"E-COMMERCE WEBSITE TO SELL SECONDHAND PRODUCT ; BIDDING SYSTEM FOR AIT."

Bid buy sell system using client-server architecture, solution for concurrent users for the website application through optimistic locking and multithreading.

Submitted to :

Dr. Chaklam Silpasuwanchai

AT70.18 Software Architecture Design

Project Members:

Suyogya Ratna Tamrakar (st121334)

Shubhangini Gontia (st121473)

Younten Tshering (st121775)

Outlines

1. Introduction
2. Architectural Challenges of our system
3. Architectural Design
4. Quality Attribute Analysis
5. Use Case Diagram and Class Diagram
6. Major User interface – Screen Mockup
7. Future Scope
8. Real-demo
9. Conclusion

Introduction

- The idea is like any **e-commerce platform** in which buying and selling of goods are done. But the core addition will be the system of **bidding before buying anything**.
- In case of AIT, this platform can be used to sell the used items. More than one person could be **interested in the same product**, but with this system, there will be a sense of competition as **one person bids over another** user's bid at the same time and the product value will also increase.
- In modern software systems, it is not uncommon to have hundreds or thousands of users independently and **concurrently interacting with the resources**. We generally want to avoid situations when changes made by one client are overridden by another one without even knowing.

Architectural Challenges of our system

- **Handling multiple concurrent transactions**

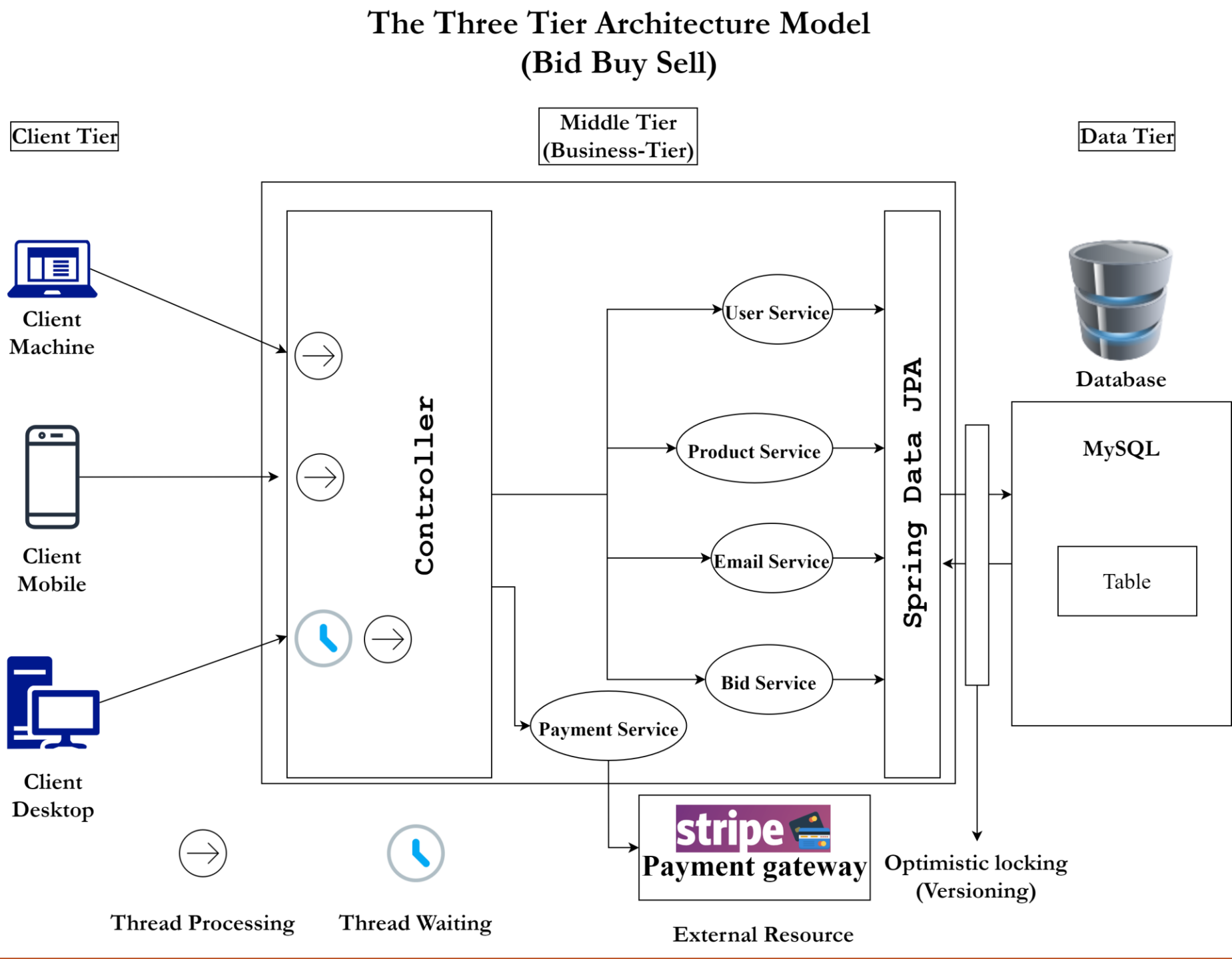
There will be many users active on the system and they will be able to bid on the products. But multiple users may place bids on the same product at the same time.

