A black background with orange and blue text

Description automatically generated**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Software Project Document

**Jewelry Production Order System - Luxuria**

|  |  |
| --- | --- |
|  | |
| **Group Members** | Trịnh Công Thái – Team Leader – SE183743  Phạm Nguyên Vũ – Team Member – SE184101  Lê Vũ Đức Lương – Team Member – SE183775  Chu Phan Nhật Long – Team Member – SE183738 |
| **Supervisor** | Mr. Tôn Thất Hoàng Minh |

**- Ho Chi Minh, May 2024 -**

**Table of Contents**

Contents

[Definition and Acronyms 4](#_Toc172644715)

[**I.** Project Introduction 5](#_Toc172644716)

[1. Project Information 5](#_Toc172644717)

[2. Project Team 5](#_Toc172644718)

[3. Product Background 5](#_Toc172644719)

[4. Existing Systems 5](#_Toc172644720)

[5. Business Opportunity 6](#_Toc172644721)

[6. Software Product Vision 6](#_Toc172644722)

[7. Project Features and Limitations 7](#_Toc172644723)

[a. Major Features 7](#_Toc172644724)

[b. Limitations 8](#_Toc172644725)

[**II.** Project Management Plan 8](#_Toc172644726)

[1. Overview 8](#_Toc172644727)

[1.1 Scope & Estimation 8](#_Toc172644728)

[1.2 Project Objectives 9](#_Toc172644729)

[1.3 Project Risks 10](#_Toc172644730)

[2. Management Approach 11](#_Toc172644731)

[2.1 Project Process 11](#_Toc172644732)

[2.2 Quality Management 11](#_Toc172644733)

[2.3 Training Plan 12](#_Toc172644734)

[3. Project Deliverables 12](#_Toc172644735)

[3. Responsibility Assignments 13](#_Toc172644736)

[4. Project Communications 14](#_Toc172644737)

[5.1 External Interface 14](#_Toc172644738)

[5. Configuration Management 15](#_Toc172644739)

[5.1 Document Management 15](#_Toc172644740)

[5.2 Source Code Management 16](#_Toc172644741)

[5.3 Tools & Infrastructures 16](#_Toc172644742)

[**III.** Software Requirements Specification 17](#_Toc172644743)

[1. Product Overview 17](#_Toc172644744)

[1.1 Guest Requirements 17](#_Toc172644745)

[1.2 Customer Requirements 17](#_Toc172644746)

[1.3 Sales Staff Requirements 17](#_Toc172644747)

[1.4 Manager Requirements 17](#_Toc172644748)

[1.5 Design Staff Requirements 18](#_Toc172644749)

[1.6 Production Staff Requirements 18](#_Toc172644750)

[1.7 Admin Requirements 18](#_Toc172644751)

[2. User Requirements 20](#_Toc172644752)

[2.1 Actors 20](#_Toc172644757)

[2.2 Use Cases 21](#_Toc172644758)

[3. System Functions 23](#_Toc172644759)

[4. Screen Authorization 30](#_Toc172644760)

[5. Entity Relationship Diagram 31](#_Toc172644761)

[6. Business Rules 35](#_Toc172644762)

[7. Application Message List 36](#_Toc172644763)

[**IV.** Software Design Description 39](#_Toc172644764)

[1. System Design 39](#_Toc172644765)

[2. Package Diagram 40](#_Toc172644766)

[3. Database Design 41](#_Toc172644767)

[4. Detailed Design 47](#_Toc172644768)

[**V.** Release Package & User Guides 55](#_Toc172644769)

[1. Deliverable Package 55](#_Toc172644770)

[1.1 Source codes & documents 55](#_Toc172644771)

[1.2 Known Issues, Limitations & Restrictions 56](#_Toc172644772)

[2. Installation Guides 56](#_Toc172644773)

[2.1 System Requirements 56](#_Toc172644774)

[2.2 Setup Files 57](#_Toc172644775)

[2.3 Installation Instruction 57](#_Toc172644776)

**Acknowledgement**

We would like to express our deepest gratitude to those who have played a pivotal role in the success of our SWP391 project.

* **Our heartfelt appreciation goes to our esteemed supervisor, Mr. Tôn Thất Hoàng Minh.** His invaluable guidance, insightful suggestions, and unwavering support have been instrumental in shaping the project from its inception to its completion.
* **We extend our sincere thanks to the esteemed FPT University lecturers, Mr. Nguyễn Quốc Khánh, Mr. Lê Nguyễn Sơn Vũ, and Mr. Nguyễn Minh Sang.** Their profound knowledge, expert guidance, and dedication to teaching have laid the foundation for our academic journey and prepared us to tackle this project with confidence.
* **We are also grateful to Mr. Nguyễn Thế Hoàng for his invaluable assistance with document structure and formatting.** His expertise ensured that our project adhered to the required guidelines and presented our work in a professional and polished manner.
* **We would like to express our gratitude to our families and friends for their unwavering support throughout this challenging endeavor.** Their encouragement, understanding, and belief in our abilities have been a source of strength and motivation.

