105

Travelus
Software Architecture Document

Version 1.0

Revision History

Date	Version	Description	Author
13/07/2023	1.0	Init Software Architecture Document	Nguyễn Phúc Thịnh

Table of Contents

Software Architecture Document

1. Introduction

a. Purpose:
This document presents a thorough description of the system's architecture, utilizing various architectural perspectives to illustrate different aspects of the system. Its purpose is to capture and communicate the important architectural choices that have been implemented in the system.

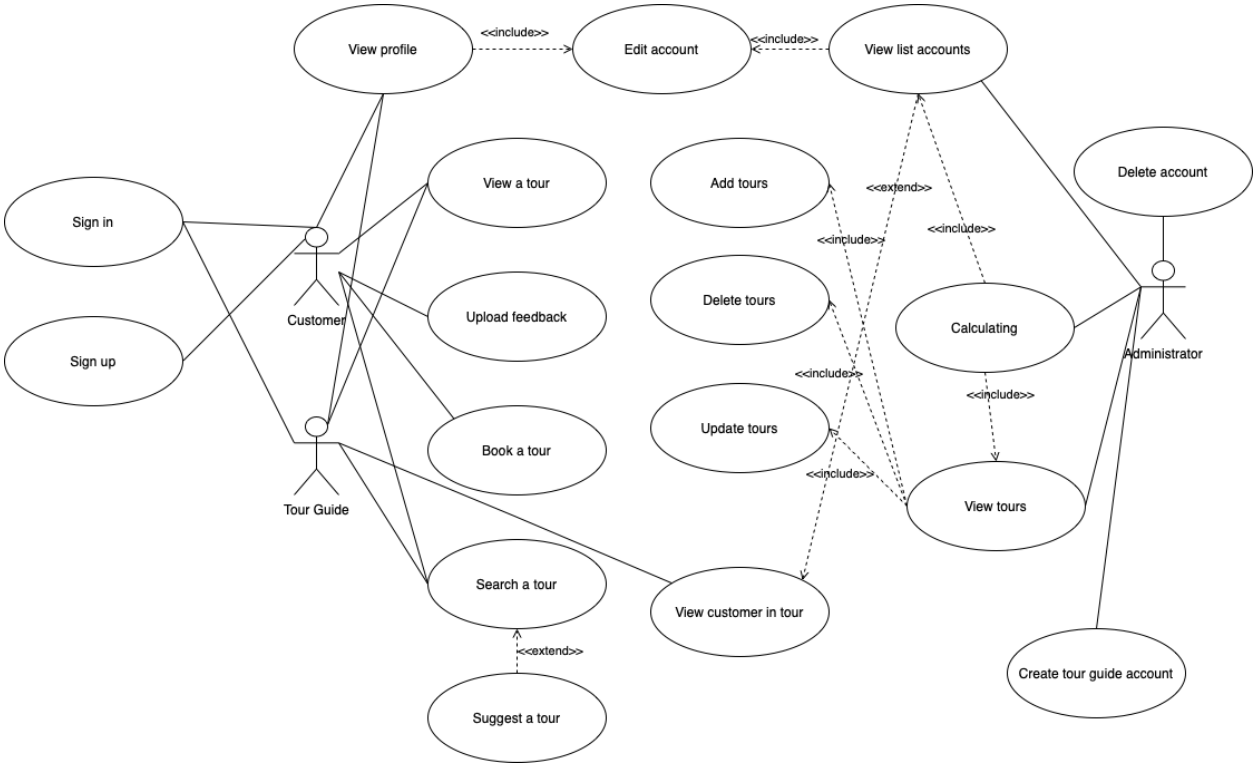
b. Scope:
This Software Architecture Document pertains to the Travelus website System, which is slated to be developed by Group 105.

c. Definitions, acronyms and abbreviations:
Customer Profile: A customer profile for a travel website is a collection of information and preferences about an individual user. It includes personal details, travel preferences, and interests, helping the website personalize recommendations and offers for a tailored travel experience.
Content: it includes all information of the tour which can be delivered to a user. It can be the name of the tour, time, destination, transportation, schedule.
Page: The page provides information about the tours which are available.
Subscription: A subscription is a booking of a customer about the tour that is coming soon.
Tour guide: A tour guide is a knowledgeable and experienced individual who leads and provides informative commentary to a group of travelers during their visit to destinations in the tour.
Website: This is a website which provides tours for those who are enthusiastic in travel.

