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

**FPT UNIVERSITY**

|  |  |
| --- | --- |
| Capstone Project Document | |
| **Scrap Collector** | |
| **Group 3 - JS** | |
| **Group members** | Nguyễn Duy Đạt – SE63093 (Leader)  Tiết Lê Bảo Khánh – SE62247  Nguyễn Hoàng Anh Trung – SE62282  Nguyễn Hoàng Nhân – SE63259 |
| **Supervisor** | Lại Đức Hùng |
| **Ext. Supervisor** | N/A |
| **Project Code** | SC |

– **Ho Chi Minh City, 13 January, 2020** –

**Table of contents**

[A. Introduction 15](#_Toc35157074)

[1. Project Information 15](#_Toc35157075)

[2. Introduction 15](#_Toc35157076)

[3. Current Situation 15](#_Toc35157077)

[4. Problem Definition 15](#_Toc35157078)

[5. Proposed Solution 16](#_Toc35157079)

[6. Functional Requirements 16](#_Toc35157080)

[7. Role & Responsibility 17](#_Toc35157081)

[B. Software Project Management Plan 17](#_Toc35157082)

[1. Problem Definition 17](#_Toc35157083)

[2. Project Organization 22](#_Toc35157084)

[3. Project Management Plan 24](#_Toc35157085)

[4. Coding Convention 24](#_Toc35157086)

[C. Software Requirement Specification 25](#_Toc35157087)

[1. User Requirement Specification 25](#_Toc35157088)

[2. System Requirement Specification 26](#_Toc35157089)

[3. Software System Attribute 78](#_Toc35157090)

[4. Conceptual Diagram 79](#_Toc35157091)

[D. Software Design Description 80](#_Toc35157092)

[1. Design Overview 80](#_Toc35157093)

[2. System Architectural Design 81](#_Toc35157094)

[3. Component Diagram 83](#_Toc35157095)

[4. Detailed Description 83](#_Toc35157096)

[5. User Interface Design 111](#_Toc35157097)

[6. Database Design 114](#_Toc35157098)

[7. Algorithms 120](#_Toc35157099)

[E. System Implementation & Test 120](#_Toc35157100)

[1. Introduction 120](#_Toc35157101)

[2. Data Relationship Diagram 121](#_Toc35157102)

[3. Test Plan 123](#_Toc35157103)

[3.1. Features To Be Tested 123](#_Toc35157104)

[3.2. Features Not To Be Tested 123](#_Toc35157105)

[4. System Testing Test Case 123](#_Toc35157106)

[F. Software User’s Manual 137](#_Toc35157107)

**List of tables**

[Table 1 - Role & Responsibility 18](#_Toc33900637)

[Table 2 - Hardware Requirements for Server 22](#_Toc33900638)

[Table 3 - Hardware requirement for PC 22](#_Toc33900639)

[Table 4 - Software Requirement 22](#_Toc33900640)

[Table 5 - Roles & Responsibilities Details 24](#_Toc33900641)

[Table 6 - Tools and techniques 25](#_Toc33900642)

[Table 7 **-** <Guest> Login use case specification 31](#_Toc33900643)

[Table 8 - <Guest> Register use case specification 33](#_Toc33900644)

[Table 9 - <Authorized User> Logout use case specification 35](#_Toc33900645)

[Table 10 - <Authorized User> View profile use case specification 36](#_Toc33900646)

[Table 11 - <Authorized User> Update profile use case specification 37](#_Toc33900647)

[Table 12 - <Authorized user> View notification use case specification 39](#_Toc33900648)

[Table 13 - <Homeowner> Post scrap use case specification 41](#_Toc33900649)

[Table 14 - <Homeowner> Chat with collector use case specification 42](#_Toc33900650)

[Table 15 - <Homeowner> Call collector use case specification 43](#_Toc33900651)

[Table 16 - <Homeowner> View booking request use case specification 44](#_Toc33900652)

[Table 17 - <Homeowner> Accept booking request use case specification 45](#_Toc33900653)

[Table 18 - <Homeowner> Reject booking request use case specification 46](#_Toc33900654)

[Table 19 - <Homeowner> Update post information use case specification 47](#_Toc33900655)

[Table 20 - <Homeowner> View post history use case specification 48](#_Toc33900656)

[Table 21 - <Homeowner> Delete post use case specification 49](#_Toc33900657)

[Table 22 - <Homeowner> Accomplish post use case specification 50](#_Toc33900658)

[Table 23 - <Homeowner> Review collector use case specification 51](#_Toc33900659)

[Table 24 - <Collector> Search near-by scraps use case specification 53](#_Toc33900660)

[Table 25 - <Collector> View scrap detail use case specification 54](#_Toc33900661)

[Table 26 - <Collector> Book scrap use case specification 55](#_Toc33900662)

[Table 27 - <Collector> Call homeowner use case specification 57](#_Toc33900663)

[Table 28 - <Collector> Chat with homeowner use case specification 58](#_Toc33900664)

[Table 29 - <Collector> View pick-up schedule use case specification 59](#_Toc33900665)

[Table 30 - <Collector> View direction use case specification 60](#_Toc33900666)

[Table 31 - <Collector> View booking response use case specification 61](#_Toc33900667)

[Table 32 - <Collector> View booking scrap list use case specification 62](#_Toc33900668)

[Table 33 - <Collector> Cancel booking use case specification 63](#_Toc33900669)

[Table 34 - <Collector> View review from Homeowner use case specification 64](#_Toc33900670)

[Table 35 - <Admin> View list of account use case specification 65](#_Toc33900671)

[Table 36 - <Admin> View account detail use case specification 66](#_Toc33900672)

[Table 37 - <Admin> View list of post use case specification 67](#_Toc33900673)

[Table 38 - <Admin> View post detail use case specification 68](#_Toc33900674)

[Table 39 - <Admin> Activate category use case specification 69](#_Toc33900675)

[Table 40 - <Admin> Deactivate category use case specification 70](#_Toc33900676)

[Table 41 - <Admin> Add new category use case specification 71](#_Toc33900677)

[Table 42 - <Admin> View category list use case specification 73](#_Toc33900678)

[Table 43 - Conceptual Diagram Data Dictionary 74](#_Toc33900679)

[Table 44 - Class Diagram Dictionary 79](#_Toc33900680)

[Table 45 - <Class Diagram Attributes> Account 79](#_Toc33900681)

[Table 46 **-** <Class Diagram Methods> Account 79](#_Toc33900682)

[Table 47 <Class Diagram Attributes> Role 80](#_Toc33900683)

[Table 48 - <Class Diagram Methods> Role 80](#_Toc33900684)

[Table 49 <Class Diagram Methods> Homeowner 80](#_Toc33900685)

[Table 50 - <Class Diagram Methods> Dictionary 80](#_Toc33900686)

[Table 51 - <Class Diagram Methods> Admin 81](#_Toc33900687)

[Table 52 <Class Diagram Attributes> Warningword 81](#_Toc33900688)

[Table 53 <Class Diagram Methods> Warningword 81](#_Toc33900689)

[Table 54 <Class Diagram Attributes> Topic 81](#_Toc33900690)

[Table 55 <Class Diagram Methods> Topic 81](#_Toc33900691)

[Table 56 <Class Diagram Attributes> Page 82](#_Toc33900692)

[Table 57 <Class Diagram Methods> Page 83](#_Toc33900693)

[Table 58 <Class Diagram Attributes> Version 83](#_Toc33900694)

[Table 59 <Class Diagram Attributes> GrammarReport 83](#_Toc33900695)

[Table 60 <Class Diagram Attributes> SpellingReport 84](#_Toc33900696)

[Table 61 <Class Diagram Methods> SpellingReport 84](#_Toc33900697)

[Table 62 <Class Diagram Attributes> WarningWordReport 84](#_Toc33900698)

[Table 63 <Class Diagram Attributes> PageReport 85](#_Toc33900699)

[Table 64 <Class Diagram Attributes> ContactReport 85](#_Toc33900700)

[Table 65 <Class Diagram Attributes> BrokenLinkReport 86](#_Toc33900701)

[Table 66 <Class Diagram Attributes> RedirectionReport 86](#_Toc33900702)

[Table 67 <Class Diagram Attributes> BrokenPageReport 86](#_Toc33900703)

[Table 68 <Class Diagram Attributes> MissingFileReport 87](#_Toc33900704)

[Table 69 <Class Diagram Attributes> FaviconReport 87](#_Toc33900705)

[Table 70 <Class Diagram Attributes> SpeedReport 88](#_Toc33900706)

[Table 71 <Class Diagram Attributes> JavascriptReport 88](#_Toc33900707)

[Table 72<Class Diagram Attributes> MobileLayoutReport 88](#_Toc33900708)

[Table 73<Class Diagram Attributes> CookieReport 89](#_Toc33900709)

[Table 74 <Fields> Login 95](#_Toc33900710)

[Table 75 <Buttons/Hyperlinks> Login 95](#_Toc33900711)

[Table 76 <Buttons/Hyperlinks> Menu Manage 96](#_Toc33900712)

[Table 77 <Fields> Manage Staff 97](#_Toc33900713)

[Table 78 <Buttons/Hyperlinks> Manage Staff 97](#_Toc33900714)

[Table 79 <Fields> Manage Website 97](#_Toc33900715)

[Table 80 <Buttons/Hyperlinks> Manage Website 98](#_Toc33900716)

[Table 81 <Buttons/Hyperlinks> Menu Staff 99](#_Toc33900717)

[Table 82 <Fields> Sitemap 100](#_Toc33900718)

[Table 83 <Buttons/Hyperlinks> Sitemap 100](#_Toc33900719)

[Table 84 <Fields> Spelling Test 100](#_Toc33900720)

[Table 85 <Buttons/Hyperlinks> Grammar Test 101](#_Toc33900721)

[Table 86 <Fields> Grammar Test 101](#_Toc33900722)

[Table 87 <Buttons/Hyperlinks> Grammar Test 101](#_Toc33900723)

[Table 88 <Fields> Broken Pages Test 102](#_Toc33900724)

[Table 89 <Buttons/Hyperlinks> Broken Pages Test 102](#_Toc33900725)

[Table 90 <Fields> Broken Links Test 103](#_Toc33900726)

[Table 91 <Buttons/Hyperlinks> Broken Links Test 103](#_Toc33900727)

[Table 92 <Fields> Missing File Test 104](#_Toc33900728)

[Table 93 <Buttons/Hyperlinks> Missing File Test 104](#_Toc33900729)

[Table 94 <Fields> Prohibited Content Test 105](#_Toc33900730)

[Table 95 <Buttons/Hyperlinks> Prohibited Content Test 105](#_Toc33900731)

[Table 96 <Fields> Mobile Layout Test 105](#_Toc33900732)

[Table 97 <Buttons/Hyperlinks> Mobile Layout Test 106](#_Toc33900733)

[Table 98 <Fields> Speed Test 106](#_Toc33900734)

[Table 99 <Buttons/Hyperlinks> Speed Test 107](#_Toc33900735)

[Table 100 <Fields> JavaScript Errors Test 107](#_Toc33900736)

[Table 101 <Buttons/Hyperlinks> JavaScript Errors Test 107](#_Toc33900737)

[Table 102 <Fields> Favicon Test 108](#_Toc33900738)

[Table 103 <Buttons/Hyperlinks> Favicon Test 108](#_Toc33900739)

[Table 104 <Fields> Cookie Test 109](#_Toc33900740)

[Table 105 <Buttons/Hyperlinks> Cookie Test 109](#_Toc33900741)

[Table 106 <Fields> Server Behavior 109](#_Toc33900742)

[Table 107 <Buttons/Hyperlinks> Server Behavior 110](#_Toc33900743)

[Table 108 <Fields> Contact Details 110](#_Toc33900744)

[Table 109 <Buttons/Hyperlinks> Contact Details Test 111](#_Toc33900745)

[Table 110 <Fields> Redirection Test 111](#_Toc33900746)

[Table 111 <Buttons/Hyperlinks> Redirection Test 111](#_Toc33900747)

[Table 112 <Fields> Page Test 112](#_Toc33900748)

[Table 113 <Buttons/Hyperlinks> Page Test 112](#_Toc33900749)

[Table 114 Physical Diagram Dictionary 118](#_Toc33900750)

[Table 115 <Test case> Manage Page Option 119](#_Toc33900751)

[Table 116<Test case> Broken Links Test 120](#_Toc33900752)

[Table 117 <Test case> Broken Pages Test 121](#_Toc33900753)

[Table 118 <Test case> Prohibited Content Test 122](#_Toc33900754)

[Table 119 <Test case> Mobile Layout Test 122](#_Toc33900755)

[Table 120 <Test case> Speed Test 123](#_Toc33900756)

[Table 121 <Test case> Javascript Errors Test 124](#_Toc33900757)

[Table 122 <Test case> Cookie Law Test 125](#_Toc33900758)

[Table 123 <Test case> Server Behavior Test 126](#_Toc33900759)

[Table 124 <Test case> Missing File Test 127](#_Toc33900760)

[Table 125 <Test case> Favicon Test 128](#_Toc33900761)

[Table 126 <Test case> Contact Details Test 129](#_Toc33900762)

[Table 127 <Test case> Redirection Test 129](#_Toc33900763)

[Table 128 <Test case> Pages Test 130](#_Toc33900764)

[Table 129 <Test case> Spelling Test 131](#_Toc33900765)

[Table 130 <Test case> Grammar Test 132](#_Toc33900766)

[Table 131 - Hardware requirement for server 133](#_Toc33900767)

[Table 132 - Software requirement for server 134](#_Toc33900768)

[Table 133 - Add Staff Steps 136](#_Toc33900769)

[Table 134 - Edit Staff Steps 136](#_Toc33900770)

[Table 135 - Delete Staff Steps 137](#_Toc33900771)

[Table 136 - Make new version Steps 138](#_Toc33900772)

[Table 137 - View Sitemap Steps 138](#_Toc33900773)

[Table 138 - Assign website steps 139](#_Toc33900774)

[Table 139 - Add warning word steps 140](#_Toc33900775)

[Table 140 - Edit warning word steps 142](#_Toc33900776)

[Table 141 - Select website steps 142](#_Toc33900777)

[Table 142 - Add page option steps 144](#_Toc33900778)

[Table 143 - Rename or Delete page option steps 145](#_Toc33900779)

[Table 144 - Do a test function steps 146](#_Toc33900780)

**List of Figures**

[Figure 1– Software process model 23](#_Toc33726577)

[Figure 2 – Use Case Diagram 29](#_Toc33726578)

[Figure 3 - Guest Overview Use Case 30](#_Toc33726579)

[Figure 4 - <Guest> Login 30](#_Toc33726580)

[Figure 5 - <Guest> Register 31](#_Toc33726581)

[Figure 6 - Authorized User Overview Use Case 34](#_Toc33726582)

[Figure 7 - <Authorized User> Logout 34](#_Toc33726583)

[Figure 8 - <Authorized User> View profile 35](#_Toc33726584)

[Figure 9 - <Authorized User> Update profile 36](#_Toc33726585)

[Figure 10 - <Authorized User> De-activate account 38](#_Toc33726586)

[Figure 11 - Homeowner Overview Use Case 39](#_Toc33726587)

[Figure 12 - <Homeowner> Post scrap 39](#_Toc33726588)

[Figure 13 - <Homeowner> Chat with collector 41](#_Toc33726589)

[Figure 14 - <Homeowner> Call collector 42](#_Toc33726590)

[Figure 15- <Homeowner> View booking request 43](#_Toc33726591)

[Figure 16 - <Homeowner> Accept booking request 44](#_Toc33726592)

[Figure 17 - <Homeowner> Reject booking request 45](#_Toc33726593)

[Figure 18 - <Homeowner> Update post information 46](#_Toc33726594)

[Figure 19 - <Homeowner> View post history 47](#_Toc33726595)

[Figure 20 - <Homeowner> Delete post 48](#_Toc33726596)

[Figure 21 - <Homeowner> Accomplish post 49](#_Toc33726597)

[Figure 22 - <Homeowner> Review collector 50](#_Toc33726598)

[Figure 23 - Collector Overview Use Case 52](#_Toc33726599)

[Figure 24 - <Collector> Search near-by scraps 52](#_Toc33726600)

[Figure 25 - <Collector> View scrap detail 53](#_Toc33726601)

[Figure 26 - <Collector> Book scrap 55](#_Toc33726602)

[Figure 27 - <Collector> Call homeowner 56](#_Toc33726603)

[Figure 28 - <Collector> Chat with homeowner 57](#_Toc33726604)

[Figure 29 - <Collector> View pick-up schedule 58](#_Toc33726605)

[Figure 30 - <Collector> View direction 59](#_Toc33726606)

[Figure 31 - <Collector> View booking response 60](#_Toc33726607)

[Figure 32 - <Collector> View booking scrap list 61](#_Toc33726608)

[Figure 33 - <Collector> Cancel booking 62](#_Toc33726609)

[Figure 34 - <Collector> View review from Homeowner 63](#_Toc33726610)

[Figure 35 – Admin Overview Use Case 64](#_Toc33726611)

[Figure 36 - <Admin> Activate account 64](#_Toc33726612)

[Figure 37 - <Admin> De-activate account 65](#_Toc33726613)

[Figure 38 - <Admin> Activate post 66](#_Toc33726614)

[Figure 39 - <Admin> Deactivate post 67](#_Toc33726615)

[Figure 40 - <Admin> Activate category 68](#_Toc33726616)

[Figure 41 - <Admin> Deactivate category 69](#_Toc33726617)

[Figure 42 - <Admin> Add new category 70](#_Toc33726618)

[Figure 43 - Conceptual Diagram 72](#_Toc33726619)

[Figure 44 Class Diagram 75](#_Toc33726620)

[Figure 45 <Sequence Diagram> Guest Login 88](file:///D:\Capstone_JS_Spring2020_FE_Doc\Document\Scrap_Collector_Full_Doc.docx#_Toc33726621)

[Figure 46 <Sequence Diagram> Staff Test Pages 88](#_Toc33726622)

[Figure 47 <Sequence Diagram> Staff Test Favicon 88](#_Toc33726623)

[Figure 48<Sequence Diagram> Staff Test Contact Details 88](#_Toc33726624)

[Figure 49 <Sequence Diagram> Staff Test Redirection 88](#_Toc33726625)

[Figure 50 <Sequence Diagram> Staff Test Missing File 90](#_Toc33726626)

[Figure 51 <Sequence Diagram> Manager Build Sitemap 91](#_Toc33726627)

[Figure 52 <Sequence Diagram> Staff Test Cookie 91](#_Toc33726628)

[Figure 53 <Sequence Diagram> Staff Test Broken Pages 91](#_Toc33726629)

[Figure 54 <Sequence Diagram> Staff Test Broken Links 91](#_Toc33726630)

[Figure 55 <Sequence Diagram> Staff Test Mobile Layout 91](#_Toc33726631)

[Figure 56 <Sequence Diagram> Staff Test Javascript Errors 92](#_Toc33726632)

[Figure 57 <Sequence Diagram> Staff Test Prohibited 92](#_Toc33726633)

[Figure 58 <Sequence Diagram> Staff Test Spelling 92](#_Toc33726634)

[Figure 59 <Sequence Diagram> Staff Test Grammar 92](#_Toc33726635)

[Figure 60 <Sequence Diagram> Staff Test Server Behavior 92](#_Toc33726636)

[Figure 61 <Sequence Diagram>Staffmeasure the speed test 92](#_Toc33726637)

[Figure 62 Login 93](#_Toc33726638)

[Figure 63 Menu Manager 94](#_Toc33726639)

[Figure 64 Manage Staff 94](#_Toc33726640)

[Figure 65 Manage Website 95](#_Toc33726641)

[Figure 66 Menu Staff 96](#_Toc33726642)

[Figure 67 Sitemap 97](#_Toc33726643)

[Figure 68 Spelling Test 98](file:///D:\Capstone_JS_Spring2020_FE_Doc\Document\Scrap_Collector_Full_Doc.docx#_Toc33726644)

[Figure 69 Grammar Test 99](#_Toc33726645)

[Figure 70 Broken Pages Test 99](#_Toc33726646)

[Figure 71 Broken Links Test 100](#_Toc33726647)

[Figure 72 Missing File Test 101](#_Toc33726648)

[Figure 73 Prohibited content test 102](#_Toc33726649)

[Figure 74 Mobile Layout Test 103](#_Toc33726650)

[Figure 75 Speed Test 104](#_Toc33726651)

[Figure 76 - JavaScript Errors Test 105](#_Toc33726652)

[Figure 77 Favicon Test 105](#_Toc33726653)

[Figure 78 Cookie Test 106](#_Toc33726654)

[Figure 79 Server Behavior 107](#_Toc33726655)

[Figure 80 - Contact Details Test 108](#_Toc33726656)

[Figure 81 - Redirection Test 109](#_Toc33726657)

[Figure 82 Page Test 109](#_Toc33726658)

[Figure 83 Entity Relationship Diagram 111](#_Toc33726659)

[Figure 84 Physical Diagram 118](#_Toc33726660)

[Figure 85 - Add Staff UI 138](#_Toc33726661)

[Figure 86 - Add Staff Popup 138](file:///D:\Capstone_JS_Spring2020_FE_Doc\Document\Scrap_Collector_Full_Doc.docx#_Toc33726662)

[Figure 87 - Edit Staff UI 139](#_Toc33726663)

[Figure 88 - Edit Staff Popup 139](#_Toc33726664)

[Figure 89 - Delete Staff UI 140](#_Toc33726665)

[Figure 90 - Delete Staff Popup 140](#_Toc33726666)

[Figure 91 - Make new version UI 140](#_Toc33726667)

[Figure 92 - View Sitemap UI 141](#_Toc33726668)

[Figure 93 - Sitemap Screen 141](#_Toc33726669)

[Figure 94 - Assign website UI 142](#_Toc33726670)

[Figure 95 - Assign website popup 142](#_Toc33726671)

[Figure 96 - Assign website popup 142](#_Toc33726672)

[Figure 97 - Add warning word UI 143](#_Toc33726673)

[Figure 98 - Add warning word popup 143](#_Toc33726674)

[Figure 99 - Add warning word popup 143](#_Toc33726675)

[Figure 100 - Edit warning word UI 144](#_Toc33726676)

[Figure 101 - Edit warning word popup 144](file:///D:\Capstone_JS_Spring2020_FE_Doc\Document\Scrap_Collector_Full_Doc.docx#_Toc33726677)

[Figure 102 - Edit warning word popup 144](file:///D:\Capstone_JS_Spring2020_FE_Doc\Document\Scrap_Collector_Full_Doc.docx#_Toc33726678)

[Figure 103 - Select website UI 145](#_Toc33726679)

[Figure 104 - Select page option UI 145](#_Toc33726680)

[Figure 105 - Add page option popup 146](#_Toc33726681)

[Figure 106 - Add page option popup 146](#_Toc33726682)

[Figure 107 - Add page option popup 146](#_Toc33726683)

[Figure 108 - Select page option UI 147](#_Toc33726684)

[Figure 109 - Select page option popup 147](#_Toc33726685)

[Figure 110 - Rename or Delete page option popup 148](#_Toc33726686)

[Figure 111 - Select page option UI 148](#_Toc33726687)

[Figure 112 - Select page option popup 149](#_Toc33726688)

[Figure 113 - Test function UI 149](#_Toc33726689)

# Introduction

## Project Information

* Project name:  **Scrap Collector**
* Project Code: **SC**
* Product Type: **Mobile Application, Website Application**
* Start Date: **January 13rd, 2020**
* End Date: **--**

## Introduction

In this document, we introduce a new solution for those who are doing the job as collecting scraps. Scrap is understood as unusable materials such as iron, aluminum, plastic bottles, cans, etc… which can be recycled. Current job of the scrap collectors is to go (is goes) around the place where they are living and ask if anybody has scrap to sell. That (This) traditional way is such an inconvenient way for both the collector and the homeowner.

Based on our researches and analysis (analyses), we came (come) to a solution for scrap collector and people who want to sell scrap in Viet Nam (Vietnam). We build a system which help the scrap collectors to find near-by scrap posted in the defined radius of 5km or 10km, and based on those location, we help the collector to find out the smartest way to go and collect all the scrap that they have dealt. Beside of that, we also provide a mobile application that help homeowner who want to sell scrap to post scrap and contact the collector easily.

## Current Situation

Now, there is no mobile application that helps both homeowner and collector to organize their work in a smarter way yet. Every day, the collector go around his/her living place to collect scrap, and the homeowner have to wait for the collector. Some large scrap collector service post their contact information onto websites, Facebook pages or E-commerce websites for those who want to sell scrap can contact. The homeowner can call and require the scrap collector service to go to their house or can bring their scrap to the address of that service.

## Problem Definition

Below are the disadvantages of the current situation:

* The collector spends all day outside but cannot find any scrap.
* The collector forgets to go to collect the scrap that has been dealt with the customer. (Collectors forget collecting discarded materials that have been disposed of to customers.)
* The collector cannot optimize his/her routes.
* The homeowner has to wait for the scrap collector to pass by his/her house and (then) call the collector.

## Proposed Solution

Building a mobile application known as a scrap collector helping system named “Scrap Collector”. It helps scrap collectors to enhance their productivity in daily work. The following workflow and features should be implemented:

#### 5.1 Feature functions

* Post scrap and request collector to pick-up
* Search for near-by scraps
* Book scrap
* Negotiate scrap price for better benefit
* Get a pick-up schedule that is optimized based on location and time
* Review collector
* User can contact to others through calling or chatting

#### 5.2 Advantages and disadvantages

The advantages and disadvantages of the proposed solution:

* **Advantages**:
  + More effective way for the collector to collect scraps
  + The good interaction between the homeowner and the collector
  + Support the homeowner to sell scraps for further recycling instead of throwing scrap
  + Save time and effort for both homeowner and collector
* **Disadvantages**:
  + Only support for scrap collectors in Viet Nam (Vietnam)
  + Not all scrap collector has smart phone to take advantage of this application

## Functional Requirements

Function requirements of the system are listed as below:

**6.1 Admin:**

* Admin can view all accounts.
* Admin can create, activate, de-activate or update category of scrap.
* Admin can view all scraps posted by homeowner.

**6.2 Collector:**

* Collector can find near-by scrap with defined radius.
* Collector can view detail of each scrap.
* Collector can contact with the owner customer of the scrap.
* Collector can negotiate the price of scrap with customer.
* Collector can view the list of dealt scrap.
* Collector can find the best route to go to collect scrap.
* Collector can update or de-active account.

**6.3 Homeowner**:

* Homeowner can post scrap for sale.
* Homeowner can contact the collector.
* Homeowner can negotiate price of scrap with collector.
* Homeowner can cancel, update status of posted scrap.
* Homeowner can review collector.
* Homeowner can update or de-activate account.

## Role & Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Lại Đức Hùng | Project Manager | Supervisor | hungld5@fe.edu.vn |
| 2 | Nguyễn Duy Đạt | Developer | Leader | datndse63093@fpt.edu.vn |
| 3 | Tiết Lê Bảo Khánh | Developer | Member | khanhtlbse62247@fpt.edu.vn |
| 4 | Nguyễn Hoàng Anh Trung | Developer | Member | trungnhase62282@fpt.edu.vn |
| 5 | Nguyễn Hoàng Nhân | Developer | Member | nhannhse63259@fpt.edu.vn |

Table - Role & Responsibility

# Software Project Management Plan

## Problem Definition

* 1. **Name of this Capstone Project**
* Official name: Scrap collector
* Vietnamese name: Ứng dụng hỗ trợ thu mua ve chai
* Abbreviation: SC
  1. **Problem Abstract**
* As current Vietnamese scrap collectors have to waste a lot of time to go around and find who has scrap to sell. This traditional way is ineffective, for example, the collectors don’t know exactly where to go, so they just go around and ask loudly for scrap, which takes their time, their force and harms to their health.
* This project is concern about implementing information technologies and mapping direction algorithms into scrap collectors’ daily work. We call it Scrap Collector Helper System (SC). SC provides the homeowner an easy way to post the scrap they want to sell with detail information and the recommended price. Besides that, SC also provides the collector a feature to find out near-by scraps, book scrap and contact with the homeowner. Moreover, SC helps the collector to save time and force by providing the optimized route to collect all booked scraps. Finally, SC gives homeowner chances to review the collector and those collector with good review will have priority in their work.
  1. **Project Overview**
     1. **Current Situation**

Below are the problems encountered in this project:

* Limit of time: our team has only 4 members and we have to get requirements, complete document, develop product and do the test in just 14 weeks.
* New technique and framework: when applying new framework into capstone project, all members of the team need an amount of time to get used to new techniques.
* Lack of User Interface and User Experience design skill: all team members are studying JS for major, no one has studied about UI and UX designing before.
* Misunderstanding in business logic: the business login of scrap collecting is quite strange with us, and all requirements are given by the mentor, who is not a scrap collector. Although we have observed many scrap collectors, there are still some misunderstood logic.
* Absence of team member: team members can have a conflict in meeting schedule because of sick, class timetable or work schedule, etc…
  + 1. **The Proposed System**

According to the technology researches, we found out that the *Chinese postman problem* (also known as route inspection problem) is very capable of resolving the problem of collecting scrap. The *Chinese postman problem* is a mathematical problem of graph theory which helps the mailman to find out the shortest route through his neighborhood that return him back to the point he starts and allow him to go through every street at least once. Because of its popularity and the high applicability, we decided to choose *Chinese postman problem* to use in application.

