MIDDLE EAST TECHNICAL UNIVERSITY

POLYHEDRA

SOFTWARE DESIGN DESCRIPTION

SE 560 SOFTWARE DEVELOPMENT STUDIO
SPRING 2019-2020

İLKYAZ ÖZER - 2064921
O. ÖZGÜN ÇOKGEZEN – 1558758
SEMA KAMIŞLI - 2340610
TAHA YAYCI - 2407716

CHANGE HISTORY

Version	Date	Authors	Remark
SDD v0a	16.04.2020	İlkyaz Özer O. Özgün Çokgezen Sema Kamışlı Taha Yaycı	SDD First Draft
SDD v0b	04.06.2020	İlkyaz Özer O. Özgün Çokgezen Sema Kamışlı Taha Yaycı	SDD Final Bundle Release

PREFACE

This description is prepared for SE560 Software Development Studio course.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SOFTWARE DESIGN DESCRIPTION

1 Design Introduction

1.1 Purpose

The purpose of the software design document (SDD) for Polyhedra is to provide design details to developers, quality assurance specialists, testers, configuration specialists, project managers and software architects. Thus, it will be utilized as a guideline by every stakeholder of the project especially for development, as well as during further stages of the project. Exceptionally, user and client will not be intended audience for this SDD, since it includes technical details which are not their concern.

1.2 Scope

SDD for Polyhedra includes the details of every component of the project. It aims to satisfy the functional and non-functional requirements that are stated in Software Requirements Specification (SRS) of Polyhedra.

In this SDD, IEEE-1016-2009-SDD (IEEE Standard for Information Technology Systems Design - Software Design Descriptions) is taken as a basis and it contains the following diagrams in order to precisely explain the design viewpoints.

- Context diagram
- Use Case diagram
- Component diagram
- Class diagram
- Entity-Relationship diagram
- MVT diagram
- Deployment diagram
- Sequence diagram

2 References

[1] IEEE-1016-2009-SDD: IEEE Standard for Information Technology Systems Design - Software Design Descriptions

[2] https://djangobook.com/mdj2-django-structure/

3 Design Viewpoints

3.1 Context Viewpoints

3.1.1 Context Diagram

The Context viewpoint describes services provided by a design subject, relationships and dependencies in the software development environment. The Context viewpoint is a way of representing the system as a black box, users, stakeholders and their interaction are represented. Context diagram is given in Figure 1.

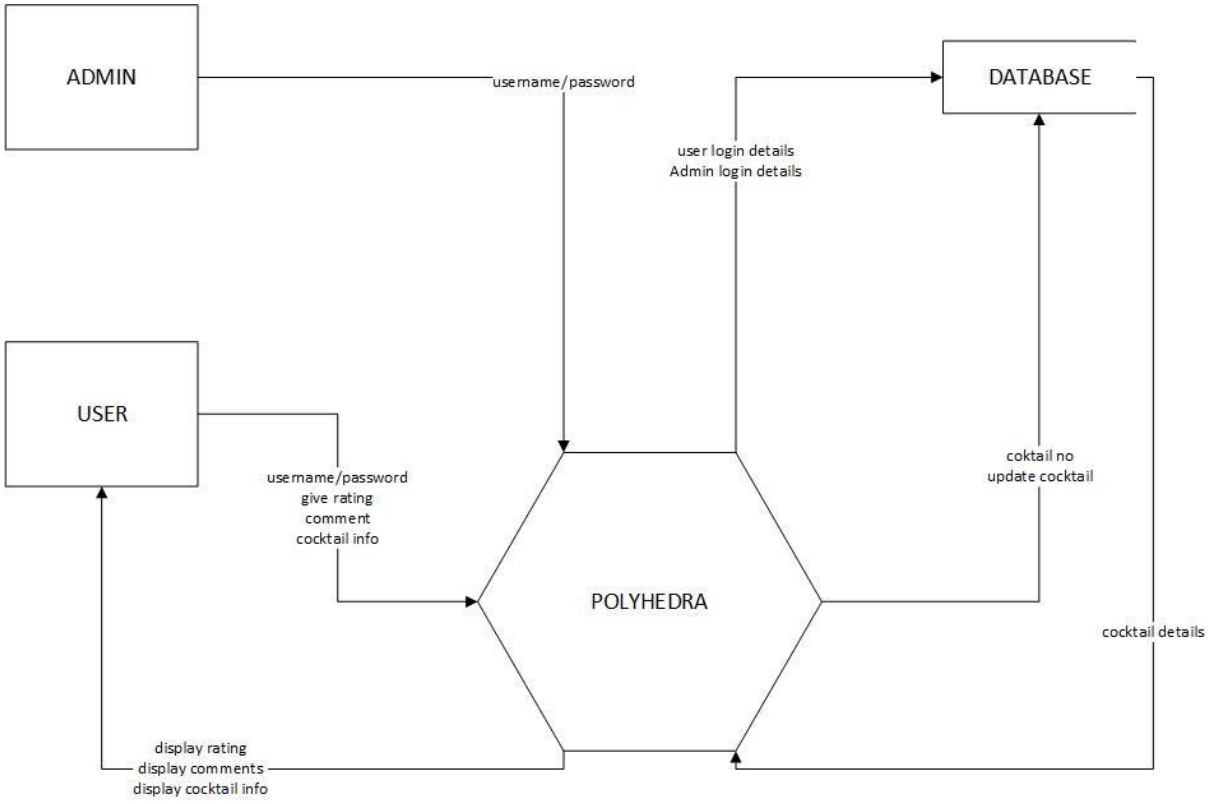

Figure 1: Context Diagram

3.1.2 Use Cases

The UML Use case Diagrams given in Figure 2 illustrates how a user interacts with the system. The detailed explanation of the use case scenarios are provided in Table 1 to Table 8.