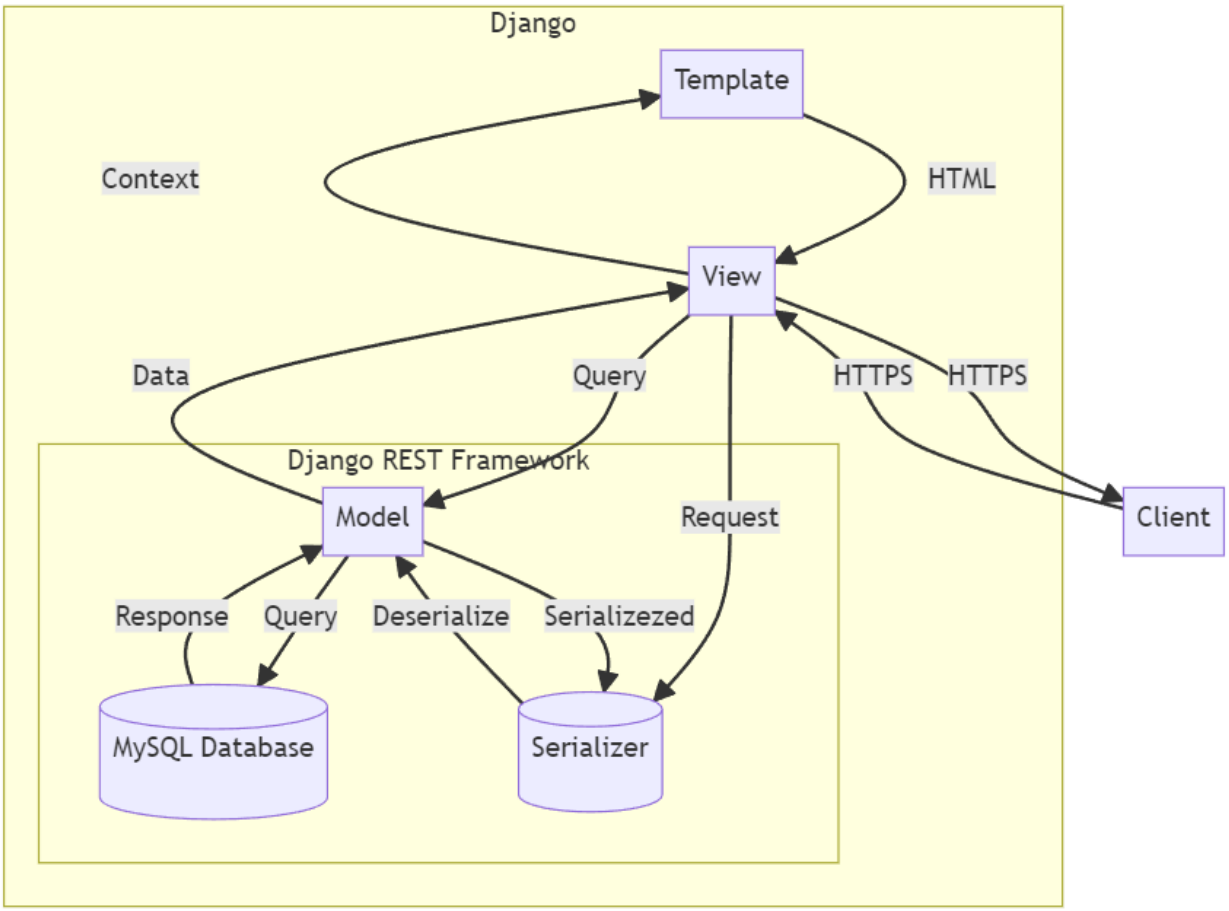
2. Architectural Goals and Constraints

- The system has to be maintainable
- Easy-to-understand schema
- Require cheaper hardware
- Use available frameworks like Django
- Safety using HTTPS to communicate Client-Server
- Security storing information users by hashing and salting
- At least running by one server.
- Implement follows MVT architecture.
- Three members implementing backend and two members implementing frontend.