We also build a high available web server to maintain the main system to work 24/7 to make sure that if mobile applications have the need of accessing to the server to get or update the information there will always be available.

We assign responsibility in vertical to make sure if any member in this problem cannot continue to work in our team there will be the least harmful to the project processes.

Our system includes 2 main subsystems:

* An online website for the admins to help them manage user account, category and scrap post.
* A mobile application for homeowner and collector to sell and buy scrap.
  + - 1. **Web application**

Web application is common tool to help admin manage user, category and scrap post. This application is used by administrator team and do followings:

* + Web Manage user account
  + Manage scrap category
  + Manage posted scrap
    - 1. **Mobile application**

Mobile application is a tool to help homeowner and collector to sell and buy scrap. Mobile application provide following features:

* For homeowner:
  + Post Scrap for sale
  + Contact the collector
  + Manage scrap
  + Negotiate scrap price
  + Update information
* For collector:
  + Find near-by scrap
  + Book scrap
  + Contact the homeowner
  + Schedule scrap
  + Negotiate price
    1. **Boundaries of the System**

The system can:

* Allow Admin to manage user accounts .
* Allow Admin to manage list category of scrap.
* Allow Admin to manage scrap posted by homeowner.
* Allow Homeowner to post scrap for sale.
* Allow Homeowner to negotiate scrap price.
* Allow Homeowner to contact the collector.
* Allow Homeowner to review the collector.
* Allow Collector to find near-by scrap.
* Allow Collector to negotiate scrap price.
* Allow Collector to schedule a pickup.
* Allow Collector to contact the homeowner.

The system cannot:

* Help collector make payment online.
  + 1. **Future Plans**

If this application becomes widely used, the team intends to implement these following features:

* Improve security of system by strict validation and authorization filters
* Allow collector to make video call with homeowner to check scrap
* Connect with scrap factory to help collect large amount of scrap
* Widen the scale of application, distribute it to another countries like countries in ASIAN
  + 1. **Development environment**
       1. **Hardware requirements**

**For server**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Minimum Requirement** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (8 Mbps) | Cable, Wi-Fi (50 Mbps or more) |
| **Operation System** | XP, Vista, 7, 10, Window Server 2008 | 10, Window server 2008 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| **Computer memory** | 4GB RAM | 32 GB RAM or more |
| **Storage space** | 1GB | 4GB or more |

Table - Hardware Requirements for Server

**For PC**

|  |  |  |
| --- | --- | --- |
| **PC** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Window 7 | Window 7 or more. |
| **Computer Processor** | Intel® Core i3 1.4GHz | Intel® Core i5 2.50GHz |
| **Computer Memory** | 1GB RAM | 2GB RAM or more |
| **Web Browser** | Chromes (v42 or higher) | Chrome latest stable version |

Table - Hardware requirement for PC

* + - 1. **Software requirement**

|  |  |  |
| --- | --- | --- |
| **Software** | **Name / Version** | **Description** |
| **Operation System** | Windows Server 2014 | Operating system and platform for development |
| **Environment** | Java EE 8.0, Node v10, npm v6, react-native v0.61 | Specification for developing web application and mobile application |
| **Modeling tool** | StarUML | Used to design diagram |
| **IDE** | Visual Studio Code 1.27.2 | Programming tools |
| **DBMS** | PostgreSQL | Used to create & manage the database for system |
| **Source control** | GitHub, GitHub Desktop, GitLab | Used for source control |
| **Web browser** | Chrome 69 or above | Testing browser |
| **Android Emulator** | Genymotion | Testing mobile device |

Table - Software Requirement

## Project Organization

#### 2.1 Software Process Model

Our team has chosen to use Scrum model for developing this project. Scrum is one of the implementations of agile methodology. Scrum allows us to focus on delivering deliverables to the customer at the end of every sprint, each sprint lasts about two or three weeks. Below are reasons why our team chooses Scrum model:

* Our team has only 4 members, and tasks are considered to be assigned vertically, which means that each member has to take part in all steps from getting requirements, designing, implementing, testing and managing. So, Scrum is the most suitable model for a medium project like Scrap Collector.
* User requirements are very difficult to be defined at first time, and product owner may change requirement or extend scope. With Scrum model, team member can adapt to changes better.
* In the project there are many new technologies that need to be learned. With the Scrum model, the team can learn and develop in parallel to meet deadline.

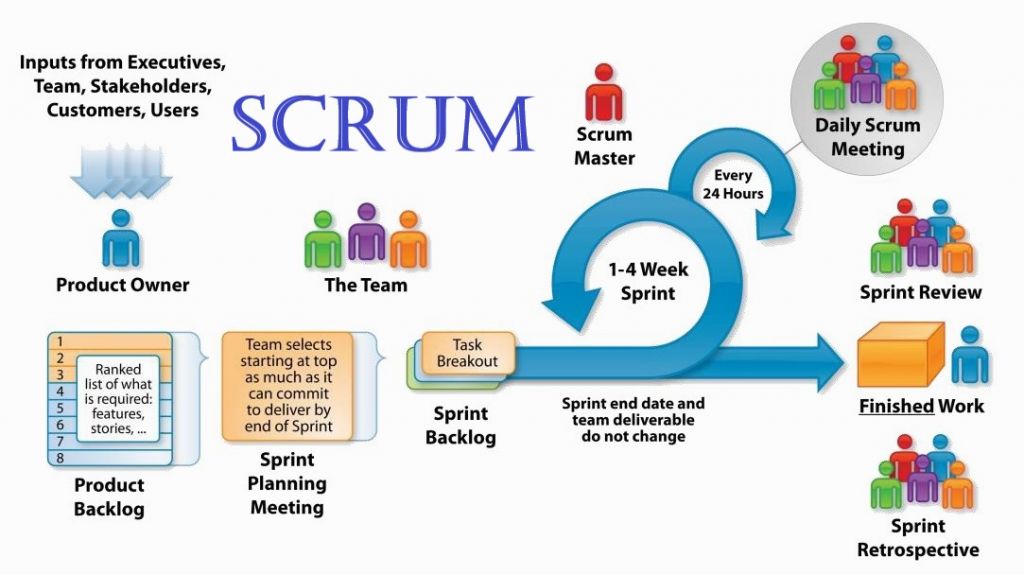


Figure – Software process model

***Reference:***[*https://www.apeironsoftware.com/scrum-roles-artifacts-and-ceremonies/*](https://www.apeironsoftware.com/scrum-roles-artifacts-and-ceremonies/)

#### 2.2 Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Lại Đức Hùng | Product Owner | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Nguyễn Duy Đạt | Scrum Master | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing * Arrange Meeting * Risk Management |
| **3** | Tiết Lê Bảo Khánh | Scrum team member | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **4** | Nguyễn Hoàng Anh Trung | Scrum team member | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **5** | Nguyễn Hoàng Nhân | Scrum team member | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |

Table - Roles & Responsibilities Details

#### 2.3 Tools and Techniques

|  |  |
| --- | --- |
| Tool/Technique | Name |
| Front-end | HTML, CSS, JavaScript, jQuery, React, Angular |
| Back-end | NestJS, Typescript, TypeORM |
| IDE | Visual Studio Code |
| DBMS | Firebase, PostgreSQL |
| Source Control | GitHub, GitHub Desktop, GitLab |
| Modelling tool | StarUML |

Table - Tools and techniques

## Project Management Plan

#### 3.1 Product Backlog

All product backlogs could be found here.

#### 3.2 Sprint Backlog

All product backlogs could be found here.

#### 3.3 Deliverables

#### 3.4 All Meeting Minutes

All sprint meeting minutes could be found here.

## Coding Convention

**Summary**:

* **Naming Convention**:
* Variable names should be short yet meaningful. The choice of a variable name should be designed to indicate to the casual observer the intent of its use.
* Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized (Camel case).
* **Indentation**:
* One declaration per line is recommended since it encourages commenting.
* In absolutely no case should variables and functions be declared on the same line.
* Do not put different types on the same line.
* **Declarations Convention:**
  + One declaration per line is recommended since it encourages commenting.
  + Using Java Code Convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

* + Using Javascript Code Convention from:

<https://google.github.io/styleguide/jsguide.html>

# Software Requirement Specification

## User Requirement Specification

#### 1.1 Guest Requirement

*Guest is a person who doesn’t have access to the system. To use all functions, guest must login. These are some functions guest can use:*

* Login.
* Register an account

#### 1.2 Authorized User Requirement

*Authorized User is an authorized user who has permission to login into the system. Authorize User can use below functions:*

* Log out
* View profile
* Update profile

#### 1.3 Homeowner Requirement

*Homeowner is an authorized user who has permission to login into the system. With Homeowner role, user can use all functions of Authorized User and below functions:*

* Post scrap for sale
* Chat with collector
* Call collector
* Review collector
* Update post information
* View post history
* Delete post
* Accomplish post
* View scrap price negotiation from Collector
* Accept scrap price negotiation
* Reject scrap price negotiation
* View scrap price negotiation from Collector
* Accept scrap price negotiation
* Reject scrap price negotiation
* Review Collector

#### 1.4 Collector Requirement

*Collector is an authorized user who has permission to login into the system. With Collector role, user can use all functions of Authorized User and below functions:*

* Search near-by scraps
* View scrap detail
* Negotiate scrap price
* Book scrap
* Call Homeowner
* Chat with Homeowner
* View pick-up schedule
* View direction
* View scrap price negotiation response
* View scrap booking response
* View booking list
* Cancel booking
* Accomplish booking
* View review

#### 1.5 Admin Requirement

*Admin is an authorized user has permission to login into the system under admin role, Admin can use below management functions:*

* Manage user account
* Manage scrap post
* Manage category

## System Requirement Specification

#### 2.1 External Interface Requirement

##### 2.1.1 User Interface

* GUI should be simple, clear, intuitive, and reminiscent.
* The interface design is an iterate process includes: designing, sketching, prototyping and user assessment.
* Website and mobile application must be responsive, suitable with all devices resolution such as: laptop, tablet or smartphone.
* Every feature should be arranged and displayed briefly.
* The User Interface uses language is Vietnamese.

##### 2.1.2 Hardware Interface

* Laptop or Personal Computer
* Smartphone or Tablet

##### 2.1.3 Software Interface

* The website works with Firefox (v57 or above), Chrome (v51 or above), Safari (v5 or above) or any web browser which supports HTML5 & CSS3.
* The mobile app work best on android version 6.0.1+.

##### 2.1.4 Communication Protocol

* Use HTTP protocol 1.1 for communication between:
  + Web application and web service
  + Mobile application and service

#### 2.2 System Overview Use Case

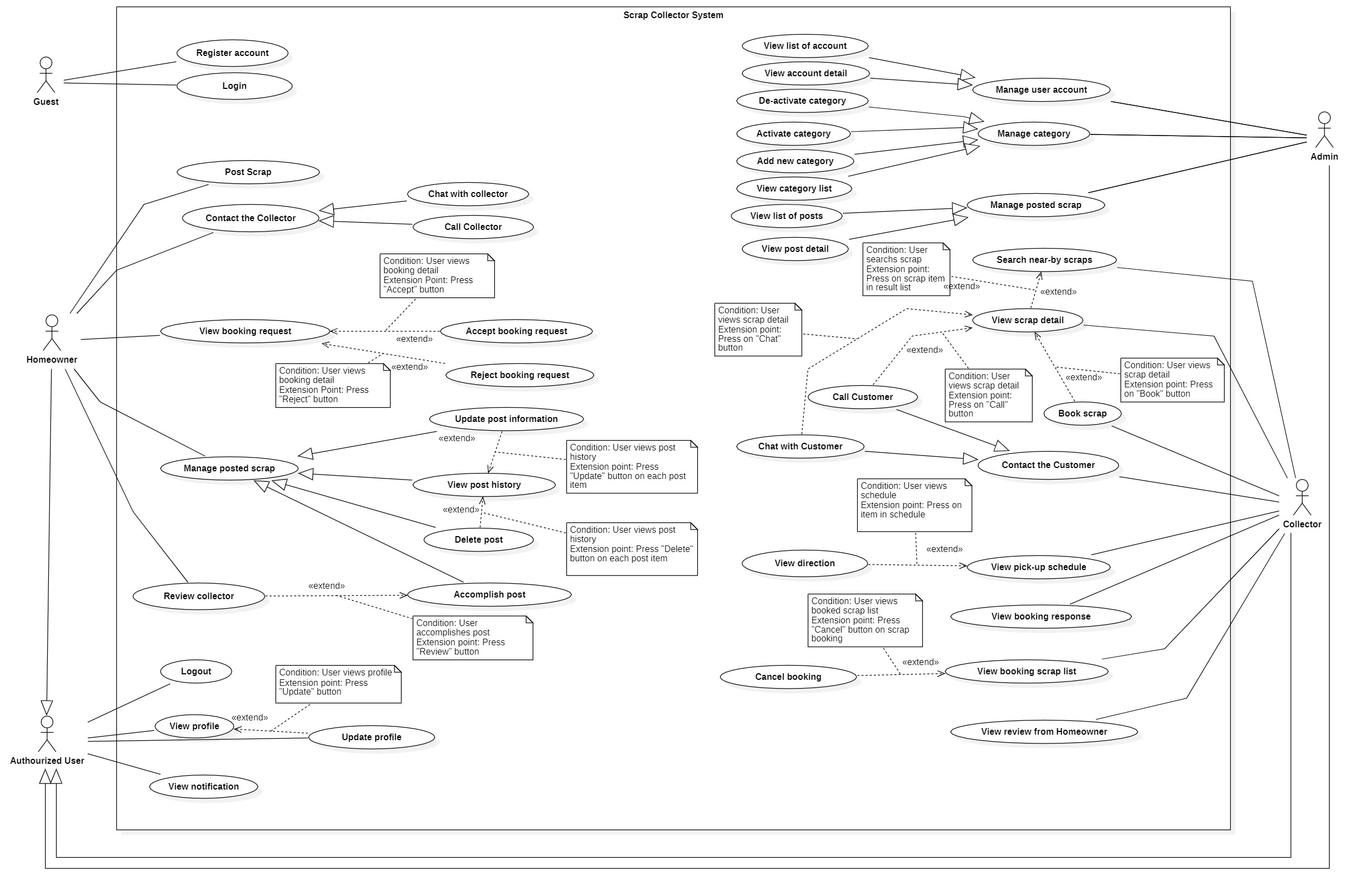


Figure – Use Case Diagram

#### 2.3 List of use case

##### 2.3.1 <Guest> Overview Use Case

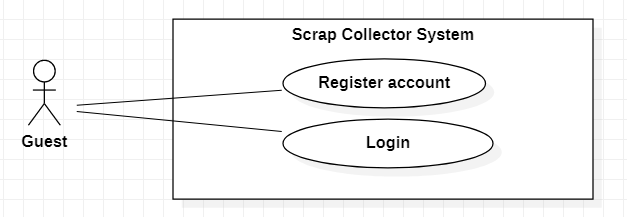


Figure 3 - Guest Overview Use Case

###### 2.3.1.1 Login

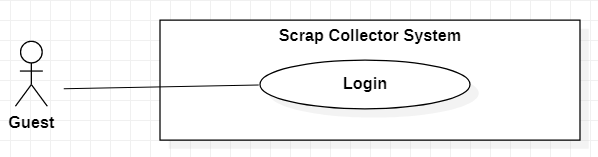


Figure 4 - <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_1** | | | |
| **Use Case No.** | SC \_UC\_1 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * Allow guest to log in the system.   **Goal:**   * Guest can log in the system with specific role.   **Triggers:**   * Guest sends login command.   **Preconditions:** N/A  **Post conditions:**   * Success: Guest is authorized with specific role in the system. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to login view. | System requires identity information from Guest:  - “Tên đăng nhập” (required): text input.  - “Mật khẩu” (required): text input.  - “Đăng nhập” button.  - “Đăng kí tài khoản” button | | 2 | Guest inputs required information. |  | | 3 | Guest presses “Đăng nhập” button. | System validate information. [Exception 1] [Exception 2][Exception 3] ][Exception 4] | | 4 |  | System let Guest to the “Trang chủ” screen based on specific role. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message the "Không có kết nối Internet" when the internet is lost | | 2 | Guest let “Tên đăng nhập” input empty | Focus on “Tên đăng nhập” filed, show error message “Không để trống thông tin này” | | 3 | Guest let “Mật khẩu” input empty | Focus on “Mật khẩu” filed, show error message “Không để trống thông tin này” | | 4 | Guest enter wrong identity information. | System shows error message “Tên đăng nhâp hoặc mật khẩu không chính xác. Vui lòng thử lại” |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted before being sent to server. * After login to system, guest will be redirected to specific view based on their role on the system: homeowner, collector or admin. * If role is “Collector”, the system will display to Collector view. * If role is “Homeowner”, the system will display to Homeowner view. * If role is “Admin”, the system will display to Admin Dashboard view. | | | |

Table 7 **-** <Guest> Login use case specification

###### 2.3.1.2 Register account

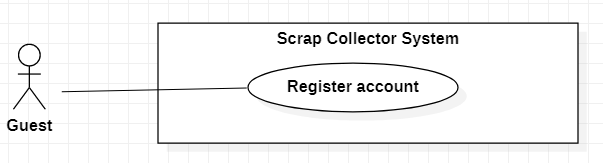


Figure 5 - <Guest> Register

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_2** | | | |
| **Use Case No.** | SC \_UC\_2 | **Use Case Version** | 1.0 |
| **Use Case Name** | Register account | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * Allow guest to create an account the system.   **Goal:**   * Guest can have an account in the system with specific role: Homeowner or Collector.   **Triggers:**   * Guest presses “Đăng kí tài khoản” button on Login screen.   **Preconditions:** N/A  **Post conditions:**   * Success: A new account is created with specific role in the system. * Fail: System shows error messages. Account is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to register view. | System shows a screen with 2 buttons:  - “Tôi thu mua ve chai” button  - “Tôi bán ve chai” button | | 2 | Guest presses on a button | System requires identity information from Guest:  - “Tên đăng nhập” (required): text input  - “Mật khẩu” (required): text input  - “Xác nhận mật khẩu” (required): text input  - “Email”(optional): text input  - “Tên” (required): text input.  - “Ảnh đại diện”(optional): Image chooser  - “Số điện thoại” (required): text input  - “Tỉnh/Thành phố” (required): selection  - “Quận/Huyện” (required): selection  - “Phường/Xã” (required): selection  - “Địa chỉ” (required): text input  System shows “Đăng kí” button | | 3 | Guest inputs full required information |  | | 4 | Guest presses “Đăng kí” button. | System validates inputs[Exception 1] [Exception 2] [Exception 3] [Exception 4] [Exception 5] [Exception 6] [Exception 7] [Exception 8] [Exception 9] [Exception 10] [Exception 11] | | 5 |  | System creates account with specific role and lets Guest to Login screen |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 | Guest let “Tên đăng nhập” input empty | Focus in “Tên đăng nhập” field. Show error message “Không để trống thông tin này” | | 3 |  | System focuses on “Tên đăng nhập” field and show error message “Đã có người sử dụng tên đăng nhập này, vui lòng chọn tên đăng nhập khác” when the username is duplicated. | | 4 | Guest let “Mật khẩu” input empty | Focus in “Mật khẩu” field. Show error message “Không để trống thông tin này” | | 5 | Guest let “Số điện thoại” input empty | Focus in “Số điện thoại” field. Show error message “Không để trống thông tin này” | | 6 | Guest let “Địa chỉ” input empty | Focus in “Địa chỉ” field. Show error message “Không để trống thông tin này” | | 7 | Guest let “Tên” input empty | Focus in “Tên” field. Show error message “Không để trống thông tin này” | | 8 | Guest let “Mật khẩu” less than 6 letters | Focus in “Mật khẩu” field. Show error message “Mật khẩu tối thiểu 6 kí tự”. | | 9 | “Xác nhận mật khẩu” is not same as “Mật khẩu”, empty or less than 6 letters. | Focus in “Xác nhận mật khẩu” field. Show error message “Xác nhận mật khẩu không khớp”. | | 10 | Guest input wrong Email format | Focus in Email field. Show error message “Địa chỉ Email không hợp lệ” | | 11 | Guest input wrong Phone number format | Focus in “Số điện thoại” field. Show error message “Số điện thoại không hợp lệ” |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted before being sent to server. * Email must contain “@” and must be valid. * Phone number must be in right format. * After register, the user will be logged in to the system and be redirected to Login screen. * If user presses on “Tôi bán ve chai” button, new account will be created with role “Homeowner”. * If user presses on “Tôi thu mua ve chai” button, new account will be created with role “Collector”. | | | |

Table 8 - <Guest> Register use case specification

##### 2.3.2 <Authorized User> Overview Use Case

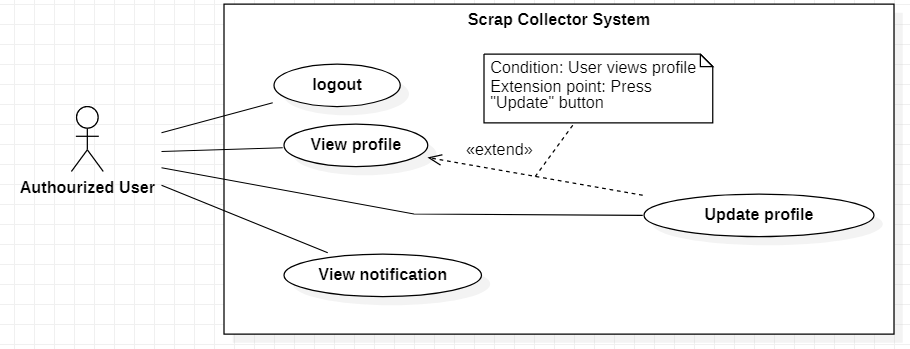


Figure - Authorized User Overview Use Case

###### 2.3.2.1 Logout

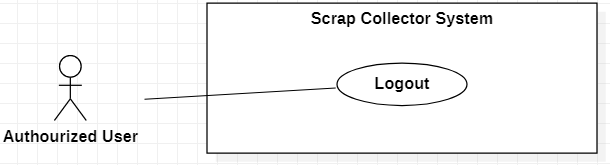


Figure - <Authorized User> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_3** | | | |
| **Use Case No.** | SC \_UC\_3 | **Use Case Version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows the Authorized User to log out of the system.   **Goal:**   * The Authorized User logs out of the system successfully.   **Triggers:**   * User presses “Đăng xuất” button.   **Preconditions:**   * User has already logged in the system with the role that has been authorized.   **Post conditions:**   * Success: The Authorized User logs out of the system successfully, system moves to Login Screen. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | The Authorized User presses on “Đăng xuất” button. | System clear the session of that Authorized User then shows Login screen. [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows error message the "Không có kết nối Internet" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Only Authorized User can log out. * After the Authorized User logs out, clear all sessions of that user. | | | |

Table - <Authorized User> Logout use case specification

###### 2.3.2.2 View profile

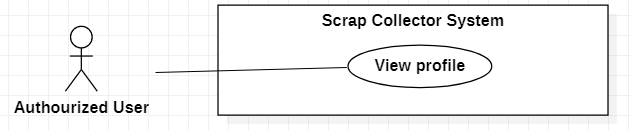


Figure - <Authorized User> View profile

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_4** | | | |
| **Use Case No.** | SC \_UC\_4 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Profile | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * Allow Authorized User to view his/her profile.   **Goal:**   * Authorized User views profile.   **Triggers:**   * User chooses “Tài khoản” tab in tab navigation.   **Preconditions:**   * User has already logged in the system with the role that has been authorized.   **Post conditions:**   * Success: System shows Profile screen. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User chooses “Tài khoản” tab in tab navigation. | System shows Profile Screen with:   * “Ảnh đại diện”: Image * “Tên”: Text * “Tên đăng nhập”: Text * “Số điện thoại”: Text * “Địa chỉ”: Text * “Tỉnh/ Thành phố”: Text * “Quận/ Huyện”: Text * “Phường/ Xã”: Text * “Chỉnh sửa” button * “Xem lịch sử hoàn thành” button * “Đăng xuất” button   [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows error message the "Không có kết nối Internet" when the internet is lost. |   **Relationships:** [SC \_UC\_5] [SC \_UC\_6]  **Business Rules:**   * Only Authorized User can view profile * Password mustn’t be shown | | | |

Table - <Authorized User> View profile use case specification

###### 2.3.2.3 Update profile

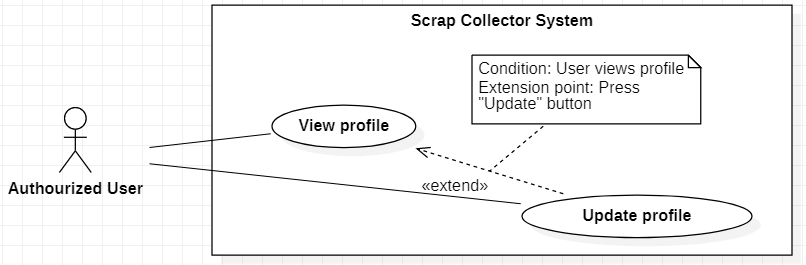


Figure - <Authorized User> Update profile

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_5** | | | |
| **Use Case No.** | SC \_UC\_5 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update Profile | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * Allow Authorized User to update his/her profile.   **Goal:**   * Authorized User updates profile.   **Triggers:**   * User presses “Chỉnh sửa” button in “Tài khoản” screen.   **Preconditions:**   * User has already logged in the system with the role that has been authorized.   **Post conditions:**   * Success: System shows Update Profile screen. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User presses “Chỉnh sửa” button in “Tài khoản” screen. | System shows Update Profile Screen with:   * “Chỉnh sửa” title * “Ảnh đại diện”: image, editable * “Tên”: required, text input * “Số điện thoại”: required, text input * “Địa chỉ”: required, text input * “Tỉnh/ Thành phố”: selection * “Quận/ Huyện”: selection * “Phường/ Xã”: selection * “Mật khẩu”: hidden text, editable * Confirm password: text, editable * “Lưu chỉnh sửa” button * “Quay lại” button | | 2 | Authorized User edit editable fields |  | | 3 | Authorized User presses “Lưu chỉnh sửa” button. [Alternative 1] | System saves new information and shows Profile screen with new information. [Exception 1][Exception 2] [Exception 3][Exception 4] [Exception 5] [Exception 6] [Exception 7] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Authorized User presses “Quay lại” button. | System shows “Tài khoản” screen. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows error message the "Không có kết nối Internet" when the internet is lost. | | 2 | Guest let “Tên” input empty | Focus in “Tên” field. Show error message “Không để trống thông tin này” | | 3 | Guest let “Số điện thoại” input empty | Focus in “Số điện thoại” field. Show error message “Không để trống thông tin này” | | 4 | Guest let “Địa chỉ” input empty | Focus in “Địa chỉ” field. Show error message “Không để trống thông tin này” | | 5 | Guest let Password less than 6 letters | Focus in Password field. Show error message “Mật khẩu tối thiểu 6 kí tự”. | | 6 | Confirmed password is not same as Password, empty or less than 6 letters. | Focus in Confirm Password field. Show error message “Xác nhận mật khẩu không khớp”. | | 7 | Guest input wrong Phone number format | Focus in Email field. Show error message “Số điện thoại không hợp lệ” |   **Relationships:** [SC \_UC\_4]  **Business Rules:**   * Only Authorized User can update profile * Password mustn’t be shown * Phone number must be in the phone number format | | | |

