

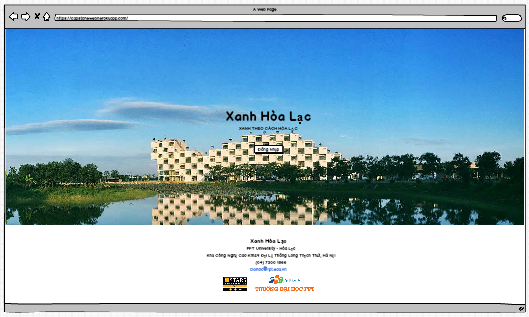


Figure 3.30: Login Screen

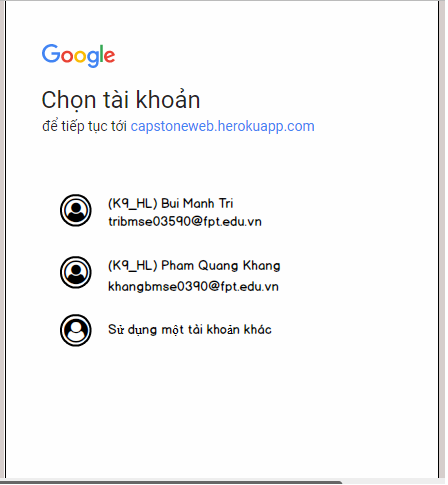


Figure 3.31: Google Sign in Screen

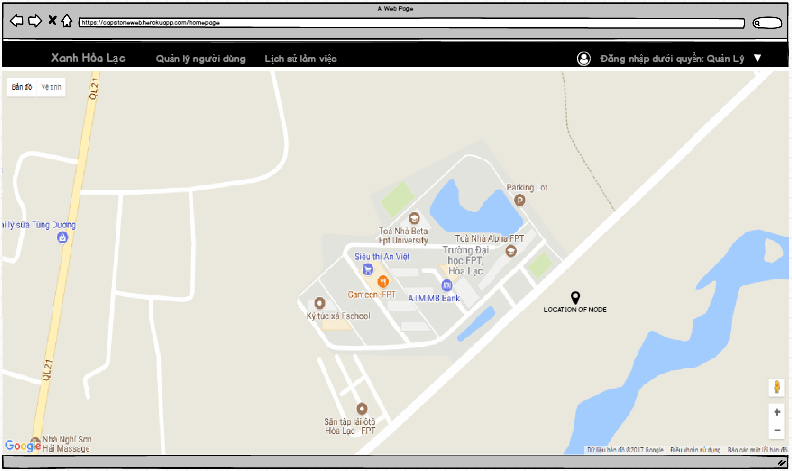


Figure 3.32: Home Page

Table 3.15: Login Screen Definition

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

###### Use case specification

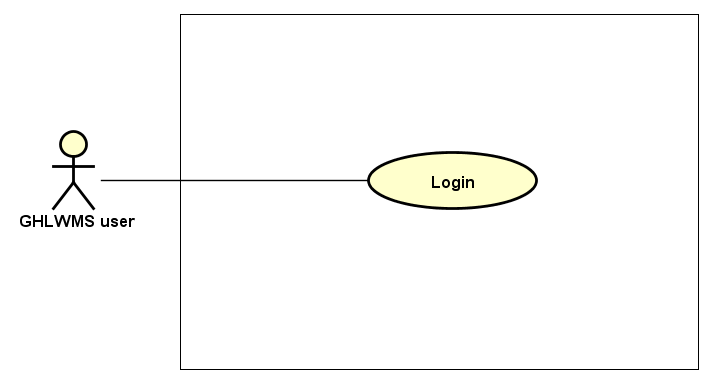


Figure 3.33: Login Use Case Diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC02 – View user’s information

###### Screen Design

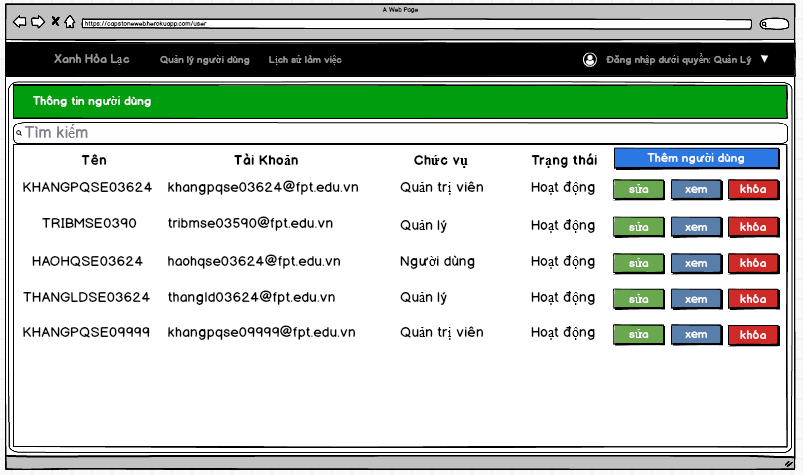


Figure 3.34: View user's information Screen

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. |

###### Use case specification

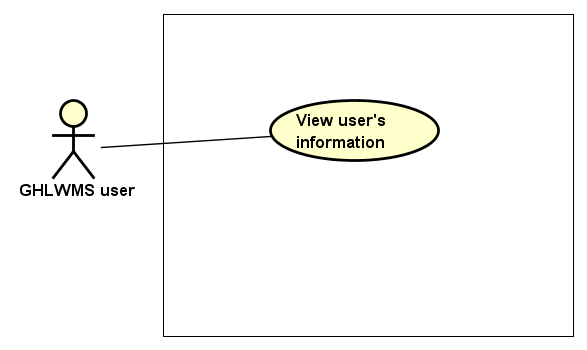


Figure 3.35: View user's information Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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) | | | | |

##### UC03 – Add user

###### Screen Design

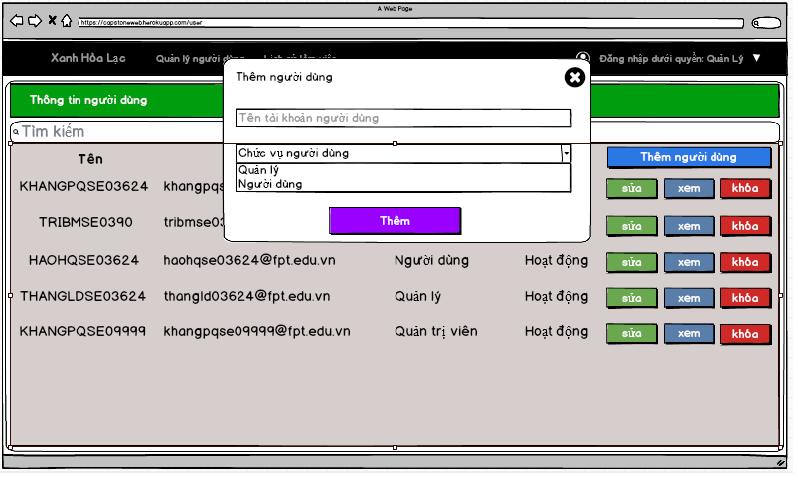


Figure 3.36: Add user screen

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 |  |  |  |

###### Use case specification

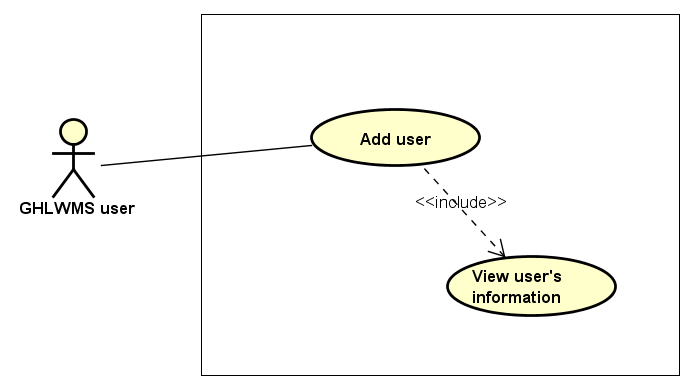


Figure 3.37: Add user Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC04 – Search user’s information

###### Screen Design

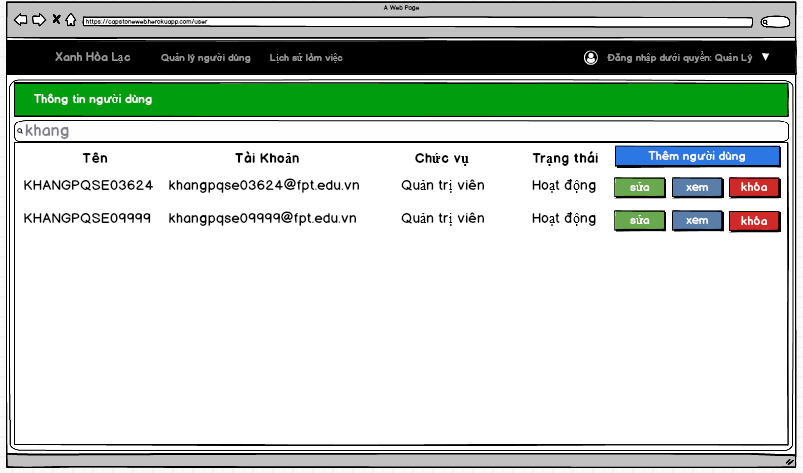


Figure 3.38: Search user's information screen

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) |

###### Use case specification

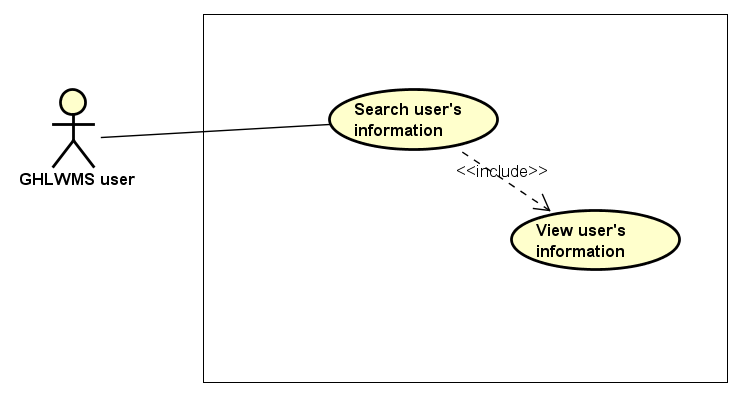


Figure 3.39: Search user's information Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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”. |

##### UC05 – Update user’s role

###### Screen Design

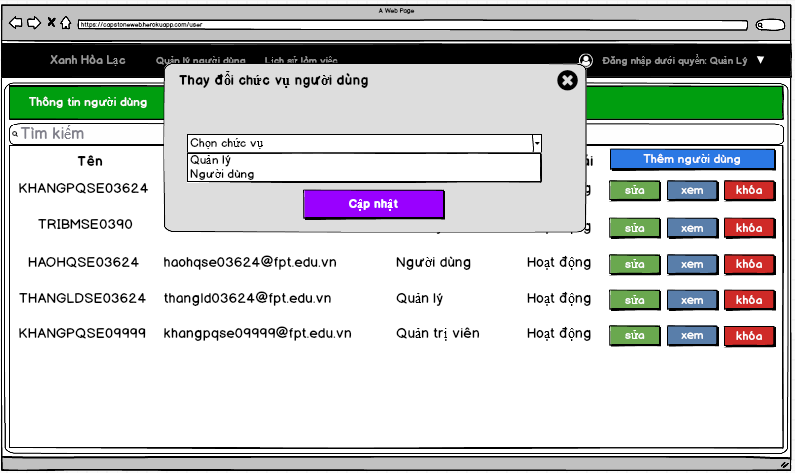


Figure 3.40: Update user's role screen

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 |  |  |  |

###### Use case specification

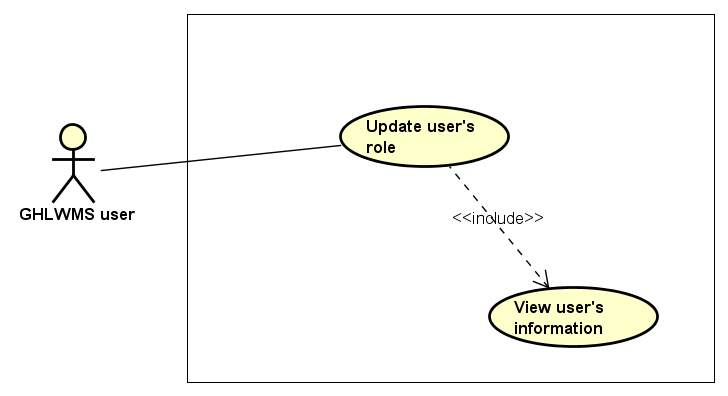


Figure 3.41: Update user's role Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC06 – View user’s history

