



MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document – Shorten Version

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

<i>A. Capstone Project Register</i>	3
<i>B. Report 1 - Introduction</i>	5
<i>C. Software Process Model</i>	8
<i>D. Conceptual Diagram</i>	10
<i>E. Use-case Diagram</i>	11
<i>F. Use-case specification</i>	12
1. Homeowner Use Case	12
2. Collector Use Case	19
3. Admin Use Case.....	25
<i>G. Architecture Diagram</i>	31
<i>H. Component Diagram</i>	32
<i>I. Class Diagram</i>	35
<i>J. Entity Relationship Diagram</i>	37
<i>K. Interactive Diagram</i>	39
<i>L. Physical Diagram – Relationship Diagram</i>	46
<i>M. Framework Architecture Diagram</i>	52
1. React Redux Framework Architecture	52
2. Angular Framework Architecture	52
<i>N. Algorithm</i>	53
1 Search nearby Scrap implementing Geohash (Proximity Searching).....	53
2. Best price recommendation by Linear Regression	57
<i>O. Future plan</i>	61

A. Capstone Project Register

CAPSTONE PROJECT REGISTER

Class: Duration time: From To

(*) Profession: <Software Engineer> Specialty: <JS>

(*) Kinds of person make registers: Lecturer Students

1. Register information for supervisor (if have)

	Full name	Phone	E-Mail	Title
Supervisor 1	Lại Đức Hùng		HungLD5@fe.edu.vn	Mr.

2. Register information for students (if have)

	Full name	Student code	Phone	E-mail	Role in Group
Student 1					
Student 2					
Student 3					
Student 4					

3. Register content of Capstone Project

(*) 3.1 Capstone Project name:

English: Scrap Collector

Vietnamese: Ứng dụng hỗ trợ thu mua ve chai.

Abbreviation:

This application helps scrap collectors to find and buy scraps easily and effectively. It will connect customers and scrap collectors. By locating local area scraps, the collectors can view scrap information, contact details.... With this application, the customers can easily request a scrap pickup.

(*) 3.2 Main proposal content (including result and product)

a) Theory and practice (document):

- Student should apply the software development process and the UML
- Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages

- 3 tiers should be applied

<https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/>

b) Program:

- Main functions:

- The collector
 - Find nearby scraps
 - View scrap details
 - Negotiate price
 - Schedule a pickup
 - Contact the customer
 - ...
- The customers
 - Post scraps
 - Negotiate price
 - Contact the collectors
 - Request a scrap pickup
 - Review the collector
 -
- Admin
 - Manage user
 - Manage category
-

c) Other products:

4. Other comment (propose all relative thing if have)

Ho Chi Minh city, 24/12/2019

Supervisor (If have)

(Sign and full name)

On behalf of Registers

(Sign and full name)

B. Report 1 - Introduction

1. Project Information

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

2. Introduction

In this document, we introduce a new solution for those who are doing the job as collecting scraps. Scrap are unusable materials such as iron, aluminum, plastic bottles, cans, etc., which can be recycled. The current job of the scrap collectors is to go around and ask their neighborhood if anybody has scrap to sell. That traditional way is inconvenient for both the collector and the homeowner.

Based on our researches and analysis, we offer a solution for scrap collectors and people who want to sell scrap in Vietnam. We build a mobile application, which helps the scrap collectors to find near-by scraps based on their location, create scrap bookings and contact to the seller via calling or chatting. The mobile application also helps the collectors to set up schedule to collect all the booked scrap.

Beside of that, our mobile application helps homeowner who wants to sell scrap to post scrap and contact the collector easily.

3. Current Situation

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

4. 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 booked with the customer.
- The collector cannot optimize his/her routes.

- The homeowner finds it difficult to find a scrap collector so he/she often throws scrap as garbage.
- The homeowner has to wait for the scrap collector to pass by his/her house and call the collector.

5. 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
 - Not all scrap collectors have smart phone to take advantage of this application

6. Functional Requirements

Function requirements of the system are listed as below:

6.1 Admin:

- Admin can view all accounts in the system.
- Admin can view all scraps posted by homeowner.
- Admin can view all categories in the system
- Admin can add new category, update category, deactivate category and activate category
- Admin can view all report about scrap or user

6.2 Collector:

- Collector can find near-by scrap.
- 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 view review from Homeowner.
- Collector can update profile.

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 profile.

7. 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 1 - Role & Responsibility

C. 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 this 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.

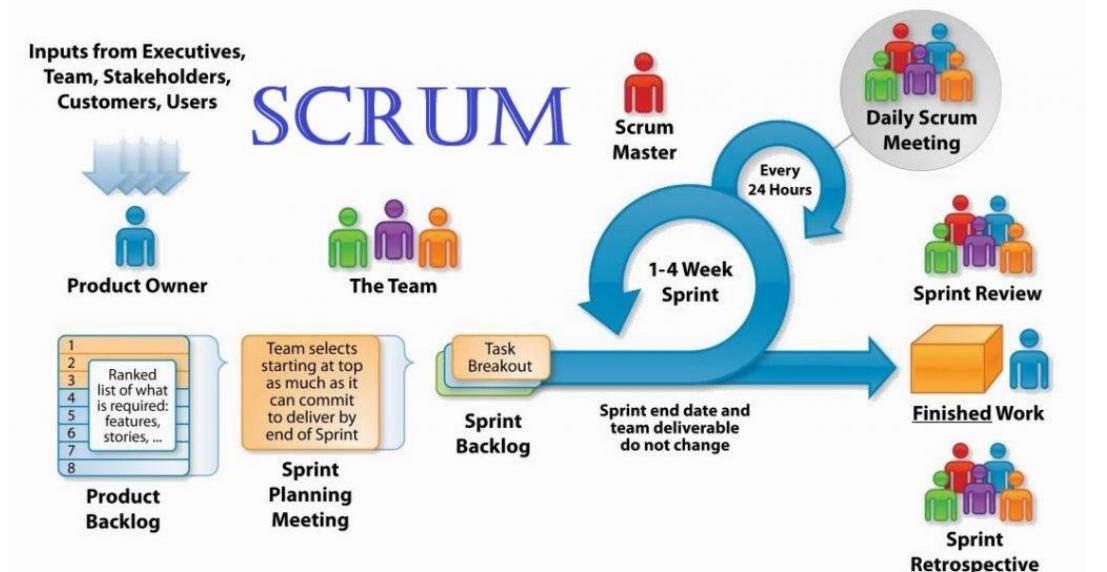


Figure 1– Software process model

Reference: <https://www.apeironsoftware.com/scrum-roles-artifacts-and-ceremonies/>

Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Lại Đức Hùng	Product Owner	<ul style="list-style-type: none"> - Specify user requirement - Control the development process - Give out technique and business analysis support

2	Nguyễn Duy Đạt	Scrum Master	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing
4	Nguyễn Hoàng Anh Trung	Scrum team member	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing
5	Nguyễn Hoàng Nhân	Scrum team member	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing

Table 2 - Roles & Responsibilities Details

D. Conceptual Diagram

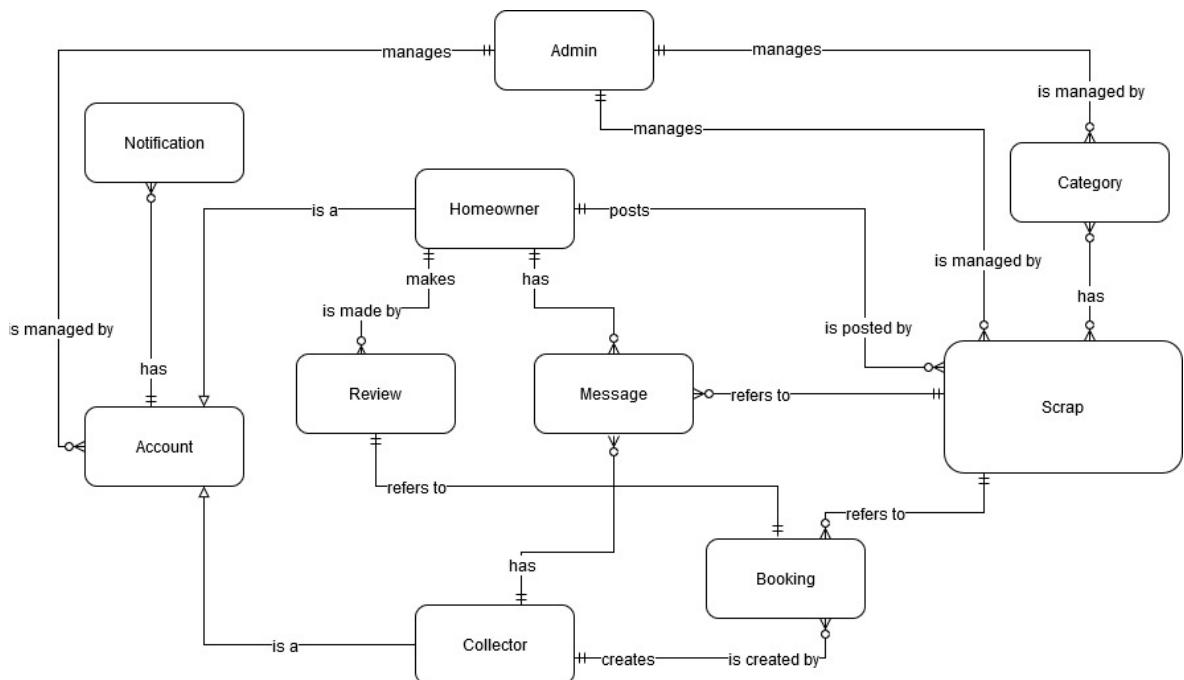


Figure 2 - Conceptual Diagram

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 3 - Conceptual Diagram Data Dictionary

E. Use-case Diagram

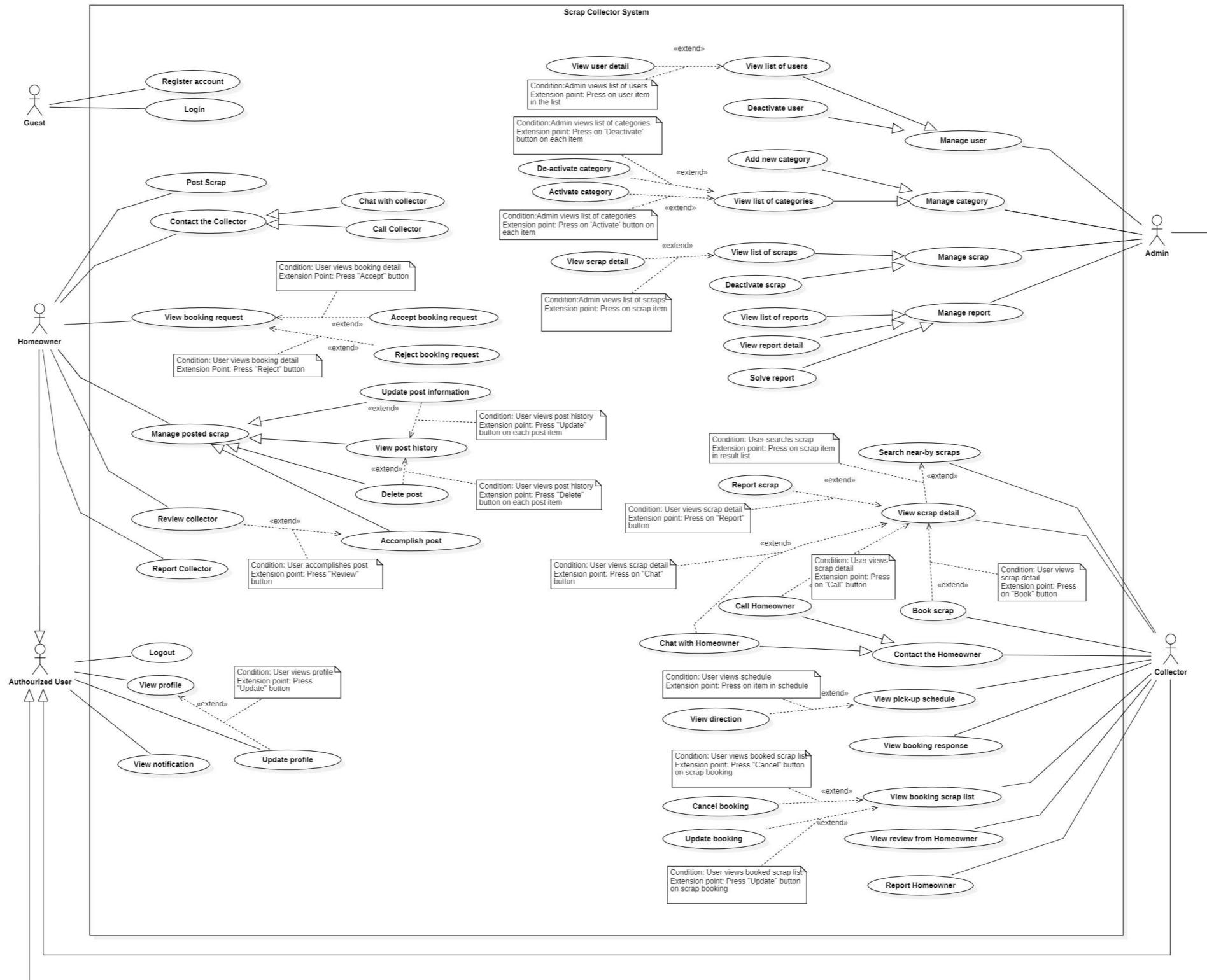


Figure 3 - Use Case diagram

F. Use-case specification

1. Homeowner Use Case

1.1 Post scrap

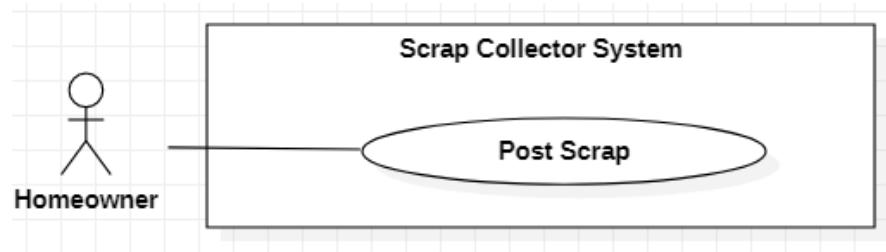


Figure 4 - <Homeowner> Post scrap Use Case

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:	<ul style="list-style-type: none"> - Homeowner 								
Summary:	<ul style="list-style-type: none"> - Allow Homeowner to post scrap for sale. 								
Goal:	<ul style="list-style-type: none"> - Homeowner posts scrap for selling. 								
Triggers:	<ul style="list-style-type: none"> - Homeowner presses “Đăng bán” button in bottom tab bar. 								
Preconditions:	<ul style="list-style-type: none"> - User must sign in as a Homeowner 								
Post conditions:	<ul style="list-style-type: none"> - Success: A new scrap is posted. - Fail: System shows error messages. Scrap is not posted. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Homeowner goes to post scrap view.</td><td> System requires identity information from Homeowner: <ul style="list-style-type: none"> - “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 Người thu mua có thể đến lấy” (required): including from time (selector), to time (selector) and day of week - “Phân loại” (required): selection - “Đơn vị”: text, auto filled when user choose category of scrap - “Khối lượng/Số lượng” (required): number input - “Địa chỉ”: text input - “Số điện thoại”: number </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Homeowner goes to post scrap view.	System requires identity information from Homeowner: <ul style="list-style-type: none"> - “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 Người thu mua có thể đến lấy” (required): including from time (selector), to time (selector) and day of week - “Phân loại” (required): selection - “Đơn vị”: text, auto filled when user choose category of scrap - “Khối lượng/Số lượng” (required): number input - “Địa chỉ”: text input - “Số điện thoại”: number
Step	Actor Action	System Response							
1	Homeowner goes to post scrap view.	System requires identity information from Homeowner: <ul style="list-style-type: none"> - “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 Người thu mua có thể đến lấy” (required): including from time (selector), to time (selector) and day of week - “Phân loại” (required): selection - “Đơn vị”: text, auto filled when user choose category of scrap - “Khối lượng/Số lượng” (required): 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 scrap and lets 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 thể trống thông tin này”.
3	Homeowner lets “Mô tả” input empty	Focus in “Mô tả” field. Show error message “Không thể trống thông tin này”.
4	Homeowner lets “Giá bán” input empty	Focus in “Giá bán” field. Show error message “Không thể 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 scrap will have “SELLING” status.
- System records time of posting scrap 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.
- If the homeowner changes the contact information of scrap, the new contact information is used by that scrap only.
- Each scrap can has one or many available time. Available time is shown as “Từ ... đến ... thứ ...”
- User can only choose categories that are activated.
- Depend on the category of scrap that the label of “Khối lượng/ Số lượng” is shown. If the unit of that category is unit of mass measurement such as “kg”, “gram”, system shows that label as “Khối lượng”, or else, system shows that label as “Số lượng”

Table 4 - <Homeowner> Post scrap Use Case Specification

1.2 Accept booking request

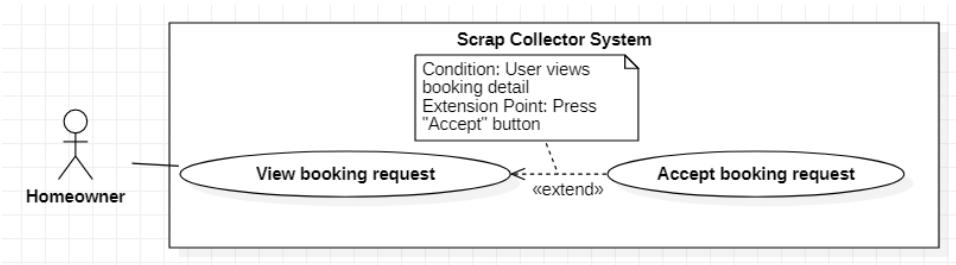


Figure 5 - <Homeowner> Accept booking request Use Case

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 accepts 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 Chấp nhận 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 status of that booking is changed to “ACCEPTED”.
- After a booking is accepted, the status of the scrap that belongs to that booking is changed to “BOOKED”.
- If a scrap has more than one booking requests from many collectors, when a booking request is accepted, other bookings are automatically rejected.
- When a booking is accepted, system sends notification to the collector of that booking.
- When a booking is rejected, system sends notification to the collector of that booking.

