

<Second-hand EV & Battery Listing Platform>

Documentation

Members:

Full name ID

Tieu Minh Duy SE161534

Pham Nguyen Quang Minh SE184082

Vu Le Truong Thinh SE193129

Huynh Dang Khoa SE193040

FUHCM, SUMMER 2025

TABLE OF CONTENTS

[1](#_Toc200703269)  [Introduction](#_Toc200703269)  [3](#_Toc200703269)

[1.1](#_Toc200703270)  [Overview](#_Toc200703270)  [3](#_Toc200703270)

[1.2](#_Toc200703271)  [Scope](#_Toc200703271)  [3](#_Toc200703271)

[1.3](#_Toc200703272)  [References](#_Toc200703272)  [3](#_Toc200703272)

[2](#_Toc200703273)  [Project scope](#_Toc200703273)  [4](#_Toc200703273)

[3](#_Toc200703274)  [Scrum Process](#_Toc200703274)  [5](#_Toc200703274)

[4](#_Toc200703275)  [Functional Requirements](#_Toc200703275)  [6](#_Toc200703275)

[4.1](#_Toc200703276)  [Roles](#_Toc200703276)  [6](#_Toc200703276)

[4.2](#_Toc200703277)  [Use Cases Diagram](#_Toc200703277)  [6](#_Toc200703277)

[5](#_Toc200703278)  [NON-FUNCTIONAL Requirements](#_Toc200703278)  [7](#_Toc200703278)

[5.1](#_Toc200703279)  [Usability](#_Toc200703279)  [7](#_Toc200703279)

[5.2](#_Toc200703280)  [Reliability](#_Toc200703280)  [7](#_Toc200703280)

[5.3](#_Toc200703281)  [Performance](#_Toc200703281)  [7](#_Toc200703281)

[5.4](#_Toc200703282)  [Supportability](#_Toc200703282)  [7](#_Toc200703282)

[5.5](#_Toc200703283)  […](#_Toc200703283)  [7](#_Toc200703283)

[6](#_Toc200703284)  [Screen Flow](#_Toc200703284)  [8](#_Toc200703284)

[7](#_Toc200703285)  [Technologies](#_Toc200703285)  [9](#_Toc200703285)

[8](#_Toc200703286)  [database design](#_Toc200703286)  [10](#_Toc200703286)

[8.1](#_Toc200703287)  [[ERD]](#_Toc200703287)  [10](#_Toc200703287)

[8.2](#_Toc200703288)  [[Database diagram]](#_Toc200703288)  [10](#_Toc200703288)

[9](#_Toc200703289)  [source control](#_Toc200703289)  [11](#_Toc200703289)

# Introduction

## Overview

The project builds an online classifieds platform for buying and selling used electric vehicles and batteries. Sellers can post ads with detailed information, images and product prices, while buyers can view and contact directly. The system applies a subscription model: Free accounts can only post one ad, Basic, Premium and VIP packages allow more ads, unlimited renewals and higher display priority. The platform also integrates a review feature for buyers to rate sellers, a report to report violations to help admins/mods handle, along with payment to record and manage transaction history, ensuring a safe, transparent and easy-to-use environment for both buyers and sellers.

## Scope

* **Objects of use** : Guest (visitor), User (buyer/seller), Admin/Moderator.
* **Main function** :
  + Register/Login, account management.
  + Post an ad to sell electric cars or batteries, browse ads.
  + View, search and contact sellers.
  + Buy subscription package (Free, Basic, Premium, VIP).
  + Manage payments and transaction history.
  + Review sellers, report violations.
  + Admin manages users, approves news, and processes reports.
* **Out of scope** : No direct payment processing between buyer and seller (only subscription fee collection), no shipping or product warranty.

## References

Good Market Website

# Project scope

**This project is to build a specialized Web Classifieds Platform for buying and selling used Electric Vehicles (EV) and Batteries, operating on the Subscription model for sellers.**