- **Updating the product price in real time for ongoing bidding processes.**

If the product is being bided by someone, there will be multiple users who will be on the same page waiting or trying to bid. If one bid is updated, other users must instantly see the updated bid amounts on their systems. This must be nearly real-time.

Note: Response time increases as concurrency increases

Architectural Design



Quality Attribute Analysis

1. Performance (H)

The output is to show who won and is bidding for the product in this system and we need to focus on concurrency and response time.

- The application should respond to a user within 2 seconds.
- The application should be able to handle 20 transactions per second in the peak load time.
- The application will be available with the uptime of 95% between 5:00 am to 1:00 am.

2. Security (M)

- All the transaction data between client and server must be encrypted.
- Since the site is buying and selling product, the payment or transactions with payment gateways will be incorporated, hence requiring high level of security mechanisms.
- The system should be able to restore backward data of 24 hours (maximum 3 Days) within 2 hours as a recovery function.

3. Usability (M)

- The web app must support latest Web browsers for any OS.
- The web app should be responsive.
- The web app should be easy to operate by users with a certain navigation menu or option. No need for a user manual.

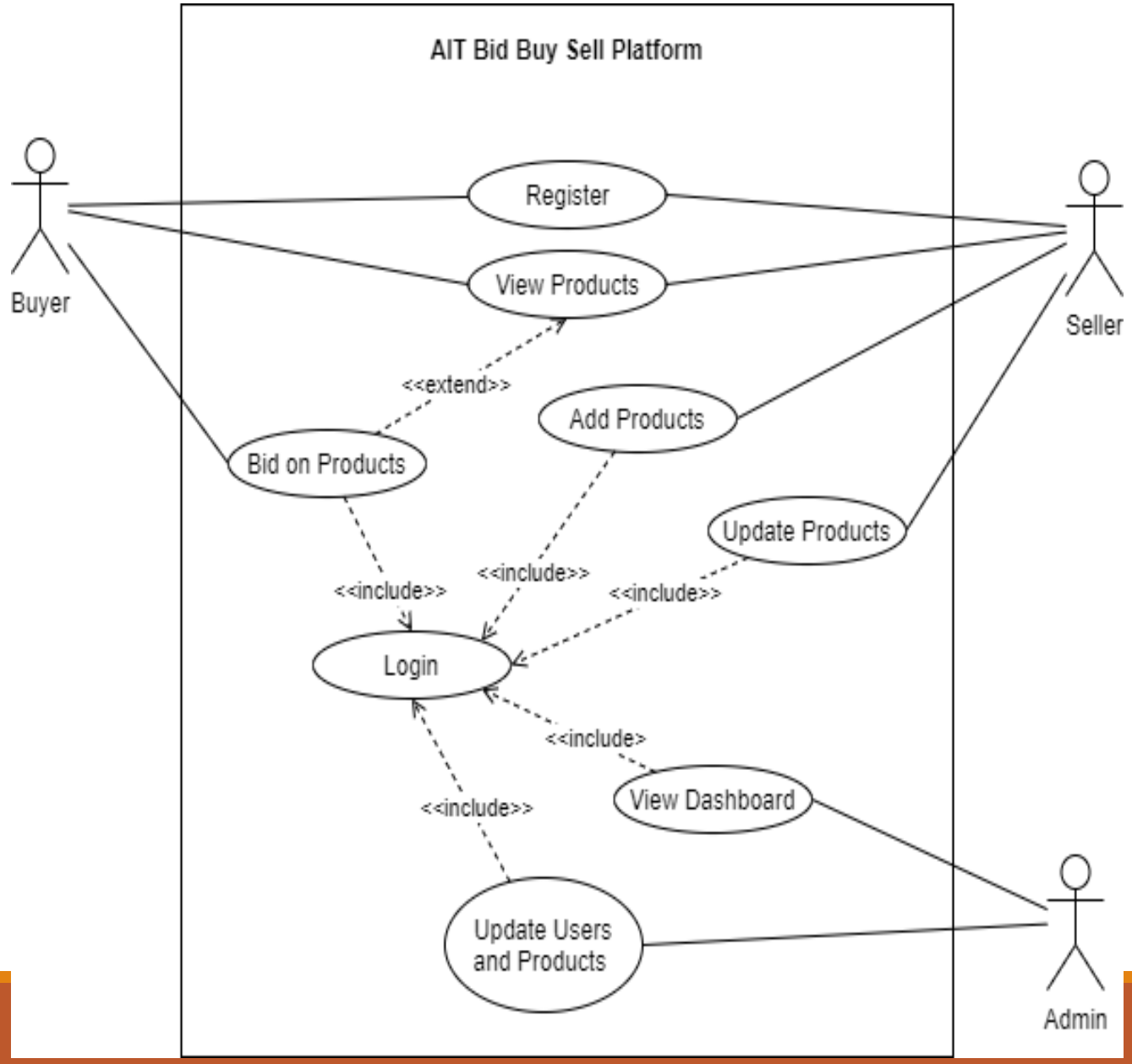
4. Interoperability (L)

- For the website we just import and integrate various information with pictures into the system from the seller.

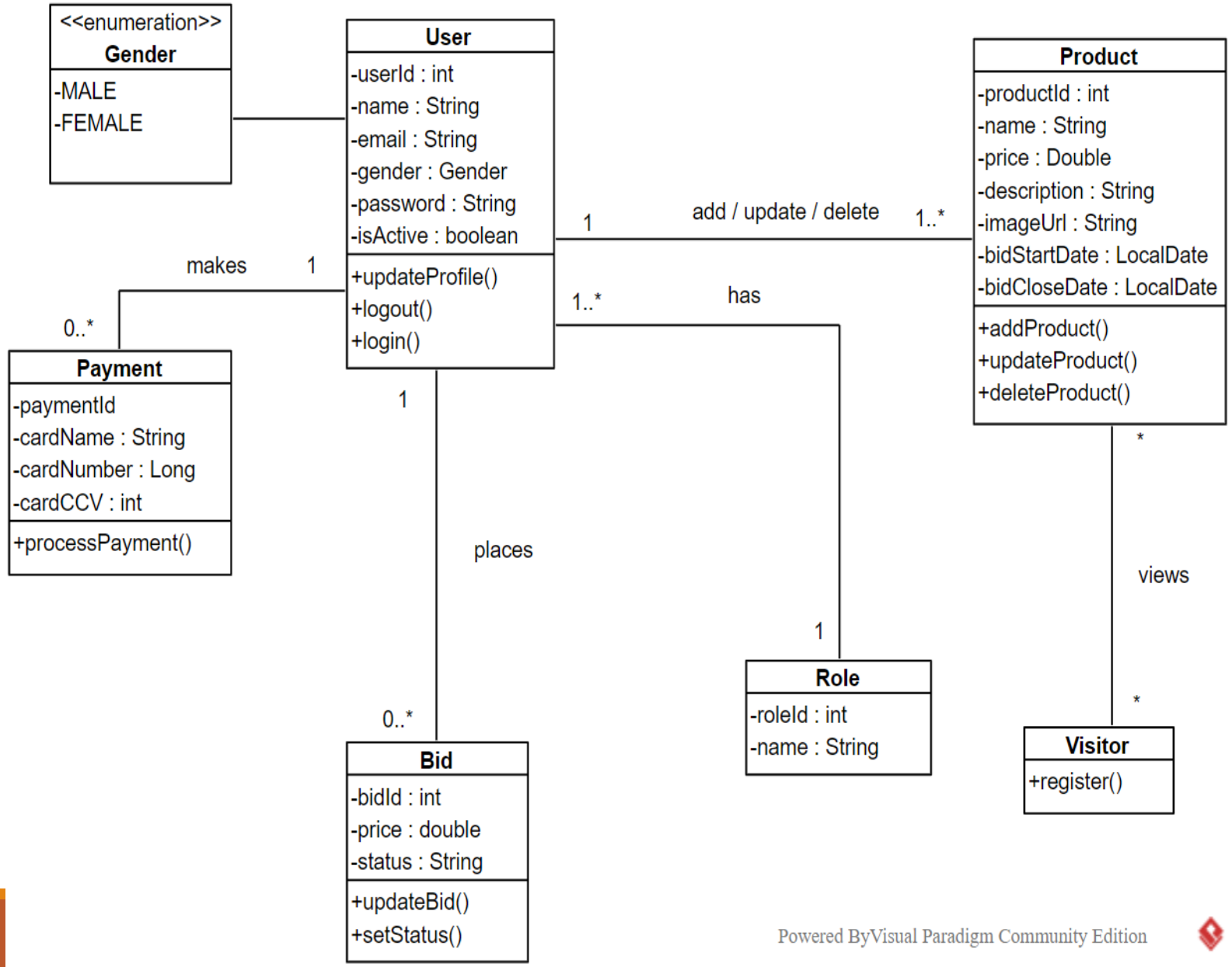
5. Scalability (L)