###### Screen Design

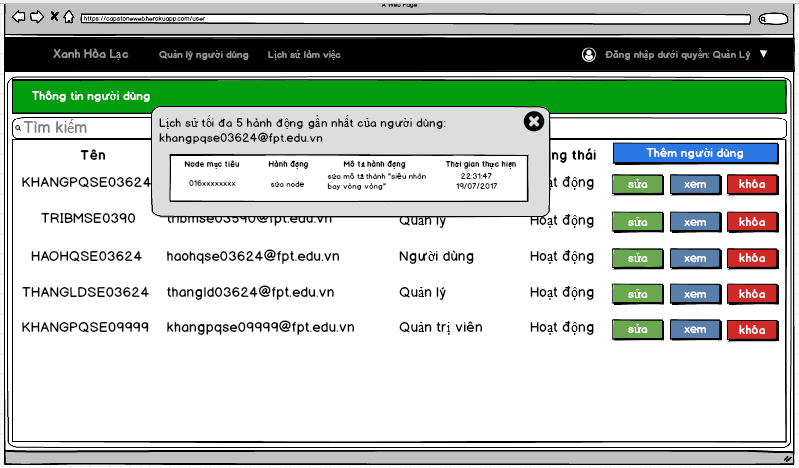


Figure 3.42: View user's history screen

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 |

###### Use case specification

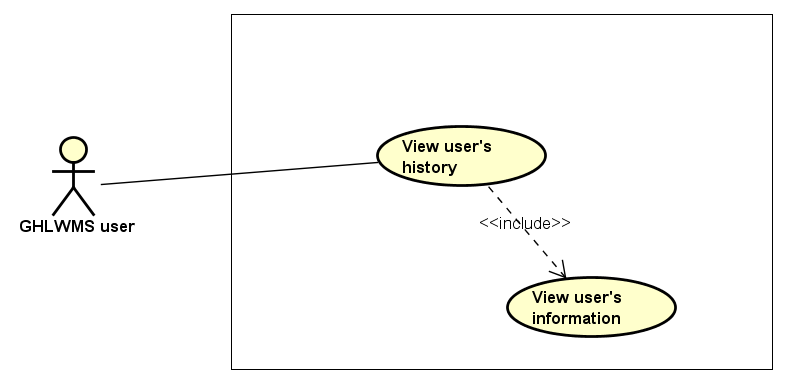


Figure 3.43: View user's history Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC07 – Lock user

###### Screen Design

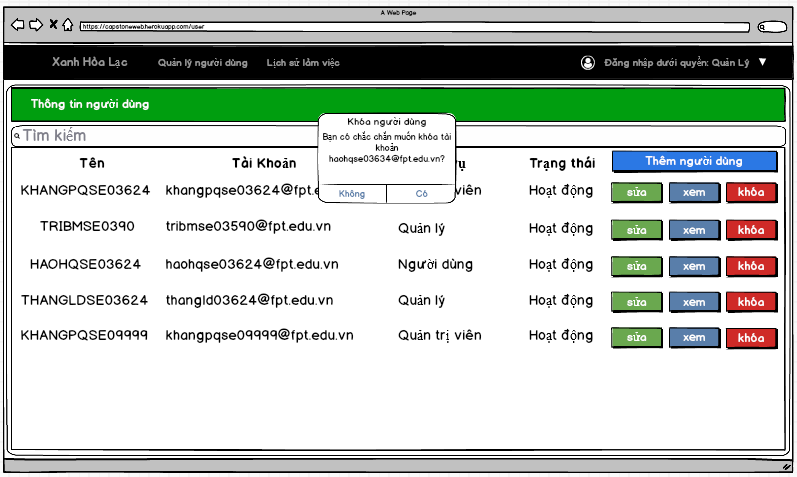


Figure 3.44: Lock user screen

Table 3.21: Log user Screen definition

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

###### Use case specification

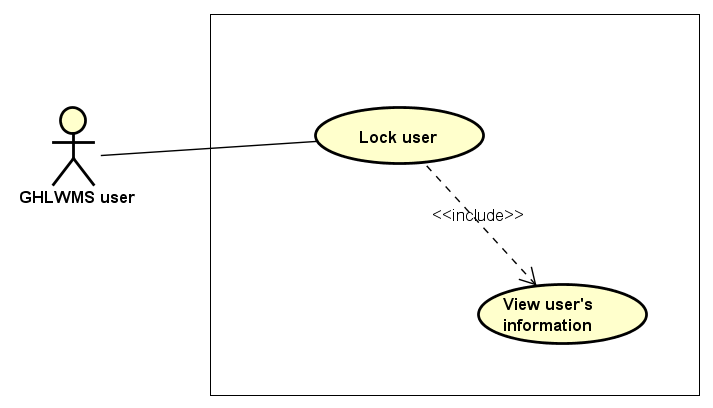


Figure 3.45: Lock user Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC08 – Unlock user

###### Screen Design

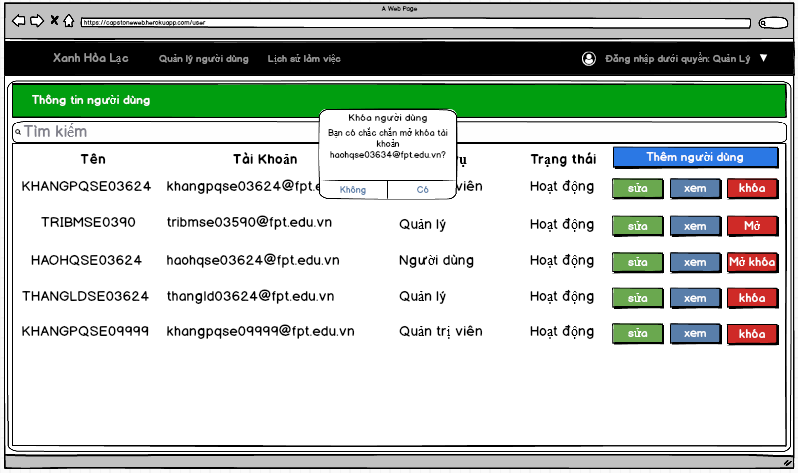


Figure 3.46: Unlock user screen

Table 3.22: Unlock user Screen definition

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

###### Use case specification

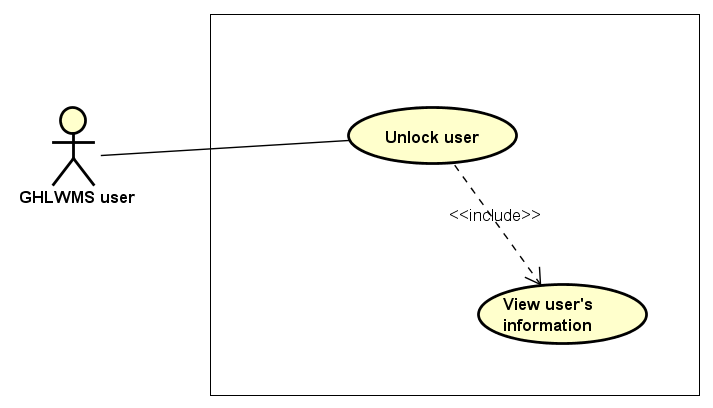


Figure 3.47: Unlock user Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC09 – View activity history

###### Screen Design

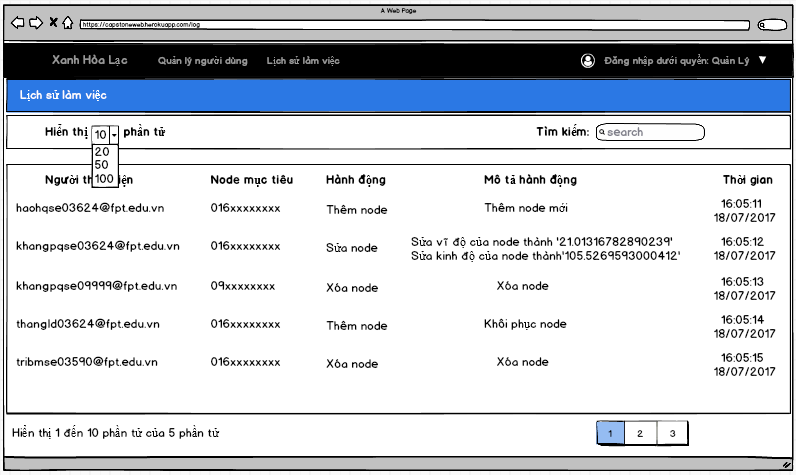


Figure 3.48: View activity history screen

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 |

###### Use case specification

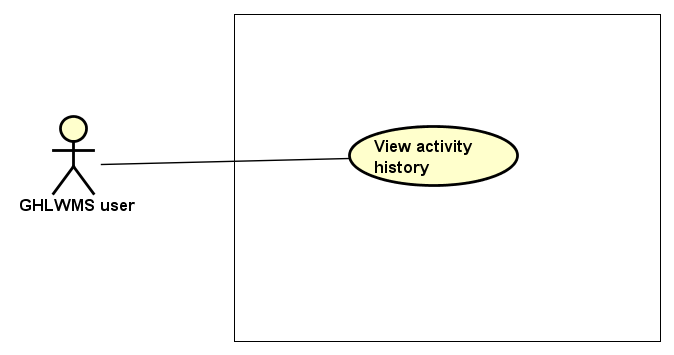


Figure 3.49: View activity history Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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. |

##### UC10 – Search activity history

###### Screen Design

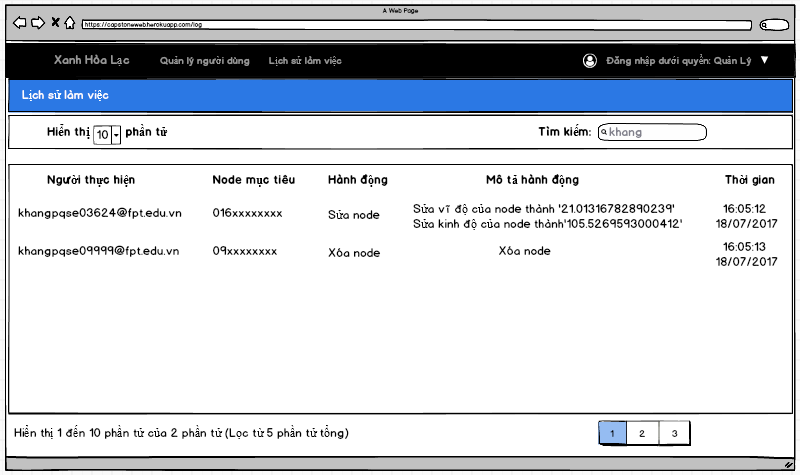


Figure 3.50: Search activity history screen

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 |

###### Use case specification

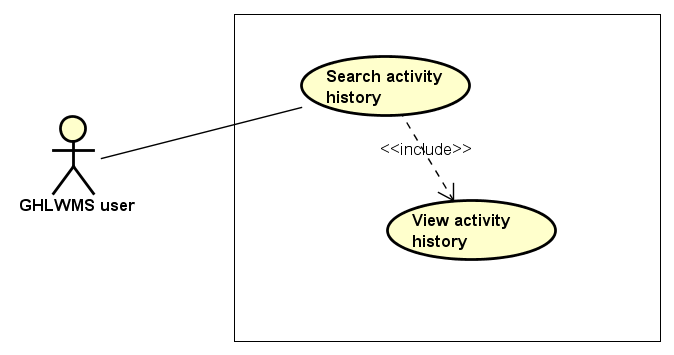


Figure 3.51: Search activity history Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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”. |