**1. User Functions:**

**o Buy/View: Search, view listings, and contact sellers (after logging in).**

**o Sell/Post: Post, edit, and manage classifieds (Batteries or Vehicles).**

**o Package Payment: Buy and manage Subscription packages (Free, Basic, Premium) via the Payment Gateway.**

**o Interaction: Review and Report listings.**

**2. Administration Functions:**

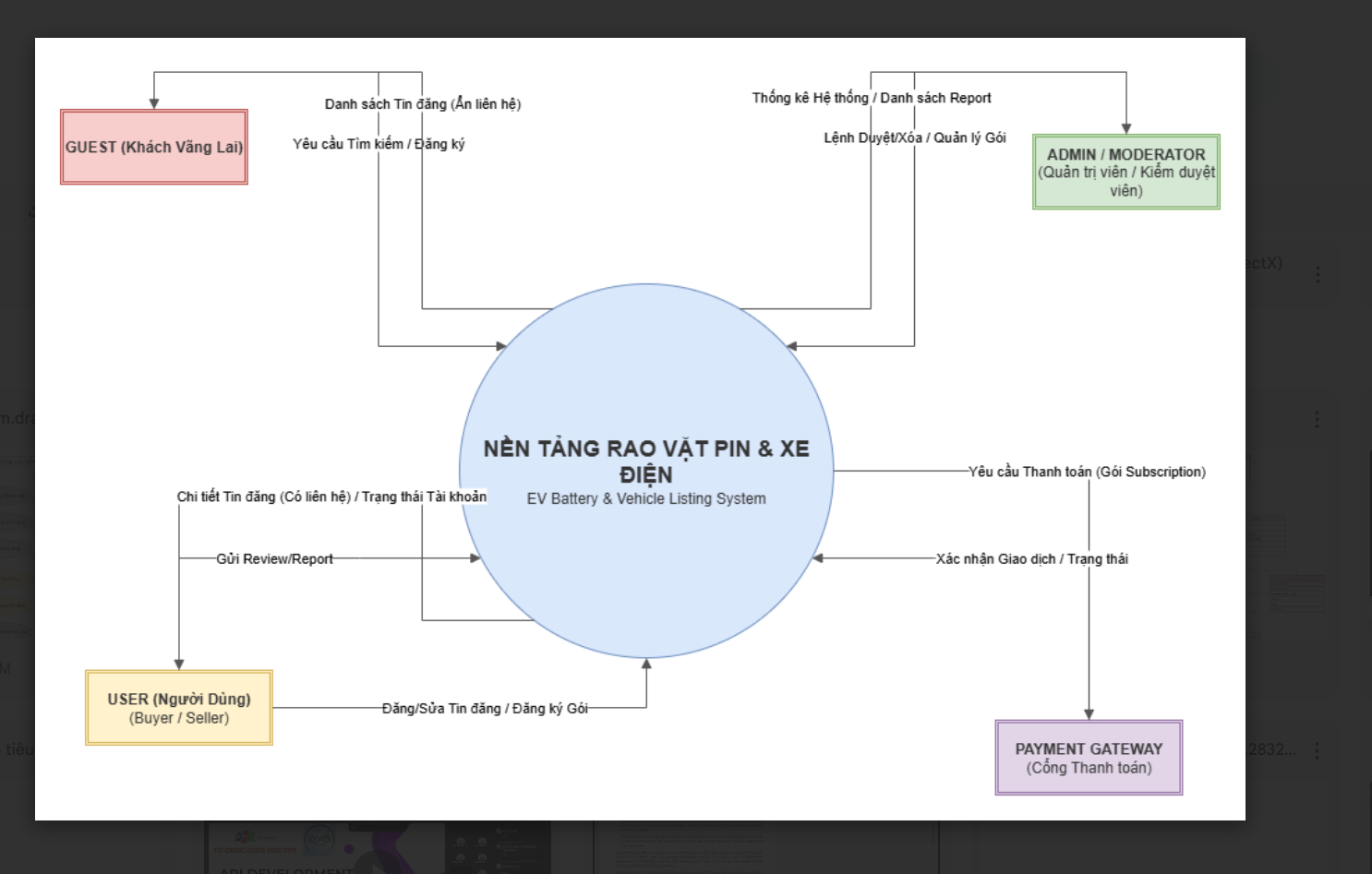
**o Browse and moderate all listings and reports.**

**o Manage User Accounts and Service Packages (Add, Edit, Delete).**

**o View basic system statistics.**

Context Diagram

Link: [Context diagram link](https://drive.google.com/file/d/1L4UAm6bSHj4ciR3YlpkhBRIJSBCc_06C/view?usp=sharing)



# Scrum Process

The development project for the **Second-hand EV & Battery Listing Platform** will adopt the **Agile Scrum framework**. Scrum is a lightweight process management framework designed to develop complex products in a rapidly changing environment.

## CORE PRINCIPLES

1. **Iterative and Incremental:** Work is divided into short, repeating cycles called **Sprints** (typically 1-4 weeks long). At the end of each Sprint, the team delivers a potentially shippable increment of the product.
2. **Transparency:** Every aspect of the process must be visible and understandable to everyone involved.
3. **Inspection and Adaptation:** The team frequently inspects the progress and quality of the product to adjust the course as necessary.

