Software Engineering course Project

AKFA University

Software Design Specification (SDS)

For

“UzWallet”

Version _____

Timur Urunbaev
Bobir Ibragimov
Elbek Erkinboev
Ravshan Zaripov
Asadbek Fazliddinov
Samandar Kambaraliev

Muhammad Irshad-Nazeer
21-12-2022

Project Code	Not applicable
Supervisor	Muhammad Irshad-Nazeer
Project Manager	Timur Urunbaev
Project Team	Samandar Kambaraliev Ravshan Zaripov Asadbek Fazliddinov Elbek Erkinboev Bobir Ibragimov
Submission Date	21-12-2022

Software Design Specification for UzWallet
AKFA University, 2022

1 Document Management **2**
1.1 Contributors 3
1.2 Version Control 3

2 Overview **4**

3 Development Tools And Standards **4**
3.1 Development Tools 4
3.2 Development Standards 4

4 System Processes **4**

5 User Interfaces **4**
5.1 Transactional Interface 5
5.2 Reporting Interface 5

6 Application Security **5**
6.1 Authentication 5
6.2 Authorisation 5
6.3 Business Objects 5

7 Database Design **5**

8 Application Interfaces **5**

9 Data **5**
9.1 Data Migration 5
9.2 Archiving Policy 5

10 Implementation **5**

1 Document Management

When completing this document, please mark any section that is not required as 'N/A'. A brief description of why the section is not required should also be included.

1.1 Contributors

Please provide details of all contributors to this document.

Role in our company	Unit	Name
System Analyst Designer (Owner)		Muhammad Irshad-Nazeer
Business Analyst		Kambaraliev Samandar
Project Manager		Urunbaev Timur
Project Sponsor		Ibragimov Bobir
Business Area Manager		Zaripov Ravshan
Lead Tech Developer		Fazliddinov Asadbek
Designer		Erkinboev Elbek

1.2 Version Control

Please document all changes that made to this document since document distribution.

Date	Version	Author	Section	Amendment
10.12.2022	1.0	U. Timur	docs	Created new repository
10.12.2022	1.1			
10.12.2022	1.2			

2 Overview

This product simulates a cardholder, where we keep our discount, membership, or loyalty cards. The application will keep all your LC on your smartphone and have access to them at any time. Also, users can open a new LC by choosing any store which provides a loyalty program. The feature will be implemented using the profile information, sending user credentials directly to issuing store. Moreover, advertisements and promotions appear in these stores, and users will be notified if they subscribe and turn on notifications.

- Template Application (Кошелек)
- Determine style and pattern
- Hardware + Software communications
- How system works

Architecture Pattern - MVC
Architecture Style -
Layers:

- Presentation Layer (UI/UX)
- Business Layer (Security, Data caching, logging, data validation, exception management)
- Data Layer (Persistence [API], Network)

3 Development Tools And Standards

3.1 Development Tools

- Language: Dart + Flutter framework
- Data Base: MySQL
- API platform: Postman
- Version control: Git, Github
- Design Tool: Figma