##### UC11 – Add new node

###### Screen Design

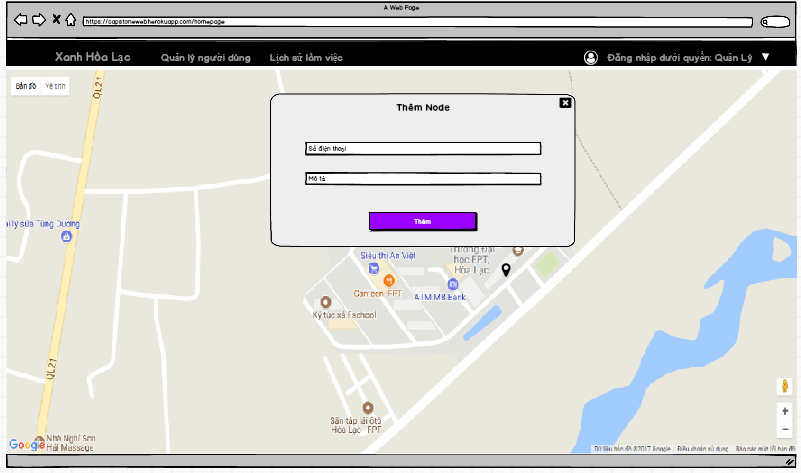


Figure 3.52: Search activity history Use case diagram

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 |  |  |  |

###### Use case specification

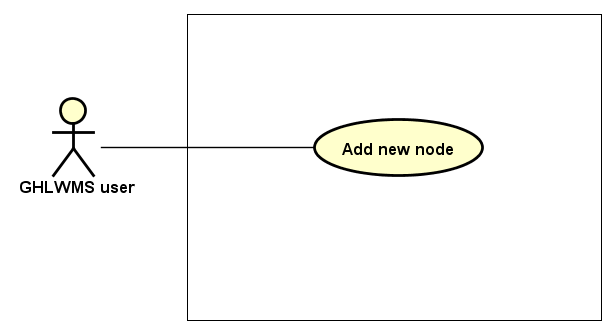


Figure 3.53: Add new node Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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. |

##### UC12 – Update node’s position

###### Screen Design

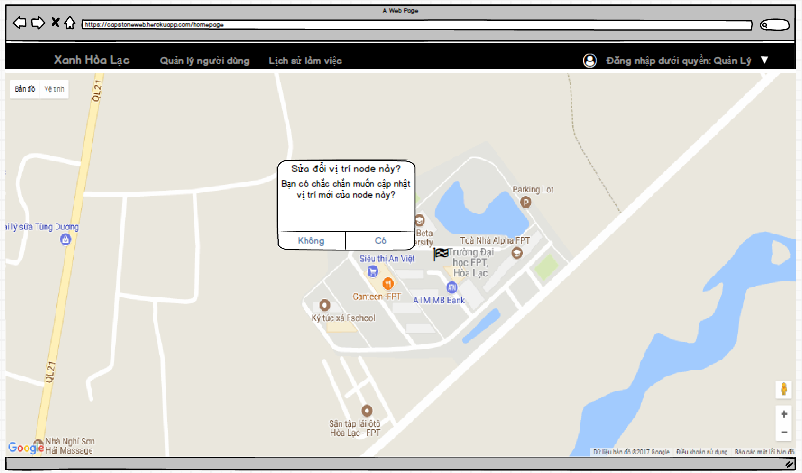


Figure 3.54: Update node's position screen

Table 3.26: Update node position Screen definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Type** | **Mandatory** | **Max Length** | **Description** |
| 1 | Update node’s position | Dialog |  |  | Confirm dialog |

###### Use case specification

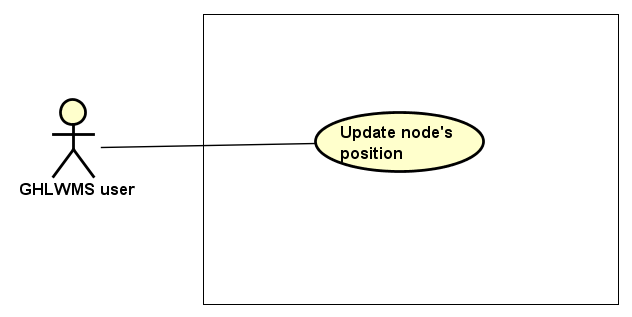


Figure 3.55: Update node's position Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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) | | | | |

##### UC13 – View node’s information

###### Screen Design

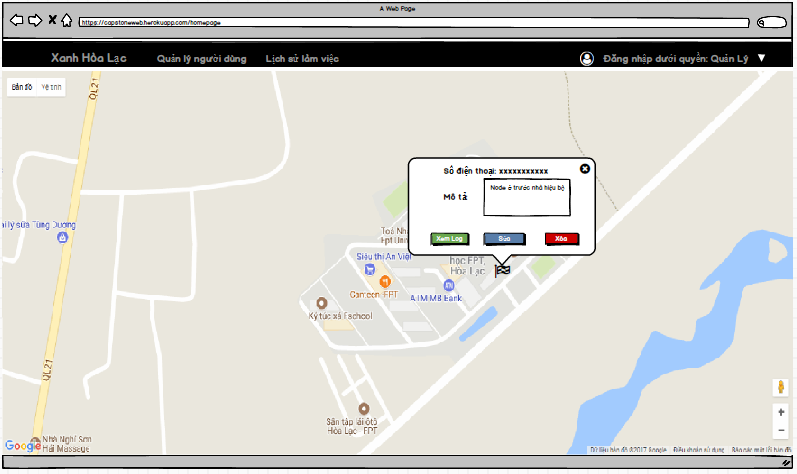


Figure 3.56: View node's information screen

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 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

###### Use case specification

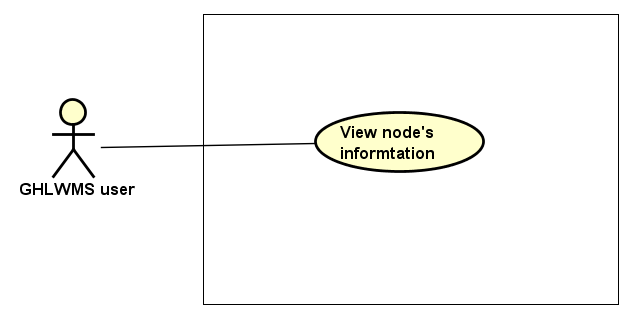


Figure 3.57: View node's information Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC14 – View node’s history

###### Screen Design

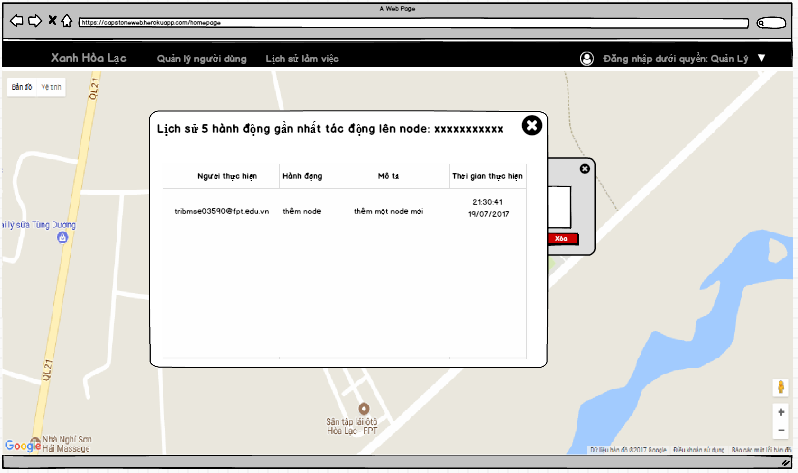


Figure 3.58: View node's history screen

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 |
|  |  |  |  |  |  |

###### Use case specification

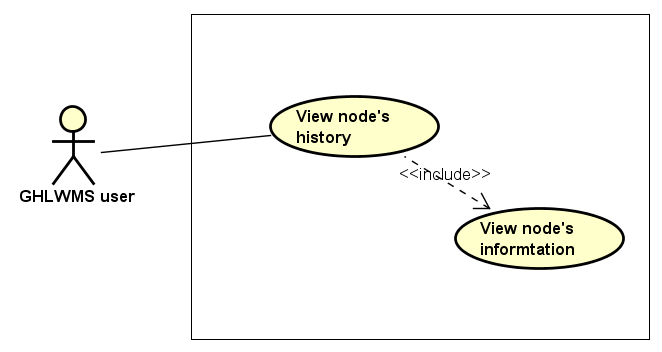


Figure 3.59: View node's history Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

##### UC15 – Update node’s information

###### Screen Design

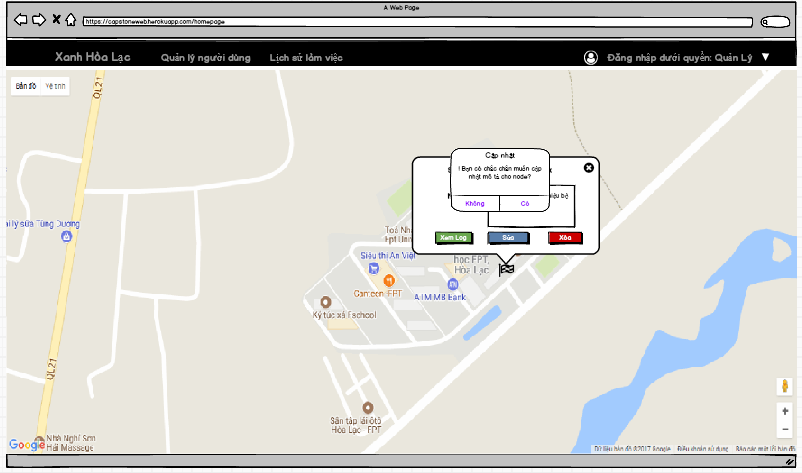


Figure 3.60: Update node's information screen

Table 3.29: Update node's information Screen definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Type** | **Mandatory** | **Max Length** | **Description** |
| 1 | Update node’s information | Dialog |  |  | Confirm dialog |

###### Use case specification

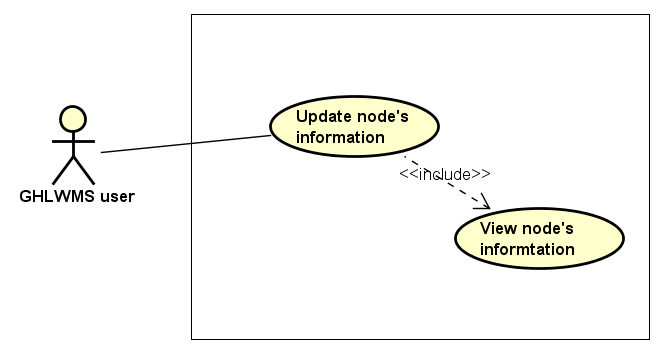


Figure 3.61: Update node's information Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

##### UC16 – Delete node

###### 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 |

###### Use case specification

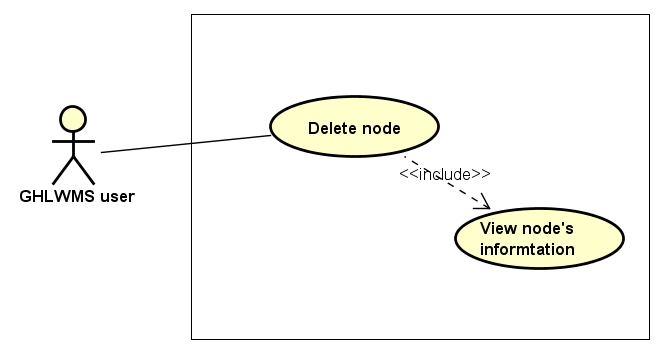


Figure 3.63: Delete node Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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” |

##### UC17 – Sort activity history

###### Screen Design

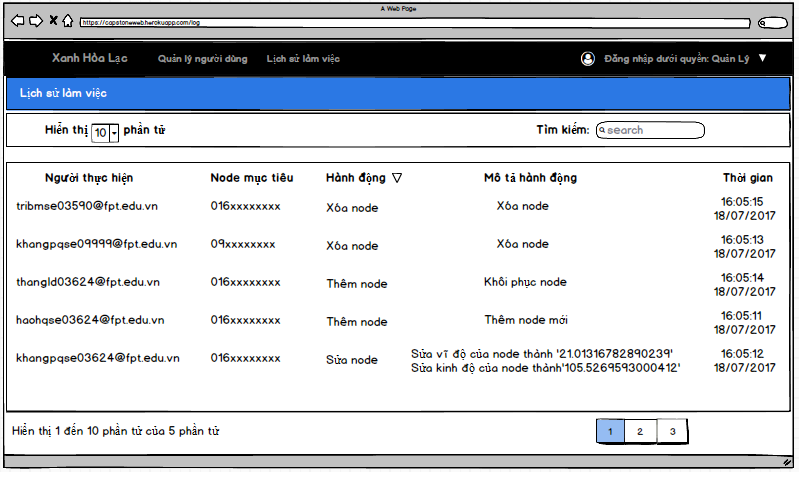


Figure 3.64: Sort activity history screen

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 |

###### Use case specification

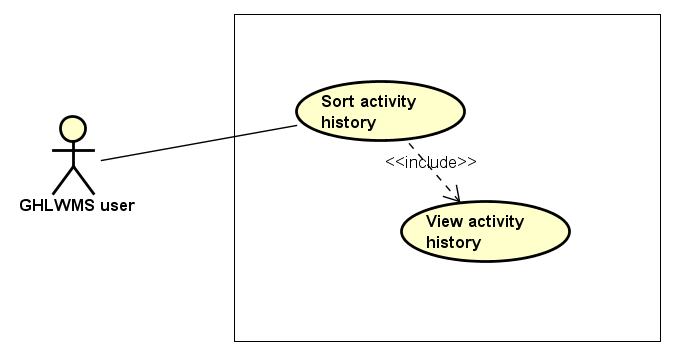


Figure 3.65: Sort activity history Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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. | | | | |