## KEY COMPONENTS IN THE PROJECT

* **Goal:** To maximize the value of the product delivered and minimize risk through continuous inspection and adaptation.
* **Benefit:** Allows the development team to respond quickly to changes in market requirements or user feedback, ensuring the product best meets customer needs.

## SCRUM EVENTS

* **Product Backlog:** A prioritized list of all the features, requirements, enhancements, and fixes needed for the product.
* **Sprint Planning:** A meeting held at the beginning of the Sprint to select items from the Product Backlog and create a **Sprint Goal**.
* **Daily Scrum (Daily Standup):** A short, 15-minute daily meeting for the Development Team to synchronize activities and plan work for the next 24 hours.
* **Sprint Review:** Held at the end of the Sprint, the team presents the results to stakeholders for feedback.
* **Sprint Retrospective:** A meeting at the end of the Sprint where the team inspects itself and creates a plan for improvements to be enacted during the next Sprint.

# FUNCTIONAL Requirements

## 4.1 Main Flows

### ****4.1 User (Buyer/Seller)****

* **Register/Login** : Users can create new accounts, log in with email/phone number and password.
* **Account Management** : Users can edit personal information, password, and account status.
* **Post for sale** :
  + Free User: can only post **01 news** .
  + Paid User (Subscription): can post multiple ads depending on the service package.
* **View and search news** : You can browse the news list, filter by category, price, and keyword.
* **Purchase Subscription Package** : User selects package, makes payment, and receives package benefits.
* **Review** : After a transaction or review, users can rate and comment on the seller.
* **Report** : Users can report violations for admin to handle.

### ****4.2 Guest (User not logged in)****

* **View listing** : Can browse and search listings, but **cannot see seller contact information** .
* **Register an account** : You can create an account to become a User.

### ****4.3 Admin/Moderator****

* **User Management** :
  + View list of users.
  + Ban/Unban the violating account.
* **Review posts** : Check and approve posts before publishing.
* **Report processing** : Receive reports from Users and decide to delete/hide violating news.
* **Subscription Management** : Add, edit, or delete service packages.
* **Payment Management** : View user payment history.

### ****4.4 Payment Gateway (external integration)****

* **Payment processing** : Receive package purchase requests from the system and return results (success/failure).
* **Confirm transaction** : Send transaction information to the system to save in the Payment table.

## Main Flows

### ****Main Flow 1: Posting (Post Listing)****

1. User logs into the system.
2. Current Subscription Package Check System:
   * Free: only allows to post a maximum of **01 news** .
   * Paid: multiple postings according to package benefits.
3. User enters information about the ad (title, description, price, image, vehicle/battery information).
4. The message is saved with the status **Pending** .
5. Admin/Moderator approves the post → changes to **Active** .
6. News displayed on the website according to the priority of the subscription package.