Table 5 - <Homeowner> Accept booking request Use Case Specification

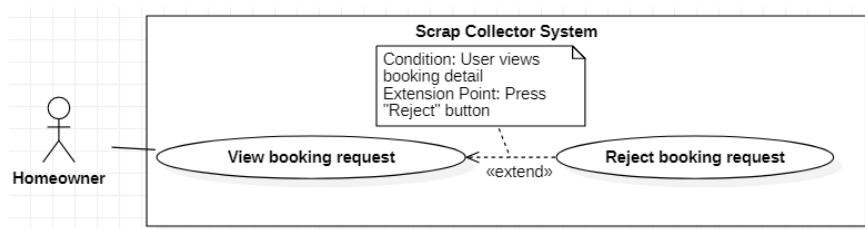
1.3 Reject booking request

Figure 6 - <Homeowner> Reject booking request Use Case

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:	<ul style="list-style-type: none"> Homeowner 								
Summary:	<ul style="list-style-type: none"> Allow Homeowner to reject scrap booking request. 								
Goal:	<ul style="list-style-type: none"> Homeowner does not allow to sell scrap for the Collector. 								
Triggers:	<ul style="list-style-type: none"> User presses “Tù chối” button in scrap booking request notification. 								
Preconditions:	<ul style="list-style-type: none"> User must be a homeowner Homeowner view scrap booking 								
Post conditions:	<ul style="list-style-type: none"> Success: The scrap booking request is canceled. Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Homeowner presses “Tù chối” button in scrap booking notification.</td><td> System shows a pop-up with: <ul style="list-style-type: none"> “Bạn muốn từ chối yêu cầu này?” message “Tù chối” button “Đóng” button </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Homeowner presses “Tù chối” button in scrap booking notification.	System shows a pop-up with: <ul style="list-style-type: none"> “Bạn muốn từ chối yêu cầu này?” message “Tù chối” button “Đóng” button
Step	Actor Action	System Response							
1	Homeowner presses “Tù chối” button in scrap booking notification.	System shows a pop-up with: <ul style="list-style-type: none"> “Bạn muốn từ chối yêu cầu này?” message “Tù chối” button “Đóng” button 							

2	Homeowner presses “Tù 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:**

- After being rejected, the status of that booking is changed to “REJECTED”.
- If a scrap has more than one booking requests from many collectors, when a booking request is rejected, other bookings are not be affected.
- After a booking is rejected, system sends notification to the collector of that booking.
- When a booking is sent to Homeowner, if after 2 days, Homeowner does not take any effect of that booking, system sends notification to that Homeowner. And after 3 days, if Homeowner does not take any effect of that booking, the booking will be automatically rejected.

Table 6 - <Homeowner> Reject booking request Use Case Specification

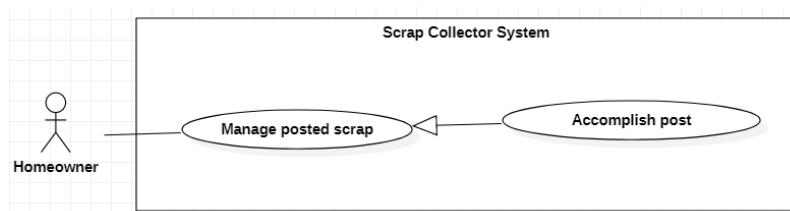
1.4 Accomplish scrap

Figure 7 - <Homeowner> Accomplish scrap Use Case

USE CASE - SC_UC_16					
Use Case No.	SC_UC_16	Use Case Version	1.0		
Use Case Name	Accomplish scrap				
Author	Nguyen Duy Dat	Priority	Normal		
Date	26/2/2020				
Actor:					
- Homeowner					
Summary:					
- Allow homeowner to accomplish a post.					
Goal:					
- Homeowner accomplishes 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
- After being accomplished, the scrap is changed to “SOLD” status, and the booking related to that scrap is changed to “DONE” status.

Table 7 - <Homeowner> Accomplish scrap Use Case Specification

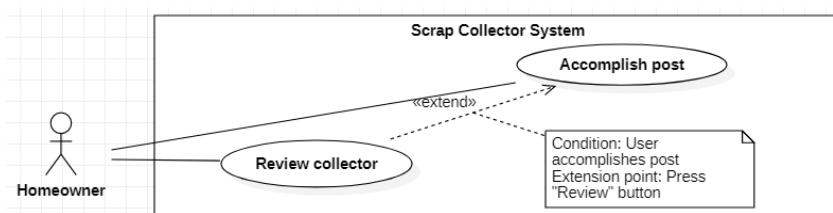
1.5 Review Collector

Figure 8 - <Homeowner> Review Collector Use Case

USE CASE - SC_UC_17			
Use Case No.	SC_UC_17	Use Case Version	1.0
Use Case Name	Review collector		
Author	Nguyen Duy Dat		
Date	26/2/2020	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Homeowner 		
Summary:	<ul style="list-style-type: none"> - Allow homeowner to review collector. 		
Goal:			

<ul style="list-style-type: none"> - Homeowner reviews Collector with star and description. <p>Triggers:</p> <ul style="list-style-type: none"> - User presses “Đánh giá” button. <p>Preconditions:</p> <ul style="list-style-type: none"> - User must be a homeowner - Homeowner accomplishes a post <p>Post conditions:</p> <ul style="list-style-type: none"> - Success: A review is created. - Fail: System shows error messages. Review is not created. <p>Main Success Scenario:</p> <table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Homeowner presses “Đánh giá” button</td><td>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” button - “Đóng” button</td></tr> <tr> <td>2</td><td>Homeowner chooses star, inputs description and presses “Gửi” button</td><td>System creates review and sends notification to Collector. [Exception 1]</td></tr> </tbody> </table> <p>Alternative Scenario:</p> <table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Homeowner presses “Đóng” button</td><td>System closes Review pop-up</td></tr> </tbody> </table> <p>Exceptions:</p> <table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td></td><td>System shows message “Không có kết nối Internet” when the Internet is lost.</td></tr> </tbody> </table> <p>Relationships: [SC_UC_16]</p> <p>Business Rules:</p> <ul style="list-style-type: none"> - Each review maps with a 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 Collector has many reviews. - Each review belongs to one Collector only - Default number of star of review is 5, which means that when the Homeowner send review without choosing the number of star, the review is created with 5 stars - After the review is created, system sends notification to the Collector 	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” button - “Đóng” button	2	Homeowner chooses star, inputs description and presses “Gửi” button	System creates review and sends notification to Collector. [Exception 1]	No	Actor Action	System Response	1	Homeowner presses “Đóng” button	System closes Review pop-up	No	Actor Action	System Response	1		System shows message “Không có kết nối Internet” when the Internet is lost.
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” button - “Đóng” button																			
2	Homeowner chooses star, inputs description and presses “Gửi” button	System creates review and sends notification to Collector. [Exception 1]																			
No	Actor Action	System Response																			
1	Homeowner presses “Đóng” button	System closes Review pop-up																			
No	Actor Action	System Response																			
1		System shows message “Không có kết nối Internet” when the Internet is lost.																			

Table 8 - <Homeowner> Review Collector Use Case Specification

2. Collector Use Case

2.1 Search near-by scraps

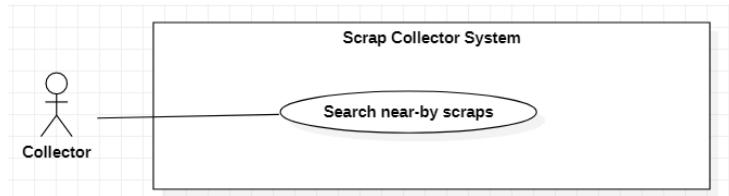


Figure 9 - <Collector> Search near-by scraps Use Case

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 Hoang Anh Trung, Nguyen Hoang Nhan											
Date	26/1/2020	Priority	Normal									
Actor:	<ul style="list-style-type: none"> - Collector 											
Summary:	<ul style="list-style-type: none"> - Allow collector to find near-by scrap by radius, scrap category. 											
Goal:	<ul style="list-style-type: none"> - Collector can see near-by scrap post to book. 											
Triggers:	<ul style="list-style-type: none"> - Collector presses “Trang chủ” icon in tab navigator. 											
Preconditions:	<ul style="list-style-type: none"> - User must login in the system with role “Collector” - Collector has registered his/her location. 											
Post conditions:	<ul style="list-style-type: none"> - Success: A list of scrap post is shown on screen. - Fail: System shows error messages. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Collector goes to “Trang chủ” screen</td><td>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]</td></tr> </tbody> </table>			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]			
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:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Collector selects type of scrap category in “Phân loại” selection</td><td>System searches scrap post with selected category and shows search result</td></tr> </tbody> </table>			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			
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:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td></td><td>System shows message “Không có kết nối Internet” when the Internet is lost.</td></tr> <tr> <td>2</td><td></td><td>System shows message the “Vui lòng bật định vị GPS để ứng dụng Ve chai có thể hoạt động tốt hơn” when user turn off GPS service of device.</td></tr> </tbody> </table>			No	Actor Action	System Response	1		System shows message “Không có kết nối Internet” when the Internet is lost.	2		System shows message the “Vui lòng bật định vị GPS để ứng dụng Ve chai có thể hoạt động tốt hơn” when user turn off GPS service of device.
No	Actor Action	System Response										
1		System shows message “Không có kết nối Internet” when the Internet is lost.										
2		System shows message the “Vui lòng bật định vị GPS để ứng dụng Ve chai có thể hoạt động tốt hơn” when user turn off GPS service of device.										

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 Collector. Collector can choose to find scrap in 5km, 10km, 15km or at most 20km.
- The default radius to search scraps when Collector goes to Home screen is 10 km.
- Admin can edit default search distance value through Web Admin.

Table 9 - <Collector> Search near-by scraps Use Case Specification

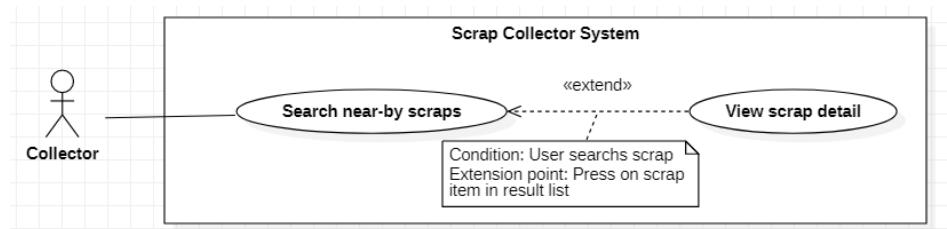
2.2 View scrap detail

Figure 10 - <Collector> View scrap detail Use Case

USE CASE - SC_UC_19									
Use Case No.	SC_UC_19	Use Case Version	1.0						
Use Case Name	View scrap detail								
Author	Nguyen Hoang Anh Trung								
Date	26/1/2020	Priority	Medium						
Actor:	<ul style="list-style-type: none"> - Collector 								
Summary:	<ul style="list-style-type: none"> - Allow Collector to view scrap detail. 								
Goal:	<ul style="list-style-type: none"> - Collector views detail of the scrap then decide whether to buy or not. 								
Triggers:	<ul style="list-style-type: none"> - Collector presses the item of scrap. 								
Preconditions:	<ul style="list-style-type: none"> - User must login in the system with role "Collector". - Collector searches for near-by scrap. 								
Post conditions:	<ul style="list-style-type: none"> - Success: A new post is created. - Fail: System shows error messages. Post is not created. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Collector press in the Item of scrap on search result list</td><td> System shows the detail of scrap with: <ul style="list-style-type: none"> - "Tiêu đề": Text - "Giá bán": Text - "Hình ảnh": Image slider - "Phân loại": Text - "Đơn vị": Text - "Số lượng": Text - "Mô tả": Text </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Collector press in the Item of scrap on search result list	System shows the detail of scrap with: <ul style="list-style-type: none"> - "Tiêu đề": Text - "Giá bán": Text - "Hình ảnh": Image slider - "Phân loại": Text - "Đơn vị": Text - "Số lượng": Text - "Mô tả": Text
Step	Actor Action	System Response							
1	Collector press in the Item of scrap on search result list	System shows the detail of scrap with: <ul style="list-style-type: none"> - "Tiêu đề": Text - "Giá bán": Text - "Hình ảnh": Image slider - "Phân loại": Text - "Đơn vị": Text - "Số lượng": Text - "Mô tả": Text 							

		<ul style="list-style-type: none"> - "Thời gian": Text - "Gọi" button - "Chat" button - "Đặt ve chai" button <p>[Exception 1] [Exception 2]</p>
--	--	---

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] [SC_UC_2.0_2]**Business Rules:**

- If Collector has booked that scrap, the "Đặt ve chai" button is disabled, and the button is displayed with message "Bạn đã đặt Ve chai này".
- When Collector views scrap detail, Collector cannot know the address of scrap.
- When Collector views scrap detail, Collector can chat with Homeowner about that scrap.
- When Collector views scrap detail, Collector can call the Homeowner for detail information.
- When Collector views scrap detail, Collector can report scrap which is not suitable.

Table 10 - <Collector> View scrap detail Use Case Specification

2.3 Book scrap

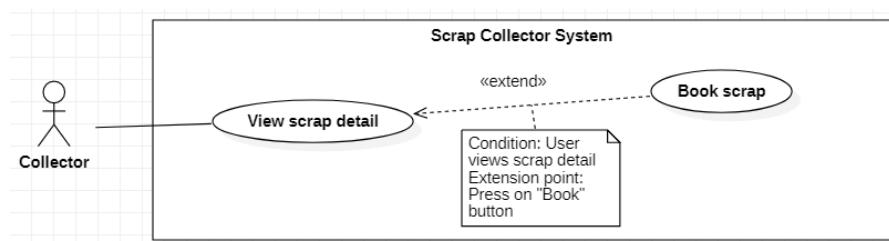


Figure 11 - <Collector> Book scrap Use Case

USE CASE - SC_UC_20			
Use Case No.	SC_UC_20	Use Case Version	1.0
Use Case Name	Book scrap		
Author	Nguyen Hoang Anh Trung		
Date	26/2/2020	Priority	High
Actor:	<ul style="list-style-type: none"> - Collector 		
Summary:	<ul style="list-style-type: none"> - Allow Collector to book scrap 		
Goal:	<ul style="list-style-type: none"> - Collector sends booking request to the Homeowner of the post. 		
Triggers:	<ul style="list-style-type: none"> - User presses "Đặt ve chai" button. 		
Preconditions:	<ul style="list-style-type: none"> - 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): ratio selection - “Đặt ve chai” button - “Hủy” button
2	Collector presses “Đặt ve chai” button [Alternative 1]	System creates booking request, send notification to Homeowner [Exception 1] [Exception 2] [Exception 3]

Alternative Scenario:

No	Actor Action	System Response
1	Collector presses “Hủy” button or “Quay lại” 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 scrap is booked.
3		System shows error message the "Bài đăng không tồn tại" when the scrap is deleted.