##### UC18 – Logout

###### Screen Design

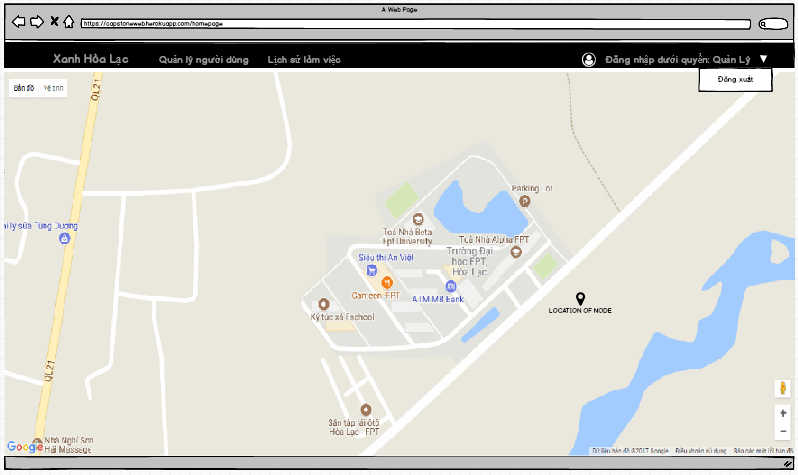


Figure 3.66: Logout screen

Table 3.32: Logout Screen definition

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

###### Use case specification

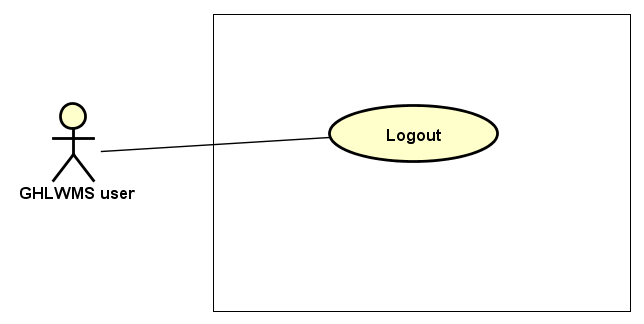


Figure 3.67: Logout Use case diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | | | | |

#### 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

### Architecture design

#### System architecture design

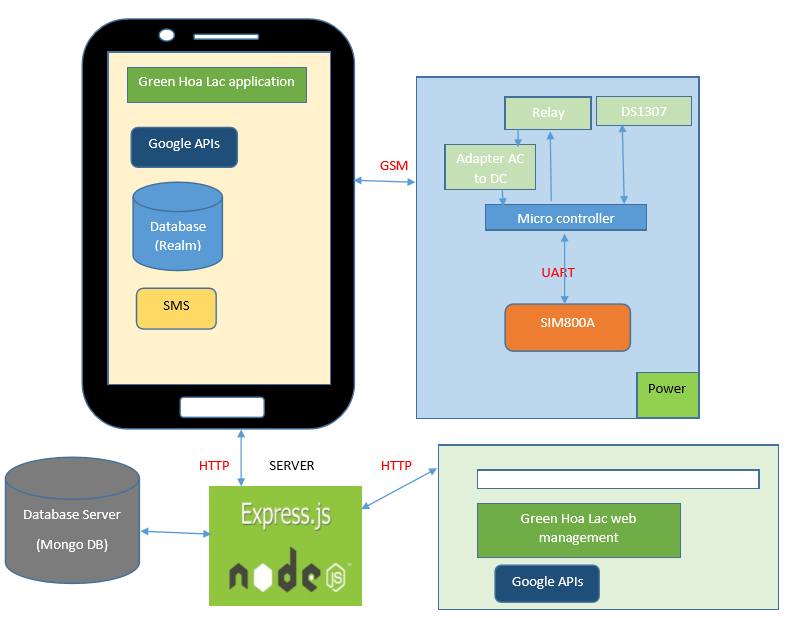


Figure 4.1: Architecture design

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.