Table - <Authorized User> Update profile use case specification

###### 2.3.2.4 View notification

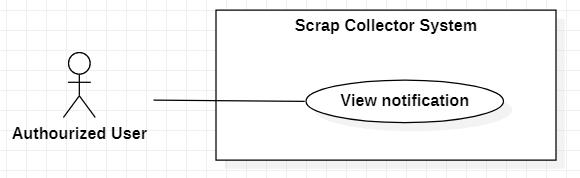


Figure - <Authorized user> View notification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_6** | | | |
| **Use Case No.** | SC \_UC\_6 | **Use Case Version** | 1.0 |
| **Use Case Name** | View notification | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 29/2/2020 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * Allow Authorized User to view notification.   **Goal:**   * Authorized User views notification.   **Triggers:**   * User receives notification * User presses on “Thông báo” tab in bottom tab bar   **Preconditions:**   * User has already logged in the system with the role that has been authorized.   **Post conditions:**   * Success: System shows notification. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User presses “Thông báo” tab in bottom tab bar. | System shows “Thông báo” Screen with list of notifications. Each notification contains:   * Time * Title of notification * Content of notification   [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows error message the "Không có kết nối Internet" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Notifications are organized and shown by time of notification, the latest notification is placed on the top. | | | |

Table - <Authorized user> View notification use case specification

##### 2.3.3 <Homeowner> Overview Use Case

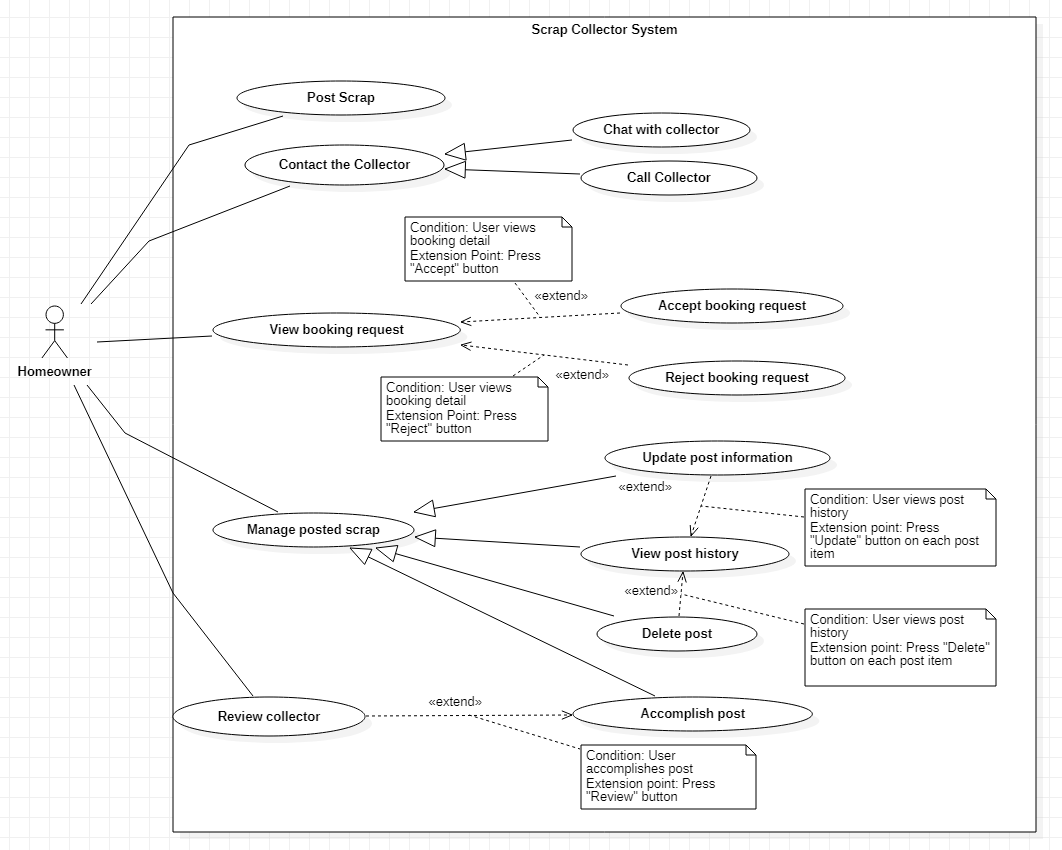


Figure 11 - Homeowner Overview Use Case

###### 2.3.3.1 Post scrap

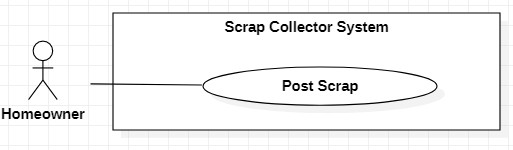


Figure - <Homeowner> Post scrap

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_7** | | | |
| **Use Case No.** | SC \_UC\_7 | **Use Case Version** | 1.0 |
| **Use Case Name** | Post scrap | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | High |
| **Actor:**   * Homeowner   **Summary:**   * Allow Homeowner to post scrap for sale.   **Goal:**   * Homeowner post one or many scraps for sale in a post.   **Triggers:**   * Homeowner presses “Đăng bán” button in bottom tab bar.   **Preconditions:**   * User must sign in as a Homeowner   **Post conditions:**   * Success: A new post is created. * Fail: System shows error messages. Post is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner goes to post scrap view. | System requires identity information from Homeowner:  - “Tiêu đề” (required): text input  - “Ảnh” (required): image chooser, at least 1 image and at most 5 images  - “Giá bán” (required): number input  - “Mô tả” (required): text input  - “Thời gian có thể lấy” (required): date time picker, can have many available times  - “Phân loại” (required): selection  - “Đơn vị”: text  - “Số lượng”: number input  - “Địa chỉ”: text input  - “Số điện thoại”: number  - “Đăng bán” button | | 2 | Homeowner inputs full required information |  | | 3 | Homeowner presses “Đăng bán” button. | System validates all required fields [Exception 1] [Exception 2] [Exception 3] [Exception 4] [Exception 5] [Exception 7] [Exception 8] | | 4 |  | System creates new post and let Homeowner to “Đang rao” tab in “Trang chủ” screen, |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 | Homeowner lets “Tiêu đề” input empty | Focus in “Tiêu đề” field. Show error message “Không để trống thông tin này”. | | 3 | Homeowner lets “Mô tả” input empty | Focus in “Mô tả” field. Show error message “Không để trống thông tin này”. | | 4 | Homeowner lets “Giá bán” input empty | Focus in “Giá bán” field. Show error message “Không để trống thông tin này”. | | 5 | Homeowner doesn’t choose any image | Focus in “Thời gian có thể lấy” field. Show error message “Vui lòng chọn ít nhất 1 ảnh”. | | 6 | Homeowner doesn’t choose any available time | Focus in “Ảnh” field. Show error message “Vui lòng chọn ít nhất 1 ảnh”. | | 7 | Homeowner does not select “Phân loại” | Focus in “Phân loại” field. Show error message “Chọn loại ve chai”. |   **Relationships:** N/A  **Business Rules:**   * Created post will have “Selling” status. * System records time of the post automatically. * The default value of address of a post is get from Homeowner’s profile. * The default value of phone number of a post is get from Homeowner’s profile. | | | |

Table - <Homeowner> Post scrap use case specification

###### 2.3.3.2 Chat with collector

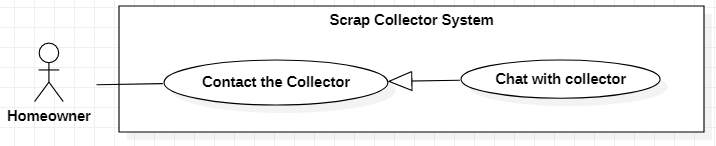


Figure - <Homeowner> Chat with collector

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_8** | | | |
| **Use Case No.** | SC\_UC\_8 | **Use Case Version** | 1.0 |
| **Use Case Name** | Chat with collector | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Homeowner   **Summary:**   * Allow homeowner to chat with collector   **Goal:**   * Homeowner chats with one or many Collectors   **Triggers:**   * User presses “Chat” tab in bottom tab bar   **Preconditions:**   * User must be a homeowner   **Post conditions:**   * Success: A conversation between Homeowner and Collector is created. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner chooses “Chat” tab in bottom tab bar | System shows “Chat” screen with all conversations of that Homeowner | | 2 | Homeowner chooses a conversation | System shows messages in chosen conversation, a text input and a “Gửi” button | | 3 | Homeowner inputs message and presses “Gửi” button | System sends message to the Collector in that conversation and shows that message [Exception 1] |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** Generalized from Contact with Collector use case  **Business Rules:**   * A Homeowner can chat with many Collectors * A conversation is related to a post | | | |

Table - <Homeowner> Chat with collector use case specification

###### 2.3.3.3 Call collector

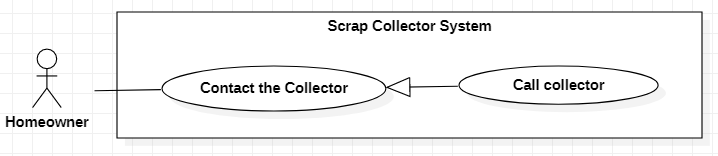


Figure - <Homeowner> Call collector

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_9** | | | |
| **Use Case No.** | SC\_UC\_9 | **Use Case Version** | 1.0 |
| **Use Case Name** | Call collector | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Homeowner   **Summary:**   * Allow Homeowner to call Collector   **Goal:**   * Homeowner calls Collector for direct communication.   **Triggers:**   * User presses “Gọi” button or button with phone icon.   **Preconditions:**   * User must be a homeowner   **Post conditions:**   * Success: Homeowner calls Collector. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Gọi” button or button with phone icon. | System shows:   * Alert message: “Bạn muốn gọi cho người thu mua” * “Đồng ý” button * “Hủy” button | | 2 | Homeowner presses “Đồng ý” button | System uses call service of phone to make a call to Collector. [Exception 1] [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Hủy” button | System goes back to previous screen |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 |  | System shows message the "Không có quyền truy cập" when the application does not have permission to make a phone call. |   **Relationships:** Generalized from Contact with Collector use case  **Business Rules:**   * Mobile Application must ask user for making phone call permission | | | |

Table - <Homeowner> Call collector use case specification

###### 2.3.3.4 View booking request

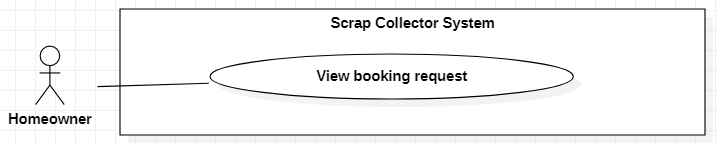


Figure - <Homeowner> View booking request

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_10** | | | |
| **Use Case No.** | SC\_UC\_10 | **Use Case Version** | 1.0 |
| **Use Case Name** | View booking request | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Medium |
| **Actor:**   * Homeowner   **Summary:**   * Allow homeowner to view scrap booking request   **Goal:**   * Homeowner view scrap booking request from Collector then accept or reject it.   **Triggers:**   * User receives booking notification.   **Preconditions:**   * User must be a Homeowner   **Post conditions:**   * Success: User view booking detail * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User presses on the booking notification | System shows booking with information:   * “Tên post”: text, title of scrap * “Với giá”: text, the price that Collector negotiate * “Thời gian lấy” * “Người đặt”: text * “Chấp nhận” button * “Từ chối” button   [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_US\_11] [SC\_US\_12]  **Business Rules:**   * Only bookings with status “waiting” is shown | | | |

Table - <Homeowner> View booking request use case specification

###### 2.3.3.5 Accept booking request



Figure - <Homeowner> Accept booking request

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_11** | | | |
| **Use Case No.** | SC\_UC\_11 | **Use Case Version** | 1.0 |
| **Use Case Name** | Accept booking request | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Medium |
| **Actor:**   * Homeowner   **Summary:**   * Allow Homeowner to accept scrap booking request.   **Goal:**   * Homeowner allows to sell scrap for the Collector.   **Triggers:**   * User presses “Chấp nhận” button in scrap booking request notification.   **Preconditions:**   * User must be a homeowner * Homeowner view scrap booking   **Post conditions:**   * Success: The scrap is moved into booking list of the Collector. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Chấp nhận” button in scrap booking notification. | System shows a pop-up with:   * “Bạn muốn đồng ý yêu cầu này?” message * “Đồng ý” button * “Đóng” button | | 2 | Homeowner presses “Đồng ý” button in pop up. [Alternative 1] | System changes status of booking to “Accepted”, shows message to Homeowner and sends a notification to the Collector [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Đóng” button in pop up. | System hides pop-up and goes back to the booking request screen. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_US\_10]  **Business Rules:**   * After being accepted, the booking has “Accepted” status * If a scrap has more than one booking request from Collector, when a booking request is accepted, other bookings are automatically rejected | | | |

Table - <Homeowner> Accept booking request use case specification

###### 2.3.3.6 Reject booking request

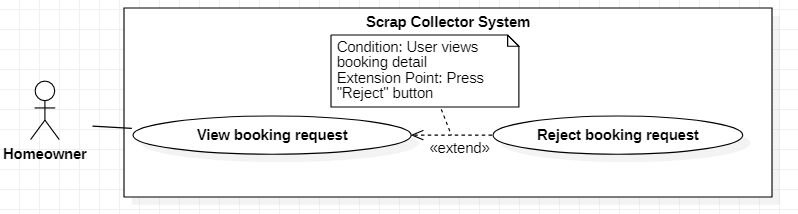


Figure - <Homeowner> Reject booking request

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_12** | | | |
| **Use Case No.** | SC\_UC\_12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Reject booking request | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Medium |
| **Actor:**   * Homeowner   **Summary:**   * Allow Homeowner to reject scrap booking request.   **Goal:**   * Homeowner does not allow to sell scrap for the Collector.   **Triggers:**   * User presses “Từ chối” button in scrap booking request notification.   **Preconditions:**   * User must be a homeowner * Homeowner view scrap booking   **Post conditions:**   * Success: The scrap booking request is canceled. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Từ chối” button in scrap booking notification. | System shows a pop-up with:   * “Bạn muốn từ chối yêu cầu này?” message * “Từ chối” button * “Đóng” button | | 2 | Homeowner presses “Từ chối” button in pop up. | System changes status of booking to “Rejected”, show message to Homeowner and sends notification to the Collector [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_US\_10]  **Business Rules:** N/A | | | |

Table - <Homeowner> Reject booking request use case specification

###### 2.3.3.7 Update post information

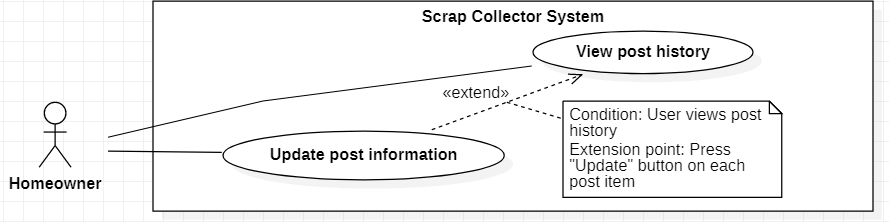


Figure - <Homeowner> Update post information

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_13** | | | |
| **Use Case No.** | SC\_UC\_13 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update post information | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Homeowner   **Summary:**   * Allow homeowner to update post information.   **Goal:**   * Homeowner change information of the post.   **Triggers:**   * User presses “Chỉnh sửa” button in post history.   **Preconditions:**   * User must be a homeowner * User views post history.   **Post conditions:**   * Success: Post information is updated. * Fail: System shows error messages. Post is not updated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Chỉnh sửa” button on each post | System shows a screen containing post information:   * “Tiêu đề”: text input, editable * “Phân loại”: selection, editable * “Giá”: number input, editable * “Thời gian đăng”: text, not editable * “Thời gian rãnh”: selection, editable * “Hình ảnh: image chooser, editable * “Lưu” button * “Hủy” button | | 2 | Homeowner updates information in editable fields. |  | | 3 | Homeowner presses “Lưu” button [Alternative 1] | System validates inputs [Exception 1], updates post information and goes to “Lịch sử bài đăng” screen [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Hủy” button | System goes back to “Lịch sử bài đăng” screen |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User inputs “Giá” with text not number | Focus in “Giá” field, shows message error “Giá không hợp lệ” | | 2 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_US\_14]  **Business Rules:**   * Only post having “Selling” status can be updated * Fields which isn’t changed will remain old information | | | |

Table - <Homeowner> Update post information use case specification

###### 2.3.3.8 View post history

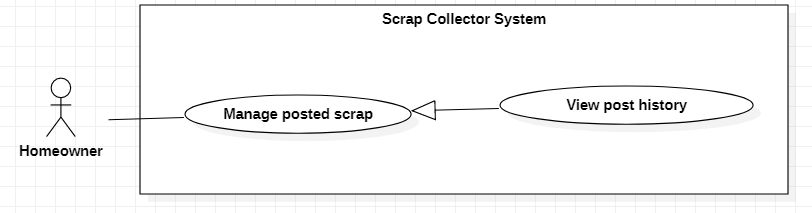


Figure - <Homeowner> View post history

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_14** | | | |
| **Use Case No.** | SC\_UC\_17 | **Use Case Version** | 1.0 |
| **Use Case Name** | View post history | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Homeowner   **Summary:**   * Allow homeowner view post history.   **Goal:**   * Homeowner view post history.   **Triggers:**   * User presses “Xem lịch sử bài đăng” button.   **Preconditions:**   * User must be a homeowner   **Post conditions:**   * Success: A screen with list of posts will be shown. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Xem lịch sử bài đăng” button | System shows “Lịch sử bài đăng” screen, with posts of that Homeowner. Posts is categorized by status “Đang rao”, “Đang được đặt”, “Đã hoàn thành”[Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_UC\_15], [SC\_UC\_16], Generalized from Manage post scrap use case  **Business Rules:** N/A | | | |

Table - <Homeowner> View post history use case specification

###### 2.3.3.9 Delete post

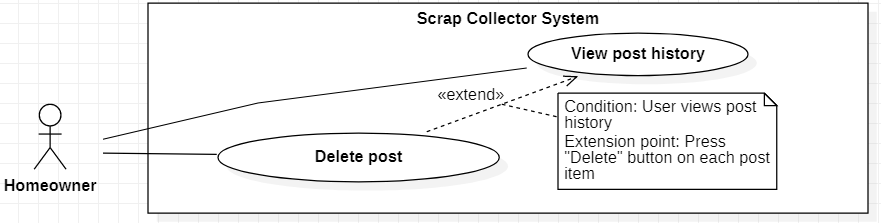


Figure - <Homeowner> Delete post

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_15** | | | |
| **Use Case No.** | SC\_UC\_15 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete post | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Homeowner   **Summary:**   * Allow homeowner to delete post.   **Goal:**   * Homeowner delete post.   **Triggers:**   * User presses “Xóa ve chai” button post history.   **Preconditions:**   * User must be a homeowner * User views post history   **Post conditions:**   * Success: A post is deleted. * Fail: System shows error messages. Post is not deleted.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Xóa ve chai” button | System shows alert with:   * “Bạn có muốn xóa bài đăng” message * “Đồng ý” button * “Không” button | | 2 | Homeowner presses “Đồng ý” button | System deletes post, deletes all the booking request of that post, shows result to the Homeowner and sends notification to the Collector. [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Hủy” button | System goes back to “Lịch sử bài đăng” screen |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_UC\_14], Generalized from Manage post scrap use case  **Business Rules:**   * Only posts with “Selling” status can be deleted * The post which is chosen to be deleted is still saved in the database, but is disabled * If that post is booked by the Collector, Homeowner can not delete that post * If that post is not booked by any Collector, systems deletes all pending booking requests of that post | | | |

Table - <Homeowner> Delete post use case specification

###### 2.3.3.10 Accomplish post

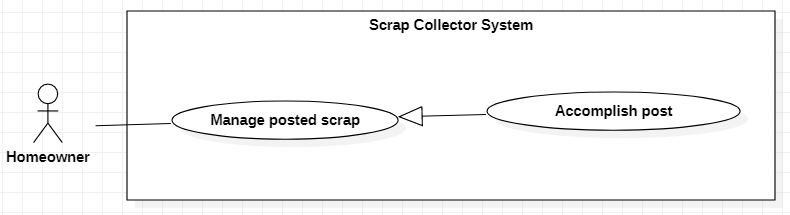


Figure - <Homeowner> Accomplish post

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_16** | | | |
| **Use Case No.** | SC\_UC\_19 | **Use Case Version** | 1.0 |
| **Use Case Name** | Accomplish post | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Homeowner   **Summary:**   * Allow homeowner to accomplish a post.   **Goal:**   * Homeowner accomplish a post.   **Triggers:**   * User presses “Hoàn thành” button.   **Preconditions:**   * User must be a homeowner   **Post conditions:**   * Success: The post is accomplished. * Fail: System shows error messages. Post is not accomplished.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Hoàn thành” button on post item. | System shows alert with:  - “Bạn có muốn hoàn thành bài đăng” message  - “Đồng ý” button  - “Không” button | | 2 | Homeowner presses “Đồng ý” button | System accomplishes post. [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Không” button | System goes back to “Lịch sử bài đăng” screen |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_UC\_17], Generalized from Manage post scrap use case  **Business Rules:**   * Only posts with “Booked” status can be accomplished | | | |

Table - <Homeowner> Accomplish post use case specification

###### 2.3.3.11 Review collector

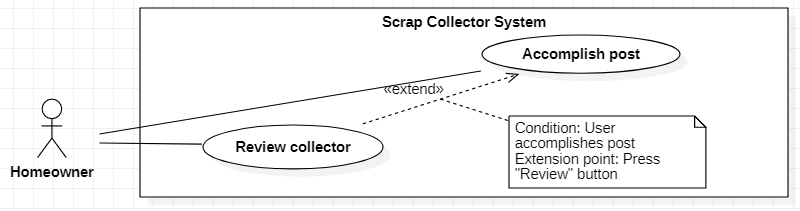


Figure - <Homeowner> Review collector

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_17** | | | |
| **Use Case No.** | SC\_UC\_20 | **Use Case Version** | 1.0 |
| **Use Case Name** | Review collector | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Homeowner   **Summary:**   * Allow homeowner to review collector.   **Goal:**   * Homeowner reviews Collector with star.   **Triggers:**   * User presses “Đánh giá” button.   **Preconditions:**   * User must be a homeowner * Homeowner accomplishes a post   **Post conditions:**   * Success: A review is created. * Fail: System shows error messages. Review is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Đánh giá” button | System shows Review pop-up with:  - “Đánh giá của bạn” title  - “Người được đánh giá”: Collector name  - “Số sao”(required): star with total 5 stars  - “Mô tả”(optional) : text input  - “Gửi đánh giá” button  - “Đóng” button | | 2 | Homeowner chooses star, inputs description and presses “Gửi đánh giá” button | Systems creates review and sends notification to Collector. [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Homeowner presses “Đóng” button | System closes Review pop-up |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_UC\_16]  **Business Rules:**   * Review maps with scrap which is accomplished. * Only after the Homeowner accomplished a scrap can Homeowner makes a review * Each Homeowner can makes one or many reviews. * Each review belongs to one Collector only * Default number of star in review pop-up is 5 | | | |

Table - <Homeowner> Review collector use case specification

##### 2.3.4 <Collector> Overview Use Case

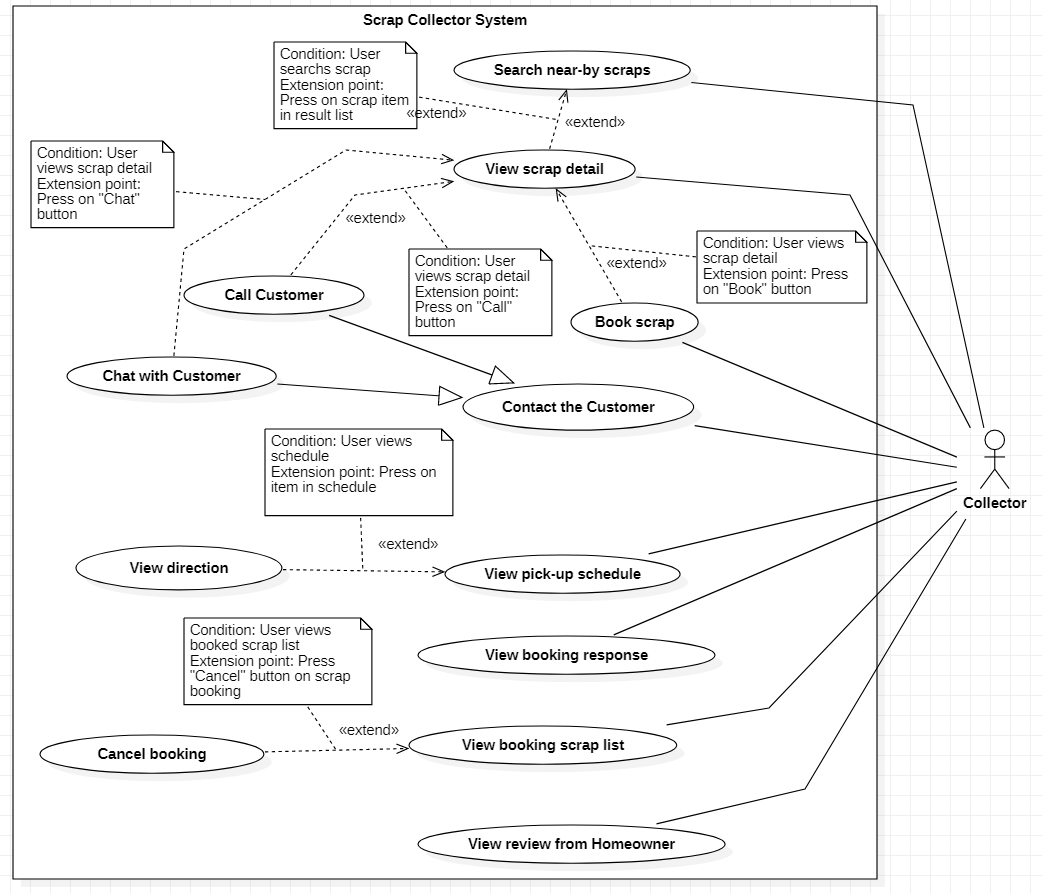


Figure - Collector Overview Use Case

###### 2.3.4.1 Search near-by scraps



Figure - <Collector> Search near-by scraps

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_18** | | | |
| **Use Case No.** | SC \_UC\_18 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search near-by scraps | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow collector to find near-by scrap by radius, scrap category.   **Goal:**   * Collector can see near-by scrap post to book.   **Triggers:**   * Collector press “Trang chủ” icon in tab navigator.   **Preconditions:**   * User must login in the system with role “Collector” * Collector have registered his/her location.   **Post conditions:**   * Success: A list of scrap post is shown on screen. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector goes to “Trang chủ” screen | System shows list of scrap posts in radius of 5km based on collector’s location, with any kind of category. [Alternative 1] [Alternative 2] [Exception 1] [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Collector selects type of scrap category in “Phân loại” selection | System searches scrap post with selected category and shows search result |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 |  | System shows message the "Không thể tải được bản đồ" when the map API is lost. |   **Relationships:** [SC\_UC\_19]  **Business Rules:**   * Only return scrap posts that have not been booked * The value of distance to search can be adjusted by Admin | | | |