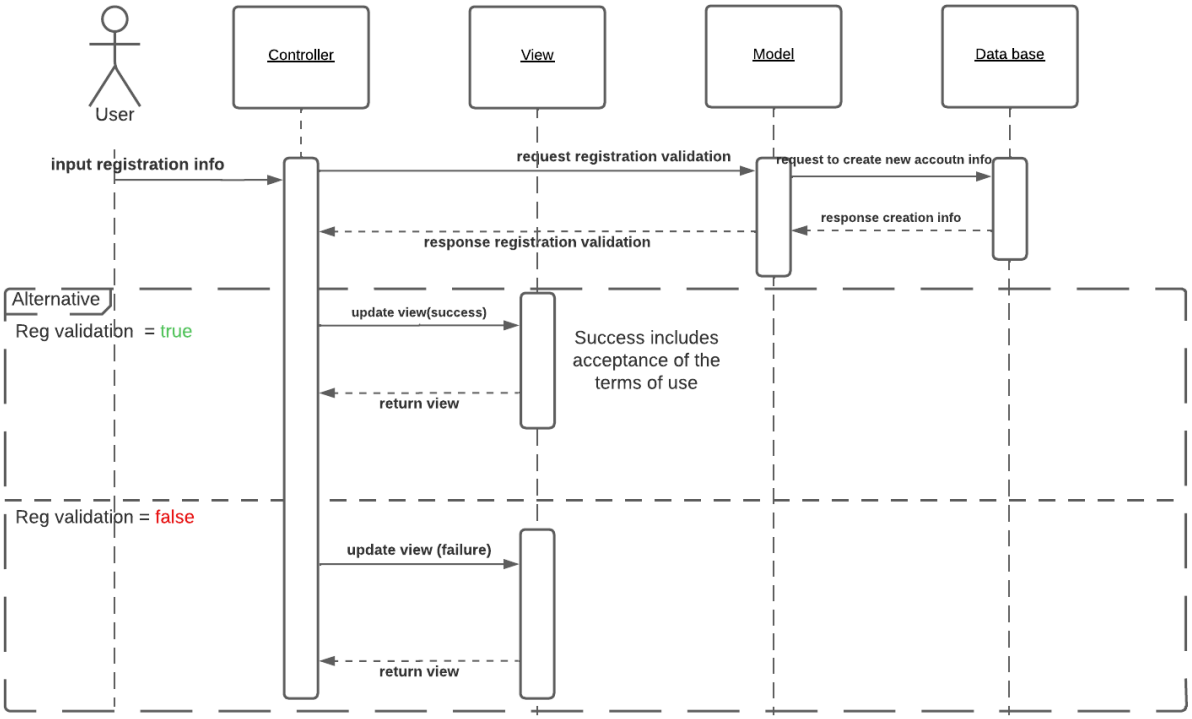
3.2 Development Standards

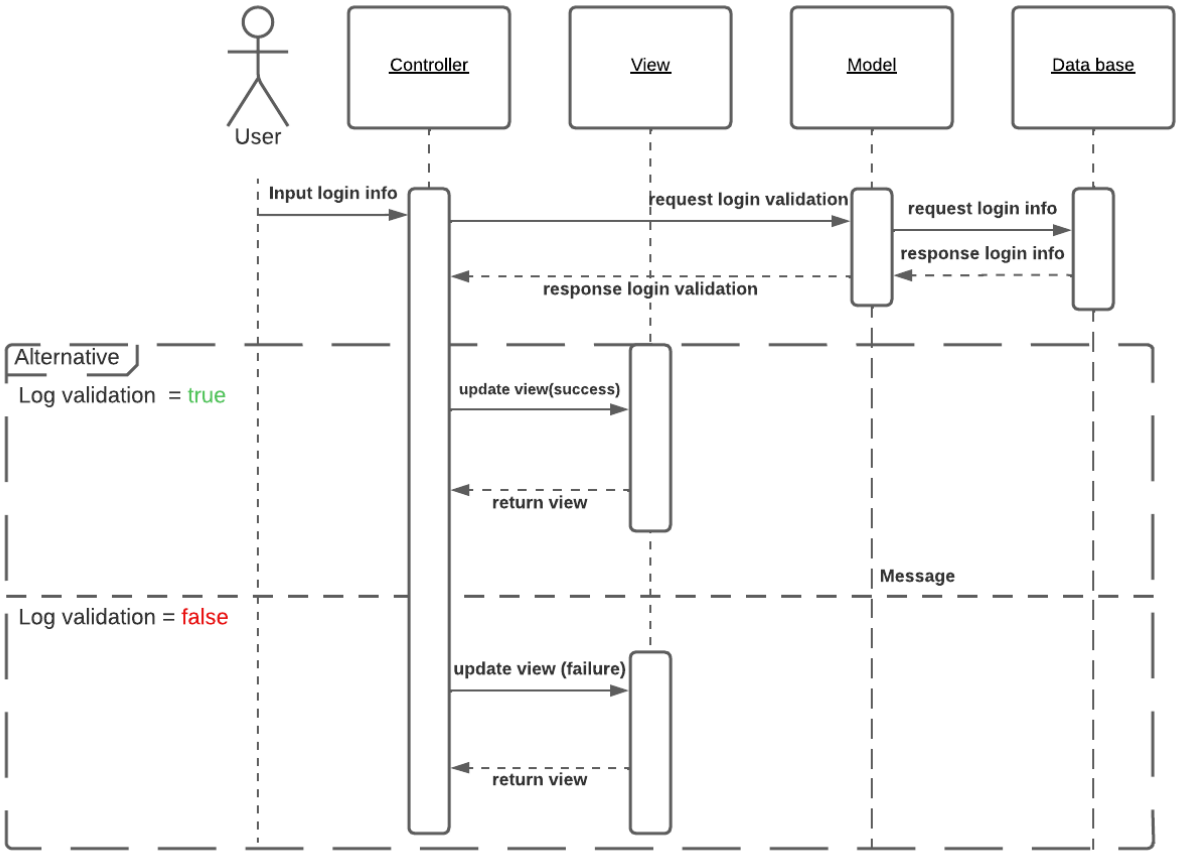
Standards being followed for this application.