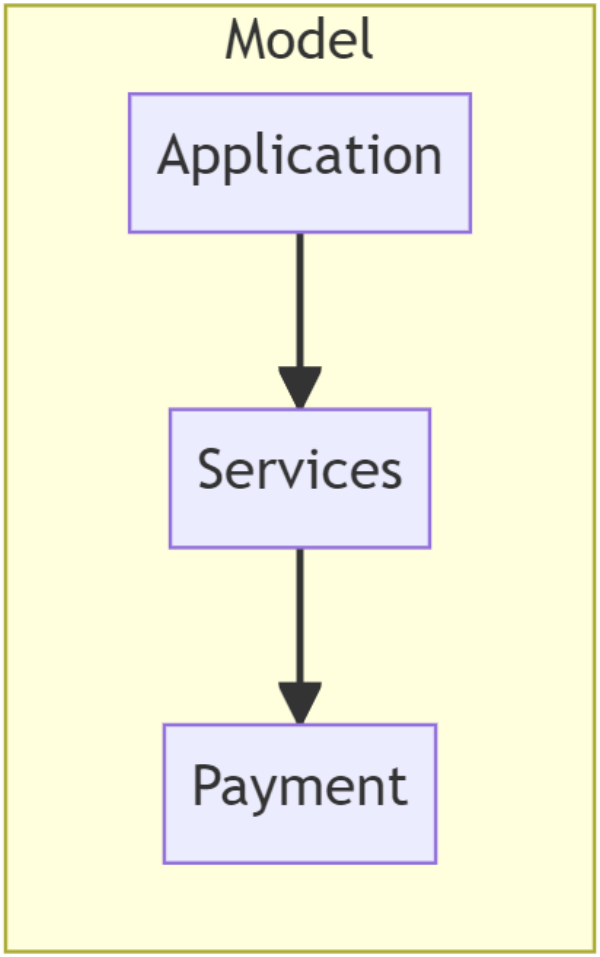
3. Use-Case Model



4. Logical View

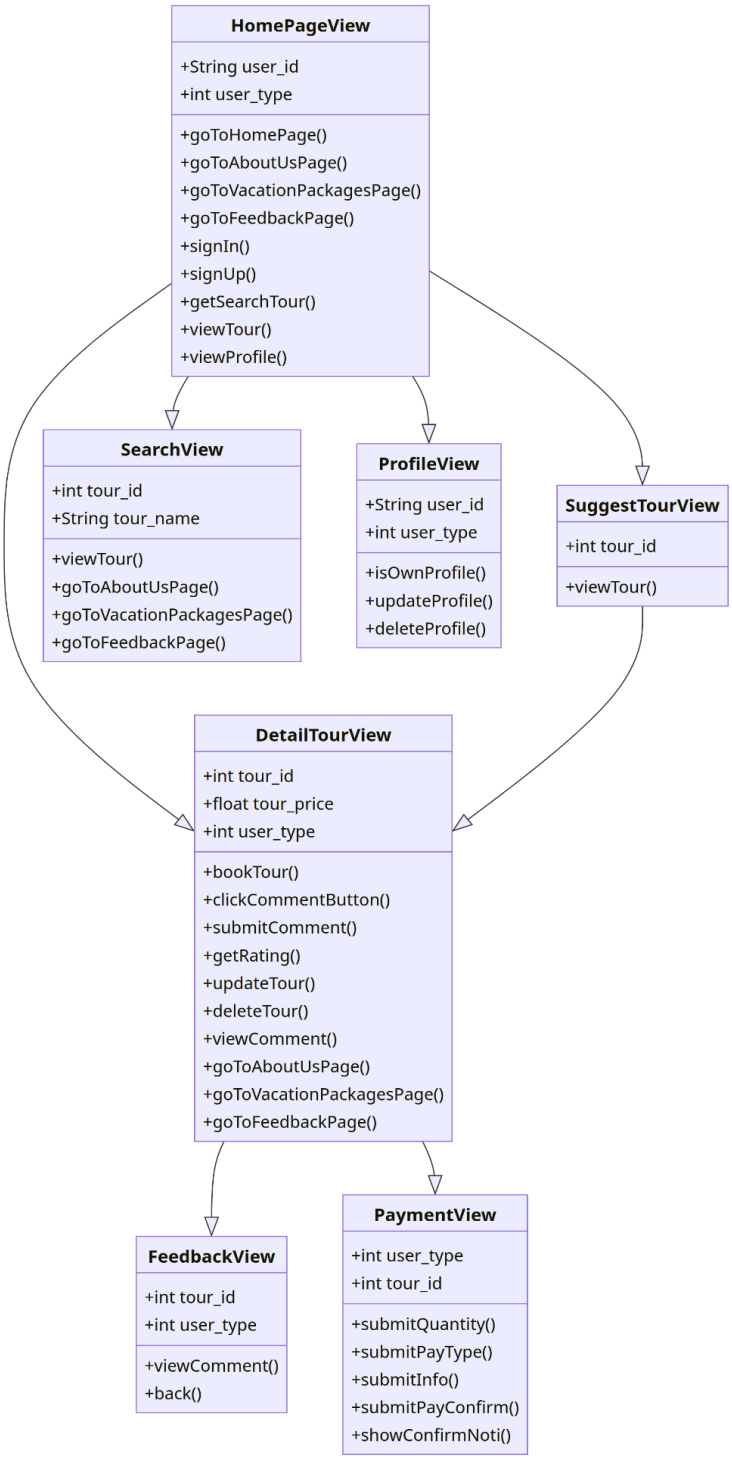


In the model package there are three smaller packages:



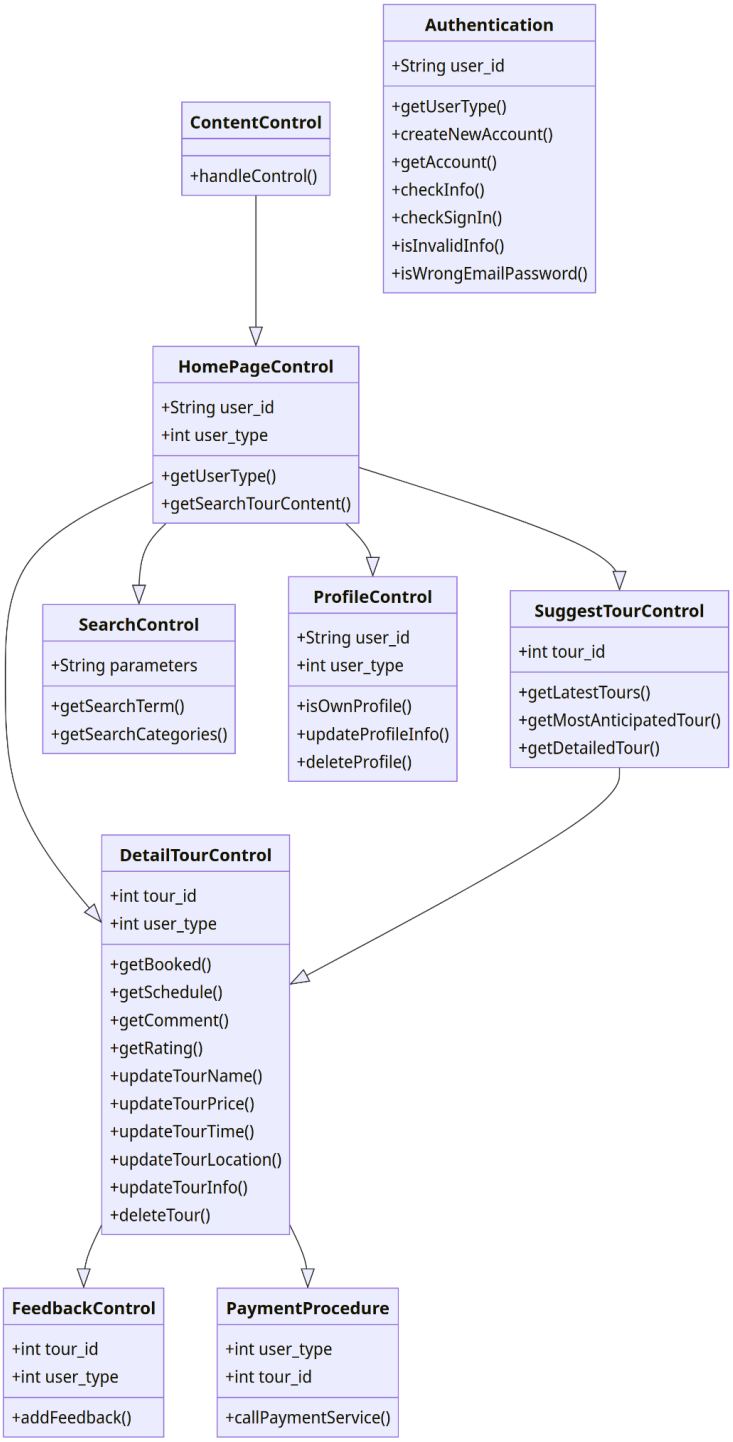
4.1 Component: View

The noticeable classes in this view are: HomePageView and DetailTourView. HomePageView is the first thing user will experience when visiting the website, it calls for other views provided on the web. DetailTourView class is the main tour