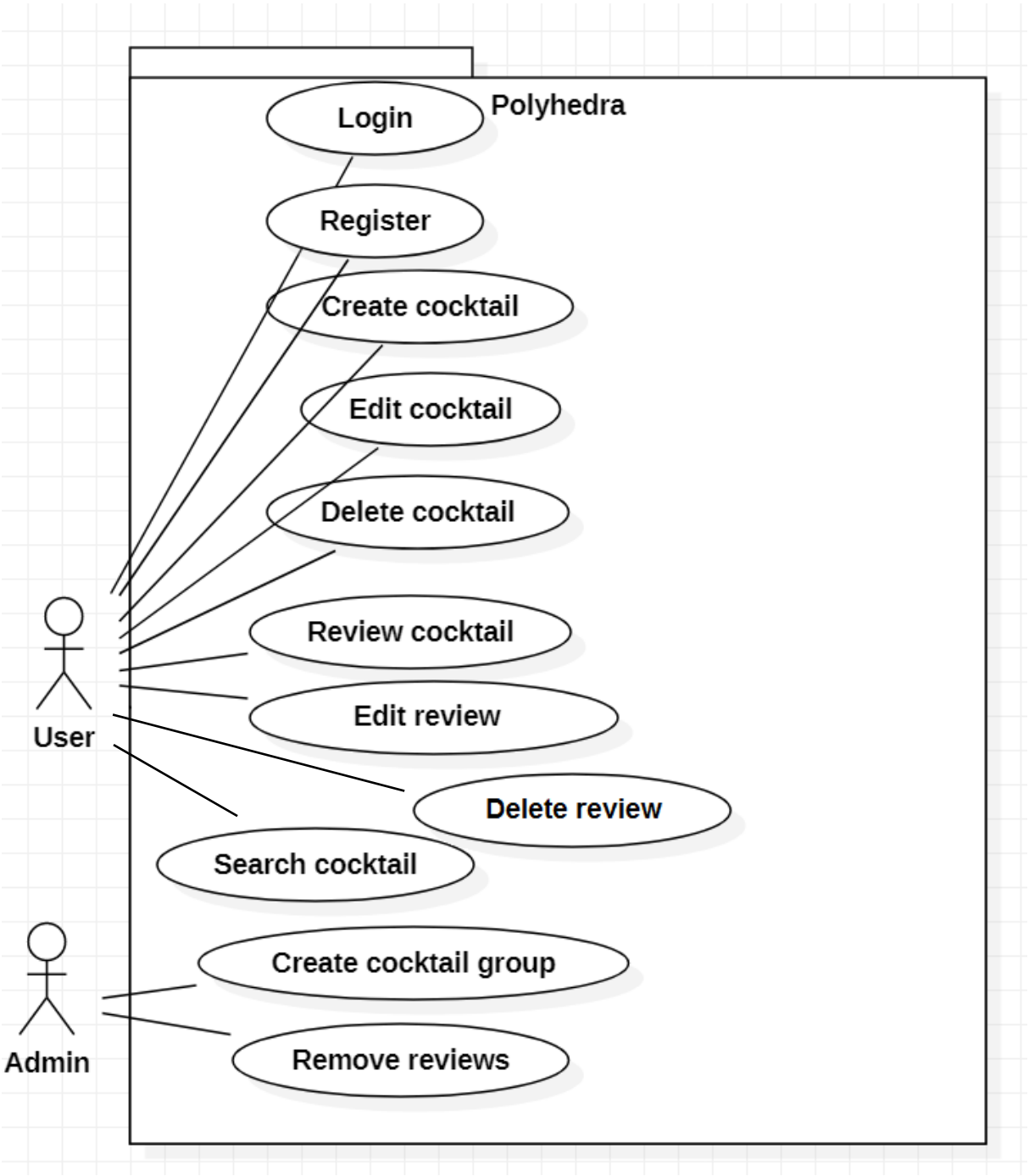

Figure 2: Use Cases

Table 1: Use Case 1

Use Case ID	1
Use Case Name	Login
Actors	Polyhedra User
Description	Users log in to the Polyhedra to create and review cocktail
Preconditions	User enter valid e-mail, username and password information to the system. User confirm the account in e-mail. User is defined in Polyhedra: user exists in the system with a valid username and password.
Postconditions	User is successfully logged in to the Polyhedra.
Normal Flow	1. User requests to enter Polyhedra system. 2. System prompts user to log in. 3. User enters his/her username and password. 4. System verifies entered information. 5. System logs user into Polyhedra system.
Exceptions	3a. If the username and/or password entered by the user is/are not valid: 1. Polyhedra system displays “Invalid username and/or password” error message. 2. Use case resumes on Step 2 of Normal Flow.

Table 2: Use Case 2

Use Case ID	2
Use Case Name	Register
Actors	Polyhedra User
Description	Users register to Polyhedra
Preconditions	None
Postconditions	User is successfully registered to Polyhedra.
Normal Flow	1. User requests to register to Polyhedra system. 2. System prompts user to register. 3. User enters his/her username, first name, last name, password and password confirmation. 4. System verifies entered information. 5. System registers user to Polyhedra.
Exceptions	3a. If the username and/or password entered by the user is/are not valid: 1. Polyhedra system displays “Invalid username and/or password” error message. 2. Use case resumes on Step 2 of Normal Flow.

Table 3: Use Case 3

Use Case ID	3
Use Case Name	Create Cocktail
Actors	Polyhedra User
Description	User creates cocktail record on the cloud environment.
Preconditions	User is defined in Polyhedra : user exists in the system with a valid username and password.
Postconditions	A cocktail record is created on the cloud environment.
Normal Flow	1. User enters cocktail name, recipe, and ingredients information into the system. 2. User selects cocktail group name from checklist. 3. User fills checkbox whether cocktail is alcoholic.(if the cocktail does not contain alcohol, user will not fill the checkbox.) 4. User clicks Save button to create a new cocktail record.
Exceptions	2.a. If user wants to create new cocktail group, user will follow the number 3 use case (use case:Create Cocktail Group) instructions. 4.a. User clicks “Cancel” button to cancel for creating a cocktail record. 4.b. User clicks “Exit” button to leave application.