We acknowledge that our limited experience may have resulted in areas where the project could be further improved. We appreciate your understanding and welcome any constructive feedback.

Thank you for your unwavering support.

Regards,

Team Fumi

# Definition and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| BA | Business Analysis |
| BR | Business Rule |
| LI | Limitations |
| ERD | Entity Relationship Diagram |
| GUI | Graphical User Interface |
| PM | Project Manager |
| SDD | Software Design Description |
| SRS | Software Requirement Specification |
| UAT | User Acceptance Test |
| UC | Use Case |
| API | Application Programming Interface |
| BE | Backend |
| FE | Frontend |

# Project Introduction

## 1. Project Information

* Project name: Jewelry Production Order System
* Software type: Web Application

## 2. Project Team

**a. Supervisors**

|  |  |  |  |
| --- | --- | --- | --- |
| Full Name | Role | Email | Mobile |
| Tôn Thất Hoàng Minh | Lecturer | minhtth5@fe.edu.vn | 0936668995 |

**b. Team Members**

|  |  |  |  |
| --- | --- | --- | --- |
| Full Name | Email | Mobile | Role |
| Trịnh Công Thái | thaitcse183743@fpt.edu.vn | 0902700316 | Leader |
| Phạm Nguyên Vũ | vupnse184101@fpt.edu.vn | 0908705620 | Member |
| Lê Vũ Đức Lương | luonglvdse183775@fpt.edu.vn | 0368984593 | Member |
| Chu Phan Nhật Long | longcpnse183738@fpt.edu.vn | 0347178790 | Member |

## 3. Product Background

The Jewelry Production Order System is a website that allows the customers to make requests for custom-made accessories by contacting our sales staffs. The website allows the customers to see various designs made by our finest designers as well as choosing the materials for their products.

## 4. Existing Systems

**a. PNJ**

* Link: <https://www.pnj.com.vn/>
* Description: PNJ is a leading brand in Vietnam in the field of crafting and retailing gold, silver, and gemstone jewelry. The company specializes in the production and trading of gold, silver, and gemstone jewelry, corporate gifts, fashion accessories, souvenirs, and watches. They also engage in the buying and selling of gold bars, provide diamond, gemstone, and precious metal certification services, and operate in real estate business.
* System Actor: Customer
* Features: Customer can view and buy products in-store, and view blog content.
* Advantages: Diverse products, user-friendly interface, many features.
* Disadvantages: Lacks a production order system.

**b. Bảo Tín K&K**

* Link: <https://www.baotinkk.com/blogs/dich-vu/gia-cong>
* Description: Bảo Tín K&K is a renowned jewelry brand in Vietnam. Established in 1995, Bảo Tín K&K specializes in trading gold, diamonds, gemstones, and pearls. They are particularly known for diamond wedding rings. Bảo Tín K&K offers a wide variety of jewelry such as rings, earrings, bracelets, bangles, necklaces, and pendants. They also provide jewelry processing and gold purity certification services.
* System Actor: Customer
* Features: Customer can view and buy products in-store, view blog content, and utilize a production order system.
* Advantages: Friendly and appealing interface, production order system, various features.
* Disadvantages: Production order system is not the main focus.

## 5. Business Opportunity

The jewelry market is developing, expected to reach 1.14 trillion USD in 2024 with an average growth rate of 3.73% (CAGR 2024-2028). According to updated data from August 2023, online channels account for 11.2% of the total distribution channels of the jewelry market in Vietnam. This presents an opportunity to develop online distribution channels as consumers are gradually forming the habit of online shopping. By leveraging apps and websites, companies can effectively promote their brand and products to the public. At the same time, they can manage orders, provide convenience, and optimize customer personalization.

## 6. Software Product Vision

* **For Customers:**
  + Access more design choices
  + Expedite product customization requests
  + Enjoy convenient ordering
  + Personalize their own products
  + Track order progress
* **For Company:**
  + Reach more customers
  + Simplify custom order requests

## 7. Project Features and Limitations

## Major Features

* **Guest:**
* FE-01: Register account.
* FE-02: Forgot password
* FE-03: View and filter product
* **Authorization User:**
* FE-04: Login/Logout
* FE-05: Manage personal profile: View profile, update profile, change password.
* **Customer:**
* FE-06: Manage request: send request, cancel request.
* FE-07: View order information, decide on the order, cancel order.
* **Sales Staff:** 
  + - FE-08: View request from customer.
    - FE-09: Manage order: create order, view order, edit order, cancel order, submit order’s price to manager.
    - FE-10: Submit order's price to manager.
    - FE-11: Export warranty.
* **Manager:**
  + - FE-12: View order information.
    - FE-13: Approve or decline order's price.
* **Design Staff:**
  + - FE-14: View 'approved' order detail.
    - FE-15: Upload 3D design for customer’s order.
* **Production Staff:**
  + - FE-16: View 'approved 3D design' order detail.
    - FE-17: Report finished order.
* **Admin:**
* FE-18: View admin dashboard.
* FE-19: Manage user and product.