#### Serial port (UART) communication protocol

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

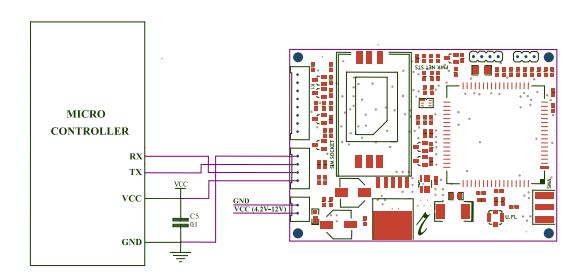


Figure 4.2: Serial port (UART) communication protocol

#### 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).

### Mechanical design

#### Design box

##### Overall

|  |  |
| --- | --- |
| Figure 4.3: Real box | Figure 4.4: Box 3D |

##### Box with all parts

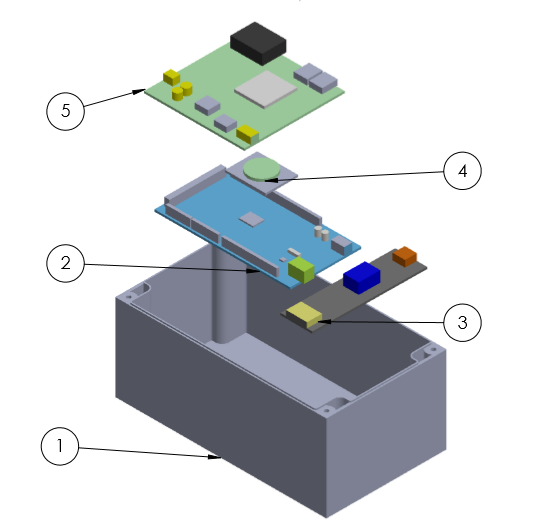


Figure 4.5: Box 3D with all parts



Figure 4.6: Note

##### Design box

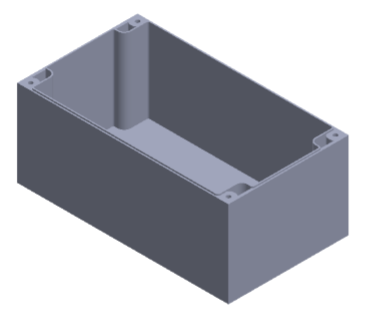


Figure 4.7: Box

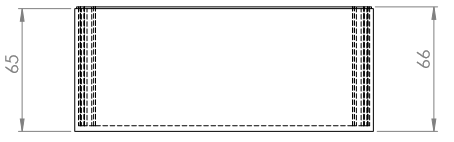


Figure 4.8: The vertical projection

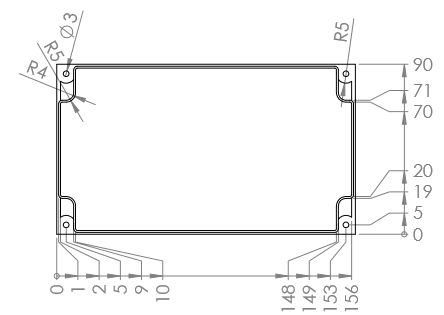


Figure 4.9: Top view

#### Schematic

##### All parts

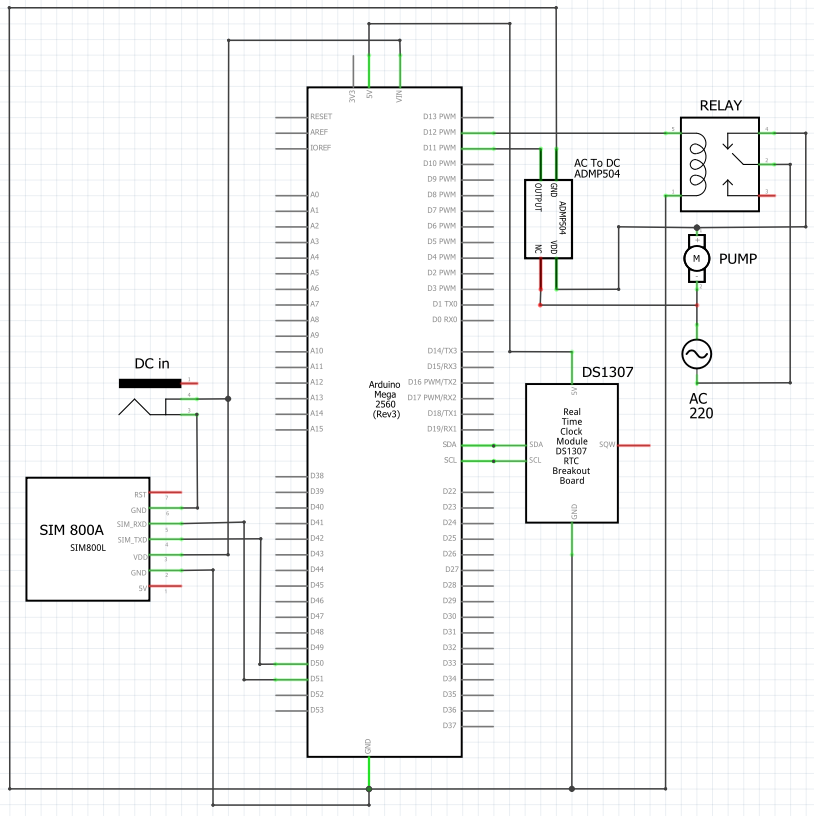


Figure 4.10: Schematic

##### Detail

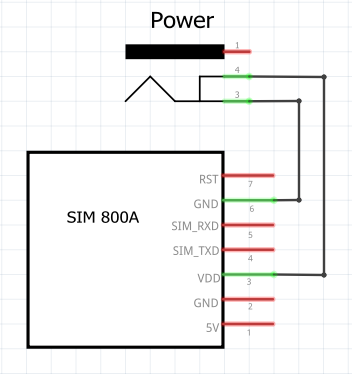


Figure 4.11: SIM and Power

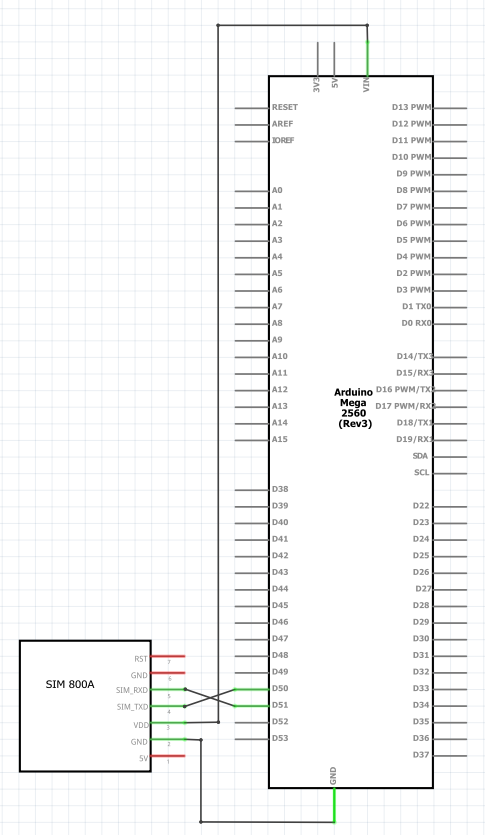


Figure 4.12: SIM and Arduino

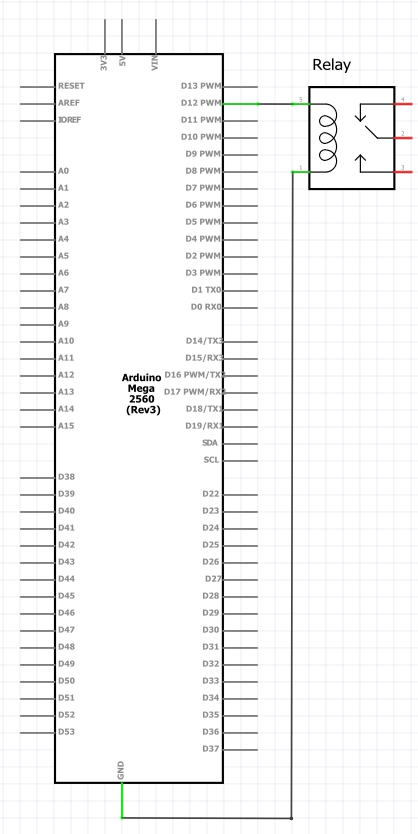


Figure 4.13: Arduino and Relay

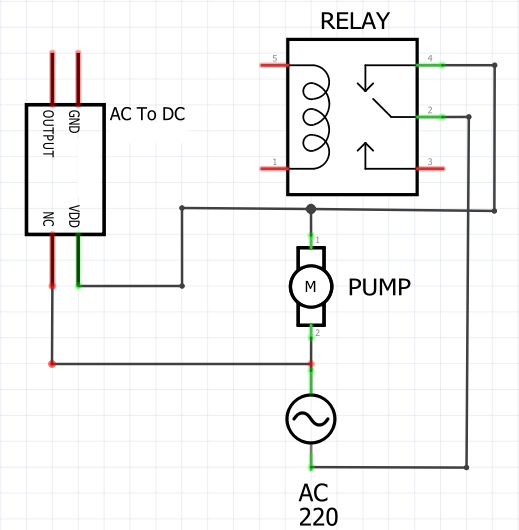


Figure 4.14: Relay, pump and adapter

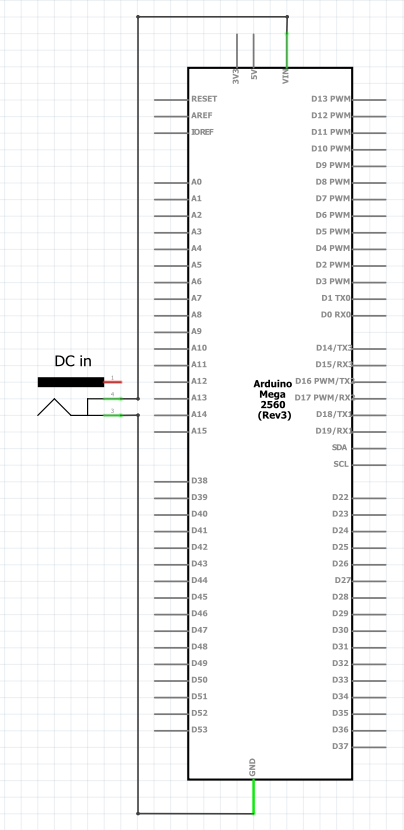


Figure 4.15: Power and Arduino

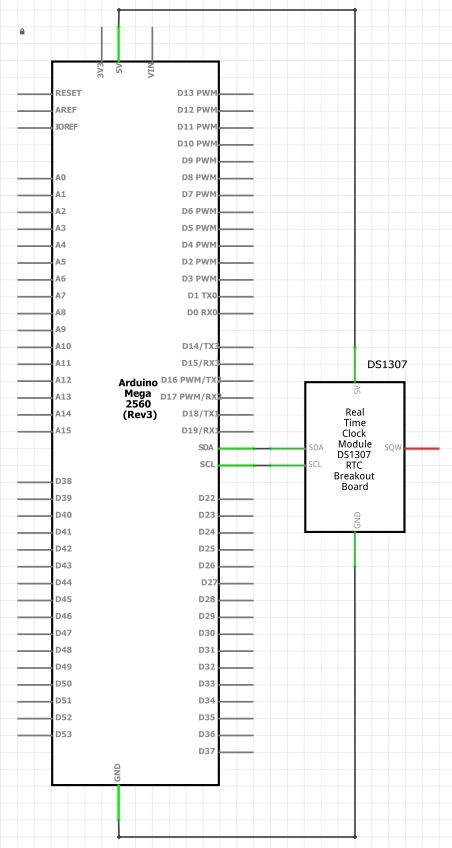


Figure 4.16: Arduino and DS1307

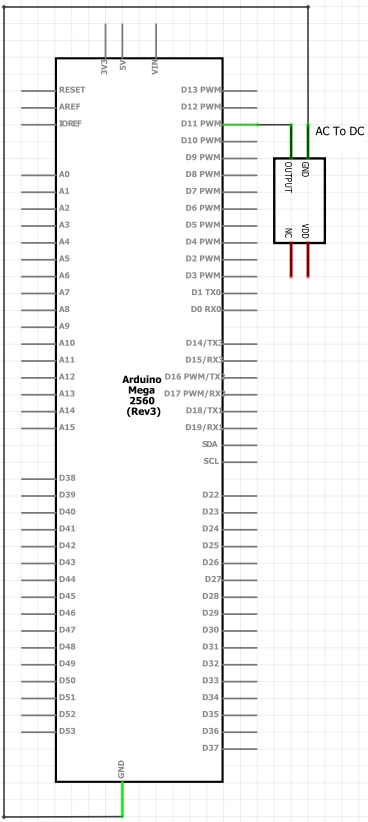
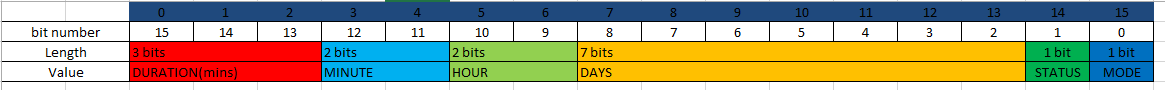


