|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

|  |
| --- |
| Capstone Project Document |
| Project title: Pet Home |
|  |
| |  |  | | --- | --- | | **We Team** | | | **Supervisor** | **Lê Phương Chi** | | **Group Members** | **Ngụy Văn Thắng - SE04385**  **Hoàng Phan Trung - SE04281**  **Tạ Đình Chiến - SE04142**  **Lâm Ngọc Khánh - SE04056** | | **Capstone Project code** | **PH** | |

# Table of Contents

[Table of Contents 2](#_Toc533460850)

[Table of Figures 5](#_Toc533460851)

[List of Images 6](#_Toc533460852)

[List of Tables 8](#_Toc533460853)

[Acknowledgements 9](#_Toc533460854)

[Definitions and Acronyms 10](#_Toc533460855)

[Chapter 1: Introduction 11](#_Toc533460856)

[1.1 Purpose 11](#_Toc533460857)

[1.2 Project information 11](#_Toc533460858)

[1.3 The people 11](#_Toc533460859)

[1.4 Background 11](#_Toc533460860)

[1.5 The problem 11](#_Toc533460861)

[1.6 Literature view 12](#_Toc533460862)

[1.7 The proposal system 13](#_Toc533460863)

[1.7.1 Our ideas 13](#_Toc533460864)

[1.7.2 The proposal 13](#_Toc533460865)

[1.7.3 System group function 14](#_Toc533460866)

[Chapter 2: Project manager plan 15](#_Toc533460867)

[2.1 Purpose 15](#_Toc533460871)

[2.2 Software process model 15](#_Toc533460872)

[2.2.1 About the iterative model 15](#_Toc533460873)

[2.2.2 Advantage and disadvantage considering 16](#_Toc533460874)

[2.3 Organization structure 16](#_Toc533460875)

[2.4 Role and responsibilities 16](#_Toc533460876)

[2.5 Tool and technique 17](#_Toc533460877)

[2.6 Project manager plan 17](#_Toc533460878)

[2.6.1 Project Schedule 17](#_Toc533460887)

[2.6.2 Meeting minutes 18](#_Toc533460897)

[2.6.3 Risk Management plan 19](#_Toc533460898)

[2.6.4 Communication plan 20](#_Toc533460899)

[2.6.5 Coding convention 21](#_Toc533460900)

[Chapter 3: Software requirement specification 22](#_Toc533460901)

[3.1 Purpose 22](#_Toc533460906)

[3.2 Definition, acronyms, and abbreviations 22](#_Toc533460907)

[3.3 User Requirement Specification 22](#_Toc533460908)

[3.4 System Requirement Specification (Specific Requirements) 22](#_Toc533460909)

[3.4.1 External interface requirements 22](#_Toc533460924)

[3.4.2 Functional requirements 23](#_Toc533460925)

[3.4.3 Non-Functional requirements 103](#_Toc533460926)

[3.5 Entity relationship diagram 104](#_Toc533460927)

[Chapter 4: Software architecture design 105](#_Toc533460928)

[4.1 Purpose 105](#_Toc533460934)

[4.2 System architecture design 105](#_Toc533460935)

[4.2.1 Choice of system architecture design 105](#_Toc533460936)

[4.2.2 Discussion of alternative design 107](#_Toc533460944)

[4.2.3 Description of system interfaces 109](#_Toc533460945)

[4.2.4 Architecture presentation 109](#_Toc533460946)

[4.3 Component diagram 109](#_Toc533460947)

[4.4 Detailed description diagram 110](#_Toc533460948)

[4.4.1 CRC card (class – responsibility - collaborators) 110](#_Toc533460957)

[4.4.2 Component 117](#_Toc533460958)

[4.5 Sequence diagram 132](#_Toc533460959)

[4.6 User interface design 143](#_Toc533460960)

[4.7 Database design 156](#_Toc533460961)

[Chapter 5: Software testing document 163](#_Toc533460962)

[5.1 Purpose 163](#_Toc533460974)

[5.2 Scope of testing 163](#_Toc533460975)

[5.2.1 Stages of testing 163](#_Toc533460983)

[5.2.2 Types of testing 163](#_Toc533460984)

[5.2.3 Range of testing 163](#_Toc533460985)

[5.3 Test plan 164](#_Toc533460986)

[5.3.1 Testing tool and environment 164](#_Toc533460995)

[5.3.2 Resource and responsibilities 165](#_Toc533461002)

[5.3.3 Test strategy 165](#_Toc533461003)

[5.3.4 Feature to be tested 168](#_Toc533461006)

[5.4 Test approach 168](#_Toc533461007)

[5.4.1 Unit testing with Postman 168](#_Toc533461017)

[5.4.2 Integration and system test 168](#_Toc533461018)

[5.5 Test report 169](#_Toc533461019)

[5.5.1 Unit tests 169](#_Toc533461021)

[5.5.2 Integration tests 169](#_Toc533461033)

[5.5.3 System tests 184](#_Toc533461035)

[Chapter 6: Software user guide 197](#_Toc533461036)

[6.1 Deployment guidelines 197](#_Toc533461044)

[6.1.1 Environment for user 197](#_Toc533461052)

[6.1.2 Deploy PH web application and web service for system admin 197](#_Toc533461053)

[6.2 User guide 199](#_Toc533461054)

[6.1.3 Location manager 199](#_Toc533461055)

[6.1.4 User 200](#_Toc533461056)

[References 206](#_Toc533461057)

# Table of Figures

[Figure 1: Pethome.vn 12](#_Toc533461058)

[Figure 2: “Đảo mèo” 12](#_Toc533461059)

[Figure 3: “Chủng tộc đại ngáo husky” 13](#_Toc533461060)

[Figure 4: Role in PH system 14](#_Toc533461061)

[Figure 5: Iterative process model development 15](#_Toc533461062)

[Figure 6: Organization Structure 16](#_Toc533461063)

[Figure 7: Tasks list 18](#_Toc533461064)

[Figure 8: Meeting minutes example 19](#_Toc533461065)

[Figure 9: Entity Relationship Diagram 104](#_Toc533461066)

[Figure 10: System Architectural Design 105](#_Toc533461067)

[Figure 11:RESTful web service architecture 105](#_Toc533461068)

[Figure 12: JSON return when call RESTful web service 107](#_Toc533461069)

[Figure 13:MVC architecture 108](#_Toc533461070)

[Figure 14:Architecture presentation 109](#_Toc533461071)

[Figure 14: Component Diagram 110](#_Toc533461072)

[Figure 15: Web login use case class diagram 118](#_Toc533461073)

[Figure 16:View user list use case class diagram 120](#_Toc533461074)

[Figure 17:View location list use case class diagram 121](#_Toc533461075)

[Figure 18:View location information use case class diagram 123](#_Toc533461076)

[Figure 19:Get post category use case class diagram 124](#_Toc533461077)

[Figure 20:Locaiton manager register use case class diagram 126](#_Toc533461078)

[Figure 21: Web forgot password use case class diagram 128](#_Toc533461079)

[Figure 22: Create a pet use case class diagram 129](#_Toc533461080)

[Figure 23: Create a post use case class diagram 131](#_Toc533461081)

[Figure 24:V-Model 166](#_Toc533461082)

# List of Images

[Image 1: Administrator use case diagram 24](#_Toc533461163)

[Image 2: Manage user function use case diagram 25](#_Toc533461164)

[Image 3: Manage location function use case diagram 29](#_Toc533461165)

[Image 4:Administrator authentication function use case diagram 37](#_Toc533461166)

[Image 5:Administrator manage category function use case diagram 39](#_Toc533461167)

[Image 6: Administrator manage report function use case diagram 43](#_Toc533461168)

[Image 7: Location Manager use case diagram 45](#_Toc533461169)

[Image 8: Location manager authentication function use case diagram 46](#_Toc533461170)

[Image 9: Manager location manage location function use case diagram 50](#_Toc533461171)

[Image 10: Location manager profile function use case diagram 55](#_Toc533461172)

[Image 11: Location manager manage category function use case diagram 57](#_Toc533461173)

[Image 12: User and Guest usecase diagram 63](#_Toc533461174)

[Image 13: User profile function use case diagram 66](#_Toc533461175)

[Image 14:User authentication function use case diagram 70](#_Toc533461176)

[Image 15:User post function use case diagram 75](#_Toc533461177)

[Image 16:User comment function use case diagram 81](#_Toc533461178)

[Image 17:User rating function use case diagram 85](#_Toc533461179)

[Image 18:User chat function use case diagram 86](#_Toc533461180)

[Image 19:User notification function use case diagram 89](#_Toc533461181)

[Image 20:User authentication function use case diagram 90](#_Toc533461182)

[Image 21:User view post function use case diagram 92](#_Toc533461183)

[Image 22:User & Guest location function use case diagram 94](#_Toc533461184)

[Image 23: User pet function use case diagram 98](#_Toc533461185)

[Image 24: Web Login sequence diagram 133](#_Toc533461186)

[Image 25: Web Logout sequennce diagram 134](#_Toc533461187)

[Image 26: View User List sequence diagram 135](#_Toc533461188)

[Image 27: View Location List sequence diagram 135](#_Toc533461189)

[Image 28: VIew Location Information sequence diagram 136](#_Toc533461190)

[Image 29: View Category List sequence diagram 136](#_Toc533461191)

[Image 30: Show/Hide Category sequence diagram 137](#_Toc533461192)

[Image 31: Location Manager register sequence diagram 138](#_Toc533461193)

[Image 32: Show/Hide Location sequence diagram 139](#_Toc533461194)

[Image 33: Edit Location sequence diagram 140](#_Toc533461195)

[Image 34: Web Forgot Password sequence diagram 141](#_Toc533461196)

[Image 35: Create Pet seqence diagram 142](#_Toc533461197)

[Image 36: Create a Post sequence diagram 143](#_Toc533461198)

[Image 38:Add Product page 144](#_Toc533461199)

[Image 39: Category Manage page 144](#_Toc533461200)

[Image 40:Location manager Home page 145](#_Toc533461201)

[Image 41: Add Location page 145](#_Toc533461202)

[Image 42: Login page 146](#_Toc533461203)

[Image 43: Register page 146](#_Toc533461204)

[Image 44: Edit Product page 147](#_Toc533461205)

[Image 45: Verify Account page 147](#_Toc533461206)

[Image 46: View Profile page 148](#_Toc533461207)

[Image 47: Login display 149](#_Toc533461208)

[Image 48: New password display 149](#_Toc533461209)

[Image 49: Register display 149](#_Toc533461210)

[Image 50:Forgot password display 149](#_Toc533461211)

[Image 51:Verify forgot password display 150](#_Toc533461212)

[Image 52:Drawer 150](#_Toc533461213)

[Image 53:Home display 150](#_Toc533461214)

[Image 54:Add post display 150](#_Toc533461215)

[Image 55:Comment display 151](#_Toc533461216)

[Image 56:Message display 151](#_Toc533461217)

[Image 57: Notification 151](#_Toc533461218)

[Image 58: Search post display 151](#_Toc533461219)

[Image 59: Send message display 152](#_Toc533461220)

[Image 60: Search display 152](#_Toc533461221)

[Image 61: Location detail display 152](#_Toc533461222)

[Image 62: Map display 152](#_Toc533461223)

[Image 63: Search filter 153](#_Toc533461224)

[Image 64: Add pet display 153](#_Toc533461225)

[Image 65: Pet manager display 153](#_Toc533461226)

[Image 66: History matched display 153](#_Toc533461227)

[Image 67: Pet management display 154](#_Toc533461228)

[Image 68:Change email display 154](#_Toc533461229)

[Image 69: change password display 154](#_Toc533461230)

[Image 70: Change display name 154](#_Toc533461231)

[Image 71: Edit profile 155](#_Toc533461232)

[Image 72: View profile 155](#_Toc533461233)

[Image 73: Database design 156](#_Toc533461234)

[Image 74: Stages of testing 163](#_Toc533461235)

[Image 75: Clone project 197](#_Toc533461236)

[Image 76:Install module 198](#_Toc533461237)

[Image 77:Install module 198](#_Toc533461238)

[Image 78:Run project 199](#_Toc533461239)

[Image 79:Login page 199](#_Toc533461240)

[Image 80:Register page 200](#_Toc533461241)

[Image 81:Register page 200](#_Toc533461242)

# List of Tables

[Table 1: Definitions and Acronyms 10](#_Toc533461243)

[Table 2: Project information 11](#_Toc533461244)

[Table 3: Advantage and Disadvantage Considering 16](#_Toc533461245)

[Table 4: Role and Responsibilities 17](#_Toc533461246)

[Table 5: Tool and Technique 17](#_Toc533461247)

[Table 6: Risk Management Plan 20](#_Toc533461248)

[Table 7: Member Information 21](#_Toc533461249)

[Table 8: Project report and Meeting 21](#_Toc533461250)

[Table 9: Project Communication 21](#_Toc533461251)

[Table 10: Coding Convention 21](#_Toc533461252)

[Table 11: Definition, Acronyms, and Abbreviations 22](#_Toc533461253)

[Table 12: Business Rule Repository 23](#_Toc533461254)

[Table 13: Administrator usecase summary 25](#_Toc533461255)

[Table 14: Location Manager usecase diagram 46](#_Toc533461256)

[Table 15: Usser and Guest usecase summary 65](#_Toc533461257)

[Table 16:Testing tools 164](#_Toc533461258)

[Table 17:Testing environment 165](#_Toc533461259)

[Table 18:Resources and responsibilities 165](#_Toc533461260)

[Table 19:Test stages 167](#_Toc533461261)

[Table 20:Test schedule 167](#_Toc533461262)

[Table 21:Deliverables 168](#_Toc533461263)

[Table 22:Unit test report 169](#_Toc533461264)

[Table 23:Integration test report 169](#_Toc533461265)

[Table 24: Administrator module integration test case 173](#_Toc533461266)

[Table 25:Location manager module integration test case 177](#_Toc533461267)

[Table 26:User module integration test case 181](#_Toc533461268)

[Table 27:Guest module integration test case 184](#_Toc533461269)

[Table 28:System test report 184](#_Toc533461270)

[Table 29:Administrator module system test case 188](#_Toc533461271)

[Table 30:Location manager module system test case 192](#_Toc533461272)

[Table 31:User module system test case 196](#_Toc533461273)

[Table 32:Environment for user 197](#_Toc533461274)

# Acknowledgements

First and foremost, we would love to express our deep gratitude to our supervisor, Mrs. Le Phuong Chi. Without his assistance and dedicated involvement in every step throughout the process, this paper would have never been accomplished. We also sincerely thank to all of our teachers who had taught us in four years at FPT University.

We also grateful to our friends whom we have had the pleasure to know and work with during this semester. We could say that the capstone project could not have been fully accomplished without their support.

Nobody has been more important to us in the pursuit of this project than our family. We would love to thank our parents, our sisters and brothers who offered unlimited encouragement and support.

# Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| **No** | **Definition and Acronyms** | **Explanation** |
| 1 | PH | Pet Home |
| 2 | Guest | People who visit the PH application without registered account. |
| 3 | Location manager | People who visit the manage location web application to add and manage their added location. |
| 4 | User | People who visit the PH application with registered account. |
| 5 | Administrator | People who manage all system. |
| 6 | SRS | Software requirement specification. |
| 7 | PM | Project manager. |
| 8 | Dev | Developer. |
| 9 | BA | Business analyst. |

Table 1: Definitions and Acronyms

# Chapter 1: Introduction

## Purpose

This chapter provides an overview of the capstone project. It includes the project information, the background of the capstone project, the problem and our proposal to solve the problem.

## Project information

|  |  |
| --- | --- |
| **Project information** | |
| Project name | Pet home |
| Project code | PH |
| Group name | We Team |
| Product type | Website and mobile application |
| Timeline | 15th August 2018 – 10th December 2018 |

Table 2: Project information

## The people

**Supervisor:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Full name** | **Phone number** | **Email** | **Title** |
| Lê Pương Chi | 097 720 0369 | ChiLP2@fe.edu.vn/ChiLP@fpt.edu.vn | Supervisor |

**Members:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Full name** | **Student ID** | **Phone number** | **Email** | **Title** |
| Ngụy Văn Thắng | SE04385 | 098 457 0673 | [Thangnvse04385@fpt.edu.vn](mailto:Thangnvse04385@fpt.edu.vn) | Team leader |
| Hoàng Phan Trung | SE04281 | 034 225 6852 | [Trunghpse04281@fpt.edu.vn](mailto:Trunghpse04281@fpt.edu.vn) | Team member |
| Tạ Đình Chiến | SE04142 | 096 925 5141 | [Chientdse04142@fpt.edu.vn](mailto:Chientdse04142@fpt.edu.vn) | Team member |
| Lâm Ngọc Khánh | SE04056 | 096 591 7064 | [Khanhlnse04056@fpt.edu.vn](mailto:Khanhlnse04056@fpt.edu.vn) | Team member |

## Background

The need of pet service is becoming more popular nowadays, according to the survey of Q&ME, 40% Vietnamese often wear clothes for their pet and many people remember and celebrate birthdays for these little friends. Many groups on social network (Facebook, …) are created for response the need of people; however, it causes some troubles for customer.

## The problem

After surfing some groups about pet on Facebook, we realize that the post management of each group are different but mostly make user so confuse when search for something, sometime user takes a lot of time to wait for the answer of the answered question. And it also costs a lot of time when user want to go to the store.

## Literature view

There are some websites provide pet service but not provides breed match and direct service:

Pethome.vn

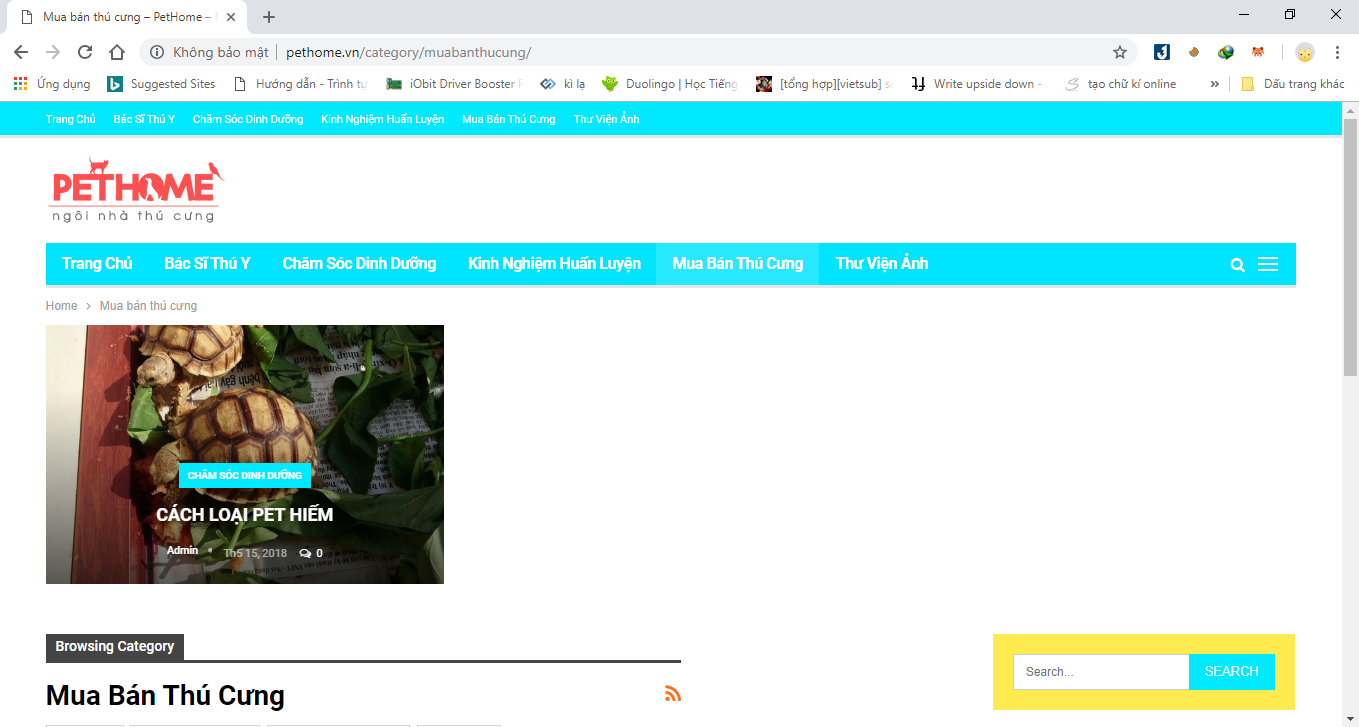


Figure 1: Pethome.vn

And some groups on Facebook:

“Đảo mèo”

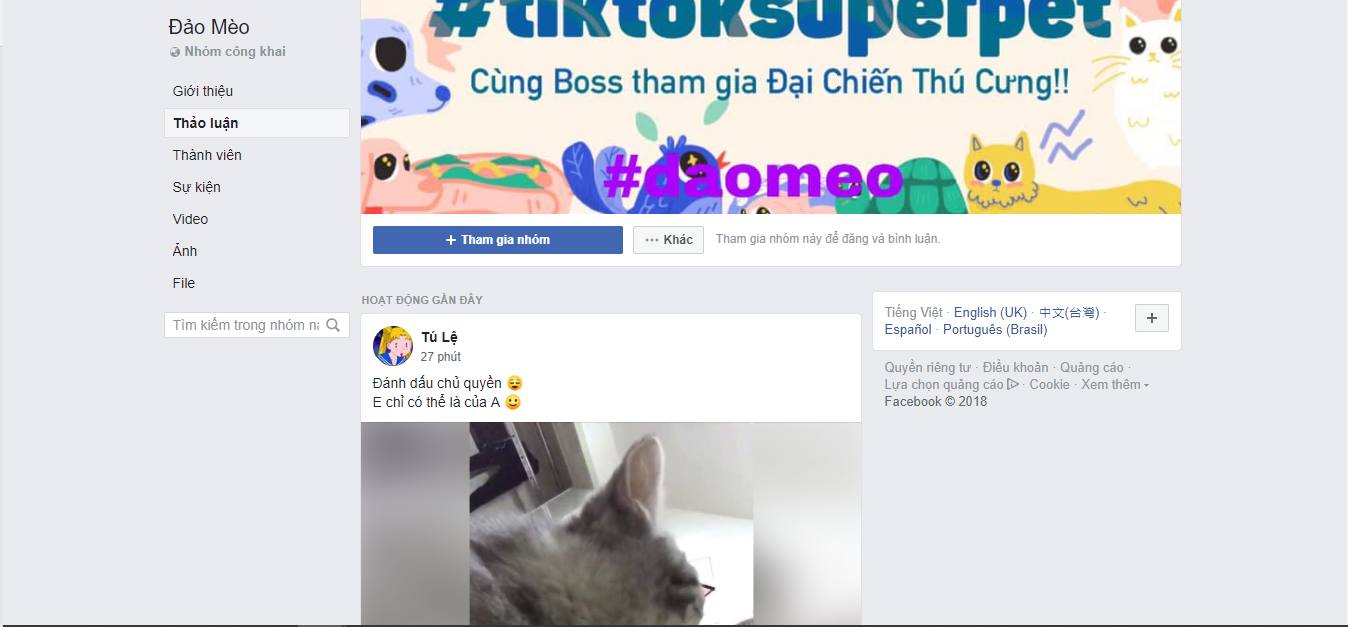


Figure 2: “Đảo mèo”

“Chủng tộc đại ngáo husky”



Figure 3: “Chủng tộc đại ngáo husky”

## The proposal system

### Our ideas

The idea of our team is create an application and web application to connect the consumers to other and the location manager to consumer

* The location managers are able to:
  + Add product or service to their location.
  + Manager their location.
* The consumers are able to:
  + Share information.
  + Interact with others.

### The proposal

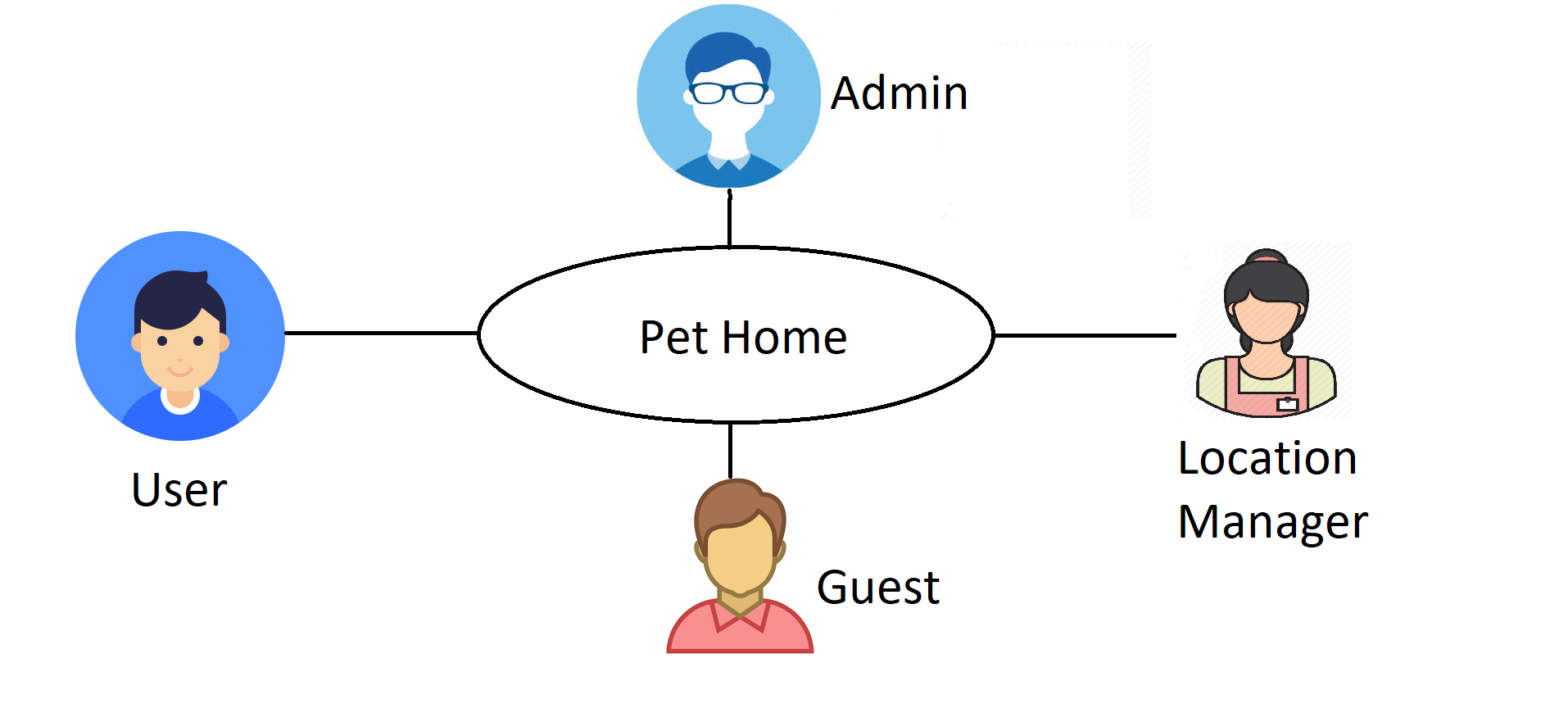


Figure 4: Role in PH system

The Pet Home system desires to provide better reaction between user with each other and the location manager with user by internet technology through following service:

* Full text search for post, location
* Real time message to communication between to user or more

### System group function

The PH has the following roles and main functions:

* Guest
  + View and search post
  + View and search location
* User
  + Create and manage their post
  + Create and manage their pet
  + View and search post, location
  + Interact with other users
* Location manager
  + Create and manage location (include: category, product, service)
* Administrator
  + Manage user
  + Create and manage public location
  + Manage report

# Chapter 2: Project manager plan



## Purpose

This section describes the software process model, team organization and management plan of the project. Team members must follow this section to complete assigned tasks and deadline.

## Software process model

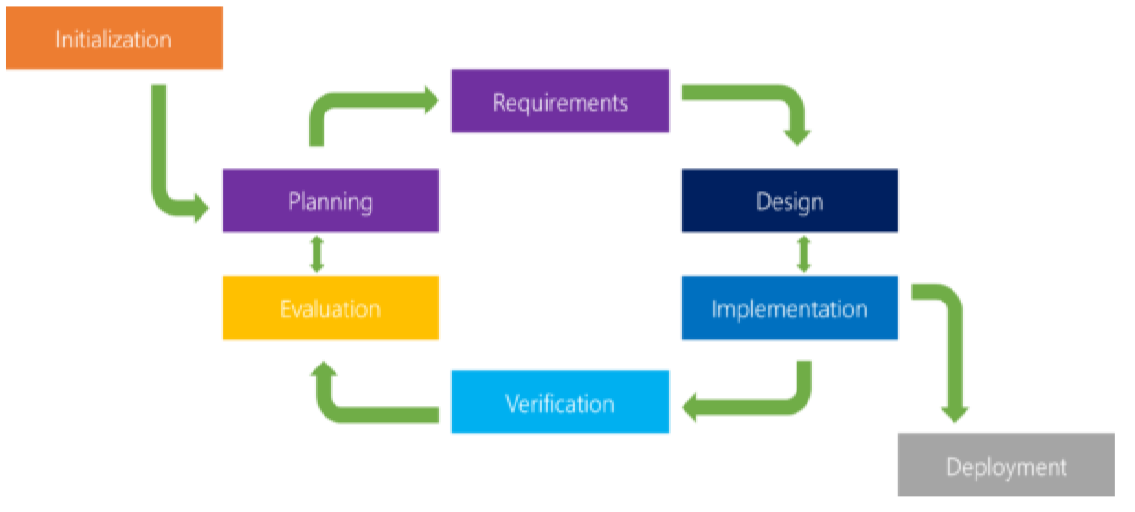


Figure 5: Iterative process model development

### About the iterative model

The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. Each cycle (iteration) ends with a usable system.

* Planning & Requirements: the first step is go through an initial planning stage to map out the specification documents.  Analysis & Design: the second step is performed to nail down the appropriate business logic, database models. The design stage also occurs here, establishing any technical requirements to meet the needs of the analysis and design stage.
* Implementation: the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project. Verification and testing: the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up.
* Evaluation: the last step is evaluating the development up to this stage. This allows the entire team to examine where the project is at, where it needs to be, what should change.