## Limitations

|  |  |
| --- | --- |
| # | Limitations |
| LI-01 | There is only one admin in the system |
| LI-02 | The system is not the best and most optimal solution |
| LI-03 | Some unimportant features and UI are still incomplete |

# Project Management Plan

## Overview

### Scope & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort (man-days)** |
| ***1*** | ***Requirement analysis*** |  | ***6*** |
| 1.1 | Define background and context of product | Complex | 2 |
| 1.2 | Find problems of users | Medium | 2 |
| 1.3 | Existed system analysis | Simple | 2 |
| ***2*** | ***Technology training*** |  | ***70*** |
| 2.1 | Learn Spring Boot framework | Complex | 42 |
| 2.2 | Learn ReactJS framework | Complex | 70 |
| ***3*** | **Workspace setup** |  | ***3*** |
| 3.1 | Setup resources storage workspace | Medium | 2 |
| 3.2 | Setup resources and version control | Medium | 1 |
| ***4*** | **Document writing** |  | ***57*** |
| 4.1 | Write project introduction | Medium | 3 |
| 4.2 | Write project management plan | Medium | 5 |
| 4.3 | Write software requirement specification | Complex | 49 |
| ***5*** | **Interface design** |  | ***70*** |

|  |  |  |  |
| --- | --- | --- | --- |
| 5.1 | Design web application interface | Complex | 63 |
| 5.2 | Design figma prototype | Medium | 7 |
| ***6*** | ***Implementation*** |  | **19** |
| 6.1 | Implement authentication/authorization  features | Complex | 3 |
| 6.2 | Implement customer features | Complex | 4 |
| 6.3 | Implement sales staff features | Complex | 4 |
| 6.4 | Implement manager features | Medium | 2 |
| 6.5 | Implement design staff features | Medium | 3 |
| 6.6 | Implement production staff features | Medium | 1 |
| 6.7 | Implement admin features | Medium | 2 |
| ***7*** | ***Testing*** |  | **7** |
| 7.1 | Unit Test | Complex | 0 |
| 7.2 | Integration Test | Complex | 7 |
| 7.3 | System Test | Complex | 0 |
| ***8*** | ***Closing*** |  | **7** |
| 8.1 | Maintenance | Medium | 0 |
| 8.2 | Prepare slides for defense | Medium | 7 |

***Total Estimated Effort (man-days)*** ***70***

Table 3 - Scope & Estimation

### Project Objectives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Testing Stage** | **No. of Defects** | **% Of Defect** | **Notes** |
| 1 | Reviewing | 0 | 0% |  |
| 2 | System Test | 0 | 0% |  |
| 3 | Acceptance Test | 0 | 0% |  |
| Total | | 0 | 0% |  |

### 1.3 Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | High weight points are skewed to Backend | Critical | Medium | Assign more members and spend more time doing backend tasks. |
| 2 | Frontend lacks experience | Critical | High | Provide training and mentors asking to guide frontend development. |

Table 4 - Project Risks

## Management Approach

### A diagram of a team Description automatically generated Project Process

References: [*https://www.scrum.org/*](https://www.scrum.org/)

Figure 1 - Scum process

This project is developed under the Scrum framework of Agile methodology. We choose this process model because of the following reasons:

+ Because customers really don't know what they want their website to look like, changing requirements often take place, we need to develop from part to part to output the most suitable process for our system to satisfy organization.

+ Using Agile for web application testing, teams get feedback almost daily and can act on that feedback immediately.

+ Our team does not have any experience in working on the website for patients which is why we need to directly communicate, research and work to deliver to customers as soon as possible so that we can get improvement feedback.

### Quality Management

In this project, to improve the project quality, we use the following approach:

* API convention:
  + Name the API Route that maps to the entity in the system
  + Each API response must return the following general information:
    - Status code: 200, 204, 400, 401, 404, ...
    - Status text: fail, success
    - Total: number of data items have been returned
    - Data: an array that contains all data items
* Coding convention:
  + Set the meaningful names for variables
  + Add comments to code can be confusing
  + Use Camel casing for naming Enum, Class and Interface
* Unit Testing: Tester member has the responsibility to write a unit test to ensure a function/method works appropriately.
* Integration Testing: After finishing coding one module/screen both members working on the client-side and server-side need to execute integration testing to ensure the module/screen can work smoothly from client-side to server-side.
* System Testing: After finishing coding for whole systems, all team members need to re-execute all test cases and free tests to ensure the system is working stable.

### Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| Spring Boot | Thái, Lương | 3 weeks | Mandatory |
| ReactJS | Vũ, Long | Along with FER202 | Mandatory |
| Document | Thái, Vũ | Along with SWR302 | Mandatory |
| Test document | Thái | Along with SWT301 | Mandatory |
| GitHub | All member | 2 days | Mandatory |
| AWS - RDS | Thái, Lương | 2 days | Mandatory |

Table 5 - Training plan

## 3. Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint #** | **Sprint objective** | **Duration** | **Deliverable** |
| # | SRS Document | 49 days | Project Planning, Software Requirement |
| 2 | Design Document | 7 days | Architecture Design, Basic Design, Detail Design and Database Design |
| 3 | Main Flow Module | 19 days | Code & System |
| 4 | Final Document | 5 days | Final document for the project |

Table 6 - Project deliverables

## Responsibility Assignments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Responsibility** | **Trịnh Công Thái** | **Phạm Nguyên Vũ** | **Lê Vũ Đức Lương** | **Chu Phan Nhật Long** |
| Back-end for system | x |  | x |  |
| Front-end web |  | x |  | x |
| Deployment | x |  |  | x |
| Documentation | x | x |  |  |
| Others | x | x | x | x |

Table 7 - Responsibility Assignments

## Project Communications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| Working with advisor | Supervisor (Ton That Hoang Minh) and team members | * Review documentation * Demonstrate features * Evaluate progress and result | 2 times per week | Face to face |
| Working in team | Team members | Raise opinions, problems   * Ask for help from other members * Report working status to leader | Always | Face to Face, Messenger, Discord, Zalo |

Table 8 - Project Communications

### 5.1 External Interface

**a. FU Contacts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person**  **(Name, position)** | **Contact address (Email, telephone)** | **Responsibility** |
| Advisor | Tôn Thất Hoàng Minh | minhtth5@fpt.edu.vn | * Provide document template * Give instruction to project team * Review deliverables * Supervise project status |