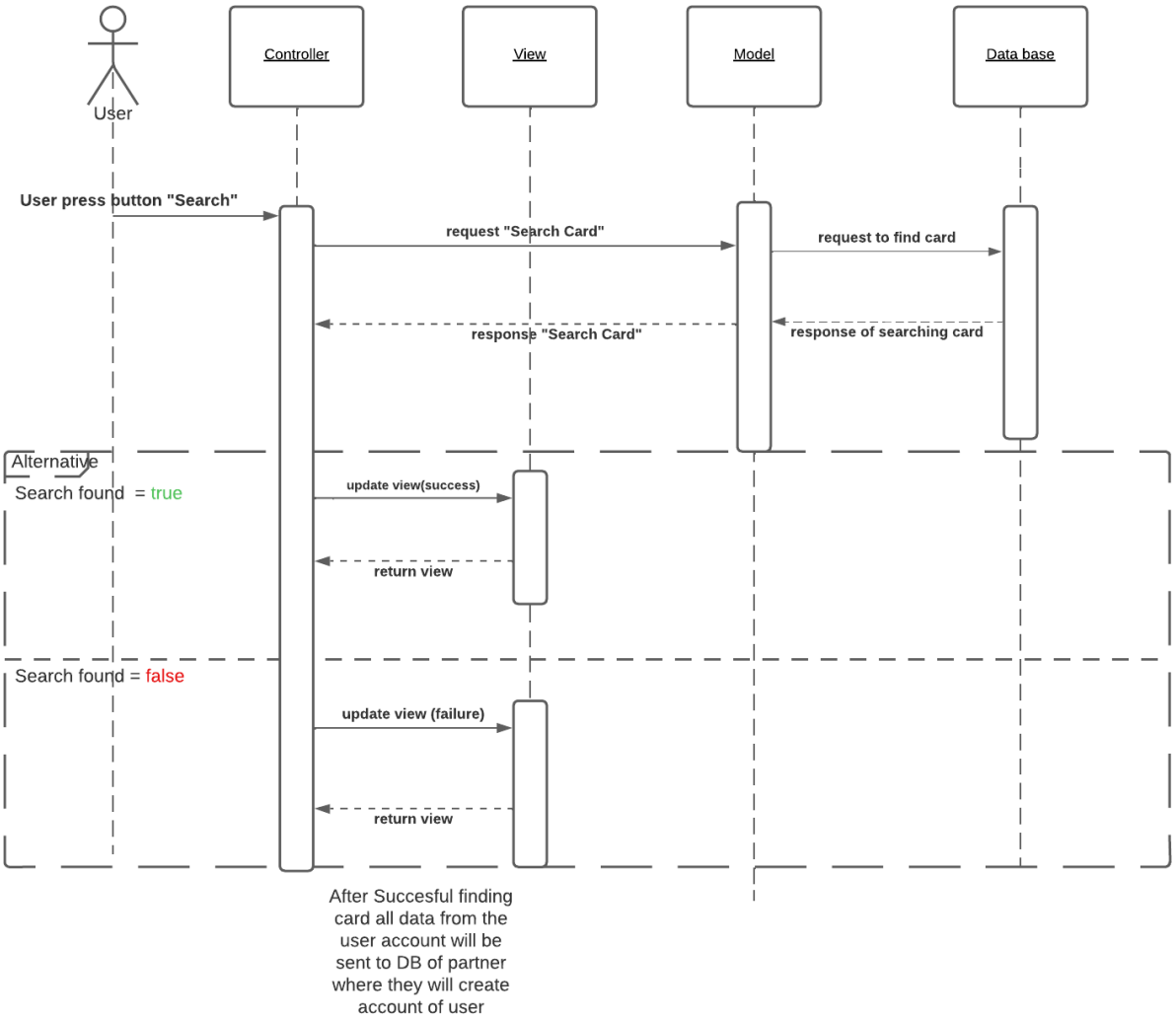
Standards:

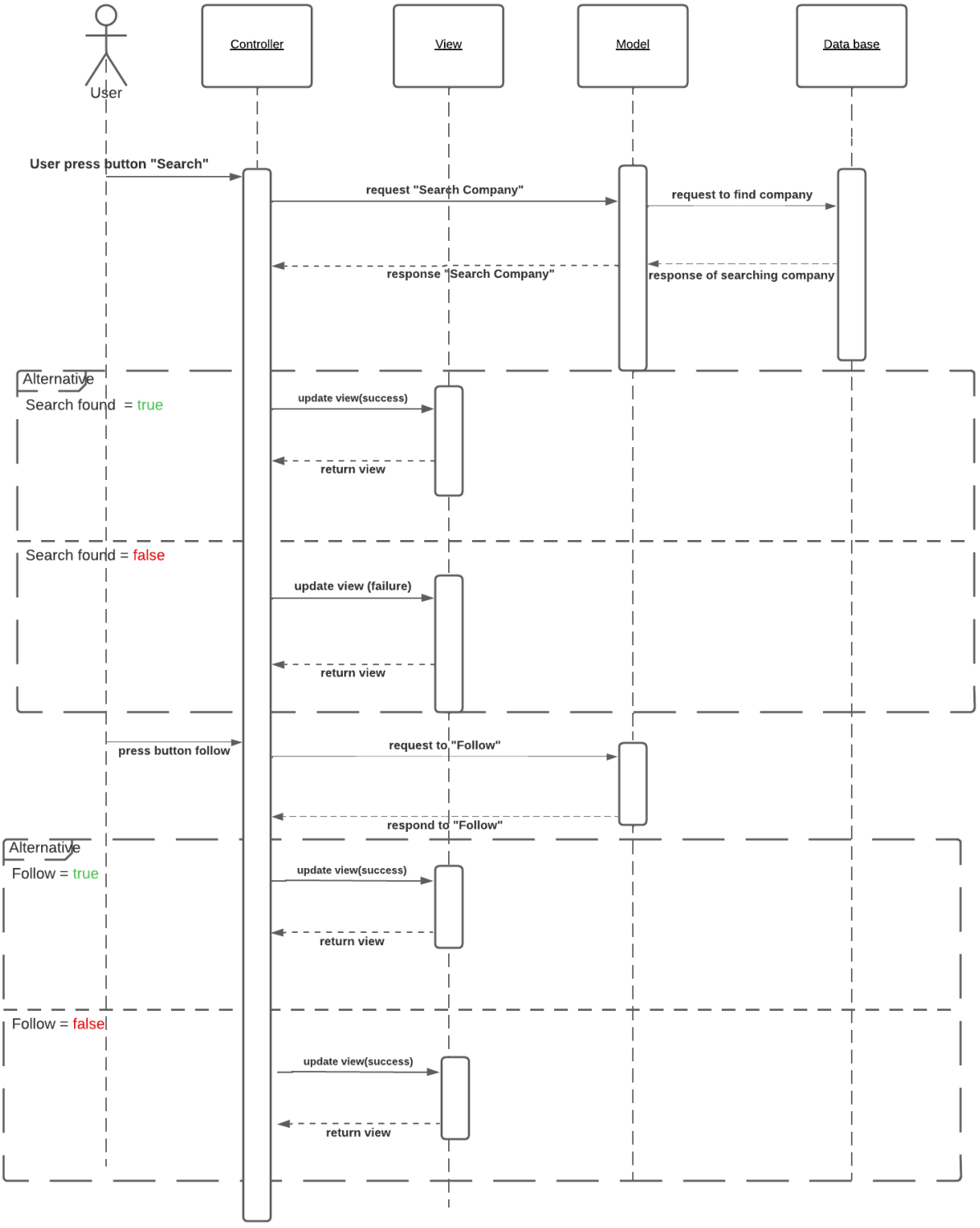
- ☑ Database Design
- ☑ Flutter
- ☑ Postman
- ☑ Accessibility
- ☑ Web Style Standards
- ☑ Supported Web Browser