Relationships: [SC_UC_19]**Business Rules:**

- After booking is created, Homeowner can accept or reject booking
- Collector can change the price in order to negotiate price
- Collector can choose only one available time to pick up scrap
- The created booking has “WAITING” status
- When Collector book a scrap, the system sends notification to the Homeowner of that scrap
- When a booking is sent to Homeowner, if after 2 days, Homeowner does not take any effect of that booking, system sends notification to that Homeowner. And after 3 days, if Homeowner does not take any effect of that booking, the booking will be automatically rejected.

Table 11 - <Collector> Book scrap Use Case Specification

2.4 Chat with Homeowner

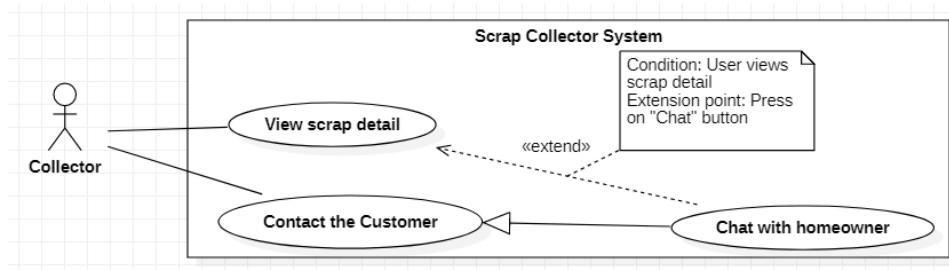


Figure 12 - <Collector> Chat with homeowner Use Case

USE CASE - SC_UC_22															
Use Case No.	SC_UC_22	Use Case Version	1.0												
Use Case Name	Chat with homeowner														
Author	Nguyen Hoang Anh Trung, Tiet Le Bao Khanh														
Date	26/2/2020	Priority	Normal												
Actor:	<ul style="list-style-type: none"> - Collector 														
Summary:	<ul style="list-style-type: none"> - Allow Collector to chat with Homeowner 														
Goal:	<ul style="list-style-type: none"> - Collector chats with one or many Homeowner. 														
Triggers:	<ul style="list-style-type: none"> - User presses "Chat" button. - User presses "Chat" tab in bottom tab bar. 														
Preconditions:	<ul style="list-style-type: none"> - User must be a Collector - Collector views post detail 														
Post conditions:	<ul style="list-style-type: none"> - Success: A conversation between Homeowner and Collector is created. - Fail: System shows error messages. 														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Collector presses "Chat" button or presses "Chat" tab in tab bar</td><td>System shows "Chat" screen with all conversations of that Collector</td></tr> <tr> <td>2</td><td>Collector chooses a conversation</td><td>System shows messages in chosen conversation, a text input and a "Gửi" button</td></tr> <tr> <td>3</td><td>Collector inputs message and presses "Gửi" button</td><td>System sends message to the Homeowner in that conversation and shows that message [Exception 1]</td></tr> </tbody> </table>			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]
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:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td></td><td>System shows message "Không có kết nối Internet" when the Internet is lost.</td></tr> </tbody> </table>			No	Actor Action	System Response	1		System shows message "Không có kết nối Internet" when the Internet is lost.						
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:															

- Each chat room between Collector and Homeowner relates to a scrap only.
- If Collector and Homeowner chat about more than one scrap, for each scrap, a new chat room is created.
- If the scrap which is being chatted about is deleted or sold, system shows message to let Collector know that scrap is deleted or sold, and Collector cannot view detail of that scrap.

Table 12 - <Collector> Chat with homeowner Use Case Specification

2.5 View pick-up schedule

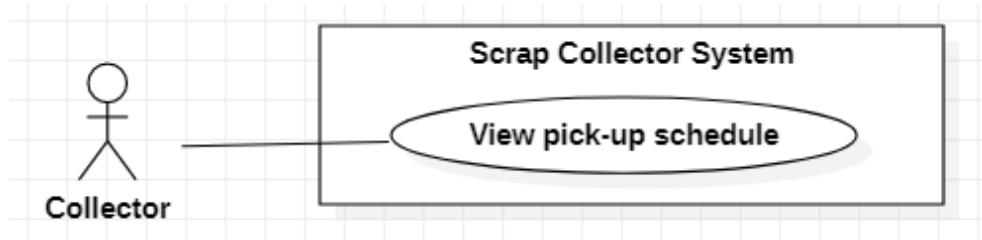


Figure 13 - <Collector> View pick-up schedule Use Case

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 Hoang Anh Trung								
Date	26/2/2020	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Collector 								
Summary:	<ul style="list-style-type: none"> - Allow Collector to view pick-up schedule 								
Goal:	<ul style="list-style-type: none"> - Collector views optimized pick-up schedule to enhance effectivity. 								
Triggers:	<ul style="list-style-type: none"> - User presses “Lịch trình” button. 								
Preconditions:	<ul style="list-style-type: none"> - User must be a Collector 								
Post conditions:	<ul style="list-style-type: none"> - Success: System shows pick-up schedule. - Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Collector presses “Lịch trình” button.</td><td> <p>System shows “Lịch trình” screen with date of current day and pick-up schedule.</p> <p>Each item in schedule contains:</p> <ul style="list-style-type: none"> - Index of scrap - Title of scrap - Address of scrap - Price of booking - Time to pick up scrap - “Chỉ đường” button - “Gọi” button <p>[Exception 1]</p> </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Collector presses “Lịch trình” button.	<p>System shows “Lịch trình” screen with date of current day and pick-up schedule.</p> <p>Each item in schedule contains:</p> <ul style="list-style-type: none"> - Index of scrap - Title of scrap - Address of scrap - Price of booking - Time to pick up scrap - “Chỉ đường” button - “Gọi” button <p>[Exception 1]</p>
Step	Actor Action	System Response							
1	Collector presses “Lịch trình” button.	<p>System shows “Lịch trình” screen with date of current day and pick-up schedule.</p> <p>Each item in schedule contains:</p> <ul style="list-style-type: none"> - Index of scrap - Title of scrap - Address of scrap - Price of booking - Time to pick up scrap - “Chỉ đường” button - “Gọi” button <p>[Exception 1]</p>							

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:**

- In "Lịch trình" screen, system only shows booking that is booked with the day of week that is the same as current day of week.
- Collector can views all the booking that Homeowner has accepted.
- All bookings with "ACCEPTED" status will be shown in schedule
- The scrap is scheduled based on available time. The scrap that has sooner available time is placed first.

Table 13 - <Collector> View pick-up schedule Use Case Specification

3. Admin Use Case

3.1 View list of users

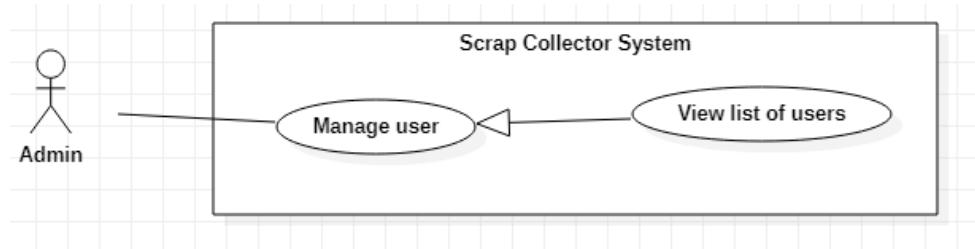


Figure 14 - <Admin> View list of users Use Case

USE CASE - SC_UC_29			
Use Case No.	SC_UC_29	Use Case Version	1.0
Use Case Name	View list of users		
Author	Tiet Le Bao Khanh		
Date	26/2/2020	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Admin 		
Summary:	<ul style="list-style-type: none"> - Allow Admin to view all users in the system 		
Goal:	<ul style="list-style-type: none"> - Admin views all users in the system, knows how many user in the system and filter user by role 		
Triggers:	<ul style="list-style-type: none"> - Admin goes to Account page on Admin website 		
Preconditions:	<ul style="list-style-type: none"> - User must be an Admin - User has logged into the website 		
Post conditions:	<ul style="list-style-type: none"> - Success: A table of users is shown on screen. - Fail: System shows error messages. 		

Main Success Scenario:		
Step	Actor Action	System Response
1	Admin go to Account page on website	<p>System show table of accounts in the System with following information:</p> <ul style="list-style-type: none"> - "Name": Full name of user - "Username": username - "Phone number" - "Role" - "Active date" - "Action" button <p>System show total of user in the system.</p> <p>System also show an option form for Admin to filter account:</p> <ul style="list-style-type: none"> - Search by Role - Sort by active time - Search by name - Search by Phone - Search by date of activation <p>[Exception 1]</p>
Alternative Scenario: N/A		
Exceptions:		
No	Actor Action	System Response
1		System shows message the "Cannot load table of user" when there is error.

Relationships: Generalized from Manage user account use case

Business Rules:

- All accounts are shown even if that account is disabled.
- System shows the total of users in the system for Admin to get an overview of system.

Table 14 - <Admin> View list of users Use Case Specification

3.2 View list of scraps

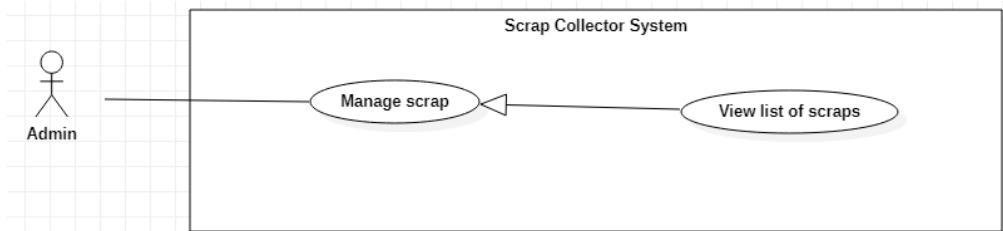


Figure 15 - <Admin> View list of scraps Use Case

USE CASE - SC_UC_31			
Use Case No.	SC_UC_31	Use Case Version	1.0
Use Case Name	View list of scraps		
Author	Tiet Le Bao Khanh		
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 page.		
Preconditions:		
- User must be an Admin		
Post conditions:		
- Success: A list of scraps is shown. - Fail: System shows error messages.		
Main Success Scenario:		
Step	Actor Action	System Response
1	Admin goes to Scrap screen	<p>System show table of accounts in the System with following information:</p> <ul style="list-style-type: none"> - "Title": Title of scrap - "Price": price of scrap - "Category": category of scrap - "Quantity": quantity or mass of scrap - "Created time": time of posting scrap - "Status": status of scrap - "Action" button <p>System show total of scrap in the system. System also show an option form for Admin to filter account:</p> <ul style="list-style-type: none"> - Filter by Category - Filter by status - Sort by created time of scrap - Search by title of scrap - Search by Price of scrap - Search by date of scrap <p>[Exception 1]</p>
Alternative Scenario: N/A		
Exceptions:		
No	Actor Action	System Response
1		System shows message the "Cannot load data" when there is error.
Relationships: Generalized from Manage posted scrap use case		
Business Rules:		
<ul style="list-style-type: none"> - System shows the total of scraps in the system for Admin to get an overview of system. 		

Table 15 - <Admin> View list of scraps Use Case Specification

3.3 Deactivate category

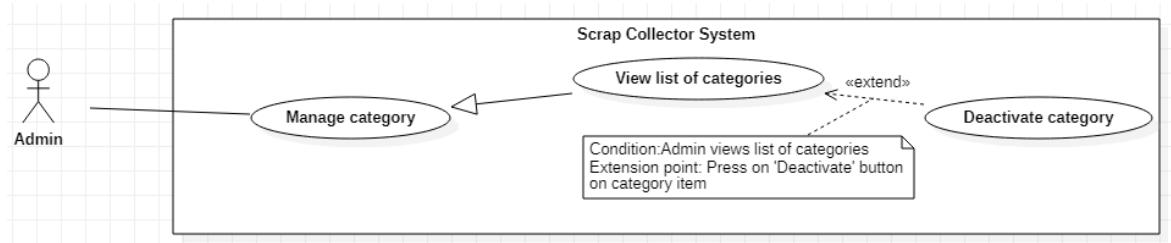


Figure 16 - <Admin> Deactivate category Use Case

USE CASE - SC_UC_34															
Use Case No.	SC_UC_34	Use Case Version	1.0												
Use Case Name	Deactivate category														
Author	Tiet Le Bao Khanh														
Date	26/2/2020	Priority	Normal												
Actor:	<ul style="list-style-type: none"> - Admin 														
Summary:	<ul style="list-style-type: none"> - Allow Admin to deactivate a category 														
Goal:	<ul style="list-style-type: none"> - Admin changes status of category so that category cannot be used. 														
Triggers:	<ul style="list-style-type: none"> - Admin update category status with status "DEACTIVE". 														
Preconditions:	<ul style="list-style-type: none"> - User must be an Admin - Admin views list of categories 														
Post conditions:	<ul style="list-style-type: none"> - Success: Category is deactivated. - Fail: System shows error messages. Category is not deactivated. 														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Admin clicks on "Update" button on category item</td><td> System shows update category form with: <ul style="list-style-type: none"> - Name (editable): text input - Unit (editable): text input - Status: selection "Update" button </td></tr> <tr> <td>2</td><td>Admin select status of scrap as "IN-ACTIVE"</td><td></td></tr> <tr> <td>3</td><td>Admin clicks "Update" button</td><td>System shows result message "Deactivate category successfully"</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Admin clicks on "Update" button on category item	System shows update category form with: <ul style="list-style-type: none"> - Name (editable): text input - Unit (editable): text input - Status: selection "Update" button	2	Admin select status of scrap as "IN-ACTIVE"		3	Admin clicks "Update" button	System shows result message "Deactivate category successfully"
Step	Actor Action	System Response													
1	Admin clicks on "Update" button on category item	System shows update category form with: <ul style="list-style-type: none"> - Name (editable): text input - Unit (editable): text input - Status: selection "Update" button													
2	Admin select status of scrap as "IN-ACTIVE"														
3	Admin clicks "Update" button	System shows result message "Deactivate category successfully"													
Alternative Scenario: N/A															
Exceptions:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td></td><td>System shows message "Deactivate category FAILED" when there is error.</td></tr> </tbody> </table>			No	Actor Action	System Response	1		System shows message "Deactivate category FAILED" when there is error.						
No	Actor Action	System Response													
1		System shows message "Deactivate category FAILED" when there is error.													
Relationships:	Generalized from Manage category use case, [SC_UC_36]														
Business Rules:	<ul style="list-style-type: none"> - Only categories that have no selling scrap can be deactivated. 														

- After a category is deactivated, Homeowner cannot post scrap belonging to that category, and Collector cannot search for scrap belonging to that category.