Table 4: Use Case 4

Use Case ID	4
Use Case Name	Edit Cocktail
Actors	Polyhedra User
Description	User edits cocktail record on the cloud environment.
Preconditions	User is defined in Polyhedra: user exists in the system with a valid username and password.
Postconditions	A cocktail record is edited on the cloud environment.
Normal Flow	1. User edits only own record. User edits cocktail name, recipe, and ingredients information. 2. User edits cocktail group name from checklist. 3. User changes checkbox whether cocktail is alcoholic.(if the cocktail does not contain alcohol, user will not fill the checkbox.) 4. User clicks Save button to update existing cocktail record.
Exceptions	4.a. User clicks “Cancel” button to cancel for updating a cocktail record. 4.b. User clicks “Exit” button to leave application.

Table 5: Use Case 5

Use Case ID	5
Use Case Name	Delete Cocktail
Actors	Polyhedra User
Description	User deletes cocktail record on the cloud environment.
Preconditions	User is defined in Polyhedra : user exists in the system with a valid username and password. The cocktail record exists in the system.
Postconditions	A cocktail record is deleted on the cloud environment.
Normal Flow	1. User deletes only own record. User deletes existing cocktail record. 2. User clicks Delete button. 3. The system show the message “the record will be deleted. are you sure to continue?” to the user. 4. Users clicks “Continue” button. 5. Cocktail record is deleted. 6. Deleted cocktail record is saved in the system as passive record. Deleted records do not disappear from database.
Exceptions	3.a. User clicks “No” button to cancel deleting operation.

Table 6: Use Case 6

Use Case ID	6
Use Case Name	Review Cocktail
Actors	Polyhedra User
Description	User review cocktail record on the cloud environment.
Preconditions	User is defined in Polyhedra: user exists in the system with a valid username and password. The cocktail record exists in the system.
Postconditions	A cocktail record is reviewed on the cloud environment.
Normal Flow	1. User adds review note. 2. User gives score to taste of the drink. 3. User gives score to cost of the drink. 4. User gives score to preparation difficulty of the drink. 5. User clicks Save button.
Exceptions	4.a. User clicks “Cancel” button to cancel for reviewing a cocktail record. 4.b. User clicks “Exit” button to leave application.

Table 7: Use Case 7

Use Case ID	7
Use Case Name	Edit Cocktail Review
Actors	Polyhedra User
Description	User edits cocktail review record on the cloud environment.
Preconditions	User is defined in Polyhedra: user exists in the system with a valid username and password. The cocktail record exists in the system. The review record exists in the system.
Postconditions	A cocktail review record is edited on the cloud environment.
Normal Flow	1. User edits review note. 2. User edits score of taste of the drink. 3. User edits score of cost of the drink. 4. User edits score of preparation difficulty of the drink. 5. User clicks Save button.
Exceptions	4.a. User clicks “Cancel” button to cancel for reviewing a cocktail record. 4.b. User clicks “Exit” button to leave application.

Table 8: Use Case 8

Use Case ID	8
Use Case Name	Delete Cocktail Review
Actors	Polyhedra User
Description	User deletes cocktail review record on the cloud environment.
Preconditions	User is defined in Polyhedra: user exists in the system with a valid username and password. The cocktail record exists in the system. The review record exists in the system.
Postconditions	A cocktail review record is deleted on the cloud environment.
Normal Flow	1. User clicks “Delete My Review” button. 2. The system deletes review data from DB.
Exceptions	1.a. User clicks “Cancel” button to cancel for reviewing a cocktail record. 1.b. User clicks “Exit” button to leave application.

Table 9: Use Case 9

Use Case ID	9
Use Case Name	Search Cocktail
Actors	Polyhedra User
Description	User searches for cocktail in Polyhedra.
Preconditions	None
Postconditions	A cocktail is searched.
Normal Flow	1. User enters search query (at least 3 letters) 2. System receives search query 3. System shows cocktails matching search query
Exceptions	1.a. User enters null (less than 3 letters) search query. 1.b. System informs users to input at least 3 letters.

Table 10: Use Case 10

Use Case ID	10
Use Case Name	Create Cocktail Group
Actors	Polyhedra Admin
Description	User creates cocktail group record on the cloud environment.
Preconditions	User is defined in Polyhedra: user exists in the system with a valid username and password.
Postconditions	A cocktail group record is created on the cloud environment.
Normal Flow	1. User enters cocktail group name into the system. (for example pre dinner cocktail, after dinner cocktail, long drink cocktail etc.) 2. User clicks Save button to create a new cocktail category record.
Exceptions	2.a. User clicks “Cancel” button to cancel for creating a cocktail category record. 2.b. User clicks “Exit” button to leave application.

Table 11: Use Case 11

Use Case ID	11
Use Case Name	Remove Review
Actors	Polyhedra Admin
Description	User removes cocktail review record on the cloud environment.
Preconditions	User is defined in Polyhedra: user exists in the system with a valid username and password.
Postconditions	A cocktail review record is removed on the cloud environment.
Normal Flow	1. User clicks “Remove Review” button. 2. The system deletes review data from DB.
Exceptions	1.a. User clicks “Cancel” button to cancel for reviewing a cocktail record. 1.b. User clicks “Exit” button to leave application.

3.2 Composition Viewpoints

UML Component Diagram is chosen to represent the composition viewpoints. The diagram given in Figure 3 allows to view the bigger picture of the system by reducing complexity and giving more abstract view of the system. Three main components, DJANGO web framework, code package and PostGreSql, database that is operational in Heroku Cloud Platform. Users only interact with website. Database has two-way interaction between DJANGO web framework. Web application utilizes external resource API in order to receive cocktail images.