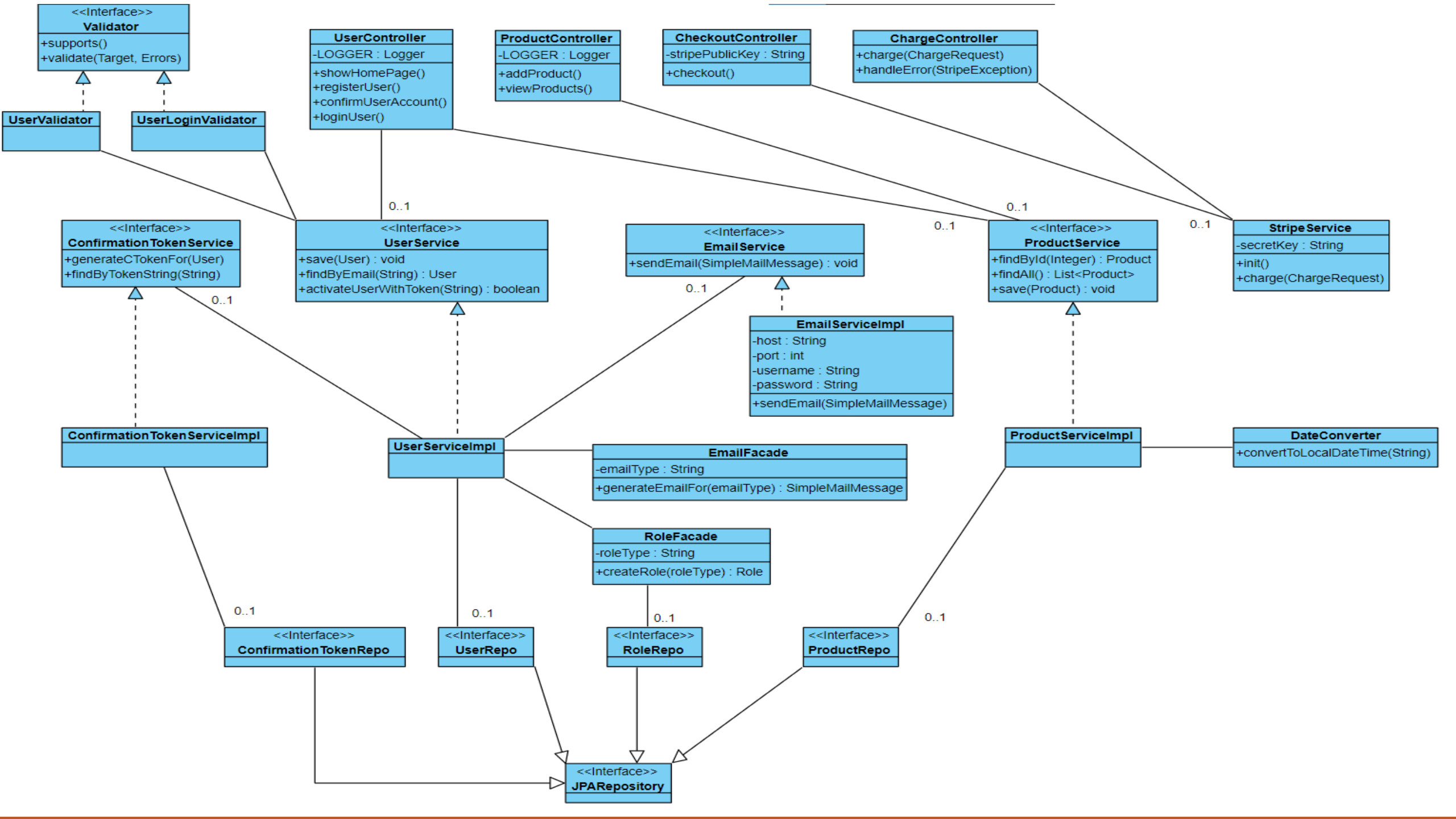
- With data, the storage size will increase but can manage before time. This app can be made vertically scalable when there is lack of memory storage.