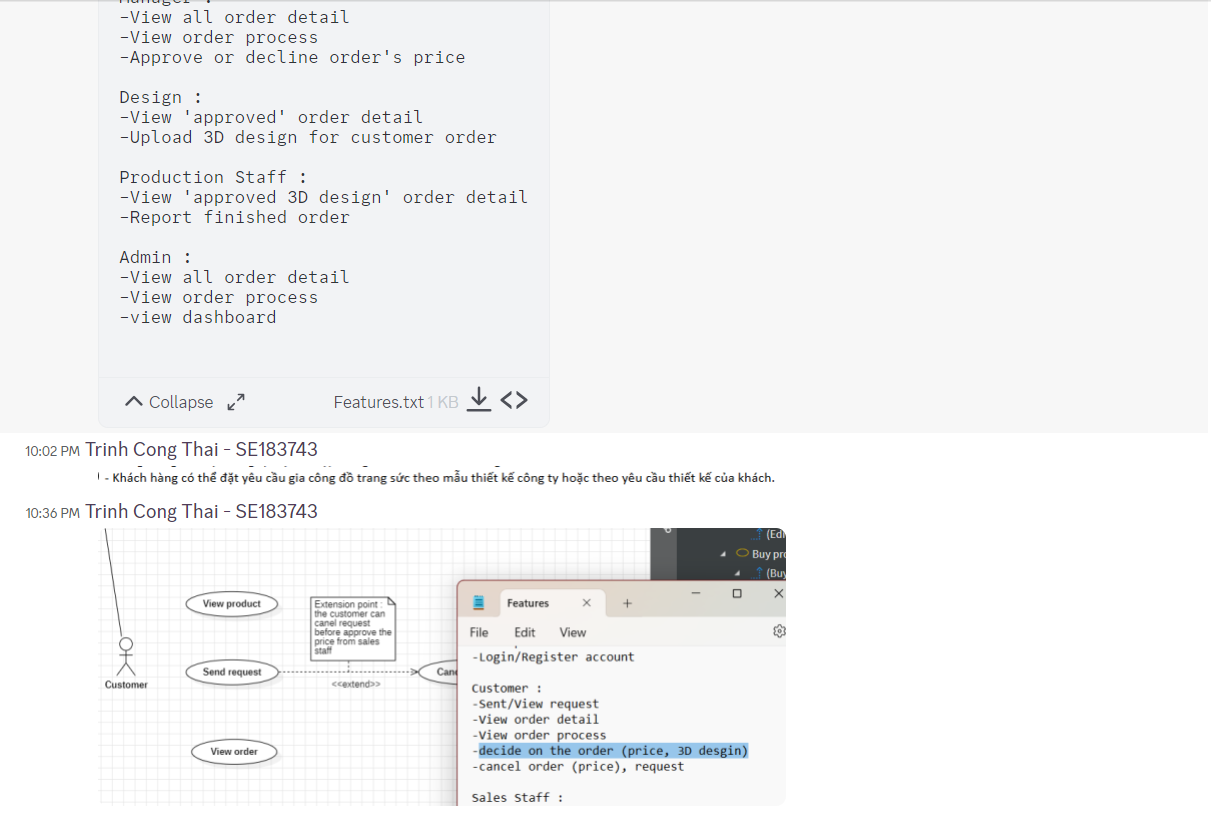
Table 9 - FU contact

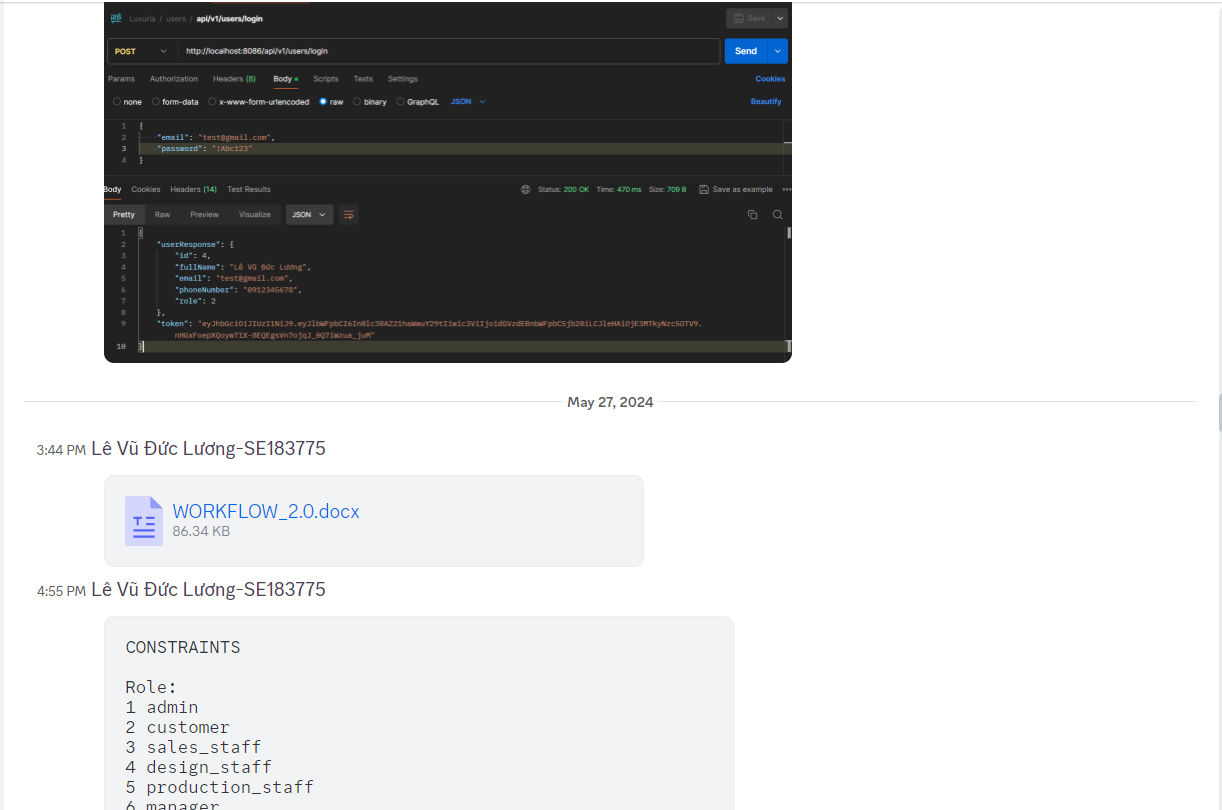
## Configuration Management

### Document Management

Because of the real-time data synchronization, we use Discord to manage documents. Discord allows team members to conveniently collaborate in real-time. Also, Discord is a great place to save images, diagrams, etc. Discord provide a channel that contains topic, comment features to discuss problems so that we can review thereafter.

-Below is our Discord Image:





### Source Code Management

We choose GitHub to manage our source code. The version control system allows members to collaboratively work on a source code efficiently, conveniently, and easily to resolve conflict, especially helpful for looking back in time and quickly identifying changes a collaborator made.

### Tools & Infrastructures

|  |  |
| --- | --- |
| **Category** | **Tools / Infrastructure** |
| **IDEs/Editors** | IntelliJ IDEA, Visual Studio Code |
| **UML tools** | Draw.io, Star UML, Visual Paradigm |
| **Source Version Control** | GitHub |
| **Project Management Tool** | Zalo, Jira |
| **Framework** | ReactJS (Frontend), Spring Boot (Backend) |
| **Programming Languages** | Java, Java Script |
| **Documentation** | MS Office, Slides |
| **DBMS** | MySQL |

Table 10 - Tools & Infrastructures

# Software Requirements Specification

## Product Overview

### Guest Requirements

For Guests who want to register an account to place orders. Our system will allow easy account creation, providing a straightforward process for new users to start using our services.

### Customer Requirements

For Customers who need a seamless way to manage their orders. The system will enable customers to send requests, view detailed order information, track order progress, make decisions on orders (including price and 3D design), and cancel orders if necessary.