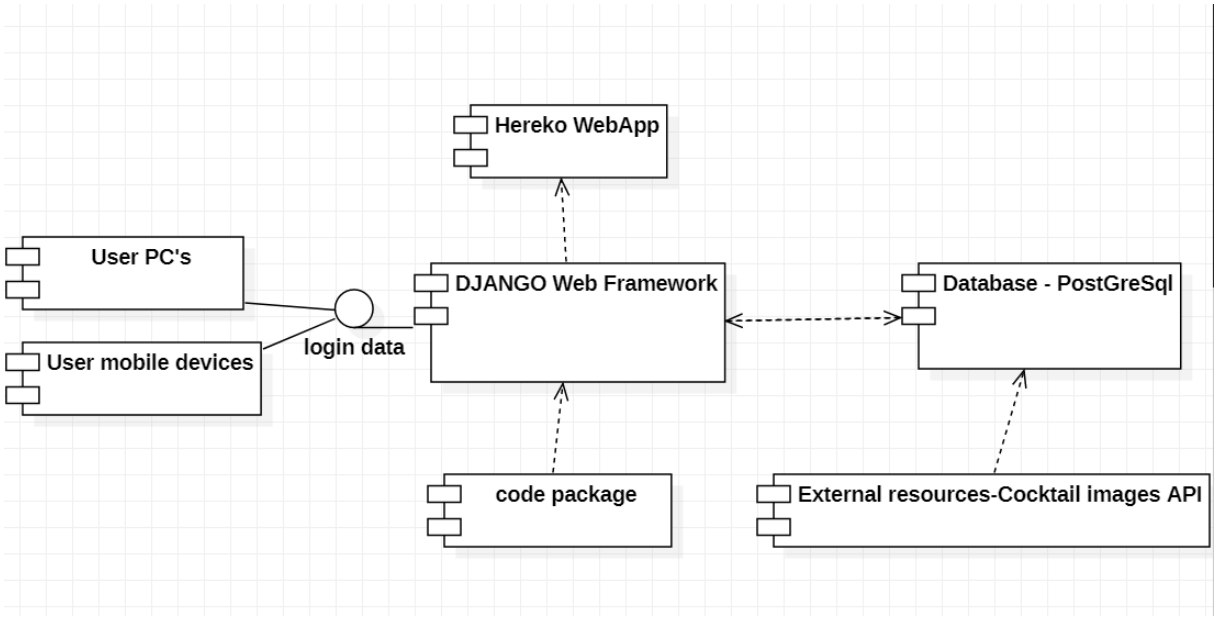

Figure 3: Component Diagram

3.3 Logical Viewpoints

Logical viewpoints of Polyhedra is represented using a UML class diagram which is given in Figure 5.

Two main actors of the system (user and admin) are the two main classes of the system. In addition, separate classes exist for cocktails, cocktail groups, cocktail reviews and user registration form. Screenshot of the user registration form is given in Figure 4.