### ****Main Flow 2: Buy Subscription****

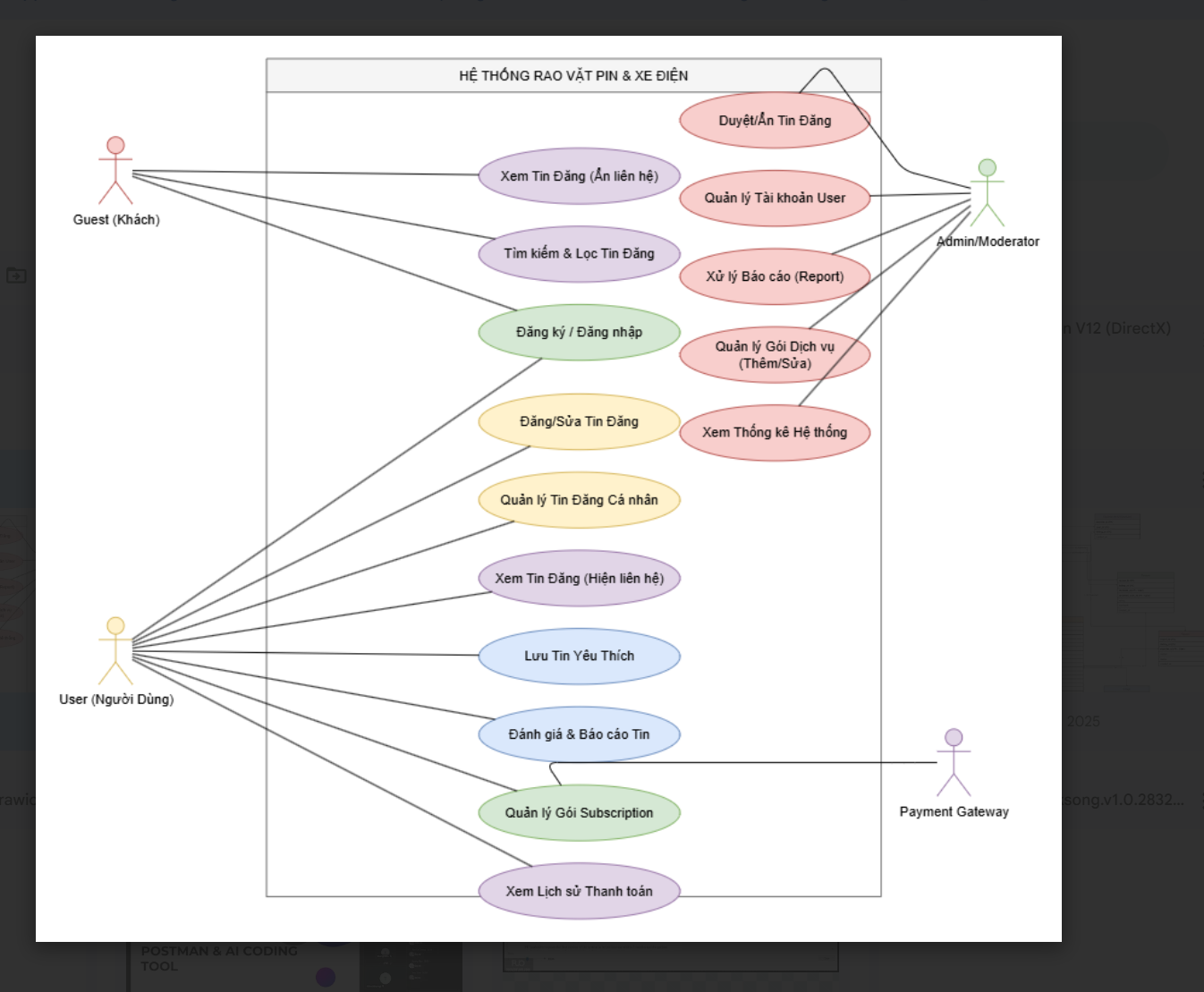
1. User selects Subscription package (Basic, Premium, VIP).
2. The system sends a payment request to **the Payment Gateway** .
3. After successful payment:
   * Create a new record in **User\_Subscription** (start\_date, end\_date).
   * Record the transaction in **Payment** .
4. The system updates User benefits according to the purchased package.

### ****Main Flow 3: Review & Report****

1. After transaction or reference, User can leave **Review** for seller (rating + comment).
2. Users can **report** a post if they detect a violation (fake news, wrong category, spam...).
3. The system saves the report and notifies the Admin/Moderator.
4. Admin/Moderator reviews and processes: can keep, hide or delete the post, and ban/unban the User if necessary.

## Use Case Diagram

link:[User case diagram](https://drive.google.com/file/d/12gpYOSVYP-UbjnC3RAwqGS8LiPYQySFJ/view?usp=sharing)



## Bussiness rule

## I.USER ACCOUNT AND ROLE RULES

1. **Unique Email:** Every user account must have a **unique** email address (User.user\_email).
2. **Password Requirement:** Passwords must be encrypted (User.password\_encoded) and must have a minimum length of **8 characters**, including uppercase letters, lowercase letters, and numbers.
3. **Default Role:** All newly registered users must be assigned the default role of **"User"** (or the equivalent role for buyers/sellers).
4. **Admin Access:** Only accounts with **Role.role\_name** as "Admin" or "Moderator" are authorized to access administrative pages.
5. **Status Restriction:** Accounts with the status status='Banned' are not allowed to log in or perform any transactions on the system.
6. **DOB Validation:** The date of birth (User.dob) must be in the past, and users must be **18 years or older** to register an account.