### Advantage and disadvantage considering

|  |  |
| --- | --- |
| **Advantages** | **Disadvantage** |
| * Innovative technology is involved – developer team can learn as they work on the project. * The major requirements are defined, while some functionalities enhancements evolve with the process of the development process. * Functional prototypes are developed early and delivery for each iteration. | * Heavily document changes when there are more and more iterations. * Issues about system architecture may turn out to be a big problem when lacking of a full requirements specification * Highly skilled resources are required for analysis skill.. |

Table 3: Advantage and Disadvantage Considering

## Organization structure

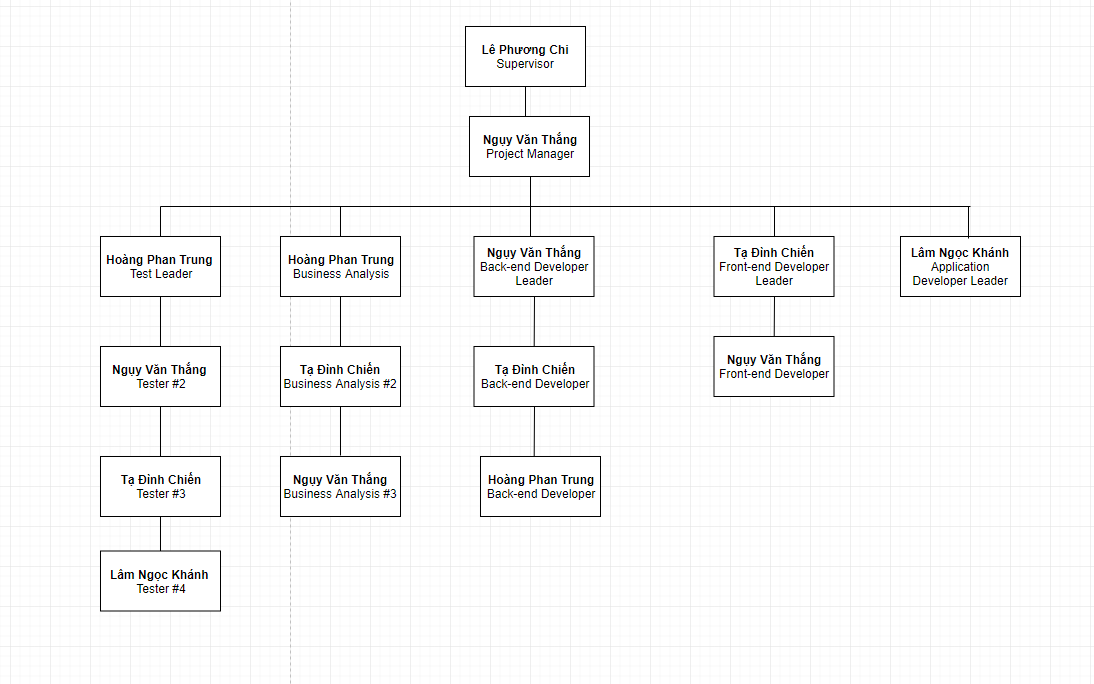


Figure 6: Organization Structure

## Role and responsibilities

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibilities** | **Full name** |
| **Project manager** | | |
| Project manager | * Guide team toward the goal of successfully passing the final capstone project. * Develop schedule and assigning task with responsibilities for each member. * Review and make sure the capstone project succeed. | Ngụy Văn Thắng |
| **Business analyst team** | | |
| BA leader | * Elicit and analyze requirement. * Define scope and create SRS template. * Design entity relationship diagram. | Hoàng Phan Trung |
| BA #1 | * Define business process flow and object state. | Tạ Đình Chiến |
| BA #2 | * Capture and specific describe use case. | Ngụy Văn Thắng |
| **Developer** | | |
| Architecture leader | * Define high level architecture base on SRS. * Implement configuration and web server. * Coding for controller, services and server API. | Ngụy Văn Thắng |
| Developer #1 | * Research and implement UX design for website. * Design and coding frontend objects for website. | Tạ Đình Chiến |
| Developer #2 | * Research and implement UX design for mobile application. * Design and coding frontend objects for mobile application. | Lâm Ngọc Khánh |
| **Tester** | | |
| Test leader | * Create template testing document. * Define test strategy, create test plan and defect log template | Hoàng Phan Trung |
| Test #1 | * Implement test case and log defect | Tạ Đình Chiến |
| Test #2 | * Implement test case and log defect | Ngụy Văn Thắng |
| Test #3 | * Implement test case and log defect | Lâm Ngọc Khánh |

Table 4: Role and Responsibilities

## Tool and technique

|  |  |
| --- | --- |
| **Software tools** | **Description** |
| IDE | Visual Studio Code, Robo 3T. |
| Git | Github. |
| Documentation tools | Microsoft word 2016, Sticky Note, Project 2016. |
| Communication tools | Slack, Facebook. |
| Manager tools | Trello. |
| UML tools | Enterprise Architect, Balsamiq Mockups 3. |

Table 5: Tool and Technique

## Project manager plan



### Project Schedule

The tasks list detail is described in: ProjectSchedule.mpp

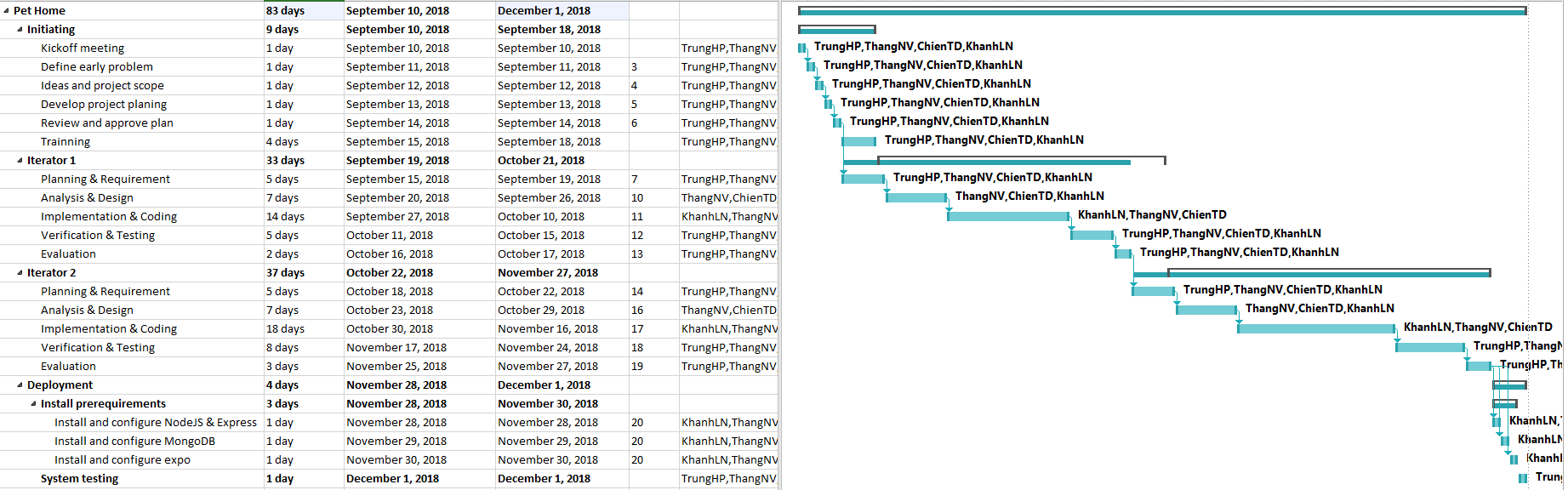


Figure 7: Tasks list



### Meeting minutes

Meeting minutes are written for any meeting. The following template is used:

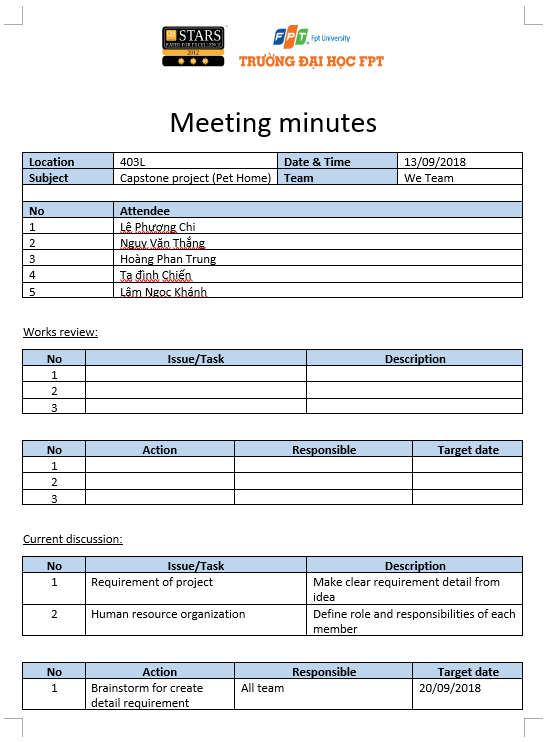


Figure 8: Meeting minutes example

### Risk Management plan

After using brainstorming, we categorized risk list in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk number (R3)** | **Risk description** | **Knowledge area** | **Root cause** | **Potential respond** |
| 1 | Poor planning work task for each member, impropriate resource allocation | Integration | Lacking experience of project manager | Discuss with team member for each task planning |
| 2 | Poor defining scope, ambiguous definition | Scope | Weak elicitation requirement of BA | Meeting with BA expert |
| 3 | Change objective during the project | Scope | Different opinion between team member | Redefine objective with each team member |
| 4 | Fail for estimating iteration time, inadequate change | Time | Developer team is lack of predict work process. | Doing overnight work to keep process continuing |
| 5 | Weak of technique | Quality | Team work with new technology | Meeting with expert for help |
| 6 | Poor solving conflict management. | Human resource | Team member not satisfy for work | Working in group team building |
| 7 | Document for meeting minute is not documented or not clear | Document change | Document is not captured. | Using notes, images, video |
| 8 | Unclear analysis of risk | Risk | Analysis risk is not clear between testers. | Discussion with team member |
| 9 | Poor assurance management | Risk | Testing leader is lack of experience | Meeting with expert |
| 10 | Meeting time conflict with domain expert. | Stakeholders | Team members don’t have appropriate time for meeting | Change the time meeting |

Table 6: Risk Management Plan

### Communication plan

Member information:

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Contact information** | **Position** |
| 1 | Ngụy Văn Thắng | Email: [Thangnvse04385@fpt.edu.vn](mailto:Thangnvse04385@fpt.edu.vn)  Phone number: 098 457 0673 | Project manager |
| 2 | Hoàng Phan Trung | Email: [Trunghpse04281@fpt.edu.vn](mailto:Trunghpse04281@fpt.edu.vn)  Phone number: 034 225 6852 | Tester |
| 3 | Tạ Đình Chiến | Email: [Chientdse04142@fpt.edu.vn](mailto:Chientdse04142@fpt.edu.vn)  Phone number: 096 925 5141 | Developer |
| 4 | Lâm Ngọc Khánh | Email: [Khanhlnse04056@fpt.edu.vn](mailto:Khanhlnse04056@fpt.edu.vn)  Phone number: 096 591 7064 | Developer |

Table 7: Member Information

Project report and meeting:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Activity** | **Stakeholder** | **Time** | **Description** |
| 1 | Weekly report | * Team member * Supervisor | Friday | * Report in meeting minutes * Review status based on weekly report * Plan for the next week |
| 2 | Daily report | Project manager | Daily | * Report issue and planning * Support to solve issues |
| 3 | Ad-hoc report | Team members | Event-based | * Report and talking face to face |

Table 8: Project report and Meeting

Project communication:

|  |  |  |
| --- | --- | --- |
| **No** | **Style** | **Description** |
| 1 | Weekly meeting schedule | Spend one day in a week for meeting to review and update solution for problems. |
| 2 | Unscheduled meeting | If any member has problems that he wants to solved immediately, we will have a chat box. |
| 3 | Communication channel | Our main communication are physical meeting, chat box, Facebook messenger. |

Table 9: Project Communication

### Coding convention

The following coding conventions are used to code the project to help improve readabilities, maintenance abilities, and code more easily.

|  |  |  |
| --- | --- | --- |
| **#** | **Style** | **Implementation component** |
| 1 | JS coding convention | Web service API |
| 2 | React coding convention | Application and web client |

Table 10: Coding Convention

# Chapter 3: Software requirement specification



## Purpose

The purpose of this chapter is to elicitation use cases to clarify how user interact with system, what are the functions and how the functions will perform. It includes the business process flow diagram, state transition, and use case detailed description of the PH system.

## Definition, acronyms, and abbreviations

|  |  |  |
| --- | --- | --- |
| **No.** | **Term** | **Definition** |
| 1 | Guest | Person who visit the application without registration. |
| 2 | Location manager | Person who logged in to website manage location |
| 3 | User | Person who visit the application with registration. |
| 4 | Administrator | Person who manage all system |
| 5 | PH | Pet home |

Table 11: Definition, Acronyms, and Abbreviations

## User Requirement Specification

* Administrator: Manager all of information by using website manager. Only admin can use this web.
* User: Login on mobile application for interactive with other users. User can also interactive with shop or public location.
* Guest: Using mobile application without login. Guest can search for posts with many options, view post but can’t comment.
* Location Manager: Login on web application to manage location.

## System Requirement Specification (Specific Requirements)



### External interface requirements

* + - 1. **User interface**
* The text is written in Vietnamese.
  + - 1. **hardware interface**
* **Mobile:**
  + Touched screen.
  + GPS: A-GPS
  + GPRS/ 3G: optional.
  + Wi-Fi: Wi-Fi 802.11 b/g/n.
* **Computer:** 
  + CPU: at least Intel Pentium D 2.4GHz.
  + RAM: 2 GB.
  + Internet browser.
    - 1. **software interface**
* Mobile device: IOS or Android
* Computer: Chrome 35 or later
  + - 1. **communication protocol**
* The system use HTTP protocol connection to communicate with server through JSON.

### Functional requirements

* + - 1. **business rule repository**

|  |  |
| --- | --- |
| **No.** | **Description** |
| BI-1 | The field must be filled by characters consist of alphabet, numbers and space. |
| BI-2 | The field must be filled by characters consist of alphabet and numbers. |
| BI-3 | The field must be filled, follow by format “X@X.Y”, where X is characters and Y is alphabets. X and Y are at least 1 character. |
| BI-4 | Start-date of a payment is after End-date. |
| BI-5 | Must enter from 6 to 50 characters. |
| BI-6 | Date follow format DD-mm-YYYY. |

Table 12: Business Rule Repository

* + - 1. **administrator**
         1. **use case diagram**

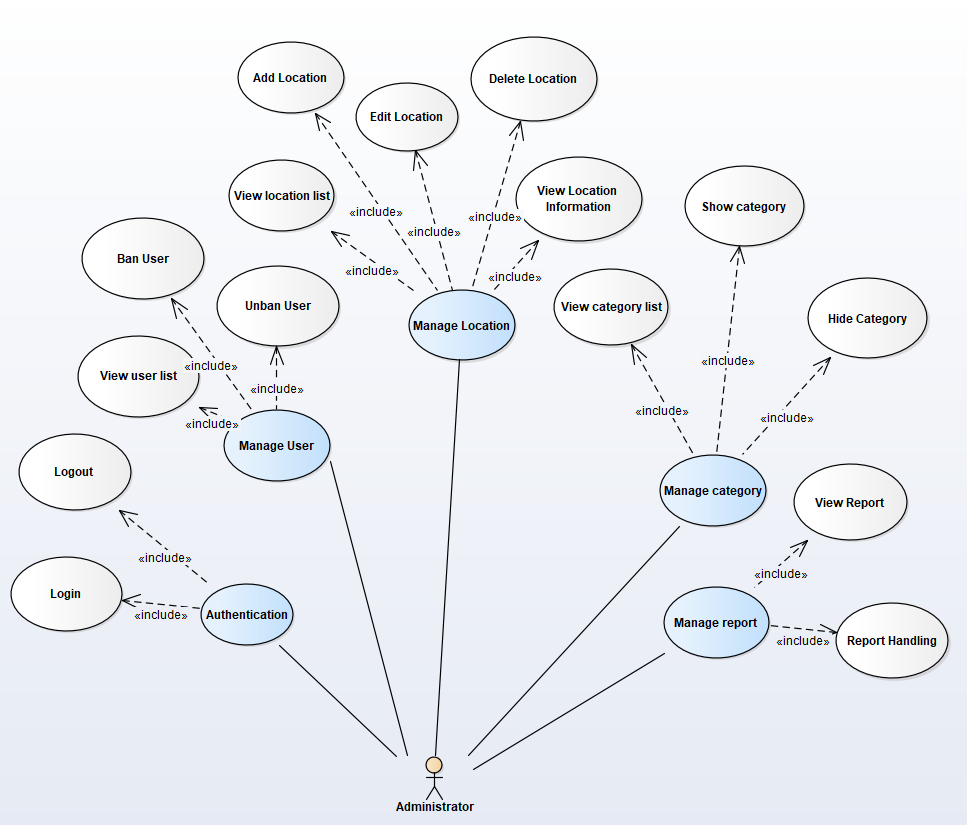


Image 1: Administrator use case diagram

* + - * 1. **use case summary**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Use case ID** | **Use case name** |
| Administrator | AD-UC-001 | View user list |
| AD-UC-002 | Ban user |
| AD-UC-003 | Add location |
| AD-UC-004 | View location list |
| AD-UC-005 | Show location |
| AD-UC-006 | Edit location |
| AD-UC-007 | Logout |
| AD-UC-008 | Login |
| LM-UC-011 | Change password |
| AD-UC-009 | View category list |
| AD-UC-010 | Show category |
| AD-UC-013 | Hide category |
| AD-UC-014 | View report |
| AD-UC-015 | View location information |
| AD-UC-016 | Hide location |
| AD-UC-017 | Handle report |
| AD-UC-018 | Unban User |

Table 13: Administrator usecase summary

* + - * 1. **manage user**

**use case diagram**

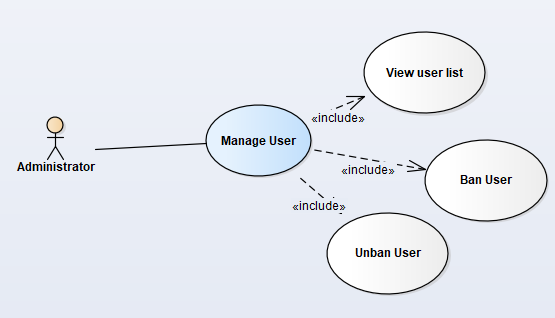


Image 2: Manage user function use case diagram

**use case specification**

View user list

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-001 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-001 | **Use case version** | V1.0 |
| **Use case name** | View user list | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to view all user registered in system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can view all user registered in system. | | |
| **Trigger** | Administrator clicks “Người dùng” link in the left side bar. | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Người dùng” link in the left side bar. |  | | **2.** |  | Display all user account and information of account. |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Ban user

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-002 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-002 | **Use case version** | V1.0 |
| **Use case name** | Ban user | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to ban reported user. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can ban reported user. | | |
| **Trigger** | Click “Chặn” button | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do AD-UC-001 |  | | **2.** |  | Display all user account and information of account. | | **3.** | Click “Chặn” button |  | | **4.** |  | Show success message and status of account display “Ngưng hoạt động”. |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Unban user

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-0018 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-002 | **Use case version** | V1.0 |
| **Use case name** | Unban user | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to unban user. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can unban reported user. | | |
| **Trigger** | Click “Bỏ chặn” button | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do AD-UC-001 |  | | **2.** |  | Display all user account and information of account. | | **3.** | Click “Bỏ chặn” button |  | | **4.** |  | Show success message and status of account display “Hoạt động”. |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **manage location**

**use case diagram**

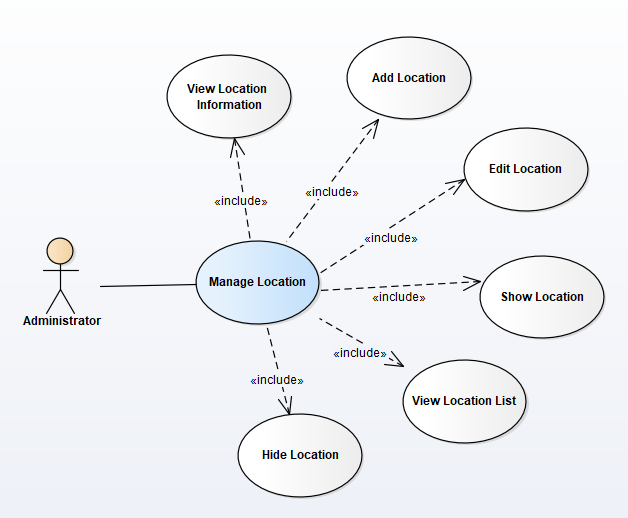


Image 3: Manage location function use case diagram

**use case specification**

Add location

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-003 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-003 | **Use case version** | V1.0 |
| **Use case name** | Add location | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to add location to system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can add location to system. | | |
| **Trigger** | Administrator click “Địa điểm” link. | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Thể loại” link |  | | **2.** |  | Show menu of “Thể loại” | | **3.** | Click “Địa điểm” link |  | | **4.** |  | Display location page. | | **5.** | Click “Thêm mới” button |  | | **6.** |  | Display add new location page. | | **7.** | Actor fill all information of location and click “Thêm” button |  | | **8.** |  | Show success message and redirect to location page |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 7 | | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Hủy” button |  | | **2.** |  | Show confirm dialog | | **3.** | Click “Có” button |  | | **4.** |  | Close confirm dialog and redirect to location page |  |  |  |  | | --- | --- | --- | | **ET1.1** | At step 3 | | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Không” button |  | | **2.** |  | Close confirm dialog |   **Exception: N/A**  **Business Rules:** [**BI-2**](#BI2) | | | |

View location list

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-004 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-004 | **Use case version** | V1.0 |
| **Use case name** | View location list | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to view all location in system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can view detail of all location to system. | | |
| **Trigger** | Administrator click “Địa điểm” link | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Thể loại” link |  | | **2.** |  | Show menu of “Thể loại” | | **3.** | Click “Địa điểm” link |  | | **4.** |  | Display location page. |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Show location

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-005 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-005 | **Use case version** | V1.0 |
| **Use case name** | Show location | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to show a location has been hide in system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can show a location has been hide in system. | | |
| **Trigger** | Administrator press “Hiện” button. | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Thể loại” link |  | | **2.** |  | Show menu of “Thể loại” | | **3.** | Click “Địa điểm” link |  | | **4.** |  | Display location page. | | **5.** | Click “Hiện” button |  | | **6.** |  | Show confirm dialog | | **7.** | Click “Có” button |  | | **8.** |  | Close confirm dialog, show success message and change button “Hiện” to “Ẩn” |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 7 | | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Không” button |  | | **2.** |  | Close confirm dialog |   **Exception: N/A**  **Business Rules: N/A** | | | |

Edit location

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-006 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-006 | **Use case version** | V1.0 |
| **Use case name** | Edit location | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to edit a location added to system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can edit information of location added to system. | | |
| **Trigger** | Administrator click “Sửa” button | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do [AD-UC-015](#ViewLocationInforAdmin) |  | | **2.** |  | Show location information page | | **3.** | Click “eye icon” button |  | | **4.** |  | Display edit location page. | | **5.** | Input information (image, text, …) press “Lưu” button |  | | **6.** |  | Show success message, direct to location information page |   **Alternative Flows: N/A**    **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 5 | | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Hủy” button |  | | **2.** |  | Show confirm dialog | | **3.** | Click “Có” button |  | | **4.** |  | Close dialog and direct to location information page |  |  |  |  | | --- | --- | --- | | **ET1.1** | At step 3 | | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Hủy” button |  | | **2.** |  | Close confirm dialog |   **Exception: N/A**  **Business Rules: N/A** | | | |

View location information

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-015 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-015 | **Use case version** | V1.0 |
| **Use case name** | View location information | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to view information of location added to system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can view information of location added to system. | | |
| **Trigger** | Click “eye icon” button | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Thể loại” link |  | | **2.** |  | Show menu of “Thể loại” | | **3.** | Click “Địa điểm” link |  | | **4.** |  | Display location page. | | **5.** | Click “eye icon” button |  | | **6.** |  | Show location information page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Hide location

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-016 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-016 | **Use case version** | V1.0 |
| **Use case name** | Hide location | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to hide a location showing on system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can hide a location showing on system. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Thể loại” link |  | | **2.** |  | Show menu of “Thể loại” | | **3.** | Click “Địa điểm” link |  | | **4.** |  | Display location page. | | **5.** | Click “Ẩn” button |  | | **6.** |  | Show confirm dialog | | **7.** | Click “Có” button |  | | **8.** |  | Close confirm dialog, show success message and change button “Ẩn” to “Hiện” |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 7 | | | **No.** | **Actor events** | **System respond** | | **1.** | Actor click “Không” button |  | | **2.** |  | Close confirm dialog |   **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **Authentication**

**use case diagram**

****

Image 4:Administrator authentication function use case diagram

**use case specification**

Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-007 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-007 | **Use case version** | V1.0 |
| **Use case name** | Logout | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to logout from manage system web. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can logout from manage system web. | | |
| **Trigger** | N/A | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the small image in the right corner |  | | **2.** |  | Show menu | | **3.** | Click “Đăng xuất” item |  | | **4.** |  | Direct to login page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Login

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-008 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-008 | **Use case version** | V1.0 |
| **Use case name** | Login | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to login to manage system web. | | |
| **Pre-condition** | Administrator has an account with permission. | | |
| **Post-condition** | Administrator can login to manage system web. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Actor input information to “Số điện thoại” and “Mật khẩu” field and press “Đăng nhập” button |  | | **2.** |  | Direct to homepage of administrator |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI-2**](#BI2)**,** [**BI-5**](#BI5) | | | |

Change password

* + - * 1. **manage category**

**use case diagram**

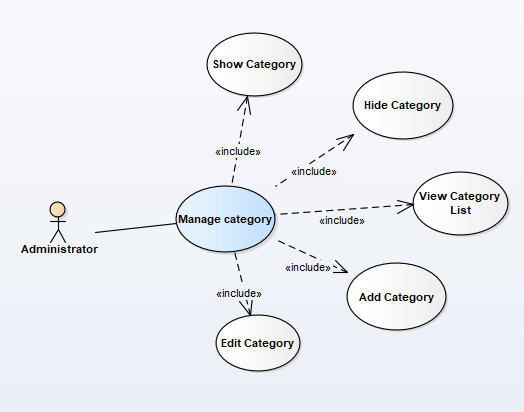


Image 5:Administrator manage category function use case diagram

**use case specification**

View category list

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-009 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-009 | **Use case version** | V1.0 |
| **Use case name** | View category list | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to view all category of system. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can view all category of system. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Quản lý category” menu at left side bar |  | | **2.** |  | Show all item of menu |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Show category

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-010 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-010 | **Use case version** | V1.0 |
| **Use case name** | Show category | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to search all category of system by default data. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can search all category of system by default data. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Quản lý category” menu at left side bar |  | | **2.** |  | Show all item of menu | | **3.** | Click on “Hiện” button |  | | **4.** |  | The button change to “Ẩn” |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Hide category

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-013 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-013 | **Use case version** | V1.0 |
| **Use case name** | Hide category | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Low |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to hide added category. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can hide added category. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Quản lý category” menu at left side bar |  | | **2.** |  | Show all item of menu | | **3.** | Click on “Ẩn” button |  | | **4.** |  | The button change to “Hiện” |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **manager report**

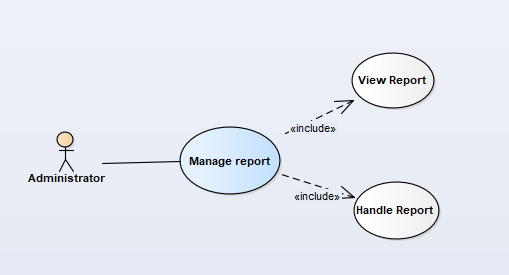


Image 6: Administrator manage report function use case diagram

**use case diagram**

**use case specification**

View report

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-014 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-014 | **Use case version** | V1.0 |
| **Use case name** | View report | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Low |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to view all report from all actor. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can view all report from all actor. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Báo cáo” menu at left side bar |  | | **2.** |  | Show report page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Handle report

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE AD-UC-017 SPECIFICATION** | | | |
| **Use case ID** | AD-UC-017 | **Use case version** | V1.0 |
| **Use case name** | Handle report | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Administrator | **Secondary actor** | N/A |
| **Description** | Allows administrator to handle all report from all actor. | | |
| **Pre-condition** | Administrator logged in to admin web application. | | |
| **Post-condition** | Administrator can handle all report from all actor. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do UC-AD-014 |  | | **2.** |  | Show all item of menu | | **3.** | Click “Bài viết bị báo cáo” link |  | | **4.** |  | Show menu action | | **5.** | Choose one item |  | | **6.** |  | Show success message and change status to “Đã xử lý” |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - 1. **location manager**
         1. **use case diagram**

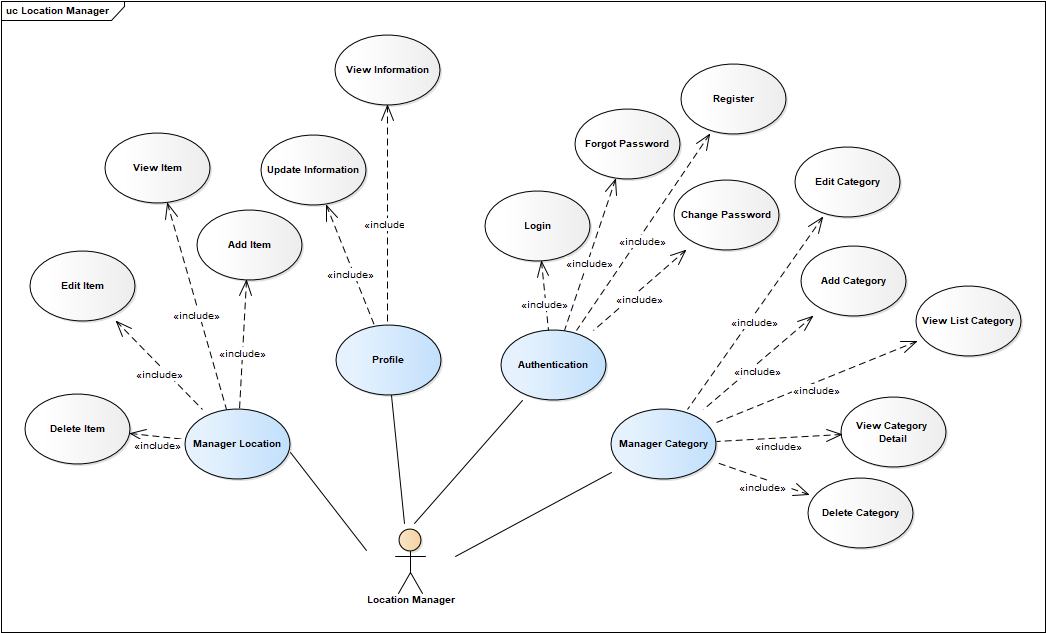
****

Image 7: Location Manager use case diagram

* + - * 1. **use case summary**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Use case ID** | **Use case name** |
| Location manager | AD-UC-007 | Logout |
| AD-UC-008 | Login |
| LM-UC-003 | Register |
| LM-UC-004 | Forgot password |
| LM-UC-005 | Delete item |
| LM-UC-006 | Add item |
| LM-UC-007 | Edit item |
| LM-UC-008 | View item |
| LM-UC-009 | Update information |
| LM-UC-010 | View information |
| LM-UC-011 | Change password |
| LM-UC-012 | Edit category |
| LM-UC-013 | Add category |
| LM-UC-014 | View list category |
| LM-UC-015 | View category detail |
| LM-UC-016 | Delete category |

Table 14: Location Manager usecase diagram

* + - * 1. **authentication**

**use case diagram**

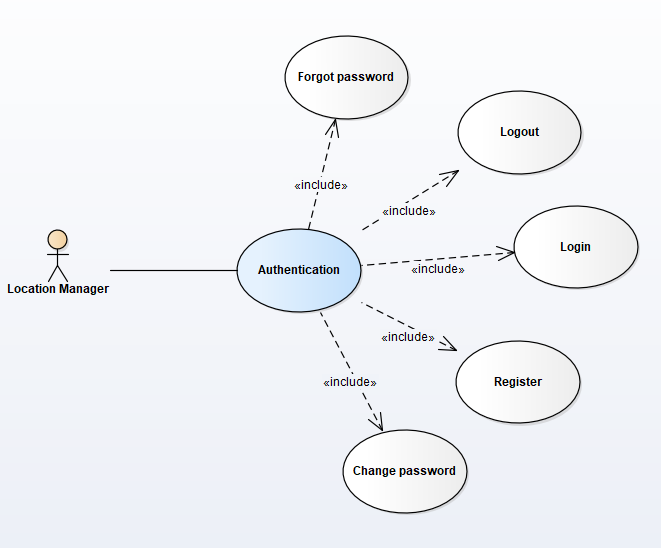


Image 8: Location manager authentication function use case diagram

**use case specification**

Register

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-003 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-003 | **Use case version** | V1.0 |
| **Use case name** | Register | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to register an account in location manager system. | | |
| **Pre-condition** | Location manager access to location manage web application. | | |
| **Post-condition** | Location manager can register an account in location manager system. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Đăng ký” |  | | **2.** |  | Show register page | | **3.** | Fill all information require in page and click “Đăng ký” button |  | | **4.** |  | Direct to verification code page and send a code include six numbers to actor’s phone at the same time | | **5.** | Input verification code and click “Xác minh” button |  | | **6.** |  | Direct to success notice page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI5**](#BI5) | | | |

Forgot password

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-004 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-004 | **Use case version** | V1.0 |
| **Use case name** | Forgot password | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Low |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to reset password of their account in location manager system. | | |
| **Pre-condition** | * Location manager access to location manage web application. * Location manager has an account with permission. | | |
| **Post-condition** | Location manager can reset password of their account in location manager system. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Quên mật khẩu” |  | | **2.** |  | Direct to verification code page and send a code include six numbers to actor’s phone at the same time | | **3.** | Input verification code and click “Xác minh” button |  | | **4.** |  | Direct to “Quên mật khẩu” page | | **5.** | Input all field require in page and click “Thay đổi” button |  | | **6.** |  | Direct to success notice page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI5**](#BI5) | | | |