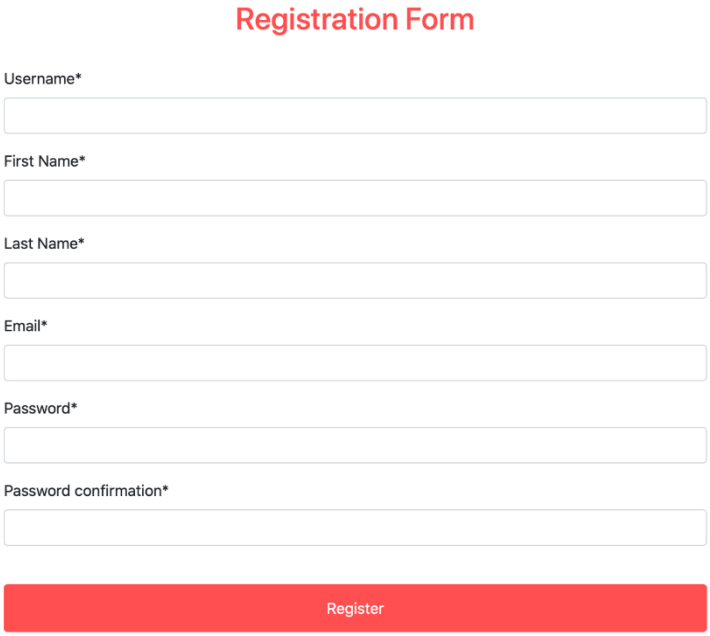

Figure 4: User Registration Form

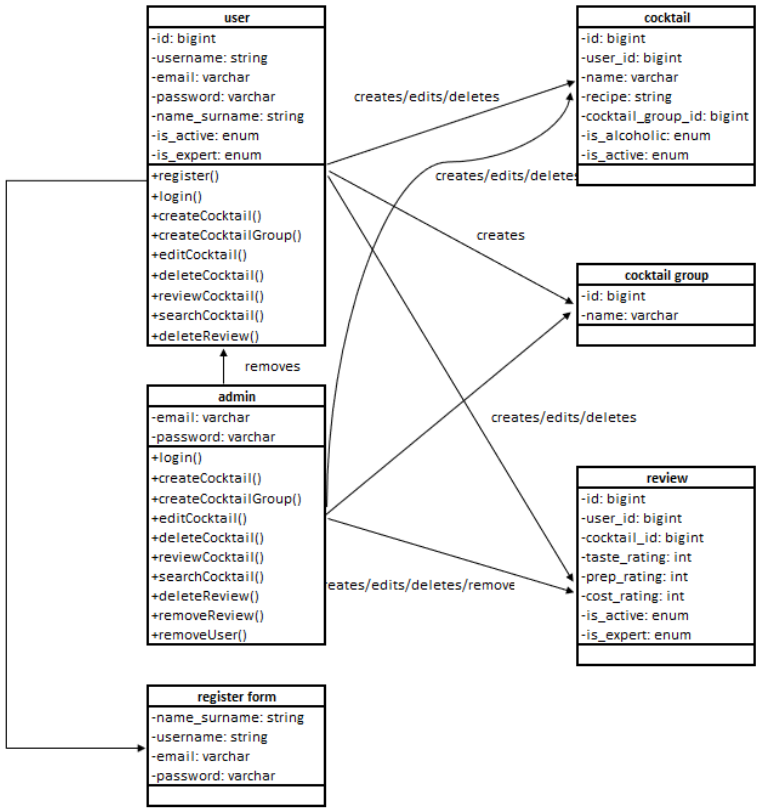

Figure 5: Class Diagram

3.4 Information Viewpoints

Entity relationship diagram for Polyhedra is given in Figure 6.

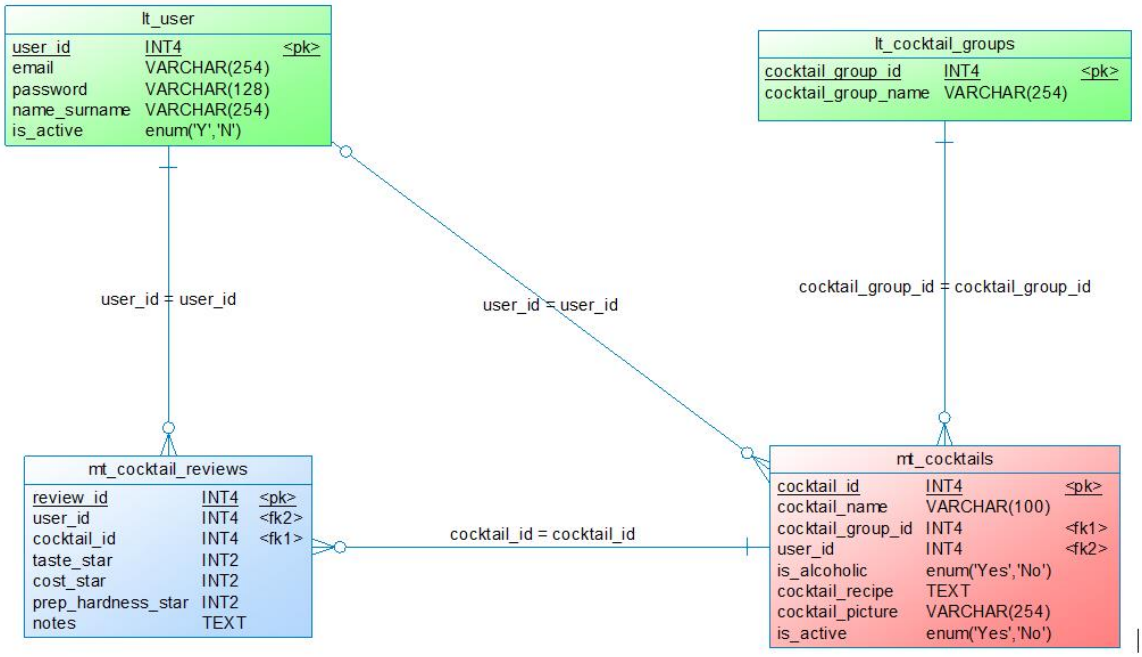

Figure 6: Entity Relationship Diagram

3.5 Patterns Viewpoints

In this section of SDD, which design pattern is opted and how it is used in Polyhedra are stated. In Polyhedra, MTV (Model-Template-View) pattern is preferred due to its suitability for developing web application for the business logic of Polyhedra.

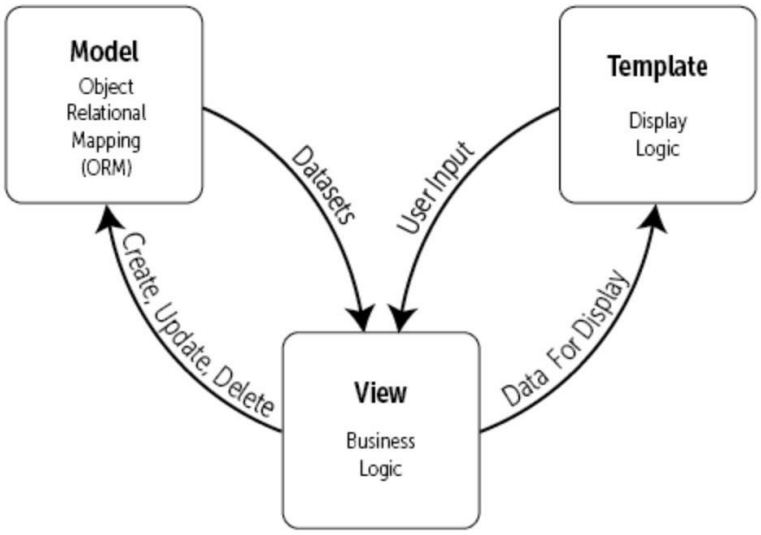

Figure 7: MVT Pattern [2]

MTV has three main components which are model, template and view which are briefly explained in the following:

- Model: It is the logical data structure behind the application and responsible for maintaining the data.
- View: It executes the business logic by fetching the data from the model and rendering the template.
- Template: It is the presentation layer which manages the complete user interface part

3.6 Interface Viewpoints

Initially, the database is created using TheCocktailDB JSON API (available at https://www.thecocktaildb.com/api.php) which includes thumbnails, ingredients, recipes and categories of 592 cocktails. Then, database is enriched by users through user interfaces. Deployment diagram to summarize interface viewpoints is given in Figure 8.