information delivery. It contains schedules, price, places, etc.



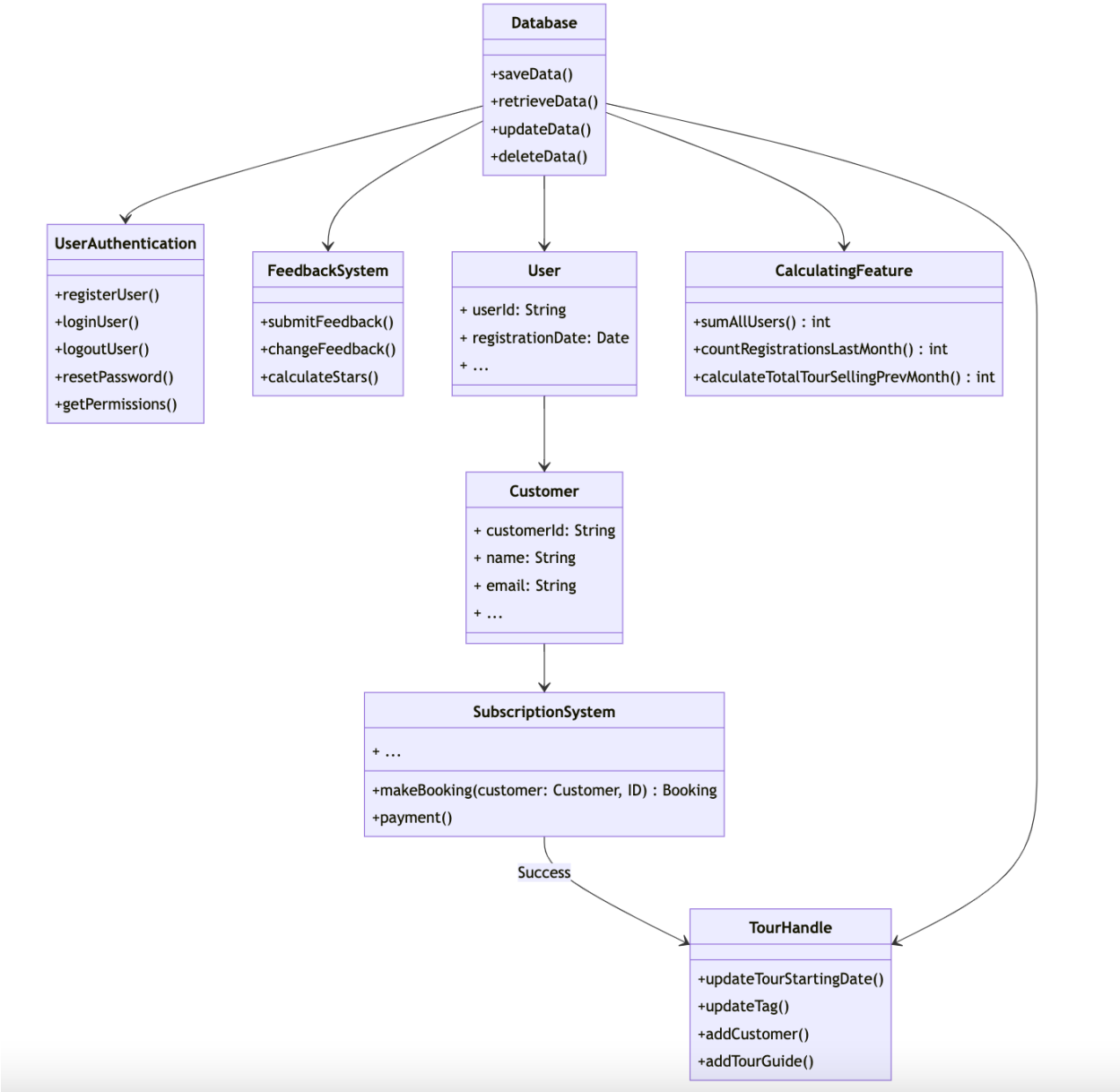
4.2 Component: Application

Content control is the most significant feature in this component. This class helps the FE call a corresponding function that matches the element user clicks.