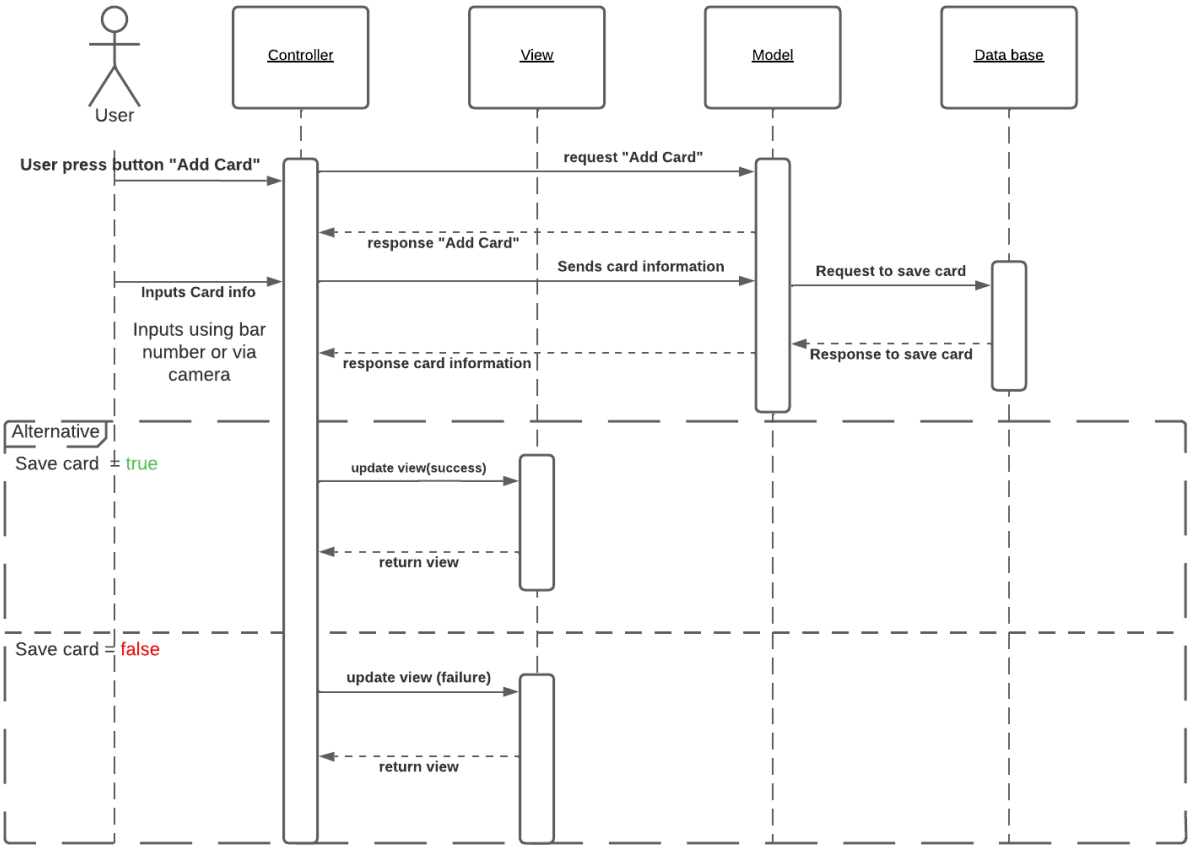
4 System Processes

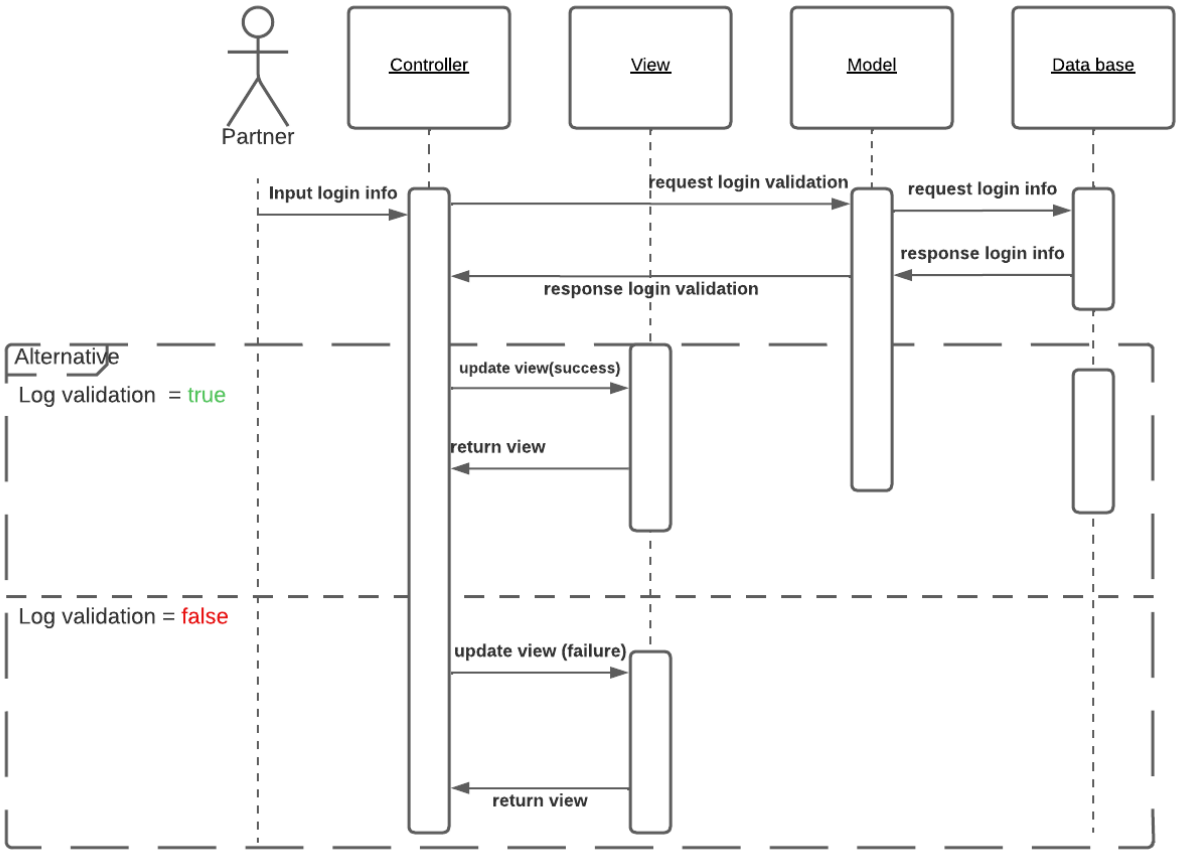


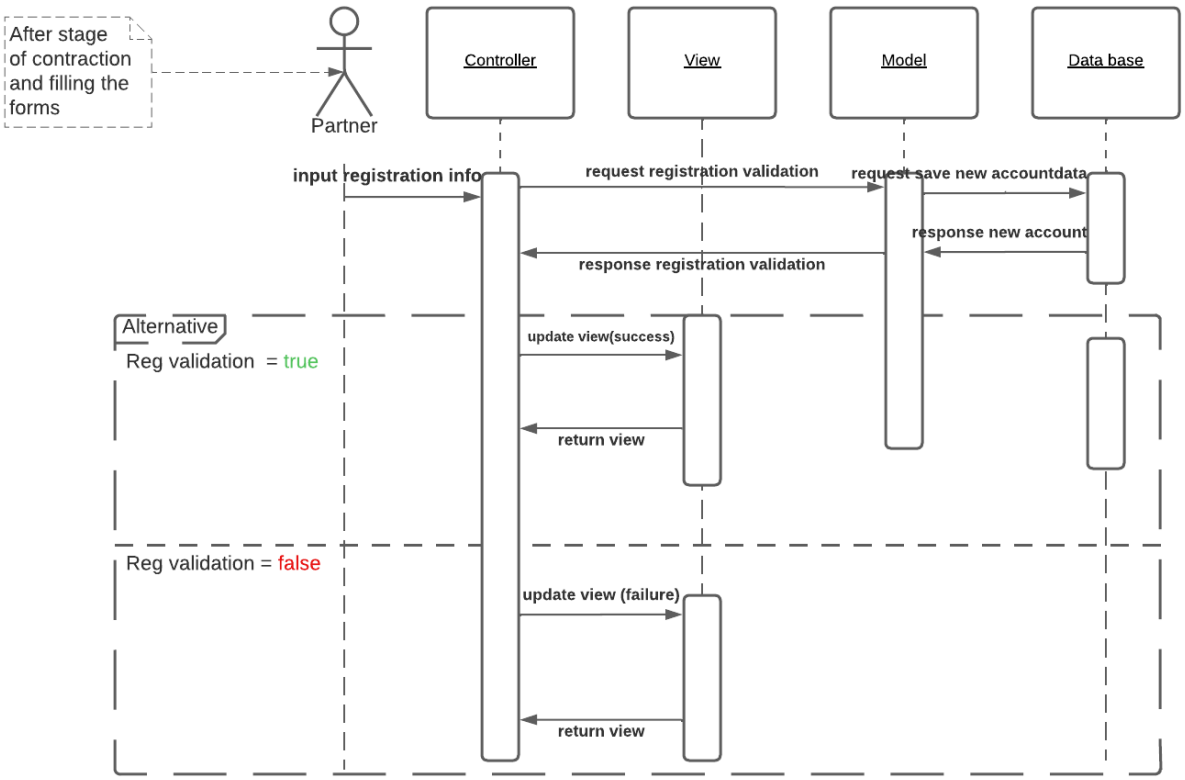

User
Controller
View
Model
Data base
Input login info
request login validation
request login info
response login info
response login validation
Alternative
Log validation = true
update view(success)
return view
Message
Log validation = false
update view (failure)
return view