### Sales Staff Requirements

For Sales Staff who require efficient order management tools. The system will provide the ability to view requests, see all order details, track order progress, edit customer orders, cancel orders, submit order prices to the manager, and export receipts and warranties.

### Manager Requirements

For Managers who need oversight and decision-making capabilities. The system will enable managers to view all order details, track order progress, and approve or decline order prices.

### Design Staff Requirements

For Design Staff who work on customer orders requiring 3D designs. The system will allow designers to view approved order details and upload 3D designs for customer orders.

### Production Staff Requirements

For Production Staff who handle the manufacturing of approved orders. The system will provide the ability to view details of orders with approved 3D designs and report finished orders.

### Admin Requirements

For Admins who need to manage and monitor the entire system. The admin website will provide comprehensive screens to view all order details, track order processes, and view a dashboard for an overview of system performance.

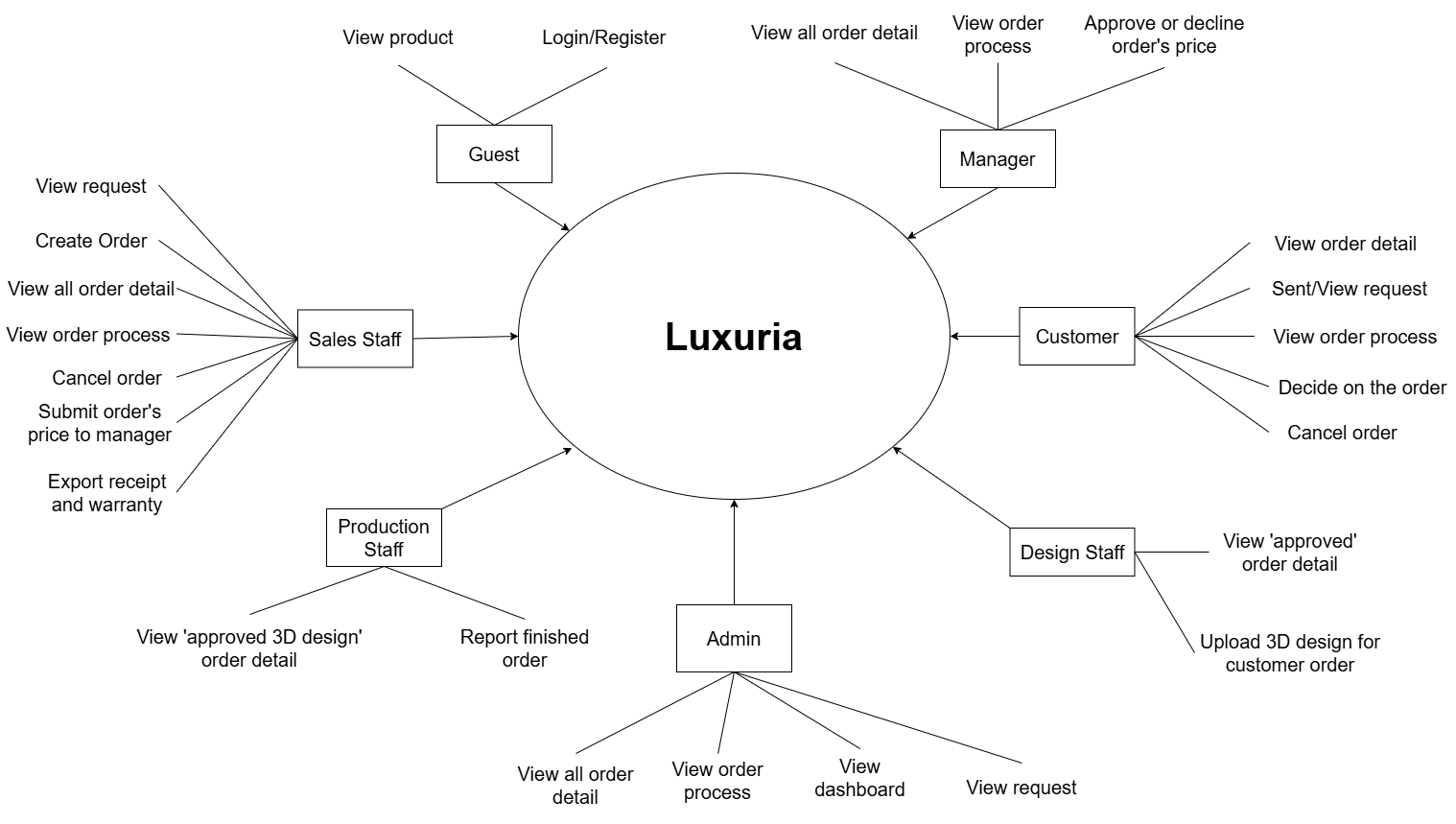


Figure 2 - Context diagram

## User Requirements



### Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Administrator | The administrators who can view all request, orders and the admin dashboard |
| 2 | Production Staff | The staff who can view “approved 3D design” order and report finished orders |
| 3 | Design Staff | The staff who can view “approved price” orders from customers and the manager. They can upload pictures of 3D designs to customer orders |
| 4 | Manager | The manager who can view orders, track order processes, and approve or decline prices submitted by sales staff |
| 5 | Sales Staff | The staff who can view and confirm or decline requests, create and view orders and order processes, edit customer orders, and submit order prices to the manager. They can also view and export receipts or warranties |
| 6 | Customer | The customers who can make and cancel requests, view orders, and cancel orders before the price is approved by sales staff. They can also approve prices from sales staff or designs from design staff |
| 7 | Guest | Guests who can visit the web pages |