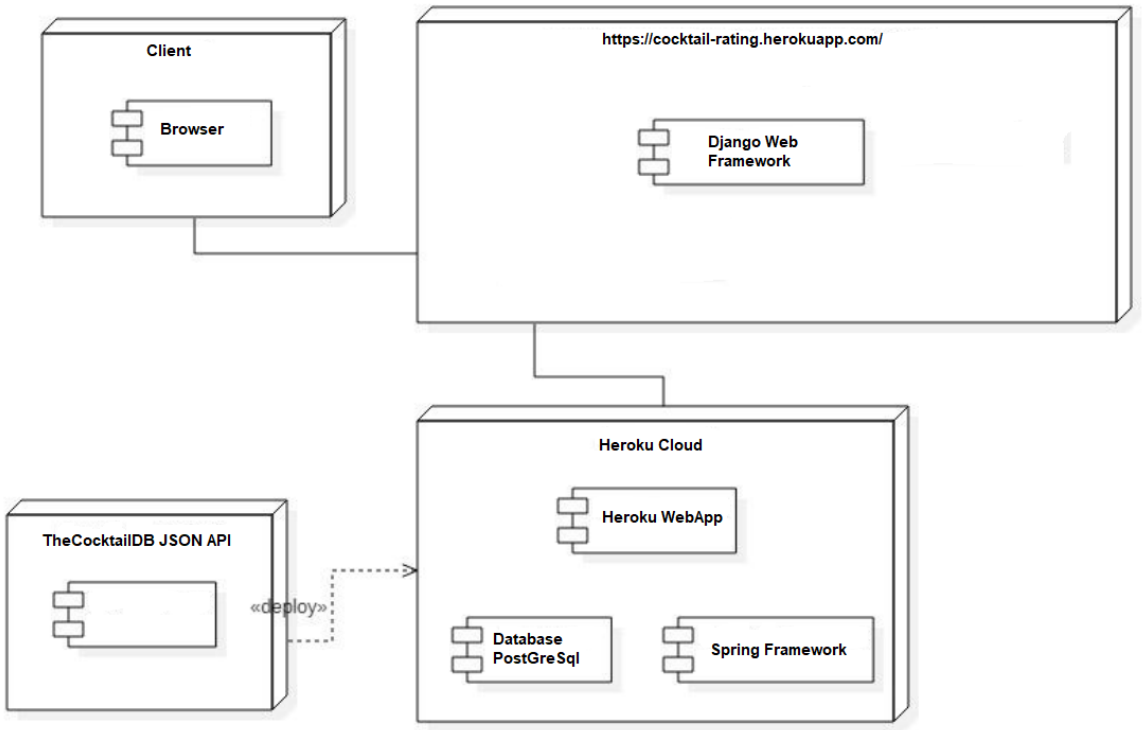

Figure 8: Deployment Diagram

3.7 Interaction Viewpoints

The interaction between system objects and user are explained in interaction viewpoint section of SDD, one by one. Therefore, sequence diagrams which are given in Figure 9 to Figure 14 are used in order to visualise the data and the workflow.