Table - <Collector> Search near-by scraps use case specification

###### 2.3.4.2 View scrap detail

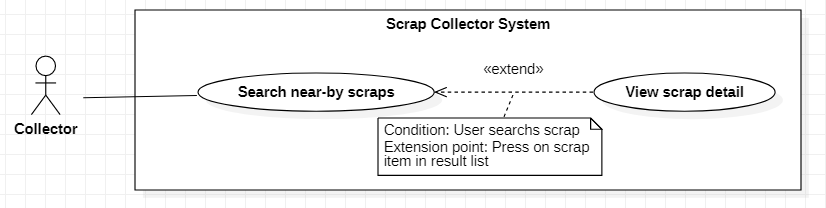


Figure - <Collector> View scrap detail

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_19** | | | |
| **Use Case No.** | SC \_UC\_19 | **Use Case Version** | 1.0 |
| **Use Case Name** | View scrap detail | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/1/2020 | **Priority** | Medium |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to view scrap detail.   **Goal:**   * Collector views detail of the scrap then decide whether to buy or not.   **Triggers:**   * Collector presses the item of scrap.   **Preconditions:**   * User must login in the system with role “Collector”. * Collector searchs near-by scrap.   **Post conditions:**   * Success: A new post is created. * Fail: System shows error messages. Post is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector press in the Item of scrap on search result list | System shows the detail of scrap with:   * “Tiêu đề”: Text * “Giá bán”: Text * “Hình ảnh”: Image slider * “Phân loại”: Text * “Đơn vị”: Text * “Số lượng”: Text * “Mô tả”: Text * “Thời gian”: Text * “Gọi” button * “Chat” button * “Đặt ve chai” button   [Exception 1] [Exception 2] |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows error message the "Không có kết nối Internet" when the internet is lost. | | 2 |  | System shows error message the "Không thể tìm thấy bài đăng" when cannot find the post. |   **Relationships:** [SC\_UC\_18] [SC\_UC\_20] [SC\_UC\_21] [SC\_UC\_22]  **Business Rules: N/A** | | | |

Table - <Collector> View scrap detail use case specification

###### 2.3.4.3 Book scrap

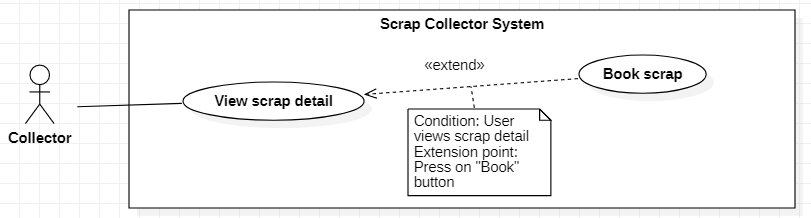


Figure - <Collector> Book scrap

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_20** | | | |
| **Use Case No.** | SC\_UC\_20 | **Use Case Version** | 1.0 |
| **Use Case Name** | Book scrap | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | High |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to book scrap   **Goal:**   * Collector send booking request to the Homeowner of the post.   **Triggers:**   * User presses “Đặt ve chai” button.   **Preconditions:**   * User must be a Collector * Collector views post detail   **Post conditions:**   * Success: A booking is created. * Fail: System shows error messages. Booking is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector presses “Đặt ve chai” button. | System shows a screen with:  - “Đặt ve chai” title  - “Giá”: editable input number, default value of this input is the price of post  - “Thời gian lấy ve chai”(required): multiple selection  - “Đặt ve chai” button  - “Hủy” button | | 2 | Collector chooses “Thời gian” selection [Alternative 1] |  | | 3 | Collector presses “Đặt ve chai” button [Alternative 2] | System creates booking request, send notification to Homeowner [Exception 1] [Exception 2] [Exception 3] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Collector edits “Giá” input | System creates booking request with price that Collector entered and send notification to Homeowner | | 2 | Collector presses “Hủy” button | System closes the screen and goes back to Scrap detail screen |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows error message the "Không có kết nối Internet" when the internet is lost. | | 2 |  | System shows error message the "Bài đăng này đã có người đặt" when the post is booked. | | 3 |  | System shows error message the "Bài đăng không tồn tại" when the post is deleted. |   **Relationships:** [SC\_UC\_19]  **Business Rules:**   * After booking is created, Homeowner can accept or reject booking * The created booking has “Pending” status | | | |

Table - <Collector> Book scrap use case specification

###### 2.3.4.4 Call Homeowner

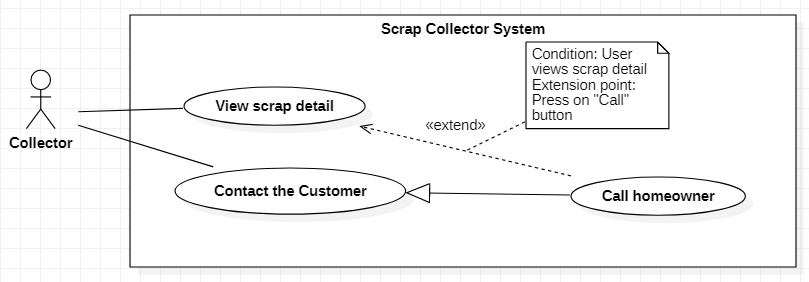


Figure - <Collector> Call homeowner

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_21** | | | |
| **Use Case No.** | SC\_UC\_21 | **Use Case Version** | 1.0 |
| **Use Case Name** | Call homeowner | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to call Homeowner.   **Goal:**   * Collector calls Homeowner for direct communication.   **Triggers:**   * User presses “Gọi” button or button with phone icon.   **Preconditions:**   * User must be a Collector * Collector views post detail   **Post conditions:**   * Success: Collector calls Homeowner. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector presses “Gọi” button or button with phone icon | Alert message: “Bạn muốn gọi cho người bán”  - “Đồng ý” button  - “Hủy” button | | 2 | Collector presses “Đồng ý” button | System uses call service of phone to make a call to Collector. [Exception 1] [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Collector presses “Hủy” button | System goes back to previous screen |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 |  | System shows message the "Không có quyền truy cập" when the application does not have permission to make a phone call. |   **Relationships:** [SC\_UC\_19]  **Business Rules:**   * Application must ask user for making phone call permission | | | |

Table - <Collector> Call homeowner use case specification

###### 2.3.4.5 Chat with Homeowner

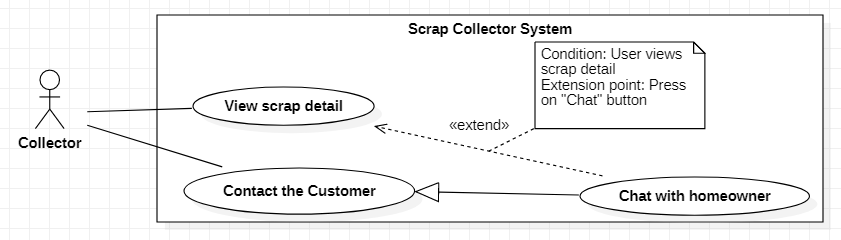


Figure - <Collector> Chat with homeowner

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_22** | | | |
| **Use Case No.** | SC\_UC\_22 | **Use Case Version** | 1.0 |
| **Use Case Name** | Chat with homeowner | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to chat with Homeowner   **Goal:**   * Collector chats with one or many Homeowner.   **Triggers:**   * User presses “Chat” button. * User presses “Chat” tab in bottom tab bar.   **Preconditions:**   * User must be a Collector * Collector views post detail   **Post conditions:**   * Success: A conversation between Homeowner and Collector is created. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector presses “Chat” button or presses “Chat” tab in tab bar | System shows “Chat” screen with all conversations of that Collector | | 2 | Collector chooses a conversation | System shows messages in chosen conversation, a text input and a “Gửi” button | | 3 | Collector inputs message and presses “Gửi” button | System sends message to the Homeowner in that conversation and shows that message [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_UC\_23]  **Business Rules:** N/A | | | |

Table - <Collector> Chat with homeowner use case specification

###### 2.3.4.6 View pick-up schedule

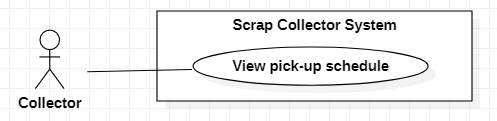


Figure - <Collector> View pick-up schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_23** | | | |
| **Use Case No.** | SC\_UC\_23 | **Use Case Version** | 1.0 |
| **Use Case Name** | View pick-up schedule | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to view pick-up schedule   **Goal:**   * Collector views optimized pick-up schedule to enhance effectivity.   **Triggers:**   * User presses “Lộ trình” button.   **Preconditions:**   * User must be a Collector   **Post conditions:**   * Success: System shows pick-up schedule. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector presses “Lộ trình” button. | System shows “Lộ trình” screen with total length of route and pick-up schedule.  Each item in schedule contains:  - Title of post  - Address of post  - Price of booking  - “Chỉ đường” button  - “Gọi” button  [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** [SC\_UC\_24]  **Business Rules:**   * All booking with “accepted” status will be shown in schedule | | | |

Table - <Collector> View pick-up schedule use case specification

###### 2.3.4.7 View direction

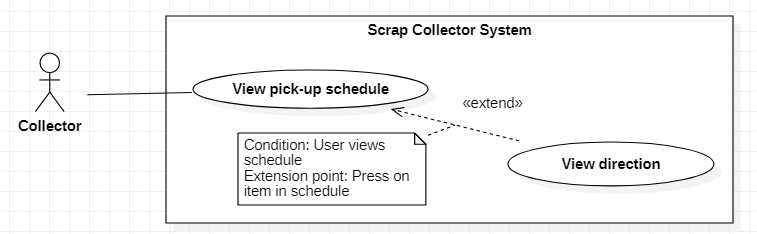


Figure - <Collector> View direction

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_24** | | | |
| **Use Case No.** | SC\_UC\_24 | **Use Case Version** | 1.0 |
| **Use Case Name** | View direction | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to view direction to the Homeowner destination.   **Goal:**   * Collector views direction to scrap’s place.   **Triggers:**   * User presses “Chỉ đường” button.   **Preconditions:**   * User must be a Collector * Collector views pick-up schedule   **Post conditions:**   * Success: System shows direction. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector presses “Chỉ đường” button on the item in schedule | System shows pop up with 2 selection:  - “Chỉ đường theo lộ trình”  - “Chỉ đường từ vị trí hiện tại”  [Exception 1][Exception 2] | | 2 | Collector presses “Chỉ đường theo lộ trình” selection | System shows “Chỉ đường” screen with direction from previous item’s location to current item’s location  [Exception 1][Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Collector presses “Chỉ đường từ vị trí hiện tại” selection | System shows “Chỉ đường” screen with direction from Collector’s current location to scrap’s location  [Exception 1][Exception 2] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 |  | System shows message the "Không thể tải bản đồ” when the Map API can’t be connected. |   **Relationships:** [SC\_US\_23]  **Business Rules:**   * Application must ask user for getting GPS location permission. | | | |

Table - <Collector> View direction use case specification

###### 2.3.4.8 View booking response

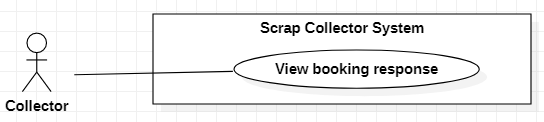


Figure - <Collector> View booking response

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_25** | | | |
| **Use Case No.** | SC\_UC\_25 | **Use Case Version** | 1.0 |
| **Use Case Name** | View booking response | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to view scrap booking response.   **Goal:**   * Collector receives notification when Homeowner accepts or rejects Collector’s booking request.   **Triggers:**   * Collector receives booking response notification.   **Preconditions:**   * User must be a Collector   **Post conditions:**   * Success: System shows response from Homeowner. * Fail: System shows error messages. Post is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector presses on the booking response notification | System shows detail of booking response that whether it is accepted or is rejected.  [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table - <Collector> View booking response use case specification

###### 2.3.4.9 View booking scrap list

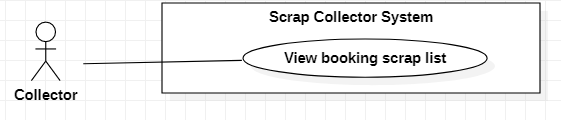


Figure - <Collector> View booking scrap list

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_26** | | | |
| **Use Case No.** | SC\_UC\_26 | **Use Case Version** | 1.0 |
| **Use Case Name** | View booking scrap list | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to view booking request list   **Goal:**   * Collector views all booking requests that they sent to Homeowner   **Triggers:**   * User presses on “Đang chờ xác nhận” tab.   **Preconditions:**   * User must be a Collector   **Post conditions:**   * Success: System shows all booking requests of that Collector. * Fail: System shows error messages. Post is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User presses on “Đang chờ xác nhận” tab | System shows all booking requests of that Collector. Each item in the list contains:  - Image of the post  - Title of the post  - Time of the post  - Distance of the post  - Booking price  - Time to pick up  - “Hủy” button  [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message the "Không có kết nối Internet" when the internet is lost |   **Relationships:** [SC\_UC\_27]  **Business Rules:**   * Only booking requests that have “Pending” status are shown | | | |

Table - <Collector> View booking scrap list use case specification

###### 2.3.4.10 Cancel booking

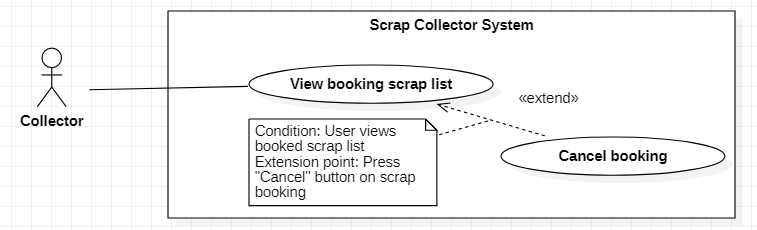


Figure - <Collector> Cancel booking

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_27** | | | |
| **Use Case No.** | SC\_UC\_27 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel booking | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to delete booking request.   **Goal:**   * Collector cancels booking request.   **Triggers:**   * Collector presses “Hủy” button on each Item in the booking request list.   **Preconditions:**   * User must be a Collector * Collector view booking scrap list   **Post conditions:**   * Success: The booking request is deleted. * Fail: System shows error messages. Booking request is not deleted.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector presses “Hủy” button on each Item in the booking request list | System shows a pop up with:  - “Bạn có muốn hủy đặt không” message  - “Có” button  - “Trở về” button | | 2 | Collector presses “Có” button on the pop up [Alternative 1] | System deletes booking request, shows result to the Collector and sends notification to the Homeowner [Exception 1] [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Collector presses “Trở về” button on the pop up | System closes the pop up and goes back to booking request list screen |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message the "Không có kết nối Internet" when the internet is lost | | 2 |  | System shows message the "Không thể hủy yêu cầu" when the booking is accepted or rejected by the Homeowner |   **Relationships:** [SC\_UC\_26]  **Business Rules:**   * Only booking requests that have “Pending” status can be deleted. | | | |

Table - <Collector> Cancel booking use case specification

###### 2.3.4.11 View review from Homeowner

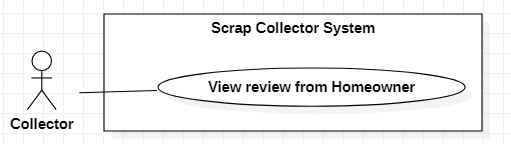


Figure 34 - <Collector> View review from Homeowner

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_28** | | | |
| **Use Case No.** | SC\_UC\_28 | **Use Case Version** | 1.0 |
| **Use Case Name** | View review from Homeowner | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Collector   **Summary:**   * Allow Collector to view his/her review   **Goal:**   * Collector views his/her review from the Homeowner   **Triggers:**   * Collector receives notification of review from Homeowner   **Preconditions:**   * User must be a Collector   **Post conditions:**   * Success: System shows Collector’s review * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Collector receives notification of review | System shows a pop-up with:  - The name of the Homeowner who made review  - The number of star on total of 5 stars  - The description from Homeowner  - “Tôi đã đọc” button  [Exception 1] | | 2 | Collector presses “Tôi đã đọc” button on the pop-up | System closes pop-up |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table - <Collector> View review from Homeowner use case specification

##### 2.3.5 <Admin> Overview Use Case

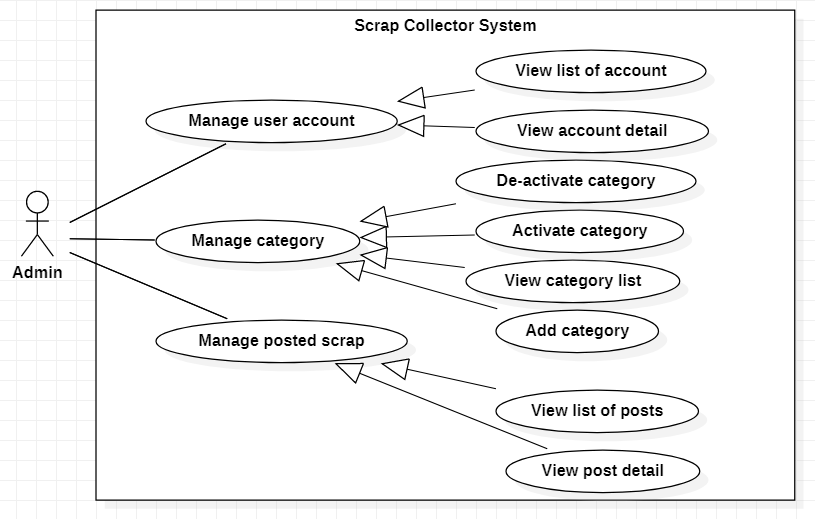


Figure – Admin Overview Use Case

###### 2.3.5.1 View list of account

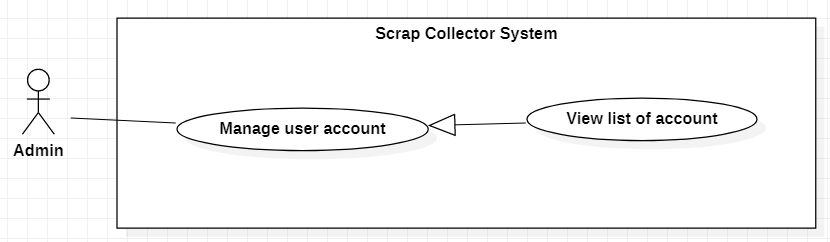


Figure - <Admin> View list of account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_29** | | | |
| **Use Case No.** | SC\_UC\_29 | **Use Case Version** | 1.0 |
| **Use Case Name** | View list of account | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to view all account in the system   **Goal:**   * Admin views all accounts in the system   **Triggers:**   * Admin go to account screen   **Preconditions:**   * User must be an Admin   **Post conditions:**   * Success: A list of account is shown on screen. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin go to Account screen on website | System show list of accounts, filter by 2 role: “Homeowner” and “Collector”.  [Exception 1] [Exception 2] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 |  | System shows message the "Không thể tải được" when there is error. |   **Relationships:** Generalized from Manage user account use case  **Business Rules:**   * All accounts are shown even if that account is disabled | | | |

Table - <Admin> View list of account use case specification

###### 2.3.5.2 View account detail

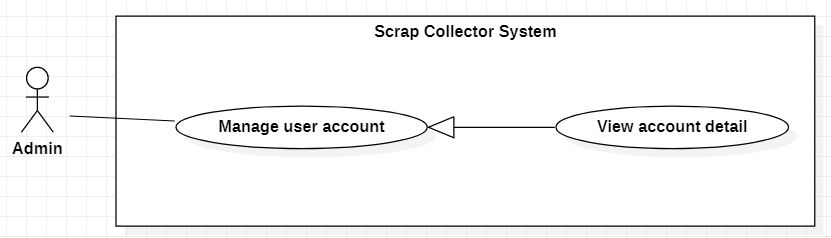


Figure - <Admin> View account detail

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_30** | | | |
| **Use Case No.** | SC\_UC\_30 | **Use Case Version** | 1.0 |
| **Use Case Name** | View account detail | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to view an account’s detail information   **Goal:**   * Admin view account detail.   **Triggers:**   * Admin click on the item of user in account list.   **Preconditions:**   * User must be an Admin * Admin view list of account   **Post conditions:**   * Success: User detail is shown. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin click on user item | System shows user’s detail information with:   * Name: text * Username: text * Phone number: text * Email: text * Detail address: text * Province: text * District: text * Ward: text * Status: is deleted or not * List of sold scrap   [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** Generalized from Manage user account use case  **Business Rules: N/A** | | | |

Table - <Admin> View account detail use case specification

###### 2.3.5.3 View list of post

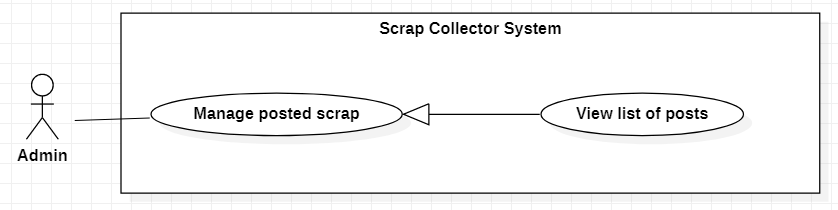


Figure - <Admin> View list of post

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_31** | | | |
| **Use Case No.** | SC\_UC\_31 | **Use Case Version** | 1.0 |
| **Use Case Name** | View list of post | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to view all scraps in the system.   **Goal:**   * Admin views all scrap.   **Triggers:**   * Admin click to go to Scrap screen.   **Preconditions:**   * User must be an Admin   **Post conditions:**   * Success: A list of scrap is shown. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin goes to Scrap screen | System shows list of scraps, filter by status: ‘Selling’, ‘Booked’ and ‘Sold’  [Exception 1] [Exception 2] | | 2 | [Alternative 1] [Alternative 2] [Alternative 3] |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin click on ‘Selling’ tab | System shows list of scraps with having status ‘Selling’ | | 2 | Admin click on ‘Booked tab | System shows list of scraps with having status ‘Booked | | 3 | Admin click on ‘Sold tab | System shows list of scraps with having status ‘Sold |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. | | 2 |  | System shows message the "Không thể tải được" when there is error. |   **Relationships:** Generalized from Manage posted scrap use case  **Business Rules: N/A** | | | |

Table - <Admin> View list of post use case specification

###### 2.3.5.4 View post detail

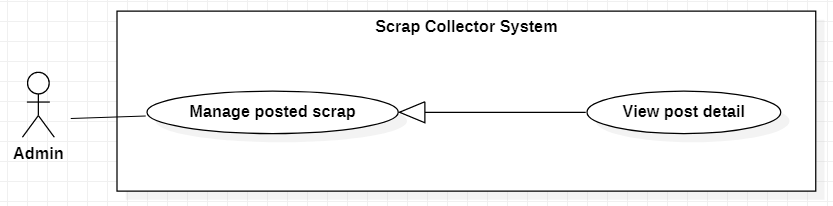


Figure - <Admin> View post detail

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_32** | | | |
| **Use Case No.** | SC\_UC\_32 | **Use Case Version** | 1.0 |
| **Use Case Name** | View post detail | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to view post detail   **Goal:**   * Admin views post detail.   **Triggers:**   * Admin click on post item in post list.   **Preconditions:**   * User must be an Admin * Admin views list of post   **Post conditions:**   * Success: Post detail is shown. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin clicks on post detail item | System shows post detail with:   * Title: text * Price: text * Date of posting: timestamp * Category: text * Quantity: text number * Image: images * Phone number: text * Address: text * Homeowner’s name: text * Homeowner’s id: text   [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message “Không có kết nối Internet” when the Internet is lost. |   **Relationships:** Generalized from Manage posted scrap use case  **Business Rules: N/A** | | | |

Table - <Admin> View post detail use case specification

###### 2.3.5.5 Activate category

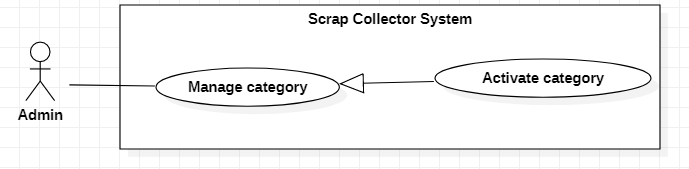


Figure - <Admin> Activate category

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_33** | | | |
| **Use Case No.** | SC\_UC\_33 | **Use Case Version** | 1.0 |
| **Use Case Name** | Activate category | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to activate a category.   **Goal:**   * Admin changes status of category so that category can be used.   **Triggers:**   * Admin clicks on “Activate” button on Category item.   **Preconditions:**   * User must be an Admin * Admin view list of category   **Post conditions:**   * Success: The category is activated. * Fail: System shows error messages. Category is not activated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin clicks on “Activate” button on Category item | System changes status of category.  [Exception 1] | | 2 |  | System shows result. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message "Không có kết nối Internet" when the Internet is lost. |   **Relationships:** Generalized from Manage category use case  **Business Rules: N/A** | | | |

Table - <Admin> Activate category use case specification

###### 2.3.5.6 Deactivate category

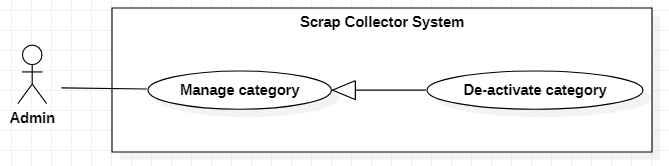


Figure - <Admin> Deactivate category

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_34** | | | |
| **Use Case No.** | SC\_UC\_34 | **Use Case Version** | 1.0 |
| **Use Case Name** | Deactivate category | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to deactivate a category   **Goal:**   * Admin changes status of category so that category cannot be used.   **Triggers:**   * Admin clicks “Deactivate” button on category item.   **Preconditions:**   * User must be a Admin * Admin views list of category   **Post conditions:**   * Success: Category is deactivated. * Fail: System shows error messages. Category is not deactivated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin clicks on “Deactivate” button on Category item | System changes status of category.  [Exception 1] | | 2 |  | System shows result. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message "Không có kết nối Internet" when the Internet is lost. |   **Relationships:** Generalized from Manage category use case  **Business Rules:**   * Only categories that have no scrap can be shown as can be deactivated. | | | |

Table - <Admin> Deactivate category use case specification

###### 2.3.5.7 Add new category

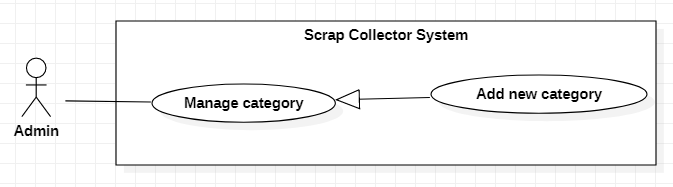


Figure - <Admin> Add new category

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_35** | | | |
| **Use Case No.** | SC\_UC\_35 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add new category | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to add new category   **Goal:**   * Admin adds a new category.   **Triggers:**   * Admin presses “Add category” button.   **Preconditions:**   * User must be a Admin * Admin views list of category   **Post conditions:**   * Success: A new category is created. * Fail: System shows error messages. Category is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin clicks “Add category” button | Systems shows a modal for Admin to add category, with:   * Category name(required): text input * Unit(required): text input * “Add” button * “Cancel” button | | 2 | Admin inputs required fields |  | | 3 | Admin clicks on “Add” button  [Alternative 1] | Systems add category and shows result  [Exception 1] [Exception 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Admin clicks on “Cancel” button | System close modal. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message "Không có kết nối Internet" when the Internet is lost. | | 2 |  | System shows message “Phân loại này đã tốn tại” when there is the inputted category is duplicated. |   **Relationships:** Generalized from Manage category use case  **Business Rules: N/A** | | | |

Table - <Admin> Add new category use case specification

###### 2.3.5.8 View category list

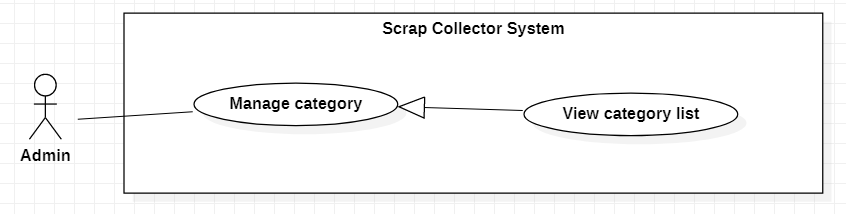


Figure - <Admin> View category list

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – SC\_UC\_36** | | | |
| **Use Case No.** | SC\_UC\_36 | **Use Case Version** | 1.0 |
| **Use Case Name** | View category list | | |
| **Author** | Nguyen Duy Dat | | |
| **Date** | 26/2/2020 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * Allow Admin to view all categoríe in the system   **Goal:**   * Admin views all categories.   **Triggers:**   * Admin goes to category screen.   **Preconditions:**   * User must be a Admin   **Post conditions:**   * Success: List of category is shown. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin goes to category screen | System shows list of category, each category is shown with:   * Name: text * Unit: text * “Activate” button * “Deactivate” button   [Exception 1] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System shows message "Không có kết nối Internet" when the Internet is lost. |   **Relationships:** Generalized from Manage category use case  **Business Rules:**   * If category is deactivated, the “Deactivate” button is disabled * If category is activated, the “Activate” button is disabled * If a category has scrap that is being sell, that category is marked as cannot be deactivated. | | | |

Table - <Admin> View category list use case specification

## Software System Attribute

#### 3.1 Usability

* The system is easy to use that user needs average 2 hours of training to use the system.
* User who does not know about computer science or programming still understands the meaning of user interface

#### 3.2 Reliability

* Every information stored in the database is permanent
* Exception rate is at minimum 1% over all working time

#### Availability

* The system must be available at any time 24/7, except for some system maintenances

#### Security

* Input data must be validated before being saved to the database
* Only Admin has fully access to the web application
* Password is encrypted before being saved to the database
* For each request to get API from user, server checks user’s authorization
* The code is based on popular design patterns in order to easy to maintain and upgrade

#### Maintainability

* The system is divided into separated sub-systems or modules for easy maintain

#### Portability

* The website is running on Windows 7 or above.
* Easy installation from APK for Android version.
* The mobile application is running on Android 6.0.1 or above.

