A PLATE

Software Requirements and Analysis

A Plate

Version 1.0

2021-11-07

Revision History

Revision	Date	Author	Reviewed By	Summary of Changes
Draft	2021/10/28	Kursad Ozduman	Resul Yuksektepe	Initial draft
Final	2021/11/05	Kazi Hasanus Safa	Hamza Koc	Final

Document Approval List

Version	Approved By	Signature	Date
0.1	Resul Yuksektepe	x	2021/11/05
1.0	Hamza Koc	x	2021/11/05
1.0	Kursad Ozduman	x	2021/11/05
1.0	Kazi Hasanus Safa	x	2021/11/05

Document Distribution List

Version	Name of the Receiver/Group	Date
1.0	Project Co-ordinator	2021/11/10

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the document is to record software requirements and designs for the "A Plate" project.

1.2 Scope

A Plate is intended to be developed primarily for web platforms but is later intended to roll out on the Android mobile platform. The project allows users to easily list the discounts offered by the restaurants, register easily, and take advantage of the opportunities by being a part of a social community. Users, who come together at the tables of the restaurants they choose, will both benefit from the discounts offered by restaurateurs who are members of the A-Plate program and expand their social networks by meeting with people they do not know. In addition, users will be able to rate restaurants through the user experience rating system. Anyone who is or is not a registered user of the program will also see these ratings, this will allow restaurants to find additional customers.

2. System Overview

The application is for everyone to expand their network while eating around a table and having fun mainly at designated restaurants. The goal of this project is to create web site for restaurants to post a deal for a special day, special table and days to give some discounts, at the same time for people who want to get discount and sit on the specified table with foreigners, to introduce themselves, having fun and adding them in their networking list.

2.1 Project Perspective

A Plate is a new self-contained project. The project will be developed by team members with collaboration.

2.2 System Context

The core part of the is a REST API with MongoDB and served at AWS hosting. The backend will be designed appropriately for the Android that will be added later. The project will be easy to design and will allow for a consistent design on any platform. Also, since the resources can work independently of each other, it will allow for simple maintenance and updating.

2.3 General Constraints

2.3.1 Business Constraints

Although we are working in cooperation, it is very difficult to get together at the same hours and days because we cannot get together due to the virus. Therefore, there are differences of opinion on the features that our project needs. The fact that we have other classes together with the project, and we must work for our basic needs causes us excessive stress towards the delivery date of our project.

2.3.2 Technical Constraints

Our technical constraints for the beginning have been determined as follows; we do not have an institutional structure support yet, such as fast response difficulties for customer service.

It is possible for users to register in our system with fake email addresses. Although we can verify this through verification, it is very easy to overcome this with today's technology.

We will need to continually improve our software to get the most out of a fluid and attractive design by inviting users to use our platform.

2.4. Assumptions and Dependencies

All the features and interfaces we have considered for our project will be accessible to users after they are completed. We will accept advertising and cooperation for the cost and maintenance of our site.

3. Functional Requirements

3.1 Functional Requirements or Features

Accounts			Services	Entry
Check User			Get restaurants	Create event/campaign
Register	Normal		Get events	Add file to the event/campaign
	Social media			
Login	Normal	Remember	Register events	Edit event/campaign
	Social media			
Logout			Get registered events	Delete event/campaign
Update account / password			Give review to restaurants	
Disable account			Give rating	
			Get registered user list	