Table 11 - Actors list and description

### Use Cases

* + - 1. ***Diagram***

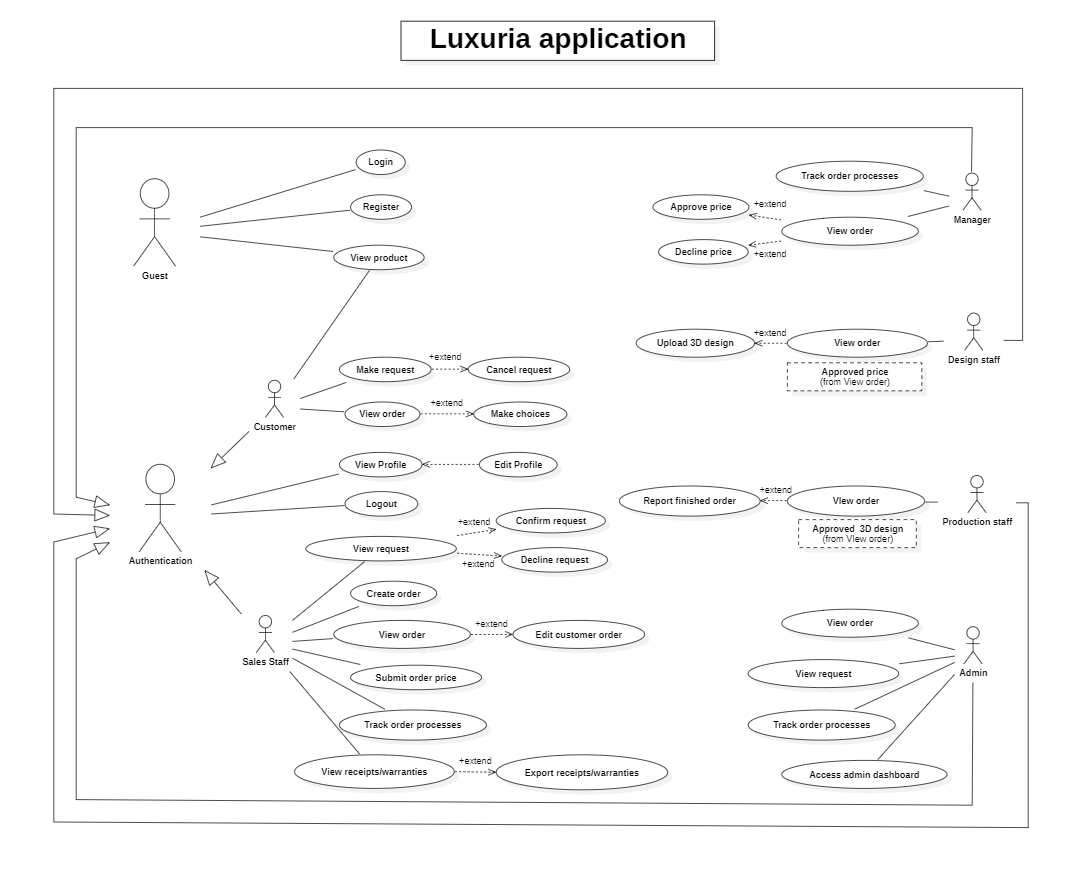


Figure 3 - <Use case overview> Luxuria Application

* + - 1. ***Use Case List***

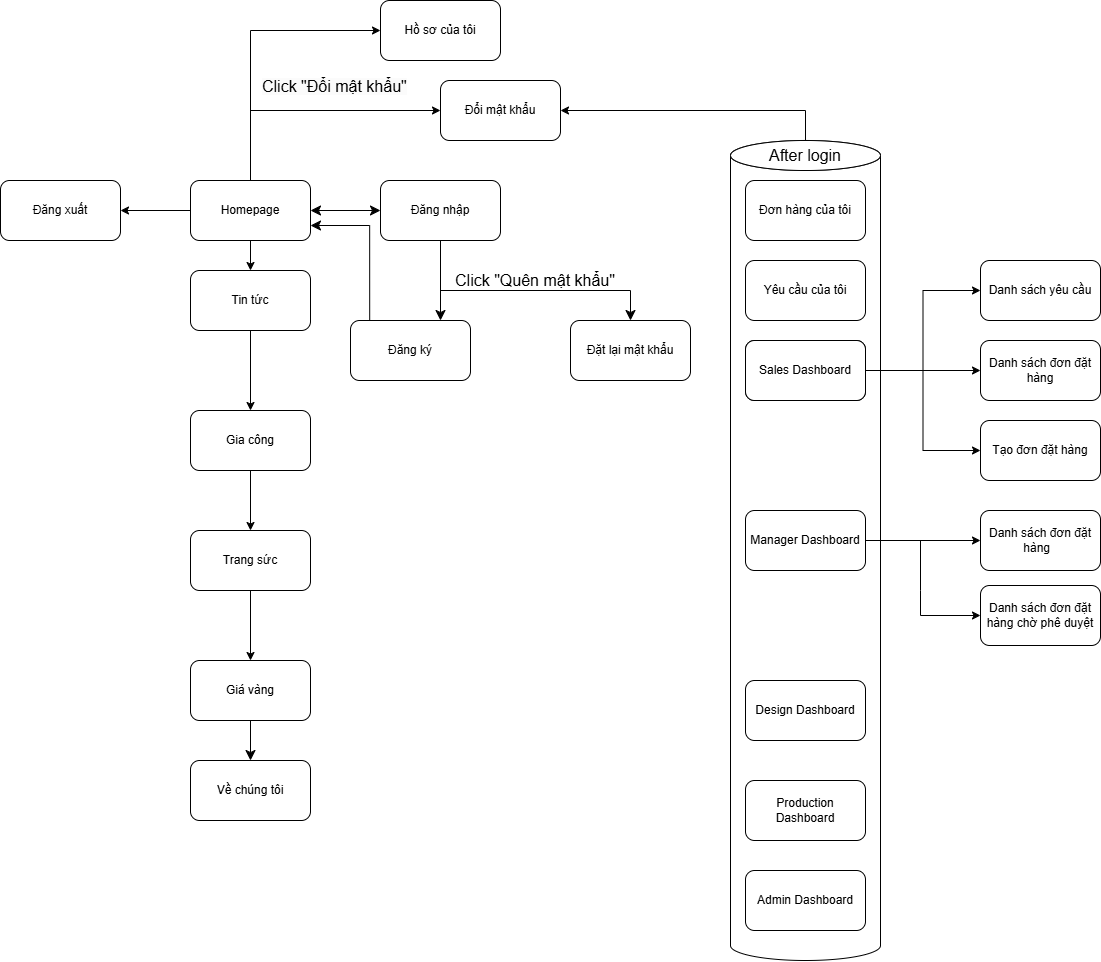
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Description** |
| UC01 | Register | Guest | Guest register for account |
| UC02 | Login | Guest | Unauthenticated user login to Luxuria to verify role, and give them access to more features |
| UC03 | Logout | All actors except Guest | This feature allows users to logout the web |
| UC04 | View Profile | All actors except Guest | This feature allows all users to view their profile |
| UC05 | Edit Profile | All actors except Guest | This feature allows users to update their profile |
| UC06 | Make Request | Customer | This feature allows the user to send request |
| UC07 | View Order | All actors except Guest | This feature allows users to view available orders |
| UC08 | View Request | Customer, Sales Staff, Manager, Admin | This feature allows certain users to view available requests |
| UC09 | Submit Order Price | Sales Staff | This feature allows the user to submit the prices of orders |
| UC10 | Track Order Process | Customer, Sales Staff, Manager, Admin | This feature allows certain users to track the processes of orders |
| UC11 | View Receipts/Warranties | Sales Staff | This feature allows user to view receipts/warranties |
| UC12 | Export Receipts/Warranties | Sales Staff | This feature allows user to export receipts/warranties |