Table 16 - <Admin> Deactivate category Use Case Specification

3.4 Add new category

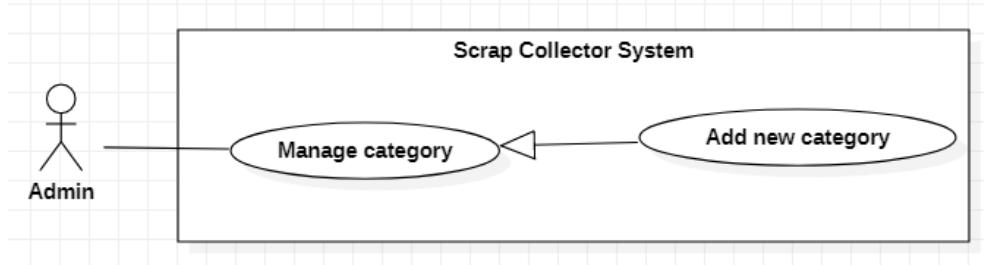


Figure 17 - <Admin> Add new category Use Case

USE CASE - SC_UC_35															
Use Case No.	SC_UC_35	Use Case Version	1.0												
Use Case Name	Add new category														
Author	Tiet Le Bao Khanh														
Date	26/2/2020	Priority	Normal												
Actor:	<ul style="list-style-type: none"> - Admin 														
Summary:	<ul style="list-style-type: none"> - Allow Admin to add new category 														
Goal:	<ul style="list-style-type: none"> - Admin adds a new category. 														
Triggers:	<ul style="list-style-type: none"> - Admin presses “Add New category” button. 														
Preconditions:	<ul style="list-style-type: none"> - User must be an Admin - Admin views list of category 														
Post conditions:	<ul style="list-style-type: none"> - Success: A new category is created. - Fail: System shows error messages. Category is not created. 														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Admin clicks “Add new category” button</td><td> Systems shows a modal for Admin to add category, with: <ul style="list-style-type: none"> - Category name(required): text input - Unit(required): text input - “Add” button </td></tr> <tr> <td>2</td><td>Admin inputs required fields (category name, unit)</td><td></td></tr> <tr> <td>3</td><td>Admin clicks on “Add” button</td><td> Systems add category and shows result “Add category successfully” [Exception 1] [Exception 2] </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Admin clicks “Add new category” button	Systems shows a modal for Admin to add category, with: <ul style="list-style-type: none"> - Category name(required): text input - Unit(required): text input - “Add” button 	2	Admin inputs required fields (category name, unit)		3	Admin clicks on “Add” button	Systems add category and shows result “Add category successfully” [Exception 1] [Exception 2]
Step	Actor Action	System Response													
1	Admin clicks “Add new category” button	Systems shows a modal for Admin to add category, with: <ul style="list-style-type: none"> - Category name(required): text input - Unit(required): text input - “Add” button 													
2	Admin inputs required fields (category name, unit)														
3	Admin clicks on “Add” button	Systems add category and shows result “Add category successfully” [Exception 1] [Exception 2]													
Alternative Scenario:	N/A														

Exceptions:		
No	Actor Action	System Response
1		System shows message "This category has existed" when the inputted category is duplicated.
Relationships: Generalized from Manage category use case		
Business Rules:		
<ul style="list-style-type: none"> - After a category is added, Homeowner can post scrap belonging to that category, and Collector can search for scrap belonging to that category 		

Table 17 - <Admin> Add new category Use Case Specification

3.5 View list of categories

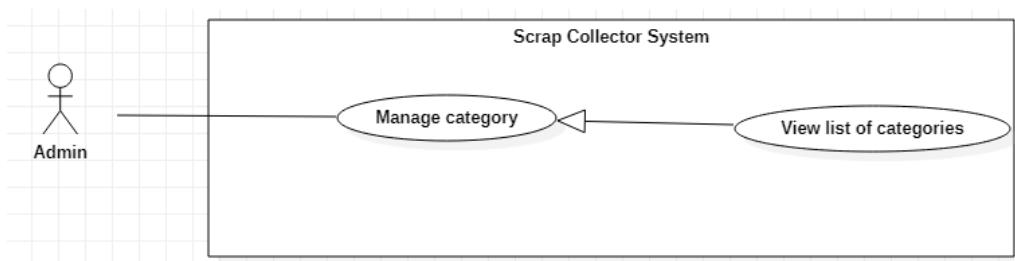


Figure 18 - <Admin> View list of categories Use Case

USE CASE - SC_UC_36									
Use Case No.	SC_UC_36	Use Case Version	1.0						
Use Case Name	View list of categories								
Author	Tiet Le Bao Khanh								
Date	26/2/2020	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Admin 								
Summary:	<ul style="list-style-type: none"> - Allow Admin to view all categories in the system 								
Goal:	<ul style="list-style-type: none"> - Admin views all categories. 								
Triggers:	<ul style="list-style-type: none"> - Admin goes to Category screen. 								
Preconditions:	<ul style="list-style-type: none"> - User must be an Admin 								
Post conditions:	<ul style="list-style-type: none"> - Success: List of categories is shown. - Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Admin goes to category screen</td> <td> <p>System shows list of category, each category is shown with:</p> <ul style="list-style-type: none"> - Category Name: text - Unit: text - Status: text </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Admin goes to category screen	<p>System shows list of category, each category is shown with:</p> <ul style="list-style-type: none"> - Category Name: text - Unit: text - Status: text
Step	Actor Action	System Response							
1	Admin goes to category screen	<p>System shows list of category, each category is shown with:</p> <ul style="list-style-type: none"> - Category Name: text - Unit: text - Status: text 							

		<ul style="list-style-type: none"> - “Action” button group with: “View” button, “Edit” button, “Deactivate” button <p>[Exception 1]</p>
Alternative Scenario: N/A		
Exceptions:		
No	Actor Action	System Response
1		System shows message "Cannot load data" when there is error.

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 18 - <Admin> View list of categories Use Case Specification

G. Architecture Diagram

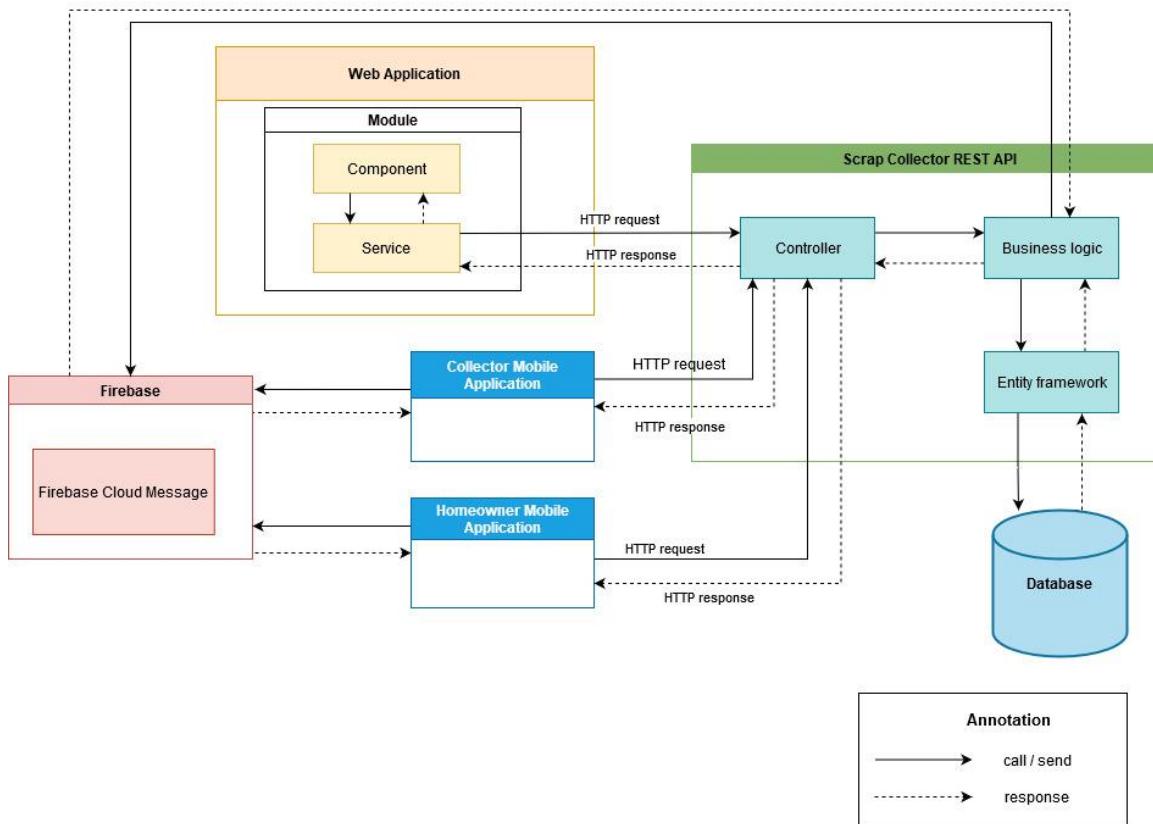


Figure 19 - System Architectural Design

H. Component Diagram

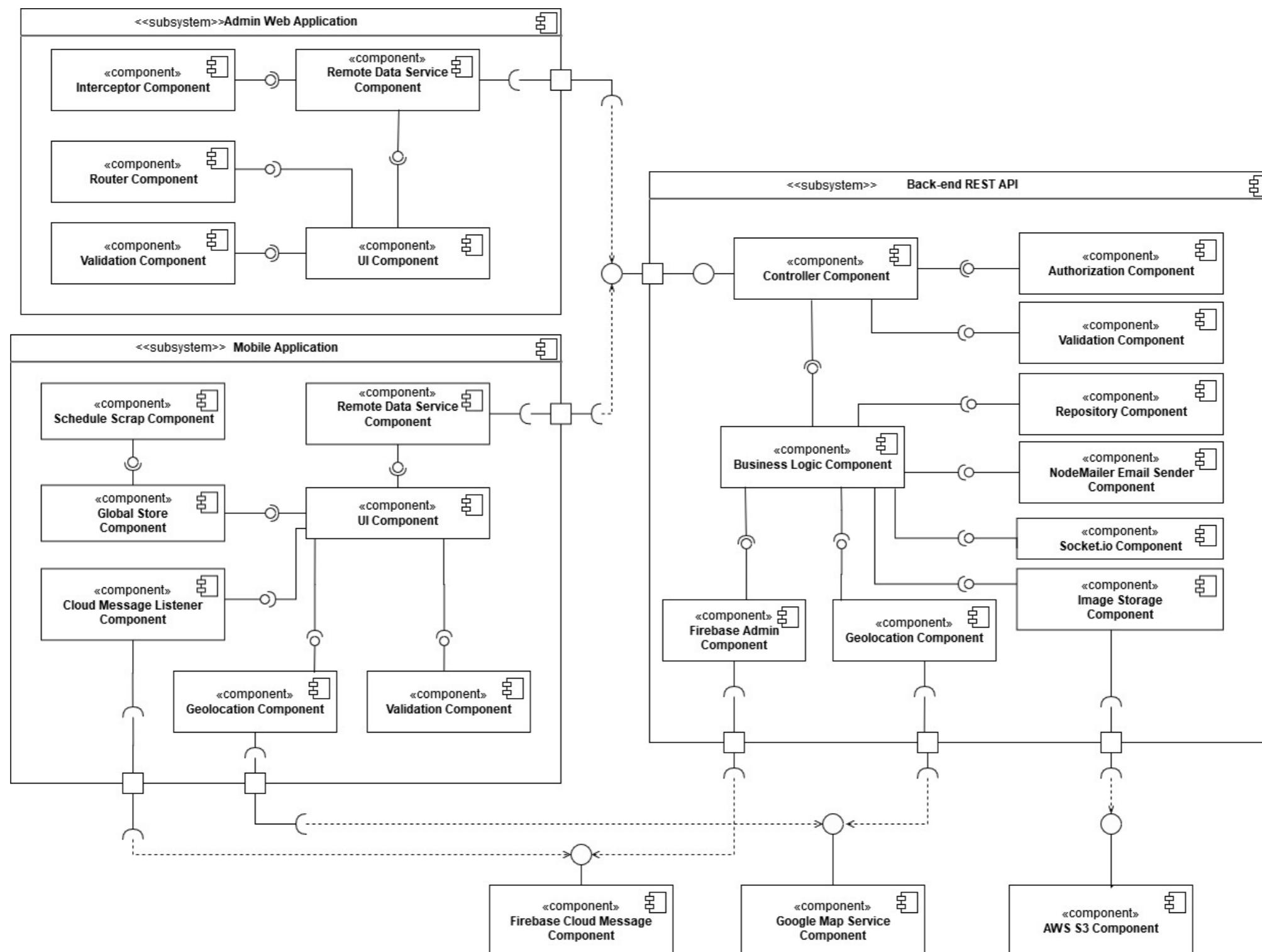


Figure 20 – Component Diagram

COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Component name	Description
Rest API	Subsystem 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	Subsystem which handles HTTP requests from Admin
Mobile Application	Subsystem which helps Homeowner and Collector implement features in the system through UI Components
Firebase Cloud Messaging	Component with Cloud messaging service to send notification to Mobile Application
Google Map Service	Component which solve HTTP request of calculating distance and getting direction
AWS S3 (Amazon Web Service: Simple Storage Service)	Component which has responsibility for storage images of scrap and avatar
UI Component	Component which has responsibility to render UI
Validation Component	Component which has responsibility to check validity of data
Remote Data Service Component	Component which has responsibility to send HTTP request and get HTTP response
Interceptor Component	Component which has responsibility to handle HTTP request and HTTP response
Router Component	Component which has responsibility to redirect user to suitable screen
Schedule Scrap Component	Component which has responsibility to make schedule to pick up scrap
Global Store Component	Component which has responsibility to store global data
Geolocation Component	Component which has responsibility to calculate distance and call Google map service
Cloud Message Listener	Component which has responsibility to handle coming notification from Firebase cloud messaging service
Controller Component	Component to handle HTTP request from Web application and Mobile application
Business Logic Component	Component which has responsibility to handle business logic in the system
Repository Component	Component which has responsibility to execute query to database and return result
Authorization Component	Component which has responsibility to check authentication and authorization

Nodemailer Email Sender Component	Component which has responsibility to send email to user
Image Storage Component	Component which has responsibility to send image to AWS S3 storage and return link of image
Firebase Admin Component	Component which has responsibility to send notification
Socket.io Component	Component which has responsibility to create real-time and event-based communication between server and client

Table 19 - Component Diagram Dictionary

I. Class Diagram

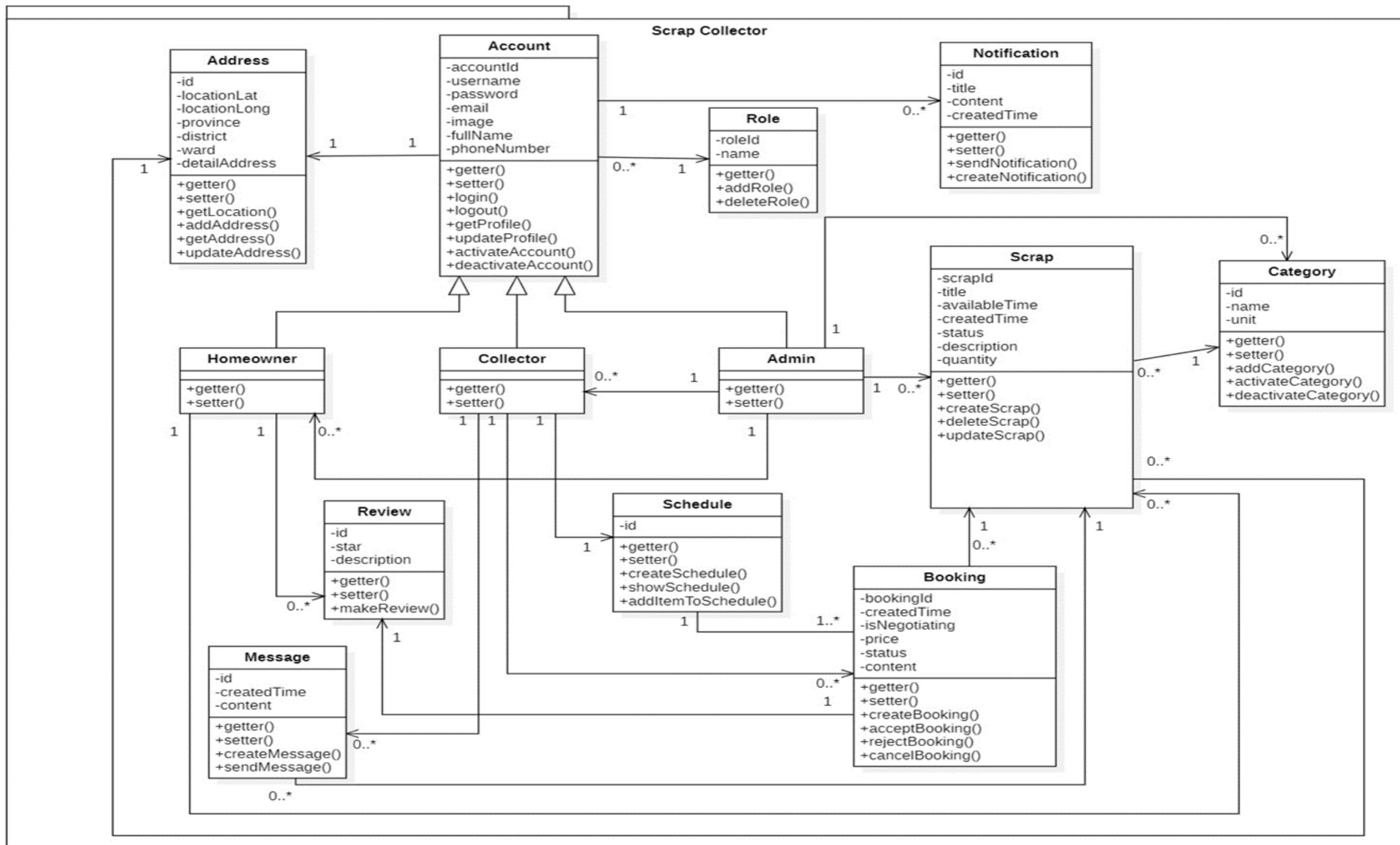


Figure 21 – Class Diagram

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 20 - Class Diagram Dictionary