3.2 Use Cases

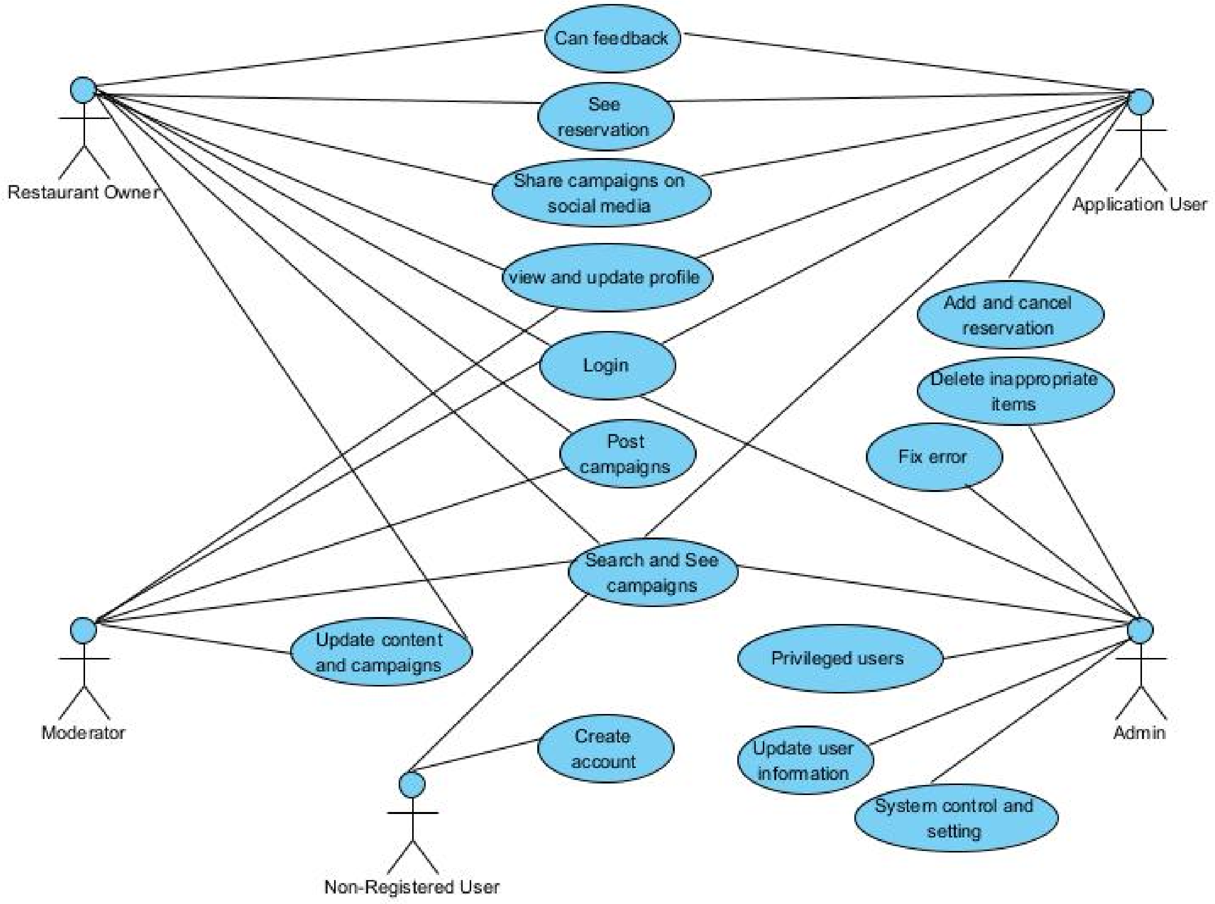
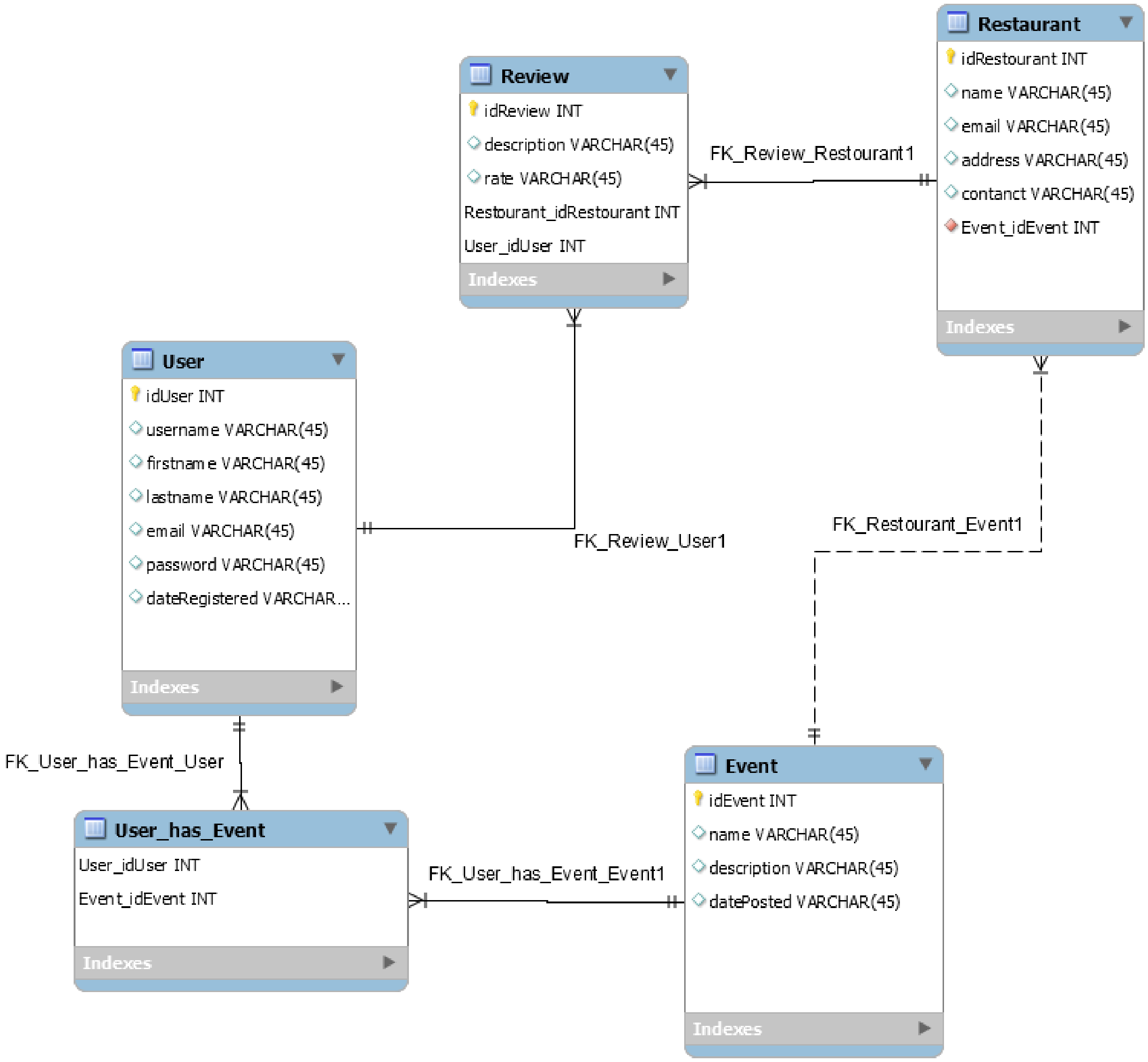