Use case Diagram



Class Diagram




<<Interface>>
Validator
+supports()
+validate(Target, Errors)
UserValidator
UserLoginValidator
UserController
-LOGGER : Logger
+showHomePage()
+registerUser()
+confirmUserAccount()
+loginUser()
ProductController
-LOGGER : Logger
+addProduct()
+viewProducts()
CheckoutController
-stripePublicKey : String
+checkout()
ChargeController
+charge(ChargeRequest)
+handleError(StripeException)
0..1
0..1
0..1
0..1
<<Interface>>
ConfirmationTokenService
+generateCTokenFor(User)
+findByTokenString(String)
0..1
<<Interface>>
UserService
+save(User) : void
+findByEmail(String) : User
+activateUserWithToken(String) : boolean
<<Interface>>
EmailService
+sendEmail(SimpleMailMessage) : void
0..1
EmailServiceImpl
-host : String
-port : int
-username : String
-password : String
+sendEmail(SimpleMailMessage)
<<Interface>>
ProductService
+findById(Integer) : Product
+findAll() : List<Product>
+save(Product) : void
StripeService
-secretKey : String
+init()
+charge(ChargeRequest)
ConfirmationTokenServiceImpl
UserServiceImpl
EmailFacade
-emailType : String
+generateEmailFor(emailType) : SimpleMailMessage
RoleFacade
-roleType : String
+createRole(roleType) : Role
ProductServiceImpl
DateConverter
+convertToLocalDateTime(String)
0..1
0..1
0..1
0..1
<<Interface>>
ConfirmationTokenRepo
<<Interface>>
UserRepo
<<Interface>>
RoleRepo
<<Interface>>
ProductRepo
<<Interface>>
JPARepository

Screen Mockup

Home Page

Home Profile Help About Tutorial

AIT Bid-Buy-Sell

Register Login

Product Name: IPhone XR

Product description: 64 GB, used for 9 months and is in perfect condition, Liquid Retina HD display

Starting Price
TBH 15000
From 15/02/2021, 9:00AM
till 18/02/2021, 9:00PM

Bid

Product Name: Boiler

Product description: 3L capacity and used for 1 year

Starting Price
TBH 600
From 19/02/2021, 9:00AM
till 20/02/2021, 9:00PM

Bid

Screen Mockup

Future Scope

1. Bidding processing with bidding service
2. Implementing the Architectural Design with optimistic locking and multithreading
3. Integrate payment system
4. Performing Testing
5. Scientific Report

Real-demo

Conclusion

- The system ‘Bid buy Sell’ will be especially designed to be used by students and employees (staff and faculties) of AIT.

Questions and Feedback