#### Performance

* System has successfully test with basic standard
* All requests are handling in maximum duration of 10 seconds
* System has successfully test with basic standard

## Conceptual Diagram

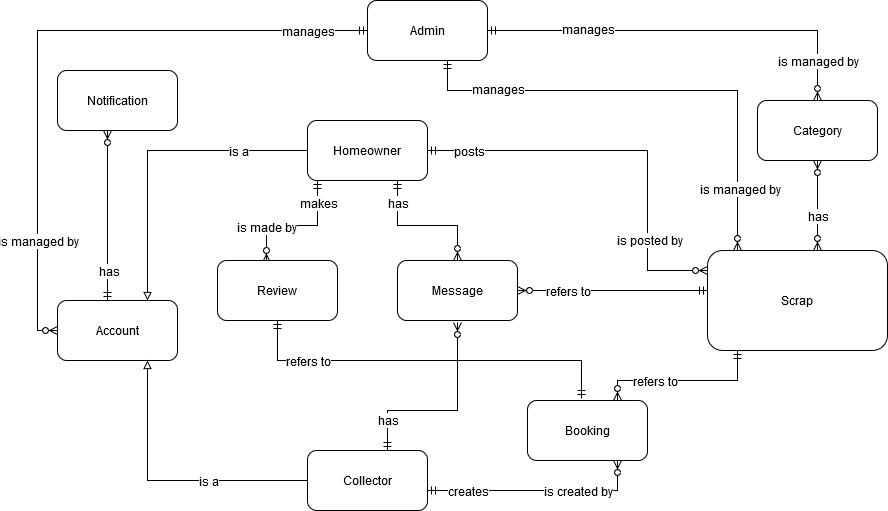
****

Figure - Conceptual Diagram

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Data dictionary: describe all content of all entities** | |
| **Entity Name** | **Description** |
| **Account** | Abstract entity describes a user in system |
| **Homeowner** | Contain the homeowner information |
| **Collector** | Contain the collector information |
| **Admin** | Contain the admin information |
| **Notification** | Contain the notification information |
| **Scrap** | Contain the scrap information |
| **Category** | Contain the category information |
| **Booking** | Contain the booking information |
| **Review** | Contain the review information |
| **Message** | Contain the message information |

Table - Conceptual Diagram Data Dictionary

# Software Design Description

## Design Overview

* This document describes the technical and user interface design of Scrap Collector System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagrams that describe the connection and integration of the system.
* Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: describe screens design.
* Section 6: describe a fully attributed ERD.
* Section 7: describe algorithms

## System Architectural Design

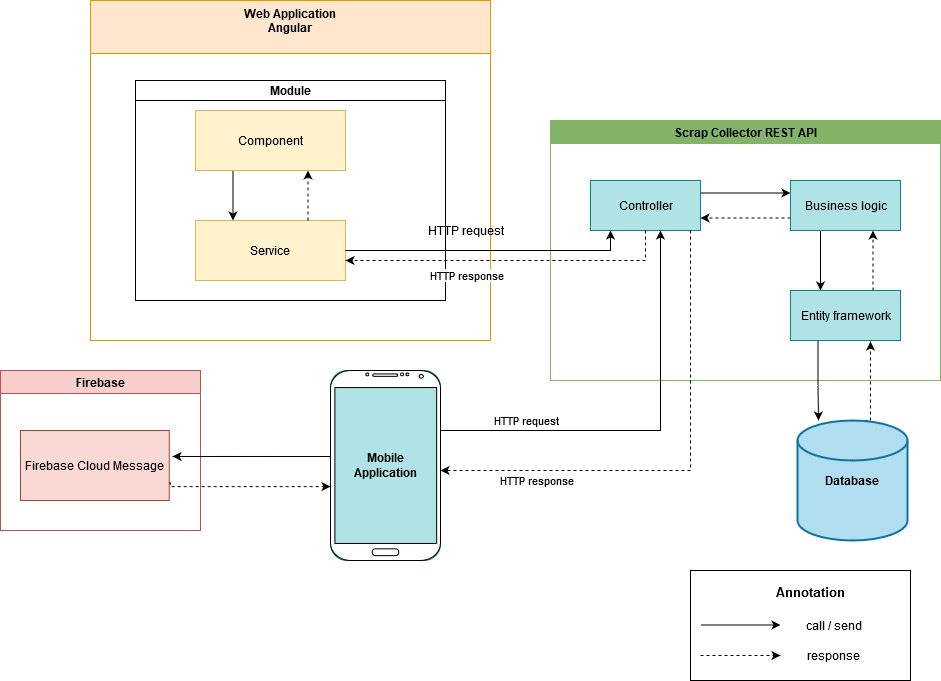


Figure - System Architectural Design

#### 2.1. Web Application Architecture Description

Web application is used by Admin only. We choose Angular to build web application. An Angular application is defined by a set of modules. An app always has at least a root module that enables bootstrapping, and typically has many more feature modules. Component defines views, which are sets of screen elements that Angular can choose among and modify according to the program logic and data. And components use service to communicate and transfer data to each other.

#### 2.2. Mobile Application Architecture Description

Mobile application is used for both Homeowner and Collector. We choose to implement mobile application with React Native because React Native supports our team to build an application that can run on both Android and iOS. Moreover, React Native supports hot-loading feature, which helps our team member to implement code and debug faster.

React introduced a Component-Based-Architecture (CBA) concept, a method for encapsulating individual parts of a large user interface into independent micro-systems. Moreover, CBA splits responsibilities vertically, which is suitable for each member’s responsibilities in capstone requirement.

For real-time notification in mobile application, we use Cloud Messaging service of Firebase. Firebase Cloud Messaging helps to send message to cross-platform devices and lets you reliably send message at no cost.



Figure – Mobile Application Architecture

In Scrap Collector Mobile Application, to handle data flow and manage state for smoothing UI, we choose to use React Redux. React Redux helps to read data from store through a global state, dispatch actions to store to update data and call web service (Rest API) through middleware service.

#### 2.3. Rest API Architecture Description

In Scrap Collector System, Mobile application and Web application are separated, so they need a web service as a middle component to retrieve and update data from database. Web service also helps to change business logic easily. In web service, we use Controller – Business Login – Entity Framework structure, which is one of the most popular structures being used in modern systems.

## Component Diagram

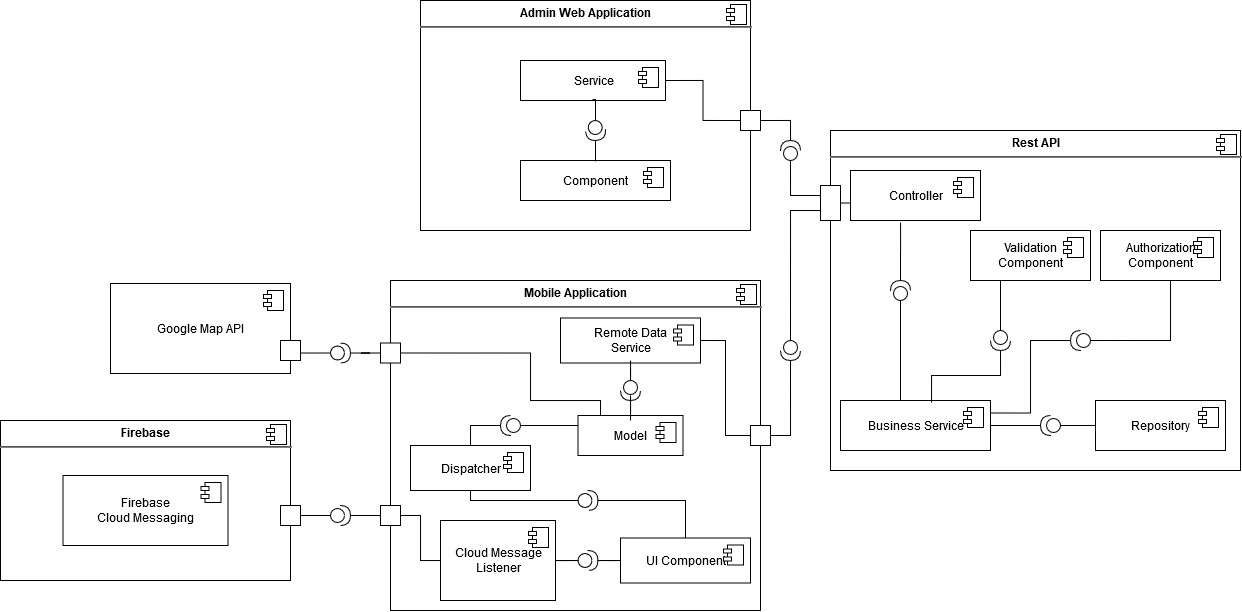


Figure – Component Diagram

|  |  |
| --- | --- |
| COMPONENT DICTIONARY: DESCRIBES COMPONENTS | |
| Component name | **Description** |
| Rest API | Component which contains core flow of Scrap Collector system. Rest API receives request from Web Application and Mobile Application, solves business logic, retrieves data from database and then returns result. |
| Admin Web Application | Component which handles HTTP requests from Admin |
| Mobile Application | Component which helps Homeowner and Collector implement features in the system through UI Components |
| Firebase | Component with Cloud messaging service to send notification to Mobile Application |
| Google Map API | Component which solve HTTP request of calculating distance and direction |

Table - Component Diagram Dictionary

## Detailed Description

#### 4.1 Class Diagram

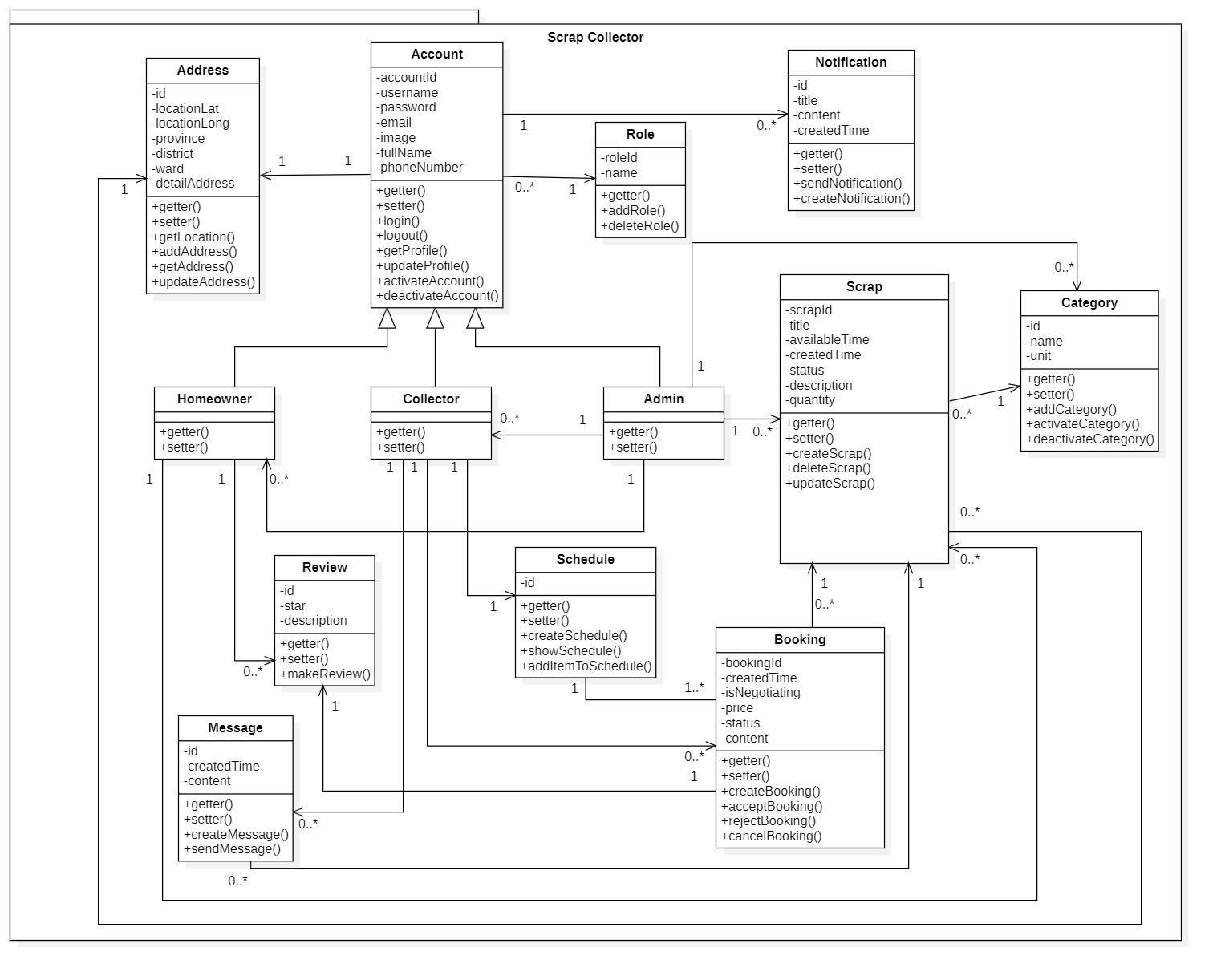


Figure 48 – Class Diagram

**Class Diagram Explanation**

|  |  |  |
| --- | --- | --- |
| **CLASS DICTIONARY: DESCRIBE CLASS** | | |
| **Class Name** | **Mapping column with Conceptual diagram** | **Description** |
| **Account** | Account | Contains the account information |
| **Role** | N/A | Does not exist in conceptual diagram, but needed in class diagram to contain the role information in general |
| **Homeowner** | Homeowner | Contains the homeowner information |
| **Collector** | Collector | Contains the collector information |
| **Admin** | Admin | Contains the admin information |
| **Address** | N/A | Does not exist in conceptual diagram, but needed in class diagram to contain the address information in general |
| **Notification** | Notification | Contains the notification information |
| **Review** | Review | Contains the review information |
| **Schedule** | N/A | Does not exist in conceptual diagram, but needed in class diagram to contain the schedule information in general |
| **Scrap** | Scrap | Contains the scrap information |
| **Booking** | Booking | Contains the booking information |
| **Category** | Category | Contains the category information |
| **Message** | Message | Contains the message information in general |

Table - Class Diagram Dictionary

##### 4.1.1 Account

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| accountId | String | Private | Unique identifier of an account |
| username | String | Private | Account’s username |
| password | String | Private | Account’s password |
| fullname | String | Private | Account’s full name |
| email | String | Private | Account’s email |
| phoneNumber | String | Private | Account’s phone number |
| image | Image | Private | Account’s image |

Table - <Class Diagram Attributes> Account

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| login | Object | Public | Login |
| lougout | Object | Public | Logout |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| updateProfile | Object | Public | Edit information |
| getProfile | Object | Public | Get profile |
| activateAccount | Object | Public | Activate account |
| deactivateAccount | Void | Public | Deactivate account |

Table 47 **-** <Class Diagram Methods> Account

##### 4.1.2 Role

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| roleId | Long | Private | Unique identifier of a role |
| name | String | Private | Role’s name |

Table 48 <Class Diagram Attributes> Role

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| addRole | Boolean | Public | Add new role |
| deleteRole | Boolean | Public | Delete role |

Table - <Class Diagram Methods> Role

##### 4.1.3 Homeowner

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |

Table 50 <Class Diagram Methods> Homeowner

##### 4.1.4 Collector

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |

Table - <Class Diagram Methods> Dictionary

##### 4.1.5 Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |

Table - <Class Diagram Methods> Admin

##### 4.1.6 Address

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Long | Private | Unique identifier of an address |
| locationLat | Float | Private | Latitude of the address |
| locationLong | Float | Private | Longitude of the address |
| province | String | Private | Name of province or city |
| district | string | Private | Name of district |
| ward | String | Private | Name of ward |
| detailAddress | String | Private | House number and street |

Table 53 <Class Diagram Attributes> Address

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| getLocation | Object | Public | Get latitude and longitude |
| addAddress | Object | Public | Add new address |
| getAddress | Object | Public | Get address by id |
| updateAddress | Object | Public | Update address information |

Table 54 <Class Diagram Methods> Address

##### 4.1.7 Notification

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Long | Private | Unique identifier of a notification |
| title | String | Private | Notification’s title |
| content | String | Private | Notification’s content |
| createdTime | Datetime | Private | Notification’s time |

Table 55 <Class Diagram Attributes> Notification

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| sendNotification | Boolean | Public | Send notification to user |
| createNotification | Object | Public | Create a notification |

Table 56 <Class Diagram Methods> Notification

##### 4.1.8 Review

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Long | Private | Unique identifier of a review |
| star | Float | Private | Number of star |
| description | String | Private | Description of the review |

Table 57 <Class Diagram Attributes> Review

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| makeReview | Object | Public | Create a review |

Table 58 <Class Diagram Methods> Review

##### 4.1.9 Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Long | Private | Unique identifier of a schedule |

Table <Class Diagram Attributes> Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| createSchedule | Object | Public | Create a schedule |
| showSchedule | Void | Public | Show schedule |
| addItemToSchedule | Boolean | Public | Add an item into schedule |

Table 60 <Class Diagram Methods > Schedule

##### 4.1.10 Scrap

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Long | Private | Unique identifier of a post |
| title | String | Private | Scrap’s title |
| price | String | Private | Scrap’s price |
| avalaibleTime | Array | Private | Time Collector can pick up |
| status | String | Private | Scrap’s status |
| quantity | Float | Private | Quantity of the scrap |
| createdTime | Datetime | Private | Created Time of the post |

Table 61 <Class Diagram Attributes> Scrap

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| createScrap | Object | Public | Create a scrap |
| deleteScrap | Boolean | Public | Delete a scrap |
| updateScrap | Object | Public | Update scrap information |

Table < Class Diagram Methods > Scrap

##### 4.1.11 Category

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Long | Private | Unique identifier of a category |
| name | String | Private | Category name |
| unit | String | Private | Unit of category |

Table 63 <Class Diagram Attributes> Category

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| addCategory | Object | Public | Add new category |
| activateCategory | Boolean | Public | Activate a category |
| deactivateCategory | Boolean | Public | Deactivate a category |

Table 64 <Class Diagram Methods> Category

##### 4.1.12 Booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| bookingId | Long | Private | Unique identifier of a booking |
| createdTime | Datetime | Private | Created time of booking |
| price | Float | Private | Price of the booking |
| status | String | Private | Page’s title |
| content | String | Private | Message from Collector to Homeowner |
| isNegotiating | Boolean | Private | Whether the price is negotiated or not |

Table 65 <Class Diagram Attributes> Booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| createBooking | Object | Public | Create new booking |
| acceptBooking | Boolean | Public | Accept booking |
| rejectBooking | Boolean | Public | Reject booking |
| cancelBooking | Boolean | Public | Cancel a booking |

Table <Class Diagram Methods> Booking

##### 4.1.13 Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Long | Private | Unique identifier of a message |
| createdTime | Datetime | Private | Created time of the message |
| content | String | Private | Content of the message |

Table 67 <Class Diagram Attributes> Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Redirect Type** | **Visibility** | **Description** |
| getter | Object | Public | Get attribute value |
| setter | Void | Public | Set attribute value |
| createMessage | Object | Public | Create new message |
| sendMessage | Void | Public | Send message |

Table <Class Diagram Methods> Message

#### 4.2 Sequence Diagram

##### 4.2.1 Get Category List

Summary: This diagram shows how the Admin gets list of category

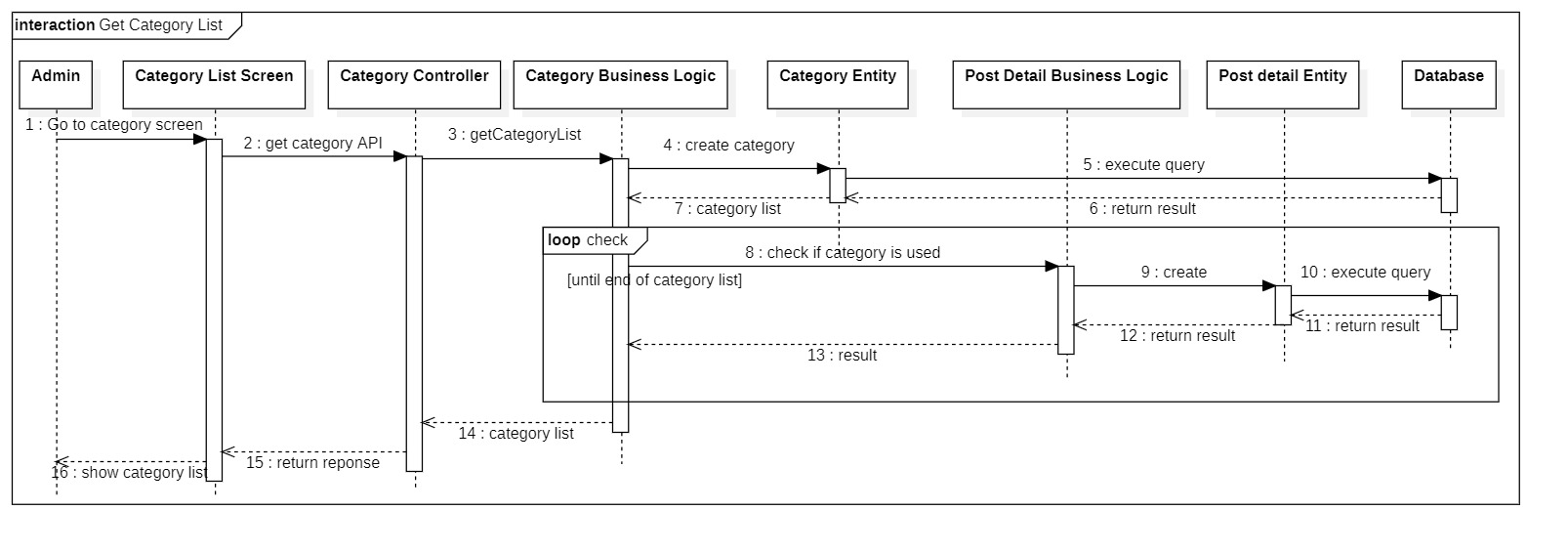
****

Figure - <Sequence Diagram> Get category list

##### 4.2.2 Add Category

Summary: This diagram shows how the Admin adds a category

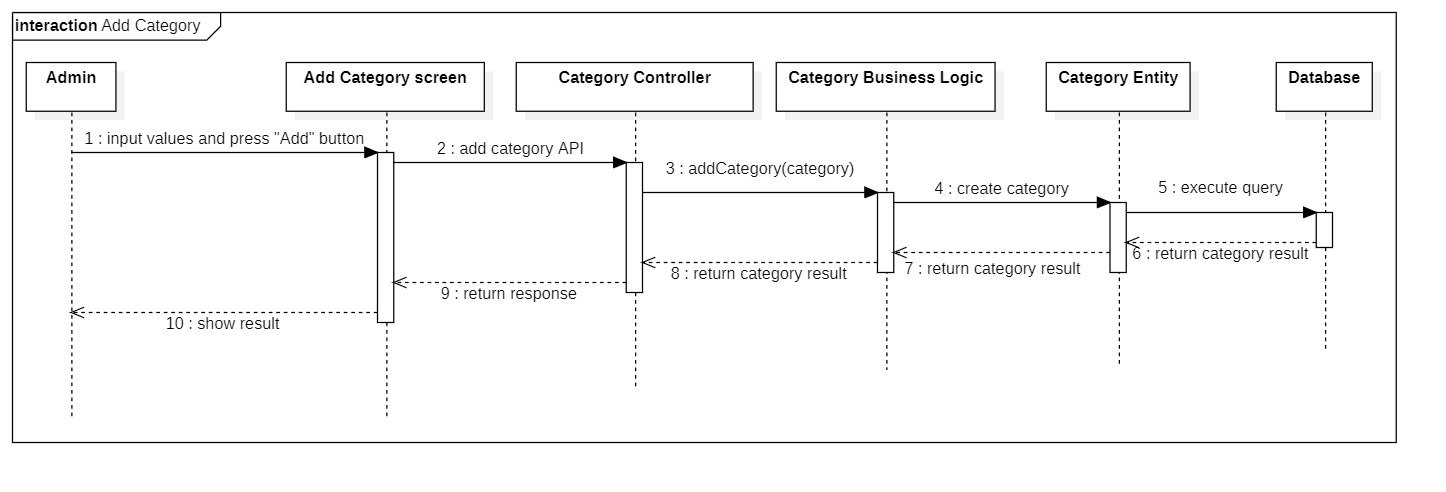


Figure - <Sequence Diagram> Add category

##### 4.2.3 Activate Category

Summary: This diagram shows how the Admin activates a category

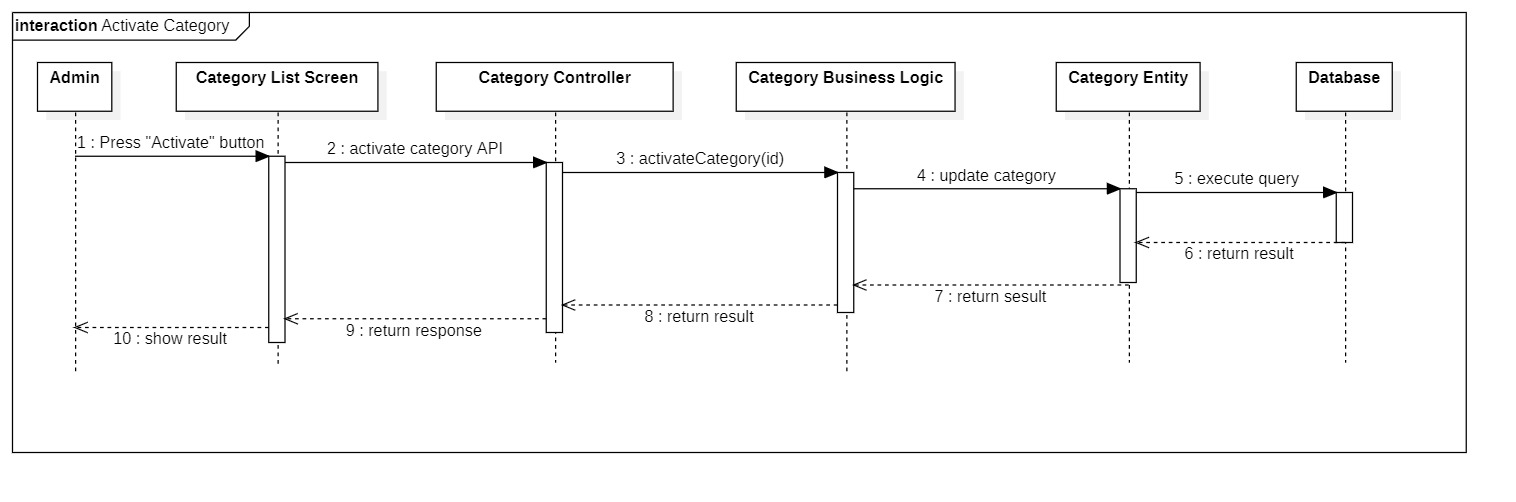


Figure - <Sequence Diagram> Activate category

##### 4.2.4 Deactivate Category

Summary: This diagram shows how the Admin deactivates a category

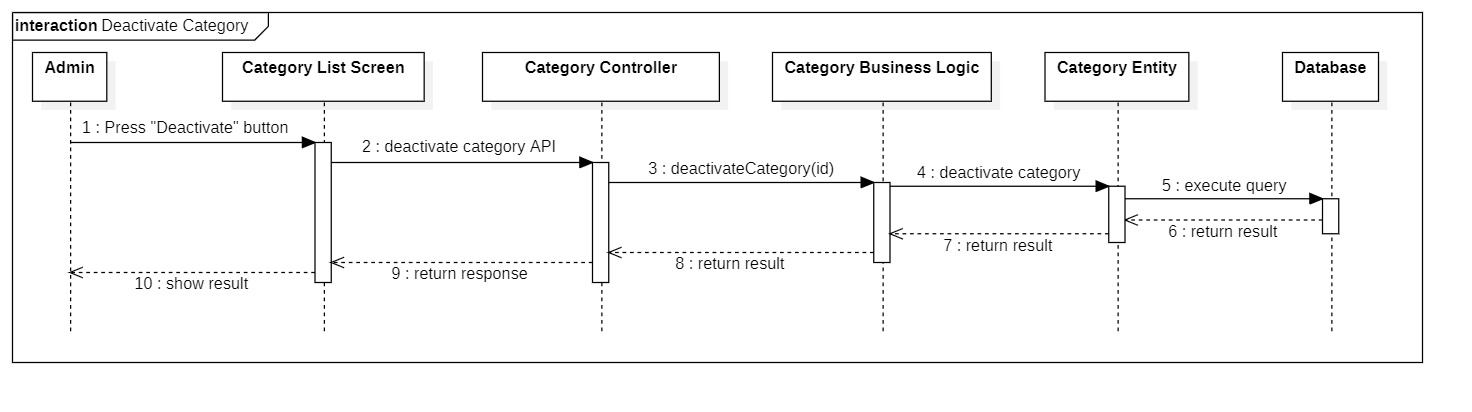


Figure - <Sequence Diagram> Deactivate category

##### 4.2.5 Get Account List

Summary: This diagram shows how the Admin gets list of accounts (Homeowner and Collector) in Scrap Collector system.