J. Entity Relationship Diagram

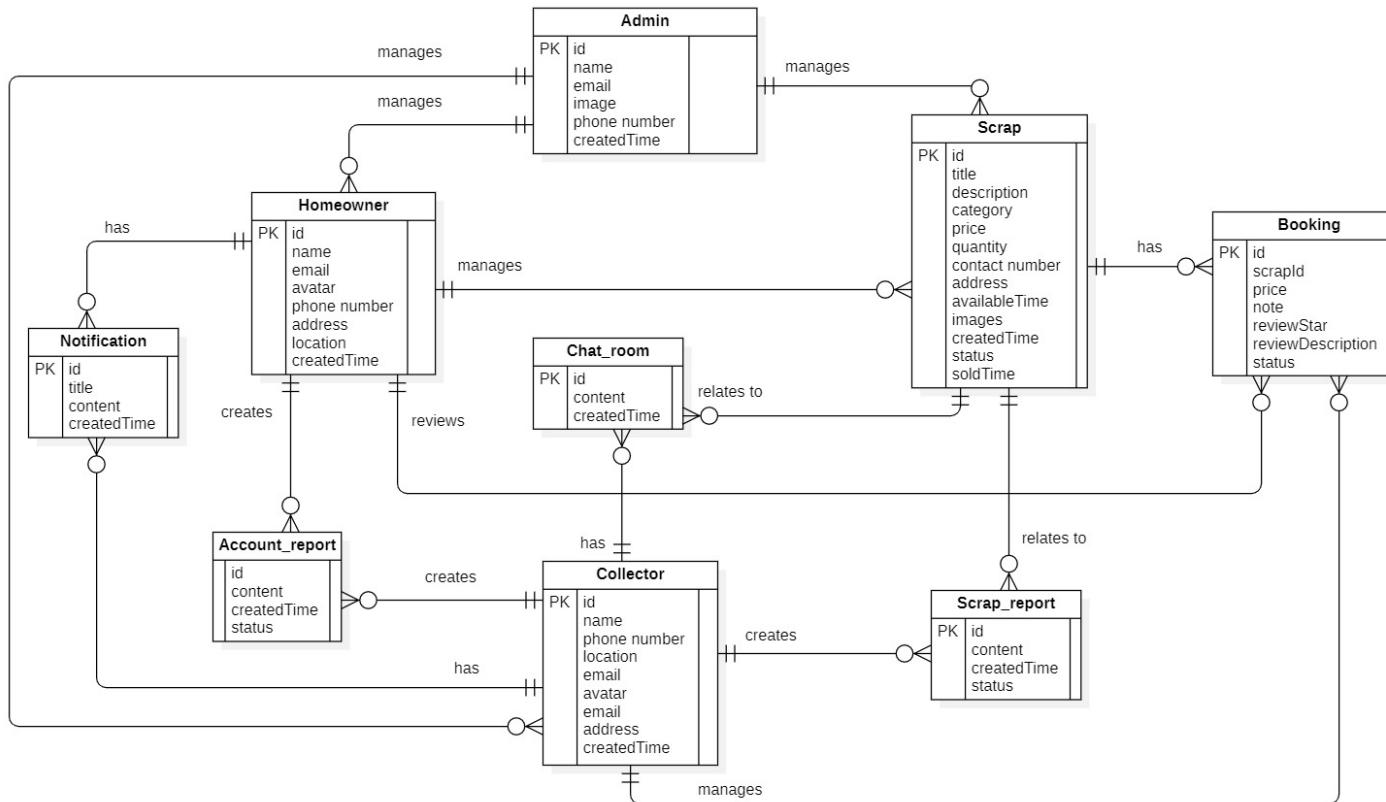


Figure 22 Entity Relationship Diagram

Entity 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
Chat_room	Contains Chat room's information
Scrap_report	Contain scrap report's information
Account_report	Contain account report's information