Change password

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-011 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-011 | **Use case version** | V1.0 |
| **Use case name** | Change password | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to change their password. | | |
| **Pre-condition** | Location manager has logged in location manage system. | | |
| **Post-condition** | Location manager can change their password. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do LM-UC-010 |  | | **2.** |  | Direct to “Thông tin cá nhân” page | | **3.** | Click “Thay đổi mật khẩu” button |  | | **4.** |  | Direct to “Thay đổi mật khẩu” page | | **5.** | Input all field require in page and click “Thay đổi” button |  | | **6.** |  | Direct to “Thông tin cá nhân” page and display success message |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **manage location**

**use case diagram**

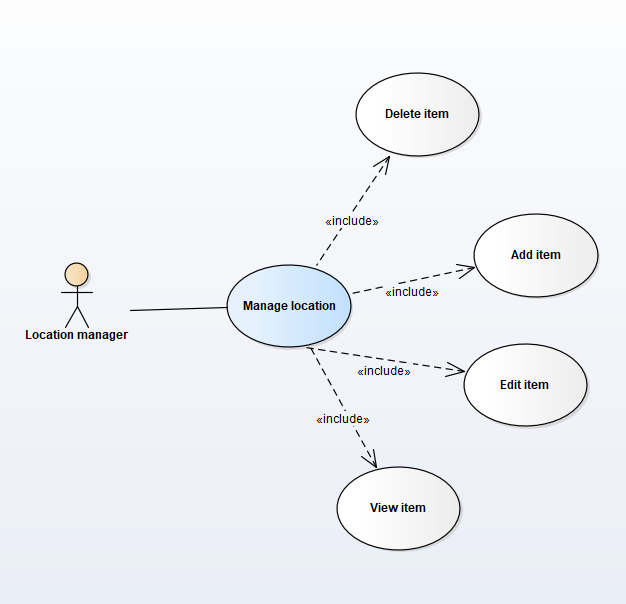


Image 9: Manager location manage location function use case diagram

**use case specification**

Delete item

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-005 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-005 | **Use case version** | V1.0 |
| **Use case name** | Delete item | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Low |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to delete items of their managed location. | | |
| **Pre-condition** | * Location manager logged in to location manage web application. * Location manager has a location registered in system. | | |
| **Post-condition** | Location manager can delete items of their managed location. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click recycle bin icon |  | | **2.** |  | Show confirm dialog | | **3.** | Click “Có” button |  | | **4.** |  | Close dialog, show success message and the item disappear. |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 3 | | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Không” button |  | | **2.** |  | Close confirm dialog |   **Exception: N/A**  **Business Rules: N/A** | | | |

Add item

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-006 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-006 | **Use case version** | V1.0 |
| **Use case name** | Add item | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to add items to their managed location. | | |
| **Pre-condition** | * Location manager logged in to location manage web application. * Location manager has a location registered in system. | | |
| **Post-condition** | Location manager can add items to their managed location. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Thêm sản phẩm” button |  | | **2.** |  | Direct to “Thêm sản phẩm” page | | **3.** | Fill all require information and click “Thêm sản phẩm” |  | | **4.** |  | Show success message and direct to home page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI1**](#BI1) | | | |

Edit item

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-007 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-007 | **Use case version** | V1.0 |
| **Use case name** | Edit item | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to edit information of items of their managed location. | | |
| **Pre-condition** | * Location manager access to location manage web application. * Location manager has a location registered in system. | | |
| **Post-condition** | Location manager can edit information of items of their managed location. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Sửa” button |  | | **2.** |  | Direct to “Sửa sản phẩm” page | | **3.** | Fill all require information and click “Lưu” |  | | **4.** |  | Show success message and direct to home page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI1**](#BI1) | | | |

View item

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-008 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-008 | **Use case version** | V1.0 |
| **Use case name** | View item | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to view information of items of their managed location. | | |
| **Pre-condition** | * Location manager access to location manage web application. * Location manager has a location registered in system. | | |
| **Post-condition** | Location manager can view information of items of their managed location. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click name of item in “Tên sản phẩm” column |  | | **2.** |  | Direct to “Chi tiết sản phẩm” page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **profile**

**use case diagram**

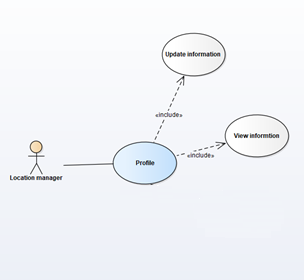


Image 10: Location manager profile function use case diagram

**use case specification**

Update information

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-009 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-009 | **Use case version** | V1.0 |
| **Use case name** | Update information | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to update information of their profile. | | |
| **Pre-condition** | Location manager logged in to location manage web application. | | |
| **Post-condition** | Location manager can view information of items of their managed location. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do LM-UC-010 |  | | **2.** |  | Direct to “Thông tin cá nhân” page | | **3.** | Click “Thay đổi thông tin cá nhân” button |  | | **4.** |  | All field change to editable field | | **5.** | Input information and click “Lưu” button |  | | **6.** |  | Show success message and display information |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

View information

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-010 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-010 | **Use case version** | V1.0 |
| **Use case name** | View information | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to view information of their profile. | | |
| **Pre-condition** | Location manager logged in to location manage web application. | | |
| **Post-condition** | Location manager can view information of their profile. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click the image in the left of top corner |  | | **2.** |  | Show menu | | **3.** | Choose “Thông tin cá nhân” item |  | | **4.** |  | Direct to “Thông tin cá nhân” page |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **manage category**

**use case diagram**

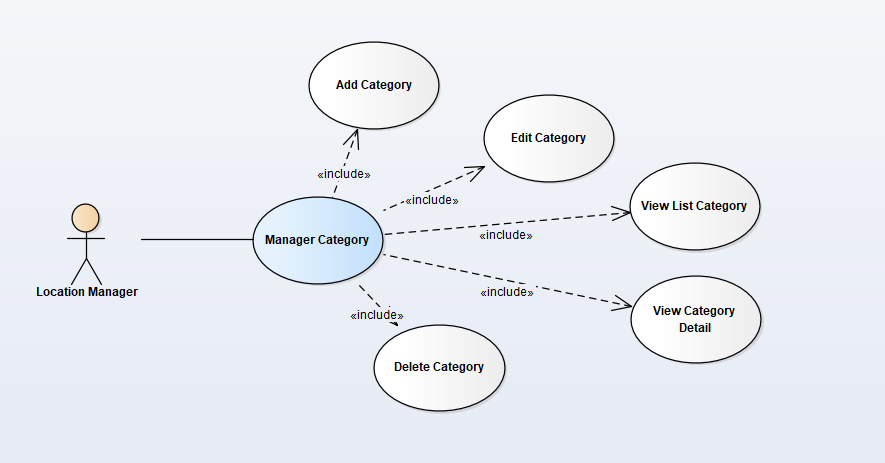


Image 11: Location manager manage category function use case diagram

**use case specification**

Edit category

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-012 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-012 | **Use case version** | V1.0 |
| **Use case name** | Edit category | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to change their created category | | |
| **Pre-condition** | * Location manager logged in to location manage web application. * Location manager click to category at the left menu | | |
| **Post-condition** | Location manager can to change their created category | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click the category |  | | **2.** |  | Show edit category page | | **3.** | Input data and click “Lưu” button |  | | **4.** |  | Show success message |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Add category

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-013 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-013 | **Use case version** | V1.0 |
| **Use case name** | Add category | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to add a category | | |
| **Pre-condition** | * Location manager logged in to location manage web application. * Location manager click to category at the left menu | | |
| **Post-condition** | Location manager can to add a category | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Thêm mới” button |  | | **2.** |  | Show add category page | | **3.** | Input data and click “Thêm” button |  | | **4.** |  | Show success message and direct to view category list page with this category at the first |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI1**](#BI1) | | | |

View list category

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-014 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-014 | **Use case version** | V1.0 |
| **Use case name** | View list category | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to view all their added category | | |
| **Pre-condition** | * Location manager logged in to location manage web application. | | |
| **Post-condition** | Location manager can view all their added category | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click category item at the left side bar |  | | **2.** |  | Show all added category |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

View category detail

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-015 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-015 | **Use case version** | V1.0 |
| **Use case name** | View category detail | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to view all their added category | | |
| **Pre-condition** | * Location manager logged in to location manage web application. * Location manager click to category at the left menu | | |
| **Post-condition** | Location manager can view all their added category | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click a category |  | | **2.** |  | Direct to view category detail with the information of this category in it. |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Delete category

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE LM-UC-016 SPECIFICATION** | | | |
| **Use case ID** | LM-UC-016 | **Use case version** | V1.0 |
| **Use case name** | Delete category | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | Location manager | **Secondary actor** | N/A |
| **Description** | Allows location manager to view all their added category | | |
| **Pre-condition** | * Location manager logged in to location manage web application. * Location manager click to category at the left menu | | |
| **Post-condition** | Location manager can view all their added category | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click a recycle bin icon next to a category name |  | | **2.** |  | Show confirm dilalog | | **3.** | Click “Có” button |  | | **4.** |  | Close confirm dialog, show success message and this category disappear. |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 3 | | | **No.** | **Actor events** | **System respond** | | **1.** | Click “Không” button |  | | **2.** |  | Close confirm dilalog |   **Exception: N/A**  **Business Rules: N/A** | | | |

* + - 1. **user, guest**
         1. **use case diagram**

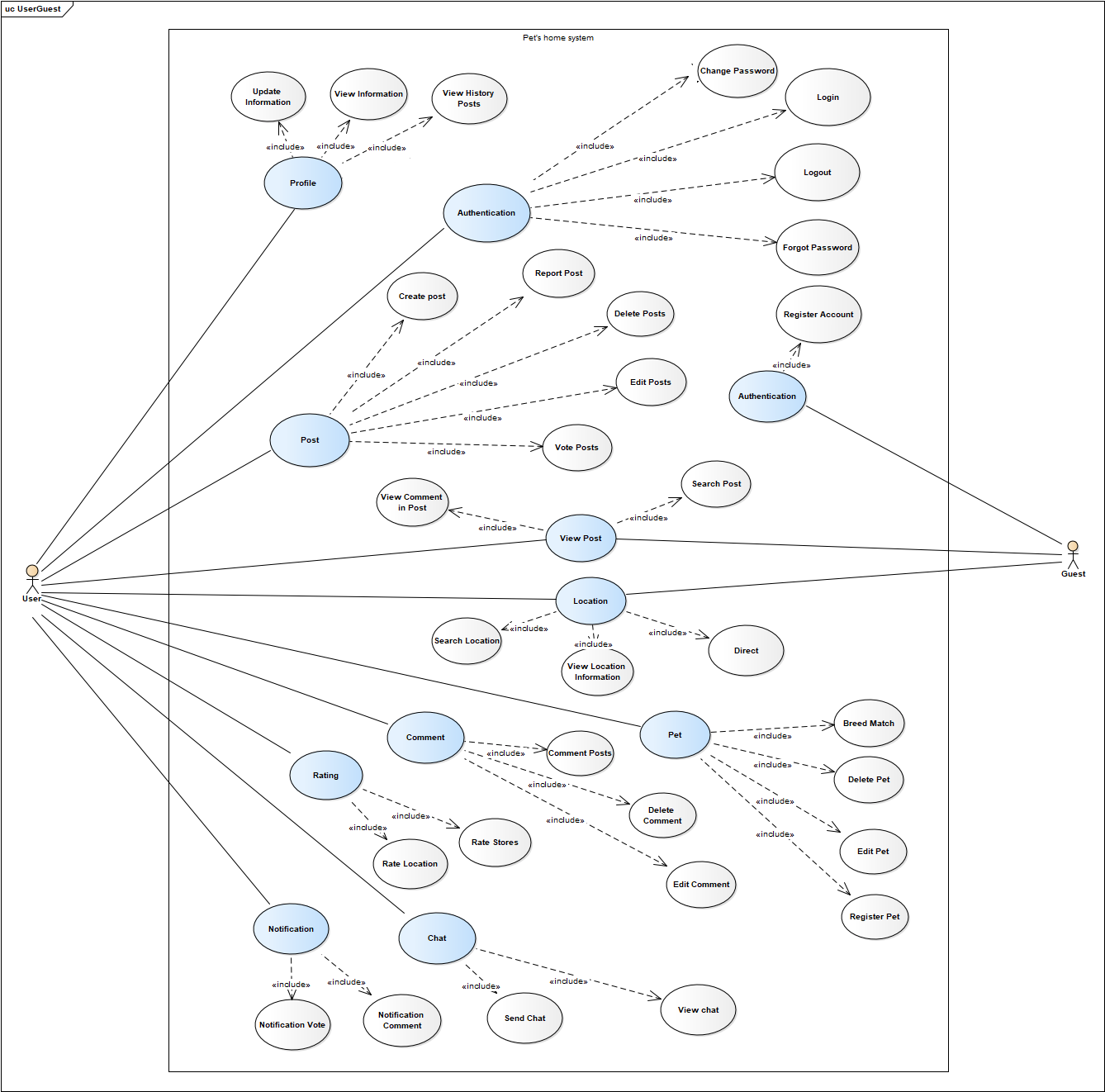
****

Image 12: User and Guest usecase diagram

* + - * 1. **use case summary**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Use case ID** | **Use case name** |
| User | US-UC-001 | Update information |
| US-UC-002 | View information |
| US-UC-003 | View history posts |
| US-UC-004 | Change password |
| US-UC-005 | Login |
| US-UC-006 | Logout |
| US-UC-007 | Forgot password |
| US-UC-008 | Create post |
| US-UC-009 | Report post |
| US-UC-010 | Delete posts |
| US-UC-011 | Edit posts |
| US-UC-012 | Vote posts |
| US-UC-013 | Breed match |
| US-UC-014 | Delete pet |
| US-UC-015 | Edit pet |
| US-UC-016 | Register pet |
| US-UC-017 | Comment posts |
| US-UC-018 | Delete comment |
| US-UC-019 | Edit comment |
| US-UC-020 | Rate stores |
| US-UC-021 | Rate location |
| US-UC-022 | Send chat |
| US-UC-023 | View chat |
| US-UC-024 | Notification comment |
| US-UC-025 | Notification vote |
| Guest | GU-UC-001 | Register account |
| User, Guest | UG-UC-001 | View comments in post |
| UG-UC-002 | Search post |
| UG-UC-003 | Search location |
| UG-UC-004 | View location information |
| UG-UC-005 | Direct |

Table 15: Usser and Guest usecase summary

* + - * 1. **profile**

**use case diagram**

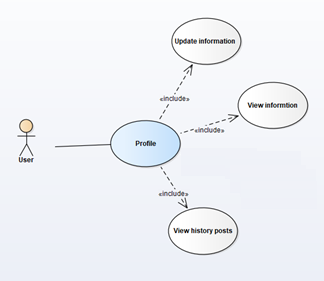
****

Image 13: User profile function use case diagram

**use case specification**

Update information

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-001 SPECIFICATION** | | | |
| **Use case ID** | US-UC-001 | **Use case version** | V1.0 |
| **Use case name** | Update information | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to update their account information. | | |
| **Pre-condition** | User logged in to application. | | |
| **Post-condition** | User can update their account information. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do US-UC-002 |  | | **2.** |  | Direct to “Thông tin cá nhân” interface | | **3.** | Press edit icon at the right of bottom corner |  | | **4.** |  | Display change information interface | | **5.** | Choose one type of information want to change |  | | **6.** |  | Direct to change type of information change | | **7.** | Input information and press save button at the right of top corner |  | | **8.** |  | Display success message and direct to information interface |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2) | | | |

View information

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-002 SPECIFICATION** | | | |
| **Use case ID** | US-UC-002 | **Use case version** | V1.0 |
| **Use case name** | View information | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to view their account information. | | |
| **Pre-condition** | User logged in to application. | | |
| **Post-condition** | User can view their account information. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the collapsible button at the top left corner |  | | **2.** |  | Show drawer | | **3.** | Choose “Thông tin cá nhân” item |  | | **4.** |  | Direct to “Thông tin cá nhân” interface |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

View history posts

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-003 SPECIFICATION** | | | |
| **Use case ID** | US-UC-003 | **Use case version** | V1.0 |
| **Use case name** | View history posts | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to view all their created post. | | |
| **Pre-condition** | User logged in to application. | | |
| **Post-condition** | User can view all their created post. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the collapsible button at the top left corner |  | | **2.** |  | Show drawer | | **3.** | Press the human icon |  | | **4.** |  | Show profile interface include all created post |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **authentication**

**use case diagram**

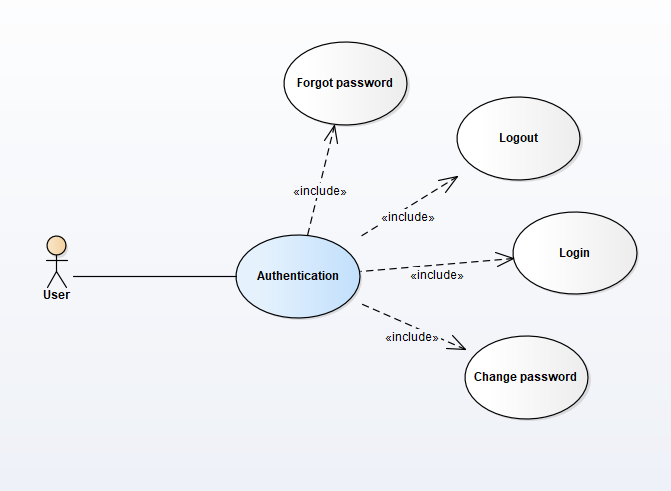


Image 14:User authentication function use case diagram

**use case specification**

Login

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-005 SPECIFICATION** | | | |
| **Use case ID** | US-UC-005 | **Use case version** | V1.0 |
| **Use case name** | Login | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to login to application. | | |
| **Pre-condition** | User access PH application. | | |
| **Post-condition** | User can login to application. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Fill phone number and password and press “Đăng nhập” button |  | | **2.** |  | Direct to home view |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI5**](#BI5) | | | |

Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-006 SPECIFICATION** | | | |
| **Use case ID** | US-UC-006 | **Use case version** | V1.0 |
| **Use case name** | Logout | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to logout from application. | | |
| **Pre-condition** | User logged in application. | | |
| **Post-condition** | User can logout from application. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the collapsible button at the top left corner |  | | **2.** |  | Show menu | | **3.** | Choose “Đăng xuất” item at the bottom |  | | **4.** |  | Direct to login view |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Forgot password

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-007 SPECIFICATION** | | | |
| **Use case ID** | US-UC-007 | **Use case version** | V1.0 |
| **Use case name** | Forgot password | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to reset their password. | | |
| **Pre-condition** | User has an account in PH system. | | |
| **Post-condition** | User can reset their password. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press “Quên mật khẩu” |  | | **2.** |  | Direct to verify forgot password interface | | **3.** | Input phone number and press “Gửi mã xác nhận button” |  | | **4.** |  | Show “Nhập mã xác nhập” pop-up and send a verification code include 6 numbers to actor’s phone at the same time | | **5.** | Input verification code and press “Xác nhận” button |  | | **6.** |  | Direct to “Cài lại mật khẩu” interface | | **7.** | Input all require field and press “Cài lại mật khẩu” button |  | | **8.** |  | Show success message and direct to login interface |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI5**](#BI5) | | | |

Change password

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-004 SPECIFICATION** | | | |
| **Use case ID** | US-UC-004 | **Use case version** | V1.0 |
| **Use case name** | Change password | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to change their password. | | |
| **Pre-condition** | User has an account in system. | | |
| **Post-condition** | User can change their password. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do US-UC-002 |  | | **2.** |  | Direct to “Thông tin cá nhân” interface | | **3.** | Press edit icon at the right of bottom corner |  | | **4.** |  | Display change information interface | | **5.** | Press “Thay đổi mật khẩu” button |  | | **6.** |  | Direct to “Đổi mật khẩu” interface | | **7.** | Fill all require field and press the save icon at the top of the right |  | | **8.** |  | Show success message and direct to change information interface |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 7 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press the left arrow at the top of the left |  | | **2.** |  | Direct to “Thông tin cá nhân” interface |   **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI5**](#BI5) | | | |

* + - * 1. **post**

**use case diagram**

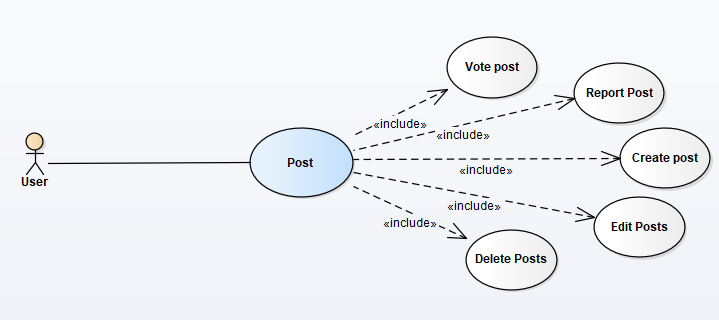
****

Image 15:User post function use case diagram

**use case specification**

Create post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-008 SPECIFICATION** | | | |
| **Use case ID** | US-UC-008 | **Use case version** | V1.0 |
| **Use case name** | Create post | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to create a post in forum. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can create a post in forum. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the plus icon at the bottom |  | | **2.** |  | Direct to “Thêm bài viết” interface | | **3.** | Input information (image, content,…) and press the plus icon it the top of the right corner |  | | **4.** |  | Direct to home interface with the added post at the top |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 3 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press the cross icon at the left of the top corner |  | | **2.** |  | Direct to home interface |   **Exception: N/A**  **Business Rules:** [**BI1**](#BI1) | | | |

Report post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-009 SPECIFICATION** | | | |
| **Use case ID** | US-UC-009 | **Use case version** | V1.0 |
| **Use case name** | Report post | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to report a post in forum. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can report a post in forum. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the three dot icon at the right of the top of the post |  | | **2.** |  | Show menu | | **3.** | Press “Báo cáo bài viết” |  | | **4.** |  | Show “Báo cáo” pop-up | | **5.** | Fill information to require field and press “Gửi” button |  | | **6.** |  | Show pop-up with the information about the report | | **7.** | Press “Xong” button |  | | **8.** |  | Close pop-up |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2) | | | |

Delete post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-010 SPECIFICATION** | | | |
| **Use case ID** | US-UC-010 | **Use case version** | V1.0 |
| **Use case name** | Delete posts | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to delete their created posts. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can delete their created posts. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the three dot icon at the right of the top of the post |  | | **2.** |  | Show menu | | **3.** | Press “Xóa bài viết” |  | | **4.** |  | Show confirm dialog | | **5.** | Press “Có” |  | | **6.** |  | Close dialog, show success message and that post disappear |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 5 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press “Không” |  | | **2.** |  | Close dialog |   **Exception: N/A**  **Business Rules: N/A** | | | |

Edit post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-011 SPECIFICATION** | | | |
| **Use case ID** | US-UC-011 | **Use case version** | V1.0 |
| **Use case name** | Edit posts | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to edit their created posts. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can edit their created posts. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the three dot icon at the right of the top of the post |  | | **2.** |  | Show menu | | **3.** | Press “Sửa bài viết” |  | | **4.** |  | Show “Sửa bài viết” interface | | **5.** | Edit content of post and press the save icon at the right of the top corner |  | | **6.** |  | Direct to home interface with this post at the top |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2) | | | |

Vote post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-012 SPECIFICATION** | | | |
| **Use case ID** | US-UC-012 | **Use case version** | V1.0 |
| **Use case name** | Vote posts | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to vote (up vote/down vote) a post in forum. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can vote (up vote/down vote) a post in forum. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press like icon |  | | **2.** |  | Like icon change color and send notification to post owner |   **Alternative Flows:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press dislike icon |  | | **2.** |  | Dislike icon change color and send notification to post owner |   **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **comment**

**use case diagram**

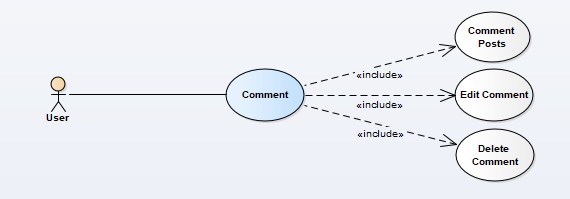


Image 16:User comment function use case diagram

**use case specification**

Comment post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-017 SPECIFICATION** | | | |
| **Use case ID** | US-UC-017 | **Use case version** | V1.0 |
| **Use case name** | Comment post | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to comment a post. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | * User can comment a post. * All user can view this post. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press speech bubble icon |  | | **2.** |  | Direct to “Bình luận” interface | | **3.** | Input comment content and press the paper air-plane icon |  | | **4.** |  | Show that comment in “Bình luận” interface and focus that comment, send notification to post owner |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 3 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press cross icon at the left of the top corner |  | | **2.** |  | Direct to home interface |   **Exception: N/A**  **Business Rules:** [**BI1**](#BI1) | | | |

Delete comment

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-018 SPECIFICATION** | | | |
| **Use case ID** | US-UC-018 | **Use case version** | V1.0 |
| **Use case name** | Delete comment | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to delete their comment. | | |
| **Pre-condition** | * User has logged in PH application. * User commented at least one post. | | |
| **Post-condition** | * User can comment a post. * All user can view this post. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press and hold the comment |  | | **2.** |  | Show pop-up menu | | **3.** | Press “Xóa bình luận” |  | | **4.** |  | Show confirm dialog | | **5.** | Press “Có” button |  | | **6.** |  | Close dialog and that comment disappear |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 5 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press “Không” button |  | | **2.** |  | Close dialog |   **Exception: N/A**  **Business Rules: N/A** | | | |

Edit comment

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-019 SPECIFICATION** | | | |
| **Use case ID** | US-UC-019 | **Use case version** | V1.0 |
| **Use case name** | Edit comment | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to edit their comment. | | |
| **Pre-condition** | * User has logged in PH application. * User commented at least one post. | | |
| **Post-condition** | User can edit their post. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press and hold the comment |  | | **2.** |  | Show pop-up menu | | **3.** | Press “Sửa bình luận” |  | | **4.** |  | Change comment to editable text field | | **5.** | Input content and press “Cập nhật” button |  | | **6.** |  | Change this comment to normal and focus to this comment |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 5 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press “Hủy” button |  | | **2.** |  | Change this comment to normal and focus to this comment |   **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **rating**

**use case diagram**

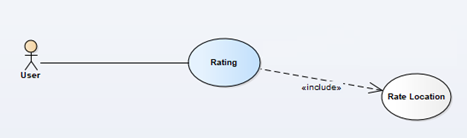
****

Image 17:User rating function use case diagram

**use case specification**

Rate location

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-021 SPECIFICATION** | | | |
| **Use case ID** | US-UC-021 | **Use case version** | V1.0 |
| **Use case name** | Rate location | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to rate a location. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can rate a location. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do UG-UC-004 |  | | **2.** |  | Direct to “Chi tiết địa điểm” interface | | **3.** | Press “Viêt đánh giá” button |  | | **4.** |  | Direct to “Viết đánh giá” interface | | **5.** | Input content, choose number of star and press “Gửi” button |  | | **6.** |  | Direct to “Chi tiết địa điểm” interface and show success message |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 5 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press “Hủy” button |  | | **2.** |  | Direct to “Chi tiết địa điểm” interface |   **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **chat**

**use case diagram**

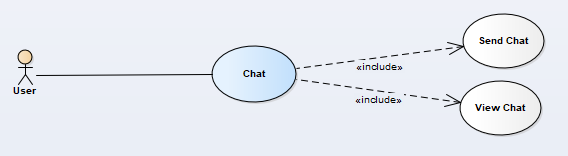
****

Image 18:User chat function use case diagram

**use case specification**

Send chat

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-022 SPECIFICATION** | | | |
| **Use case ID** | US-UC-022 | **Use case version** | V1.0 |
| **Use case name** | Send chat | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to send message to another user. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can send message to another user. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click the collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press speech bubble icon |  | | **4.** |  | Show “Tin nhắn” interface | | **5.** | Press one communication |  | | **6.** |  | Show “Tin nhắn” pop-up | | **7.** | Type message and press “Gửi” button |  | | **8.** |  | The message appear in pop-up |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

View chat

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-023 SPECIFICATION** | | | |
| **Use case ID** | US-UC-023 | **Use case version** | V1.0 |
| **Use case name** | View chat | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to view message from another user. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can view message from another user. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Click the collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press speech bubble icon |  | | **4.** |  | Show “Tin nhắn” interface |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **notification**

**use case diagram**

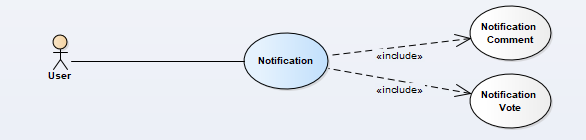


Image 19:User notification function use case diagram

**use case specification**

Notification comment

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-024 SPECIFICATION** | | | |
| **Use case ID** | US-UC-024 | **Use case version** | V1.0 |
| **Use case name** | Notification comment | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to receive notice when someone comment their posts. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can receive notice when someone comment their posts. | | |
| **Trigger** |  | | |
| **Main flow: N/A**  **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Notification vote

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-025 SPECIFICATION** | | | |
| **Use case ID** | US-UC-025 | **Use case version** | V1.0 |
| **Use case name** | Notification vote | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to receive notice when someone vote their posts. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can receive notice when someone vote their posts. | | |
| **Trigger** |  | | |
| **Main flow: N/A**  **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **authentication**

**use case diagram**

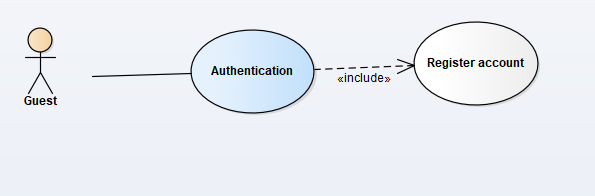


Image 20:User authentication function use case diagram

**use case specification**

Register account

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE GU-UC-001 SPECIFICATION** | | | |
| **Use case ID** | GU-UC-001 | **Use case version** | V1.0 |
| **Use case name** | Register account | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | Guest | **Secondary actor** | N/A |
| **Description** | Allows guest to register an account in PH system. | | |
| **Pre-condition** | Guest open PH application. | | |
| **Post-condition** | Guest can register an account in PH system with their Facebook account. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press “Đăng ký” |  | | **2.** |  | Show “Đăng nhập bằng Facebook” interface | | **3.** | Fill all the require field and press “Đăng nhập” button |  | | **4.** |  | Direct to home interface |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI1**](#BI1)**,** [**BI2**](#BI2)**,** [**BI5**](#BI5) | | | |

* + - * 1. **view post**

**use case diagram**

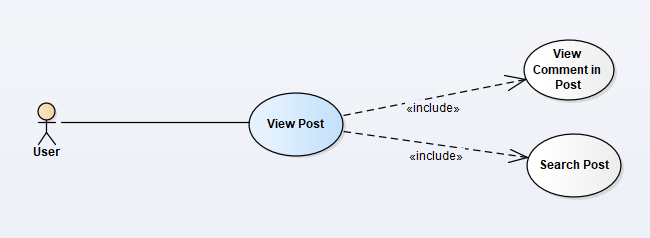
****

Image 21:User view post function use case diagram

**use case specification**

View comments in post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE GU-UC-001 SPECIFICATION** | | | |
| **Use case ID** | UG-UC-001 | **Use case version** | V1.0 |
| **Use case name** | View comments in post | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User, Guest | **Secondary actor** | N/A |
| **Description** | Allows actor to view comment in a post. | | |
| **Pre-condition** | Actor open PH application. | | |
| **Post-condition** | Actor can view comment in a post. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the speech bubble at the left bottom corner of a post |  | | **2.** |  | Show 20 newest comment | | **3.** | Press “Xem thêm bình luận trước” |  | | **4.** |  | Direct to comment interface with more than 20 comment |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Search post

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE GU-UC-002 SPECIFICATION** | | | |
| **Use case ID** | UG-UC-002 | **Use case version** | V1.0 |
| **Use case name** | Search post | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User, Guest | **Secondary actor** | N/A |
| **Description** | Allows actor to search all post in PH system with default data. | | |
| **Pre-condition** | Actor open PH application. | | |
| **Post-condition** | Actor can search all post in PH system with default data. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the magnifying glass icon at the bottom |  | | **2.** |  | Show search post interface | | **3.** | Input data |  | | **4.** |  | Show all post fit with input data |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2) | | | |