## II. LISTING AND CATEGORY RULES

1. **Listing Required Fields:** When submitting a new listing, the fields **title**, **description**, **price**, **brand**, and **category\_id** are mandatory (NOT NULL).
2. **Listing Price Range:** The sale price (Listing.price) must be a positive numeric value and must not exceed the maximum limit (e.g., $1,000,000,000).
3. **Battery Life Data:** For the Battery category, the fields **battery\_life\_original** and **battery\_life\_remaining** must be provided, and **remaining** must be less than or equal to **original**.
4. **Image Minimum:** Each listing must have a minimum of **1 linked image** (Image).
5. **Image Maximum:** Each listing is not allowed to have more than **5 images** (limiting the number of records in the Image table linked to Listing.listing\_id).
6. **Post Limit by Subscription:** The maximum number of active listings for a seller (Listing.seller\_id) must be determined by their current **Subscription** package. (e.g., Free package maximum 1 listing).
7. **Unique Listing Title:** Within the scope of a single seller, the listing title (Listing.title) must be **unique** (to avoid duplicate content).
8. **Approval Requirement:** New listings must have the status **status='Pending'** and must be changed to **status='Active'** by an Admin/Moderator before public display.
9. **Listing Status Consistency:** Only listings with the status **status='Active'** will be displayed on the homepage and search pages.

## III. SUBSCRIPTION AND PAYMENT RULES

1. **Subscription Status:** A user can only have one **User\_Subscription** package with the status **status='Active'** at any given time.
2. **Subscription Duration:** When a package is activated (User\_Subscription.start\_date), the system must automatically calculate the **end\_date** based on the **Subscription.duration** field of the selected package.
3. **Payment Precedence:** For a User\_Subscription record to be created, there must be a successfully linked **Payment** record (Payment.status = 'Completed').
4. **Refund Period:** Subscription package payments are not refundable after **7 days** from the payment date (Payment.payment\_date).
5. **Price Non-Negotiable:** The subscription price (Subscription.sub\_price) cannot be changed for an active **User\_Subscription** package.
6. **Expired Listing:** If the **User\_Subscription** package expires (end\_date < Current Date), all listings (Listing) exceeding the Free package limit must be automatically changed to the status **status='Inactive'** or **status='Expired'**.
7. **Payment Amount Match:** The Payment.amount field must exactly match the Subscription.sub\_price at the time of the transaction.

## IV. REVIEW AND REPORT RULES

1. **Review Per Listing:** Each user (reviewer\_id) is only allowed to submit **one review** (Review) per listing (listing\_id).
2. **No Self-Review:** The seller of a listing (Listing.seller\_id) is **not allowed** to review their own listing.
3. **Rating Range:** The review rating (Review.rating) must be within the range of **1 to 5**.
4. **Report Restriction:** A user (reporter\_id) is only allowed to submit **one unique report** (Report) per listing (listing\_id).
5. **Report Action:** When a listing receives **5 or more reports** (Report), the Listing status must automatically change to **status='Under Review'** and notify the Admin.
6. **Admin Review:** Only Admin/Moderator have the authority to change the status of a Report (Report.status) from 'Pending' to 'Resolved' or 'Rejected'.
7. **Favorite Duplication:** A user (Favorite.user\_id) cannot add the same listing (Favorite.listing\_id) to their favorite list more than **once**.
8. **Reviewer Must be Active:** Only users with an account status of status='Active' are allowed to submit Reviews and Reports.

# Non-Functional Requirements

### ****5.1 Performance Requirements****

* The system must allow at least **100 simultaneous users** to access without interruption.
* Response time for main operations (login, search, view news) is no more than **3 seconds** .

### ****5.2 Reliability & Availability****

* The system must achieve **uptime ≥ 99%** during operation.
* Important data (user, listing, payment) must be backed up periodically every day.

### ****5.3 Security Requirements****

* User password must be **encrypted (hash + salt)** .
* Payment transactions are made via **secure Payment Gateway (HTTPS, SSL/TLS)** .
* Only Admin/Moderator has the right to delete/ban users and handle reports.