Figure 4.17: Arduino and Adapter

### Firmware design

#### 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



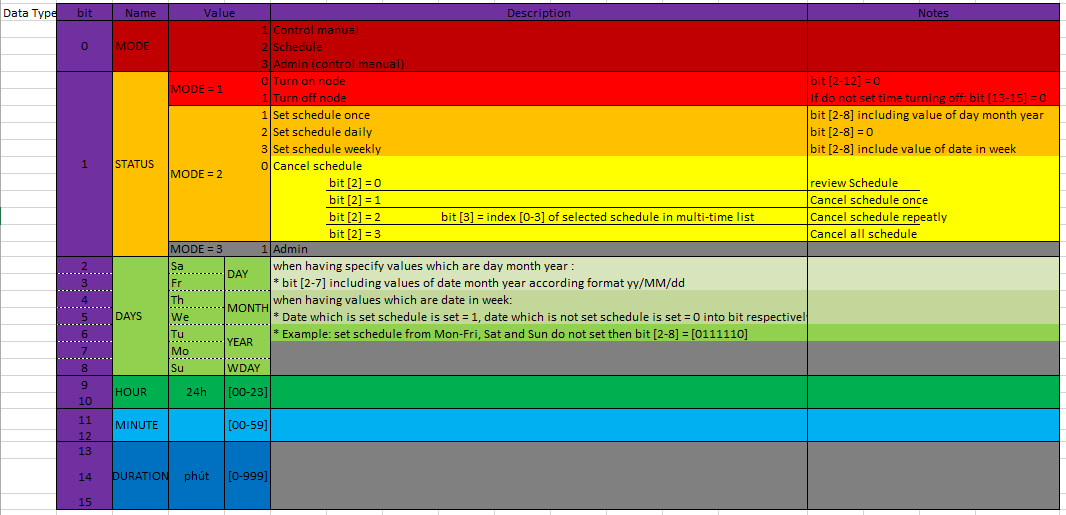


Figure 4.18: SMS syntax

#### Some main flows

##### Get on

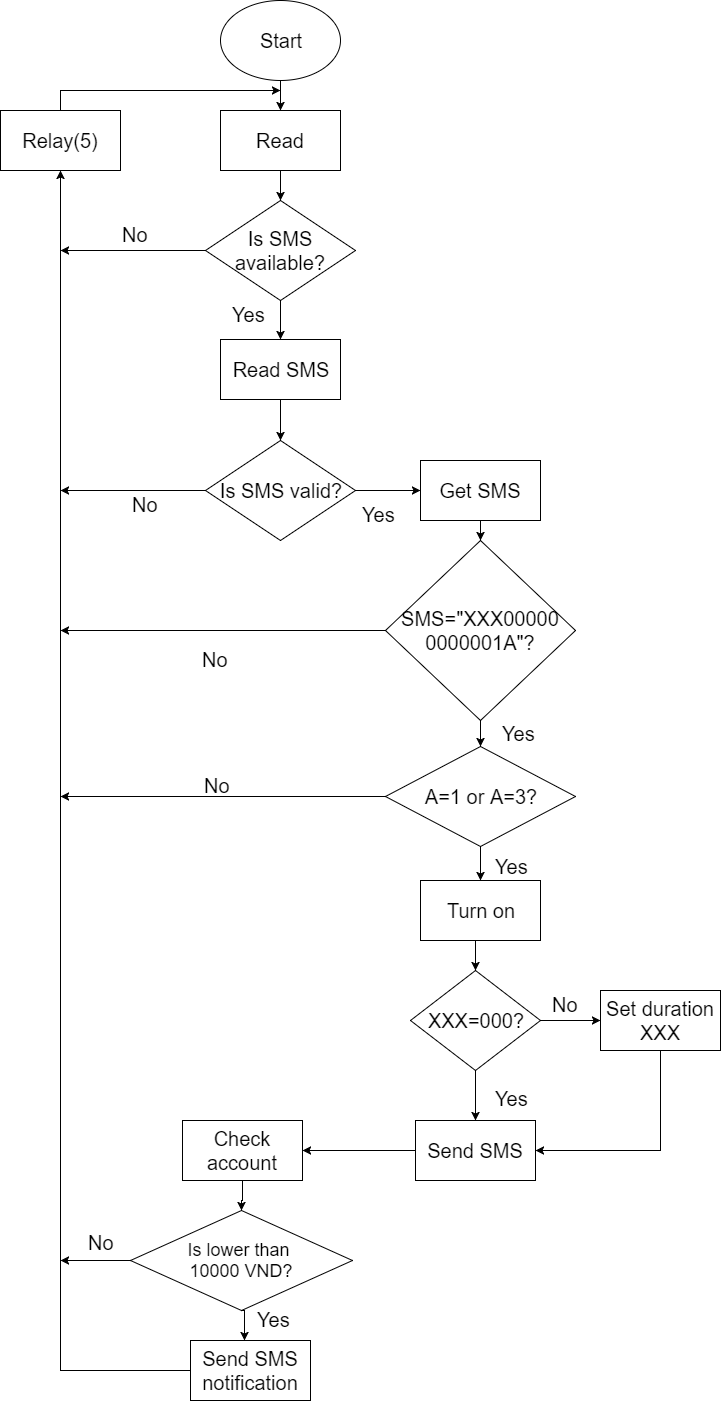


Figure 4.19: Get on

##### Get off

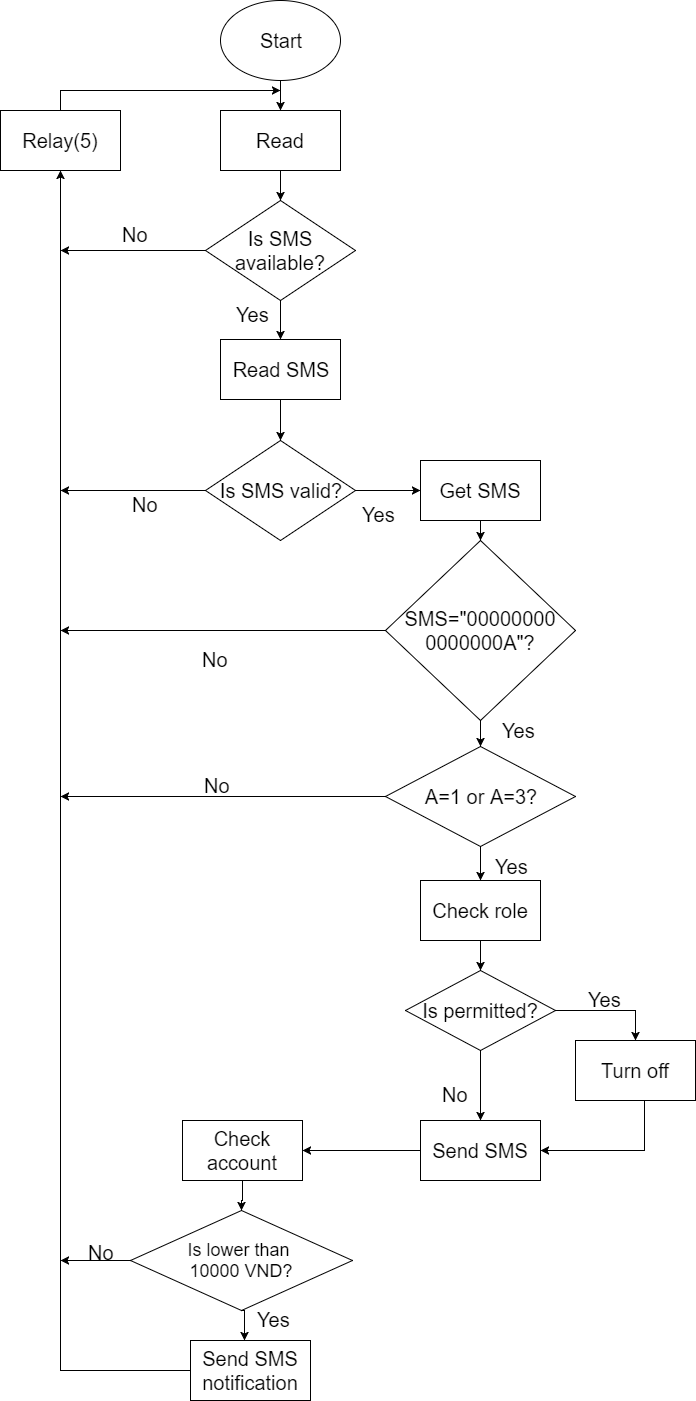


Figure 4.20: Get off

##### Set schedule

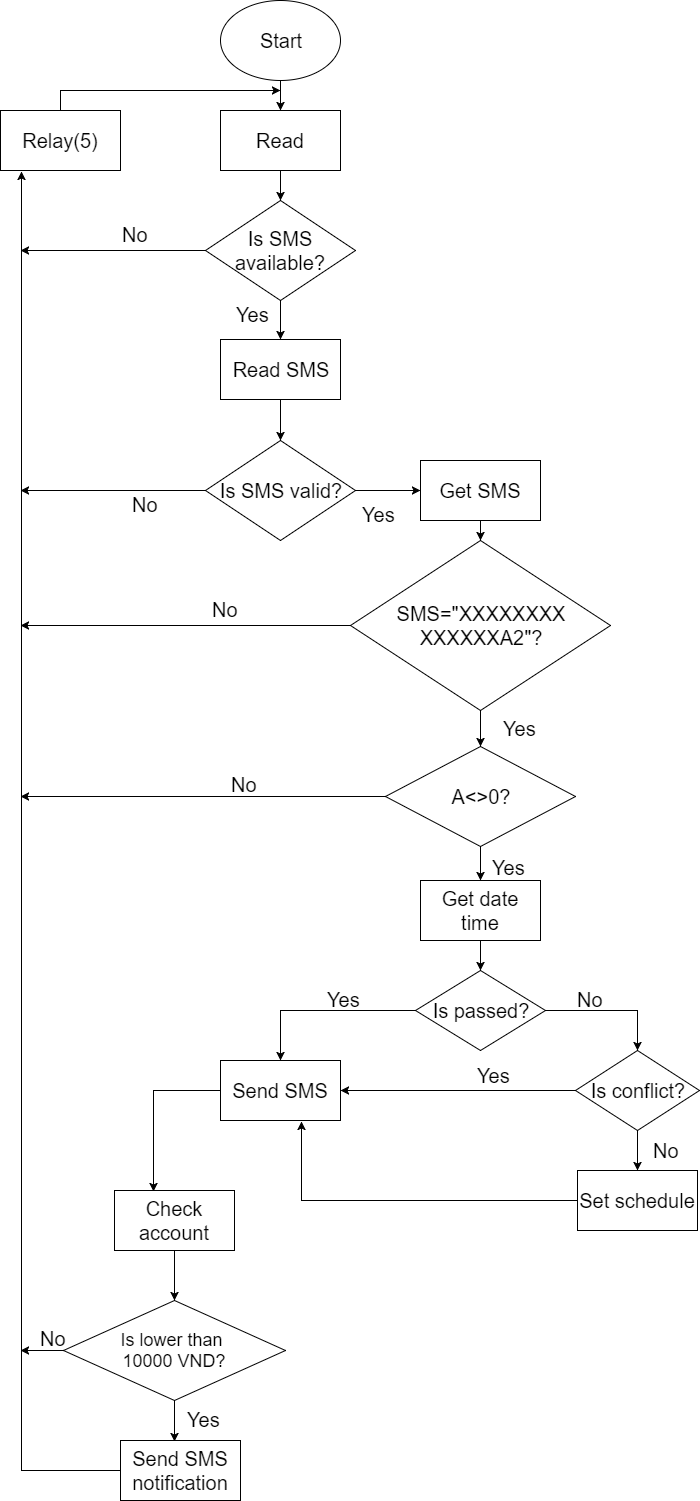


Figure 4.21: Set schedule

##### Get schedule

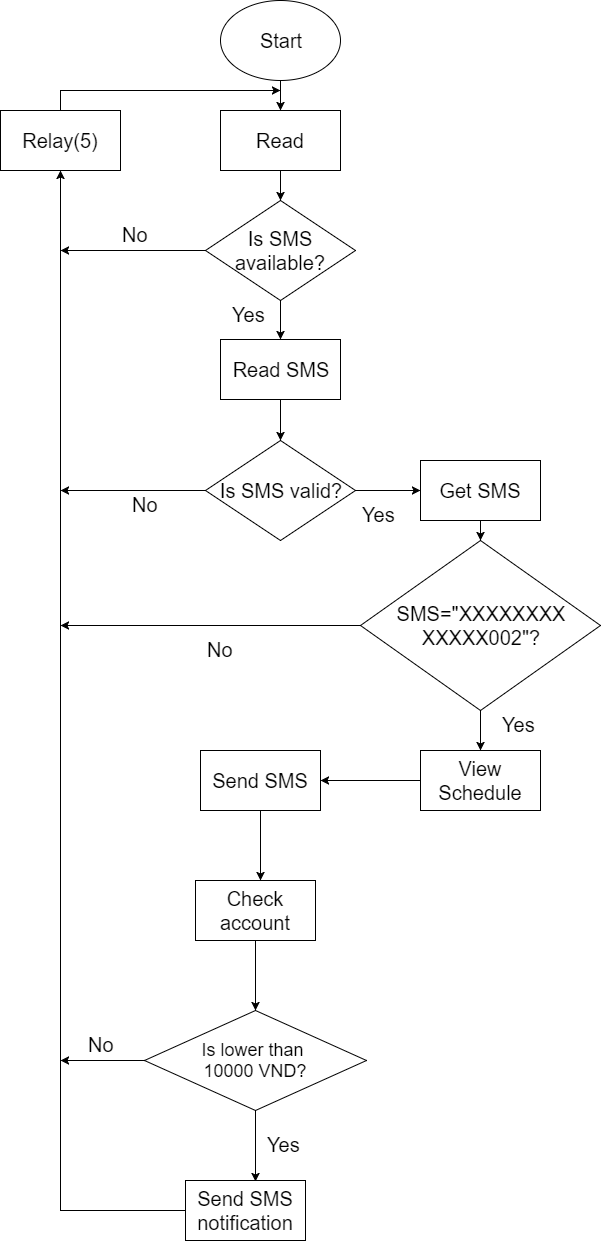


Figure 4.22: Get schedule

##### Cancel schedule

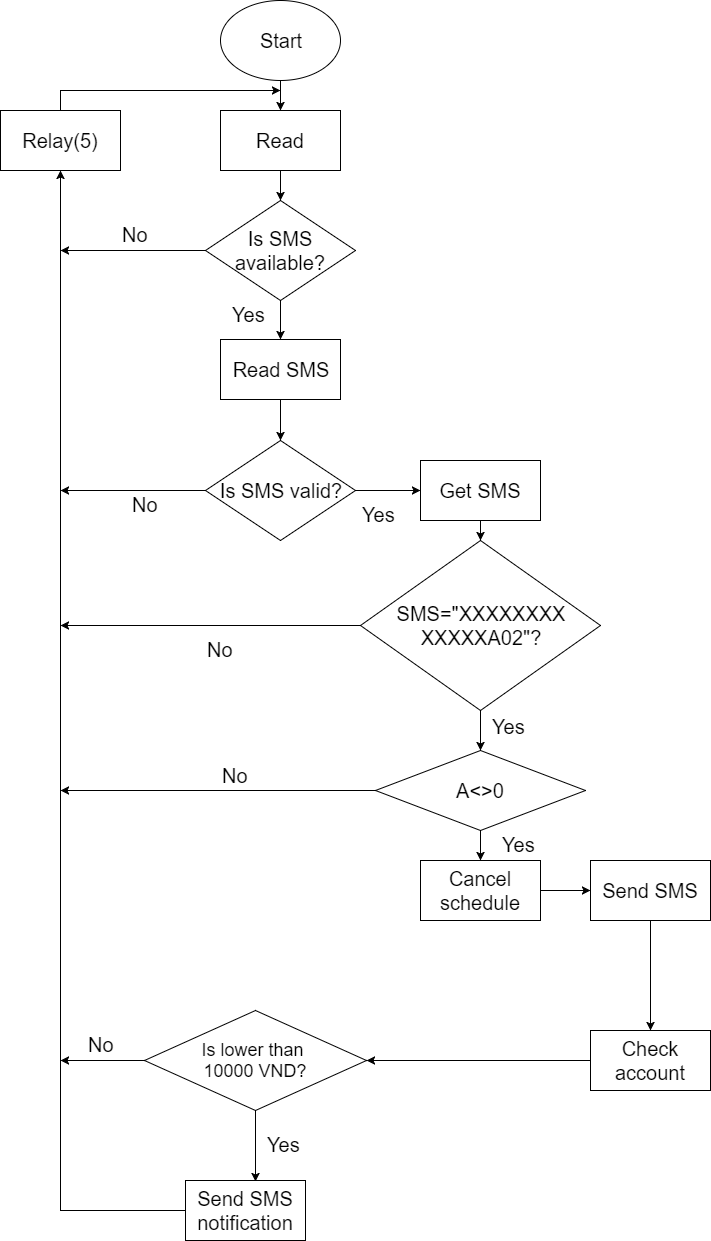


Figure 4.23: Cancel schedule

### Software design

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

#### Login

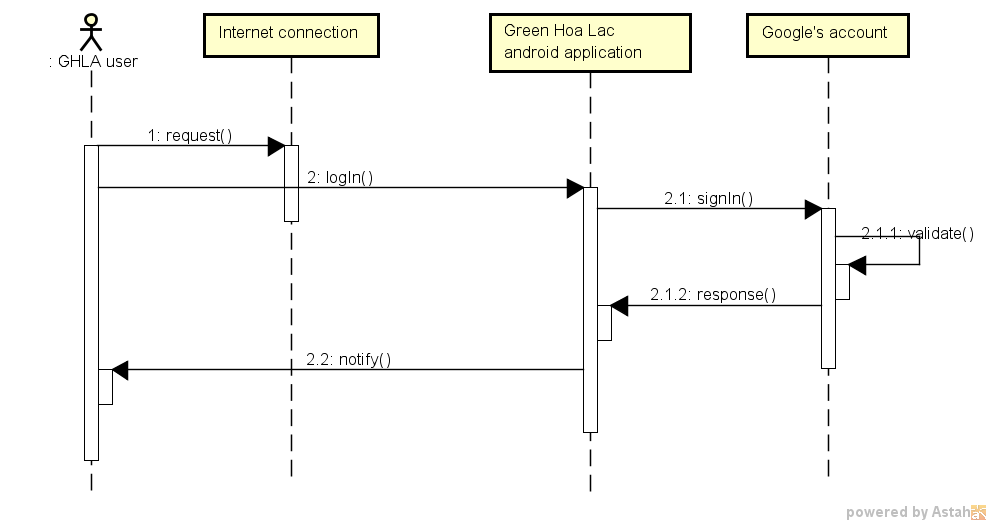


Figure 4.24: Login sequence diagram

#### Turn on

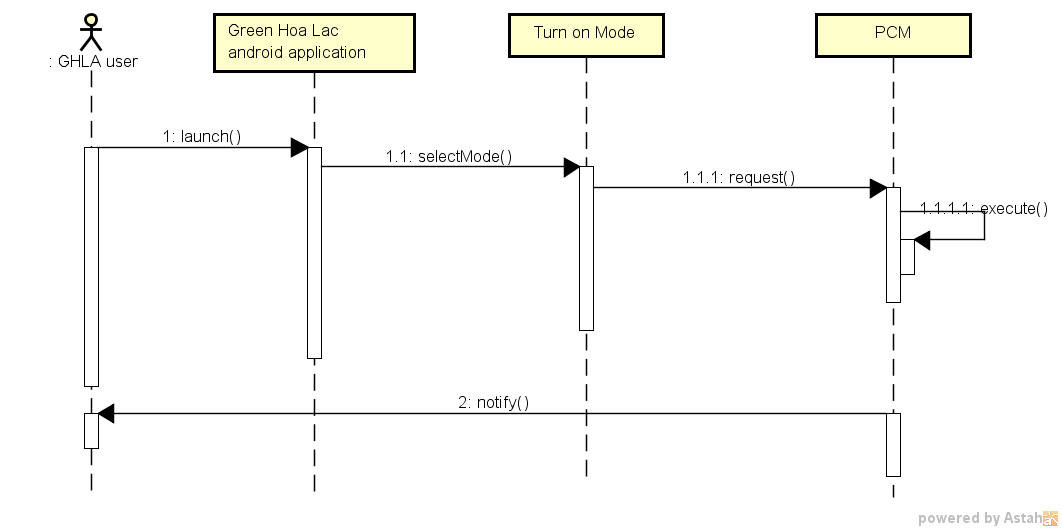


Figure 4.25: Turn on node sequence diagram

#### Turn off

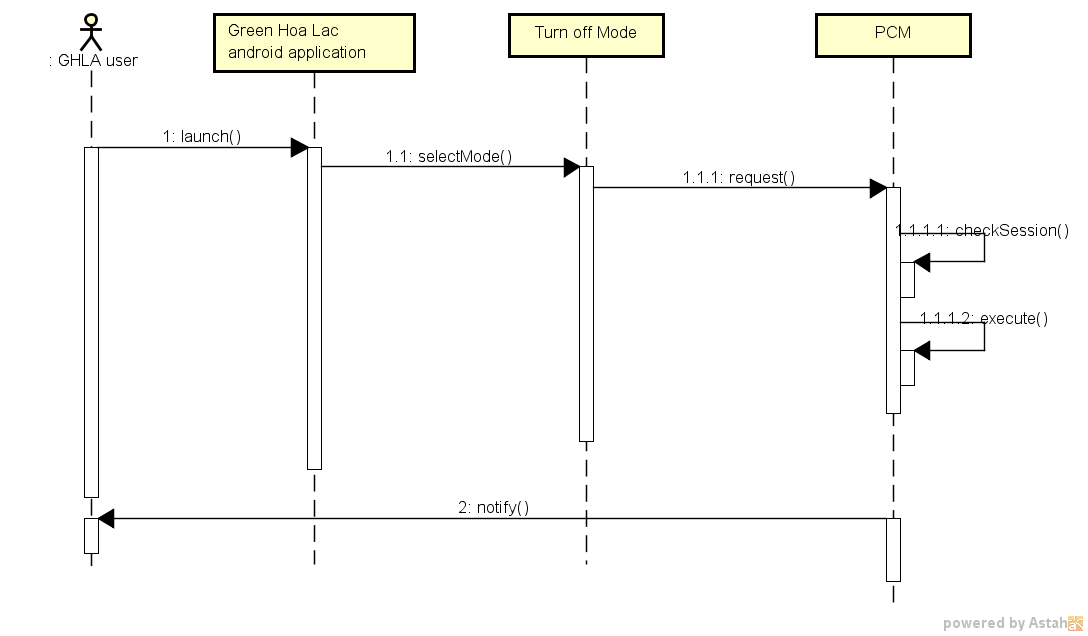