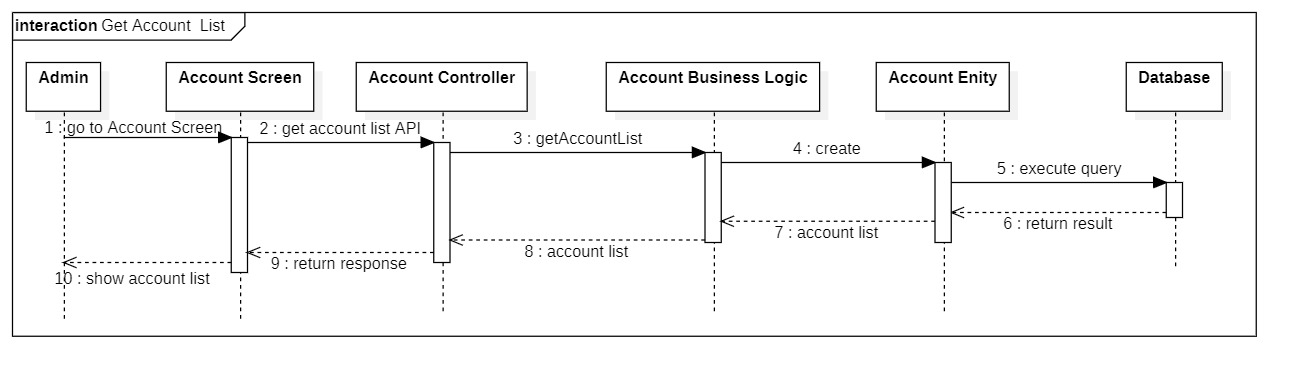


Figure - <Sequence Diagram> Get account list

##### 4.2.6 Get Account Detail

Summary: This diagram shows how the Admin views the information of an account in the system.

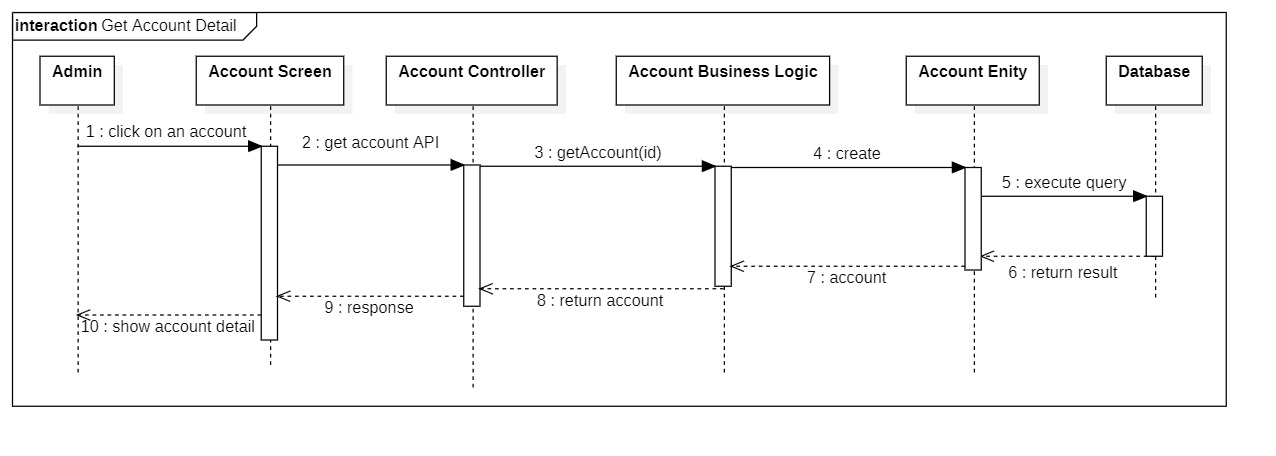


Figure - <Sequence Diagram> Get account detail

##### 4.2.7 Get Post List

Summary: This diagram shows how the Admin gets list of post

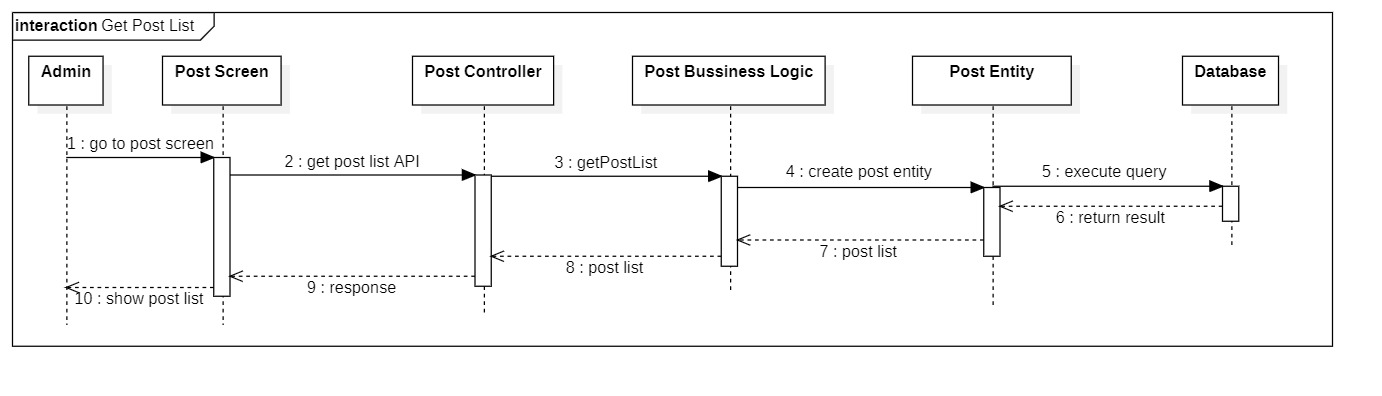


Figure - <Sequence Diagram> Get post list

##### 4.2.8 Get Post Detail

Summary: This diagram shows how the Admin views detail information of a post in the system.

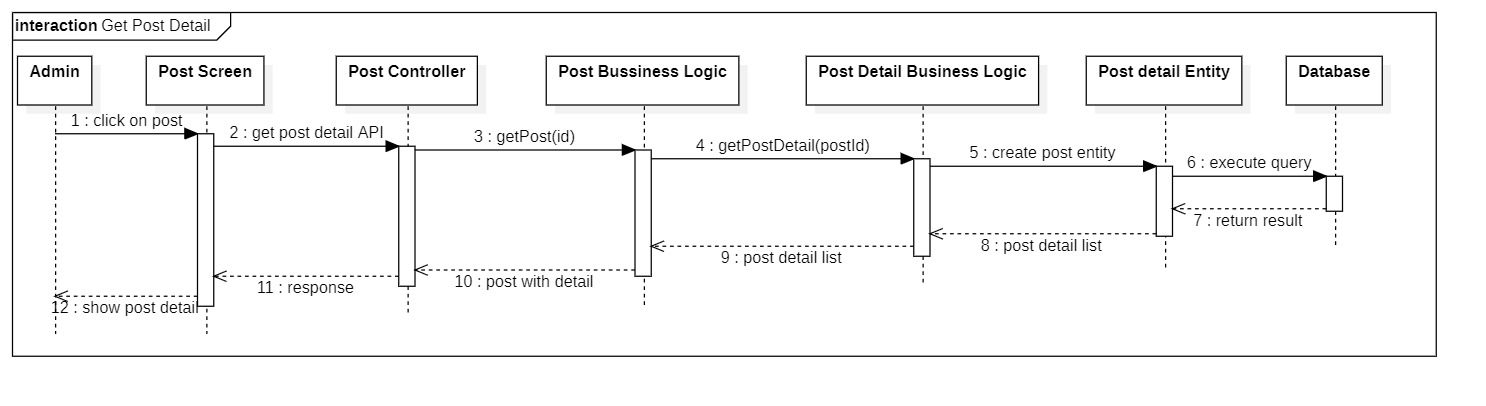


Figure - <Sequence Diagram> Get post detail

#### 4.3 Activity Diagram

##### 4.3.1 Login

Summary: This diagram shows how a user logins into System.



Figure - <Activity Diagram> Login

##### 4.3.2 Register

Summary: This diagram shows how a guest registers an account.

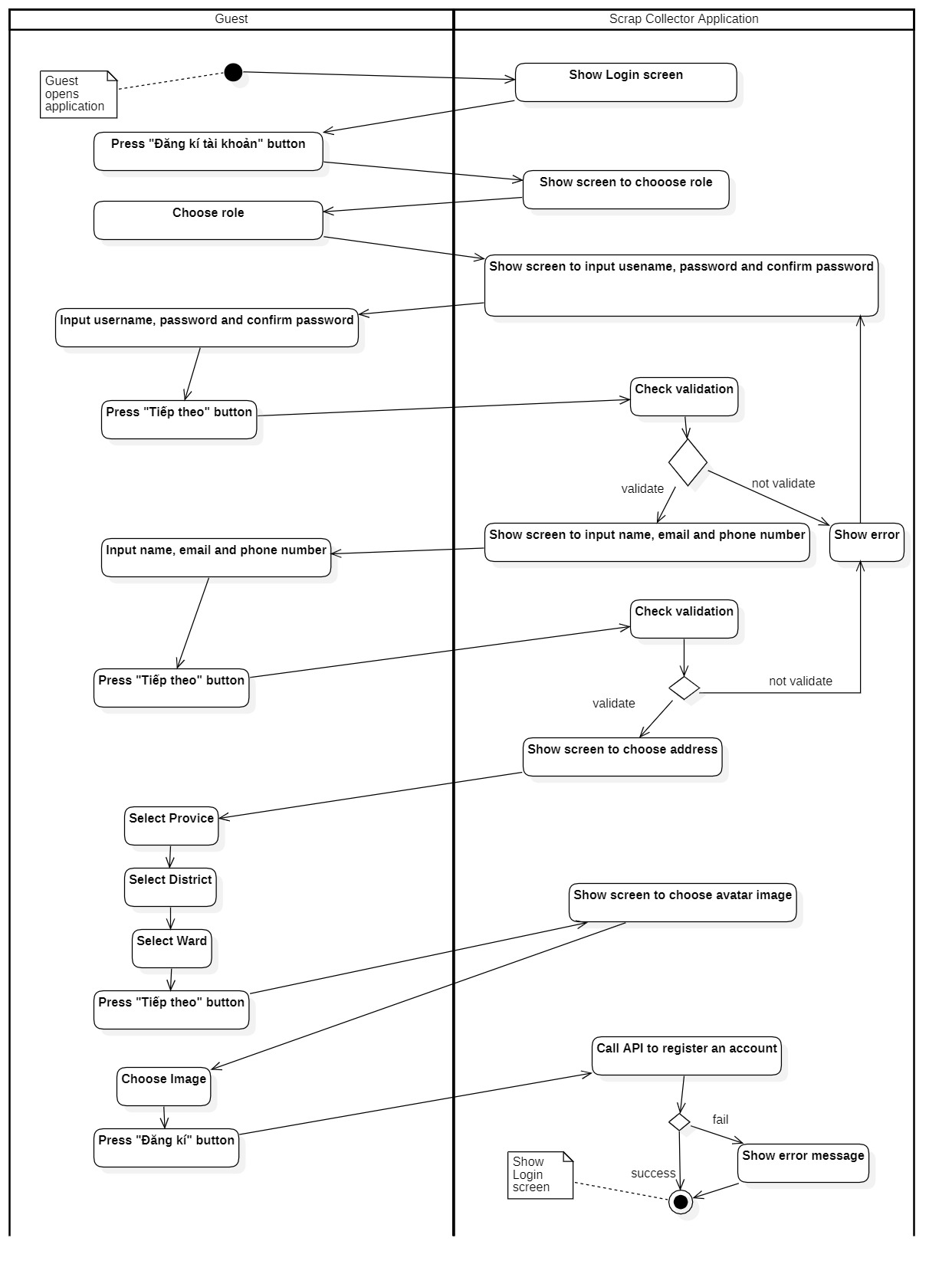


Figure - <Activity Diagram> Register

##### 4.3.3 Post scrap

Summary: This diagram shows how a Homeowner creates a post to sell scrap.

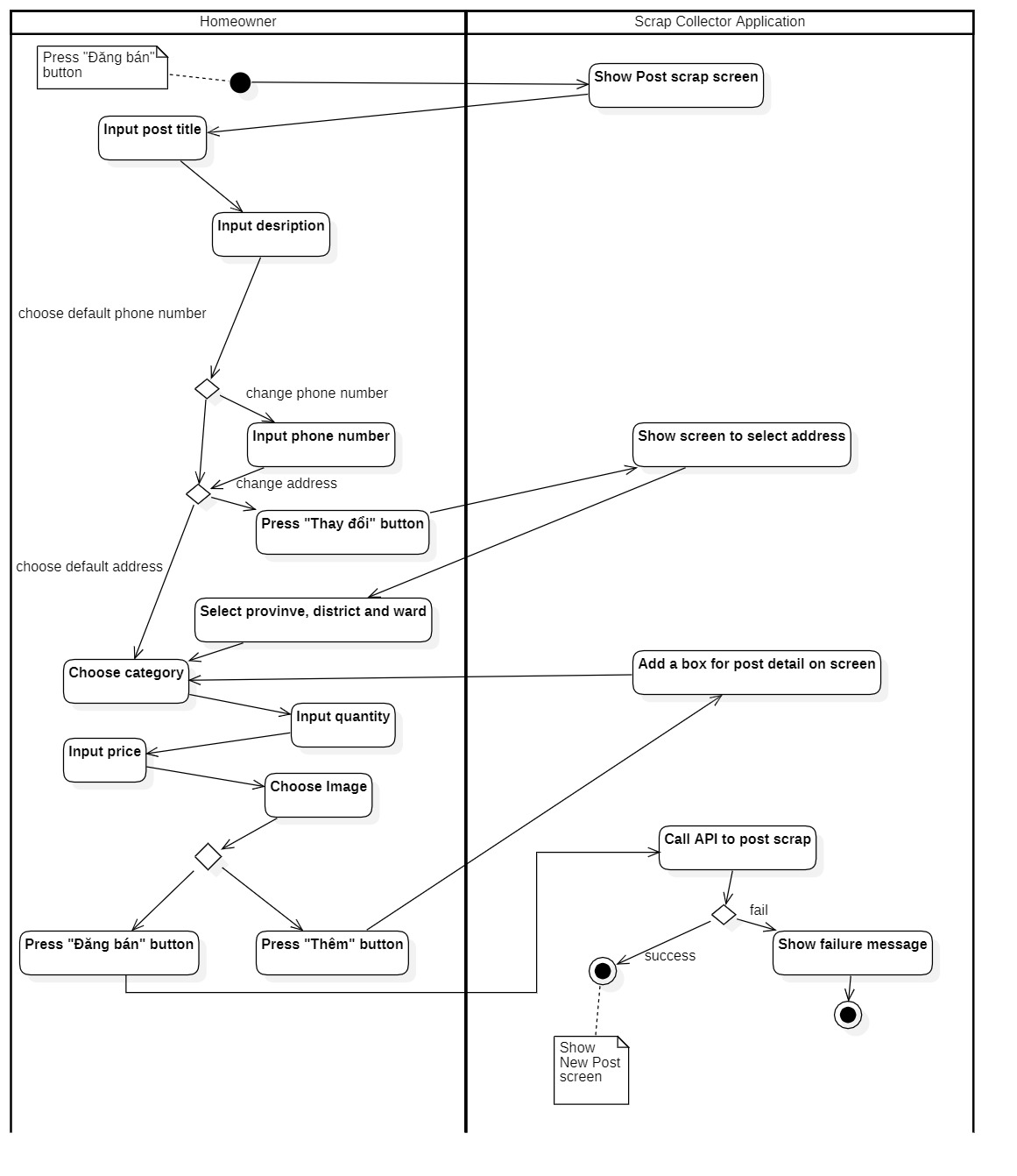


Figure - <Activity Diagram> Post scrap

##### 4.3.4 View Booking request

Summary: This diagram shows how a Homeowner views all waiting booking requests.

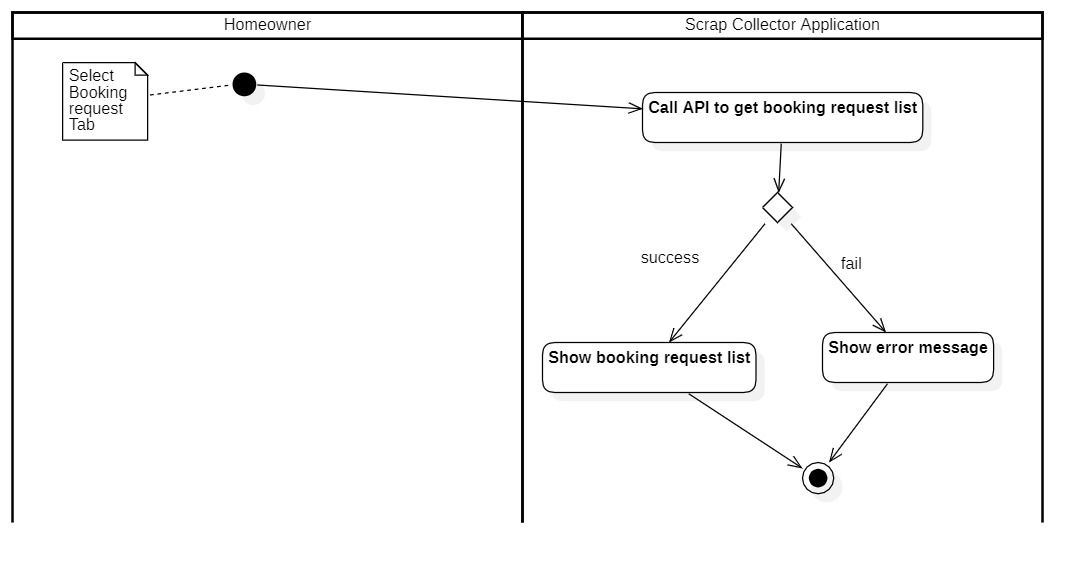
****

Figure - <Activity Diagram> View booking request

##### 4.3.5 Accept Booking request

Summary: This diagram shows how a Homeowner accepts a booking request.

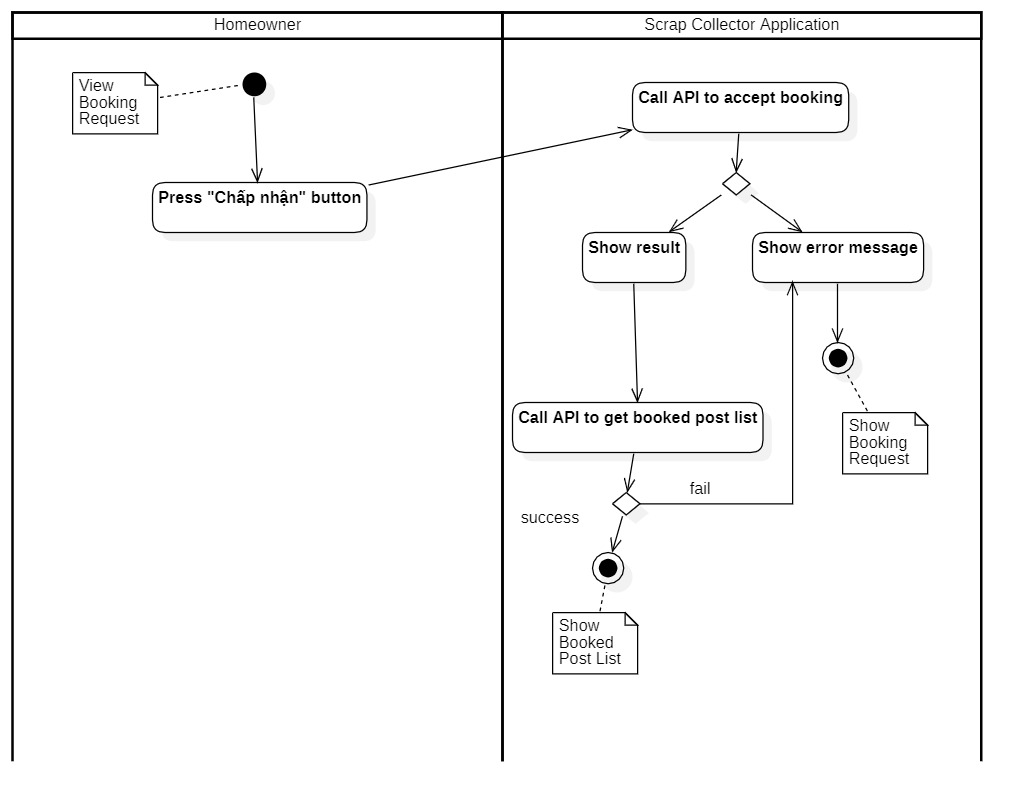


Figure - <Activity Diagram> Accept booking request

##### 4.3.6 Reject Booking request

Summary: This diagram shows how a Homeowner rejects a booking request.

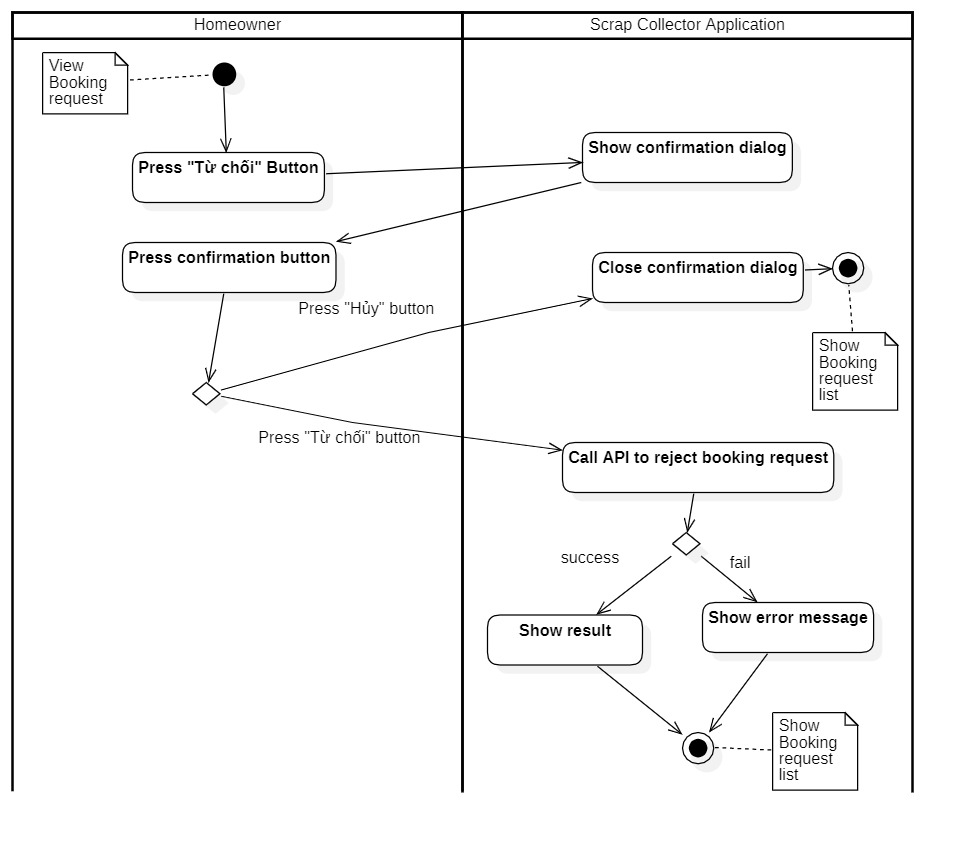


Figure - <Activity Diagram> Reject booking request

##### 4.3.7 Update Profile

Summary: This diagram shows how an user updates his/her profile.

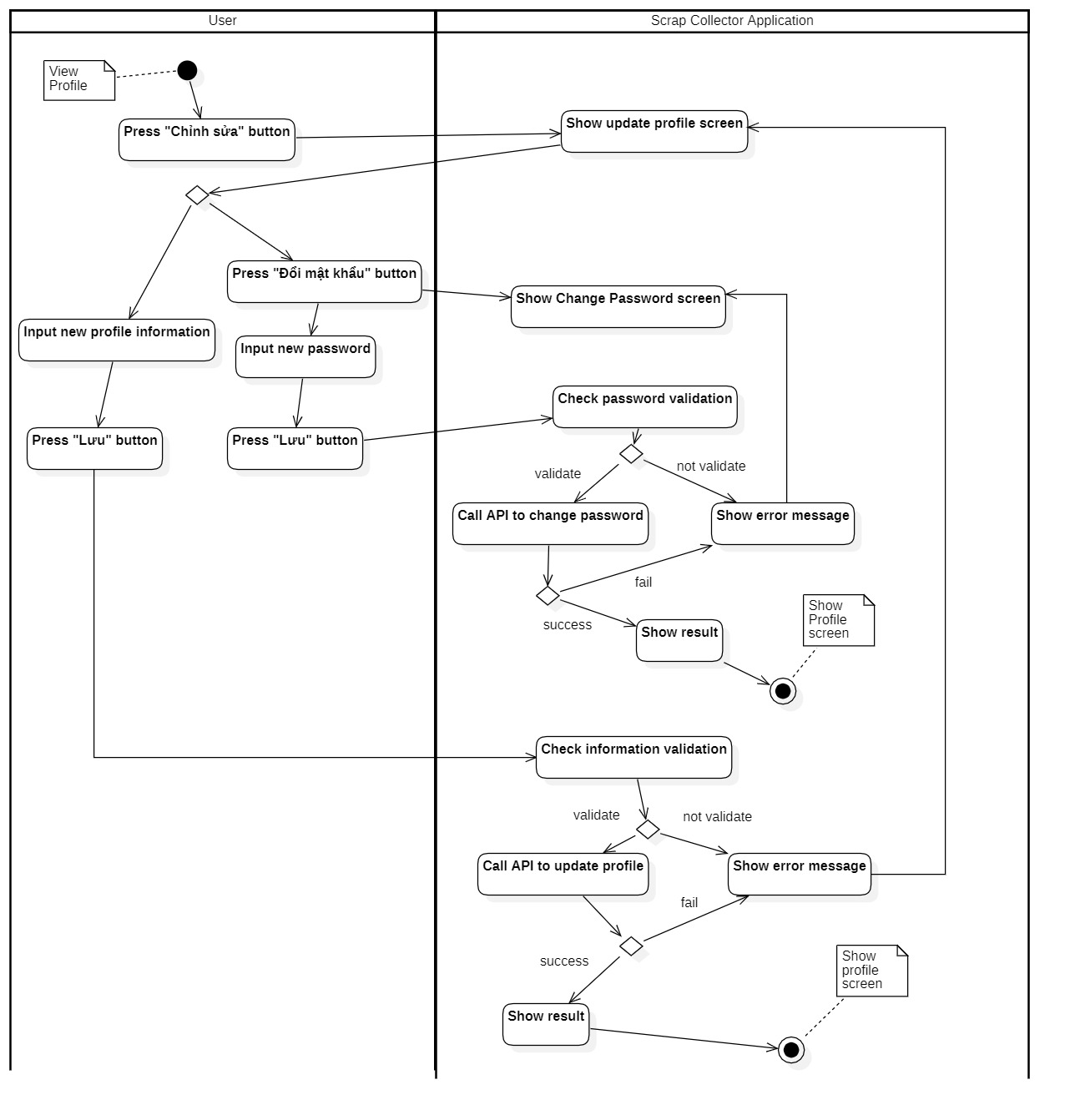


Figure - <Activity Diagram> Update profile

##### 4.3.8 Book Scrap

Summary: This diagram shows how a Collector books scrap.