* + - * 1. **location**

**use case diagram**

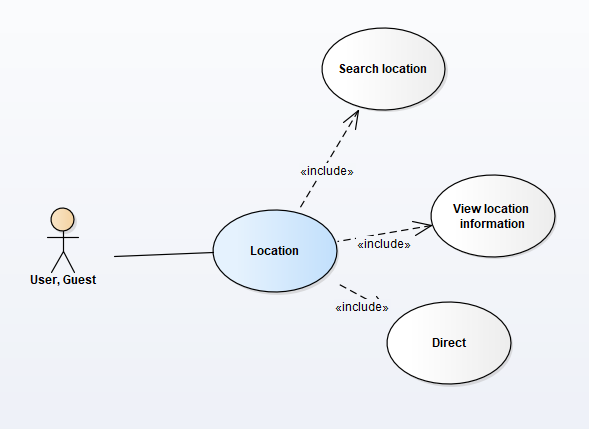


Image 22:User & Guest location function use case diagram

**use case specification**

Search location

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE GU-UC-003 SPECIFICATION** | | | |
| **Use case ID** | UG-UC-003 | **Use case version** | V1.0 |
| **Use case name** | Search location | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User, Guest | **Secondary actor** | N/A |
| **Description** | Allows actor to search all location in PH system with default data. | | |
| **Pre-condition** | Actor open PH application. | | |
| **Post-condition** | Actor can search all location in PH system with default data. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press the earth icon |  | | **4.** |  | Show list location in location view interface | | **5.** | Press search bar |  | | **6.** |  | Show search interface | | **7.** | Input data and press magnifying glass icon |  | | **8.** |  | Show list location fit with input data |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules:** [**BI2**](#BI2)**,** [**BI1**](#BI1) | | | |

View location information

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE GU-UC-004 SPECIFICATION** | | | |
| **Use case ID** | UG-UC-004 | **Use case version** | V1.0 |
| **Use case name** | View location information | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User, Guest | **Secondary actor** | N/A |
| **Description** | Allows actor to view information of location in system. | | |
| **Pre-condition** | Actor open PH application. | | |
| **Post-condition** | Actor can view information of location in system. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press the earth icon |  | | **4.** |  | Show list location in location view interface | | **5.** | Press one location |  | | **6.** |  | Show location detail interface |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Direct

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE GU-UC-005 SPECIFICATION** | | | |
| **Use case ID** | UG-UC-005 | **Use case version** | V1.0 |
| **Use case name** | Direct | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User, Guest | **Secondary actor** | N/A |
| **Description** | Allows actor to know the way to reach the location from current location. | | |
| **Pre-condition** | Actor open PH application. | | |
| **Post-condition** | Actor can reach the location with shortest way. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Do UG-UC-004 |  | | **2.** |  | Show location detail interface | | **3.** | Press icon at the top of the right corner |  | | **4.** |  | Show the shortest way from the actor to the location |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

* + - * 1. **pet**

**use case diagram**

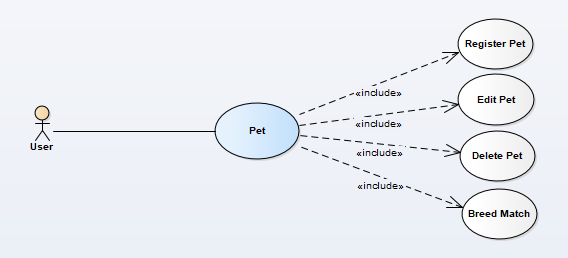


Image 23: User pet function use case diagram

**use case specification**

Breed match

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-013 SPECIFICATION** | | | |
| **Use case ID** | US-UC-013 | **Use case version** | V1.0 |
| **Use case name** | Breed match | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to find another’s pet fit with registered information. | | |
| **Pre-condition** | * User has logged in PH application. * User has at least on registered pet. | | |
| **Post-condition** | User can find another’s pet fit with registered information. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press heart icon |  | | **4.** |  | Show “Ghép đôi” interface | | **5.** | Choose a pet a press “Ghép” button |  | | **6.** |  | Show pet fit with data |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Delete pet

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-014 SPECIFICATION** | | | |
| **Use case ID** | US-UC-014 | **Use case version** | V1.0 |
| **Use case name** | Delete pet | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to delete their registered pet. | | |
| **Pre-condition** | * User has logged in PH application. * User has at least on registered pet. | | |
| **Post-condition** | User can delete their registered pet. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press heart icon |  | | **4.** |  | Show “Ghép đôi” interface | | **5.** | Choose “Thú nuôi của tôi” button |  | | **6.** |  | Show list of pet | | **7.** | Press the trash bin icon |  | | **8.** |  | Show confirm dialog | | **9.** | Press “Có” button |  | | **10.** |  | Close dialog, show success message and this pet disappear |   **Alternative Flows: N/A**  **Extension Flows:**   |  |  |  | | --- | --- | --- | | **ET1** | At step 9 | | | **No.** | **Actor events** | **System respond** | | **1.** | Press “Không” button |  | | **2.** |  | Close dialog |   **Exception: N/A**  **Business Rules: N/A** | | | |

Edit pet

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-015 SPECIFICATION** | | | |
| **Use case ID** | US-UC-015 | **Use case version** | V1.0 |
| **Use case name** | Edit pet | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | Medium |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to edit their registered pet. | | |
| **Pre-condition** | * User has logged in PH application. * User has at least on registered pet. | | |
| **Post-condition** | User can edit their registered pet. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press heart icon |  | | **4.** |  | Show “Ghép đôi” interface | | **5.** | Choose “Thú nuôi của tôi” button |  | | **6.** |  | Show list of pet | | **7.** | Press the edit icon |  | | **8.** |  | Show edit pet interface | | **9.** | Edit data and press “Lưu” button |  | | **10.** |  | Show list of pet |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

Register pet

|  |  |  |  |
| --- | --- | --- | --- |
| **USECASE US-UC-016 SPECIFICATION** | | | |
| **Use case ID** | US-UC-016 | **Use case version** | V1.0 |
| **Use case name** | Register pet | | |
| **Author** | Hoang Phan Trung | | |
| **Date** | 03/10/2018 | **Priority** | High |
| **Primary actor** | User | **Secondary actor** | N/A |
| **Description** | Allows user to register a pet. | | |
| **Pre-condition** | User has logged in PH application. | | |
| **Post-condition** | User can register a pet. | | |
| **Trigger** |  | | |
| **Main flow:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor events** | **System respond** | | **1.** | Press the collapsible button |  | | **2.** |  | Show drawer | | **3.** | Press heart icon |  | | **4.** |  | Show “Ghép đôi” interface | | **5.** | Choose “Thú nuôi của tôi” button |  | | **6.** |  | Show list of pet | | **7.** | Press the plus icon |  | | **8.** |  | Show add pet interface | | **9.** | Input data and press the plus icon at the top of the right |  | | **10.** |  | Show list of pet with this pet at the top |   **Alternative Flows: N/A**  **Extension Flows: N/A**  **Exception: N/A**  **Business Rules: N/A** | | | |

### Non-Functional requirements

* + - 1. **usability**
* GUI:
* All the text has to be shown in Vietnamese.
* The mobile application’s interface and website administrator must be clear and simple.
* The mobile application and website administrator are friendly and easy to use.
* Usability for end users:
* The end user’s interface has to friendly, easy to use on each function.
* Search capabilities fast and exactly.
* Usability for administrator:
* Interface has to friendly, easy to use on each function.
* Installation:
* The system is easy to deploy.
* The document for deploying must be included. It describes detail steps to install the system for end user and administrator.
  + - 1. **availability**
* The system active every day.
  + - 1. **security**
* User Password is invisible for the system administration.
* When user password is saved in database, it must be encrypted.
* The system must provide methods for users to recovery their password when they forgot their password.
* The server encountered an internal.
  + - 1. **maintainability**
* Coding standards and naming conventions
* Output of the project must include coding standards and naming conventions documentations. Implementation code must be easy to maintain.
* All code must be clearly commented, including class, method documentations.
* If some components are reused, the documentations of those components must also be included.
* Design
* The system architecture should be designed using self-contained components that may readily be changed: Therefore, the system architecture should be flexible to plug addition modules, packages.
* Exception handler: The system should include an exception handler for all of the unexpected exception.
  + - 1. **performance**
* The mobile application work well in Android 4.1 and IOS 12.
* The web application work well with Chrome 35.

## Entity relationship diagram

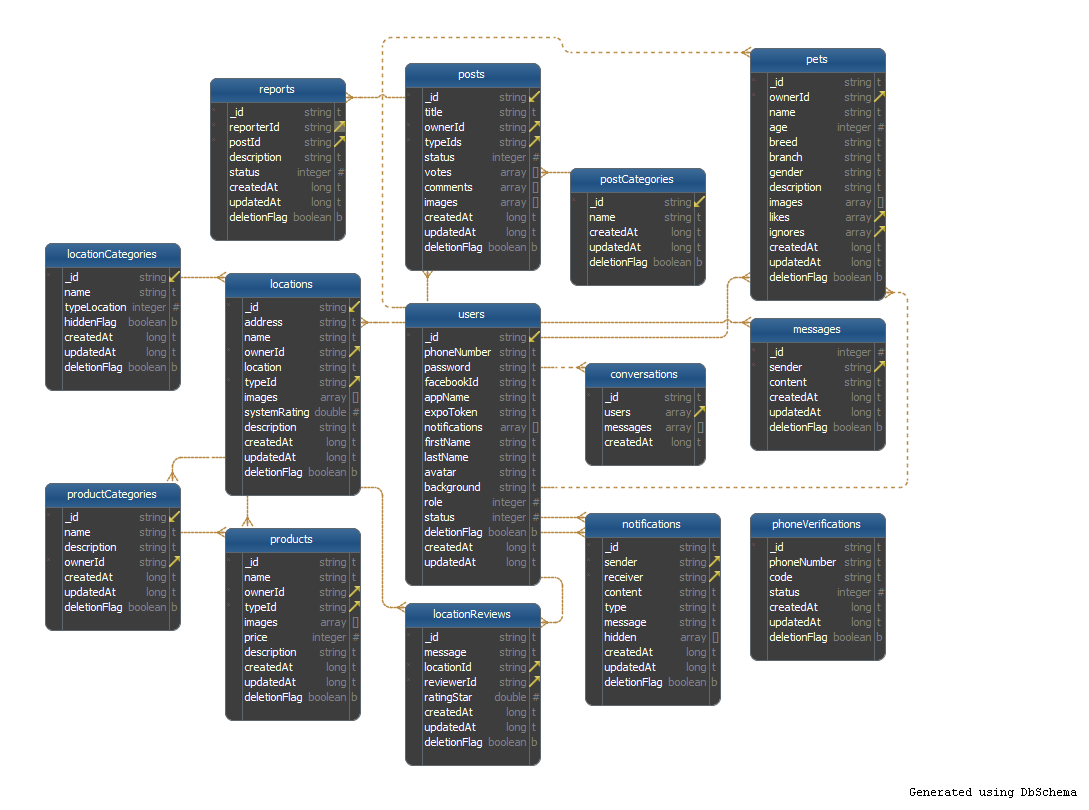


Figure 9: Entity Relationship Diagram

# Chapter 4: Software architecture design



## Purpose

This purpose of this chapter shows the software architecture system design using a top-down approach, from the high-level design to detail design description for each section.

## System architecture design

### Choice of system architecture design

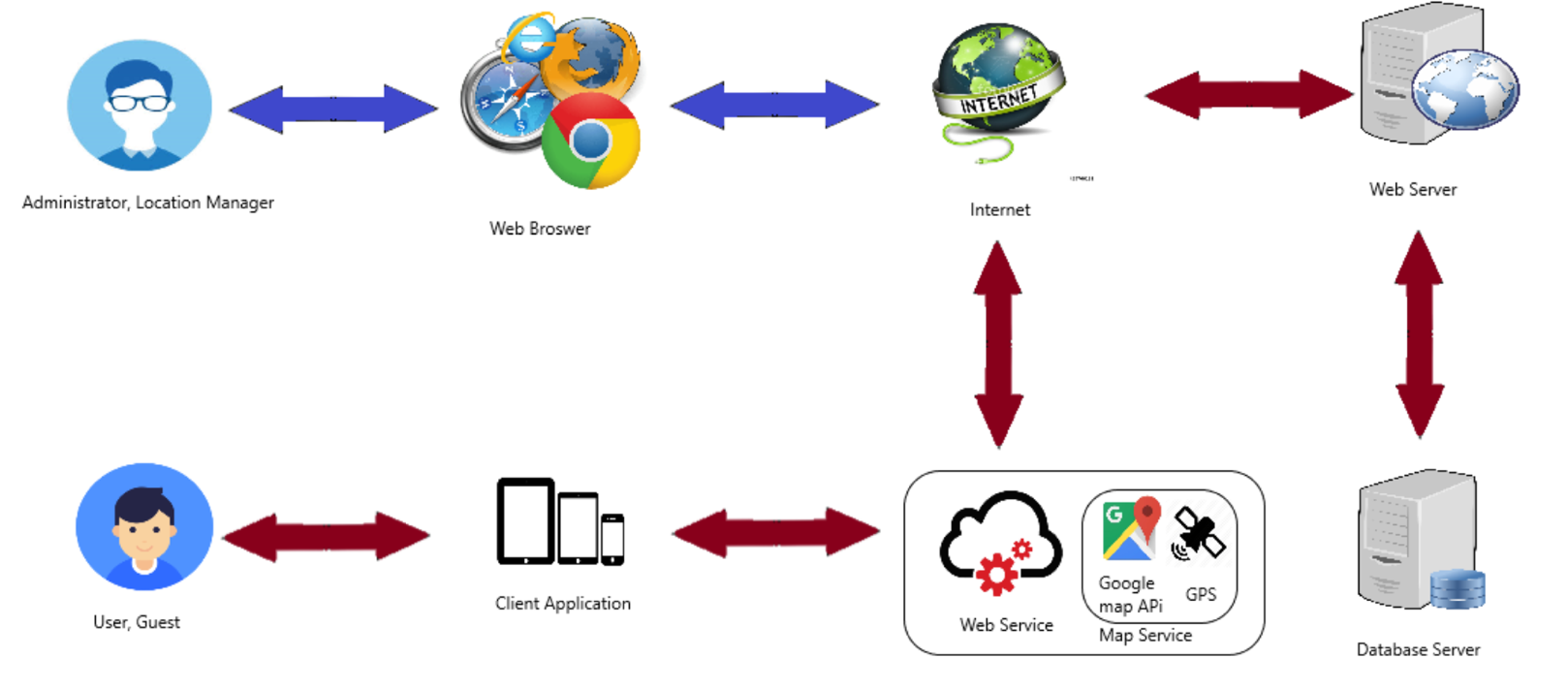
****

Figure 10: System Architectural Design

* + - 1. **services**

In PH, the service used contain: RESTful Web Services with Node.js , Expressjs and Google Map API web service.

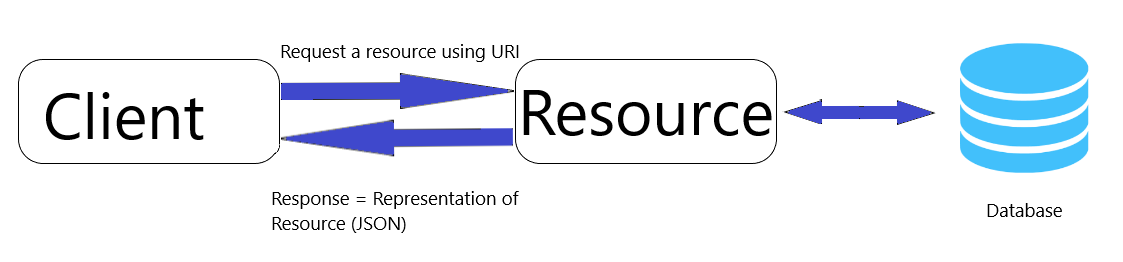


Figure 11:RESTful web service architecture

**What is REST?**

REST stands for Representational State Transfer is an architectural style which is based on web-standards and the HTTP protocol. REST was first described by Roy Fielding in 2000.

In REST based architecture everything is a resource. A resource is accessed via a common interface based on the HTTP standard methods.

In a REST based architecture typically have a REST server which provides access to the resources and a REST client which accesses and modifies the REST resources.

Every resource should support the HTTP common operations. Resources are identified by global IDs (which are typically URIs).

REST allows that resources have different representations, e.g., text, XML, JSON etc.

The REST client can ask for a specific representation via the HTTP protocol (content negotiation).

**HTTP methods:**

The PUT, GET, POST and DELETE methods are typical used in REST based architectures.

The following table gives an explanation of these operations.

* **GET**: It defines a reading access of the resource without side-effects. This operation is idempotent i.e. They can be applied multiple times without changing the result
* **PUT**:  It creates a new resource. It must also be idempotent.
* **DELETE:** It removes the resources. The operations are idempotent i.e. they can get repeated without leading to different results.
* **POST**: It updates an existing resource or creates a new resource.

**RESTful web services**

A RESTful web services are based on HTTP methods and the concept of REST. A RESTful web service typically defines the base URI for the services, the supported MIME-types (XML, text, JSON, user-defined ...) and the set of operations (POST, GET, PUT, DELETE) which are supported.

**Sample:**

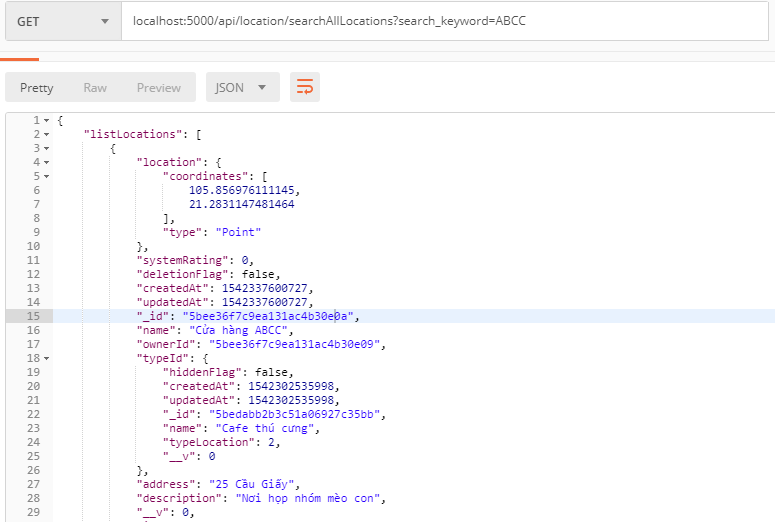


Figure 12: JSON return when call RESTful web service

* + - 1. **Client – mobile application**

Application running on IOS Phone and will interact with the web service via normal protocols (HTTP, HTTPS…).

* + - 1. **web manager**

This is website for administrator and location manager, allow administrator management database through web interface easier and allow location manager to manage their added location, service. Website does intervene directly in the database and it doesn’t through web service. Website is developed by ReactJS.

* + - 1. **database**

Data on server is the central data storage of the system. It may be running the same machine or in a separate server to the Application Server. Database on server using MongoDB 4.0.



### Discussion of alternative design

**Model – Controller – View**

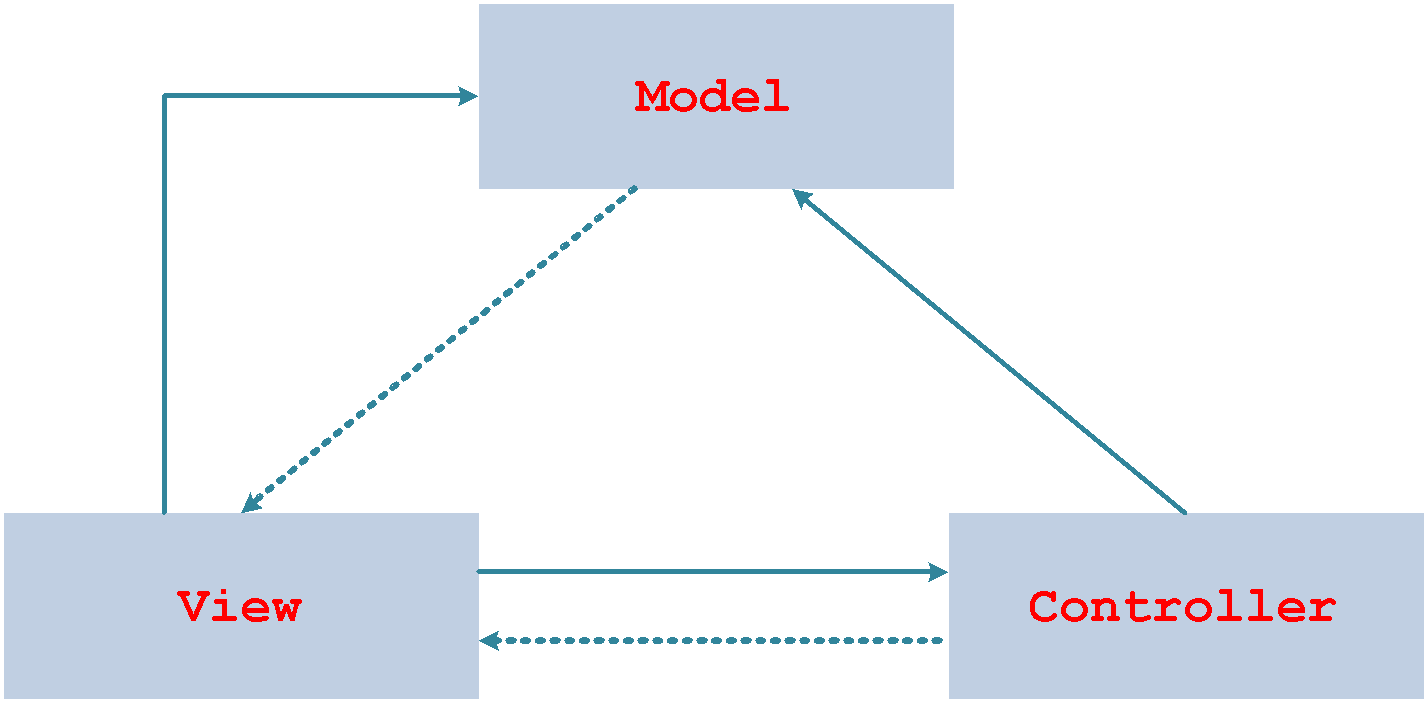


Figure 13:MVC architecture

**What is MVC model?**

MVC is a software pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. The central component, the model, consists of application data, business rules, logic, and functions. A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants. The third part, the controller, accepts input and converts it to commands for the model or view.

* A **controller** can send commands to the model to update the model's state (e.g., editing a document). It can also send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document).
* A **model** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them.
* A **view** requests information from the model that it needs for generating an output representation to the user.

**Advantages and Disadvantages**

* **Advantages:** the MVC model demonstrates professionalism in programming and design analysis. It is divided into independent components to help develop applications faster, simpler, easier upgrades and maintenance.
* **Disadvantages:** for small projects applying MVC model caused cumbersome, time consuming in development process. Time consuming to transits data between components.

**Why do we choose MVC model?**

* The first thing is that it fits the pattern of PH: receive request from the user, call appropriate action, retrieve data and display on view for the user.
* Easily manage the complexity of application by dividing the application into three components: model, view and controller.
* Better support for test driven development.
* PH is not complete system, now. We built the system that towards extensibility and maintainability in the future.
* It is good support for application built by project team has many developers and designers but still managed application features

### Description of system interfaces

SRS document also refers very detail of PH interface. In this description, we will generalize 3 main areas of PH interface, which will be detail so far in this document:

* Web services: to communicate between server and mobile application.
* Web administrator: is the administrator web application for admin to manage category, manage location, manage user, …. It also for location manager to manage their location and the service attached.
* Mobile Application: is client application interacting with user and guest.

### Architecture presentation

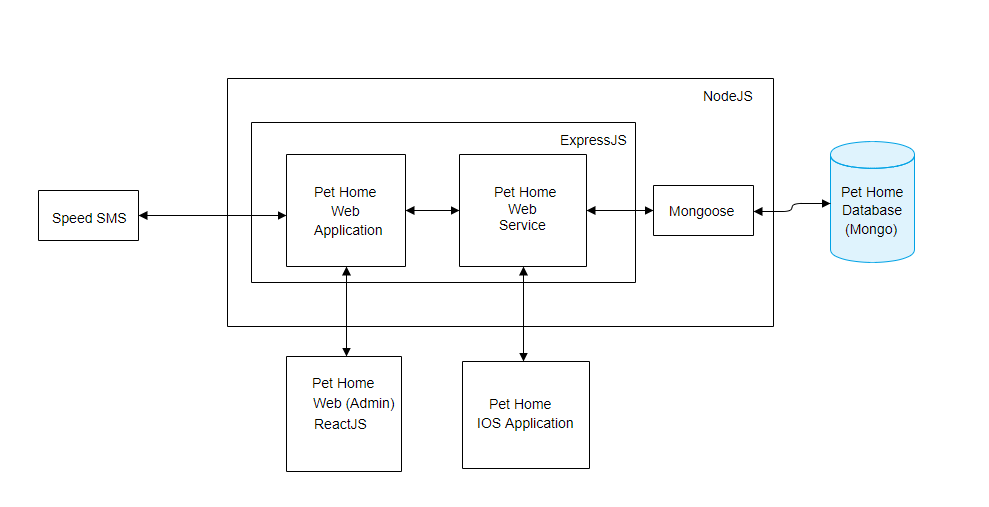


Figure :Architecture presentation

### Technology description

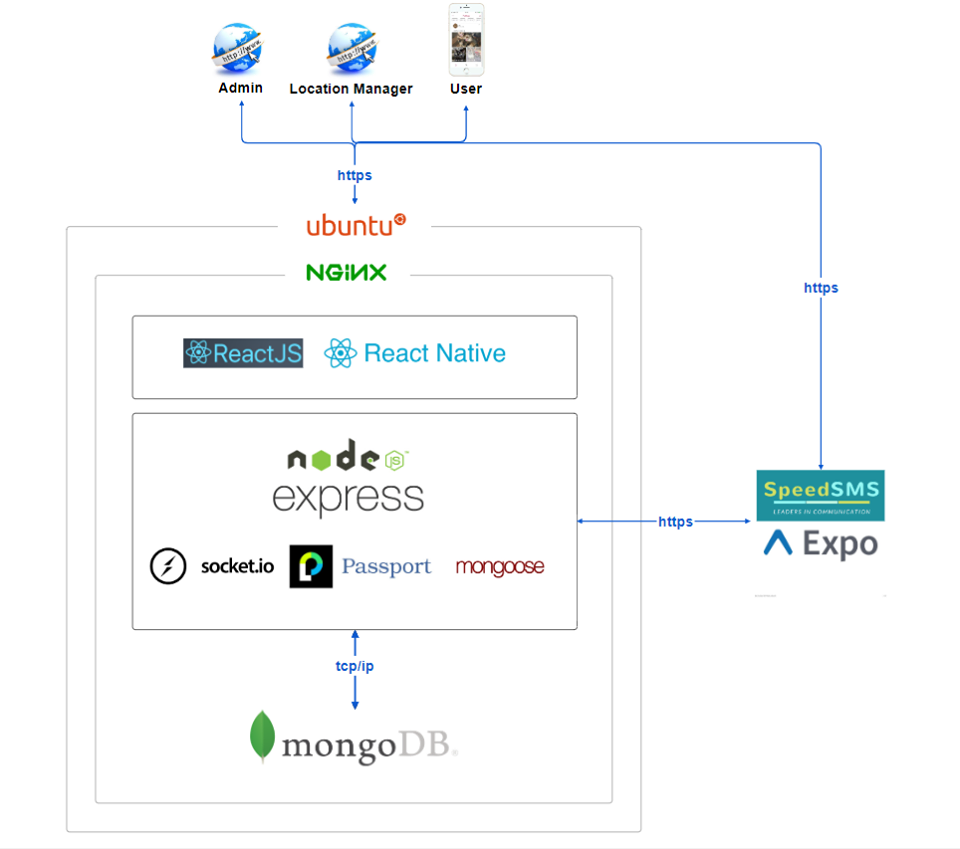


Figure : Technology description

## Component diagram

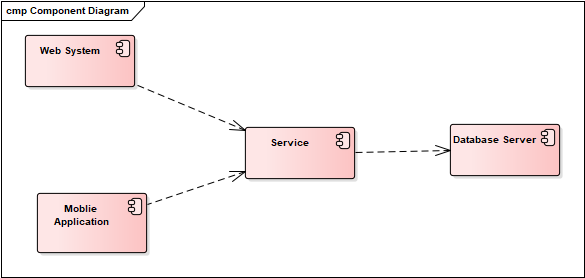
****

Figure 16: Component Diagram

## Detailed description diagram



### CRC card (class – responsibility - collaborators)

Web login

|  |  |
| --- | --- |
| Login View | |
| * Knows user name * Knows password * Know Login.js * Handle click event of “Đăng nhập” button | * Login.js |

|  |  |
| --- | --- |
| Login.js | |
| * Valid phone number with format * Send request and data to AuthController.js * Know AuthController.js | * AuthController.js |

|  |  |
| --- | --- |
| AuthController.js | |
| * Handle the login request (compare the data with database) * Know HomePage * Know LoginPage | * AuthController.js * HomePage * LoginPage |

Add new category

|  |  |
| --- | --- |
| Add New Category Page | |
| * Knows category name * Handle click event of “Thêm mới” button * Knows CategoryController.js | * CategoryController.js |

|  |  |
| --- | --- |
| CategoryController.js | |
| * Handle the add new request (create a new category with inputted data) * Knows View All Category Page * Knows Category.js | * Category.js * View All Category Page |

View location list

|  |  |
| --- | --- |
| Location Page | |
| * Knows LocationController.js * Handle start event of user | * LocationController.js |

|  |  |
| --- | --- |
| LocationController.js | |
| * Handle the get data request (return all location in system) * Knows Location Page * Knows Location.js | * Location.js |

Administrator view report

|  |  |
| --- | --- |
| View Report Page | |
| * Know AdminController .js * Handle start event of user | * AdminController .js |

|  |  |
| --- | --- |
| AdminController .js | |
| * Handle the get data request (return all reports) * Knows View Report Page | * View Report Page * Report.js |

Administrator ban or unban user

|  |  |
| --- | --- |
| View User Page | |
| * Handle click event of “Chan” button * Handle click event of “Bo Chan” button * Knows AdminController .js | * AdminController .js |

|  |  |
| --- | --- |
| AdminController.js | |
| * Handle the change status request (create a new user and update into database) * Know View User Page | * View User Page |

Change password of web application

|  |  |
| --- | --- |
| Change Password Page | |
| * Knows old password * Knows new password * Knows re-enter password * Knows AuthController.js * Handle click event of “Thay doi” button | * AuthController.js |

|  |  |
| --- | --- |
| AuthController.js | |
| * Handle the update request (update data into database) * Knows Change Password Page | * Change Password Page |

Edit location

|  |  |
| --- | --- |
| Edit Location Page | |
| * Knows location images, location name, address, type * Handle click event of “Luu” button | * LocationController.js |

|  |  |
| --- | --- |
| LocationController.js | |
| * Handle the update request * Knows Edit Location Page | * Edit Location Page |

Forgot password web application

|  |  |
| --- | --- |
| Login Page | |
| * Handle click event of “Quen mat khau” button * Knows Forgot Page | * Forgot Page |

|  |  |
| --- | --- |
| Forgot Page | |
| * Knows phone number * Handle click event of “Gui ma xac nhan” button * Knows Login Page * Knows PhoneController.js | * Login Page * PhoneController.js |

|  |  |
| --- | --- |
| PhoneController.js | |
| * Handle the request * Call function send verification code * Valid phone number * Knows Verify Code Page | * Verify Code Page |

|  |  |
| --- | --- |
| Verify Code Page | |
| * Knows verification code * Handle click event of “Xac nhan” button * Knows AuthController.js | * AuthController.js |

|  |  |
| --- | --- |
| AuthController.js | |
| * Handle the request * Compare verification code * Knows Change Password Page | * Change Password Page |

|  |  |
| --- | --- |
| Change Password Page | |
| * Knows old password * Knows new password * Knows re-enter password * Knows AuthController.js * Handle click event of “Thay doi” button | * AuthController.js |

|  |  |
| --- | --- |
| AuthController.js | |
| * Handle the update request (update data into database) * Knows Change Password Page | * Change Password Page |