Table 21 - Entity Data Dictionary

K. Interactive Diagram

1. Get list of category

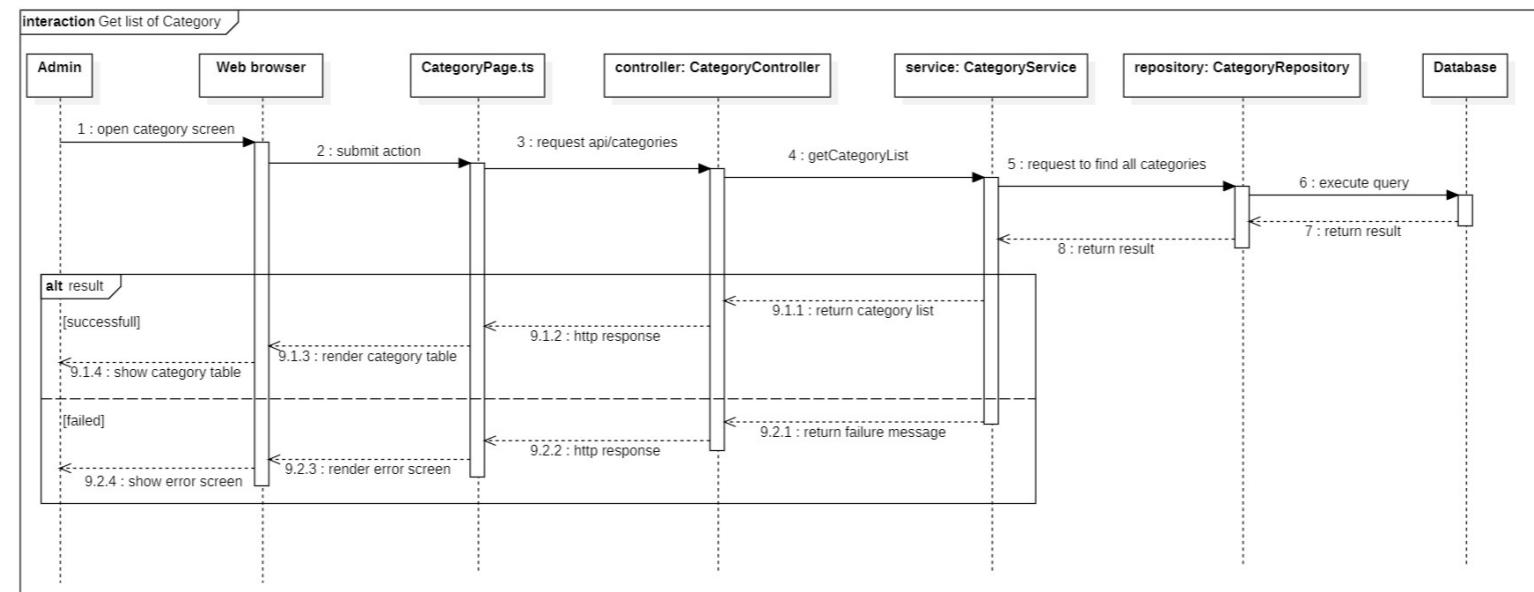


Figure 23 - <Sequence Diagram> Get list of category

2. Add Category

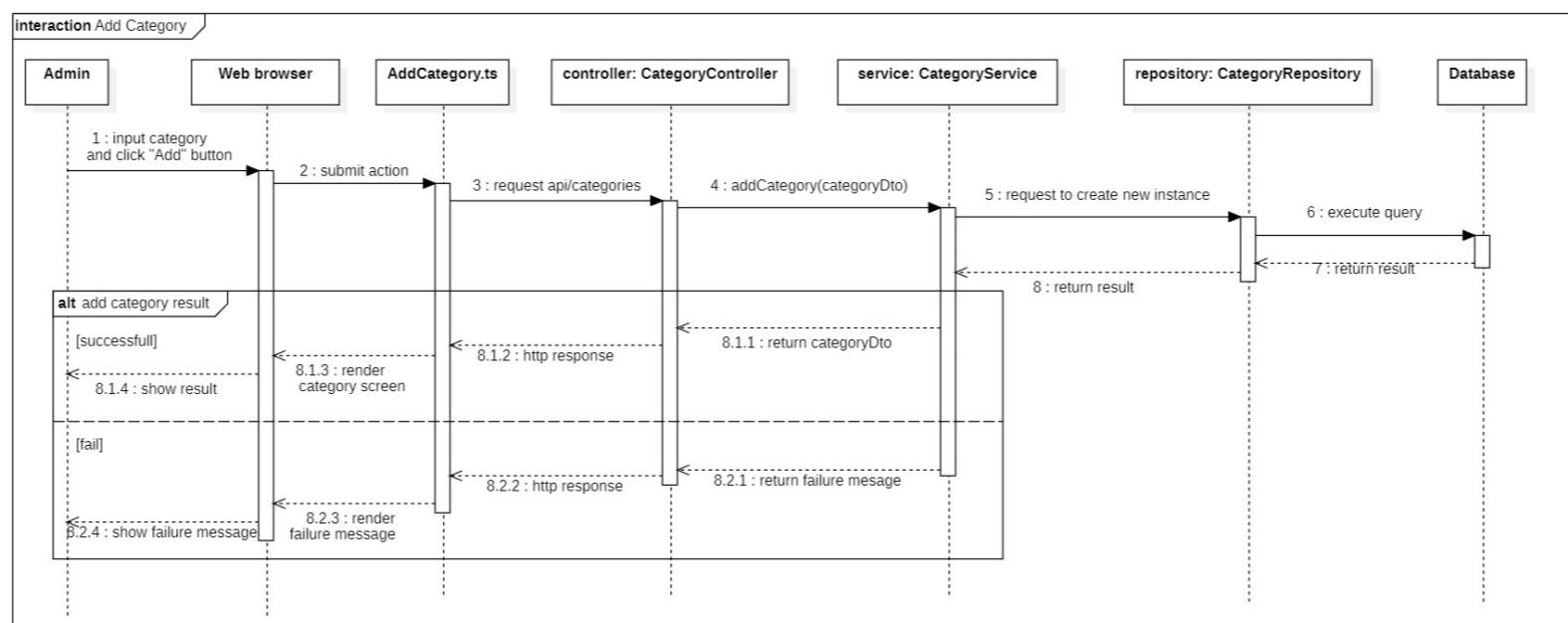


Figure 24 - <Sequence Diagram> Add category

3. Get list of Account

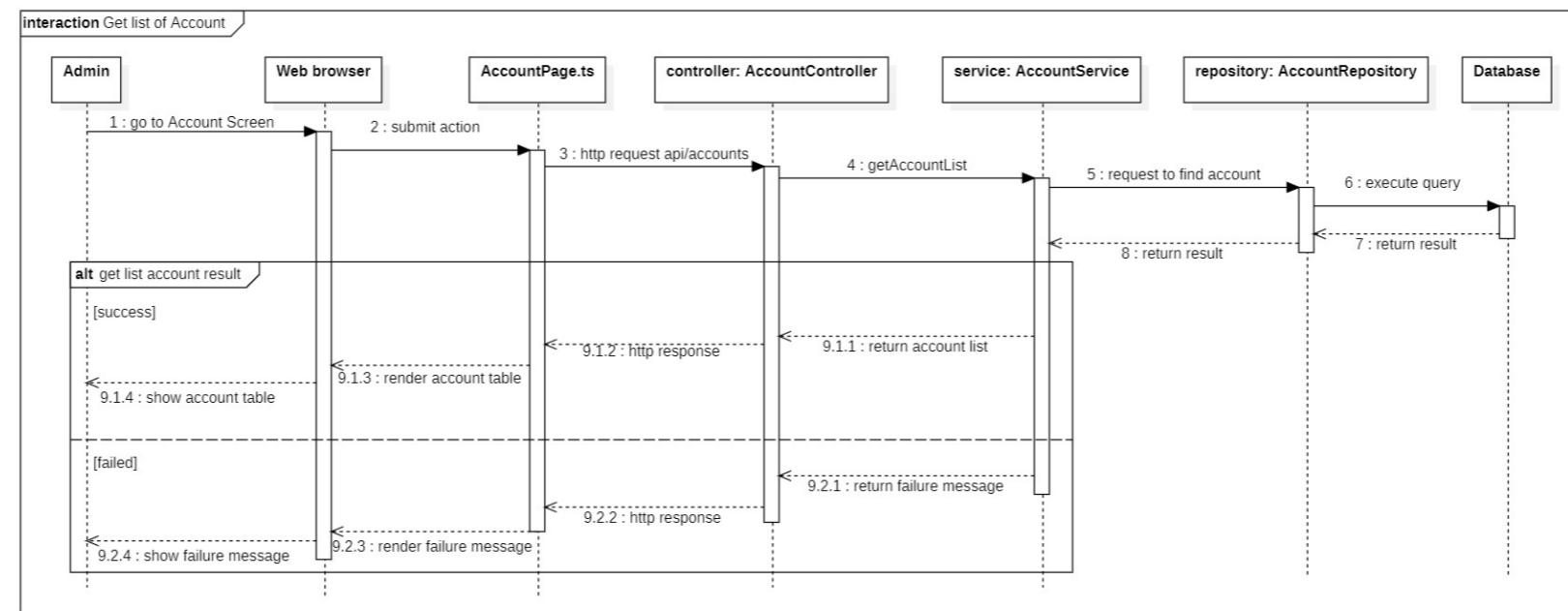


Figure 25 - <Sequence Diagram> Get list of account

4. Get list of Scrap

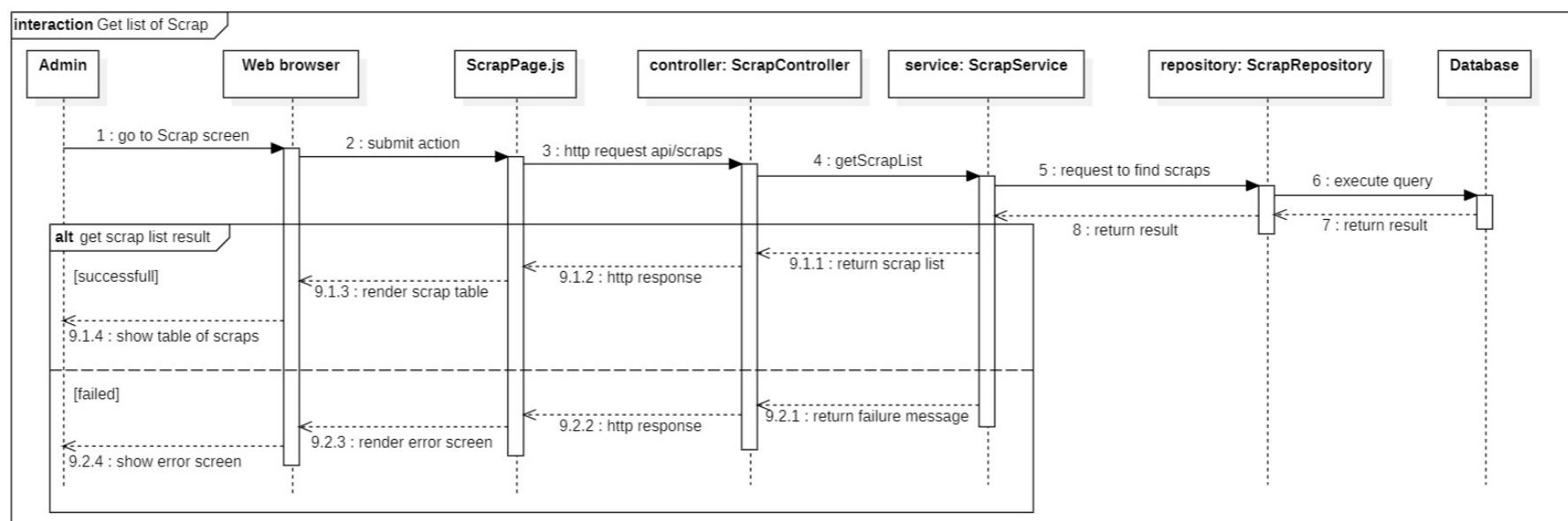


Figure 26 - <Sequence Diagram> Get list of scrap

5. Post scrap

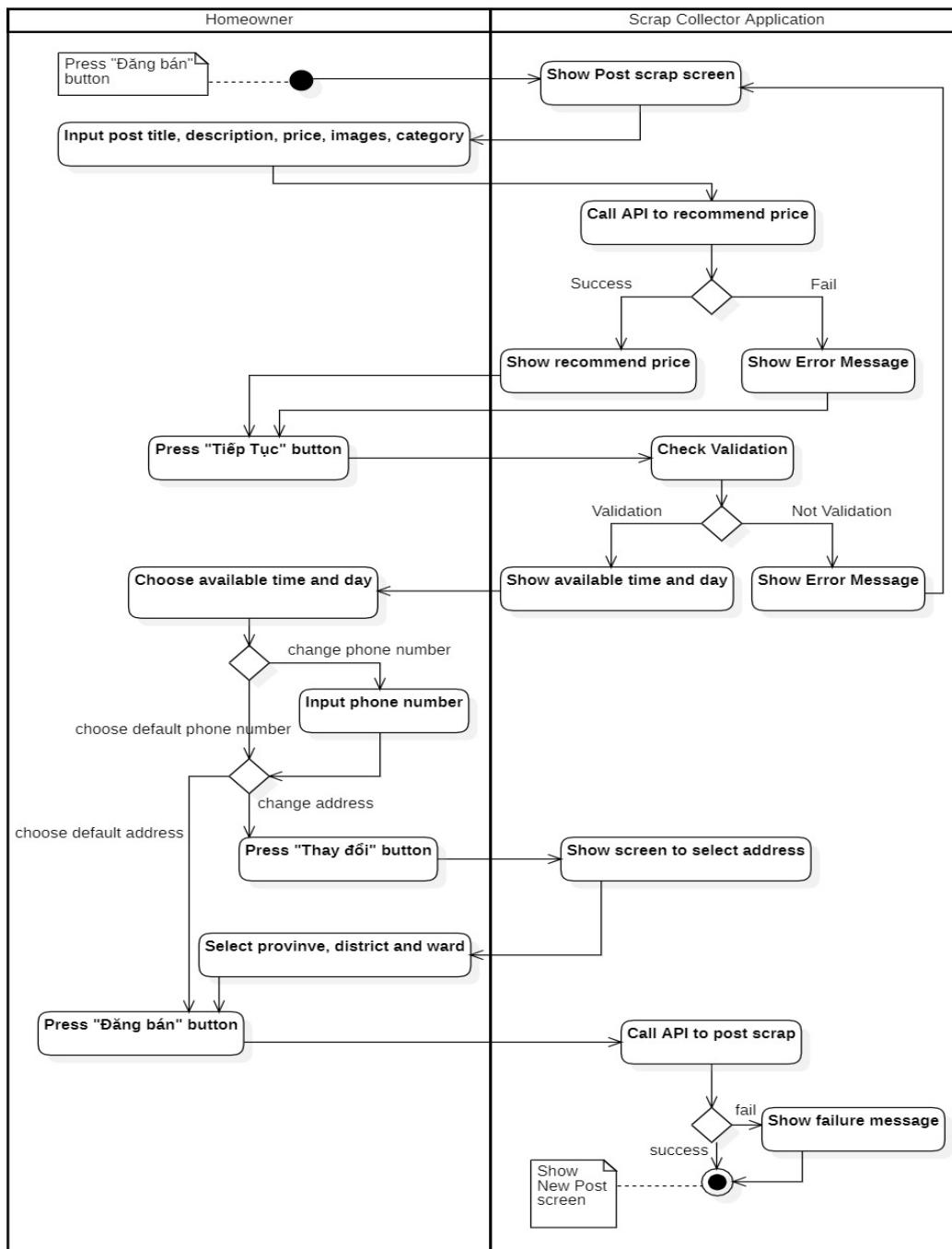


Figure 27 - <Activity Diagram> Post scrap

6. View booking request

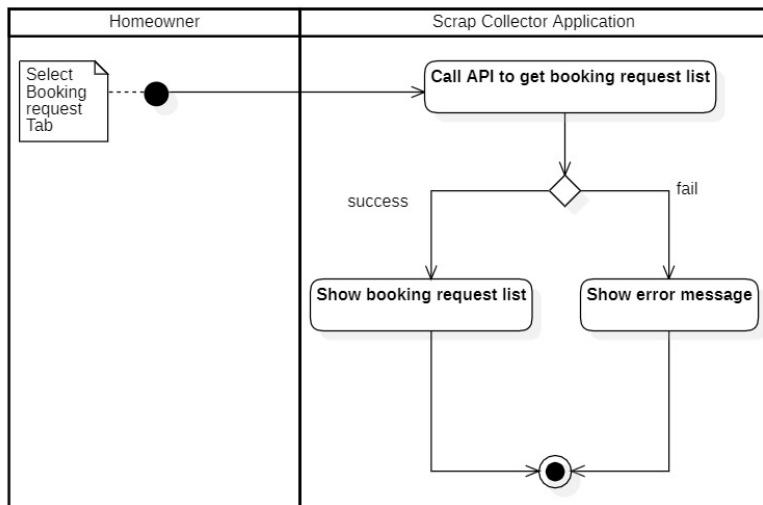


Figure 28 - <Activity Diagram> View booking request

7. Book Scrap

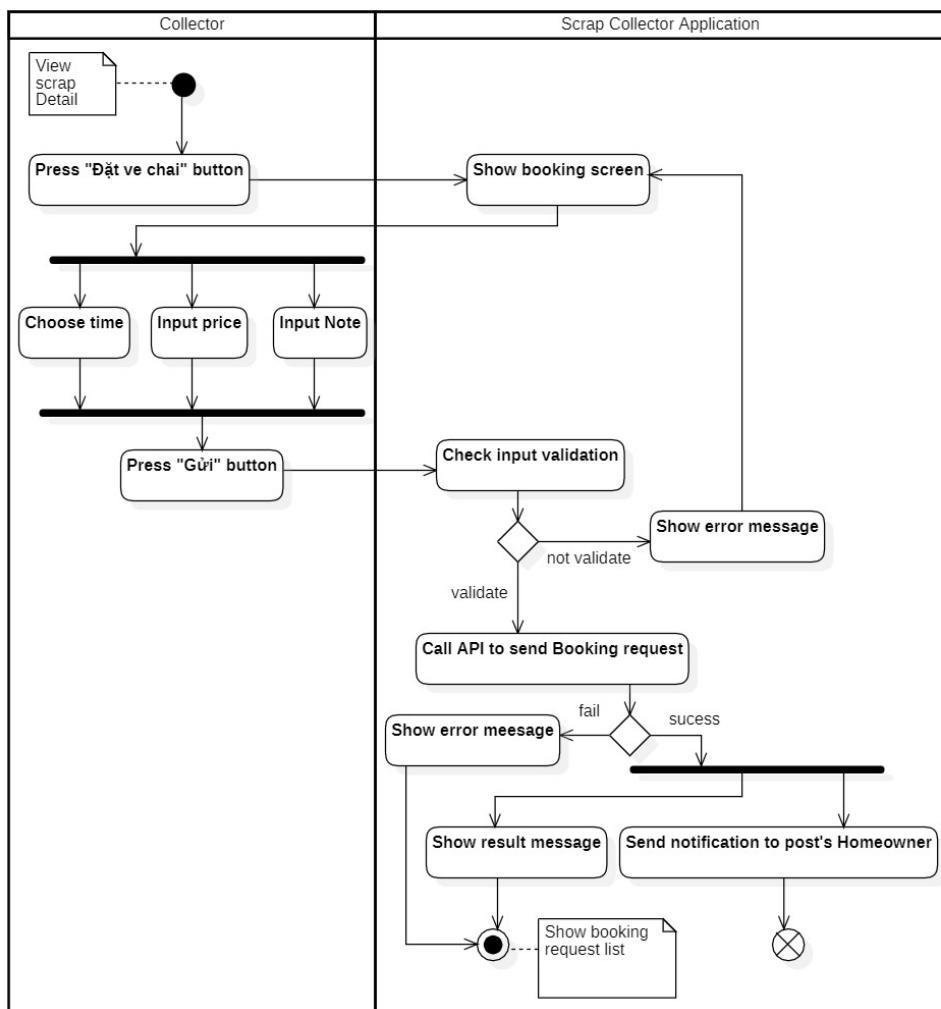


Figure 29 - <Activity Diagram> Book scrap

8. Search near-by scrap

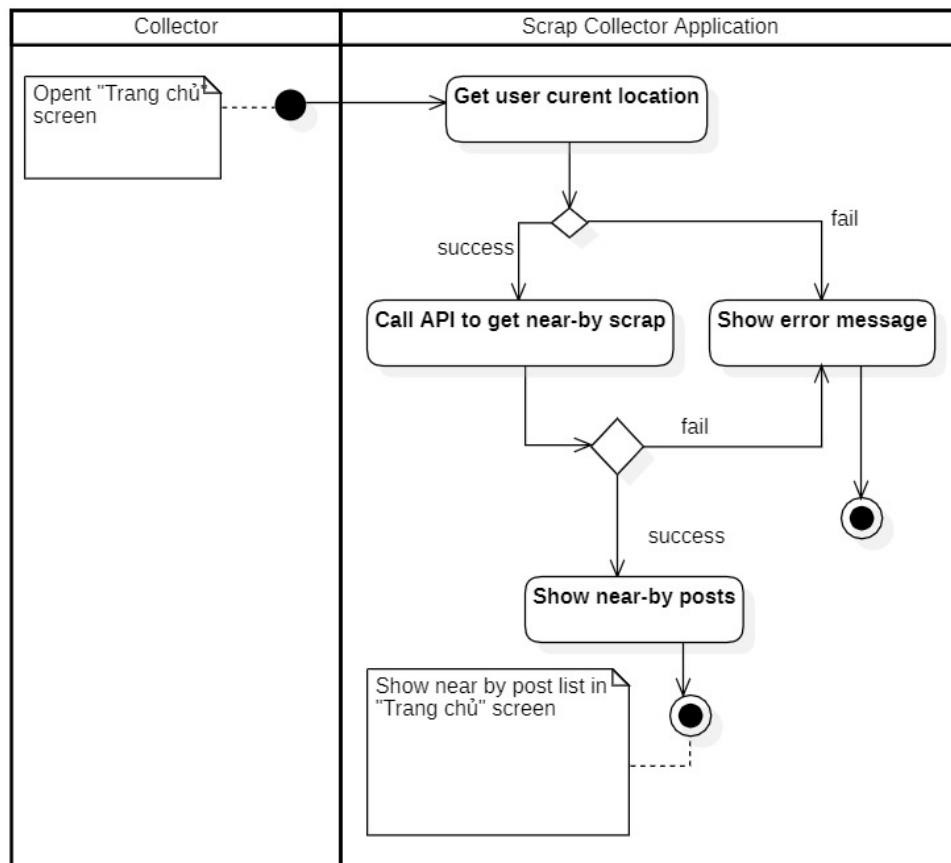


Figure 30 - <Activity Diagram> Search near-by scrap

9. View schedule

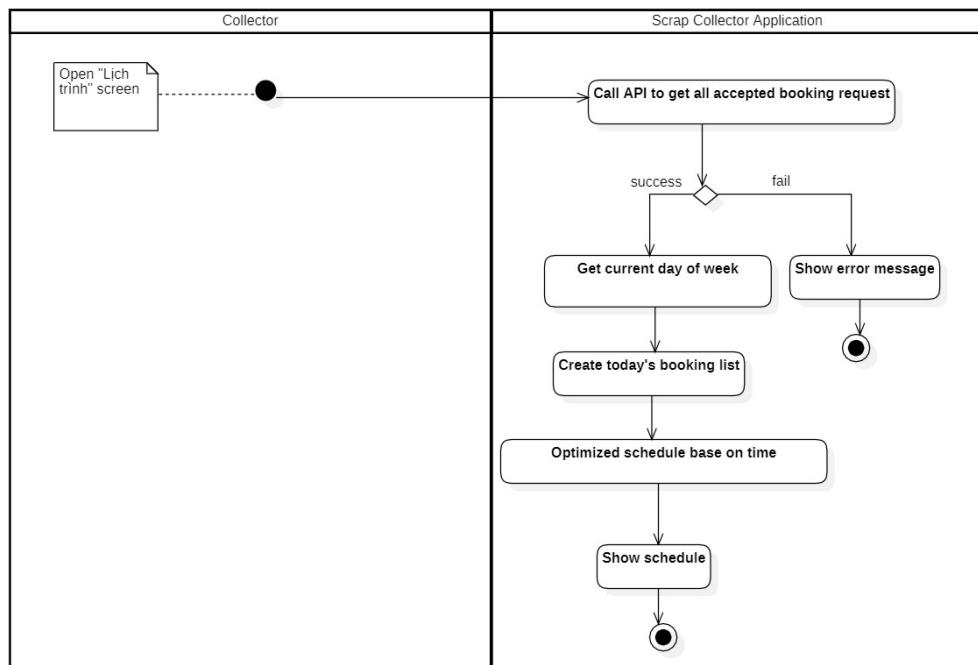


Figure 31 - <Activity Diagram> View schedule

10. Review Collector

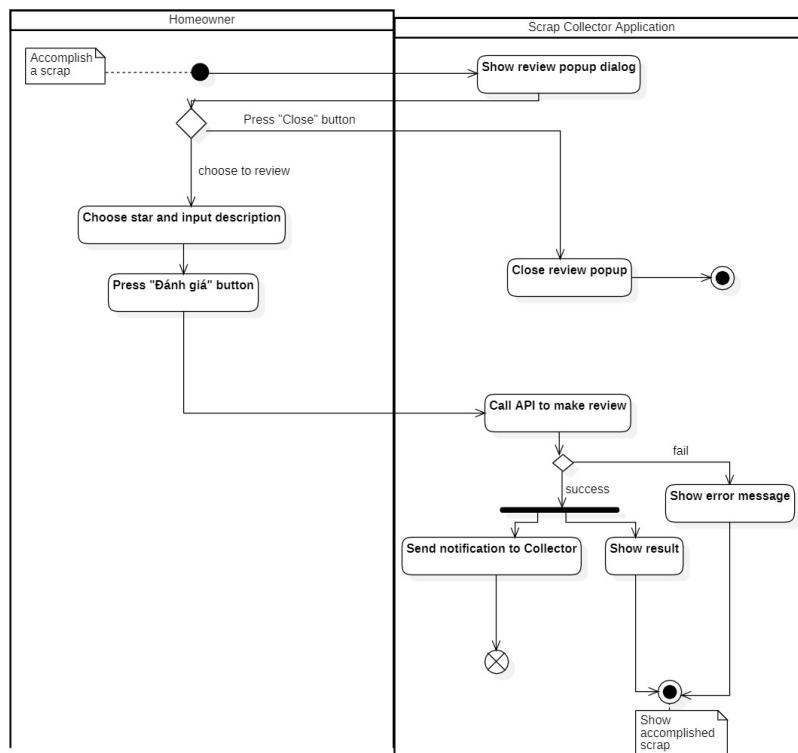


Figure 32 - <Activity Diagram> Review Collector

11. Scrap State machine Diagram

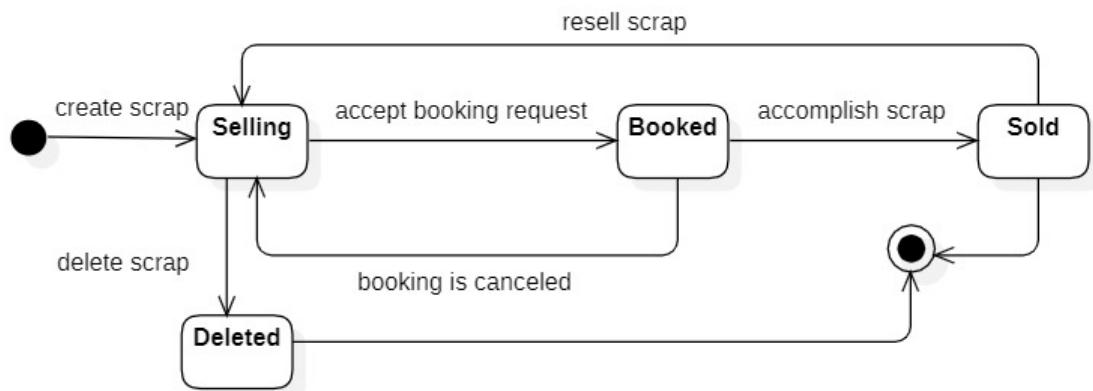


Figure 33 – Scrap State machine diagram

12. Booking State machine Diagram

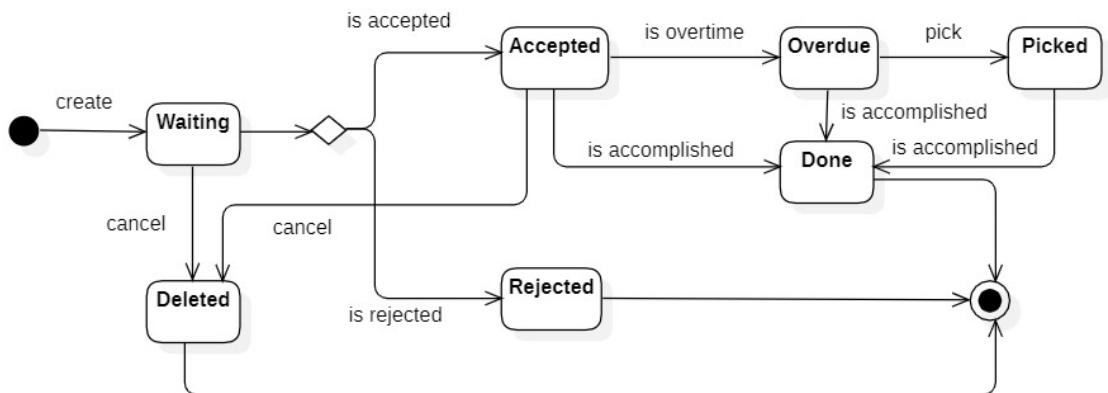


Figure 34 – Booking State machine diagram

L. Physical Diagram – Relationship Diagram

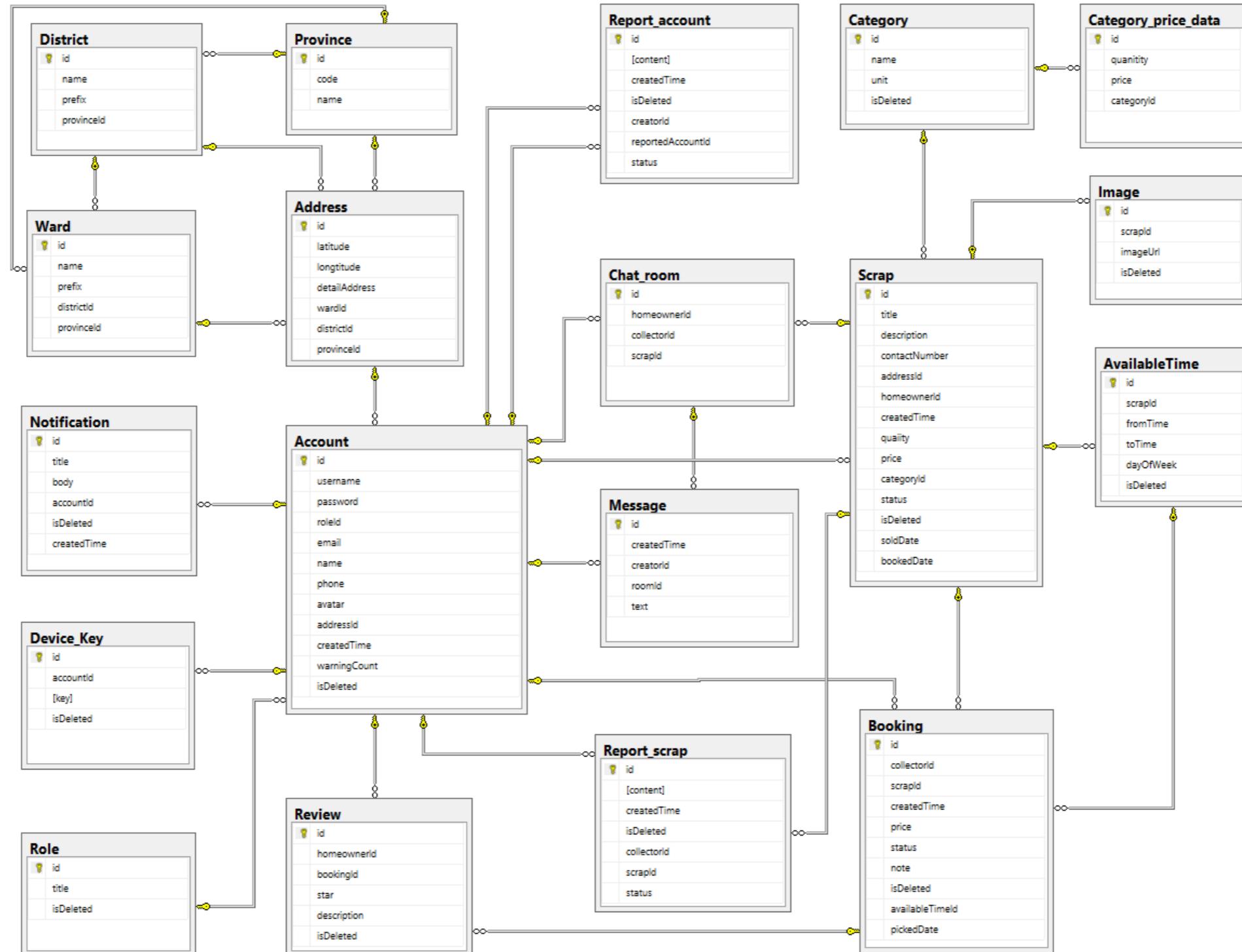


Figure 35 Physical Diagram

Data Dictionary

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 to which the user log in. The device key is stored for Firebase Cloud Message.
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
15	chat_room	Contain chat room information
16	category_price_data	Contain data for Linear Regression Algorithm
17	report_scrap	Contain scrap report information
18	report_account	Contain account report information

Table 22 - 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
	warningCount	Number of warning report of that account	int	Y
	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 province of address	int	N
	districtId	Id of district of address	int	N
	wardId	Id of ward of address	int	N
	latitude	Latitude of address	double	N
province	longitude	Longitude of address	double	N
	id	Unique id of province	int	N
	code	Code of province	varchar(20)	N
district	name	Name of province	varchar(50)	N
	id	Unique id of district	int	N
	provinceId	Id of province	int	N
	prefix	Prefix name of district	varchar(50)	N
ward	name	Name of district	varchar(50)	N
	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
	creatorId	Id of creator	varchar(255)	N
	roomId	Id of chat room that the message belongs to	int	N
	text	Content of message	varchar(255)	N
	createdTime	Time of creation of message	datetime	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
	bookedDate	Time at which scrap is booked	datetime	N
	soldDate	Time at which scrap is sold	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	time	N
	toTime	To time	time	N