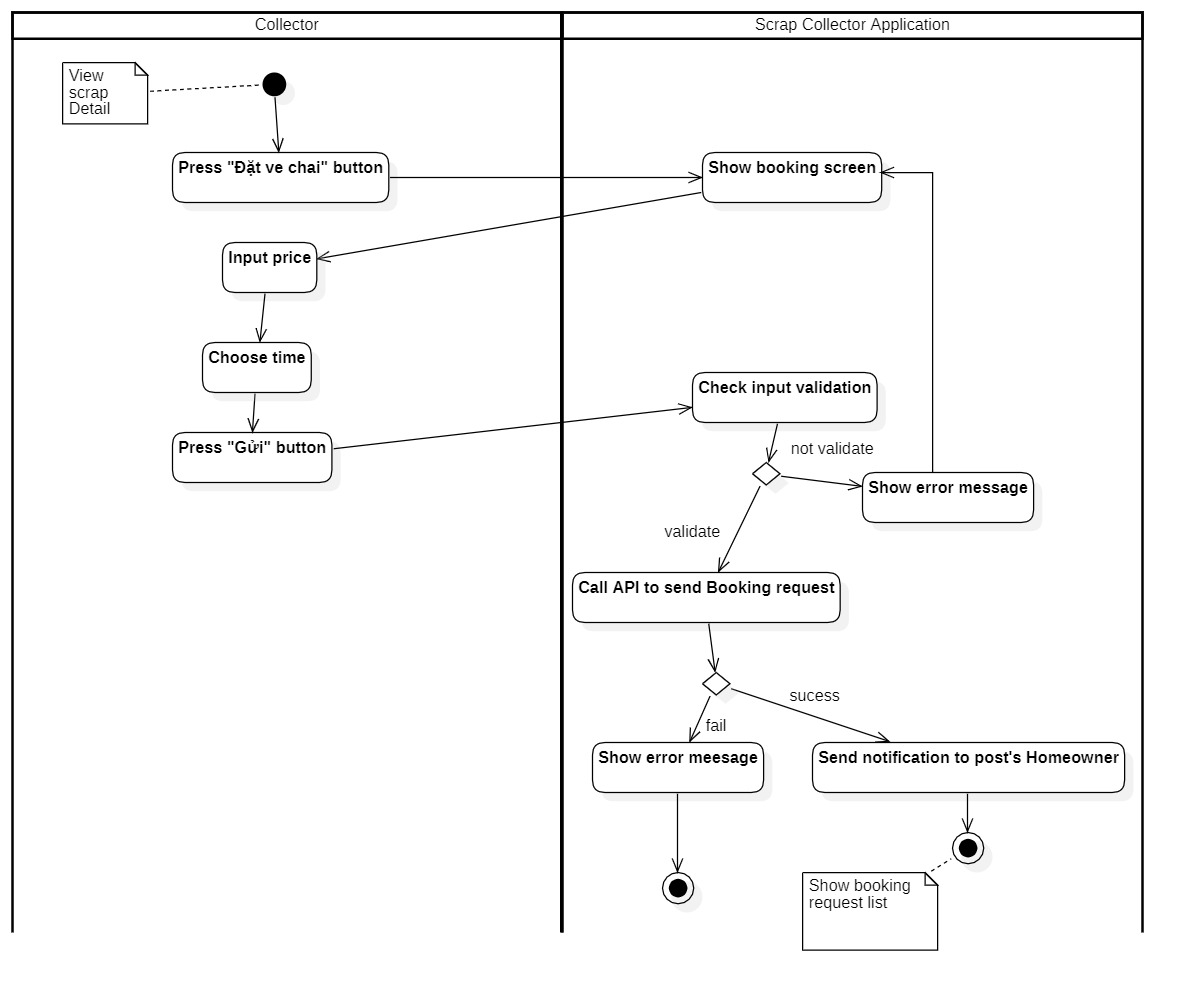


Figure - <Activity Diagram> Book scrap

##### 4.3.9 Search near-by scrap

Summary: This diagram shows how a Collector searchs for near-by scraps.

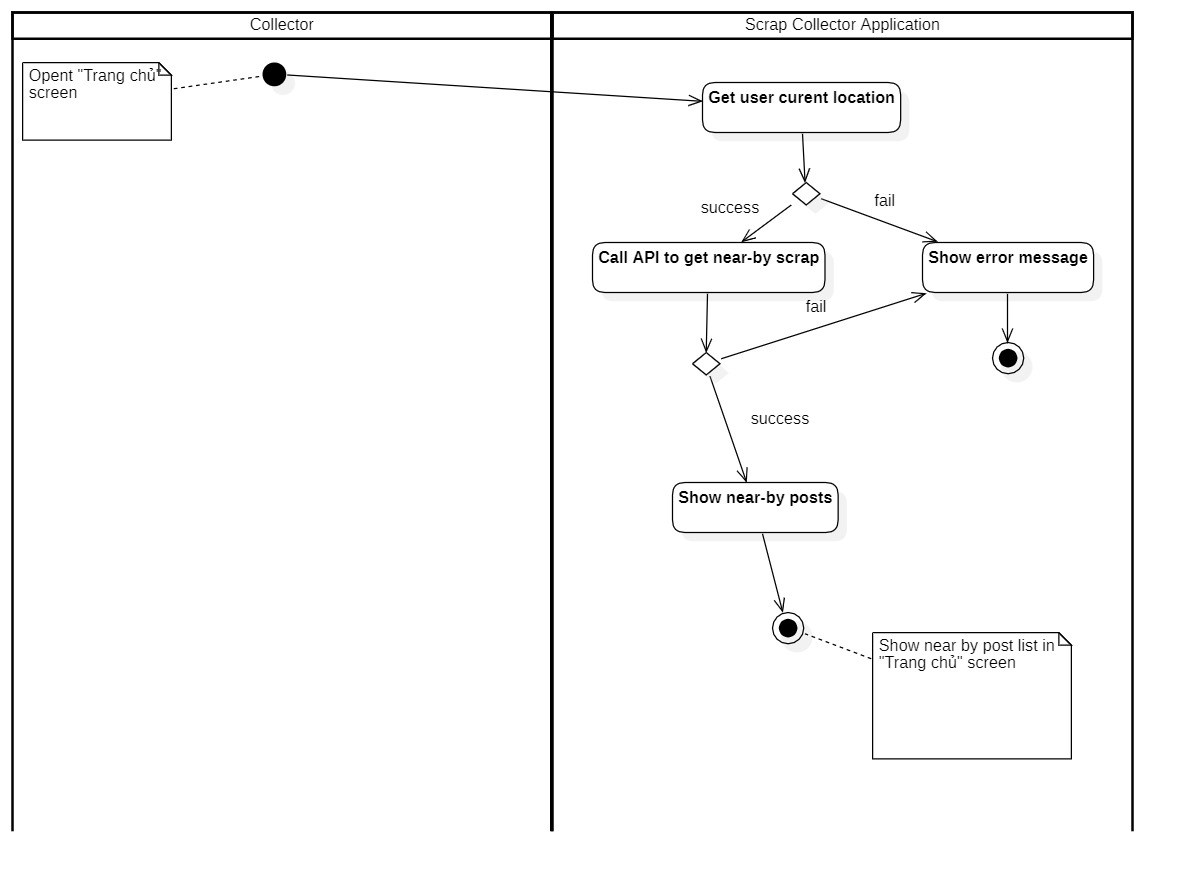


Figure - <Activity Diagram> Search near-by scrap

##### 4.3.10 View schedule

Summary: This diagram shows how a Collector views schedule to pick up scrap.

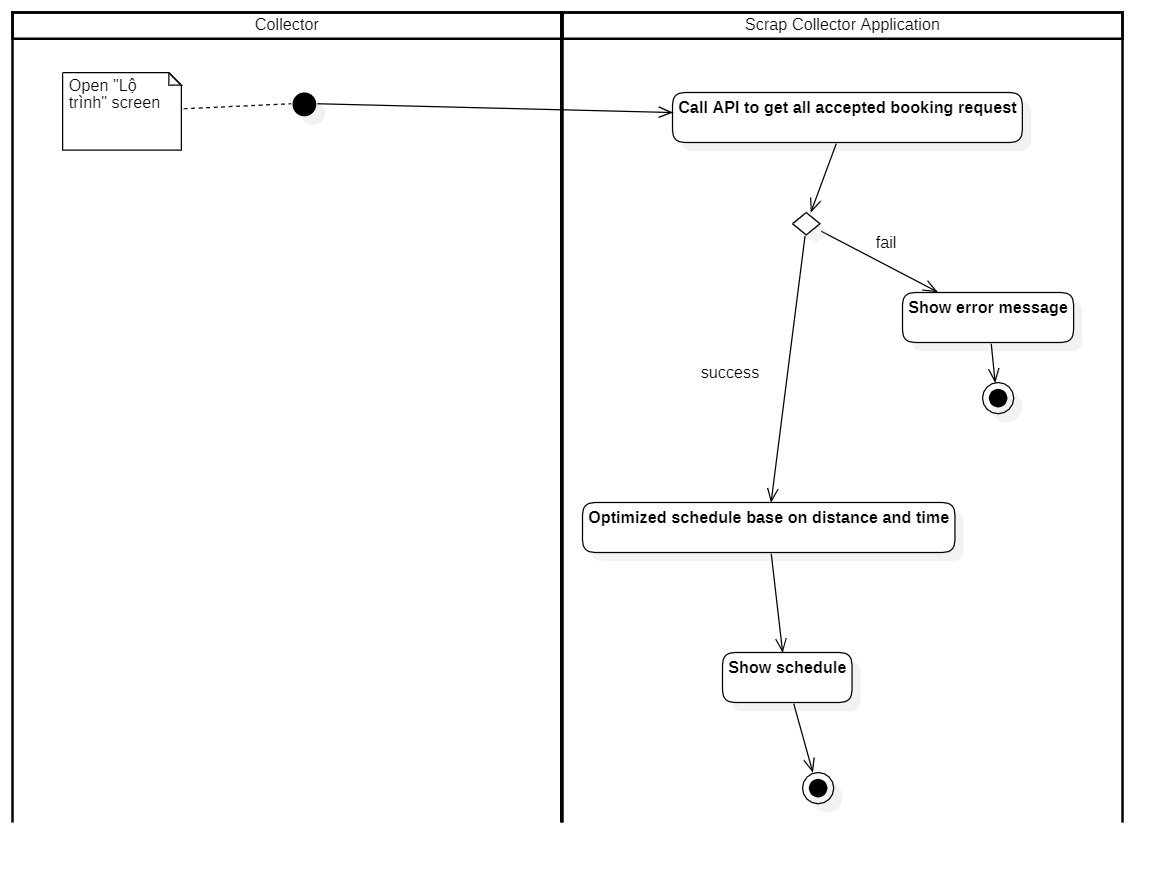


Figure - <Activity Diagram> View schedule

##### 4.3.11 Get Direction

Summary: This diagram shows how a Collector gets direction to the scrap’s location.

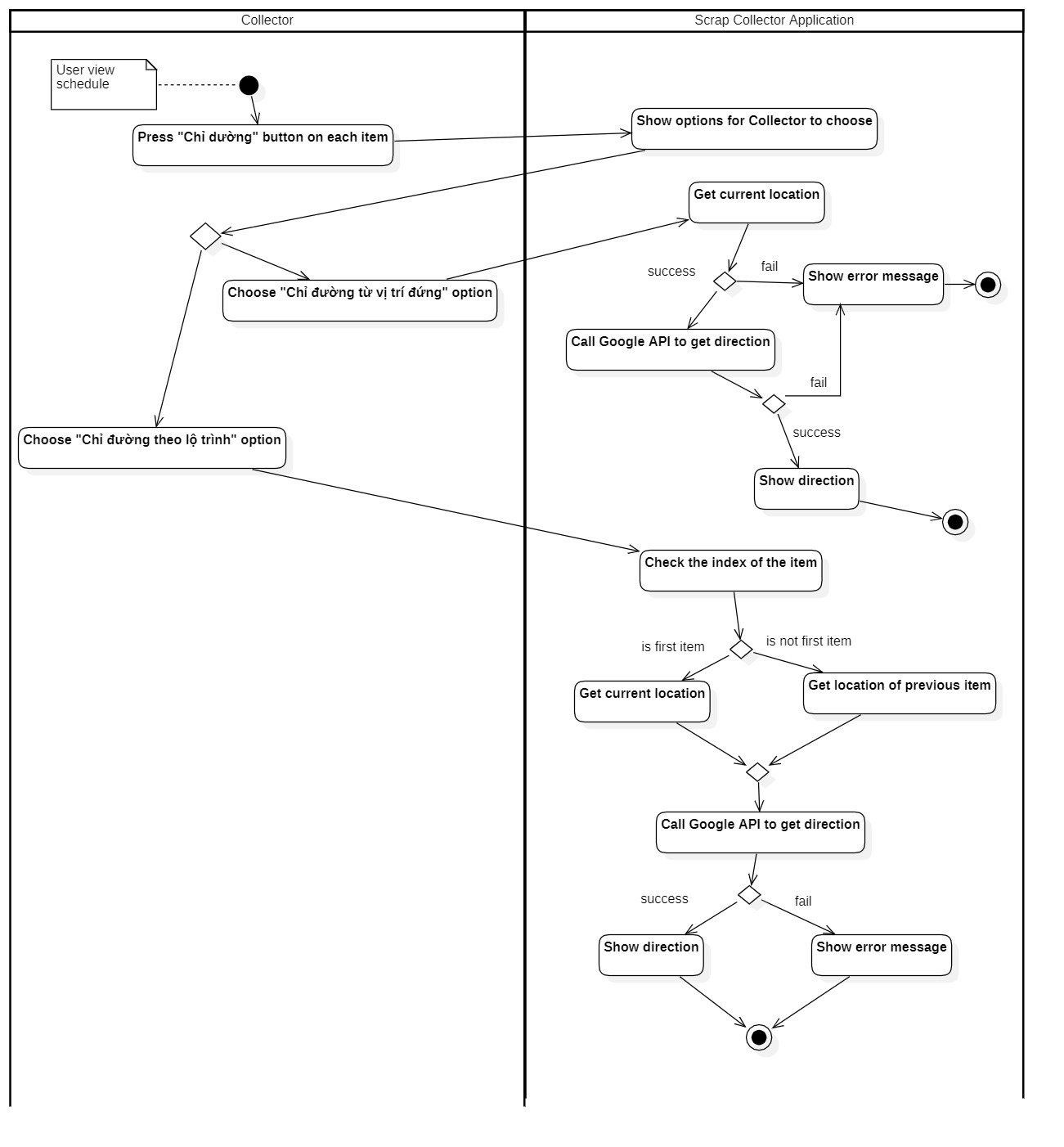


Figure - <Activity Diagram> Get Direction

##### 4.3.12 Review Collector

Summary: This diagram shows how a Homeowner reviews Collector.



Figure - <Activity Diagram> Review Collector

##### 4.3.13 View history post

Summary: This diagram shows how a user views history of post.



Figure - <Activity Diagram> View history post

##### 4.3.14 Logout

Summary: This diagram shows how a user logout.

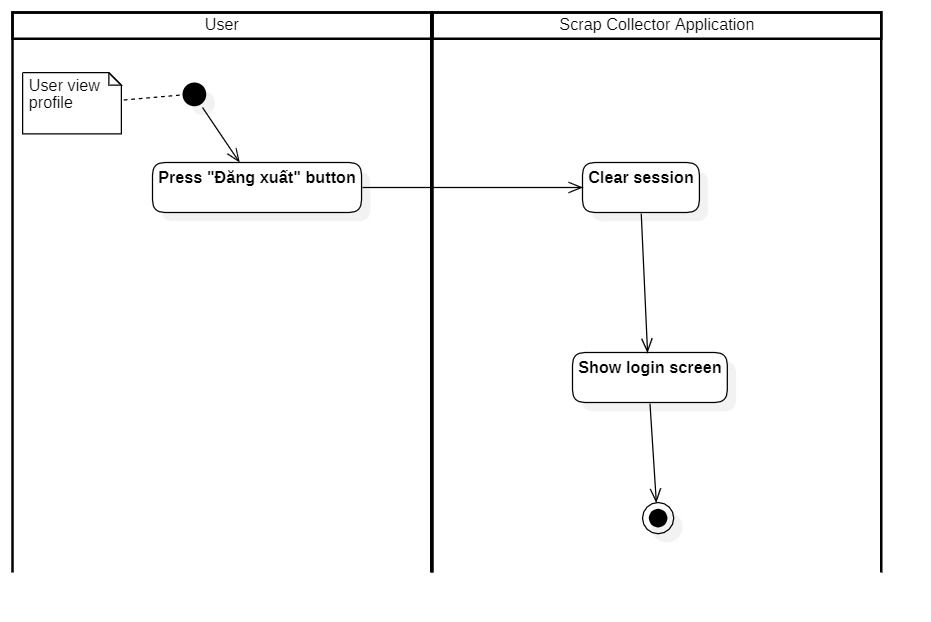


Figure - <Activity Diagram> Logout

## User Interface Design

#### 5.1 Mobile Application

##### 5.1.1 Login

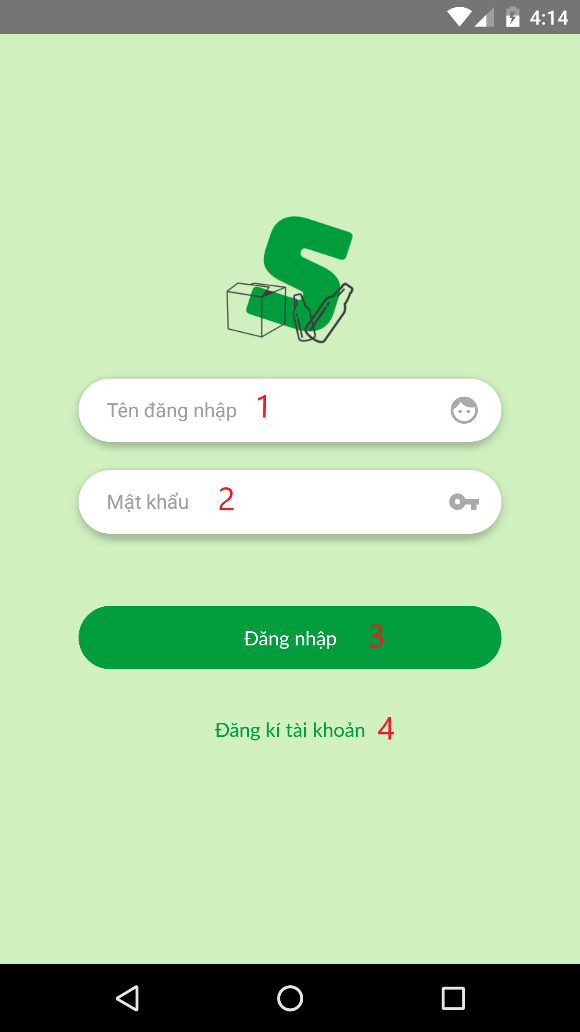


Figure 71 Login

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Tên đăng nhập | Username to login | No | Yes | TextInput | String |
| 2 | Mật khẩu | Password to login | No | Yes | TextInput | String |

Table 69 <Fields> Login

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Đăng nhập | Login into the system | Required | Transfer to Home screen |
| 4 | Đăng kí tài khoản | Register an account | No | Transfer to Register screen |

Table <Buttons/Hyperlinks> Login

##### 5.1.2 Register

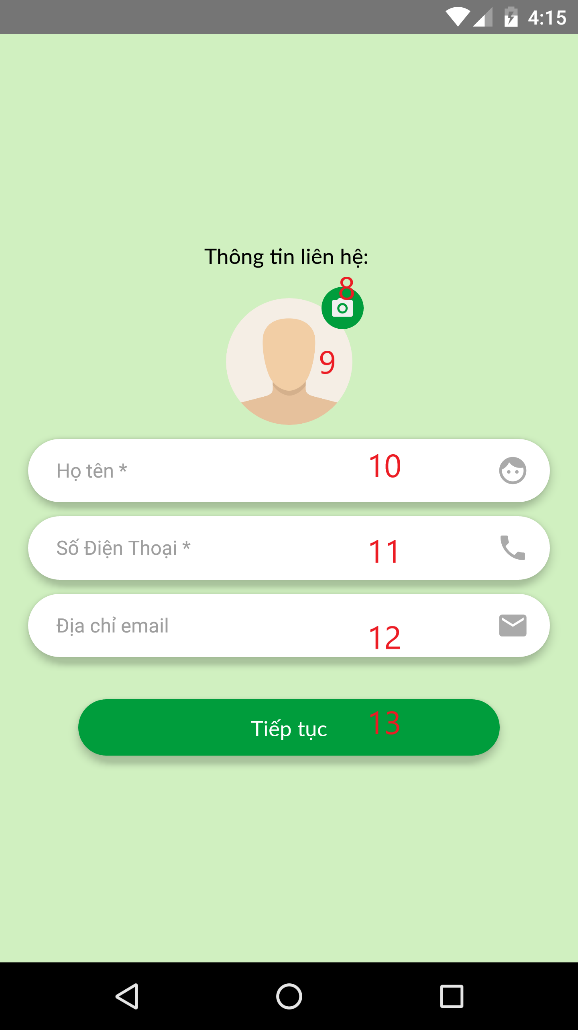
 

Figure 72 Register

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Tên đăng nhập | Username for regitration | No | Yes | TextInput | String |
| 2 | Mật khẩu | Password for registration | No | Yes | TextInput | String |
| 3 | Xác nhận mật khẩu | Confirm password for registration | No | Yes | TextInput | String |
| 9 | Ảnh đại diện | Avatar | No | No | Image | String |
| 10 | Họ tên | Full name for registration | No | Yes | TextInput | String |
| 11 | Số điện thoại | Phone number for registration | No | Yes | TextInput | String |
| 12 | Địa chỉ Email | Email for registration | No | No | TextInput | String |
| 14 | Địa chỉ chi tiết | Detail address for registration | No | Yes | TextInput | String |

Table 71 <Fields> Register

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 4 | Đăng kí | Register account with username and password | Required | Transfer to Choose role registration screen |
| 5 | Tôi đã có tài khoản | Direct user to go back to Login screen | N/A | Transfer to Login screen |
| 6 | Người thu mua ve chai | User register as a Collector | N/A | Transfer to Detail information registration screen |
| 7 | Người bán ve chai | User register as a Homeowner | N/A | Transfer to Detail information registration screen |
| 8 | Chọn ảnh | Take an image from device as avatar | N/A | Open Image picker |
| 13 | Tiếp tục | Go to next screen | Required | Transfer to Address registration screen |
| 15 | Chọn Tỉnh/ Thành phố | Choose province or city | N/A | Transfer to Choose address registration screen |
| 16 | Chọn Quận/ huyện | Choose district | Province or city is chosen | Transfer to Choose address registration screen |
| 17 | Chọn Phường/ xã | Choose ward | District is chosen | Transfer to Choose address registration screen |
| 18 | Đăng kí tài khoản | Finish registration | Required | Transfer to Login screen |

Table 72 <Buttons/Hyperlinks> Register

##### 5.1.3 View profile

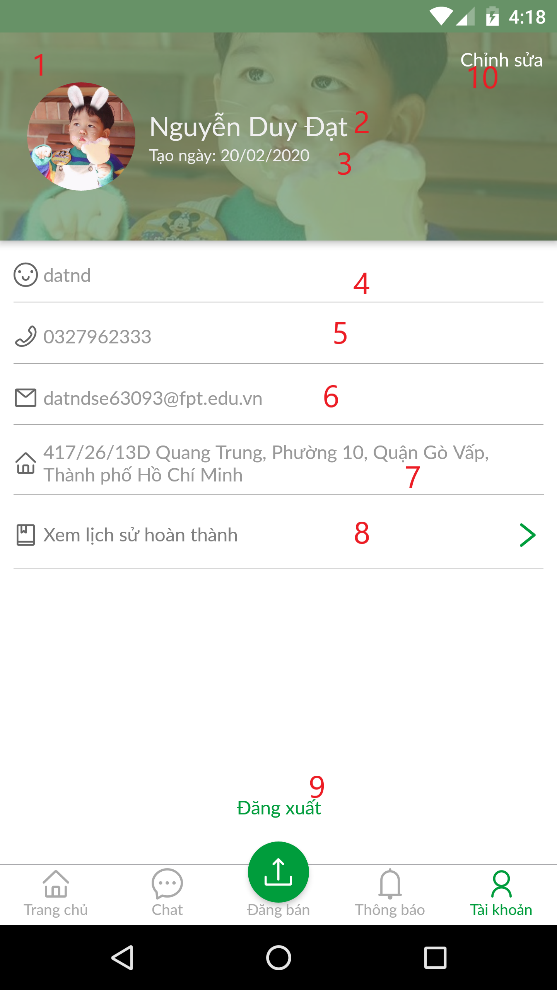


Figure 73 View profile

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Ảnh đại diện | Avatar of user | Yes | Yes | Text | String |
| 2 | Tên | Full name of user | Yes | Yes | Text | String |
| 3 | Ngày tạo | Date of creation of user | Yes | Yes | Text | String |
| 4 | Tên đăng nhập | Username of user | Yes | Yes | Text | String |
| 5 | Số điện thoại | Phone number of user | Yes | Yes | Text | String |
| 6 | Địa chỉ Email | Email Address of user | Yes | No | Text | String |
| 7 | Địa chỉ | Address of user | Yes | Yes | Text | String |

Table 73 <Fields> View profile

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Xem lịch sử hoàn thành | View all accomplished scrap of that user | N/A | Transfer to View history screen |
| 9 | Đăng xuất | Clear session of user and sign out | N/A | Transfer to Login screen |
| 10 | Chỉnh sửa | Update user’s profile | N/A | Transfer to Update profile screen |

Table 74 <Buttons/Hyperlinks> View profile

##### 5.1.4 Update profile

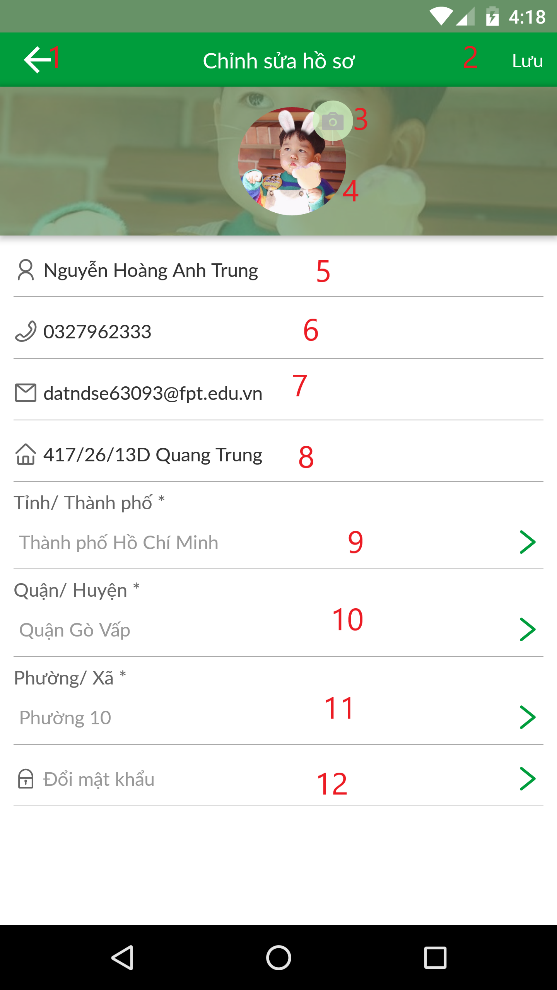


Figure 74 View profile

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 4 | Avatar | Avatar of user | No | No | Image | String |
| 5 | Full name | Full name of user | No | Yes | TextInput | String |
| 6 | Phone number | Phone number of user | No | Yes | TextInput | String |
| 7 | Email | Email address of user | No | No | TextInput | String |
| 8 | Detail address | Detail address of user | No | Yes | TextInput | String |