Fig: Use Case Diagram

Powered By Visual Paradigm Community Edition

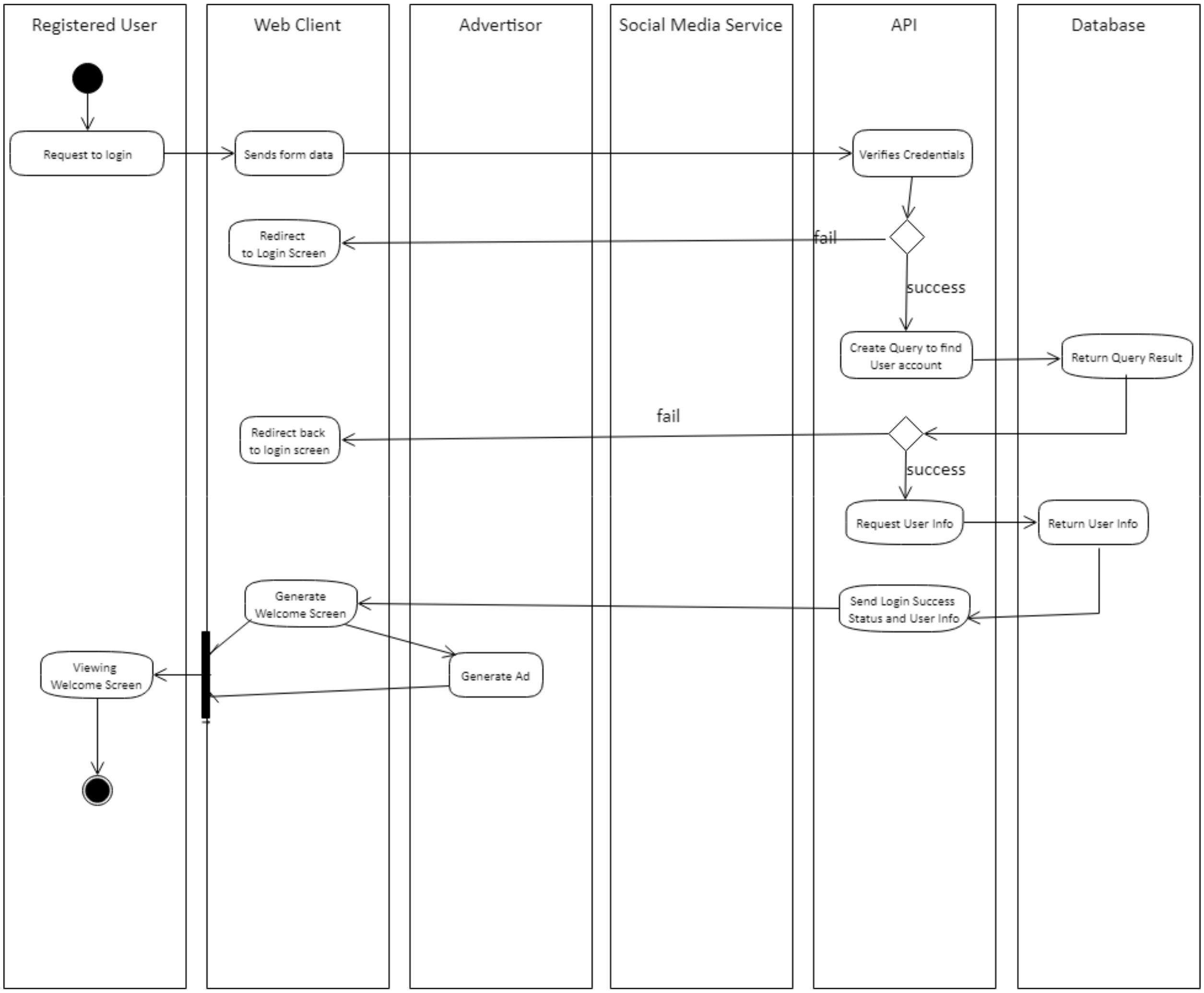
3.3 Data Modelling

3.3.1 Normalized Data Model Diagram

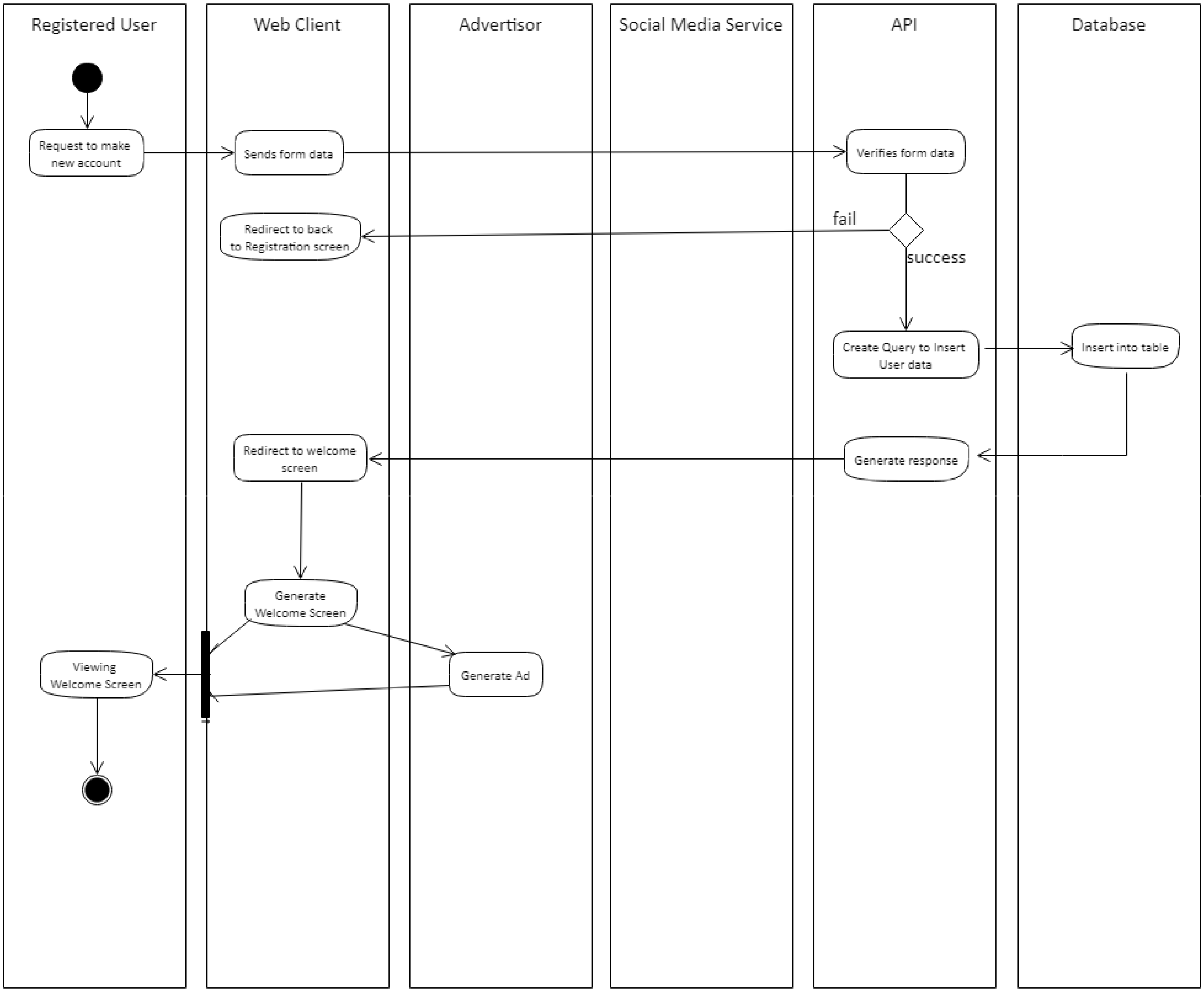


3.3.2 Activity Diagrams