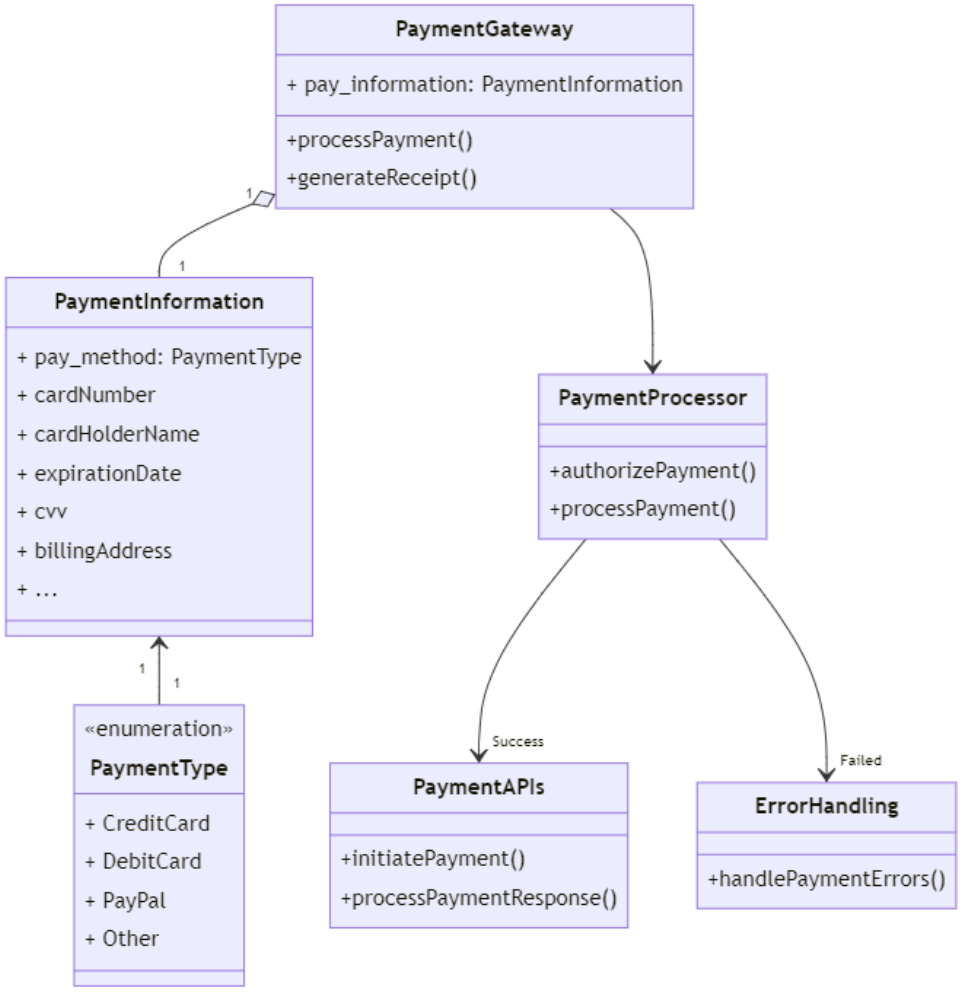
4.3 Component: Services

Services components get the profile of the user and handle all kinds of use-case functionality in it. This is the main business component of the system. All classes in this component are equally important, but the database class stands out a bit more. This will be implemented with lazy initialization and error handling to withstand a moderate number of users.



4.4 Component: Payment

The payment feature is holding the payment information of the user, this class has to be managed correctly to ensure customer satisfaction and security.



5. Deployment

[Leave this section blank for PA3.]

6. Implementation View

[Leave this section blank for PA3.]