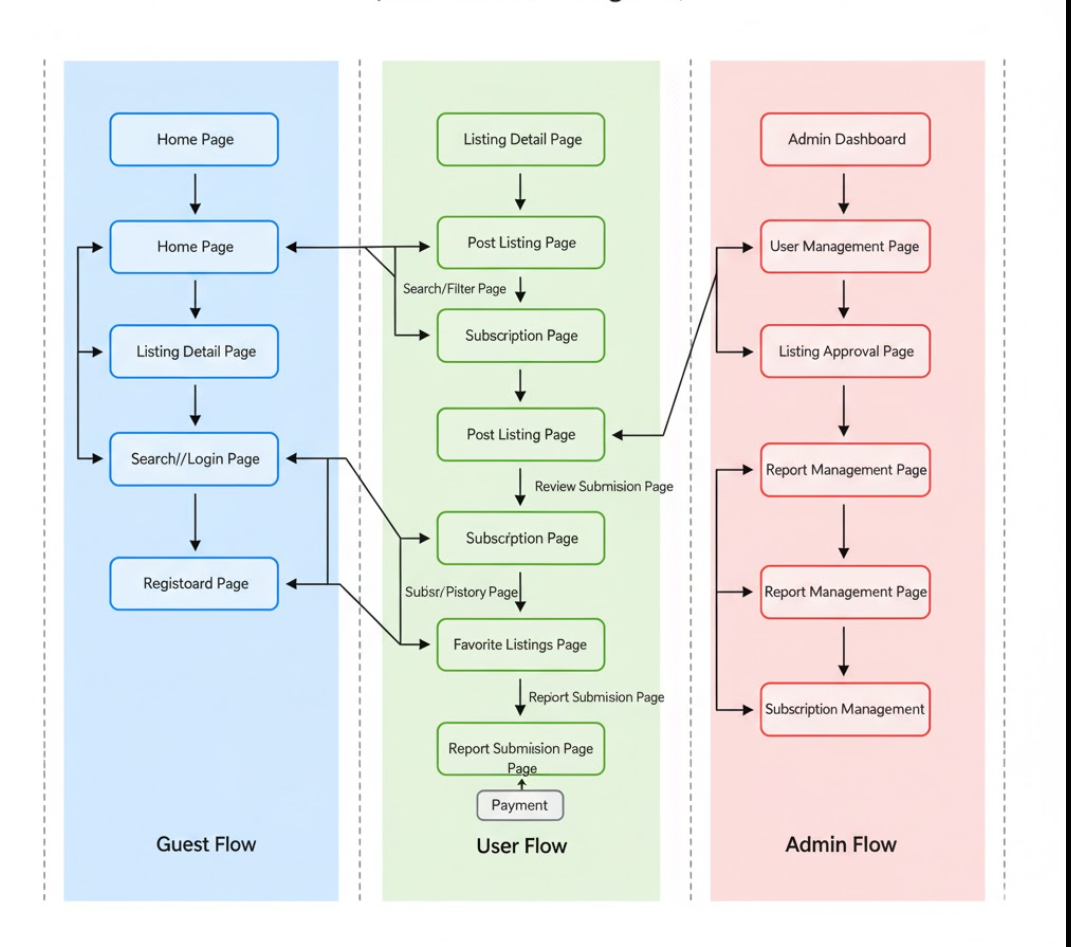
### ****5.4 Usability Requirements****

* The web interface must be **friendly and easy to use** for both sellers and buyers.
* **Multi-device** support system (PC, laptop, mobile).
* Error/violation messages must be displayed clearly and easily understood.

### ****5.5 Maintainability & Scalability****

* Scalable to support **new product types** (beyond vehicles and batteries) in the future.
* The database can scale to **tens of thousands of listings** .

# Screen Flow



### ****6.1 Guest (Not logged in)****

* **Home Page** : displays list of posts, but hides contact information.
* **Search/Filter Page** : search by category, keyword, price.
* **Listing Detail Page** : view post details, only see product information, no contact.
* **Register/Login Page** : create an account or log in to become a User.
* **Payment History Page** : view user's payment history.

### ****6.2 User (Buyer/Seller)****

* **Dashboard Page** : manage account, subscription package, list of posted news.
* **Post Listing Page** : posting form (title, description, price, image, vehicle/battery information).
* **Subscription Page** : choose package (Free, Basic, Premium, VIP), pay via Payment Gateway.
* **Listing Detail Page** : view listing details (with contact information).
* **Review/Report Page** : leave a review for the seller, or report violations.

### ****6.3 Admin/Moderator****

* **Admin Dashboard** : quick user statistics, listing, report.
* **User Management Page** : view user list, ban/unban accounts.
* **Listing Approval Page** : browse/hide/delete listings.
* **Report Management Page** : processes reports submitted by users.
* **Subscription Management Page** : add/edit/delete service packages.
* **Payment History Page** : view user's payment history.