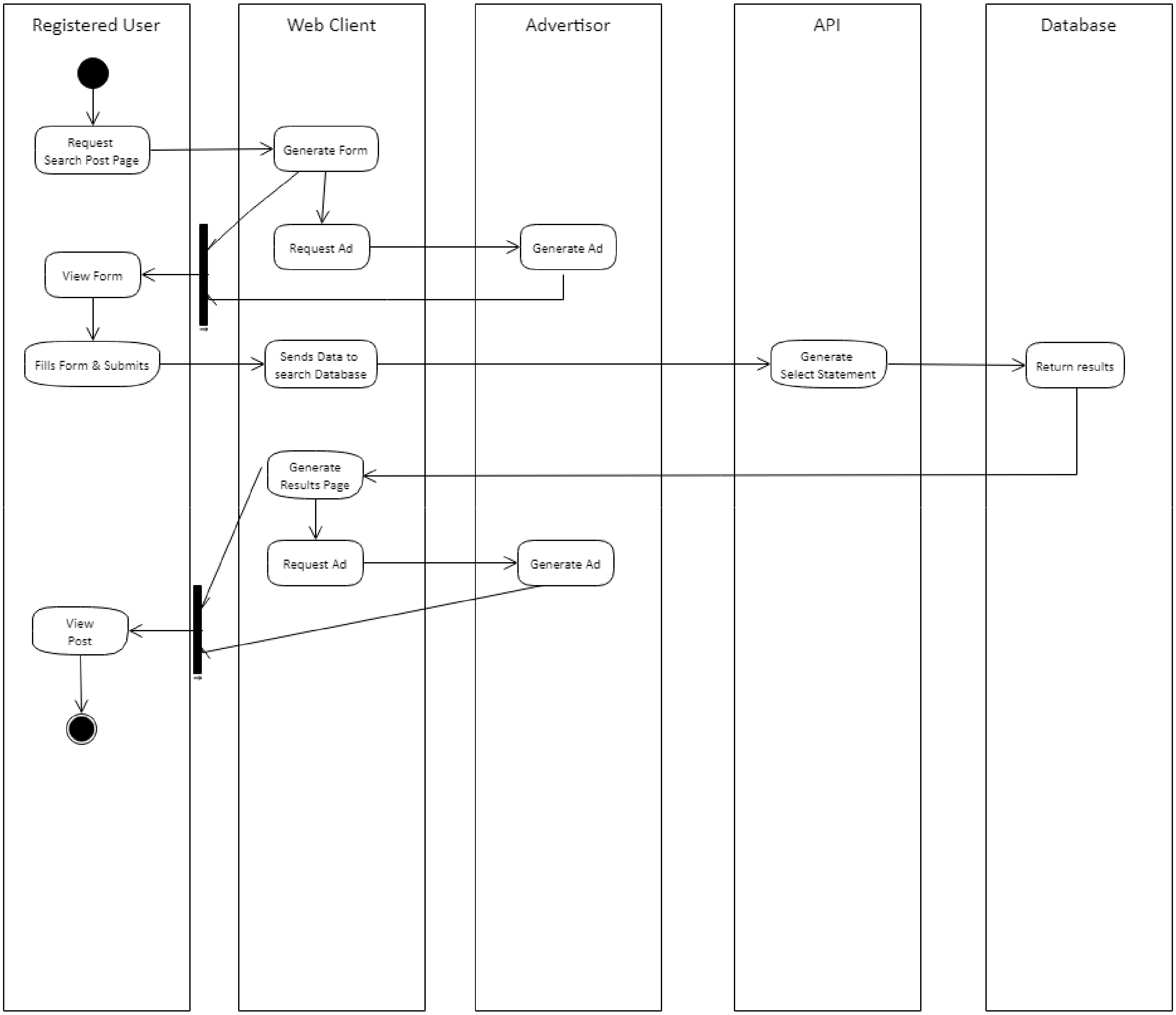
3.3.2.1 Login Activity Diagram



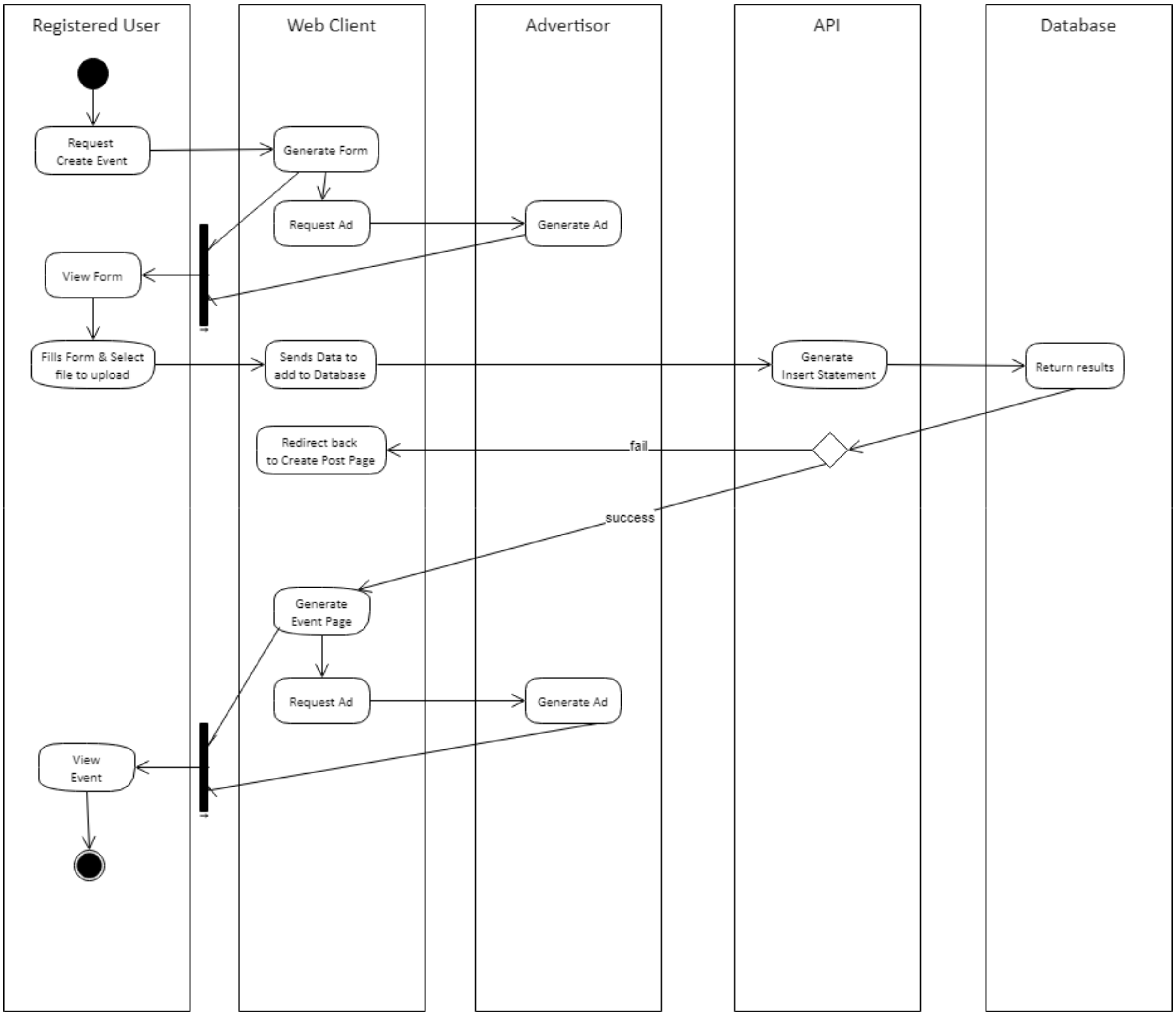
3.3.2.2 Registration Activity Diagram



3.3.2.3 Search Event Activity Diagram

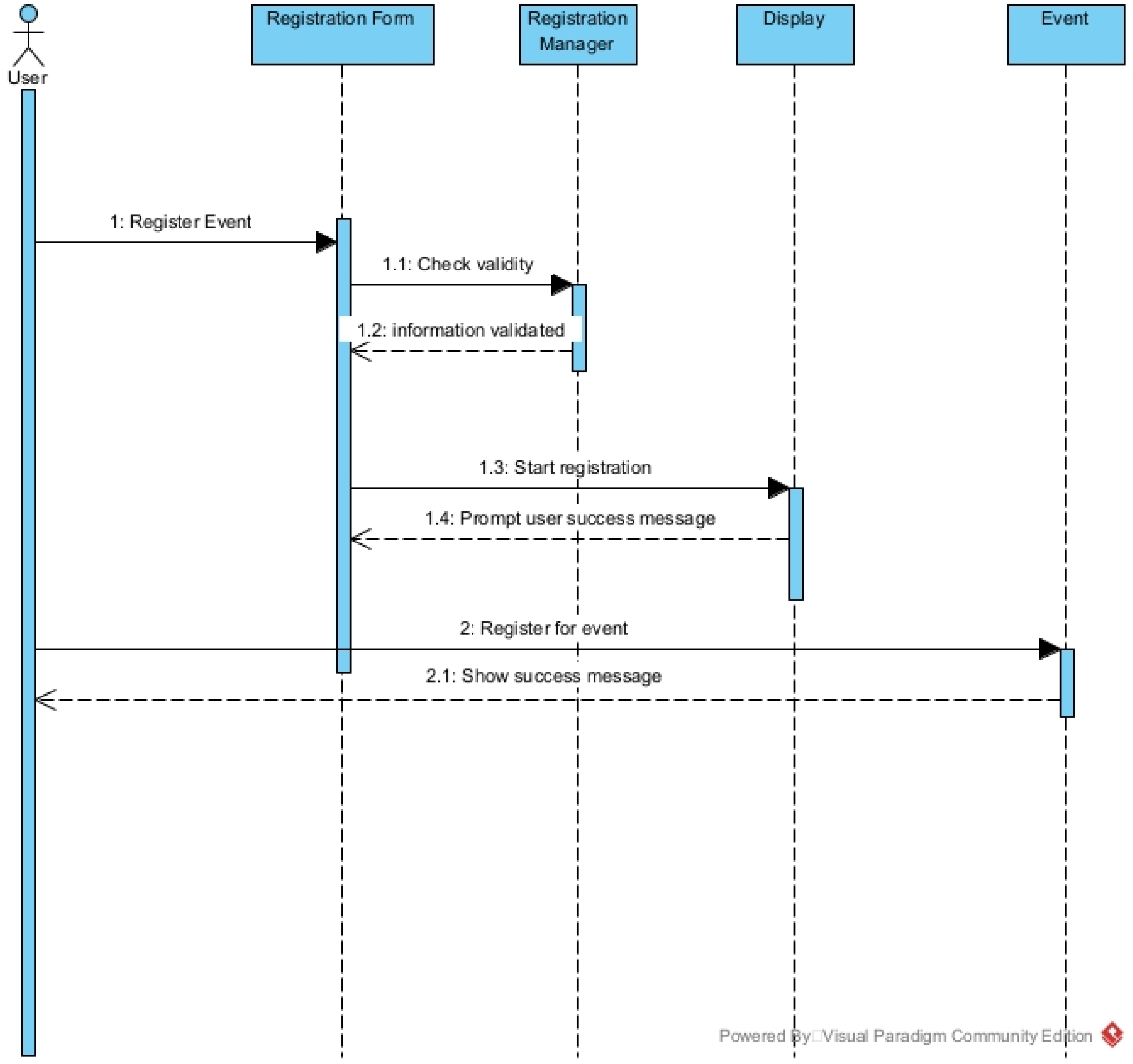