Hide or show location

|  |  |
| --- | --- |
| Location Page | |
| * Handle click event of “Hien” button * Handle click event of “An” button * Knows LocationController.js | * LocationController.js |

|  |  |
| --- | --- |
| LocationController.js | |
| * Handle the update request * Knows Location Page | * Location Page |

Web application register

|  |  |
| --- | --- |
| Register Page | |
| * Knows phone number, password, location name * Handle click event of “Dang ky” button * Knows AuthController.js | * AuthController.js |

|  |  |
| --- | --- |
| AuthController.js | |
| * Handle the request * Knows Register Page * Valid the input data * Create new User.js to insert to database | * Register Page * User.js |

Show or hide category

|  |  |
| --- | --- |
| View Category Page | |
| * Knows CategoryController.js * Handle click event of “Ẩn” button * Handle click event of “Hiện” button | * CategoryController.js |

|  |  |
| --- | --- |
| CategoryController.js | |
| * Handle the update request * Knows View Category Page | * View Category Page |

View category list

|  |  |
| --- | --- |
| View Category Page | |
| * Knows CategoryController.js * Show all category | * CategoryController.js |

|  |  |
| --- | --- |
| CategoryController.js | |
| * Handle the update request * Knows View Category Page | * View Category Page |

View location information

|  |  |
| --- | --- |
| Location Detail Page | |
| * Knows LocationController.js * Handle open – event of user | * LocationController.js |

|  |  |
| --- | --- |
| LocationController.js | |
| * Knows Location Detail Page * Handle the query request | * Location Detail Page |

View location list

|  |  |
| --- | --- |
| Location Page | |
| * Knows LocationController.js * Handle open – event of user | * LocationController.js |

|  |  |
| --- | --- |
| LocationController.js | |
| * Knows Location Page * Handle the query request | * Location Page |

View user list

|  |  |
| --- | --- |
| User Page | |
| * Knows AdminController.js * Handle the open event of user | * AdminController.js |

|  |  |
| --- | --- |
| AdminController.js | |
| * Handle the query request * Knows User Page | * User Page |

Create a pet

|  |  |
| --- | --- |
| Add Pet Interface | |
| * Knows PetController.js * Handle press event of “Thêm” button * Knows pet name, species, gender, breed, age, images | * PetController.js |

|  |  |
| --- | --- |
| PetController.js | |
| * Knows Add Pet Interface * Handle the create request * Create a Pet.js to insert into database | * Add Pet Interface * Pet.js |

Create a post

|  |  |
| --- | --- |
| New Post Interface | |
| * Knows PostController.js * Knows content * Knows images * Handle the press event of save button | * PostController.js |

|  |  |
| --- | --- |
| PostController.js | |
| * Knows New Post Interface * Handle the add request * Create a Post.js to insert into database * Knows Post.js | * New Post Interface * Post.js |

Guest register

|  |  |
| --- | --- |
| Register Interface | |
| * Knows UserController.js * Knows phone number, display name, password * Handle the press event of “Đăng ký” button | * UserController.js |

|  |  |
| --- | --- |
| UserController.js | |
| * Handle the insert request * Knows Register Interface * Create a User.js to insert into database | * Register Interface * User.js |

### Component

* + - 1. **Class diagram**

Web Login

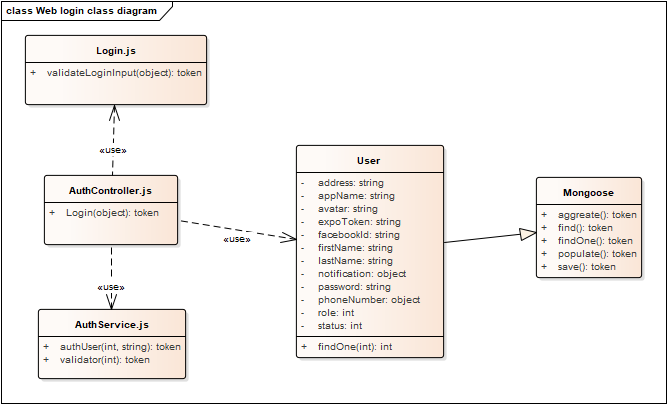


Figure : Web login use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Login | | |
| Description | Service class | | |
| Source file | Login.js | | |
| Physical address | validation\login.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | validateLoginInput | Data: object | Object |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AuthController | | |
| Description | Controller class | | |
| Source file | AuthController.js | | |
| Physical address | controllers\AuthController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | Login | Data: object | Object |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AuthService | | |
| Description | Service class | | |
| Source file | AuthService.js | | |
| Physical address | services\AuthService.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | authUser | phoneNumber: int  avatar: string | Object |
|  | Validator | phoneNumber: Int | Boolean |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | User | | |
| Description | Model class | | |
| Source file | User.js | | |
| Physical address | models\User.js | | |
| Attribute | Name | | Data type |
|  | Address | | String |
|  | appName | | String |
|  | Avatar | | String |
|  | expoToken | | String |
|  | facebookId | | String |
|  | frstName | | String |
|  | lastName | | String |
|  | Notification | | Object |
|  | Password | | String |
|  | phoneNumber | | Object |
|  | Role | | Int |
|  | Status | | Int |
| Methods | Name | Parmeter | Return type |
|  | findOne | phoneNumber: int | Token |

View user list

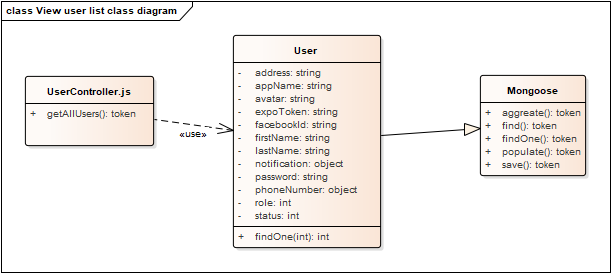


Figure :View user list use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | UserController | | |
| Description | Controller class | | |
| Source file | UserController.js | | |
| Physical address | controllers\UserController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | getAllUser |  | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | User | | |
| Description | Model class | | |
| Source file | User.js | | |
| Physical address | models\User.js | | |
| Attribute | Name | | Data type |
|  | Address | | String |
|  | appName | | String |
|  | Avatar | | String |
|  | expoToken | | String |
|  | facebookId | | String |
|  | frstName | | String |
|  | lastName | | String |
|  | Notification | | Object |
|  | Password | | String |
|  | phoneNumber | | Object |
|  | Role | | Int |
|  | Status | | Int |
| Methods | Name | Parmeter | Return type |
|  | findOne | phoneNumber: int | Token |

View location list

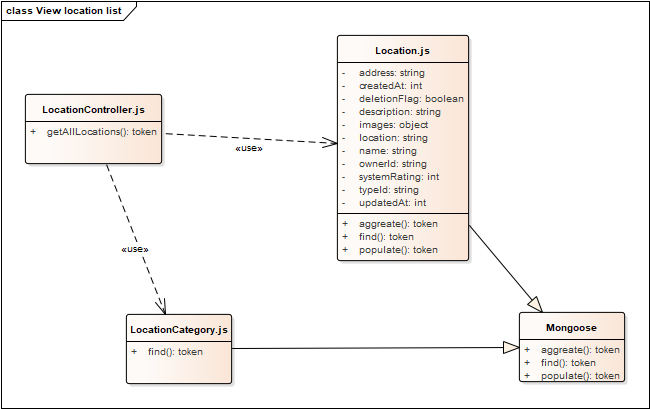


Figure :View location list use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LocationController | | |
| Description | Controller class | | |
| Source file | LocationController.js | | |
| Physical address | controllers\ LocationController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | getAllLocation |  | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Location | | |
| Description | Model class | | |
| Source file | Location.js | | |
| Physical address | models\Location.js | | |
| Attribute | Name | | Data type |
|  | Address | | String |
|  | createdAt | | Int |
|  | deletionFlag | | Boolean |
|  | Description | | String |
|  | Images | | Object |
|  | Location | | String |
|  | Name | | String |
|  | ownerId | | String |
|  | systemRating | | Int |
|  | typeId | | Int |
|  | updatedAt | | Int |
| Methods | Name | Parmeter | Return type |
|  | Aggreate |  | Token |
|  | Find |  | Token |
|  | Populate |  | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LocationCategory | | |
| Description | Controller class | | |
| Source file | LocationCategory.js | | |
| Physical address | models\LocationCategory.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | Find |  | Token |

View location information

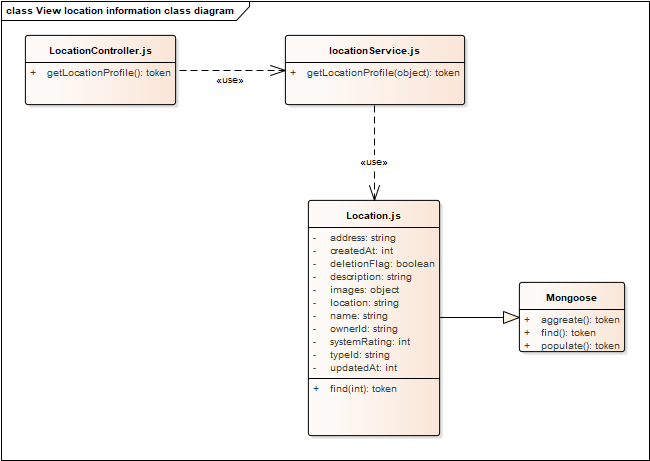


Figure :View location information use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LocationController | | |
| Description | Controller class | | |
| Source file | LocationController.js | | |
| Physical address | controllers\ LocationController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | getLocationProfile |  | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LocationService | | |
| Description | Controller class | | |
| Source file | LocationController.js | | |
| Physical address | controllers\ LocationController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | getLocationProfile | ownerId: object | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Location | | |
| Description | Model class | | |
| Source file | Location.js | | |
| Physical address | models\Location.js | | |
| Attribute | Name | | Data type |
|  | Address | | String |
|  | createdAt | | Int |
|  | deletionFlag | | Boolean |
|  | Description | | String |
|  | Images | | Object |
|  | Location | | String |
|  | Name | | String |
|  | ownerId | | String |
|  | systemRating | | Int |
|  | typeId | | Int |
|  | updatedAt | | Int |
| Methods | Name | Parmeter | Return type |
|  | Aggreate |  | Token |
|  | Find |  | Token |
|  | Populate |  | Token |

Get post category

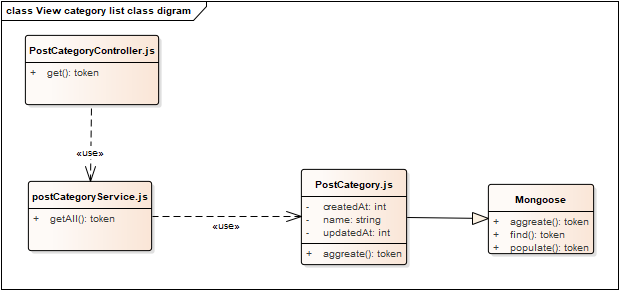


Figure :Get post category use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PostCategoryController | | |
| Description | Controller class | | |
| Source file | PostCategoryController.js | | |
| Physical address | controllers\PostCategoryController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | get |  | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | postCategoryService | | |
| Description | Service class | | |
| Source file | postCategoryService.js | | |
| Physical address | services\PostCategoryService.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | getAll |  | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PostCategory | | |
| Description | Model class | | |
| Source file | PostCategory.js | | |
| Physical address | models\PostCategory.js | | |
| Attribute | Name | | Data type |
|  | createdAt | | Int |
|  | Name | | String |
|  | updatedAt | | Int |
| Methods | Name | Parmeter | Return type |
|  | Aggregate |  | Token |

Location manager register

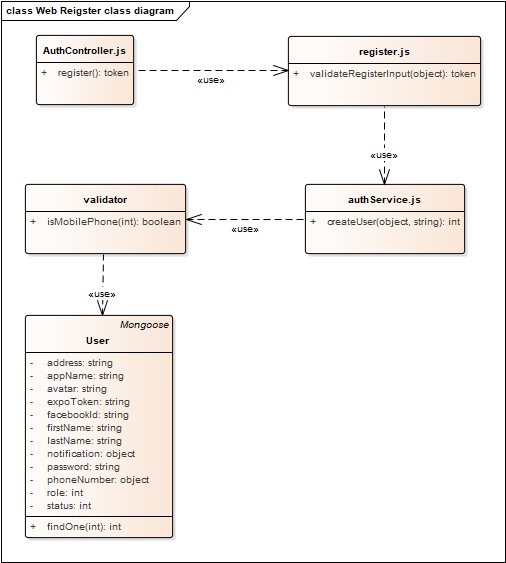


Figure :Locaiton manager register use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AuthController | | |
| Description | Controller class | | |
| Source file | AuthController.js | | |
| Physical address | controllers\ AuthController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | Register |  | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Register | | |
| Description | Model class | | |
| Source file | Register.js | | |
| Physical address | validation\register.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | validateRegisterInput | Data: object | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | authService | | |
| Description | Service class | | |
| Source file | authService.js | | |
| Physical address | services\AuthService.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | createUser | Data: object  Image: string | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | User | | |
| Description | Model class | | |
| Source file | User.js | | |
| Physical address | models\User.js | | |
| Attribute | Name | | Data type |
|  | Address | | String |
|  | appName | | String |
|  | Avatar | | String |
|  | expoToken | | String |
|  | facebookId | | String |
|  | frstName | | String |
|  | lastName | | String |
|  | Notification | | Object |
|  | Password | | String |
|  | phoneNumber | | Object |
|  | Role | | Int |
|  | Status | | Int |
| Methods | Name | Parmeter | Return type |
|  | findOne | phoneNumber: int | Token |

Web forgot password

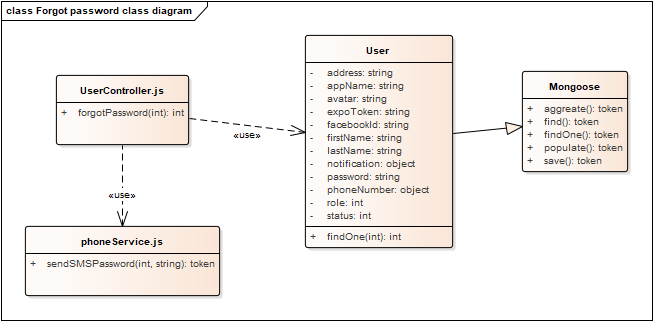


Figure : Web forgot password use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | UserController | | |
| Description | Controller class | | |
| Source file | UserController.js | | |
| Physical address | controllers\ UserController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | forgotPassword | phoneNumber: int | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | phoneService | | |
| Description | Service class | | |
| Source file | phoneService.js | | |
| Physical address | services\PhoneService.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | sendSMSPassword | phoneNumber: int  newPassword: String | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | User | | |
| Description | Model class | | |
| Source file | User.js | | |
| Physical address | models\User.js | | |
| Attribute | Name | | Data type |
|  | Address | | String |
|  | appName | | String |
|  | Avatar | | String |
|  | expoToken | | String |
|  | facebookId | | String |
|  | frstName | | String |
|  | lastName | | String |
|  | Notification | | Object |
|  | Password | | String |
|  | phoneNumber | | Object |
|  | Role | | Int |
|  | Status | | Int |
| Methods | Name | Parmeter | Return type |
|  | findOne | phoneNumber: int | Token |

Create a pet

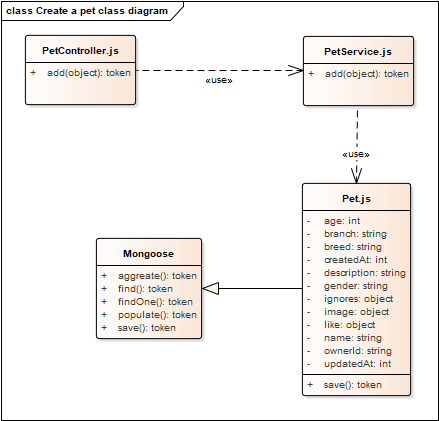


Figure : Create a pet use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PetController | | |
| Description | Controller class | | |
| Source file | PetController.js | | |
| Physical address | controllers\ PetController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | add | data: object | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PetService | | |
| Description | Service class | | |
| Source file | PetService.js | | |
| Physical address | services\PetService.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | save | data: object | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Pet | | |
| Description | Model class | | |
| Source file | Pet.js | | |
| Physical address | models\Pet.js | | |
| Attribute | Name | | Data type |
|  | Age | | Int |
|  | Branch | | String |
|  | Breed | | String |
|  | createdAt | | Int |
|  | Description | | String |
|  | Gender | | String |
|  | Ignores | | Object |
|  | Images | | Object |
|  | Likes | | Object |
|  | Name | | String |
|  | ownerId | | String |
|  | updatedAt | | Int |
| Methods | Name | Parmeter | Return type |
|  | add | data: object | Token |

Create a post

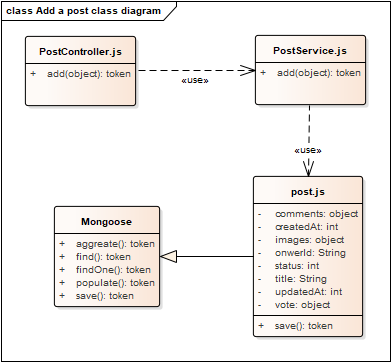


Figure : Create a post use case class diagram

Class specification

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PostController | | |
| Description | Controller class | | |
| Source file | PostController.js | | |
| Physical address | controllers\ PostController.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | add | data: object | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PostService | | |
| Description | Service class | | |
| Source file | PostService.js | | |
| Physical address | services\ PostService.js | | |
| Attribute | Name | | Data type |
|  |  | |  |
| Methods | Name | Parmeter | Return type |
|  | save | data: object | Token |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Post | | |
| Description | Model class | | |
| Source file | Post.js | | |
| Physical address | models\ Post.js | | |
| Attribute | Name | | Data type |
|  | Comments | | Object |
|  | createdAt | | Int |
|  | Images | | Object |
|  | onwerId | | String |
|  | Status | | Int |
|  | Title | | String |
|  | updatedAt | | Int |
|  | Vote | | Object |
| Methods | Name | Parmeter | Return type |
|  | save |  | Token |