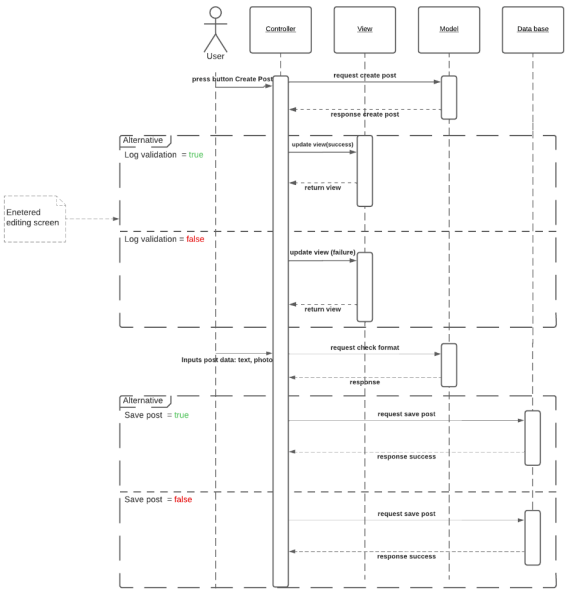

User
Controller
View
Model
Data base
User press button "Search"
request "Search Card"
request to find card
response "Search Card"
response of searching card
Alternative
Search found = true
update view(success)
return view
Search found = false
update view (failure)
return view
After Succesful finding card all data from the user account will be sent to DB of partner where they will create account of user

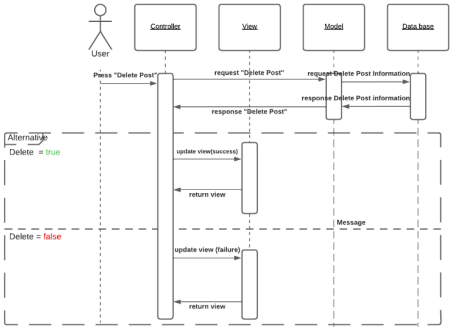

User
Controller
View
Model
Data base
User press button "Search"
request "Search Company"
request to find company
response "Search Company"
response of searching company
Alternative
Search found = true
update view(success)
return view
Search found = false
update view (failure)
return view
press button follow
request to "Follow"
respond to "Follow"
Alternative
Follow = true
update view(success)
return view
Follow = false
update view(success)
return view


User
Controller
View
Model
Data base
User press button "Add Card"
request "Add Card"
response "Add Card"
Sends card information
Request to save card
Inputs Card info
Inputs using bar number or via camera
response card information
Response to save card
Alternative
Save card = true
update view(success)
return view
Save card = false
update view (failure)
return view


Partner
Controller
View
Model
Data base
Input login info
request login validation
request login info
response login info
response login validation
Alternative
Log validation = true
update view(success)
return view
Log validation = false
update view (failure)
return view







5 User Interfaces

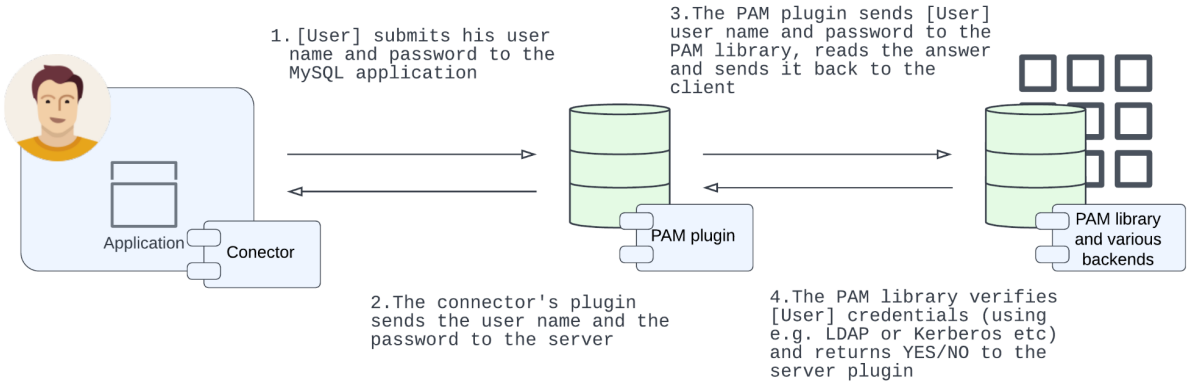
5.1 Transactional Interface

5.2 Reporting Interface

6 Application Security

6.1 Authentication

By default, MySQL uses the built-in mysql_native_password authentication plugin, which performs authentication using the native password hashing method.