3.3.2.4 Create Event Activity Diagram

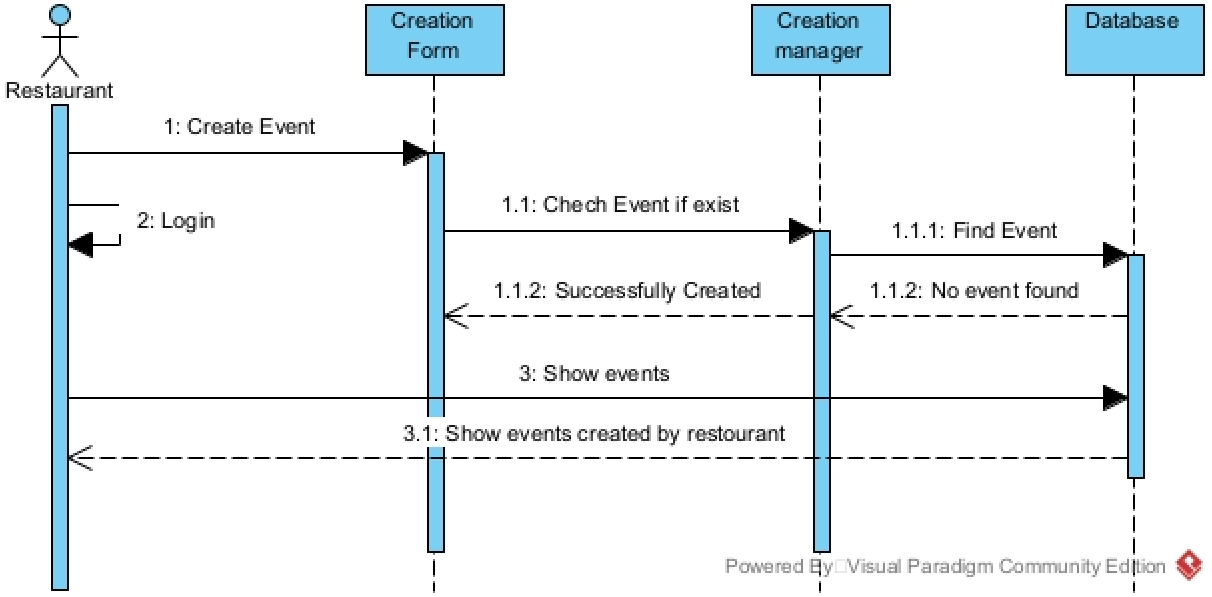


3.3.3 Sequence Diagrams

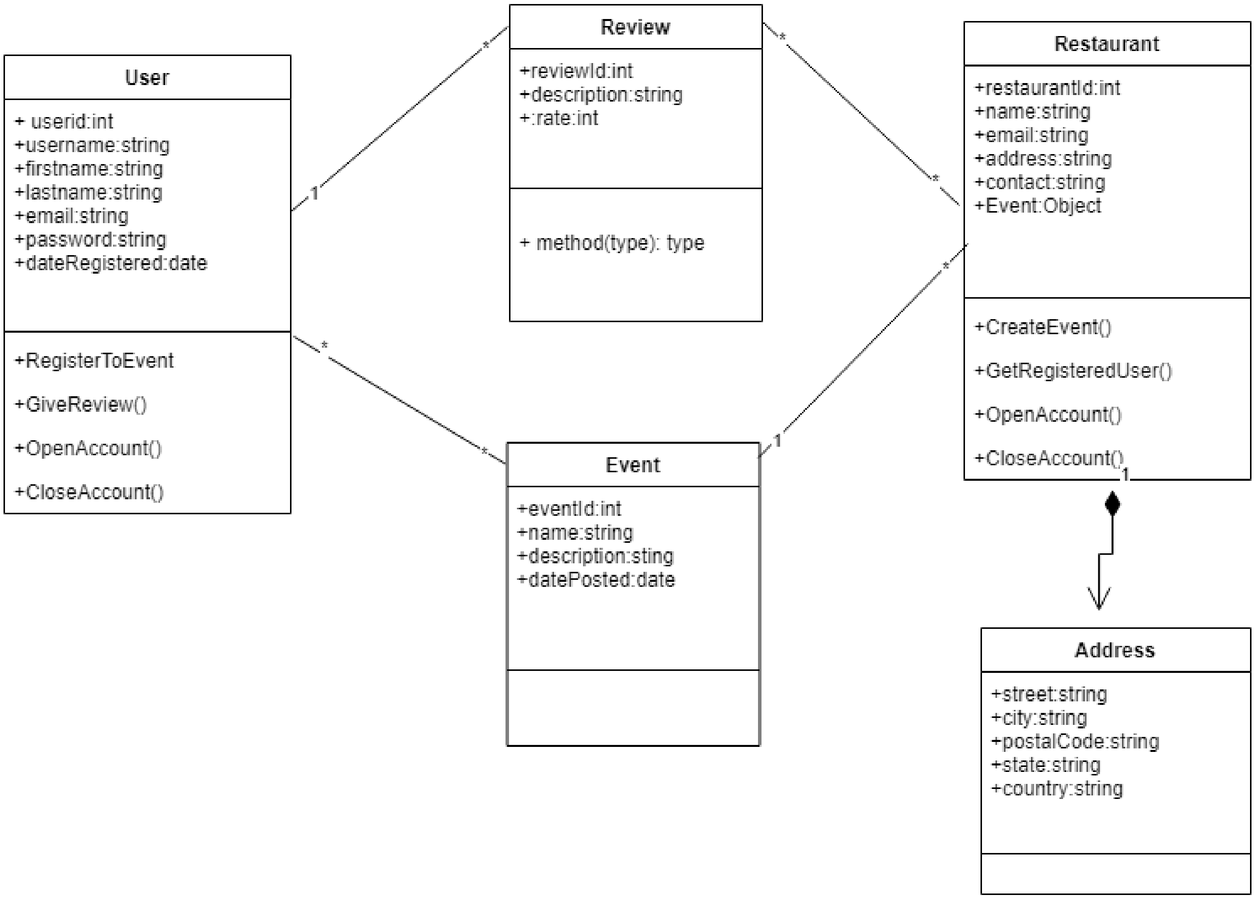
3.3.3.1 User Sequence Diagrams



3.3.3.2 Restaurant Sequence Diagrams


Restaurant
Creation Form
Creation manager
Database
1: Create Event
2: Login
1.1: Chech Event if exist
1.1.1: Find Event
1.1.2: Successfully Created
1.1.2: No event found
3: Show events
3.1: Show events created by restourant
Powered By Visual Paradigm Community Edition