## Sequence diagram

Web Login

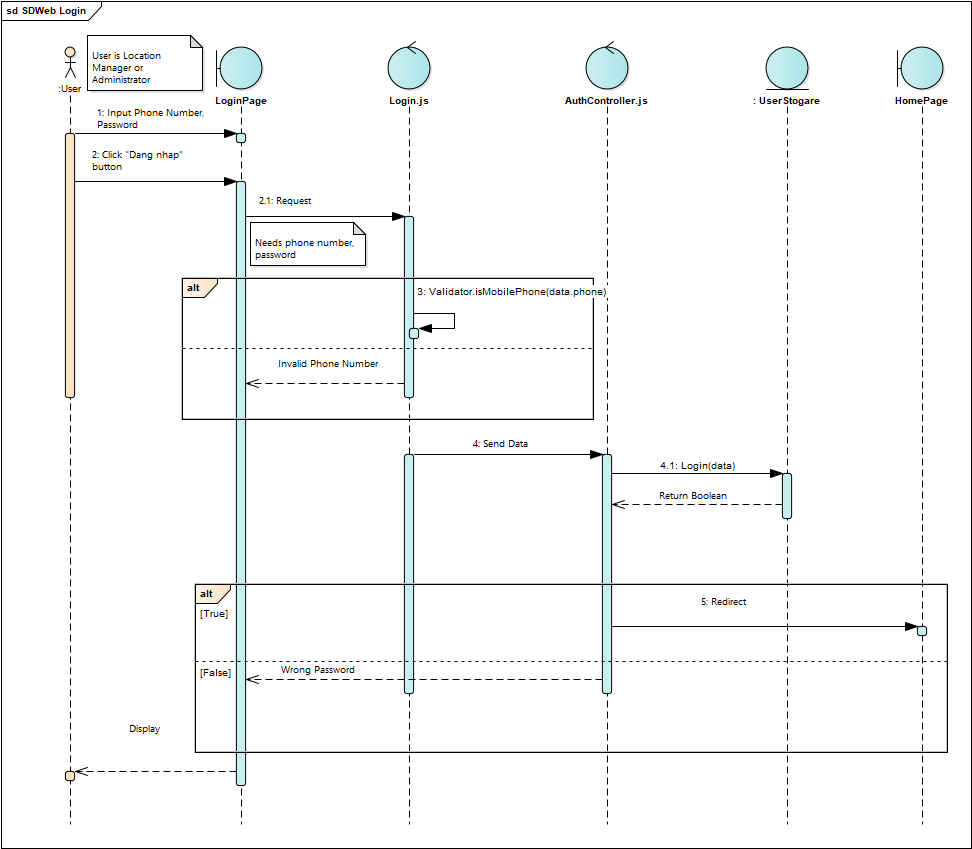
****

Image 24: Web Login sequence diagram

Web logout

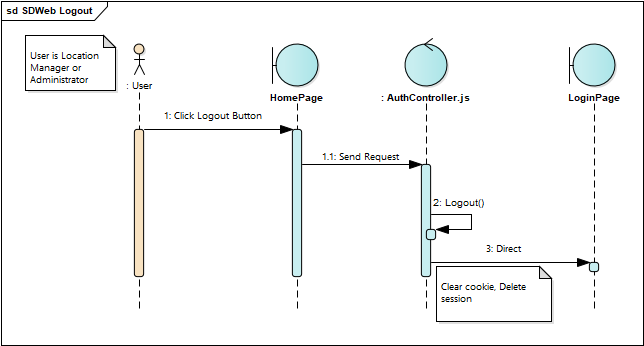
****

Image 25: Web Logout sequennce diagram

View user list

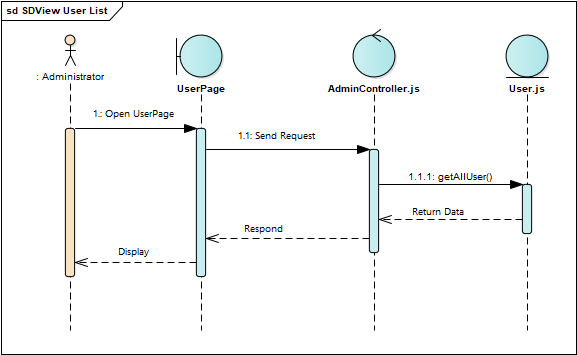
****

Image 26: View User List sequence diagram

View location list

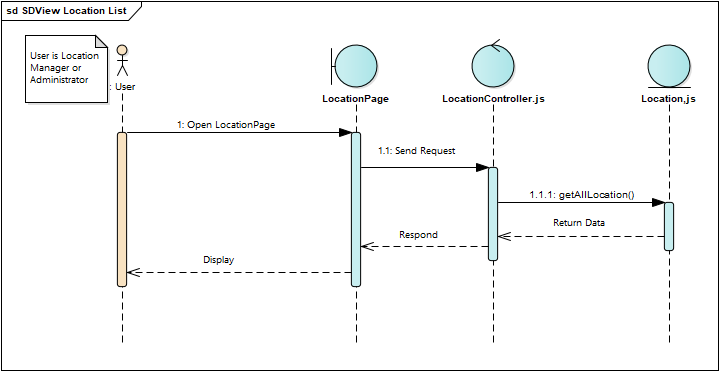
****

Image 27: View Location List sequence diagram

View location information

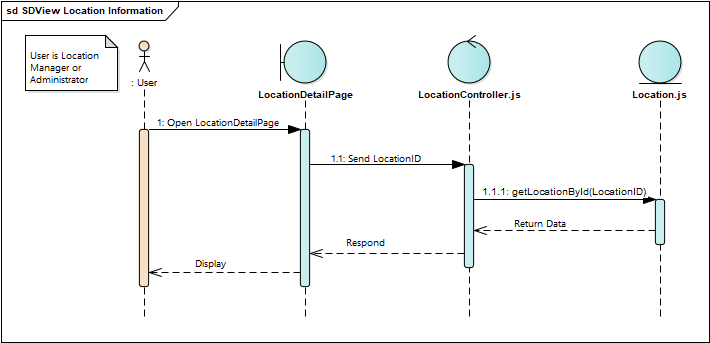
****

Image 28: VIew Location Information sequence diagram

View category list

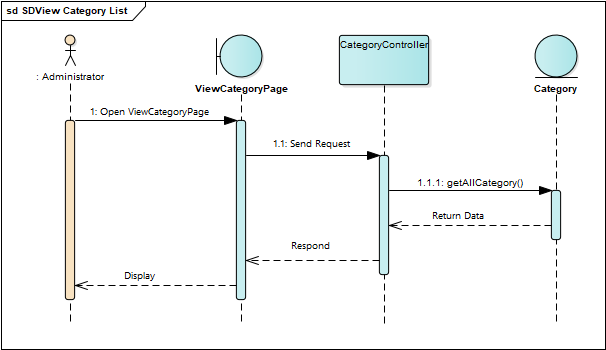
****

Image 29: View Category List sequence diagram

Show/Hide category

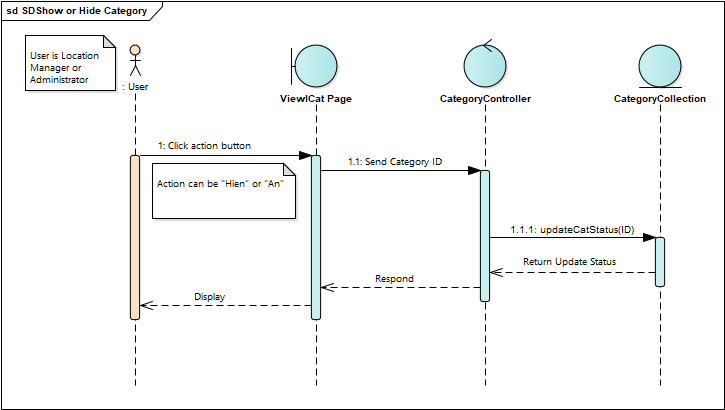
****

Image 30: Show/Hide Category sequence diagram

Location manager register

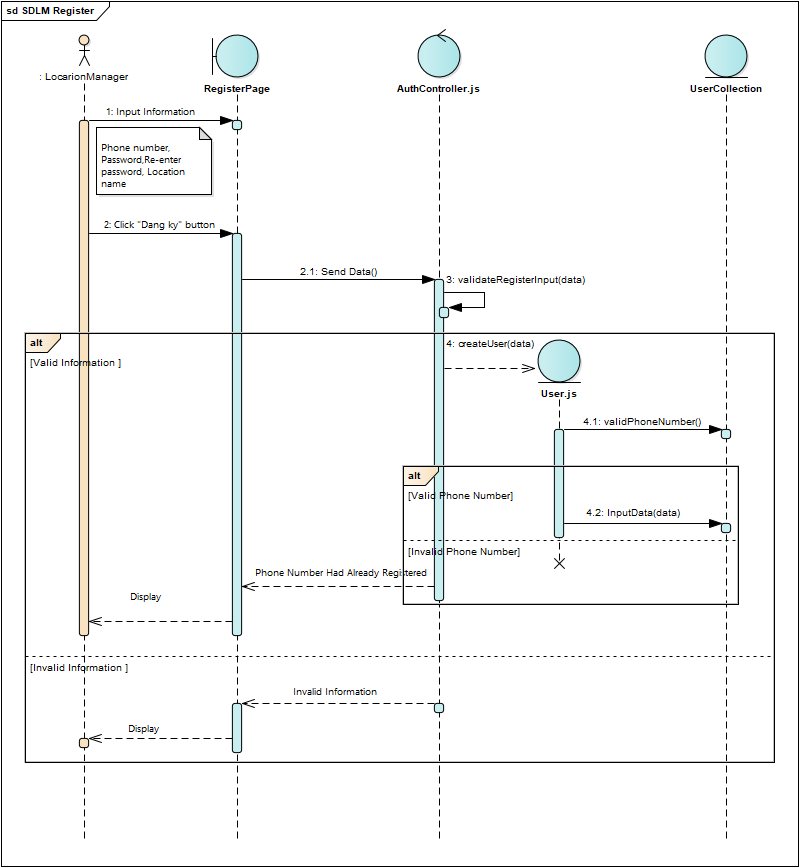
****

Image 31: Location Manager register sequence diagram

Show/Hide location

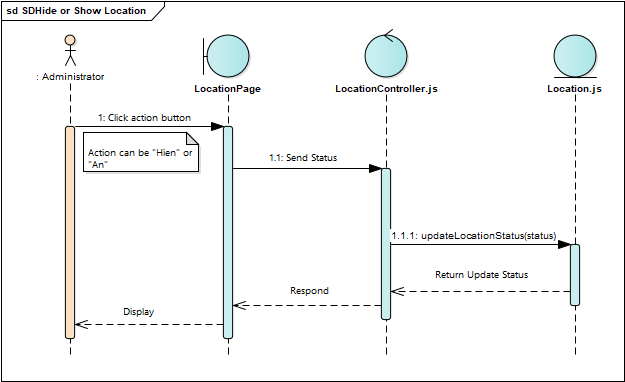
****

Image 32: Show/Hide Location sequence diagram

Edit location

****

Image 33: Edit Location sequence diagram

Web forgot password

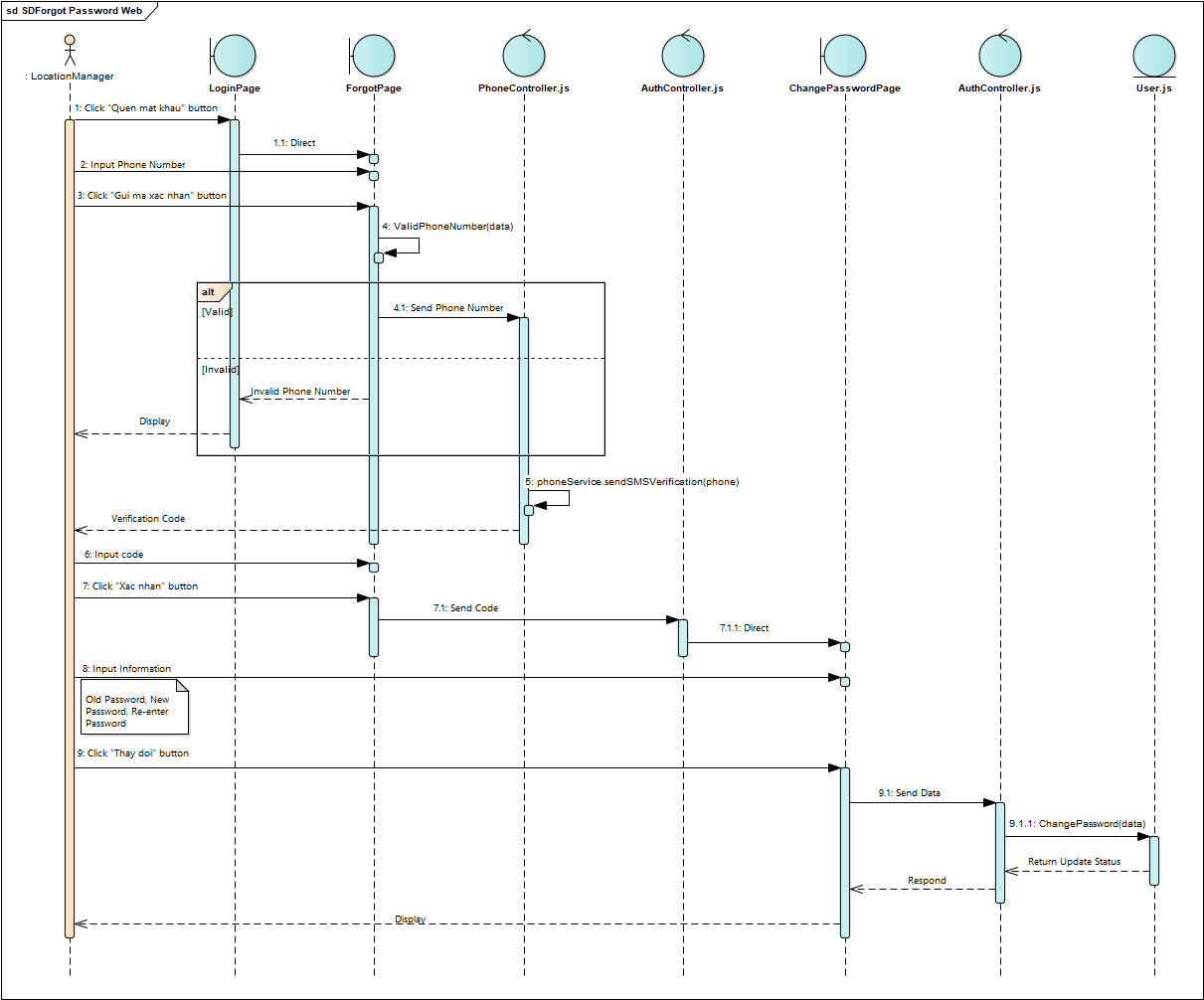
****

Image 34: Web Forgot Password sequence diagram

Create a pet

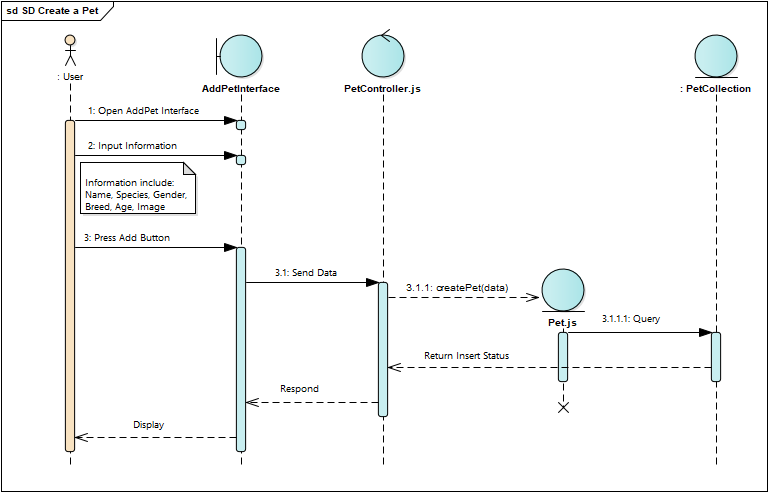
****

Image 35: Create Pet seqence diagram

Create a post

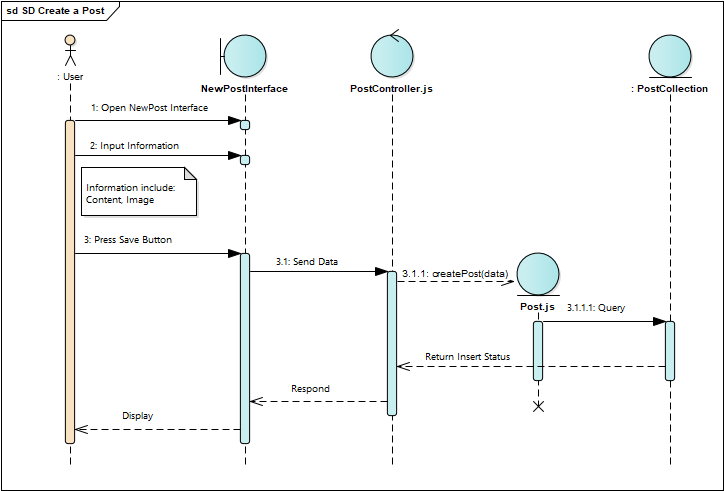
****

Image 36: Create a Post sequence diagram

## User interface design

Add Product page

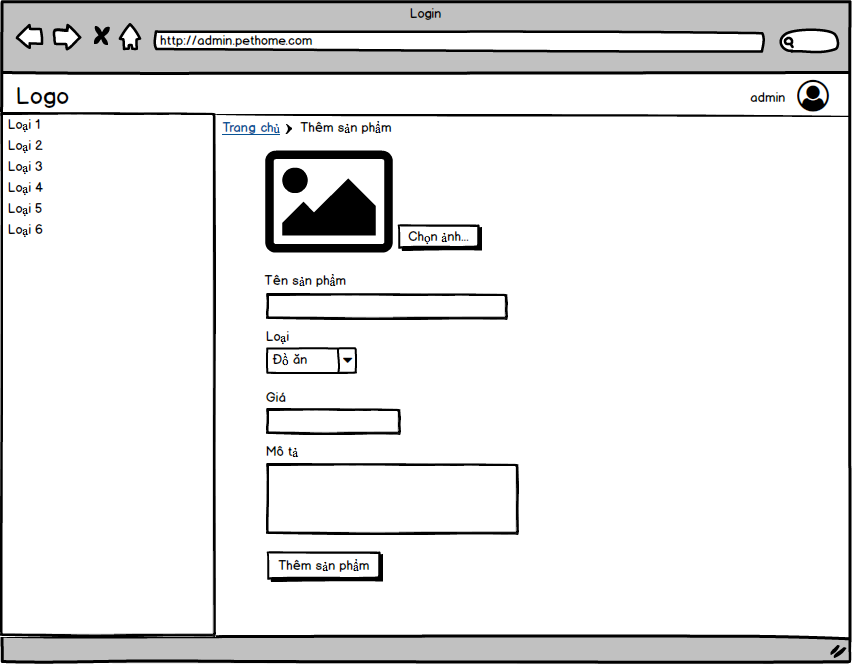


Image 38:Add Product page

Category Manage page

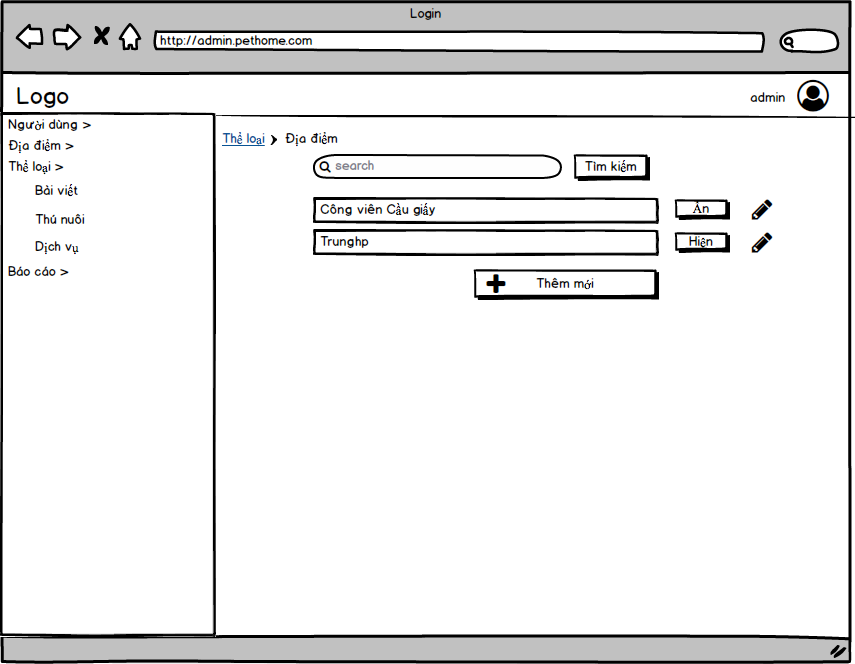


Image 39: Category Manage page

Location Manager Home page

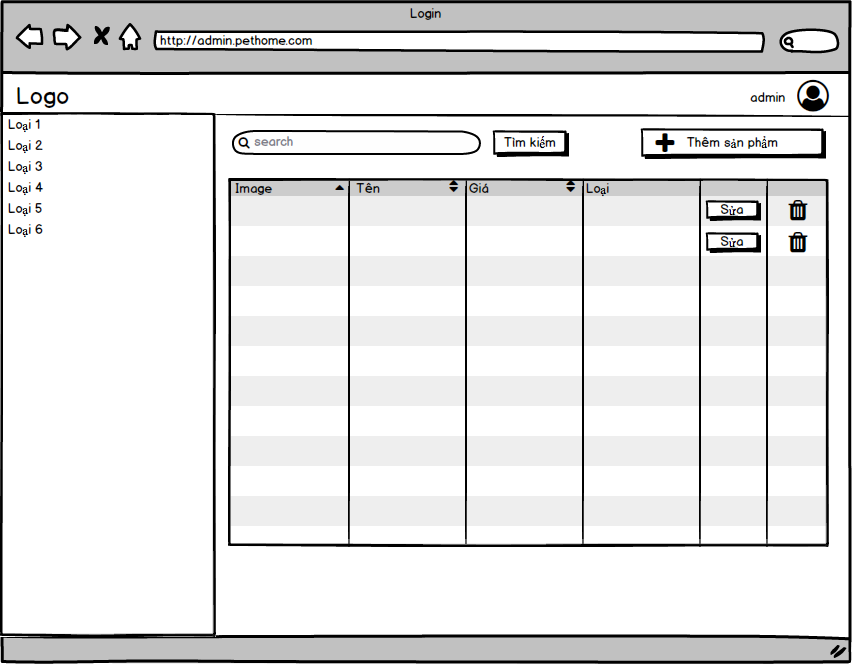


Image 40:Location manager Home page

Add Location page

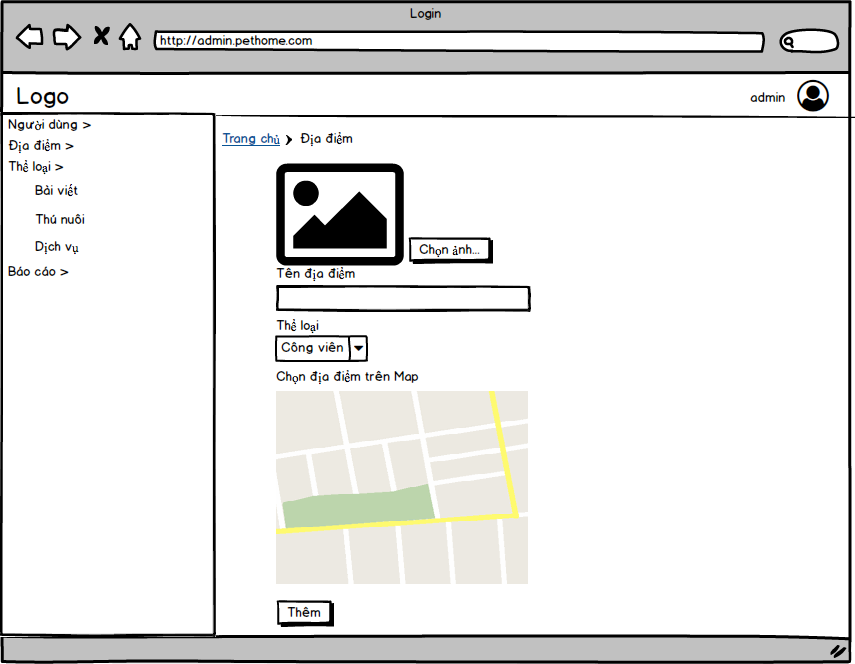


Image 41: Add Location page

Login page

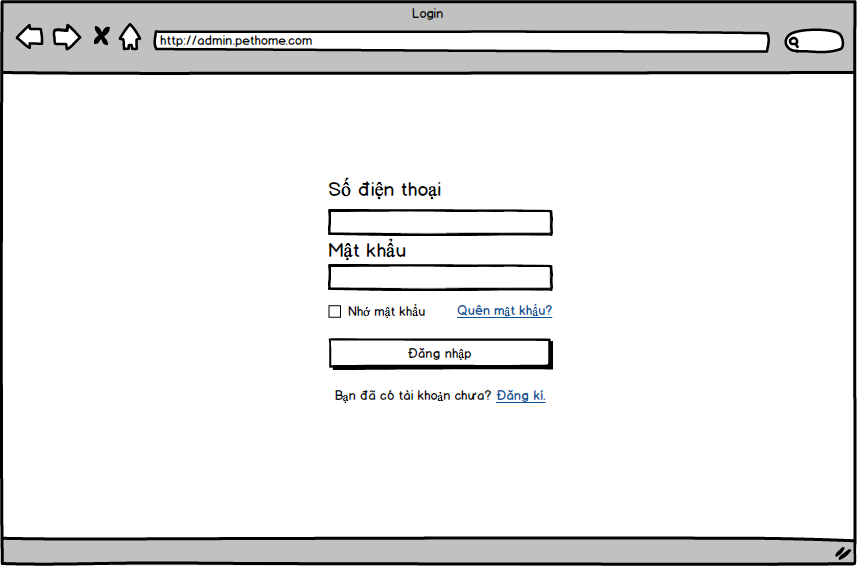


Image 42: Login page

Register page

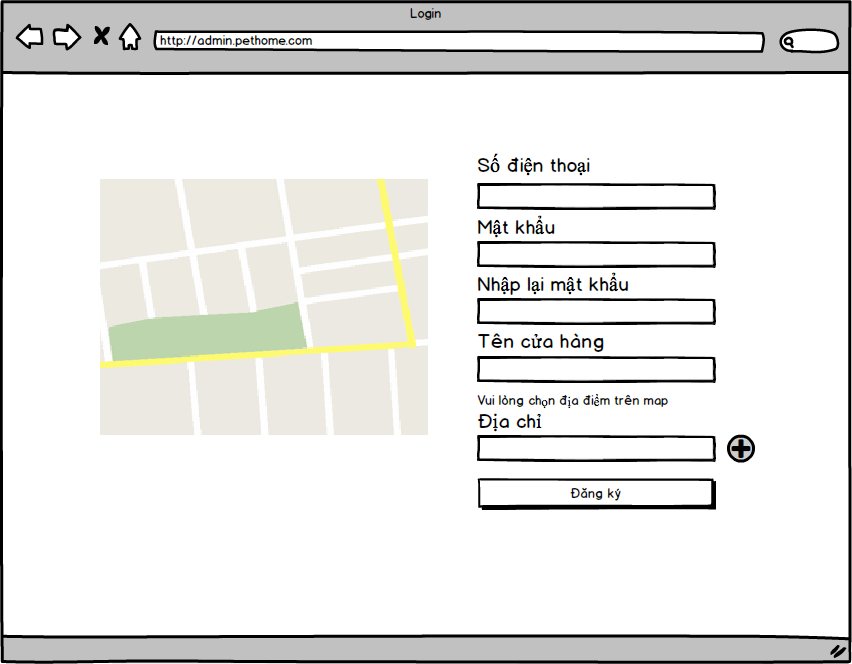


Image 43: Register page

Edit Product page

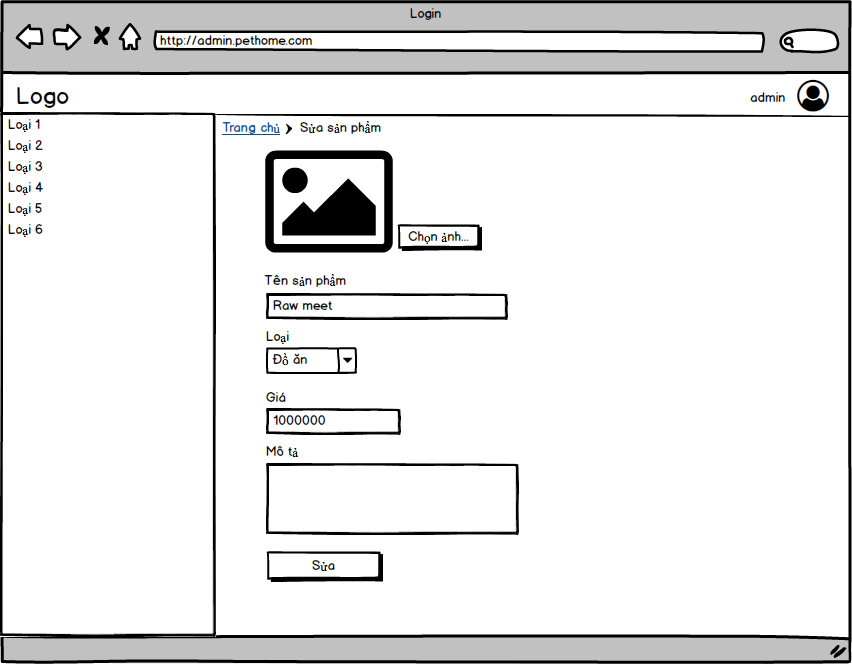


Image 44: Edit Product page

Verify Account page

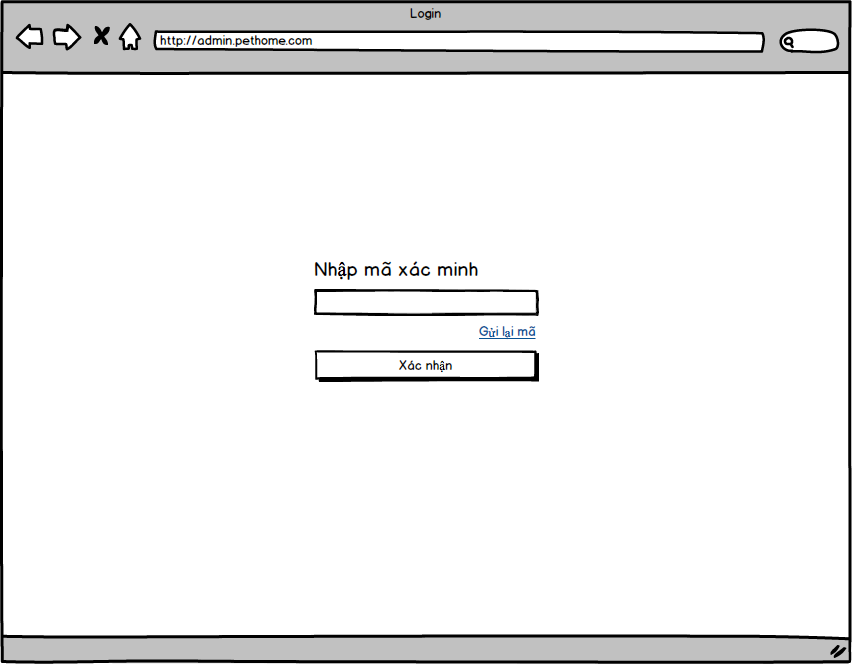


Image 45: Verify Account page

View Profile page

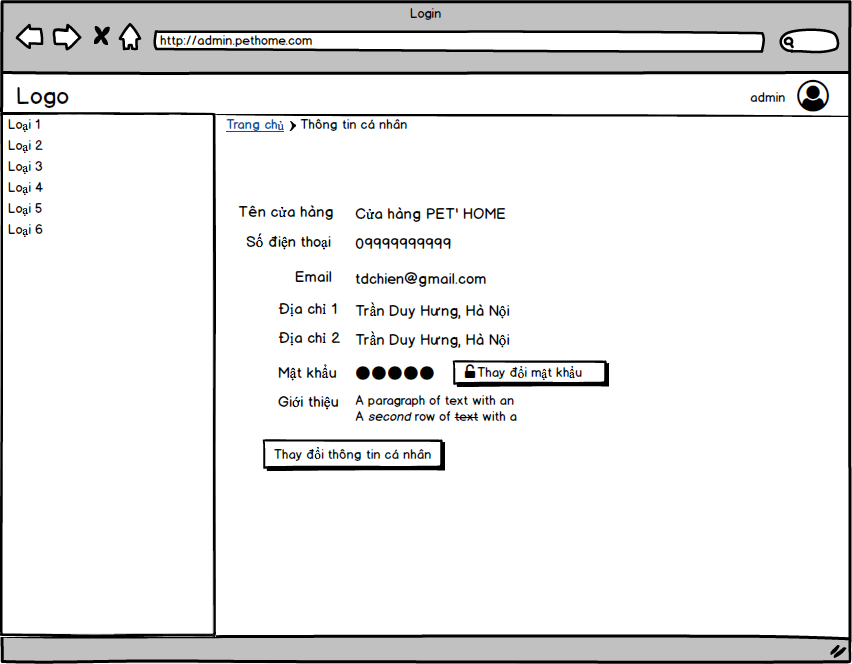


Image 46: View Profile page

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\auth\Login.png  Image 47: Login display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\auth\New password.png  Image 48: New password display |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\auth\Register.png  Image 49: Register display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\auth\Verify forget password.png  Image 50:Forgot password display |

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\auth\Verify forget passowrd enter code.png  Image 51:Verify forgot password display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\drawer\drawer.png  Image 52:Drawer |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\home\Home .png  Image 53:Home display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\home\Add Post.png  Image 54:Add post display |

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\home\Comment.png  Image 55:Comment display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\home\Message.png  Image 56:Message display |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\home\Notification.png  Image 57: Notification | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\home\Search Post.png  Image 58: Search post display |

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\home\Send message.png  Image 59: Send message display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\location\List View.png  Image 60: Search display |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\location\Location detail.png  Image 61: Location detail display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\location\Map View copy.png  Image 62: Map display |

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\location\Search Filter.png  Image 63: Search filter | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\pet\Add pet.png  Image 64: Add pet display |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\pet\Pet manager.png  Image 65: Pet manager display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\pet\Match History.png  Image 66: History matched display |

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\pet\Pet manager.png  Image 67: Pet management display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\profile\Change email.png  Image 68:Change email display |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\profile\Change password.png  Image 69: change password display | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\profile\Change username.png  Image 70: Change display name |

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\profile\Profile settings.png  Image 71: Edit profile | C:\Users\Asus\Dropbox\SWP490\Design\App Design-20181204T091447Z-001\App Design\profile\Profile.png  Image 72: View profile |
|  |  |

## Database design

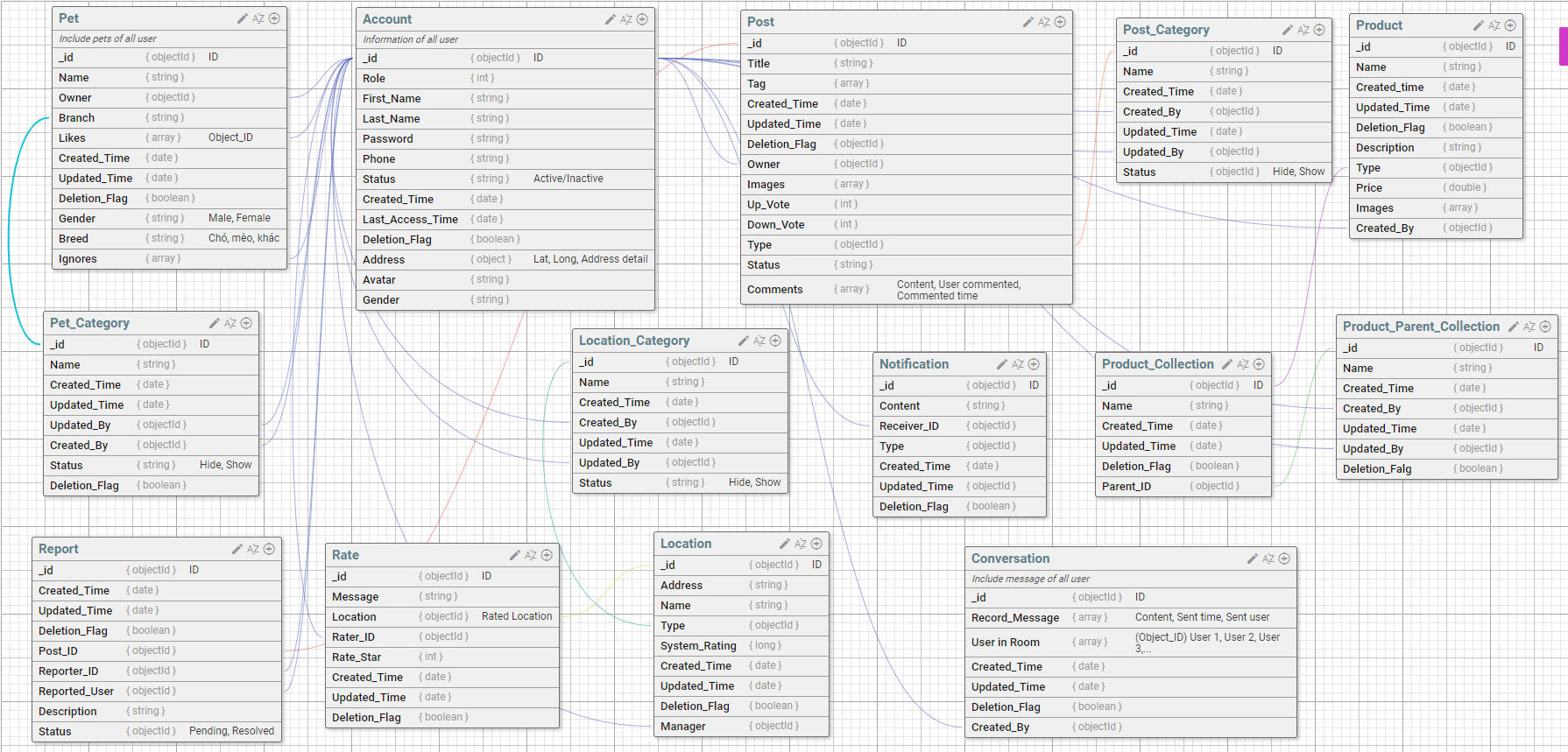


Image 73: Database design

Pets collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Name | String | Yes | Name of pet |
| 3 | Owner | ObjectId | Yes | The owner of pet  Foreign key references to \_id in Accounts collection. |
| 4 | Branch | String | Yes | Branch of pet |
| 5 | Likes | Array | Yes | All users like this pet  Foreign key references to \_id in Accounts collection. |
| 6 | Created\_Time | Date | Yes | The create time of pet |
| 7 | Updated\_Time | Date | Yes | The update time of pet |
| 8 | Gender | String | Yes | Male  Female |
| 9 | Breed | ObjectId | Yes | The breed of pet  Foreign key references to \_id in Pet\_Categories collection. |
| 10 | Ignores | Array | Yes | All users dislike this pet  Foreign key references to \_id in Accounts collection. |

Pet\_Categories collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Name | String | Yes | Name of pet category |
| 3 | Created\_Time | Date | Yes | The create time of category |
| 4 | Updated\_Time | Date | Yes | The update time of category |
| 5 | Updated\_By | ObjectId | Yes | User update that category  Foreign key references to \_id in Accounts collection. |
| 6 | Created\_By | ObjectId | Yes | User create that category  Foreign key references to \_id in Accounts collection. |
| 7 | Status | String | Yes | Status of category  Hide  Show |
| 8 | Deletion\_Flag | Boolean | Yes | Allow delete or not  True: allow  False: not allow |

Reports collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Created\_Time | Date |  | The create time of report |
| 3 | Updated\_Time | Date |  | The update time of report |
| 4 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |
| 5 | Post\_ID | ObjectId |  | ID of reported post  Foreign key references to \_id in Posts collection. |
| 6 | Reporter\_ID | ObjectId |  | ID of user report  Foreign key references to \_id in Accounts collection. |
| 7 | Reported\_User | ObjectId |  | ID of reported user  Foreign key references to \_id in Accounts collection. |
| 8 | Description | String |  | Description of report |
| 9 | Status | String |  | Status of report  Pending  Resolved |

Accounts collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Role | Int |  | Role of account  1: Location Manager  2: User  3: Administrator |
| 3 | First\_Name | String |  | First name of user |
| 4 | Last\_Name | String |  | Last name of user |
| 5 | Password | String |  | Password of account |
| 6 | Status | Int |  | Status of account  1: Active  2: Inactive |
| 7 | Created\_Time | Date |  | The create time of account |
| 8 | Last\_Access\_Time | Date |  | Last time account logged in |
| 9 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |
| 10 | Address | Object |  | Address of user |
| 11 | Avatar | String |  | URL of user’s avatar |
| 12 | Gender | String |  | Gender of user  Male  Female  Other |

Location\_Categories collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Name | String |  | Name of category |
| 3 | Created\_Time | Date |  | The create time of category |
| 4 | Created\_By | ObjectId |  | User create category  Foreign key references to \_id in Accounts collection. |
| 5 | Updated\_Time | Date |  | The update time of category |
| 6 | Updated\_By | ObjectId |  | User update that category  Foreign key references to \_id in Accounts collection. |
| 7 | Status | String |  | Status of category  Hide  Show |

Rates collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Message | String |  | Message of user |
| 3 | Location | ObjectId |  | Rated location  Foreign key references to \_id in Locations collection. |
| 4 | Rater\_ID | ObjectId |  | User rate  Foreign key references to \_id in Accounts collection. |
| 5 | Rate\_Star | Int |  | Number of star of rated |
| 6 | Created\_Time | Date |  | The create time of rate |
| 7 | Updated\_Time | Date |  | The update time of rate |
| 8 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |

Locations collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Address | String |  | Address of location |
| 3 | Name | String |  | Name of location |
| 4 | Type | ObjectId |  | Type of location  Foreign key references to \_id in Location\_Categories collection. |
| 5 | System\_Rating | Long |  | Average rated star of location |
| 6 | Created\_Time | Date |  | The create time of location |
| 7 | Updated\_Time | Date |  | The update time of location |
| 8 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |
| 9 | Manager | ObjectId |  | Owner of location  Foreign key references to \_id in Accounts collection. |
| 10 | Created\_By | ObjectId |  | User create location  Foreign key references to \_id in Accounts collection. |

Posts collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Title | String |  | Title of post |
| 3 | Tag | Array |  | Tags of post |
| 4 | Created\_Time | Date |  | The create time of post |
| 5 | Updated\_Time | Date |  | The update time of post |
| 6 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |
| 7 | Owner | ObjectId |  | Owner of post  Foreign key references to \_id in Accounts collection. |
| 8 | Images | Array |  | URL of all images in post |
| 9 | Up\_Vote | Int |  | Number of user like post |
| 10 | Down\_Vote | Int |  | Number of user dislike post |
| 11 | Type | ObjectId |  | Type of post  Foreign key references to \_id in Post\_Categories collection. |
| 12 | Status | String |  | Status of post |
| 13 | Comments | Array |  | All comments in post  Foreign key references to \_id in Accounts collection. |

Notifications collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Content | String |  | Content of notification |
| 3 | Receiver\_ID | ObjectId |  | User receive the notice  Foreign key references to \_id in Accounts collection. |
| 4 | Type | Int |  | Type of notification  1: Message  2: Up vote  3: Down vote  4: Report |
| 5 | Created\_Time | Date |  | The create time of notice |
| 6 | Updated\_Time | Date |  | The update time of notice |
| 7 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |
| 8 | Sender\_ID | ObjectId |  | User actioned  Foreign key references to \_id in Accounts collection. |

Conversations collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Record\_Message | Array |  | Messages in conversation |
| 3 | User in Room | Array |  | Users in conversation |
| 4 | Created\_Time | Date |  | The create time of conversation |
| 5 | Updated\_Time | Date |  | The update time of conversation |
| 6 | Created\_By | ObjectId |  | User create conversation  Foreign key references to \_id in Accounts collection. |
| 7 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |

Post\_Categories collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Name | String |  | Name of category |
| 3 | Created\_Time | Date |  | The create time of category |
| 4 | Created\_By | ObjectId |  | User create category  Foreign key references to \_id in Accounts collection. |
| 5 | Updated\_Time | Date |  | The update time of category |
| 6 | Updated\_By | ObjectId |  | User update that category  Foreign key references to \_id in Accounts collection. |
| 7 | Status | String |  | Status of category  Hide  Show |

Product\_Collections collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Name | String |  | Name of product collection |
| 3 | Created\_Time | Date |  | The create time of product collection |
| 4 | Updated\_Time | Date |  | The update time of product collection |
| 5 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |
| 6 | Parent\_ID | ObjectId |  | Parent of product collection  Foreign key references to \_id in Product\_Parent\_Collections collection. |

Product\_Parent\_Collections collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Type** | **Allow Null** | **Description** |
| 1 | \_id | ObjectId | No | Auto generated ID  Unique |
| 2 | Name | String |  | Name of product collection |
| 3 | Created\_Time | Date |  | The create time of product collection |
| 4 | Updated\_Time | Date |  | The update time of product collection |
| 5 | Deletion\_Flag | Boolean |  | Allow delete or not  True: allow  False: not allow |
| 6 | Updated\_By | ObjectId |  | The user update product collection  Foreign key references to \_id in Accounts collection. |

# Chapter 5: Software testing document



## Purpose

This chapter provides details related to testing of the system, including the scope of testing, test plan, test case and test report. Through testing, system defects can be discovered and fixed, allowing for better software quality. Using the contents of this chapter, developers and testers can work together to ensure that the system is thoroughly tested and functions as intended.

## Scope of testing



### Stages of testing

There are 3 phases in Testing Process: Unit testing, Integration testing, System testing.

|  |  |
| --- | --- |
| **Test Level** | **Description** |
| Unit Test | Unit Tests are performed to ensure that individual units of source code function as intended. Unit tests are primarily used to test complicated algorithms and automatically test important functions. Postman is used to perform unit tests. |
| Integration Test | Integration Tests are performed by PH testers to test whether combined units function as intended. |
| System Test | System Tests are performed by PH testers to test the system as a whole and determine whether or not the system meets the requirements. No specific tools are used for system tests. |

Image 74: Stages of testing

### Types of testing

* The following types of testing are performed:
* Function testing
  + GUI testing
  + Regression testing

### Range of testing

All features as described in the Software Requirement Specification are tested.

