

IT - 314 Software Engineering

LAB - 6

Group - 21

Renting System

Domain analysis models

Identify boundary, entity, control object.

Boundary Objects

These would include those objects which serve as an interface between the entities. It would include the following objects:

- UserRegistrationForm
It would be used to collect the details of a new user when they are signing up on the website.
- Authentication Interface
It would be used for authenticating the user when they try to login.
- Payment Gateway
It would be used for processing the payments when a borrower is paying the lender for the product

Entity Objects

- Borrower
 - It includes the details of those users who are using the website for borrowing products.
Attributes : First Name
Last Name
Email Id
Password
Contact Number

Address
Products borrowed

- Lender

- It includes the details of those users who are lending their products on the website.

Attributes : First Name
Last Name
Email Id
Password
Contact Number
Address
Products lent

- Product

- It includes the details of the products listed on the website which are both currently borrowed and are available to borrow.

Attributes : Name
Product ID
Description
Price
Image
Category
Availability
Purchase Date
Duration

- Order

- It includes the details of the products currently purchased or purchased in the past .

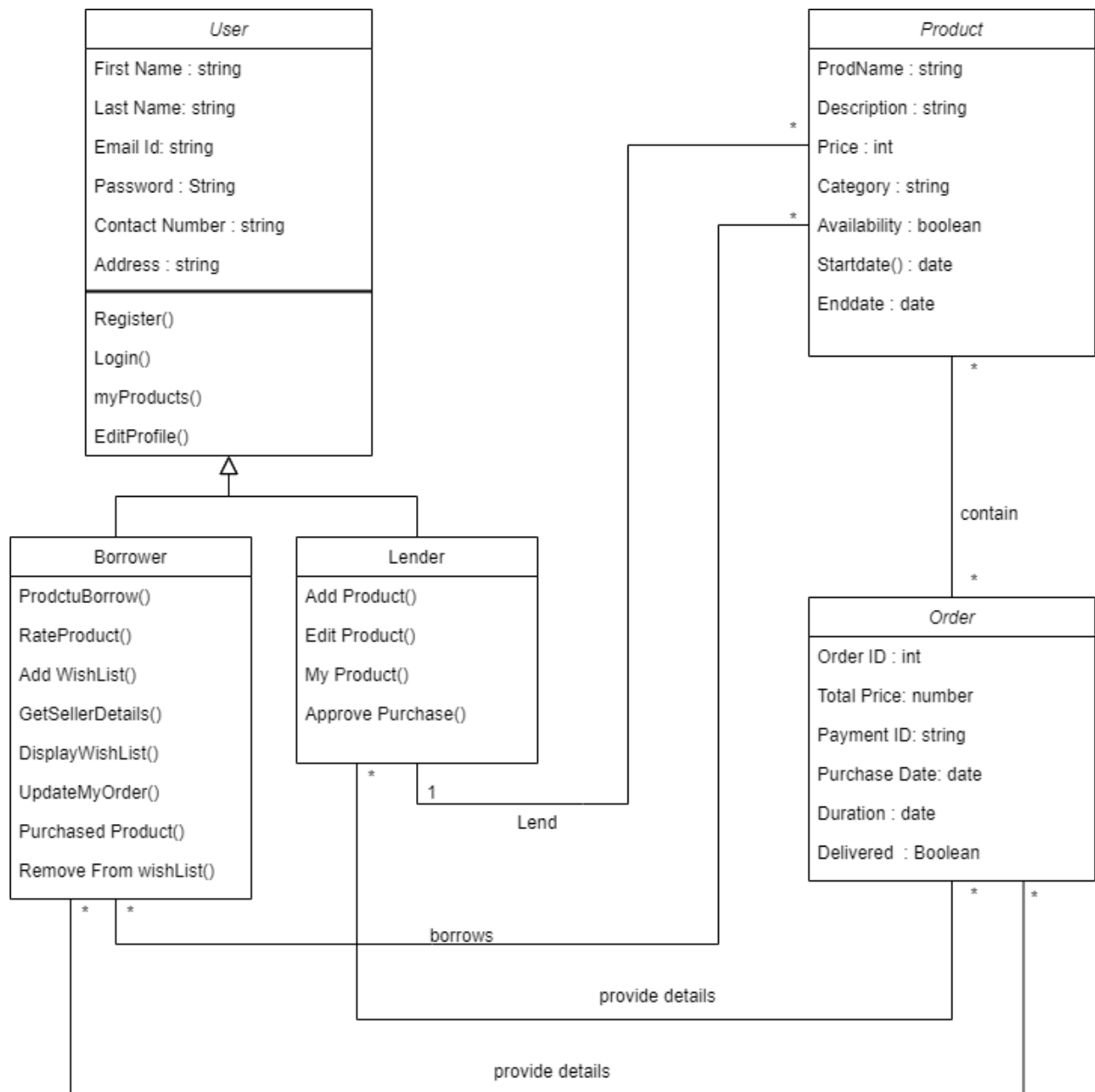
Attributes : Product Id
Total Price (Price + Delivery Cost)
Payment Id
Purchase date
Duration
Delivered