Table 75 <Fields> Update Profile

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Go back | Go back to Profile screen | N/A | Transfer to Profile screen |
| 2 | Lưu | Save changes and update profile | Required | Update profile and transfer to Profile screen |
| 3 | Choose image | Choose avatar of user | N/A | Change image in the image box |
| 9 | Chọn Tỉnh/ Thành phố | Choose Province or City | N/A | Change Province or City name in the button |
| 10 | Chọn Quận/ Huyện | Choose District | Province or City must be chosen | Change District name in the button |
| 11 | Chọn Phường/ Xã | Choose Ward | District must be chosen | Change Ward name in the button |
| 12 | Đổi mật khẩu | Go to Change password screen | N/A | Transfer to Change password screen |

Table 76 <Buttons/Hyperlinks> Update Profile

##### 5.1.5 Change password

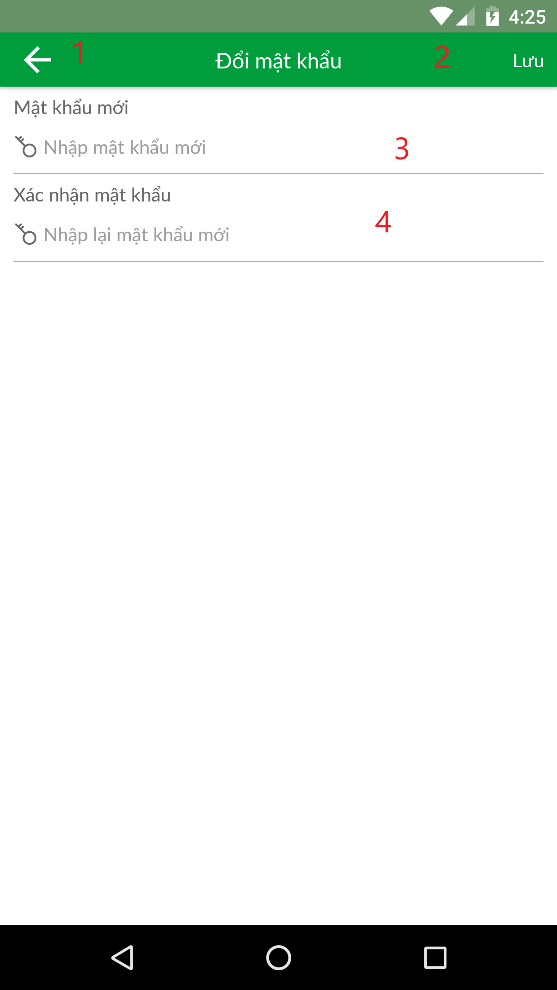


Figure 73 Change Password

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 3 | Mật khẩu mơi | New password to update | No | Yes | TextInput | String |
| 4 | Xác nhận mật khẩu | Confirm new password to update | No | Yes | TextInput | String |

Table 77 <Fields> Change Password

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Go Back | Go back to update profile screen | N/A | Transfer to Update profile screen |
| 2 | Lưu | Save new password | Required | Update password of user |

Table 78 <Buttons/Hyperlinks> Change Password

##### 5.1.6 Choose Address

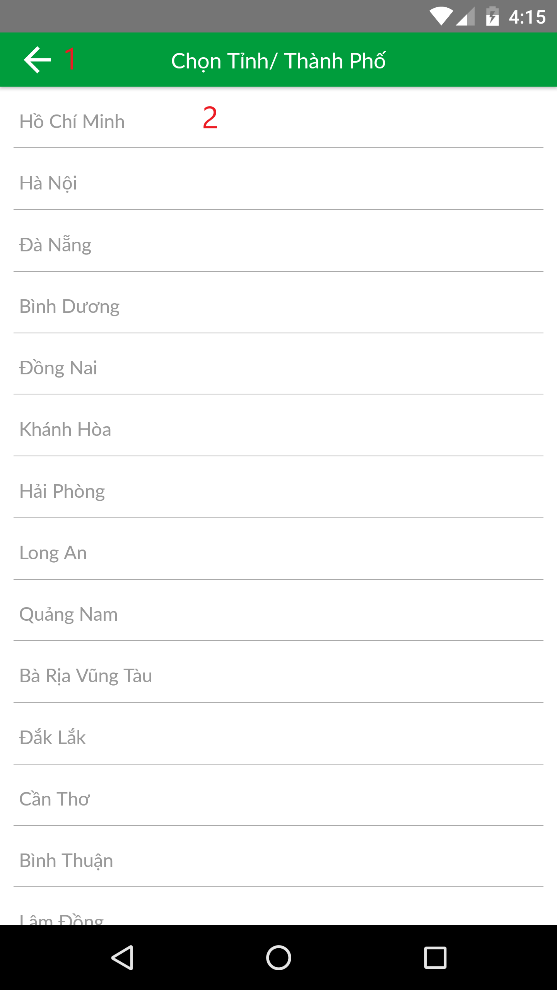
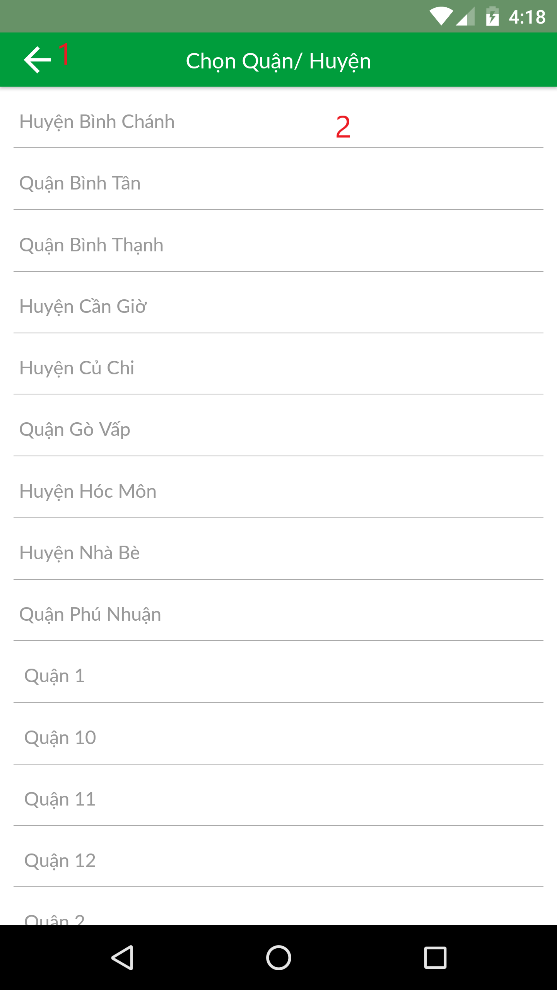
 

Figure 73 Choose Address

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Go back | Go back to previous screen | N/A | Transfer back to previous screen |
| 2 | Choose province/ district or ward | Choose province or district or ward | N/A | Save province/ district or ward and transfer back to previous screen |

Table 79 <Buttons/Hyperlinks> Choose Address

#### 5.2 Web Application

## Database Design

#### 6.1 Entity relationship diagram (ERD)

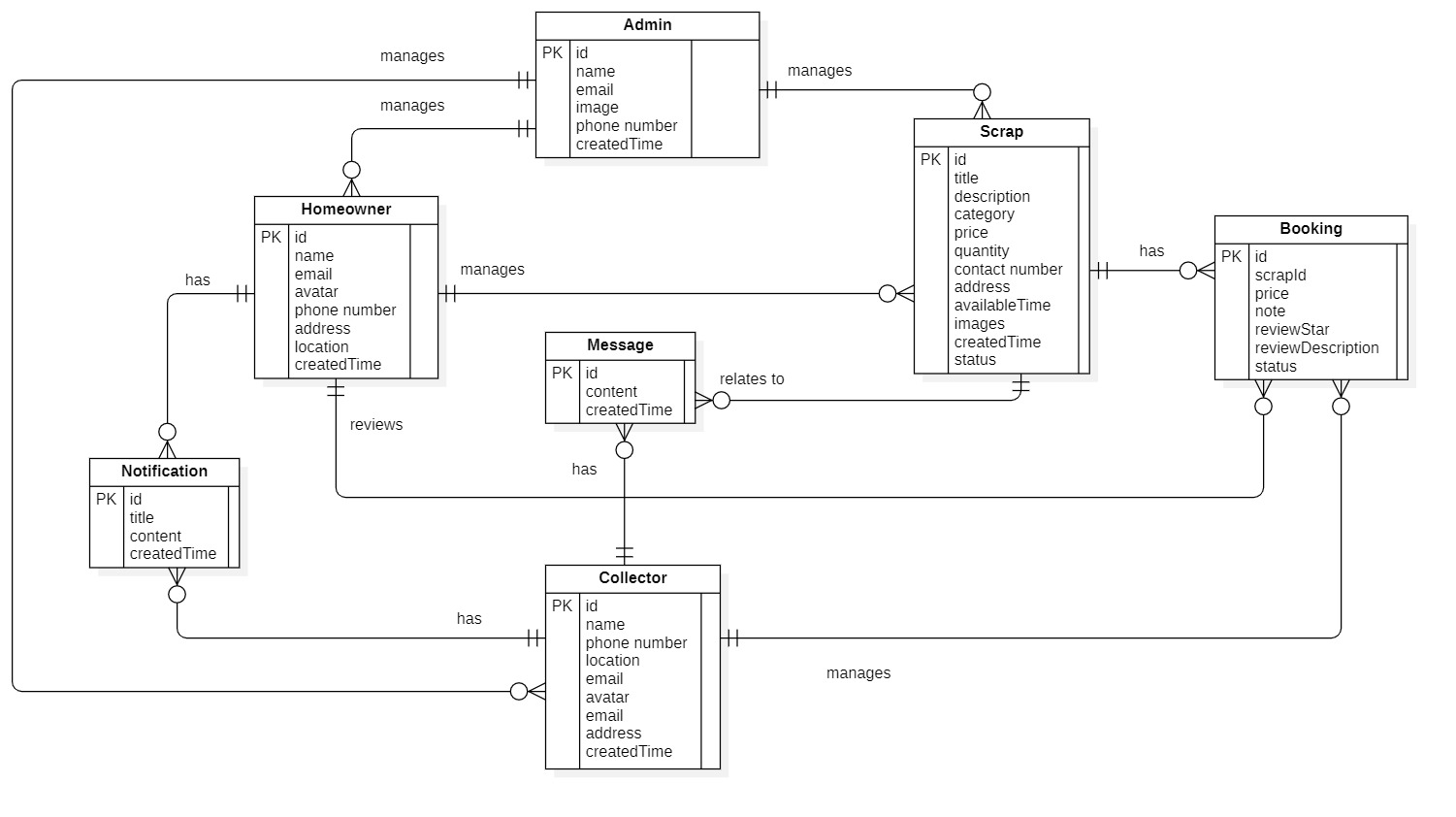


Figure 74 Entity Relationship Diagram

#### 6.2 Data Dictionary

|  |  |
| --- | --- |
| **Entity Data Dictionary: describe content of all entities** | |
| **Entity name** | **Description** |
| **Admin** | Contains admin’s information |
| **Homeowner** | Contains homeowner information |
| **Collector** | Contains collector information |
| **Scrap** | Contains scrap information |
| **Booking** | Contains booking information |
| **Notification** | Contains review’s information |
| **Message** | Contains message’s information |

Table - Entity Relationship Diagram Dictionary

## Algorithms

#### 7.1 Algorithms 1

##### 7.1.1 Defininition

##### 7.1.2 Define problem

##### 7.1.3 Solution

##### 7.1.4 Complexity

##### 7.1.5 Flowchart

# System Implementation & Test

## Introduction

* 1. **Overview**

This section describes approach and methodologies used by our team to plan, organize and manage testing process of WCS.

It provides all necessary information about implementation and testing procedure of the system including test plans, test cases, test results and pass/fail criteria as well as testing flow to cover all possible cases.

* 1. **Test Approach**
* **Goal:** Test main features in the whole WCS based on core flow
* **Method:** Black-box testing

## Data Relationship Diagram

Figure 75 Physical Diagram

|  |  |  |
| --- | --- | --- |
| **Physical Diagram Data Dictionary: Describe content of all tables** | | |
| **No.** | **Table Name** | **Description** |
| 1 | account | Contain account information |
| 2 | role | Contain role information |
| 3 | address | Contain address information |
| 4 | province | Contain province information of all provinces in Vietnam |
| 5 | district | Contain district information of all districts in Vietnam |
| 6 | ward | Contain ward information of all wards in Vietnam |
| 7 | device\_key | Contain key of the device which the user log in to |
| 8 | message | Contain message of user |
| 9 | category | Contain category information |
| 10 | scrap | Contain scrap information |
| 11 | image | Contain image information |
| 12 | available\_time | Contain available time information |
| 13 | booking | Contain booking information |
| 14 | review | Contain review information |

Table - Physical Diagram Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Null** |
| account | id | Unique id of each account | varchar(255) | N |
| username | Unique username of each account | varchar(50) | N |
| password | Password of account | varchar(50) | N |
| roleId | Id of the role of account | int | N |
| email | Email address of account | varchar(255) | Y |
| name | Full name of account | varchar(255) | N |
| phone | Phone number of account | varchar(15) | N |
| avatar | Link of avatar of account | varchar(255) | Y |
| addressId | Id of address of account | int | N |
| createdTime | Time of creation of account | datetime | N |
| isDeleted | Check if the account is deleted or not | boolean | N |
| role | id | Unique id of each role | int | N |
| name | Name of role | varchar(50) | N |
| isDeleted | Check if the role is deleted or not | boolean | N |
| address | id | Unique id of address | int | N |
| detailAddress | Detail address of address: home number, street name, building name | varchar(255) | N |
| provinceId | Id of provine of address | int | N |
| districtId | Id of district of address | int | N |
| wardId | Id of ward of address | int | N |
| latitute | Latitute of address | double | N |
| longitute | Longitute of address | double | N |
| province | id | Unique id of province | int | N |
| code | Code of province | varchar(20) | N |
| name | Name of province | varchar(50) | N |
| district | id | Unique id of district | int | N |
| provinceId | Id of province | int | N |
| prefix | Prefix name of district | varchar(50) | N |
| name | Name of district | varchar(50) | N |
| ward | id | Unique id of ward | int | N |
| districtId | Id of district | int | N |
| provinceId | Id of province | int | N |
| prefix | Prefix name of ward | int | N |
| name | Name of ward | int | N |
| device\_key | id | Unique id of device key | int | N |
| accountId | Id of account | varchar(255) | N |
| key | Key of device | varchar(255) | N |
| isDeleted | Check if the device key is deleted or not | boolean | N |
| message | id | Unique id of message | varchar(255) | N |
| collectorId | Id of collector | varchar(255) | N |
| scrapId | Id of scrap belonged to the message | varchar(255) | N |
| content | Content of message | varchar(255) | N |
| createdTime | Time of creation of message | datetime | N |
| isDeleted | Check if the message is deleted or not | boolean | N |
| category | id | Unique id of category | int | N |
| name | Name of category | varchar(255) | N |
| unit | Unit of category | varchar(255) | N |
| isDeleted | Check if the category is deleted or not | boolean | N |
| scrap | id | Unique id of scrap | varchar(255) | N |
| title | Title of scrap | varchar(255) | N |
| description | Description of scrap | varchar(255) | N |
| price | Price of scrap | double | N |
| quantity | Quanity of scrap | double | N |
| categoryId | Id of category | int | N |
| homeownerId | Id of the homeowner posting scrap | varchar(255) | N |
| contactNumber | Phone number | varchar(20) | N |
| addressId | Id of address | int | N |
| status | Status of scrap | varchar(20) | N |
| createdTime | Time of posting scrap | datetime | N |
| isDeleted | Check if scrap is deleted or not | boolean | N |
| image | id | Unique id of each image | int | N |
| scrapId | Id of scrap | varchar(255) | N |
| imageUrl | Url link of image | varchar(255) | N |
| isDeleted | Check if the image is deleted or not | boolean | N |
| available\_time | id | Unique id of available time | int | N |
| scrapId | Id of scrap | varchar(255) | N |
| fromTime | From time | datetime | N |
| toTime | To time | datetime | N |
| isDeleted | Check if the available time is deleted or not | boolean | N |
| booking | id | Unique id of booking | int | N |
| fromCollector | Id of Collector | varchar(255) | N |
| scrapId | Id of scrap | varchar(255) | N |
| price | Price of booking | double | N |
| note | Note of booking | varchar(255) | N |
| status | Status of booking | varchar(20) | N |
| createdTime | Time of creating booking | datetime | N |
| isDeleted | Check if the booking is deleted or not | boolean | N |
| review | id | Unique id of each review | int | N |
| fromHomeowner | Id of Homeowner | varchar(255) | N |
| bookingId | Id of booking | varchar(255) | N |
| star | Number of star | int | N |
| description | Description of review | varchar(255) | Y |
| isDeleted | Check if the review is deleted or not | boolean | N |

Table - Physical Diagram Attribute Data Dictionary

## Test Plan

The overall purpose of testing is to ensure WCS meets its entire technical, functional and business requirement. The purpose of this document is to describe the test plan and strategy for testing the WCS. The following part will describe which features to be tested and which will not.

## Features To Be Tested

## Features Not To Be Tested

## System Testing Test Case

***4.1 Staff Test Case  
 4.1.1 Manage page option***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Table <Test case> Manage Page Option

***4.1.2 Broken links test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Table <Test case> Broken Links Test

***4.1.3 Broken pages test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Table <Test case> Broken Pages Test

***4.1.4 Prohibited content test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Table <Test case> Prohibited Content Test

***4.1.5 Mobile layout test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Table <Test case> Mobile Layout Test

***4.1.6 Speed test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *ST\_1* | Staff send speed test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Page”, “Interactive time”, “Load time”, “Size” of root page. | N/A | Pass | 26/11/2018 |  |
| *ST\_2* | Staff send speed test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Page”, “Interactive time”, “Load time”, “Size” of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Speed Test

***4.1.7 JavaScript test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *JST\_1* | Staff send JavaScript test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Error message”, “Type”, “Page” of root page. | N/A | Pass | 26/11/2018 |  |
| *JST\_2* | Staff send JavaScript test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Error message”, “Type”, “Page” of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Javascript Errors Test

***4.1.8 Cookie law test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *CLT\_1* | Staff send cookie law test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Name”, “Category”, “Party” and “Description” of root page. | N/A | Pass | 26/11/2018 |  |
| *CLT\_2* | Staff send cookie law test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Name”, “Category”, “Party”, “Description” of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Cookie Law Test

***4.1.9 Server behavior test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *SBT\_1* | Staff send server behavior test command with page option unselected. | 1. Staff click “Check” button. | System return test result of root page. | N/A | Pass | 26/11/2018 |  |
| *SBT\_2* | Staff send server behavior test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Server Behavior Test

***4.1.10 Missing File test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *MFT\_1* | Staff send missing file test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Files”, “Description”, “Pages” and “Action” of root page. | N/A | Pass | 26/11/2018 |  |
| *MFT\_2* | Staff send missing file test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Files”, “Description”, “Pages” and “Action” of all page in page option. | N/A | Pass | 26/11/2018 |  |
| *MFT\_3* | Staff send missing file test command with page option unselected and select specific type to test. | 1. Staff select type to test.  2. Staff click “Check” button. | System return test result with information: “Files”, “Description”, “Pages” and “Action” of root page and type selected. | N/A | Pass | 26/11/2018 |  |
| *MFT\_3* | Staff send missing file test command with page option selected and select specific type to test. | 1. Staff select page option.  2. Staff select type to test.  3. Staff click “Check” button | System return test result with information: “Files”, “Description”, “Pages” and “Action” of all page in page option and type selected. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Missing File Test

***4.1.11 Favicons test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *CLT\_1* | Staff send favicons test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Favicon”, “URL”, “Width”, “Type” and “Pages” of root page. | N/A | Pass | 26/11/2018 |  |
| *CLT\_2* | Staff send favicons test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Favicon”, “URL”, “Width”, “Type” and “Pages” of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Favicon Test

***4.1.12 Contact Details test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *CDT\_1* | Staff send contact details test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Contact”, “Page affected” and “Action” of root page. | N/A | Pass | 26/11/2018 |  |
| *CDT\_2* | Staff send contact details test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Contact”, “Page affected” and “Action” of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Contact Details Test

***4.1.14 Redirection test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *RET\_1* | Staff send redirection test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Web Address”, “Directs to”, “Type” and “Code” of root page. | N/A | Pass | 26/11/2018 |  |
| *RET\_2* | Staff send redirection test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Web Address”, “Directs to”, “Type” and “Code” of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Redirection Test

***4.1.14 Pages test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *CDT\_1* | Staff send pages test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Title”, “Web Address”, “Canonical URL” and “HTTP” of root page. | N/A | Pass | 26/11/2018 |  |
| *CDT\_2* | Staff send pages test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Title”, “Web Address”, “Canonical URL” and “HTTP” of all page in page option. | N/A | Pass | 26/11/2018 |  |

Table <Test case> Pages Test

***4.1.15 Spelling test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *SPT\_1* | Staff send spelling test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Word”, “Suggestions”, “Found in”, “Page” and “Action” of root page. | N/A |  |  |  |
| *SPT\_2* | Staff send spelling test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Word”, “Suggestions”, “Found in”, “Page” and “Action” of all page in page option. | N/A |  |  |  |

Table 98 <Test case> Spelling Test

***4.1.16 Grammar test***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | *Inter Dependence -test Case* | Result | Test Date | Note |
| *GRT\_1* | Staff send grammar test command with page option unselected. | 1. Staff click “Check” button. | System return test result with information: “Excerpt”, “Issue”, and “Action” of root page. | N/A |  |  |  |
| *GRT\_2* | Staff send grammar test command with page option selected. | 1. Staff select page option.  2. Staff click “Check” button. | System return test result with information: “Excerpt”, “Issue”, and “Action” of all page in page option. | N/A |  |  |  |

Table 99 <Test case> Grammar Test

# Software User’s Manual

1. **Installation Guide**
   1. **Setting up enviroment at server side**
      1. **Hardware requirements**

|  |  |  |
| --- | --- | --- |
| **Server** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (8 Mbps) | Cable, Wi-Fi (50 Mbps or more) |
| **Operating System** | XP, Vista, 7, 10, Window Server 2008 | 10, Window Server 2008 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| **Computer Memory** | 4GB RAM | 32GB or more |
| **Storage space** | 1GB | 5GB or more |

Table 100 - Hardware requirement for server

* + 1. **Software requirements**

|  |  |  |
| --- | --- | --- |
| **Name** | **Name/Version** | **Description** |
| **Environment** | Java EE 8.0,Node v10,npm v6 | Specification for developing web application |
| **Operating System** | Window 10 | Operating system and platform for development |
| **Modeling tool** | StarUML | Used to design diagram |
| **IDE** | IntelliJ IDEA 2018.1.5, Visual Studio Code 1.27.2 | Programming tools |
| **DBMS** | MySQL 8.0 | Used to create & manage the database for system |
| **Web server** | Apache Tomcat 8 | Deployment environment |
| **Web browser** | Chrome 69 or above. | Testing browser |

Table 101 - Software requirement for server

* 1. **Deployment at server side**

**1.2.1 Prepare deployement package**

**Step 1:** Download and Install Java 8

Link download: <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html>

**Step 2:** Download and setup Tomcat 8

Link download: <https://tomcat.apache.org/download-80.cgi>

**Step 3:** Download and Install Node v10

Link download: https://nodejs.org/en/download/

**Step 4:** Download and Install IntelliJ IDE

Link download: <https://www.jetbrains.com/idea/download/#section=windows>

**Step 5:** Download, Install and configure MySQL 8.0

Link download: <https://dev.mysql.com/downloads/installer/>

**1.2.2 Configure server before deploy**

Step 1: Go to %Tomcat folder%\conf\

Step 2: Edit port to 8080 in server.xml at

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

Step 3: Delete all files and folders in %Tomcat folder%\webapps

Step 4: Rename production java build to ROOT.WAR and copy to

%Tomcat folder%\webapps.

* 1. **Setting up enviroment at client side**

The client devices should have one of the following browsers to access the website:

* + - Google Chrome

Link download: <https://www.google.com/chrome/>

* + - Firefox

Link download: <https://www.mozilla.org/vi/firefox/channel/desktop/>

**2. User Guide**

Appendix

* + - 1. **Spring Framework** *[Online]* Available: [https:/spring.io/docs/reference](https://spring.io/docs/reference).
      2. **The Unified Certification** *[Online]* Available: <https://www.omg.org/ocup-2/index.htm>.
      3. **MySQL** *[Online]* Available: <https://dev.mysql.com/doc/>.
      4. **ReactJS** *[Online]* Available: <https://reactjs.org/docs/getting-started.html>
      5. **IntelliJ IDEA** *[Online]* Available: <https://www.jetbrains.com/idea/documentation/>
      6. **Visual Studio Code** *[Online]* Available: <https://code.visualstudio.com/docs>
      7. **Jsoup** *[Online]* Available: <https://jsoup.org/apidocs/overview-summary.html>
      8. **Selenium** *[Online]* Available: https://www.seleniumhq.org/docs/
      9. **Nodejs** *[Online]* Available: <https://nodejs.org/en/docs/>
      10. **HtppCode** *[Online]* Available: <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
      11. **Regular Expression** *[Online]* Available: <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>
      12. **Cookie Law** *[Online]* Available: <https://www.cookielaw.org/faq/>
      13. **DFS and BFS Algorithms using Stacks and Queues, Professor Lawrence L. Larmore** *[Online]* Available: <http://web.cs.unlv.edu/larmore/Courses/CSC477/bfsDfs.pdf>
      14. **The Breadth-First Search Algorithm, Khan Academy** *[Online]* Available: <https://www.khanacademy.org/computing/computer-science/algorithms/breadth-first-search/a/the-breadth-first-search-algorithm>
      15. **Data Structure — Breadth First Traversal, TutorialsPoint** *[Online]* Available: <https://www.tutorialspoint.com/data_structures_algorithms/breadth_first_traversal.htm>
      16. **Binary tree: Level Order Traversal, mycodeschool** *[Online]* Available: <https://www.youtube.com/watch?v=86g8jAQug04>
      17. **Breadth-First Traversal of a Tree, Computer Science Department of Boston University** *[Online]* Available: <https://www.cs.bu.edu/teaching/c/tree/breadth-first/>
      18. **Digit-based sorting and data structures, Professor Avrim Blum** *[Online]* Available: <https://www.cs.cmu.edu/~avrim/451f11/recitations/rec0921.pdf>
      19. **Lecture Notes on Tries, Professor Frank Pfenning** *[Online]* Available: <https://www.cs.cmu.edu/~fp/courses/15122-f10/lectures/18-tries.pdf>
      20. **Algorithms: Tries, Robert Sedgewick and Kevin Wayne** *[Online]* Available: <https://algs4.cs.princeton.edu/lectures/52Tries.pdf>
      21. **Tries, Brilliant Learning** *[Online]* Available: <https://brilliant.org/wiki/tries/>
      22. **Tries, Daniel Ellard** *[Online]* Available: <http://ellard.org/dan/www/libsq/cb_1998/c06.pdf>
      23. **Tries, Harvard CS50** *[Online]* Available: <https://www.youtube.com/watch?v=TRg9DQFu0kU>
      24. **Sitemap, Garenne Bigby** *[Online]* Available: https://dynomapper.com/blog/16-sitemap-generator/164-top-50-sitemap-generators-for-creating-sitemaps