# Technologies

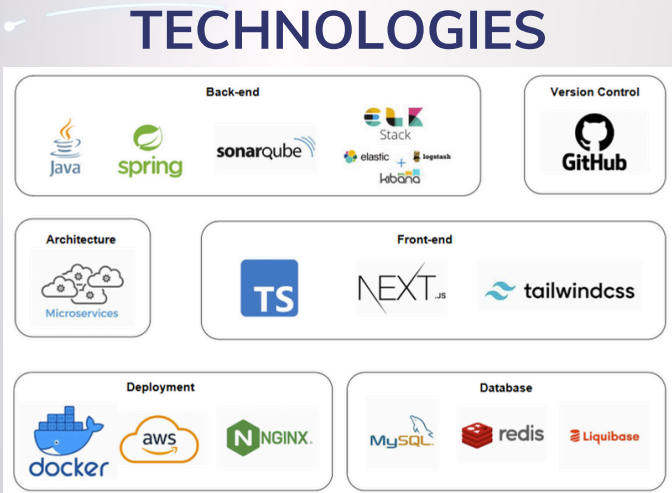
[Font-End]

github

[Back-End]

Springboot, sqlserver, github

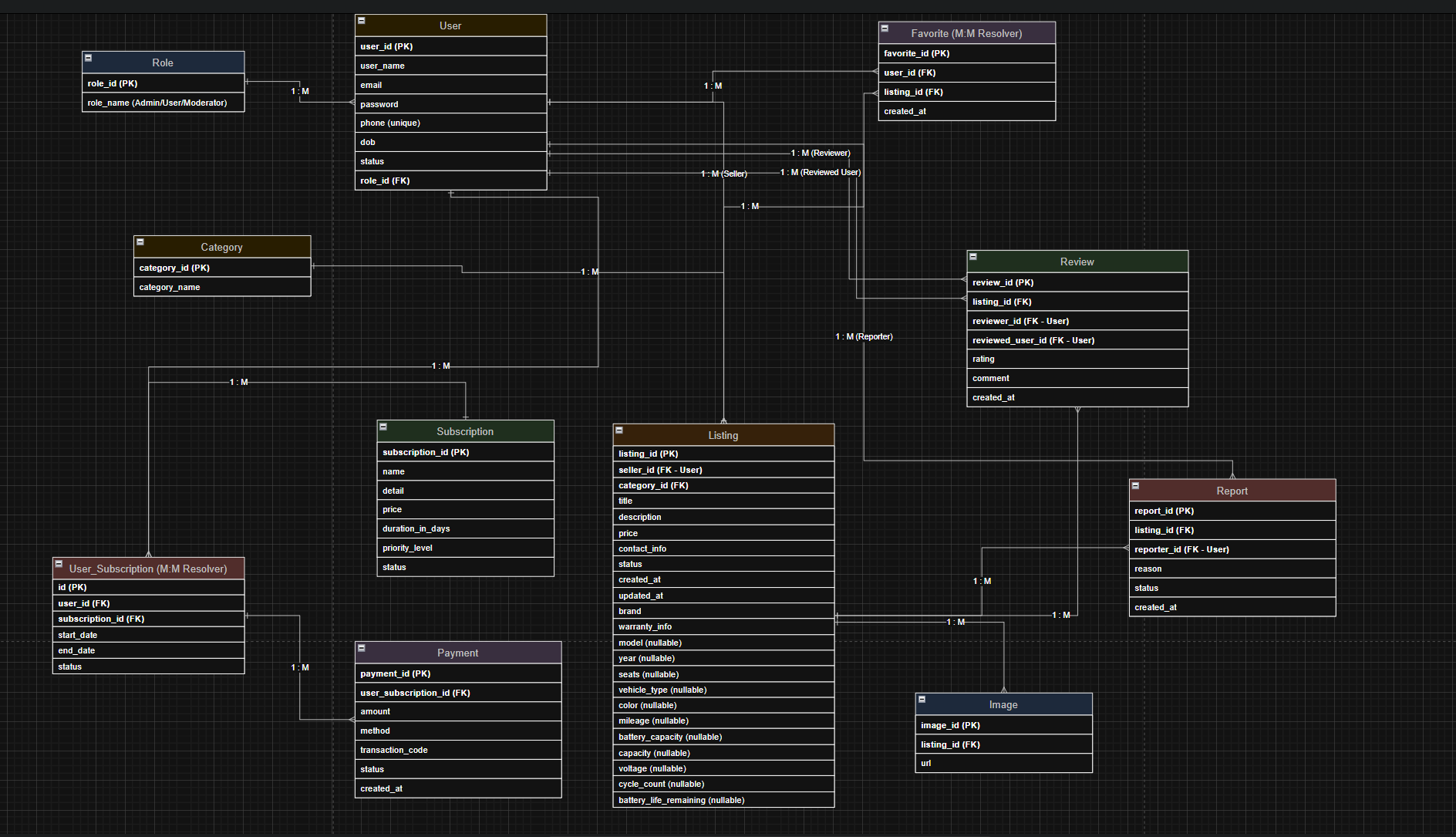
Example:



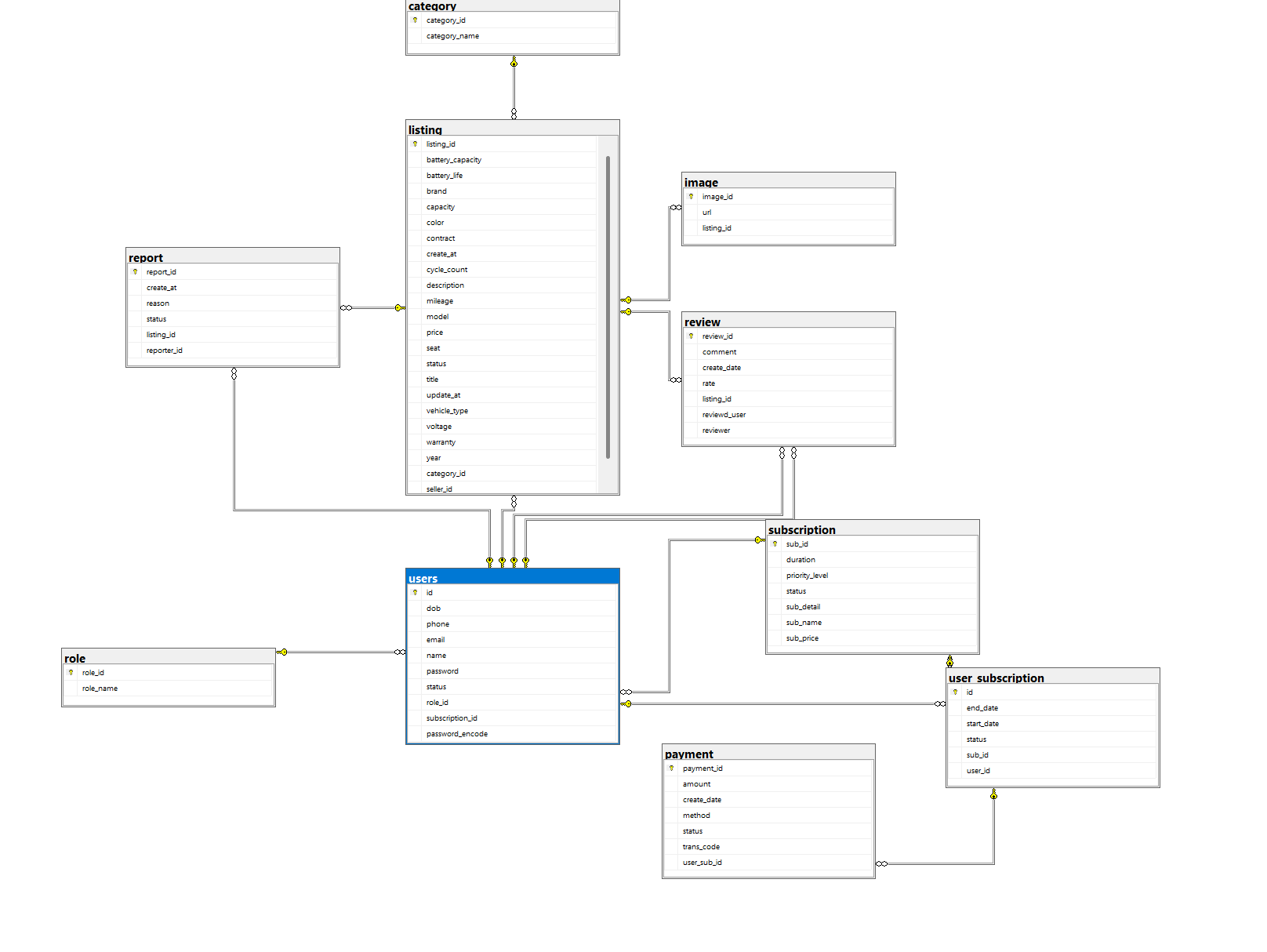
# database design

## [ERD]

Link:[ERD link](https://drive.google.com/file/d/1JzlEW1Jw4_WGqzWvhdOC22AVQovGjya_/view?usp=sharing)



## [Database diagram]



# source control

Fe:

<https://github.com/segciest/swp391_fa25_fe_2>

Beige:

<https://github.com/segciest/swp391_fa25_be>