- AddProduct()
- EditProduct()
- ProductBorrow()
- ProductLend()

Control Object

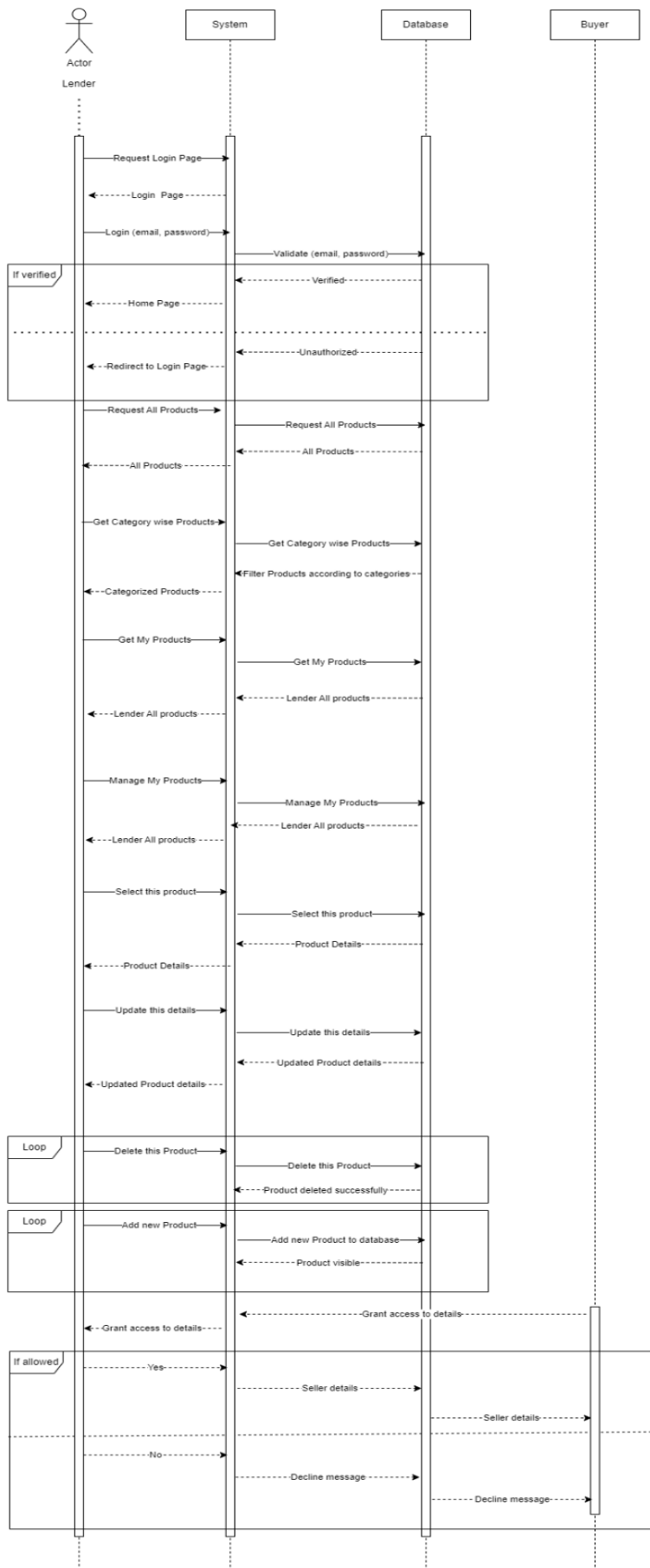
- BorrowManager
 - The admin would have a master access over the system.
 - It would consist of a BorrowMangager which would look after the borrowing and lending of products among the users of the website.
 - The BorrowManager would have the following functions to look after this:
 - SendNotification()
 - SendReminder()
 - AuthenticateUser()
 - CreateOrder()

Class diagrams:

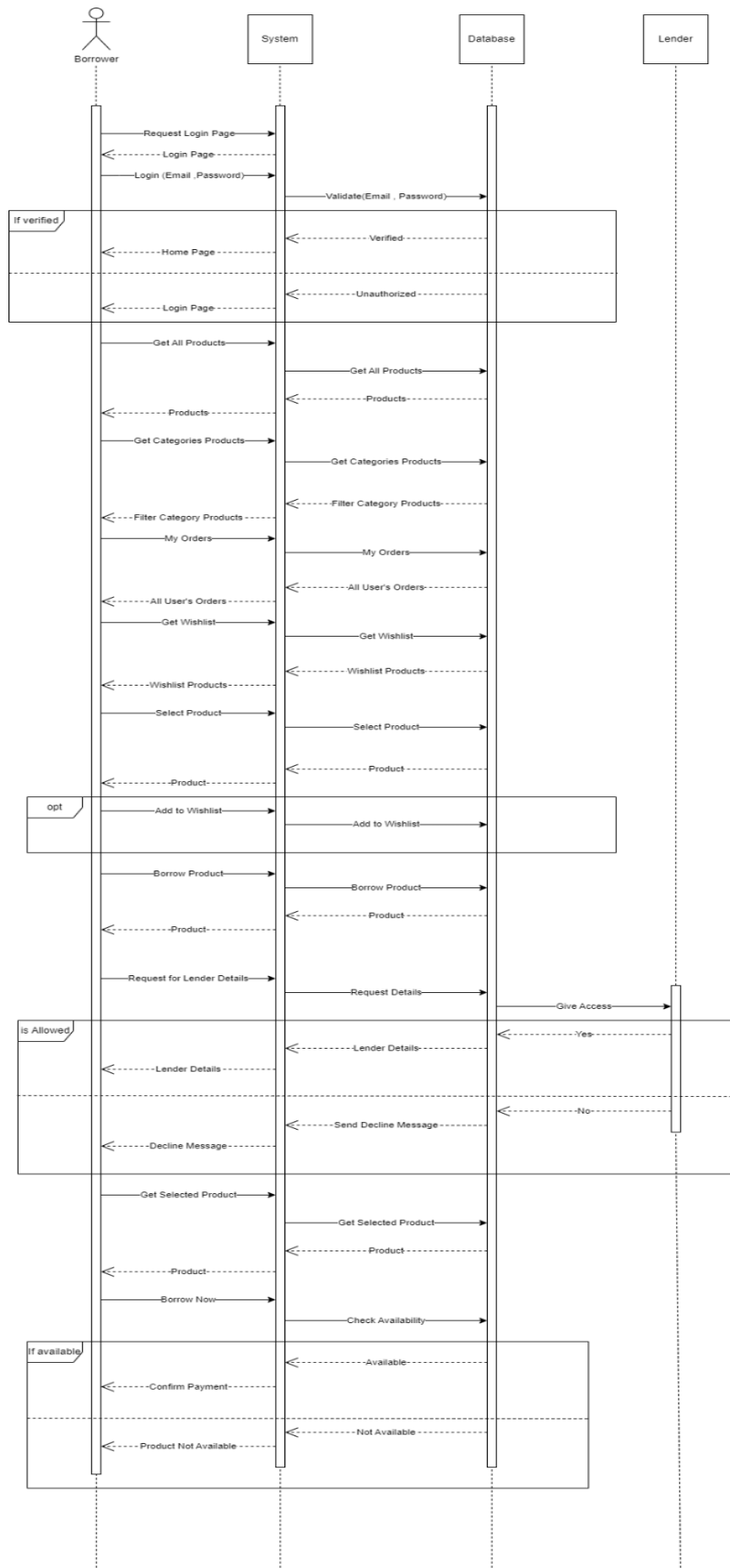


Sequence diagrams:

For Lender:



For borrower:



Design goals

- **Reliability:**
 - Reliability is a major concern as low reliability leads to higher risk of errors and failure which can potentially give wrong information to Buyers or Sellers.
- **Maintainability:**
 - The system should be easy to maintain and manage as it should provide service 24x7 to users.
- **Efficiency:**
 - The system should provide a seamless and hassle-free experience for both lenders and buyers ,while optimizing the use of resources and minimizing waste and still meeting the needs of the customers.
- **Fault tolerance:**
 - The system should be resistant to any potential failures and in case of failures, it needs to be recovered using backups and other methods to ensure that the user's data is not lost and if so, it can be retrieved easily.
- **Flexibility:**
 - The system should be flexible enough to adapt to new changes in the future as there will always be some evolving requirements by the users based on their preferences.
- **User Friendliness:**
 - Users, in this case Buyers and Sellers should not face any difficulty in using various features such as placing items for renting ,searching items ,categorisation of them etc.
- **Understandability:**
 - System should have good user experience and the rental process should be as simple and straightforward as possible, so that users can understand and use the system very easily.
- **Robustness**
 - The system should continue to function properly even in the face of unexpected events or failures. This means the system should be able to detect and recover from errors, prevent data loss, and maintain system availability.