## Test plan



### Testing tool and environment

* + - 1. **Testing tools**

The following tools are used for testing:

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| Google Chrome v70.0 | Use to view the web page, bug logging page, etc.. |
| Microsoft Excel | Use to manage test cases and manage bugs. |
| Postman | Use to perform unit tests. |
| Expo | Use to test application interface, login with Facebook, etc… |

Table 16:Testing tools

* + - 1. **Testing Environment**

The following table describes the testing environment:

|  |  |  |
| --- | --- | --- |
| **Type of Testing** | **Software** | **Hardware** |
| Unit Test | Postman | Laptop Asus:   * Window 10 Pro 64-bit * Intel Core i5 2.5Ghz * 8GB Memory * 1,161GB Disk Space   Laptop Lenovo:   * Window 10 Pro 64-bit * Intel Core i5 2.4Ghz * 8GB Memory * 661GB Disk Space |
| Integration Test | Google Chrome v70.0 | Laptop Asus:   * Window 10 Pro 64-bit * Intel Core i5 2.5Ghz * 8GB Memory * 1,161GB Disk Space * Screen solution: 1920 x 1080 |
| Expo | Iphone 6s:   * IOS 12.0.1 * Apple A9 1,8Ghz * 2GB Memory * 64GB Disk Space |
| System Test | Google Chrome v70.0 | Laptop Asus:   * Window 10 Pro 64-bit * Intel Core i5 2.5Ghz * 8GB Memory * 1,161GB Disk Space * Screen solution: 1920 x 1080   Laptop Lenovo:   * Window 10 Pro 64-bit * Intel Core i5 2.4Ghz * 8GB Memory * 661GB Disk Space |
| Expo | Iphone 6s:   * IOS 12.0.1 * Apple A9 1,8Ghz * 2GB Memory * 64GB Disk Space * Screen solution: 1334 x 750 |

Table 17:Testing environment



### Resource and responsibilities

|  |  |  |
| --- | --- | --- |
| **ID** | **Resource** | **Responsibilities** |
| 1 | Test Leader | * Review test cases and test report. * Review overall quality of the project. |
| 2 | Tester | * Preforming the actual system testing * Create test plan * Creat test cases * Perform tests * Write test reports * Log bugs |
| 3 | Developer | * Create and perform Unit Tests. * Fix bugs. |

Table 18:Resources and responsibilities

### Test strategy

* + 2. 1. **Test model**

This project follows the V-Model for testing

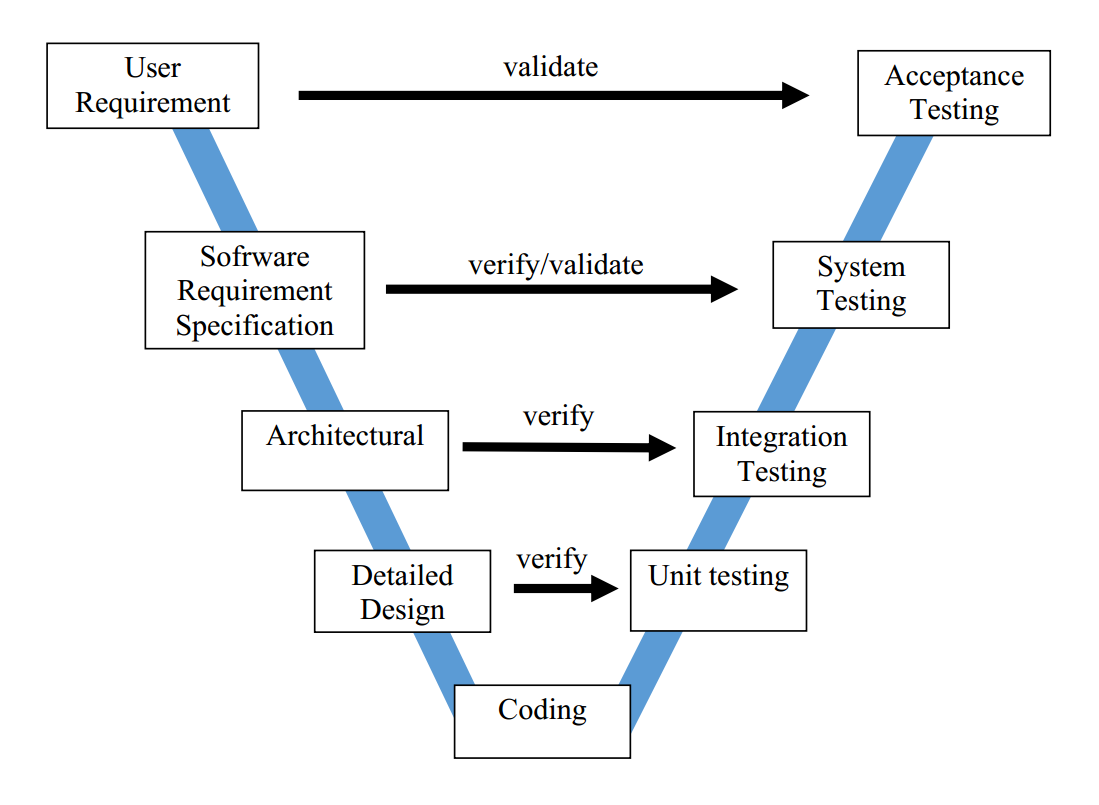


Figure 26:V-Model

* + - 1. **Testing types**

The following types of testing are performed:

* Functional testing:
  + Tests whether or not functions of the system work as intended on the PH system.
  + Function testing is a type of black-box testing and as such, internal structure is rarely considered.
  + GUI (Graphical user interface) testing:
    - Tests whether or not elements of the websites appear and work correctly, and are positioned correctly.
    - Tests whether or not the website is usable and appears as intended with various browser sizes:
      * 1366 x 768
      * 1920 x 1080
    - Tests whether or not the application is usable and appears as intended with various screen sizes:
      * 1334 x 750
      * 1080 x 1920
    - This test is targeted to cover the verification of the overall look and feel of the PH system including initial position, font, text size, color, focus, initial button, tab order, label, screen sizes and sentences width.
    - Check all the Graphical User Interface elements for size, position, width, length and acceptance of characters or numbers:
* Check if Error Messages are displayed correctly
* Check if font used in application is readable.
* Check if the alignment of the text is proper.
* Check if the color of the font and warning messages is clear and not confusing.
* Check if the images have good clarity.
* Check if the images are aligned properly.
* Regression testing:
  + - Regression testing ensures that bugs are fixed.
    - Regression testing ensures that the fixing of a bug does not impact other parts of the system.
      1. Stages of testing

|  |  |  |  |
| --- | --- | --- | --- |
| Test Type | Stages of testing | | |
| Unit Test | Integration Test | System Test |
| Regression testing | ✓ | ✓ | ✓ |
| Functional testing |  | ✓ | ✓ |
| GUI testing |  | ✓ | ✓ |

Table 19:Test stages

* + - 1. test Schedule

Table below are the Test Schedule for PH project. That consist of two iterations:

|  |  |  |
| --- | --- | --- |
| **Test Schedule** | **Start Date** | **End Date** |
| ***Iteration 1*** | 25/11/2018 | 05/12/2018 |
| Functional testing | 25/11/2018 | 28/11/2018 |
| GUI testing | 29/11/2018 | 30/11/2018 |
| System testing | 01/12/2018 | 03/12/2018 |
| Regression testing | 04/12/2018 | 05/12/2018 |
| ***Iteration 2*** | 12/12/2018 | 31/12/2018 |
| Functional testing | 12/12/2018 | 20/12/2018 |
| GUI testing | 21/12/2018 | 24/12/2018 |
| System testing | 25/12/2018 | 29/12/2018 |
| Regression testing | 30/12/2018 | 31/12/2018 |

Table 20:Test schedule

* + - 1. deliverables

Table below are the Deliverables for PTO Project:

|  |  |  |
| --- | --- | --- |
| **Deliverables** | **Responsibilities** | **Completion date** |
| Test plan | Test Leader | 30/11/2018 |
| Test cases | Testers | 19/12/2018 |
| Test case review | Testers + Test Leader | 20/12/2018 |
| Test data( Data Set) | Testers | 22/12/2018 |
| Defect report | All members | 23/12/2018 |
| Final test report | Test Leader | 30/12/2018 |

Table 21:Deliverables

### Feature to be tested

All features described in the list of use cases in the Software Requirement Specification are to be tested.

## Test approach



### Unit testing with Postman

Unit tests are done by the developers in order to test individual units, and approved by the test leader. PH uses Postman to execute unit test.

PH embrace this feature in order to gain the following advantages:

* Reduce the level of bugs in production code
* Save development time
* Easier to change and refactor code
* Unit Tests are a form of documentation

### Integration and system test

Integration and system tests are done by testers to ensure that combined units work correctly and that the system as a whole function as intended. Each test case is tested using Google Chrome v70.0 and Expo

GUI testing is also done during this process to ensure that elements load and function correctly, text is readable.

Test case details are described in the file:

* **PH\_Integration Test Case.xlsx**
* **PH\_System Test Case.xlsx**

## Test report



### Unit tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Iteration 1** | | **Iteration 2** | |
| **Before** | **After** | **Before** | **After** |
| **Pass** | 170 | 183 | 220 | 241 |
| **Fail** | 13 | 0 | 21 | 0 |
| **Untested** | 0 | 0 | 0 | 0 |
| **Total** | **183** | **183** | **241** | **241** |

Table 22:Unit test report

For each of the tested units, all branches are tested and code coverage is 100%.



### Integration tests

The number of test cases is shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module** | **Pass** | **Fail** | **Untested** | **Total** |
| **Administrator** | 39 | 0 | 0 | 39 |
| **Location Manager** | 39 | 0 | 0 | 39 |
| **User** | 92 | 0 | 0 | 92 |
| **Guest** | 20 | 0 | 0 | 20 |
| **Total** | **175** | **0** | **0** | **175** |

Table 23:Integration test report

List of test case: (Administrator module)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** |
|  | **Authentication** |  |  |  |
| [Administrator-1] | Check Login page | 1. Actor enters URL to view admin page | 1. System display the Login page include all elements: - "Số điện thoại" textfield - "Mật khẩu" textfield - "Đăng nhập" button - "Dăng kí" button - "Quên mật khẩu?" button | Actor is currently not logged in. |
| [Administrator-2] | Login successfully | 1. Actor enters URL to view login page of admin 2. Actor fill valid data for "Số điện thoại" and "Mật khẩu" textfield 3. Actor click "Đăng nhập" button | System direct to Home Page of Administrator | Actor is currently not logged in. |
| [Administrator-3] | Login with blank "Tên đăng nhập" field or "Mật khẩu" field | 1. Actor enters URL to view login page of admin 2. Actor does not fill data for "Số điện thoại" or "Mật khẩu" textfield 3. Actor click "Đăng nhập" button | System display error message: "Vui lòng nhập số điện thoại và mật khẩu hợp lệ" | Actor is currently not logged in. |
| [Administrator-4] | Login with blank "Tên đăng nhập" field and "Mật khẩu" field | 1. Actor enters URL to view login page of admin 2. Actor does not fill data for "Số điện thoại" and "Mật khẩu" textfield 3. Actor click "Đăng nhập" button | System display error message: "Vui lòng nhập số điện thoại và mật khẩu hợp lệ" | Actor is currently not logged in. |
| [Administrator-5] | Login with invalid phone number | 1. Actor enters URL to view login page of admin 2. Actor inputs invalid phone numbe for "Số điện thoại" field 3. Actor inputs valid data for following fields: - Mật khẩu 4. Actor click "Đăng nhập" button | System display error message: "Vui lòng nhập số điện thoại và mật khẩu hợp lệ" | Actor is currently not logged in. |
| [Administrator-6] | Logout | 1. Actor click to phone number at the header 2. Actor clicks "Đăng xuất" | System direct to Login page | Actor logged in with administrator account |
| [Administrator-7] | Check Home page | 1. Actor enters URL to view login page of admin 2. Actor fill valid data for "Số điện thoại" and "Mật khẩu" textfield 3. Actor click "Đăng nhập" button | System display Home page include elemnts: | Actor logged in with administrator account |
| [Administrator-8] | Check Logo Home page | 1. Actor enters URL to view login page of admin 2. Actor fill valid data for "Số điện thoại" and "Mật khẩu" textfield 3. Actor click "Đăng nhập" button 4. Actor click the logo at the left top of Home page | System refresh Home page | Actor logged in with administrator account |
| [Administrator-9] | Check left side menu | 1. Actor enters URL to view login page of admin 2. Actor fill valid data for "Số điện thoại" and "Mật khẩu" textfield 3. Actor click "Đăng nhập" button | 1. System display Home page 2. The left side menu of Home page include elements: - "Người dùng" - "Thể loại" + "Bài viết" + "Thú nuôi" + "Dịch vụ" + "Địa điểm" - "Quản lý danh mục" - "Báo cáo" | Actor logged in with administrator account |
| **System managerment** | | | | |
| **User managerment** | | | | |
| [Administrator-10] | View all user in system | 1. Actor clicks"Người dùng" at the left side bar | System display User Managerment page include elements: - A table with 5 columns: + "Tên"  + "Số điện thoại" + "Email" + "Trạng thái" + "Hành động" - In the 5th colunm of table has "Chặn" or "Bỏ chặn" button- At "Tên", "Số điện thoại", "Email" cell has black up and down arrow | Actor logged in with administrator account |
| [Administrator-11] | Ban user | 1. Actor clicks"Người dùng" at the left side bar  2. Actor clicks "Chặn" button (if have) at the line of corresponding user | 1. System display User Managerment page. 2. System show "Chặn người dùng thành công" message and "Chặn" button change to "Bỏ chặn" button | Actor logged in with administrator account |
| [Administrator-12] | Unban user | 1. Actor clicks"Người dùng" at the left side bar  2. Actor clicks "Bỏ chặn" button (if have) at the line of corresponding user | 1. System display User Managerment page. 2. System show "Chặn người dùng thành công" message and "Bỏ chặn" button change to "Chặn" button | Actor logged in with administrator account |
| [Administrator-13] | Sort user by name | 1. Actor clicks"Người dùng" at the left side bar  2. Actor clicks "Tên" cell of table | 1. System display User Managerment page. 2. All user in table sort by name with alphabet order | Actor logged in with administrator account |
| [Administrator-14] | Sort user by phone number | 1. Actor clicks"Người dùng" at the left side bar  2. Actor click "Số điện thoại" cell of table | 1. System display User Managerment page. 2. All user in table sort by phone number | Actor logged in with administrator account |
| [Administrator-15] | Sort user by email | 1. Actor clicks"Người dùng" at the left side bar  2. Actor click "Email" cell of table | 1. System display User Managerment page. 2. All user in table sort by email with alphabet order | Actor logged in with administrator account |
| **Category managerment** | | | | |
| [Administrator-16] | View all category | 1. Actor clicks"Quản lý thể loại" at the left side bar | System display Category Managerment page include elements: - Filter + Filter label + Filter combobox - All category in system - "Ẩn" button or "Hiện" button - "Eye icon" button | Actor logged in with administrator account |
| [Administrator-17] | View Post category | 1. Actor clicks "Bài viết" at the left side bar | System display Category Managerment page include elements: - Filter + Filter label + Filter combobox display "Bài viết" - All category in system - "Ẩn" button or "Hiện" button - "Eye icon" button | Actor logged in with administrator account |
| [Administrator-18] | View Pet category | 1. Actor clicks "Thú nuôi" at the left side bar | System display Category Managerment page include elements: - Filter + Filter label + Filter combobox display "Thú nuôi" - All category in system - "Ẩn" button or "Hiện" button - "Eye icon" button | Actor logged in with administrator account |
| [Administrator-19] | View Service category | 1. Actor clicks "Dịch vụ" at the left side bar | System display Category Managerment page include elements: - Filter + Filter label + Filter combobox display "Dịch vụ" - All category in system - "Ẩn" button or "Hiện" button - "Eye icon" button | Actor logged in with administrator account |

Table 24: Administrator module integration test case

List of test case: (Location manager module)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** |
| **Authentication** | | | | |
| **Register** | | | | |
| [LocationManager-1] | Check "Đăng ký" page | 1. Actor enter URL to access the web application 2. Actor click "Đăng ký" 3. Actor input phone number and press "Gửi mã xác minh" 4. Actor input verification code and press "Xác nhận" | 1. System display Login page 2. System display "Đăng ký" page include elements: - "Số điện thoại" 3. System display "Mã Xác minh" 4. System display "Đăng ký cửa hàng" include elements: - "Tên địa điểm" - "Mật khẩu" - "Xác nhận mật khẩu" - "Địa chỉ chi tiết" - "Loại địa điểm" - Map |  |
| [LocationManager-2] | Check register when actor input wrong phone number data | 1. Actor enter URL to access the web application 2. Actor click "Đăng ký" 3. Actor input phone number like example: - 34556463453453  4. Actor press "Gửi mã xác minh" | 1. System display Login page 2. System display "Đăng ký" page include elements: - "Số điện thoại" 3. System display error message: - "Số điện thoại không hợp lệ" |  |
| [LocationManager-3] | Check register when actor input invalid verification code | 1. Actor enter URL to access the web application 2. Actor click "Đăng ký" 3. Actor input phone number and press "Gửi mã xác minh" 4. Actor input verification code like example: - 123456 | 1. System display Login page 2. System display "Đăng ký" page include elements: - "Số điện thoại" 3. System display "Mã Xác minh" 4. System display error message: - "Mã xác minh không đúng" |  |
| [LocationManager-4] | Check register when actor not input data | 1. Actor enter URL to access the web application 2. Actor click "Đăng ký" 3. Actor input phone number and press "Gửi mã xác minh" 4. Actor input verification code and press "Xác nhận" 5. Actor press "Đăng ký" | 1. System display Login page 2. System display "Đăng ký" page include elements: - "Số điện thoại" 3. System display "Mã Xác minh" 4. System display "Đăng ký cửa hàng" page 5. System show error messages: - "Vui lòng điền địa điểm"  - "Vui lòng nhập mật khẩu - "Vui lòng nhập lại mật khẩu" - "Vui lòng chọn một địa điểm trên bản đồ" |  |
| [LocationManager-5] | Check register when actor input "Mật khẩu" not fit with "Xác nhận mật khẩu" | 1. Actor enter URL to access the web application 2. Actor click "Đăng ký" 3. Actor input phone number and press "Gửi mã xác minh" 4. Actor input verification code and press "Xác nhận" 5. Actor press "Đăng ký" 6.Actor input - "Trung's Store" for "Địa điểm" field - "123456" for "Mật khẩu" field - "1234567" for "Xác nhận mật khẩu" - "Linh Lang, Cống vị, Ba Đình, Hà Nội" for "Địa chỉ" field - Create a dot in map 7. Actor press "Đăng ký" button | 1. System display Login page 2. System display "Đăng ký" page include elements: - "Số điện thoại" 3. System display "Mã Xác minh" 4. System display "Đăng ký cửa hàng" page 5. System show error message:  - "Mật khẩu không khớp" |  |
| [LocationManager-6] | Check actor register with valid data | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number 4. Actor click "Gửi mã xác minh" button 5. Actor input verification code: 47261 (for example )  6. Actor pess "Xác nhận" button 7. Actor input - "Trung's Store" for "Địa điểm" field - "123456" for "Mật khẩu" field - "123456" for "Xác nhận mật khẩu" - "Linh Lang, Cống vị, Ba Đình, Hà Nội" for "Địa chỉ" field - Create a dot in map 8. Actor press "Đăng ký" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display:  - "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát"  And also display:  - "Mã xác minh field" - "Xác nhận" button  4. System display:  - "Vui lòng nhập chính xác mã xác minh" message 5. System direct to "Đăng ký cửa hàng" page 6. System show success message:  - "Đăng ký thành công" and direct to Login page |  |
| **Login** | | | | |
| [LocationManager-7] | Check login view | 1. Actor enter URL to access the web application | 1. System display "Đăng nhập" page include elements: - "Số điện thoại" - "Mật khẩu" |  |
| [LocationManager-8] | Check login when actor not input data | 1. Actor enter URL to access the web application 2. Actor clicks "Đăng nhập" | 1. System display "Đăng nhập" page  2. System display error message: - "Số điện thoại và mật khẩu không hợp lệ" |  |
| [LocationManager-9] | Check login when actor input wrong "Số điện thoại" | 1. Actor enter URL to access the web application 2. Actor input data like example: - Số điện thoại: 34534534553 - Mật khẩu: 123345 3. Actor clicks "Đăng nhập" | 1. System display "Đăng nhập" page  2. System display error message: - "Số điện thoại không hợp lệ" |  |
| [LocationManager-10] | Check login when actor input wrong "Mật khẩu" | 1. Actor enter URL to access the web application 2. Actor input data like example: - Số điện thoại: 0342256852 - Mật khẩu: trhtynghj 3. Actor clicks "Đăng nhập" | 1. System display "Đăng nhập" page  2. System display error message: - "Mật khẩu không đúng" |  |
| [LocationManager-11] | Check login when actor input valid "Số điện thoại" and "Mật khẩu" | 1. Actor enter URL to access the web application 2. Actor input data like example: - Số điện thoại: 0342256852 - Mật khẩu: 123456 3. Actor clicks "Đăng nhập" | 1. System display "Đăng nhập" page  2. System direct to "Quản lý sản phẩm" page |  |
| **Product Management** | | | | |
| [LocationManager-12] | Check "Quản lý sản phẩm" page | 1 Actor logged in to system | 1. System show "Quản lý sản phẩm" view include elements: - Search area - List of products - "Thêm sản phẩm" button |  |
| [LocationManager-13] | Check "Thêm sản phẩm" page | 1. Actor clicks "Thêm sản phẩm" button | 1. System show "Thêm sản phẩm" view include elements: - "Ảnh" - "Tên sản phẩm" - "Giá" - "Loại" - "Mô tả" |  |
| [LocationManager-14] | Check add product when actor not input data | 1. Actor clicks "Thêm sản phẩm" button 2. Actor clicks "Thêm sản phẩm" button | 1. System show "Thêm sản phẩm" page 2. System show error messages: - "Vui lòng tải ảnh lên" - "Vui lòng nhập tên sản phẩm" - "Vui lòng nhập giá sản phẩm" |  |
| [LocationManager-15] | Check add product when actor add image and input valid data for all field | 1. Actor Input data like example: - Ảnh: image - Tên sản phẩm: thức ăn mèo me-o 350g - Giá: 23.000 - Loại: Đồ ăn - Mô tả: thức ăn từ hãng danh tiếng cho mèo 2. Actor clicks "Thêm sản phẩm" | 1. System show success message: - "Thêm sản phẩm thành công" 2. System direct to "Quản lý sản phẩm" page with added product at first line |  |
| [LocationManager-16] | Check edit product when actor not input data | 1. Actor clicks edit button 2. Actor delete all data 3. Actor clicks "Lưu sản phẩm" button | 1. System show "Sửa sản phẩm" page 2. System show error messages: - "Vui lòng tải ảnh lên" - "Vui lòng nhập tên sản phẩm" - "Vui lòng nhập giá sản phẩm" |  |
| [LocationManager-17] | Check edit product when actor add image and input valid data for all field | 1. Actor Input data like example: - Ảnh: image - Tên sản phẩm: thức ăn mèo me-o 350g - Giá: 23.000 - Loại: Đồ ăn - Mô tả: thức ăn từ hãng danh tiếng cho mèo 2. Actor clicks "Lưu sản phẩm" | 1. System show success message: - "Thêm sản phẩm thành công" 2. System direct to "Quản lý sản phẩm" page |  |
| [LocationManager-19] | Check search product when actor input keyword | 1. Actor Input data like example: - Tìm sản phẩm: "Đồ ăn" | 1. System show all products have name or type contain the keyword |  |
| [LocationManager-20] | Check delete product | 1. Actor clicks delete button 2. Actor cliks "Có" button | 1. System show confirm dialog with message: - "Bạn có chắc chắn muốn xóa sản phẩm này" 2. System display success message: - "Xóa sản phẩm thành công" |  |

Table 25:Location manager module integration test case

List of test case: (User module)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** |
| **Home view** | | | | |
| [User -1] | Check Home interface of logged in user | 1. Actor press an icon to run application | 1. Application show Home interface with elements: - "Home" title at the center of the top of the display. - Drawer icon at the left of title. The bell icon at the right of title. - 4 types of post + "Tất cả" + "Sự kiện" + "Buôn bán" + "Thảo luận" + "Hài hước" + "Tin tức" + "Địa điểm" - Some post of orther user with "Tất cả" type as default. | Actor logged in |
| [User -2] | Check drawer of logged in user | 1. Actor press an icon to run application 2. Actor press the drawer icon at the left top of display | 1. Application show Home interface  2. Application show the drawer include elements: - User icon - Home icon - Map icon - Pet icon - Bell icon - Speech bubble icon - Logout icon | Actor logged in |
| [User -3] | Check bell icon | 1. Actor press an icon to run application 2. Actor press the bell icon at the right top of display | 1. Application show Home interface  2. Show activites interface include emelents: - "Hoạt động" title at the top center. - Two tabs  + "Thông báo" + "Tin nhắn" - Display "Thông báo" tab as default. | Actor logged in |
| [User -4] | Check "Thông báo" tab | 1. Actor press an icon to run application 2. Actor press the bell icon at the right top of display 3. Actor pess "Thông báo" title | 1. Application show Home interface  2. Show activites interface. 3. Show notification relate with actor | Actor logged in |
| [User -5] | Check "Tin nhắn" tab | 1. Actor press an icon to run application 2. Actor press the bell icon at the right top of display 3. Actor pess "Tin nhắn" title | 1. Application show Home interface  2. Show activites interface. 3. Show all conversation | Actor logged in |
| [User -6] | Check bottom tab | 1. Actor press an icon to run application | 1. Application show Home interface with the bottom tab include elements: - Home icon - Search icon - Create post icon | Actor logged in |
| [User -7] | Check Create post icon | 1. Actor press an icon to run application 2. Actor press Create post icon at the bottom tab | 1. Application show Home interface. 2. Application show Create post interface include elements: - "Tạo bài viết" title - Drawer icon - Add icon - Six type of post - Text area - Camera icon - Image icon | Actor logged in |
| **Check Post** | | | | |
| [User -8] | Check create a post with valid information | 1.  - Actor choose type of post  - Actor fill content of post - Actor upload/take a picture - Actor press Add icon | 1. Application show "Đăng bài viết thành công" message and direct to Home interface | Actor logged in |
| [User -9] | Check create a post with blank content and image | 1. Actor press Add icon | 1. Application show "Bạn vui lòng nhập nội dung bài viết" message | Actor logged in |
| [User -10] | Check create a post with blank content | 1.  - Actor update a picture - Actor press Add icon | 1. Application show "Bạn vui lòng nhập nội dung bài viết" message | Actor logged in |
| [User -11] | Check create a post without image | 1.  - Actor fill content for text area - Actor press Add icon | 1. Application show "Bạn vui lòng thêm ảnh vào bài viết" | Actor logged in |
| [User -12] | Check edit post view | 1. Actor press to three dots icon 2. Actor press "Chỉnh sửa bài viết" | 1. Application show menu 2. Application show Edit Post interface include elements:  - "Chỉnh sửa bài viết" title - Drawer icon - Save icon - Six type of post - Text area - Camera icon - Image icon | Actor logged in |
| [User -13] | Check edit post with valid data | 1. Actor press to three dots icon 2. Actor press "Chỉnh sửa bài viết" 3. Actor fill data for all field and press save icon button | 1. Application show menu 2. Application show Edit Post interface 3. Show "Lưu thay đổi thành công" message | Actor logged in |
| [User -14] | Check edit post with blank title | 1. Actor press to three dots icon 2. Actor press Sửa bài viết" 3. Actor add images, delete content and press save icon button | 1. Application show menu 2. Application show Edit Post interface 3. Show "Bạn vui lòng nhập nội dung bài viết" message | Actor logged in |
| [User -15] | Check edit post with no image | 1. Actor press to three dots icon 2. Actor press "Chỉnh sửa bài viết" 3. Actor edit content, delete all images and press save icon button | 1. Application show menu 2. Application show Edit Post interface 3. Show "Bạn vui lòng thêm ảnh vào bài viết" message | Actor logged in |
| [User -16] | Check delete post | 1. Actor press to three dots icon 2. Actor press "Xóa bài viết" 3. Actor press "Có" button | 1. Application show menu 2. Application show confirm dialog 3. Show "Xóa bài viết thành công" message | Actor logged in |
| [User -17] | Check up vote post | 1. Actor press to like icon | 1. Like icon change and number of like increase 1 | Actor logged in |
| [User -18] | Check down vote post | 1. Actor press to dislike icon | 1. Like icon change and number of dislike increase 1 | Actor logged in |
| [User -19] | Check up vote when down voted post | 1. Actor press to like icon | 1. Like icon change and number of like increase 1 2. Dislike icon change and number of dislike decrease 1 | Actor logged in |
| [User -20] | Check down vote when up voted post | 1. Actor press to dislike icon | 1. Dislike icon change and number of dislike increase 1 2. Like icon change and number of like decrease 1 | Actor logged in |

Table 26:User module integration test case

List of test case: (Guest module)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** |
| **Home page** | | | | |
| [Guest-1] | Check drawer without log in | 1. Actor press an icon to run application 2. Actor press the drawer icon at the left top of display | 1. Application show Home interface  2. Application show the drawer include elements: - Login icon - Home icon - Map icon |  |
| [Guest-2] | Check Home page without log in | 1. Actor press an icon to run application | 1. Application show Home interface with elements: - "Home" title at the center of the top of the display. - Drawer icon at the left of title. - 4 types of post + "Tất cả" + "Sự kiện" + "Buôn bán" + "Thảo luận" + "Hài hước" + "Tin tức" + "Địa điểm" - Some posts of orther user with "Tất cả" type as default. |  |
| [Guest-3] | Check botttom bar without log in | 1. Actor press an icon to run application | 1. Application show Home interface with bottom bar include elements: - Home icon - Search icon - Add icon |  |
| [Guest-4] | Check Home icon | 1. Actor press an icon to run application 2. Actor press Home icon at the bottom tab | 1. Application show Home interface. 2. Application refresh Home interface. |  |
| [Guest-5] | Check Search icon | 1. Actor press an icon to run application 2. Actor press Search icon at the bottom tab | 1. Application show Home interface. 2. Application show Search interface include elements: - Search text field with "Gõ từ khóa bạn muốn tìm" as place holder - "Tìm" button |  |
| [Guest-6] | Check Add icon | 1. Actor press an icon to run application 2. Actor press Add icon at the bottom tab | 1. Application show Home interface. 2. Application show message "Bạn vui lòng đăng nhập để tạo bài viết" |  |
| [Guest-7] | View post by type | 1. Actor press an icon to run application 2. Actor press one of six types | 1. Application show Home interface  2. Application show post fit with the choosen type |  |
| [Guest-8] | Check view type have no post | 1. Actor press an icon to run application 2. Actor press one of six types | 1. Application show Home interface  2. Application show "Không có bài viết" |  |
| **Check Post** | | | | |
| [Guest-9] | View post at Home interface | 1. Actor press an icon to run application. 2. Actor wipes up. | 1.  - Application show Home interface - Post at Home interface iclude elements: + Post owner + Posted time + Three lines of content + Post images + Up vote icon + Speech bubble icon + Down vote icon 2. Application show more post |  |
| [Guest-10] | Check view post with long content | 1. Actor press an icon to run application. 2. Actor press "Xem thêm" | 1. Application show three first lines of post content 2. Application show all content of post |  |
| [Guest-11] | View post image detail | 1. Actor press an icon to run application 2. Actor press the image of post 3. Actor wipe to right 4. Actor wipe to left | 1. Application show Home interface  2. Application show the first image as default  3. Application show the next image 4. Application show the previous image |  |
| [Guest-12] | View post content detail | 1. Actor press an icon to run application 2. Actor press the content of post | 1. Application show Home interface  2. Application show all content of post |  |
| [Guest-13] | View comments in post | 1. Actor press an icon to run application 2. Actor press the speech bubble icon of post | 1. Application show Home interface  2.  - Application show the Post Detail interface. - Each comment is post include elements:  + Comment owner + Comment content + Commented time |  |
| [Guest-14] | Check view comment with long content | 1. Actor press an icon to run application 2. Actor press the speech bubble icon of post 3. Actor press "Đọc thêm" | 1. Application show Home interface  2.  - Application show the Post Detail interface. - Show three first lines of comment cotent 3. Application show all content of comment |  |
| **Check search** | | | | |
| [Guest-15] | Check search post with valid data | 1. Actor press Search icon at the bottom bar 2. - Actor type key word for search text field - Actor press "Tìm" button | 1. Application show Search interface 2. Show post has title or description contain the key word |  |
| [Guest-16] | Check search post with blank search text field | 1. Actor press Search icon at the bottom bar 2. Actor press "Tìm" button | 1. Application show Search interface 2. Application show message "Bạn vui lòng nhập từ khóa muốn tìm" |  |
| [Guest-17] | Check search location with valid data | 1. Actor press Search icon at the bottom bar 2. - Actor type key word for search text field - Actor press "Tìm" button | 1. Application show Search interface 2. Show location has name or description contain the key word |  |
| [Guest-18] | Check search location with blank search text field | 1. Actor press Search icon at the bottom bar 2. Actor press "Tìm" button | 1. Application show Search interface 2. Application show message "Bạn vui lòng nhập từ khóa muốn tìm" |  |
| [Guest-19] | Check direct from current location to choosen location | 1. Actor press to the location 2. Actor pess "Chỉ đường" button | 1. Application show information of location 2. Application show the shortest way from current location to choosen location on map |  |
| [Guest-20] | Check view information of location | 1. Actor press to the location on map | 1. Application show the location detail interface include elements: - Images - Name of location - Average rate star - Rate of orther user - "Chỉ đường" button |  |

Table 27:Guest module integration test case

For the full list of test cases and more details, see the file **PH\_Integration Test Case.xlsx**.