3.3.4 UML Class Diagram



3.4 Process Modelling

3.4.1. Data Flow Diagram

3.4.1.1 Level 0

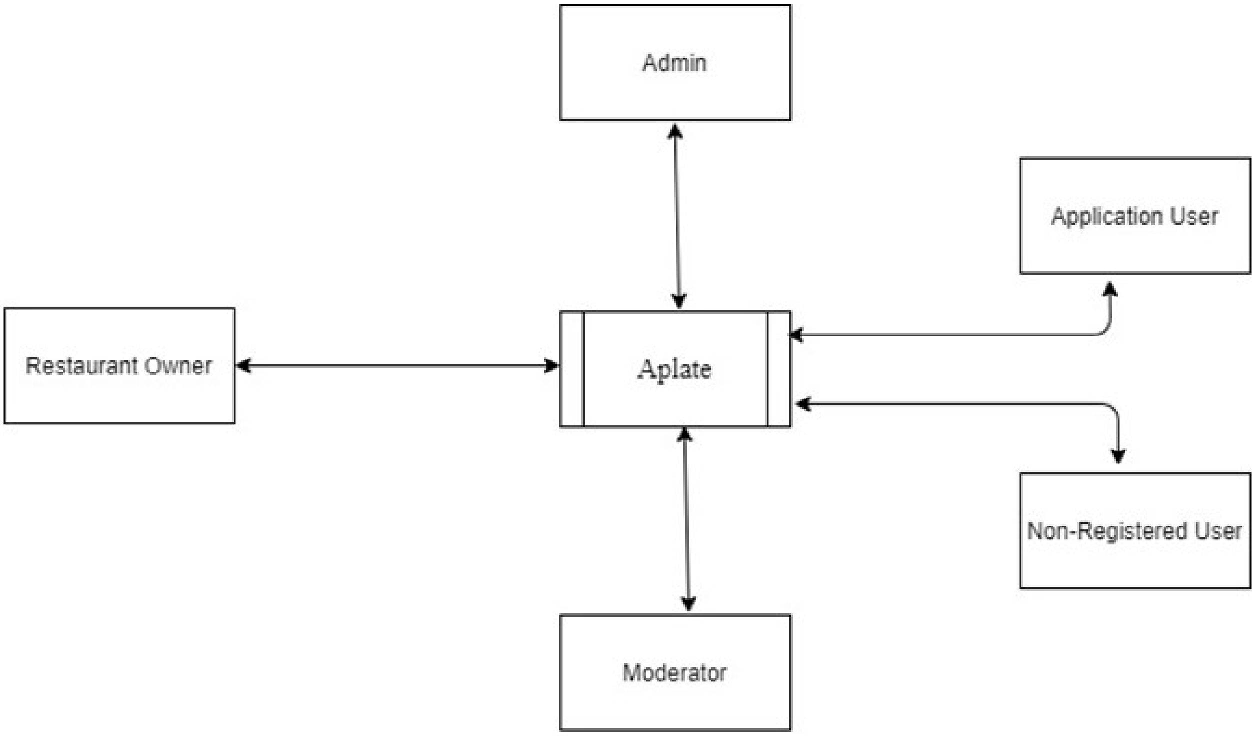
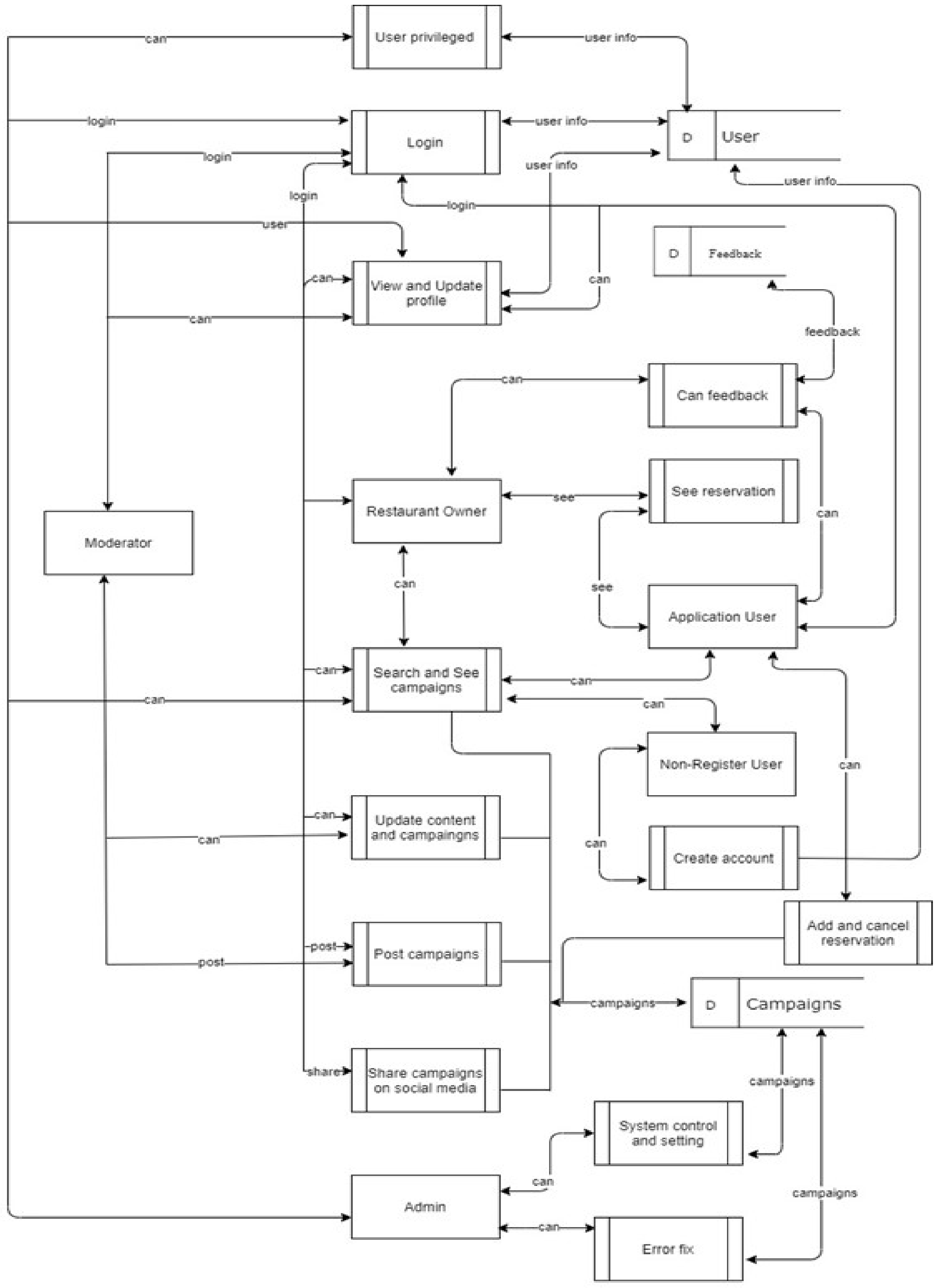