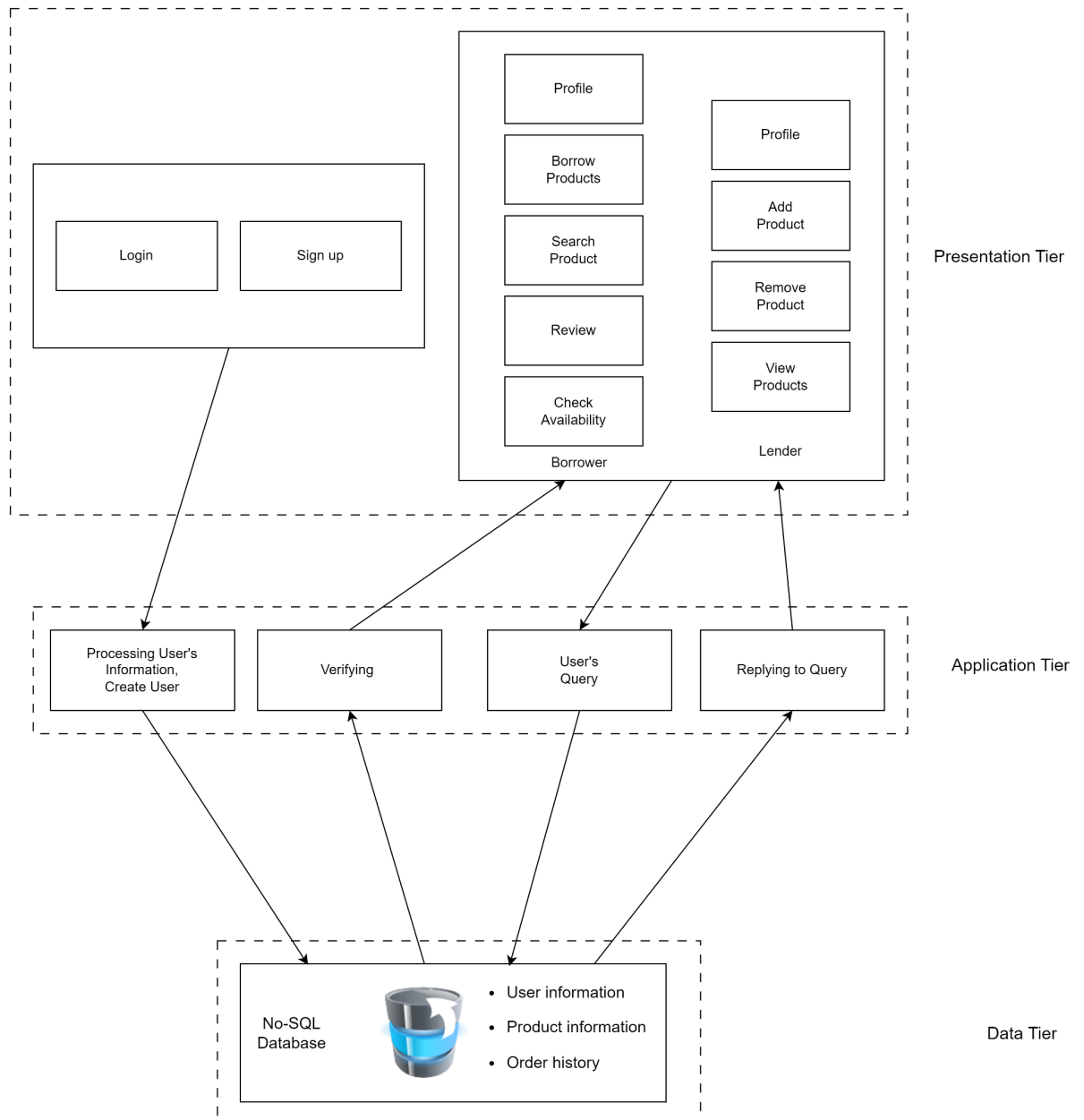
- **Well Defined interfaces:**
 - An interface is the point of interaction between the user and the system, and it should be designed to be intuitive, efficient, and easy to use. When the interface is well-defined, users can navigate the website and complete tasks without confusion, frustration, or errors.
- **Ease of Learning:**
 - Ease of learning affects the user's ability to quickly understand and use the website. When users find it easy to learn how to use the website, they are more likely to continue using it and completing their tasks successfully.

High Level System Design:

Client-server architecture will be utilized by this application. A presentation tier, an application tier, and a data tier make up the three levels of the application architecture that we'll use.

The presentation tier is a graphical user interface, while the data tier stores information and manages logic. (GUI)

- **Presentation tier** :
 - ☐ ReactJS is used to build the presentation tier, which is transmitted to a computing device via a web browser or a web-based application. The major method of communication between the presentation tier and the higher levels is through calls to the application programming interfaces (APIs).
- **Application tier** :
 - ☐ The business logic that underpins the program's primary functionalities is located in the application tier, also known as the logic tier, which is developed in Node.js with embedding express.js.
 - ☐ This decision was made because it enables widespread app usage and offers flexibility for future expansion.
- **Data tier** :
 - ☐ The data tier consists of a database and software for managing read and write access to databases. For this tier, MongoDB will be used.



*****Thankyou*****