	dayOfWeek	Day of week of available time	int	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
	pickedDate	Time when collector pick up scrap	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
chat_room	id	Unique id of each chat room	int	N
	homeownerId	Id of Homeowner	varchar(255)	N
	collectorId	Id of Collector	varchar(255)	N
	scrapId	Id of scrap belonging to that chat room	varchar(255)	N
category_price_data	id	Unique id of each data	int	N
	quantity	Quantity of scrap	int	N
	price	Price of scrap	int	N
	categoryId	Id of category of scrap	int	N
report_scrap	id	Id of report	int	N
	content	Content of report	varchar(MAX)	N
	collectorId	Id of collector who created report	varchar(255)	N
	scrapId	Id of scrap which is reported	varchar(255)	N
	createdTime	Time of creating report	datetime	N
	status	Status of report	boolean	N
	isDeleted	Check if report is deleted or not	boolean	N
report_account	id	Id of report	int	N
	content	Content of report	varchar(MAX)	N

	creatorId	Id of account who created report	varchar(255)	N
	reportedAccountID	Id of account which is reported	varchar(255)	N
	createdTime	Time of creating report	datetime	N
	status	Status of report	boolean	N
	isDeleted	Check if report is deleted or not	boolean	N

Table 23 - Physical Diagram Attribute Data Dictionary

M. Framework Architecture Diagram

1. React Redux Framework Architecture

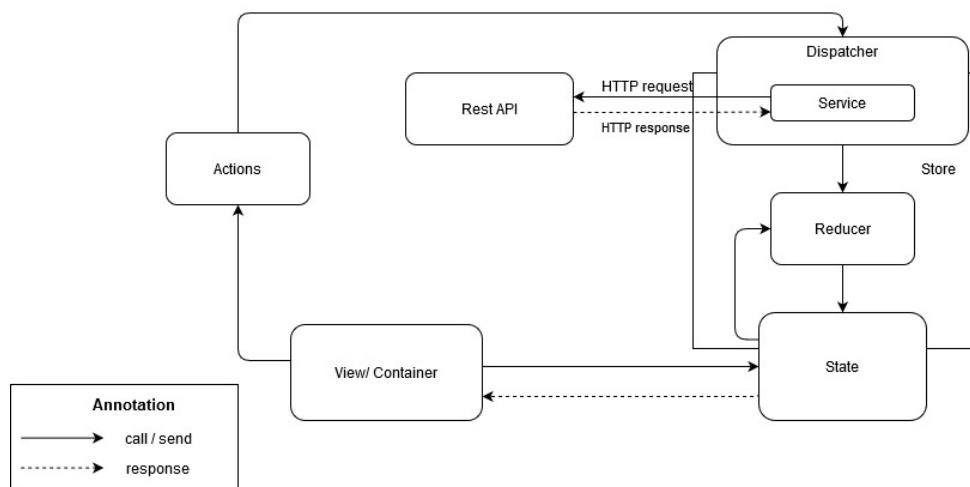


Figure 36 React Redux Framework Architecture

2. Angular Framework Architecture

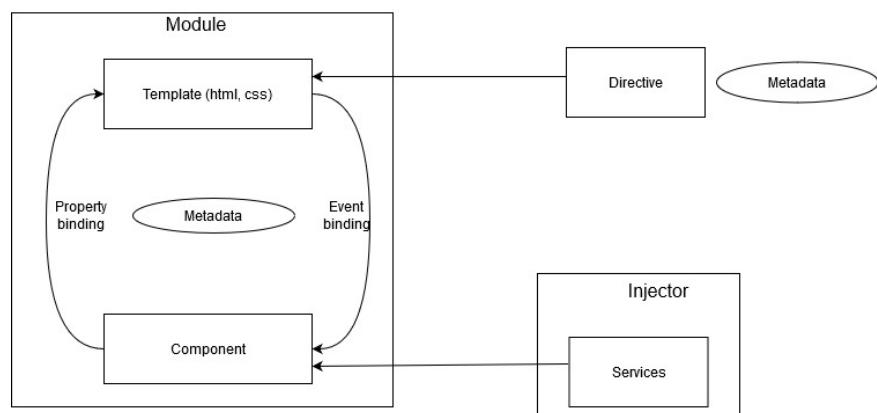


Figure 37 Angular Framework Architecture

N. Algorithm

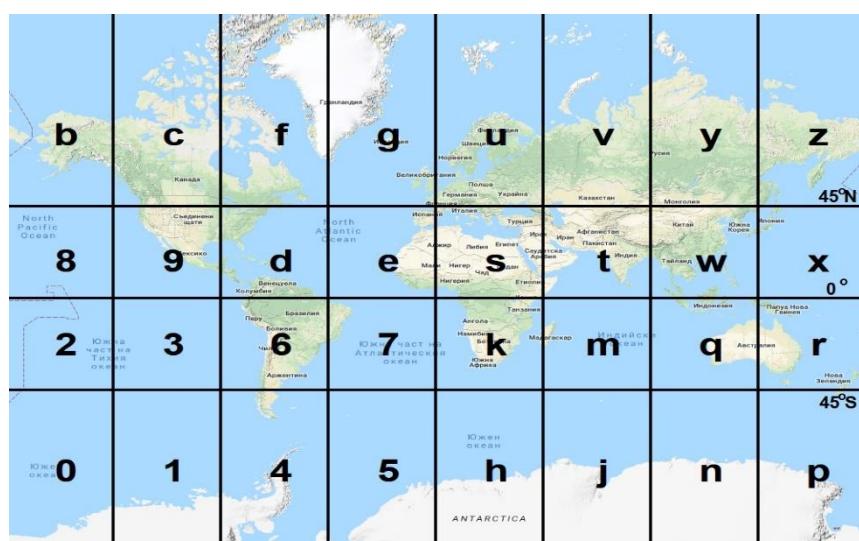
1 Search nearby Scrap implementing Geohash (Proximity Searching)

1.1 Definition

Geohash uses Base-32 alphabet encoding (characters can be from 0 to 9 and from b to z, except "a", "i", "l" and "o") to encode latitude and longitude of a coordinate.

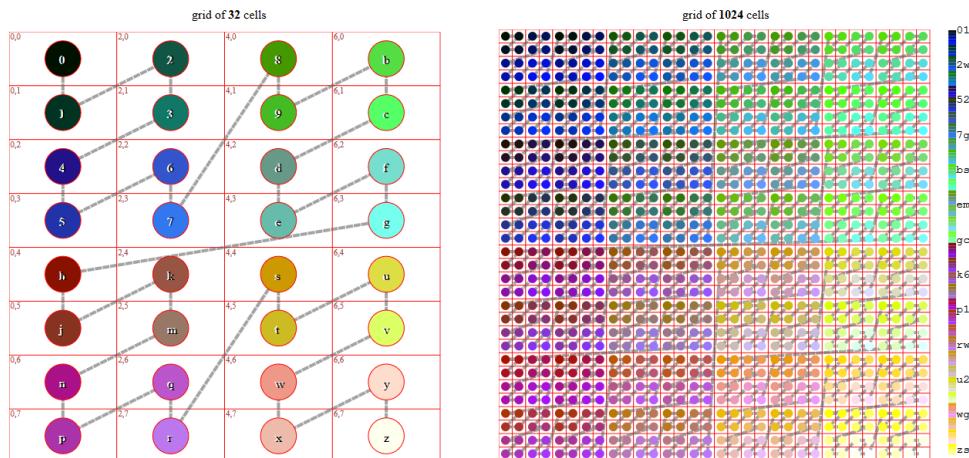
Imagine the world is divided into a grid with 32 cells. The first character in a geohash identifies the initial location as one of the 32 cells. This cell will also contain 32 cells, and each one of these will contain 32 cells (and so on repeatedly). Adding characters to the geohash sub-divides a cell, effectively zooming in to a more detailed area.

For example, the geohash code for the coordinate of *FPT University (10.84205101, 106.80930885)* is **w3gvwt8pt2c1**, the geohash code for the coordinate of *7elevent store in FPT University (10.841523, 106.810463)* is **w3gvwt9h2nme**.



Reference: <https://petrov.free.bg/academic/publication/geohash-eas-modified-geohash-geocoding-system-equal-area-spaces/>

Figure 38 - Initial Earth map division into 32 spaces using the original Geohash system



Reference: <https://en.wikipedia.org/wiki/Geohash>

Figure 39 – How Geohash works

1.2 Define problem

For those collectors who buy scrap, the most important thing is how they can know which scraps are for sale in their living place, may be about 5km or 10km, because scrap collectors do not want to go too far just for buying scrap.