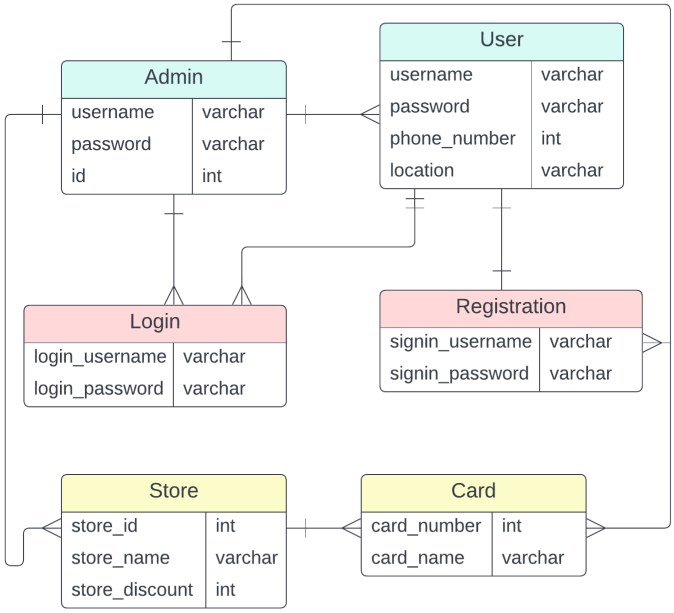
6.2 Authorisation

Once a client's credentials are authenticated, MySQL establishes a connection and then enters the second part of the access control system to determine authorization. MySQL authorization is an ongoing process that checks each command against the user account's specific privileges. If the commands fall within the scope of the user's privileges, the action is allowed. If not, the server denies the request.

6.3 Business Objects

7 Database Design

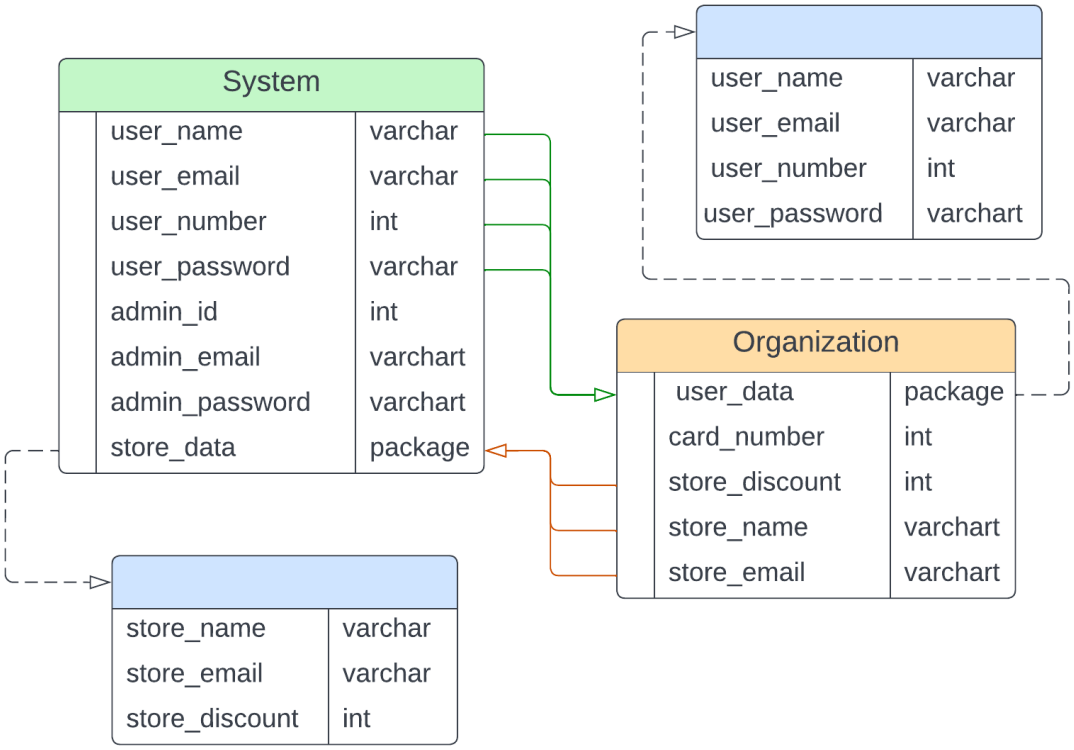
7.1 Entity-Relationship Diagram of system