3.7.1 Create Cocktail Sequence Diagram

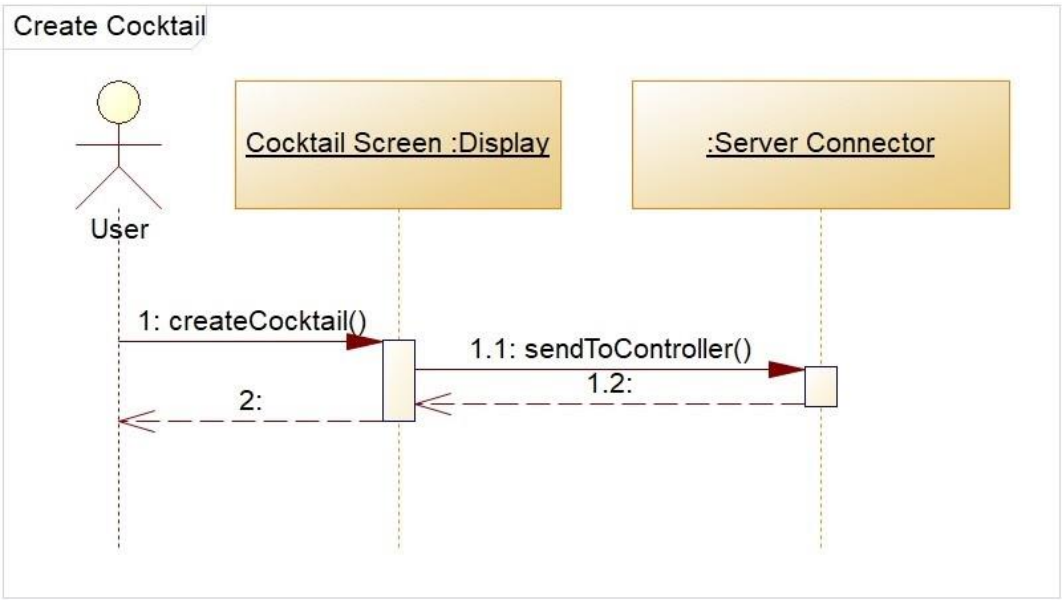

Figure 9: Create Cocktail Sequence Diagram

3.7.2 Create Cocktail Group Sequence Diagram

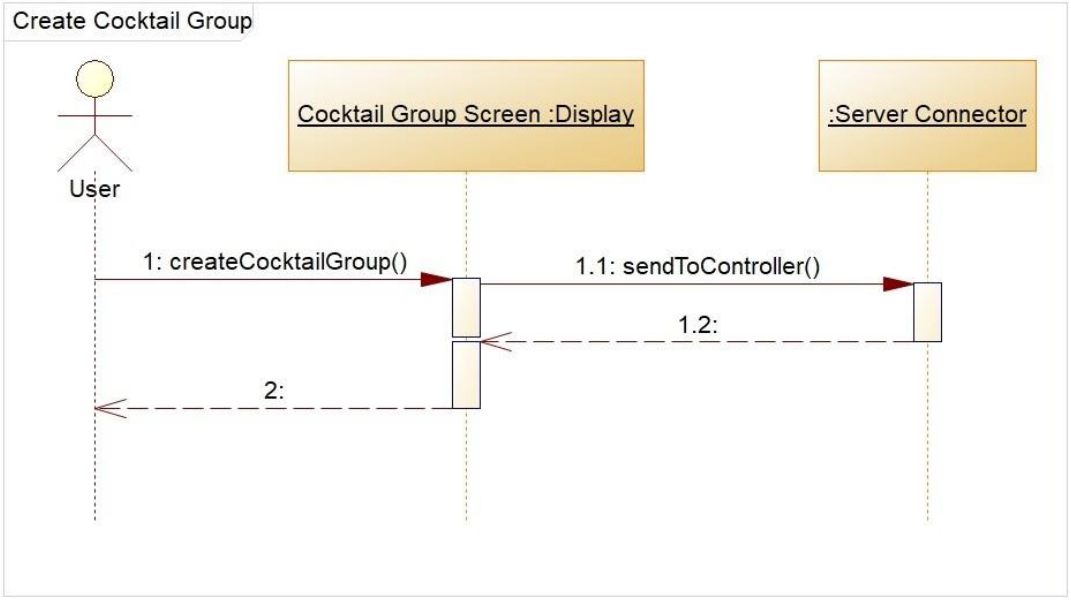

Figure 10: Create Cocktail Group Sequence Diagram

3.7.3 Delete Cocktail Sequence Diagram

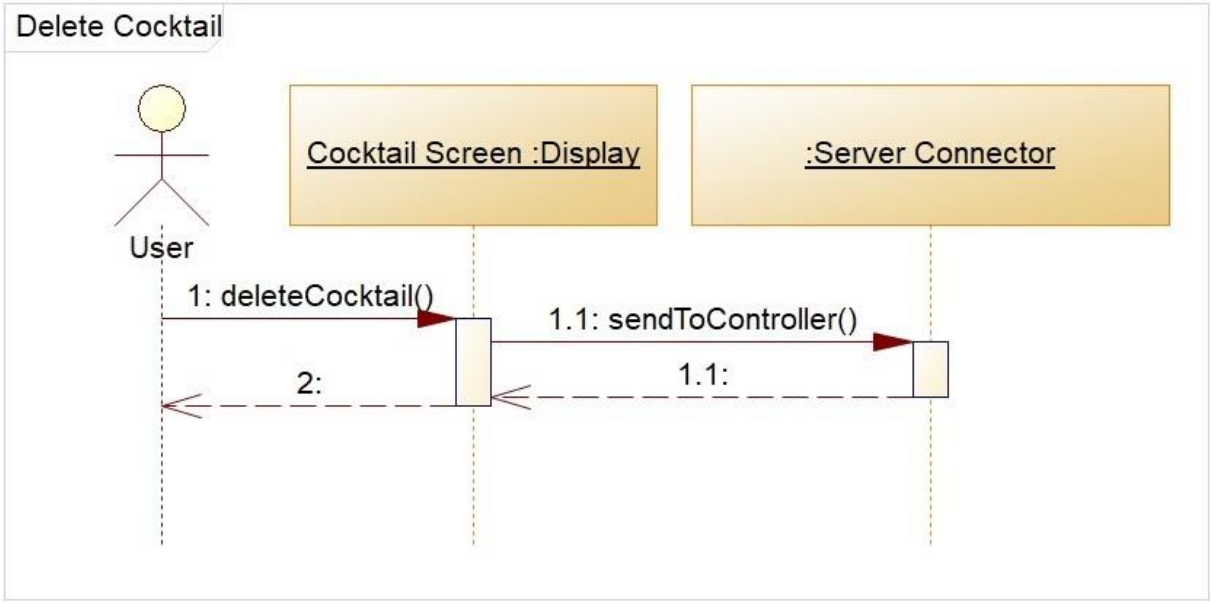

Figure 11: Delete Cocktail Sequence Diagram