Figure 4.26: Turn off node sequence diagram

#### Set schedule

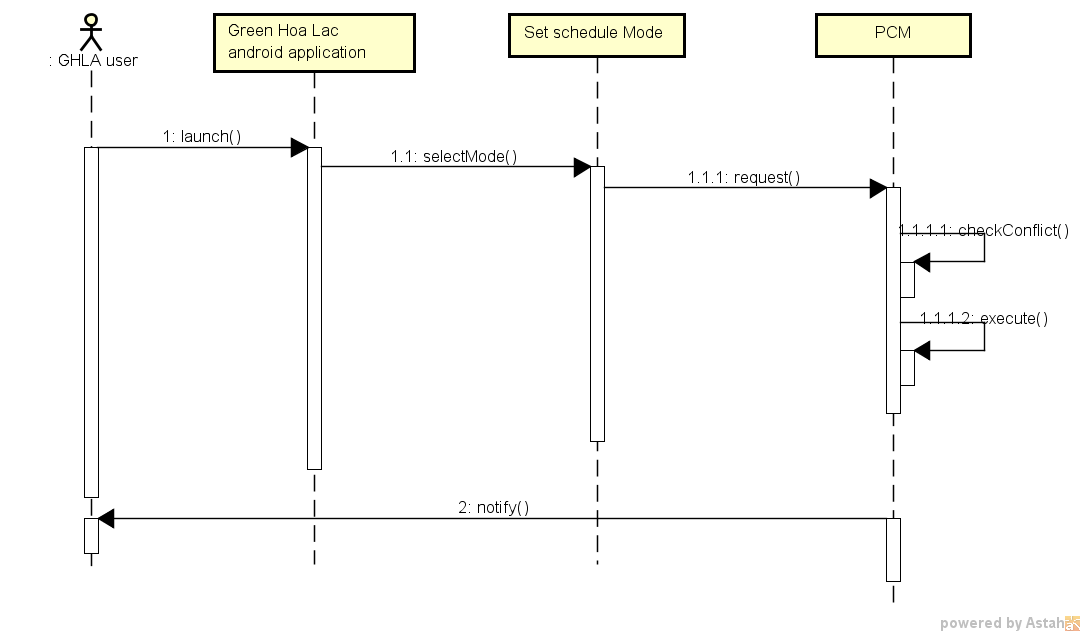


Figure 4.27: Set schedule sequence diagram

#### Cancel Schedule

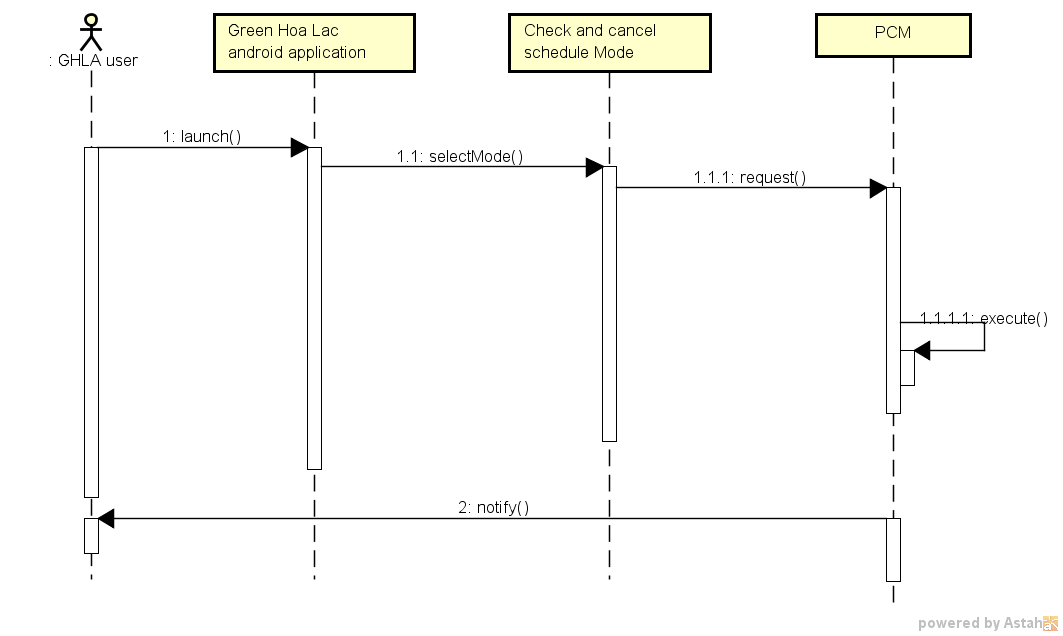


Figure 4.28: Cancel Schedule

#### Synchronize node

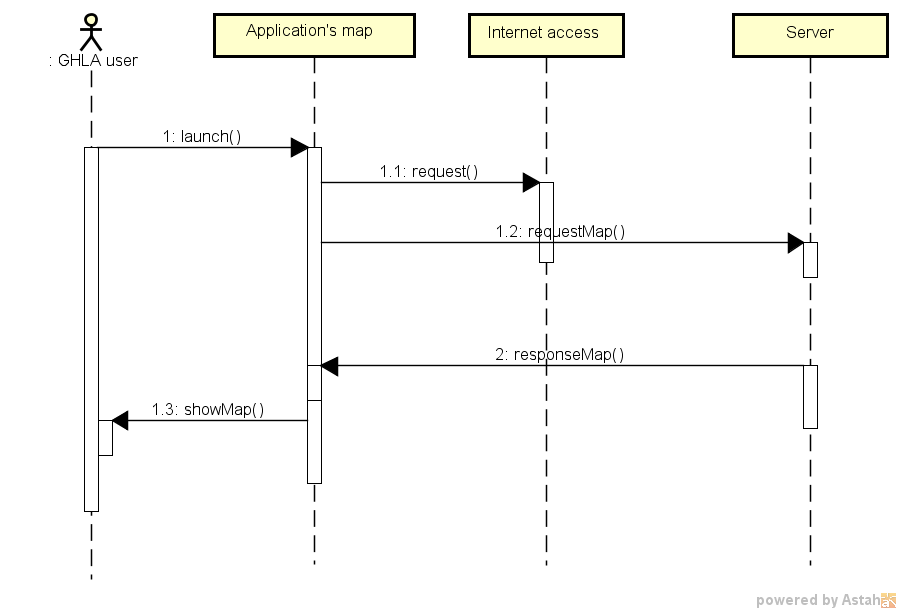


Figure 4.29: Synchronize node sequence diagram

### Database design

#### 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.

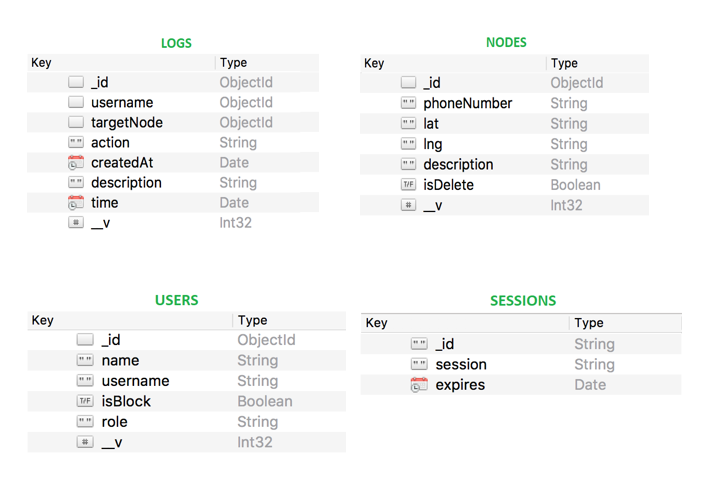


Figure 4.30: Server database (MongoDB)

In above design, we have:

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

#### 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

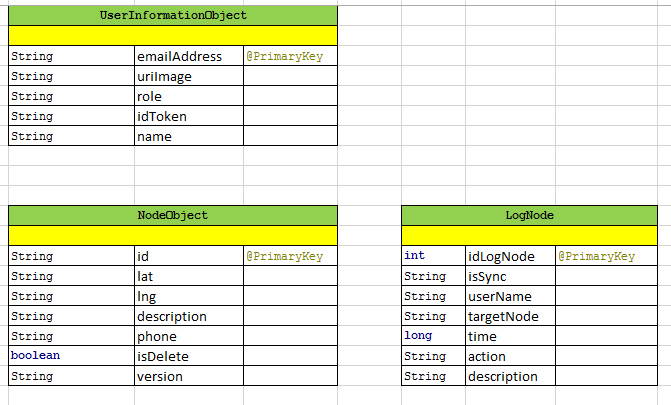


Figure 4.31: Realm database design

## Chapter 5: Implement and testing

### Implement

#### Version 1.0

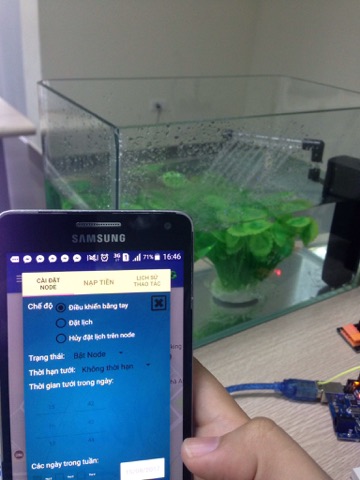
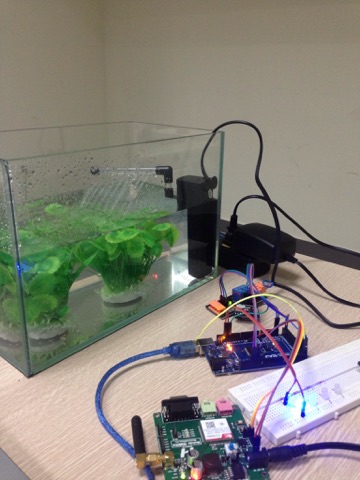
We have the first version of GHLS with features like:

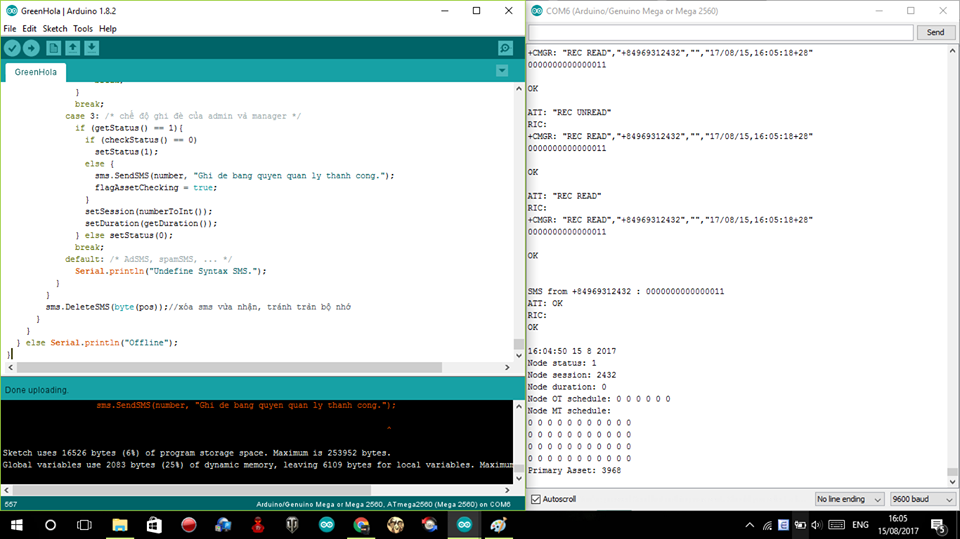
* Turn On/Of Node
* Set schedules
* Cancel schedules
* Notify account

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.

#### Version 1.1

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

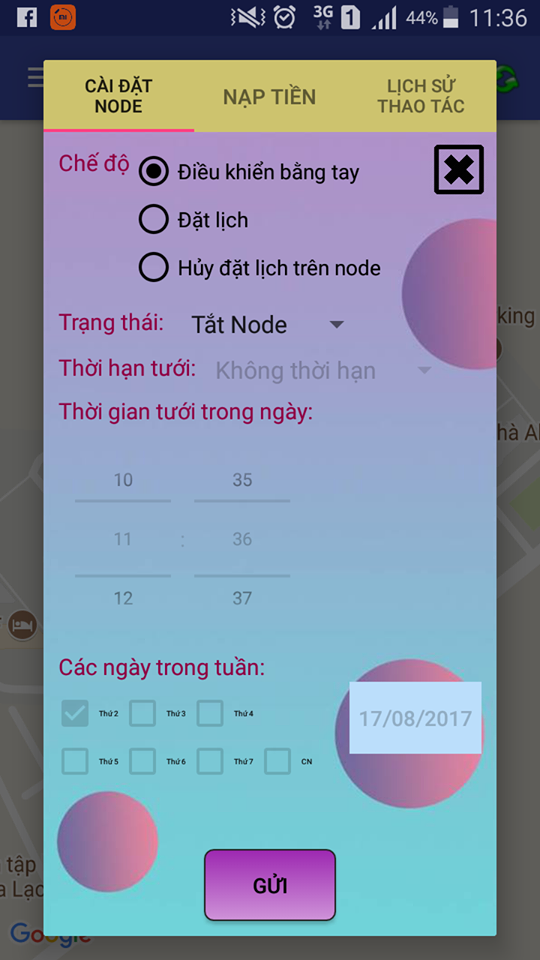
* + Turn On/Of Node
  + Set schedules
  + Cancel schedules
  + Notify account

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.

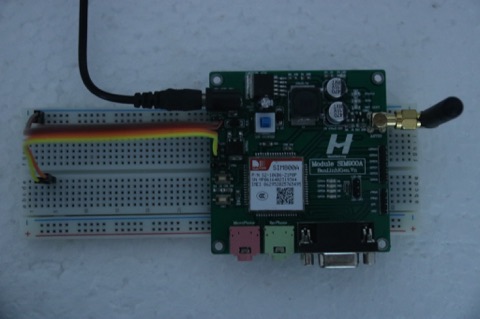




### Evidences

#### Test Module SIM800A power

##### Image

##### Expected results

Module SIM does not work

##### Actual results

Module SIM lacked power

##### Cause

Power supply cord is loose

##### Solution

Plugged in cord again

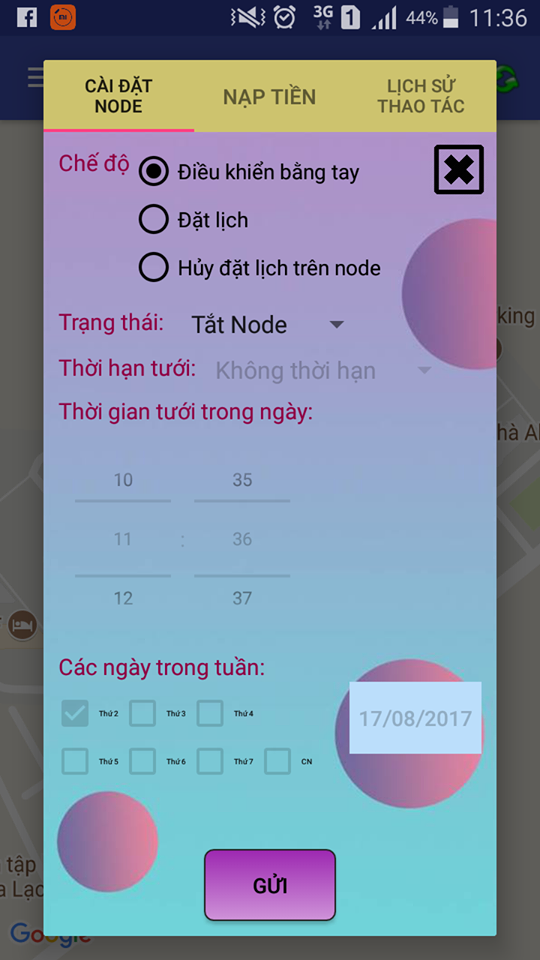
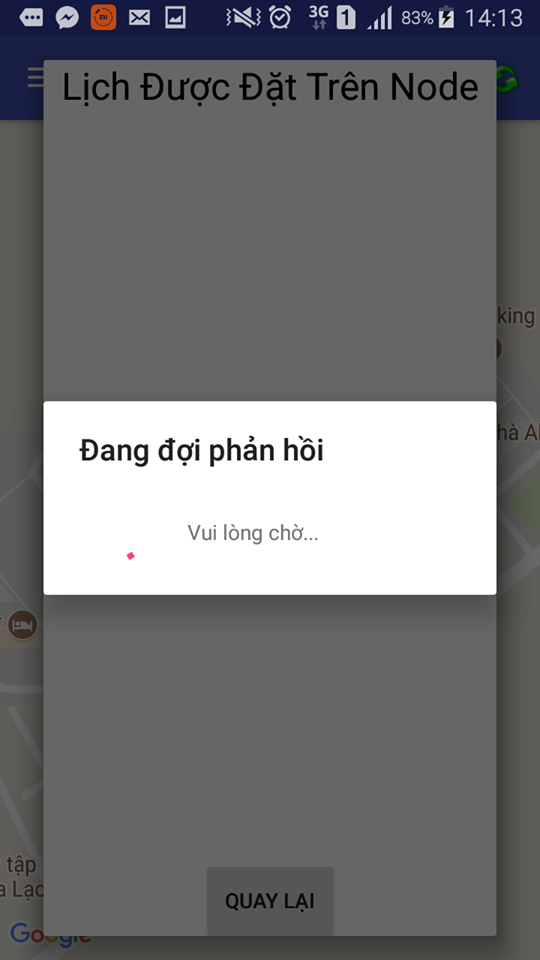
#### Can’t turn OFF by admin account

##### Step details

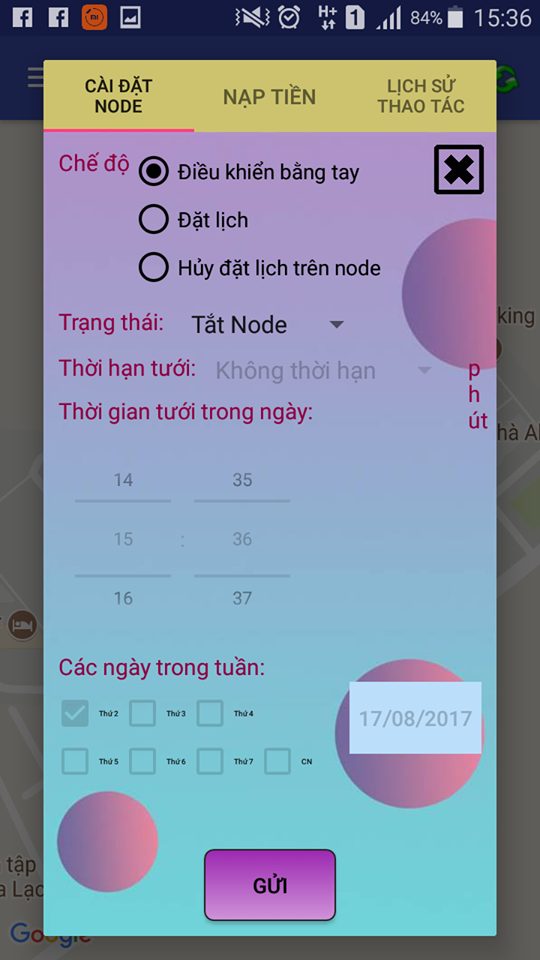
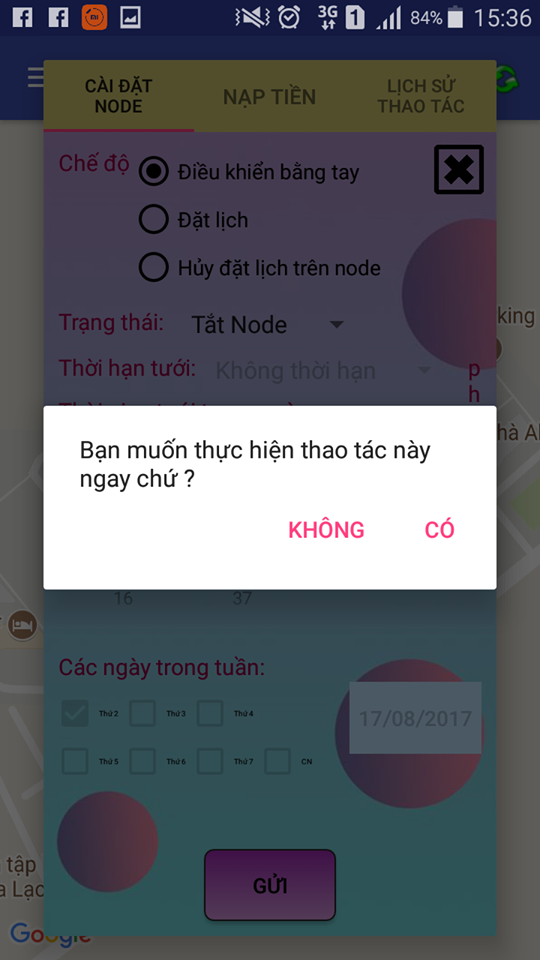
* Choose mode for Node.(Manual control)
* Choose status mode is ON.
* Set time working of Node.
* Click button Send to turn ON Node.
* After about 2-3 minutes, choose status mode is OFF.
* Click button Send to turn OFF Node.

##### Images

Get bug:

Fix bug:

##### Expected Results

“Cancel Schedule” doesn’t appear.

##### Actual Results

“Cancel Schedule” appeared.

##### 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.

##### Solution

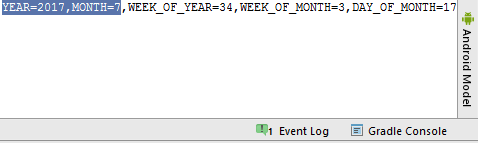
Change mode of admin account different with mode of cancel schedule.

#### Set timer in future is not working.

##### 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.

##### Images

Fix Bug:



##### Expected Results

Node must turn ON at start time moment.

##### Actual Results

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

##### Cause

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

##### 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]. |