8 Application Interfaces

9 Data

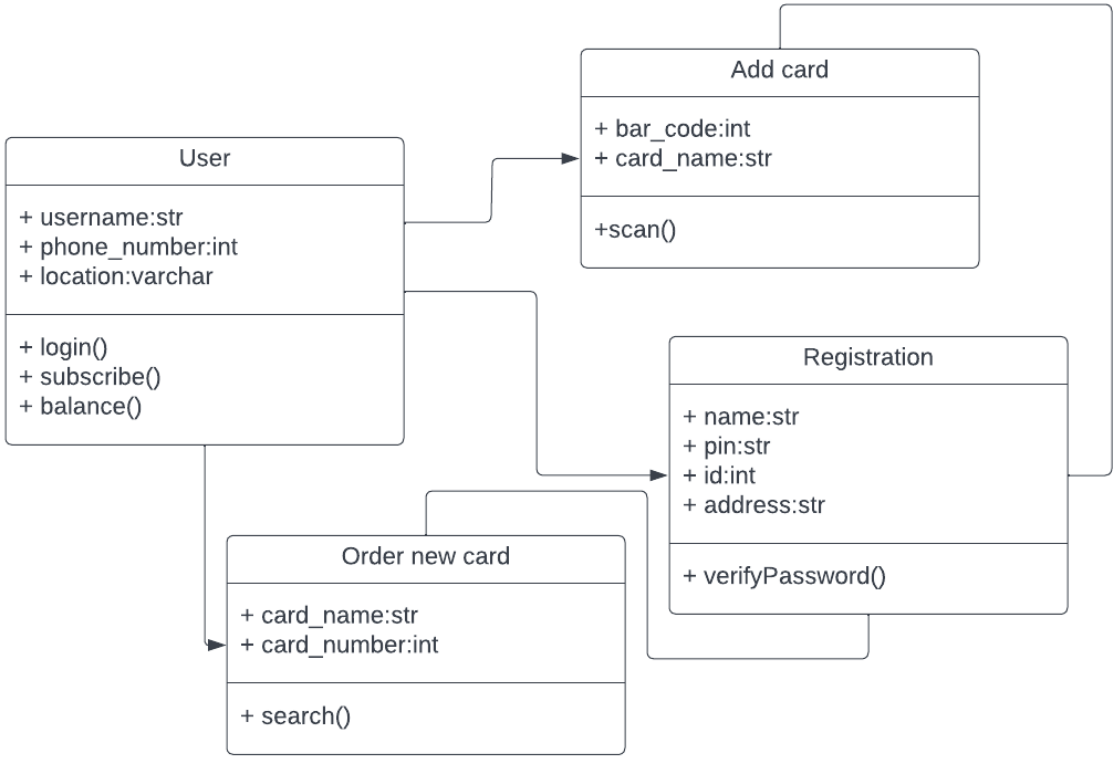
9.1 Data Migration

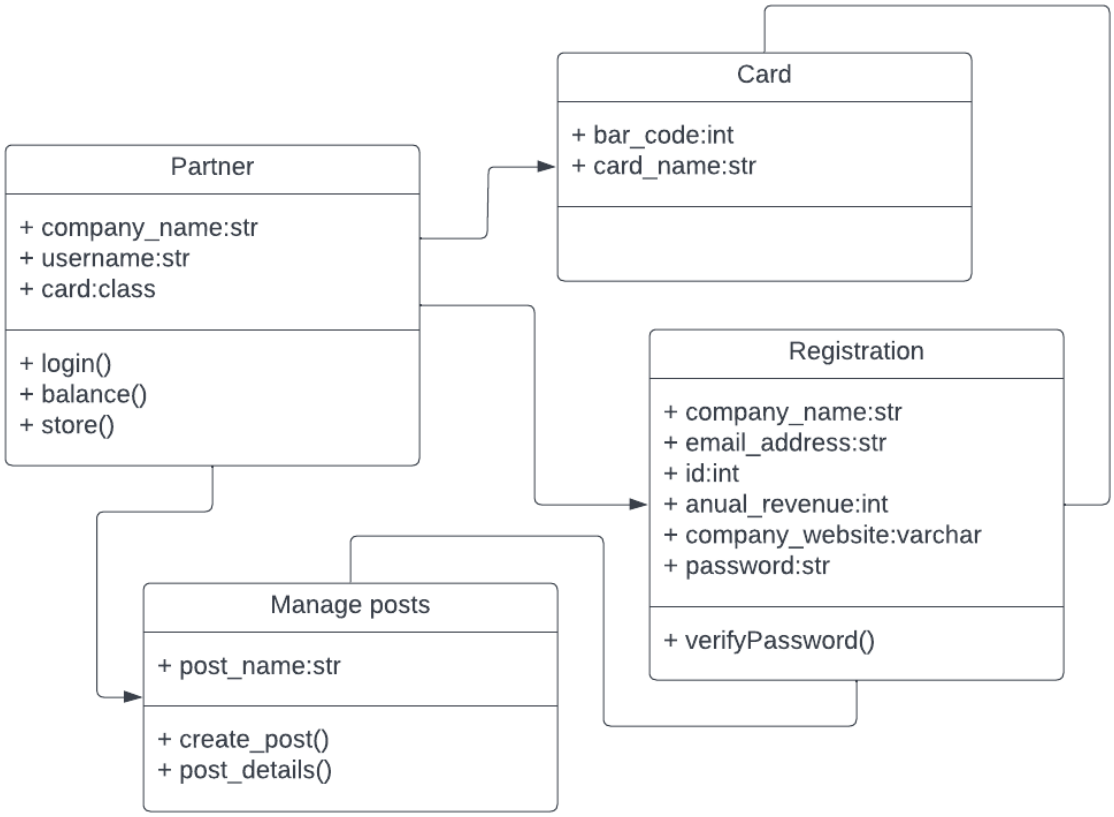


9.2 Archiving Policy

10 Implementation

We do not implement the code, because of assignment description, consequently we decided to not describe something that is not done.


Add card
+ bar_code:int
+ card_name:str
+scan()
User
+ username:str
+ phone_number:int
+ location:varchar
+ login()
+ subscribe()
+ balance()
Registration
+ name:str
+ pin:str
+ id:int
+ address:str
+ verifyPassword()
Order new card
+ card_name:str
+ card_number:int
+ search()


Card
+ bar_code:int
+ card_name:str
Partner
+ company_name:str
+ username:str
+ card:class
+ login()
+ balance()
+ store()
Registration
+ company_name:str
+ email_address:str
+ id:int
+ anual_revenue:int
+ company_website:varchar
+ password:str
+ verifyPassword()
Manage posts
+ post_name:str
+ create_post()
+ post_details()