3.7.4 Edit Cocktail Sequence Diagram

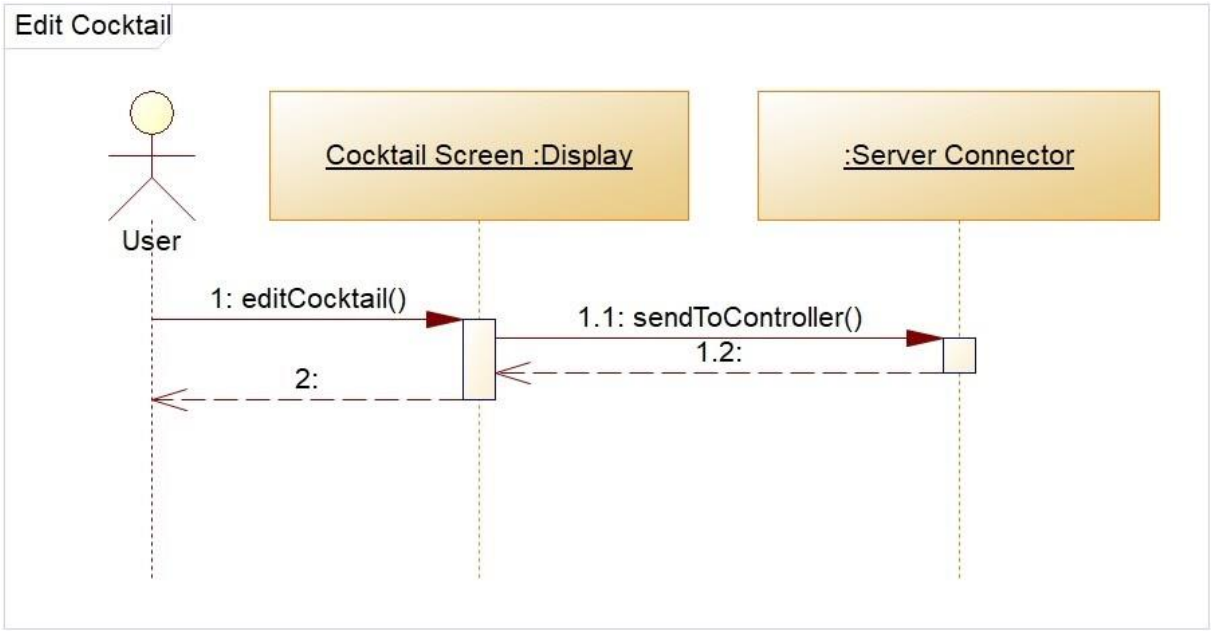

Figure 12: Edit Cocktail Sequence Diagram

3.7.5 Login Sequence Diagram

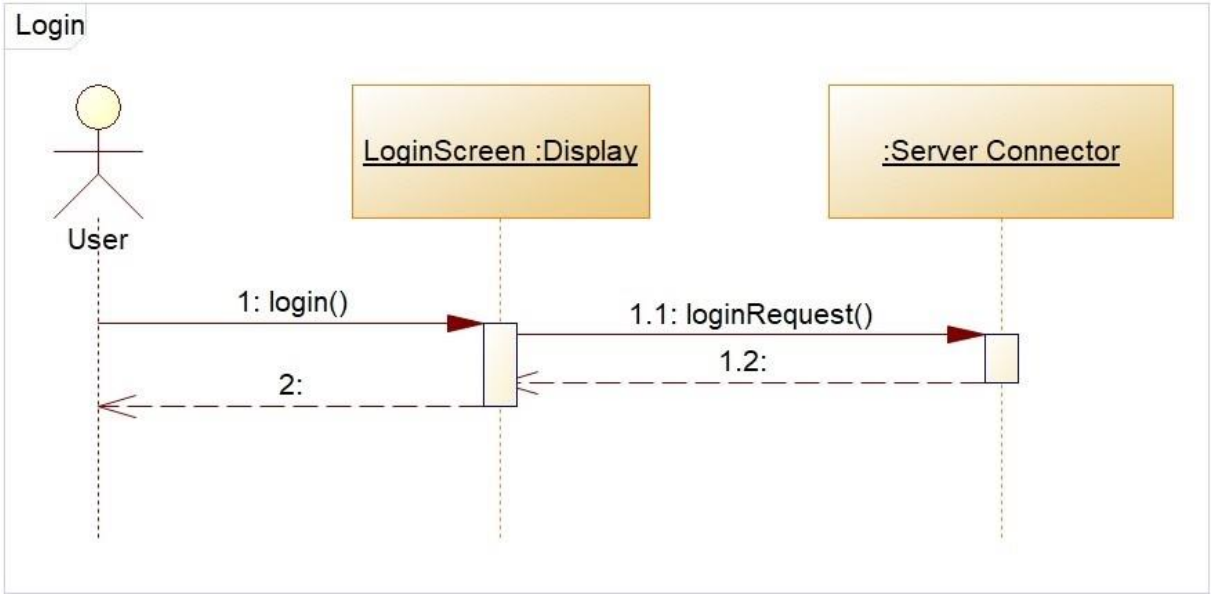

Figure 13: Login Sequence Diagram

3.7.6 Review Cocktail Sequence Diagram

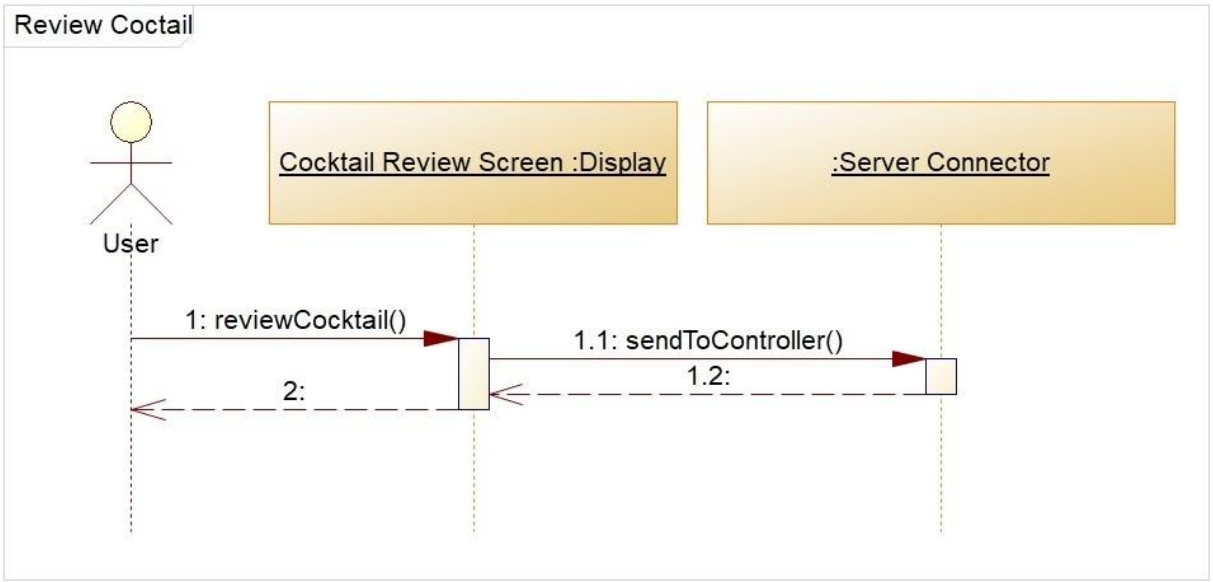

Figure 14: Review Cocktail Sequence Diagram