### System tests

The number of test cases is shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module** | **Pass** | **Fail** | **Untested** | **Total** |
| **Administrator** | 53 | 0 | 0 | **53** |
| **Location Manager** | 44 | 0 | 0 | **44** |
| **User** | 93 | 0 | 0 | **93** |
| **Total** | **173** | **0** | **0** | **173** |

Table 28:System test report

List of test case: (Administrator module)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** |
| **Login** | | | | |
| [Administrator-1] | Check view Login page | 1. Actor enters URL to view Login page. | 1. The Login page is displayed |  |
| [Administrator-2] | Check login with blank "Số điện thoại" and "Mật khẩu" field | 1. Actor access Login page 2. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại và mật khẩu không hợp lệ" | [Administrator-1] |
| [Administrator-3] | Check login with blank "Số điện thoại" field | 1. Actor access Login page 2. Actor input data for "Mật khẩu" field like: - "123456"  3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại và mật khẩu không hợp lệ" | [Administrator-1] |
| [Administrator-4] | Check login with blank "Mật khẩu" field | 1. Actor access Login page 2. Actor input data for "Số điện thoại" field like: - "0342256852"  3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại và mật khẩu không hợp lệ" | [Administrator-1] |
| [Administrator-5] | Check login with valid "Số điện thoại" but incorrect "Mật khẩu" | 1. Actor access Login page 2. Actor input data like: - Số điện thoại: 0342256852 - Mật khẩu: 1234567 3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Mật khẩu không đúng" | [Administrator-1] |
| [Administrator-6] | Check login with invalid "Số điện thoại" và "Mật khẩu" | 1. Actor access Login page 2. Actor input data like: - Số điện thoại: abc - Mật khẩu: 1234567 3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại không hợp lệ" | [Administrator-1] |
| [Administrator-7] | Check login when actor input correct data for "Số điện thoại" and "Mật khẩu" field | 1. Actor access Login page 2. Actor input data like: - Số điện thoại: 0342256852 - Mật khẩu: 123456  3. Actor click "Đăng nhập" button | 1. System display Login page 2. System direct to "Quản lý người dùng" page | [Administrator-1] |
| **User Management** | | | | |
| [Administrator-8] | Check "quản lý người dùng" view | 1. Actor logged in to system | 1. System display "Quản lý người dùng" page include elements: - "Danh sách người dùng" - "Tìm kiếm" | [Administrator-7] |
| [Administrator-9] | Check search user by name | 1. Actor logged in to system 2. Actor input keyword for "Tìm kiếm" like example: - "Lâ" | 1. System display "Quản lý người dùng" page  2. System display all user has name fit with inputted keyword | [Administrator-7] |
| [Administrator-10] | Check ban user | 1. Actor logged in to system 2. Actor clicks "Cấm" 3. Actor clicks "Có" | 1. System display "Quản lý người dùng" page  2. System show confirm dialog with message "Bạn có chắc muốn cấm người dùng này" 3. System display success message: - "Cấm người dùng thành công" 4. "Cấm" button change to "Bỏ cấm" button | [Administrator-7] |
| [Administrator-11] | Check unban user | 1. Actor logged in to system 2. Actor clicks "Bỏ cấm" 3. Actor clicks "Có" | 1. System display "Quản lý người dùng" page  2. System show confirm dialog with message "Bạn có chắc muốn bỏ cấm người dùng này" 3. System display success message: - "Bỏ cấm người dùng thành công" 4. "Bỏ cấm" button change to "Cấm" button | [Administrator-7] |
| **Category management** | | | | |
| **Post category management** | | | | |
| [Administrator-12] | Check view "Danh mục bài viết" page | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục" 3. Actor click "Bài viết" | 1. System display "Quản lý người dùng" page  2. System show menu include elements: - "Bài viết" - "Địa điểm" 3. System show "danh mục bài viết" page iclude elements: - Search area - Filter - "Thêm mới" button - Post category list | [Administrator-7] |
| [Administrator-13] | Check search category when actor not input data for "Tìm theo tên" | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục" 3. Actor click "Bài viết" | 1. System display "Quản lý người dùng" page  2. System show menu  3. System show "danh mục bài viết" page | [Administrator-7] |
| [Administrator-14] | Check search category when actor input data for "Tìm theo tên" | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục" 3. Actor click "Bài viết" 4. Actor input data like example: - Tìm theo tên: Đồ | 1. System display "Quản lý người dùng" page  2. System show menu  3. System show "danh mục bài viết" page  4. System show all post categorys have name include inputted keyword | [Administrator-7] |
| [Administrator-15] | Check filter post categorys when user check to checkbox | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục 3. Actor click "Bài viết" 4. Actor check to checkbox before "Hoạt động" | 1. System display "Quản lý người dùng" page  2. System show menu  3. System show "danh mục bài viết" page  4. System show all post categorys have status is show | [Administrator-7] |
| [Administrator-16] | Check filter post categorys when user check to checkbox | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục 3. Actor click "Bài viết" 4. Actor check to checkbox before "Không hoạt động" | 1. System display "Quản lý người dùng" page  2. System show menu  3. System show "danh mục bài viết" page  4. System show all post categorys have status is hide | [Administrator-7] |
| [Administrator-17] | Check add post category view | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục" 3. Actor click "Bài viết" 4. Actor clicks "Thêm mới" button | 1. System display "Quản lý người dùng" page  2. System show menu  3. System show "danh mục bài viết" page  4. System show add category dialog include elements: - "Tên danh mục" | [Administrator-7] |
| [Administrator-18] | Check add post category when actor not input "Tên danh mục" | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục" 3. Actor click "Bài viết" 4. Actor clicks "Thêm mới" button 5. Actor clicks "Lưu" button | 1. System display "Quản lý người dùng" page  2. System show menu  3. System show "danh mục bài viết" page  4. System show add category dialog  5. System display error message: - "Vui lòng nhập Tên danh mục" | [Administrator-7] |
| [Administrator-19] | Check add post category when actor input "Tên danh mục" | 1. Actor logged in to system 2. Actor clicks "Quản lý danh mục" 3. Actor click "Bài viết" 4. Actor clicks "Thêm mới" button 5. Actor input data like example: - Tên danh mục: Đồ ăn 6. Actor clicks "Lưu" button | 1. System display "Quản lý người dùng" page  2. System show menu  3. System show "danh mục bài viết" page  4. System show add category dialog  5. System display success message: - "Thêm danh mục thành công" 6. System display added category at the first line | [Administrator-7] |

Table 29:Administrator module system test case

List of test case: (Location manager module)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** |
| **Authentication** | | | | |
| **Register** | | | | |
| [Location Manager-1] | View Register screen | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom | 1. The Login page is displayed 2. System redirects the user to Register page include  - Số điện thoại field - "Gửi mã xác minh" button |  |
| [Location Manager-2] | Check actor register when "Số điện thoại" field blank | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor click "Gửi mã xác minh" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display "Vui lòng nhập số điện thoại để xác minh" message | [Location Manager-1] |
| [Location Manager-3] | Check actor register when valid phone number inputted | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number: 0342256852 4. Actor click "Gửi mã xác minh" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát" and display "Mã xác minh field", "Xác nhận" button | [Location Manager-1] |
| [Location Manager-4] | Check actor register with blank "Mã xác minh" field | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number 4. Actor click "Gửi mã xác minh" button 5. Actor pess "Xác nhận" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát" and display "Mã xác minh field", "Xác nhận" button  4. System display "Vui lòng nhập chính xác mã xác minh" message | [Location Manager-3] |
| [Location Manager-5] | Check actor register with wrong "Mã xác minh" field | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number 4. Actor click "Gửi mã xác minh" button 5. Actor pess "Xác nhận" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát" and display "Mã xác minh field", "Xác nhận" button  4. System display "Vui lòng nhập chính xác mã xác minh" message | [Location Manager-3] |
| [Location Manager-6] | Check actor register with valid "Mã xác minh" field | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number 4. Actor click "Gửi mã xác minh" button 5. Actor input verification code: 47261 (for example )  6. Actor pess "Xác nhận" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display:  - "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát"  And also display:  - "Mã xác minh field" - "Xác nhận" button  4. System display:  - "Vui lòng nhập chính xác mã xác minh" message 5. System direct to "Đăng ký cửa hàng" page include elements:  - Tên địa điểm - Mật khẩu - Xác nhận mật khẩu - Địa chỉ - Map | [Location Manager-3] |
| [Location Manager-7] | Check actor register with all blank field | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number 4. Actor click "Gửi mã xác minh" button 5. Actor input verification code: 47261 (for example )  6. Actor pess "Xác nhận" button 7. Actor press "Đăng ký" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display:  - "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát"  And also display:  - "Mã xác minh field" - "Xác nhận" button  4. System display:  - "Vui lòng nhập chính xác mã xác minh" message 5. System direct to "Đăng ký cửa hàng" page 6. System show error messages: - "Vui lòng điền địa điểm"  - "Vui lòng nhập mật khẩu - "Vui lòng nhập lại mật khẩu" - "Vui lòng chọn một địa điểm trên bản đồ" | [Location Manager-6] |
| [Location Manager-8] | Check actor register with "Mật khẩu" not fit with "Xác nhận mật khẩu" | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number 4. Actor click "Gửi mã xác minh" button 5. Actor input verification code: 47261 (for example )  6. Actor pess "Xác nhận" button 7. Actor input - "Trung's Store" for "Địa điểm" field - "123456" for "Mật khẩu" field - "1234567" for "Xác nhận mật khẩu" - "Linh Lang, Cống vị, Ba Đình, Hà Nội" for "Địa chỉ" field - Create a dot in map 8. Actor press "Đăng ký" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display:  - "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát"  And also display:  - "Mã xác minh field" - "Xác nhận" button  4. System display:  - "Vui lòng nhập chính xác mã xác minh" message 5. System direct to "Đăng ký cửa hàng" page 6. System show error message:  - "Mật khẩu không khớp" | [Location Manager-6] |
| [Location Manager-9] | Check actor register with valid data | 1. Actor enters URL to view Login page. 2. Actor clicks "Đăng ký" link at the bottom 3. Actor input phone number 4. Actor click "Gửi mã xác minh" button 5. Actor input verification code: 47261 (for example )  6. Actor pess "Xác nhận" button 7. Actor input - "Trung's Store" for "Địa điểm" field - "123456" for "Mật khẩu" field - "123456" for "Xác nhận mật khẩu" - "Linh Lang, Cống vị, Ba Đình, Hà Nội" for "Địa chỉ" field - Create a dot in map 8. Actor press "Đăng ký" button | 1. The Login page is displayed 2. System redirects the user to Register page 3. System display:  - "Mã xác minh đã được gửi đến bạn thành công, vui lòng đợi trong giây lát"  And also display:  - "Mã xác minh field" - "Xác nhận" button  4. System display:  - "Vui lòng nhập chính xác mã xác minh" message 5. System direct to "Đăng ký cửa hàng" page 6. System show success message:  - "Đăng ký thành công" and direct to Login page | [Location Manager-6] |
| **Login** | | | | |
| [Location Manager-10] | Check view Login page | 1. Actor signup the system successful | 1. System display success message: - "Đăng ký thành công" 2. System direct to Login page include elements: - Số điện thoại - Mật khẩu | [Location Manager-10] |
| [Location Manager-11] | Check login with blank "Số điện thoại" and "Mật khẩu" field | 1. Actor access Login page 2. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại và mật khẩu không hợp lệ" | [Location Manager-10] |
| [Location Manager-12] | Check login with blank "Số điện thoại" field | 1. Actor access Login page 2. Actor input data for "Mật khẩu" field like: - "123456"  3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại và mật khẩu không hợp lệ" | [Location Manager-10] |
| [Location Manager-13] | Check login with blank "Mật khẩu" field | 1. Actor access Login page 2. Actor input data for "Số điện thoại" field like: - "0342256852"  3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại và mật khẩu không hợp lệ" | [Location Manager-10] |
| [Location Manager-14] | Check login with valid "Số điện thoại" but incorrect "Mật khẩu" | 1. Actor access Login page 2. Actor input data like: - Số điện thoại: 0342256852 - Mật khẩu: 1234567 3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Mật khẩu không đúng" | [Location Manager-10] |
| [Location Manager-15] | Check login with invalid "Số điện thoại" và "Mật khẩu" | 1. Actor access Login page 2. Actor input data like: - Số điện thoại: abc - Mật khẩu: 1234567 3. Actor click "Đăng nhập" button | 1. System display Login page 2. System display error message:  "Số điện thoại không hợp lệ" | [Location Manager-10] |
| [Location Manager-16] | Check login when actor input correct data for "Số điện thoại" and "Mật khẩu" field | 1. Actor access Login page 2. Actor input data like: - Số điện thoại: 0342256852 - Mật khẩu: 123456  3. Actor click "Đăng nhập" button | 1. System display Login page 2. System direct to Home page | [Location Manager-10] |
| **Product managerment** | | | | |
| [Location Manager-17] | View "Quản lý sản phẩm" page | 1. Actor logged in to system | 1. Show "Quản lý sản phẩm" page include elements: - Search container - "Danh sách sản phẩm" | [Location Manager-16] |
| [Location Manager-18] | Check search product | 1. Actor logged in to system 2. Actor input key word for "Tìm sản phẩm" field | 1. System show "Quản lý sản phẩm" page 2. System show all products have data fit with inputed keyword | [Location Manager-16] |
| [Location Manager-19] | Check "Thêm sản phẩm" view | 1. Actor logged in to system 2. Actor click "Thêm sản phẩm" button | 1. System show "Quản lý sản phẩm" page 2. System direct to "Thêm sản phẩm" page include elements: - Product image - Product name - Price (in VND) - Type - Description | [Location Manager-16] |

Table 30:Location manager module system test case

List of test case: (User module)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test case Dependence** |
| **Common** | | | | |
| [User -1] | Check Home display view | 1. Actor open the application | 1. Application show Home display include elements: - Drawer - Top bar - Some posts - Bottom bar |  |
| [User -2] | Check load more post | 1. Actor open the application 2. Actor wipe up | 1. Application show Home display  2. Load more posts | [User -1] |
| [User -3] | Check search post view | 1. Actor open the application 2. Actor press the search icon at the bottom bar | 1. Application show Home display  2. Show search display include elements: - Search bar - All categories of post | [User -1] |
| [User -4] | Check search post | 1. Actor open the application 2. Actor press the search icon at the bottom bar 3. Actor input keyword like example: - Tìm kiếm: "thuốc" | 1. Application show Home display  2. Show search display 3. Application show some posts have content or type contain inputed keyword | [User -1] |
| **Authentication** | | | | |
| **Register** | | | | |
| [User -5] | Check register view | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page include elements: - Username - Password | [User -1] |
| [User -6] | Check register when actor leave user name and password blank | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon 4. Actor cliks "Đăng nhập" button | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page | [User -1] |
| [User -7] | Check register when actor input incorrect user and password | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon 4. Actor input data like example: - Username: 0342256749 - Password: hgfdhdfgh 5. Actor cliks "Đăng nhập" button | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page | [User -1] |
| [User -8] | Check register when user input correct data | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon 4. Actor input data like example: - Username: 0342256749 - Password: 123456 5. Actor cliks "Đăng nhập" button 6. Actor clicks "Tiếp tục" button | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page  4. Application display confirm view 5. Application direct to home display | [User -1] |
| **Login** | | | | |
| [User -9] | Check login view | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page include elements: - Username - Password | [User -8] |
| [User -10] | Check login when actor leave user name and password blank | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon 4. Actor cliks "Đăng nhập" button | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page | [User -8] |
| [User -11] | Check login when actor input incorrect user and password | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon 4. Actor input data like example: - Username: 0342256749 - Password: hgfdhdfgh 5. Actor cliks "Đăng nhập" button | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page | [User -8] |
| [User -12] | Check login when user input correct data | 1. Actor open the application 2. Actor clicks the collapsible button at the top left 3. Actor press login icon 4. Actor input data like example: - Username: 0342256749 - Password: 123456 5. Actor cliks "Đăng nhập" button 6. Actor clicks "Tiếp tục" button | 1. Application show Home display 2. Application display drawer 3. Application display Facebook login page  4. Application display confirm view 5. Application direct to home display | [User -8] |
| **Profile** | | | | |
| [User -13] | Check profile view | 1. Actor logged in to application 2. Actor clicks to collapsible button 3. Actor click profile icon | 1. Application show Home display 2. Application show drawer 3. Application show profile view include elements: - Cover image - Avatar - Display name - Some created post - "Three dot" icon | [User -12] |
| [User -14] | Check user information view | 1. Actor logged in to application 2. Actor clicks to collapsible button 3. Actor clicks profile icon 4. Actor clicks "Three dot" icon  5. Actor clicks "Xem thông tin" | 1. Application show Home display 2. Application show drawer 3. Application show profile view  4. Application show menu  5. Application show user information interface include elements: - Avatar - All information of user | [User -12] |
| [User -15] | Check edit information of user view | 1. Actor logged in to application 2. Actor clicks to collapsible button 3. Actor clicks profile icon 4. Actor clicks "Three dot" icon  5. Actor clicks "Thay đổi thông tin" | 1. Application show Home display 2. Application show drawer 3. Application show profile view  4. Application show menu  5. Application show "Chỉnh sửa thông tin" view include elements: - "Thay đổi tên hiển thị" - "Thay đổi Email" - "Thay đổi số điện thoại" - "Thay đổi mật khẩu" | [User -12] |
| [User -16] | Check edit "Tên hiển thị" when user not input data | 1. Actor logged in to application 2. Actor clicks to collapsible button 3. Actor clicks profile icon 4. Actor clicks "Three dot" icon  5. Actor clicks "Thay đổi thông tin" 6. Actor clicks "Thay đổi tên hiển thị" 7. Actor clicks save button | 1. Application show Home display 2. Application show drawer 3. Application show profile view  4. Application show menu  5. Application show "Chỉnh sửa thông tin" view  6. Application show "Sửa tên hiển thị" view include elements: - "Tên hiển thị" text field 7. Application show error message: - "Tên hiển thị không thể là khoảng trống" | [User -12] |
| [User -17] | Check edit "Tên hiển thị" when user input valid data | 1. Actor logged in to application 2. Actor clicks to collapsible button 3. Actor clicks profile icon 4. Actor clicks "Three dot" icon  5. Actor clicks "Thay đổi thông tin" 6. Actor clicks "Thay đổi tên hiển thị" 7. Actor input data like example: - Tên hiển thị: "Khánh Lâm"  8. Actor clicks save button | 1. Application show Home display 2. Application show drawer 3. Application show profile view  4. Application show menu  5. Application show "Chỉnh sửa thông tin" view  6. Application show "Sửa tên hiển thị" view include elements: - "Tên hiển thị" text field 7. Application show success message: - "Thay đổi tên hiển thị thành công" | [User -12] |
| [User -18] | Check edit email when user not input data | 1. Actor logged in to application 2. Actor clicks to collapsible button 3. Actor clicks profile icon 4. Actor clicks "Three dot" icon  5. Actor clicks "Thay đổi thông tin" 6. Actor clicks "Thay đổi email" 7. Actor clicks save button | 1. Application show Home display 2. Application show drawer 3. Application show profile view  4. Application show menu  5. Application show "Chỉnh sửa thông tin" view  6. Application show "Sửa email" view include elements: - Email text field 7. Application show error message: - "Email không thể là khoảng trống" | [User -12] |

Table 31:User module system test case

For the full list of test cases and more details, see the file **PH\_System Test Case.xlsx**.

# Chapter 6: Software user guide



## Deployment guidelines



### Setting server.

### Ubuntu on Virtual Private Server.

### MongoDB on Ubuntu.

In this guide, we will demonstrate how to install Postgres on an Ubuntu 16.04 VPS instance and go over some basic ways to use it.

* Import the public key used by the package management system.

sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4

* Create a list file for MongoDB.

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/4.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.0.list

* Reload local package database.

sudo apt-get update

* Install the MongoDB packages.

sudo apt-get install -y mongodb-org

Now that our software is installed, we can go over how it works and how it may be different from similar database management systems may have used.

Start MongoDB

sudo service mongod start

Accessing a MongoDB Prompt

mongo --host 139.162.77.151:27017

Create user

use admin

db.createUser(

{

user: " pethome",

pwd: "abc123",

roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]

}

)

### Nodejs on Ubuntu.

Our project will launch on Node Js platform. Node.js is a JavaScript platform for general-purpose programming that allows users to build network applications quickly. By leveraging JavaScript on both the front- and back-end, development can be more consistent and designed within the same system.

In order to get this version, we just have to use the apt package manager. We should refresh our local package index first, and then install from the repositories:

sudo apt-get update

sudo apt-get install nodejs

If the package in the repositories suits your needs, this is all you need to do to get set up with Node.js. In most cases, you'll also want to also install npm, which is the Node.js package manager. You can do this by typing:

sudo apt-get install npm

This will allow you to easily install modules and packages to use with Node.js. To check which version of Node.js you have installed after these initial steps, type:

node -v

### Nginx on Ubuntu.

* Install nginx

Nginx is one of the most popular web servers in the world and is responsible for hosting some of the largest and highest-traffic sites on the internet. It is more resource-friendly than Apache in most cases and can be used as a web server or a reverse proxy.

In this guide, we'll discuss how to get Nginx installed on your Ubuntu 16.04 server. we can install nginx:

npm sudo apt-get update

sudo apt-get install nginx

After accepting the procedure, apt-get will install Nginx and any required dependencies to your server.

* Config nginx for project

Start by creating and opening a new server block in Nginx's sites-available directory:

sudo nano /etc/nginx/sites-available/default

Here are config for vinakol.ga:

proxy\_http\_version 1.1;

proxy\_set\_header Connection "";

upstream pethome {

server 139.162.77.151:5000 max\_fails=0 fail\_timeout=10s weight=1;

ip\_hash;

keepalive 64;

}

server {

listen 80;

server\_name pethome;

location / {

# Set this to your upstream module.

proxy\_pass http://pethomefu;

# Headers to pass to proxy server.

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $host;

proxy\_set\_header X-NginX-Proxy true;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_cache\_bypass $http\_upgrade;

proxy\_http\_version 1.1;

proxy\_redirect off;

# Go to next upstream after if server down.

proxy\_next\_upstream error timeout http\_500 http\_502 http\_503 http\_504;

proxy\_connect\_timeout 5s;

# Gateway timeout.

proxy\_read\_timeout 20s;

proxy\_send\_timeout 20s;

# Buffer settings.

proxy\_buffers 8 32k;

proxy\_buffer\_size 64k;

}

access\_log off;

expires 1h;

}

}

### Pm2 on Ubuntu.

PM2 provides an easy way to manage and daemonize applications (run them in the background as a service).

We will use npm, a package manager for Node modules that installs with Node.js, to install PM2 on our server. Use this command to install PM2:

sudo npm install -g pm2

### Deploy project.

### Install Git.

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. We use git and github to manage and deploy code.

app-get install git-core

You may be asked to confirm the download and installation; simply enter y to confirm. It’s that simple, git should be installed and ready to use!

Confirm Git the installation. with the main installation done, first check to ensure the executable file is setup and accessible. The best way to do this is simply to run git with the version command.

git –version

### Deploy project.

Then we get project to deploy server by use this command (use authenticated github account):

git clone https://github.com/thangnvse/pet-home-be.git

Install node module by using command:

npm install

Run project with pm2: cd into root folder of project and run bellow the command

pm2 start npm –name “pethome” -i 0 -- run start

## User guide

### Location manager

Register

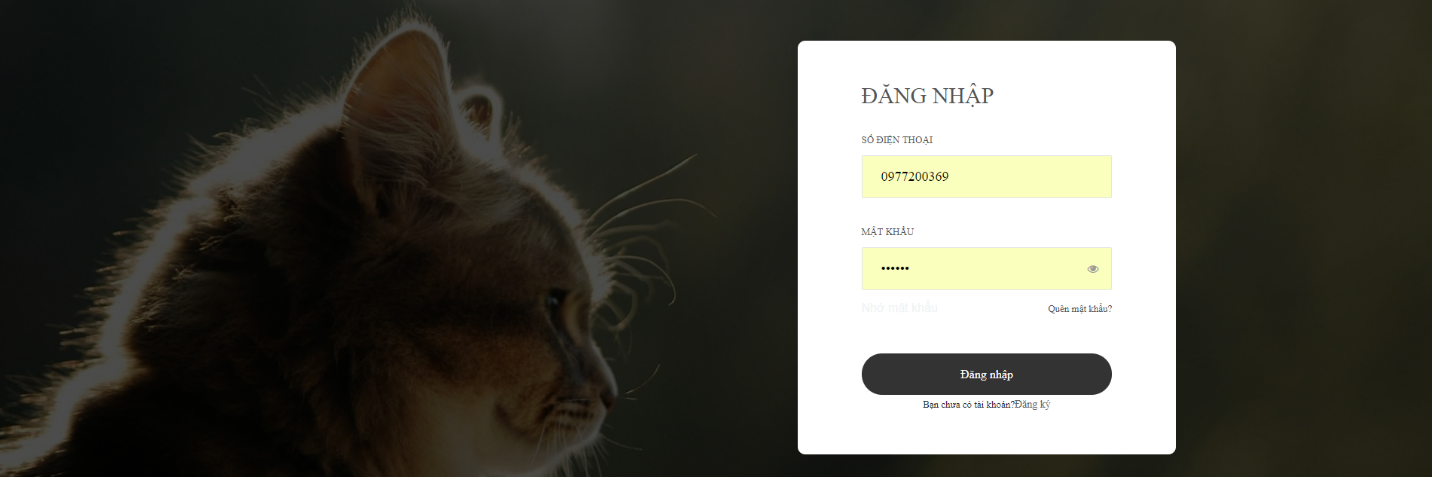


Image 79:Login page

* In Pet Home web application, user click on “Đăng ký” link.

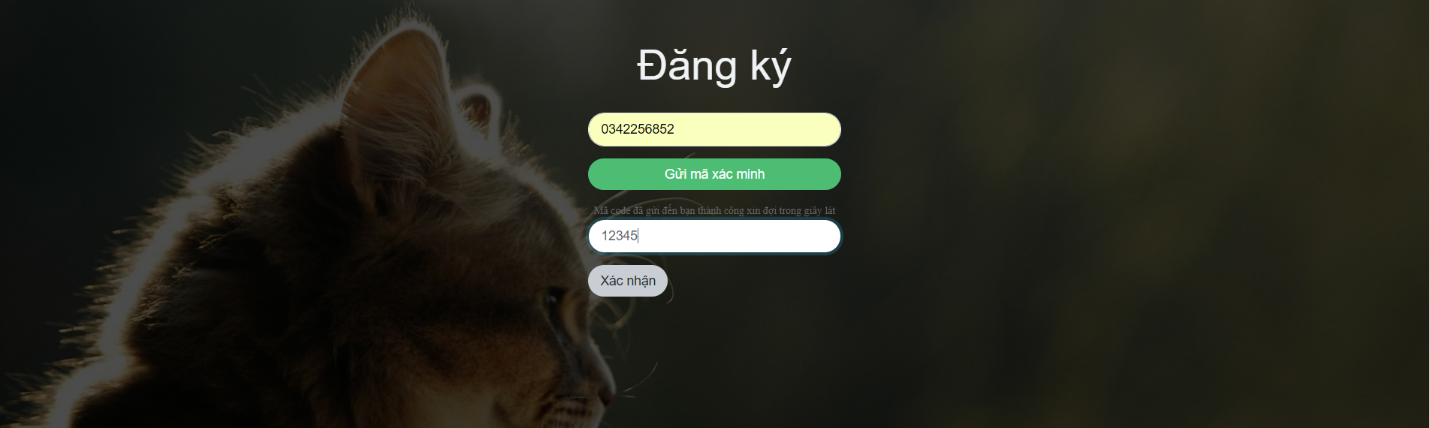


Image 80:Register page

* Input phone number
* Wait for second and input verification code



Image 81:Register page

* User fills information to text field
  + “Tên địa điểm” field: name of location
  + “Mật khẩu” field: password
  + “Xác nhận mật khẩu” field: re-enter password
  + “Địa chỉ chi tiết” field: location’s address detail
  + “Cửa hàng thú cưng” combobox: type of location
  + Create a dot in map
  + Press “Đăng ký” button

### User

Register

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(1).jpg  Step 1: Press to the collapsible icon | C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(2).jpg  Step 2: Clicks to rounded icon |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(3).jpg  Step 3: Input Facebook account for first time to register  Add post   |  |  | | --- | --- | | C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(1).jpg  Step 1: Press to rounded icon | C:\Users\Asus\Dropbox\SWP490\Guide\UserAddPost(1).jpg  Step 2:   * Choose a type of post * Input content of post * Upload at least one image * Press the rounded icon | | |

Search post

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(1).jpg  Step 1: Press to search icon | C:\Users\Asus\Dropbox\SWP490\Guide\UserSearchPost.jpg  Step 2: Type keyword to search |

Comment post

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(1).jpg  Step 1: Press to speech bubble | C:\Users\Asus\Dropbox\SWP490\Guide\UserCommentPost.jpg  Step 2: Type content and press the air plane icon |

View notification

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Guide\HomeApp.jpg  Step 1: Press the bell icon | C:\Users\Asus\Dropbox\SWP490\Guide\UserNotice.jpg  Step 2: Notification |

Add pet

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(2).jpg  Step 1: Press pet icon | C:\Users\Asus\Dropbox\SWP490\Guide\UserAddPet.jpg  Step 2: Press “+” icon |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserAddPet(2).jpg  Step 3: Fill all information and press “+” icon | |

Chat

|  |  |
| --- | --- |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserRegister(2).jpg  Step 1: Press speech bubble icon | C:\Users\Asus\Dropbox\SWP490\Guide\UserChat(1).jpg  Step 2: Pick one user to chat |
| C:\Users\Asus\Dropbox\SWP490\Guide\UserChat(2).jpg  Step 3: Enter content and press “Send” | |

# References

**Technology:**

[1] Nodejs - <https://nodejs.org/en/>

[2] Express - <https://expressjs.com/>

[3] Passport - <http://www.passportjs.org/>

[4] socket.io - https://socket.io/

[5] React - <https://reactjs.org/>

[6] React Native - <https://facebook.github.io/react-native/>

[7] Expo - https://expo.io/

[8] MongoDB - <https://www.mongodb.com/>

[9] Mongoose - <https://mongoosejs.com>

[10] Git - <https://git-scm.com/doc>

**Document:**

[11] UML - https://www.ibm.com/developerworks/rational/library/3101.html

[12] Data Modeling Introduction - <https://docs.mongodb.com/v3.0/core/data-modeling-introduction/>