Fig: Context Diagram/Level 0

3.4.1.2 Level 1



Level 1

3.4.2.3 Level 2

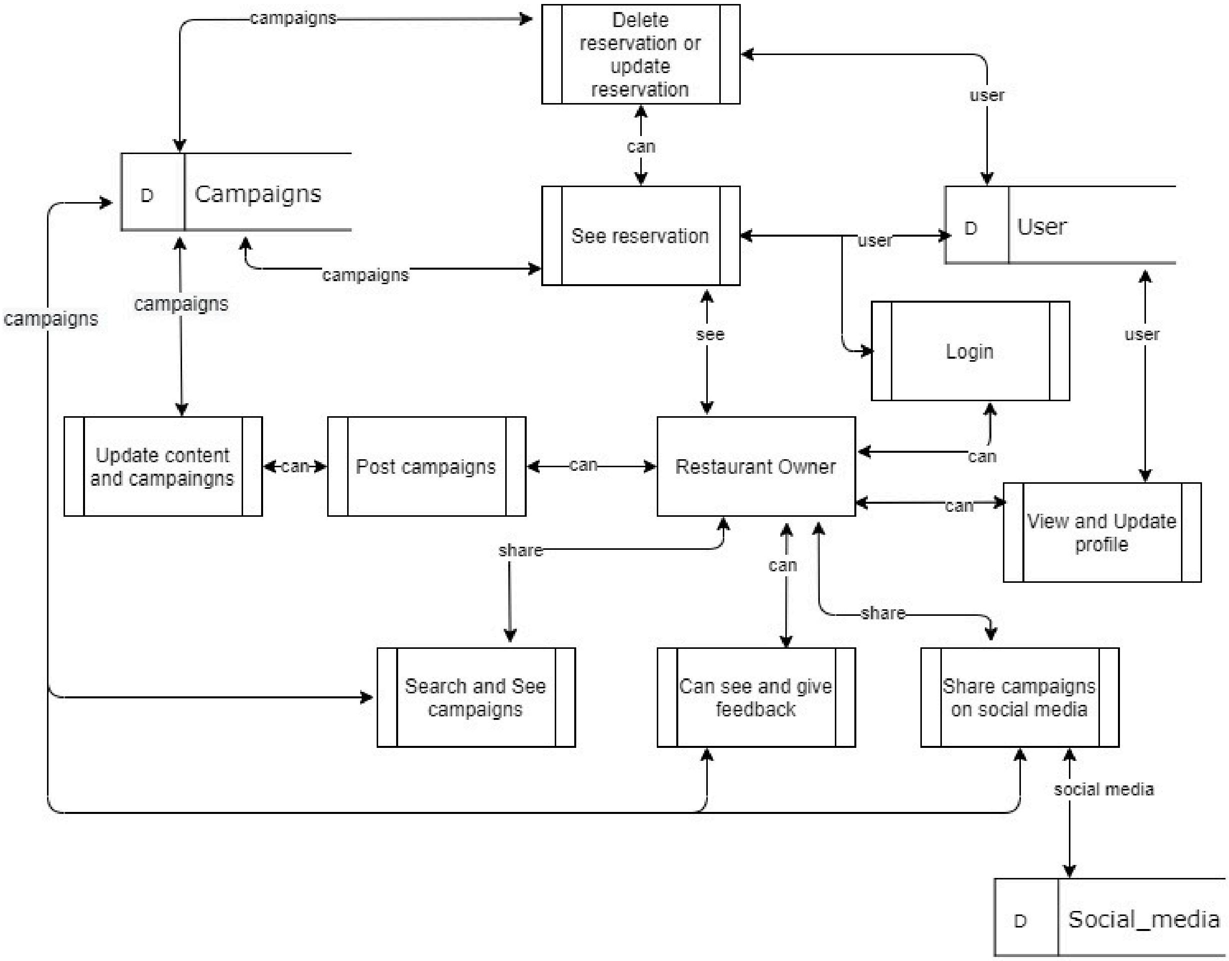


Fig: Level 2

4. Non-Functional

Operation	Maintenance	Usablility
We will periodically query our database to generate user activity reports. This allows us to have a basis for future improvements and monitoring activities.	All pages will be tested and necessary maintenance and updating will be done without damaging other codes.	All campaigns and restaurant information will be up to date.
All files running on our system will be audited and these files will be legal. We will ensure this with user login, continuous database checks and validation on login.	A small change to be made should not adversely affect the database in any way.	There will be improvements in the service provided with the scoring system given by the users.
Necessary infrastructure works will be carried out to minimize site traffic.		

5. Logical Database Requirements

An open-source database which is one of the NoSQL-based database systems named MongoDB will be used. Since it is a document-based and scalable application, it was preferred instead of structures that traditional relational databases could not keep up with and became cumbersome, due to its fast and open-source code. Multiple copies of the original data will be created with MongoDB, thus preventing data loss. It is aimed to process the data in bulk.

6. Approval

Project Role	Name	Signature	Date
Lead Developer	Hamza Koc	X	2021/11/08
UX Designer	Kazi Hasanus Safa	X	2021/11/08
Project Manager	Resul Yuksektepe	X	2021/11/08
Production/Deployment	Kursad Ozduman	X	2021/11/08