Moreover, with traditional searching algorithms, the Scrap Collector system has to get all for-sale-scrap, then the system has to check if the distance from collector's current positon to scrap's position is suitable then creates a list of scrap to be shown to the collector. The longer time the system runs, the greater amount of scrap, and the longer it takes the collector to find nearby scraps.

1.3 Solution

Step 1: Encode latitude and longitude to geohash code

A geohash is a series of bits that repeatedly bisects a search space of latitudes and longitudes. The first bit bisects the longitude, the next one latitude, the next longitude... which means that the even bits are taken for the longitude code and the odd bits are taken for the latitude code.

Pseudocode:

Input the latitude and longitude of scrap

Input the number of bit of result

Initialize minimum latitude as -90 and maximum latitude as 90

Initialize minimum longitude as -180 and maximum longitude as 180

Initialize result as 0

Initialize counter as 0

While counter is less than the number of bit of result

If the counter is an *even* number

Calculate the middle point = $(\min \text{ longitude} + \max \text{ longitude}) / 2$

If inputted *longitude* less than the middle point, push a 0 bit to result and reduce the max longitude to the middle point, otherwise, push a 1 bit to the result and increase the min longitude to the middle point

Increase counter by 1

Otherwise (the counter is an *odd* number)

Calculate the middle point = $(\min \text{ latitude} + \max \text{ latitude}) / 2$

If inputted *latitude* less than the middle point, push a 0 bit to result and reduce the max latitude to the middle point, otherwise, push a 1 bit to the result and increase the min latitude to the middle point

Increase counter by 1

Convert geohash from string of bits to base32 encode

Return the geohash code

Step 2: Search and get near-by scraps based on geohash code

After the coordinate of scrap is encoded in geohash format, we encode the coordinate of current position of the collector, then do the compare to check if that scrap is in collector region or not. As a consequence of geohashing, nearby places often present similar prefixes. The longer a shared prefix is, the closer the two places are.

Pseudocode:

Input the latitude and longitude of collector's current position and the radius distance

Check the inputted radius distance value, if value is not null, get radius value, otherwise, get the default configure radius value

Get the id of scrap, latitude and longitude of scraps with "SELLING" status.

For each scrap in the scraps result list, hash scrap's position to geohash.

Get all 8 neighbors (north, northern east, east, southern east, south, southern west, west and northern west) around collector's position based on radius.

Based on the set of neighbors, create a set of ranges to search scrap, each range contains *lower range geohash* and *upper range geohash*.

Initialize the array of scraps as empty array.

For each range in the set of ranges

Initialize the result as empty array

Search for scrap that has position's geohash between the *lower range geohash* and *upper range geohash*.

If the result array is not null

Push the result array into array of scraps

For each result in the array of scraps, get detail information from database based on the id of scrap.

Return the array of nearby scraps.

1.4 Complexity

For encoding n positions to geohash coding by geohash, the complexity is $O(\log(n))$.

For searching near-by scrap based on hash code, with Binary search algorithm, in average and worst case, the complexity is $O(\log(n))$; in the best case, because we have build a hashmap based on geohash code for scrap's position, the system can access data directly, so the time complexity in the best case is $O(1)$.

1.5 Flowchart

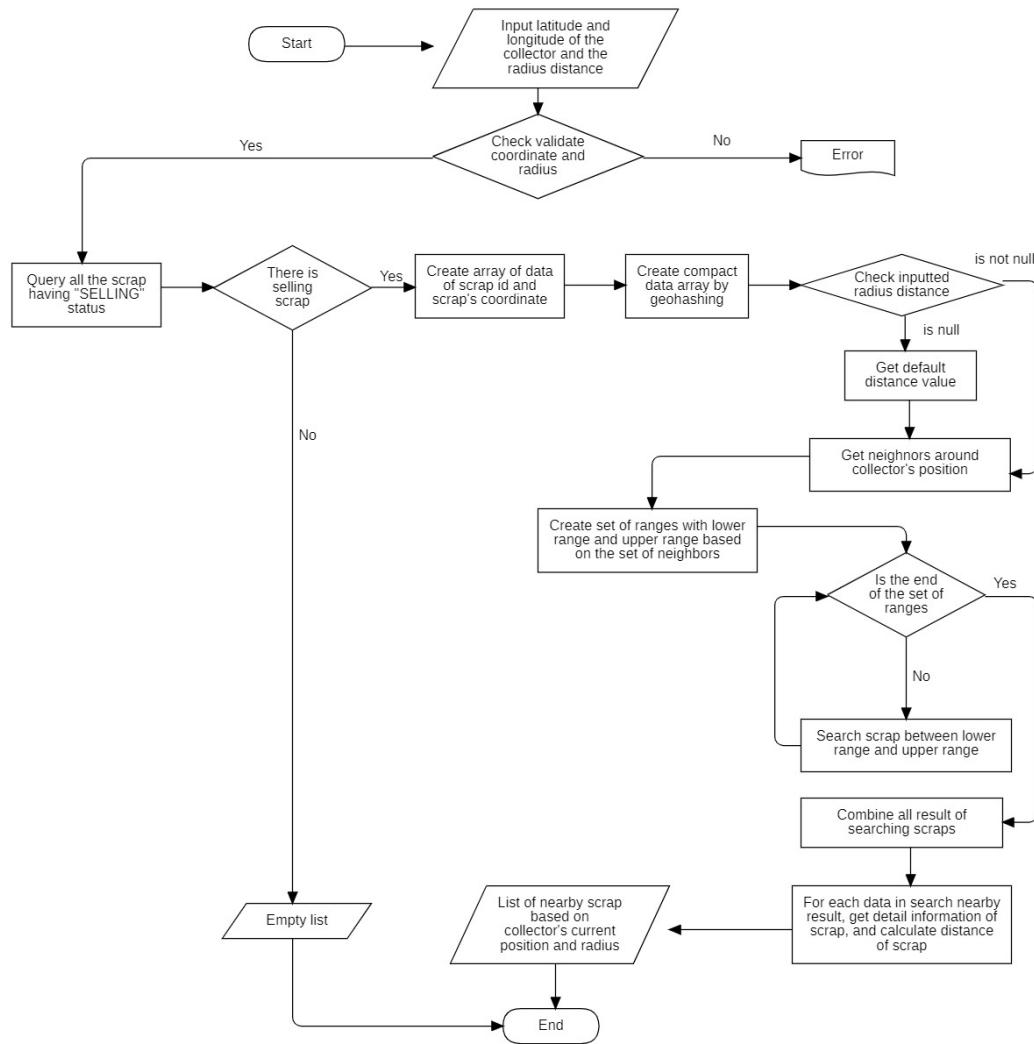


Figure 40 – Flowchart – Search nearby scrap implementing Geohash

2. Best price recommendation by Linear Regression

2.1 Definition

Linear regression is a linear approach to modeling the relationship between a *dependent variable* (also called *scalar response*) and one or more *independent variables* (or *explanatory variables*). In *Linear Regression*, the relationships are modeled by *linear predictor functions* in which unknown model *parameters* are estimated from real data collected by researches.

One of the most popular practical uses of Linear Regression is *prediction*. Linear Regression can be used to fit a predictive model for an observed data set of values of the dependent variable (the response) and independent variables (explanatory

variables). After developing a linear model, it can use that model to make prediction of the response from inputted explanatory variables.

2.2 Define problem

For the Homeowner who wants to post scrap for selling, they usually don't know about price of scrap, so they find it very hard to set the price of scrap. Therefore, by implementing Linear Regression, we develop a Best Price Recommendation for Homeowner so that they can sell scrap at the most suitable price.

2.3 Solution

In fitting a linear regression model, it is much more convenient to express the mathematical operations using **matrix notation**. Suppose that there are k independent variables and n observations, the model relating the independent variables to the response is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} \quad i = 1, 2, \dots, n$$

This model of n equations can be expressed in matrix notation as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$$

Where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

According to the least squares estimate of $\boldsymbol{\beta}$, we have the equation:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' \mathbf{y}$$

To implement in our Best Price Recommendation, there are two parameters including the mass of Scrap and the price of Scrap. The collected data helps form linear equation in which the independent variable is *mass of Scrap* and the dependent variable is *price of Scrap*.

The mass of Scrap and the price of Scrap of specific Category are collected from bookings that are accomplished and from observations.

=> Hence, for each category of scrap in system, there is an equation:

$$Y = \beta_0 + \beta_1 X$$

Y: the recommended price

X: the mass or quantity of scrap

β_0 : the slope of the linear equation, depicting the increasing level of Y where X decreases by 1 unit.

β_1 : depicts the estimated value of Y if the value of X = 0, which means the value of Y does not depend on the value of X

Step 1: Define 3 matrices: X, Y and β . In which, n is the number of observations. X is a $(n \times 2)$ matrix, Y is a $(n \times 1)$ matrix and β is a (2×1) matrix

$$X = \begin{bmatrix} 1 & x_{11} \\ 1 & x_{21} \\ 1 & x_{31} \\ \dots & \dots \\ 1 & x_{n1} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_n \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

Step 2: Define the transpose of matrix X, call X' . Then we have X' , a $(2 \times n)$ matrix

$$X = \begin{bmatrix} 1 & x_{11} \\ 1 & x_{21} \\ 1 & x_{31} \\ \dots & \dots \\ 1 & x_{n1} \end{bmatrix} \Rightarrow X' = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_{11} & x_{21} & x_{31} & \dots & x_{n1} \end{bmatrix}$$

Step 3: Define matrix $X'X$ by multiply matrix transpose X' with matrix X. Then we have $X'X$, a (2×2) square matrix.

Step 4: Invert the matrix $X'X$, then we have $(X'X)^{-1}$, a (2×2) square matrix.

Step 5: Multiply the inversion matrix $(X'X)^{-1}$ with matrix X' , we have $(X'X)^{-1}X'$, a $(2 \times n)$ matrix

Step 6: Multiply the result $(X'X)^{-1}X'$ in step 5 with matrix Y, we have $(X'X)^{-1}X'Y$, a (2×1) matrix

Step 7: result is a matrix has 1 column and 2 rows. 2 items in matrix is β_0 and β_1 .

2.4 Complexity

Declare n the number of observations and p the number of weights.

The problem of finding the vector of weights β in a linear regression boils down to evaluating the following equation:

$$\beta = (X'X)^{-1}X'Y$$

X is a $(n \times p)$ matrix, and X' is a $(p \times n)$ matrix

The most computationally intensive part is to evaluate the product $X'X$, which is done in $p * n * p = p^2n$ operations. After that, we invert the result of $X'X$, which is a $(p \times p)$ square matrix, and the inversion is finished in p^3 operations.

Though most implementations prefer to use a gradient descent to solve the system of equations $(X'X)\beta = X'Y$, the complexity remains the same.

In conclusion, the overall complexity of “Best price recommendation by Linear Regression” should be $p^2n + p^3$.

2.5 Example

For scrap of crude iron category, we have a table of data as below:

Mass of Scrap (X)	Price of Scrap (Y)
0.5 kg	7,000
0.3 kg	5,000
1 kg	15,000
1 kg	16,000
0.8 kg	12,000

The equation we have to find is: $Y = \beta_0 + \beta_1 * X$, in which, Y is the recommended price and X is the mass of scrap

$$\begin{aligned} \beta &= (X'X)^{-1} X' Y = \begin{bmatrix} 0.6 & 0.97 & -0.32 & -0.32 & 0.051 \\ -0.56 & -1.08 & 0.72 & 0.72 & 0.206 \end{bmatrix} * \begin{bmatrix} 7000 \\ 5000 \\ 15000 \\ 16000 \\ 12000 \end{bmatrix} \\ &= \begin{bmatrix} -134.02 \\ 15463.91 \end{bmatrix} \end{aligned}$$

Which means that $\beta_0 = -134.02$ and $\beta_1 = 15463.91$

Then, we have equation for recommended price of scrap belonging to crude iron category:

$$\text{recommended price} = -134.02 + 15463.91 * \text{mass of scrap} \quad (2)$$

If we want to know what price is suitable for selling 2 kilogram of crude iron, just set the mass of scrap as 2 in equation (2), then we can know the recommended

price is: 30,793 VND. Then the system will round up the price and suggest the Homeowner to sell scrap at price of 31,000 VND.

2.6 Flowchart

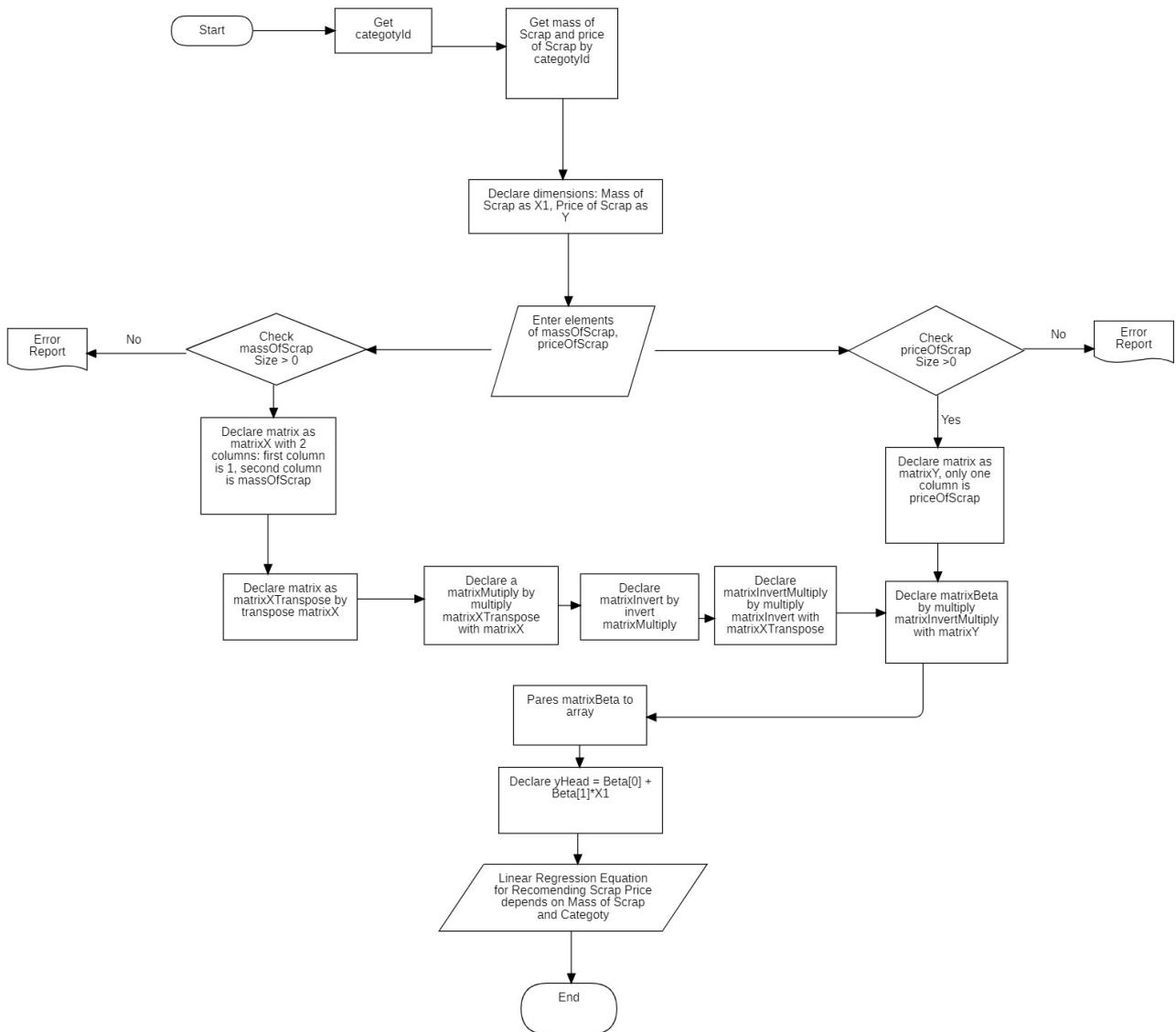


Figure 41 – Flowchart - Recommendation Scrap Price by Linear Regression

O. Future plan

Current system aims to sell scrap only, with current architecture of mobile application, we can extend the scale of system to sell things such as old technical devices, old clothes...

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

- Develop the application to make it more flexible and can be run on low-platform device
- Implement “Chinese Postman Problem” algorithm to help Collector to find the shortest path to visit and pick all scraps in a day
- 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