| UC13 | Upload 3D design | Design Staff | This feature allows user to upload 3D design |
| UC14 | Report finished order | Production Staff | This feature allows user to report finished order |
| UC15 | Access admin dashboard | Admin | This feature allow user to access admin dashboard |

## System Functions

* 1. Screen detail

|  |  |  |
| --- | --- | --- |
| **Feature** | **Screen** | **Description** |
| Public | Home Page | The starting page of the system |
| Public | Tin tức | List of shared posts |
| Public | Chi tiết tin tức | Detail of the shared post |
| Public | Gia công | Details on production processes and send requests |
| Public | Trang sức | List of premade jewelry |
| Public | Giá vàng | List of gold prices |
| Public | Về chúng tôi | Information about the location of our stores, experiences and skills |
| Public | Vòng/lắc tay | List of premade bracelet/bangle designs |
| Public | Mặt dây/dây chuyền | List of premade pendant/necklace designs |
| Public | Nhẫn | List of premade ring designs |
| Public | Bông tai | List of premade earring designs |
| Common | Đăng nhập | Authenticate the user to give him/her to access authorized features later on |
| Common | Đăng ký | Register a new user into the system, verified by the registered email |
| Common | Đổi mật khẩu | Change user’s login password |
| Common | Hồ sơ của tôi | View & edit/update user profile |
| Common | User Authorization | Authorization user into the system functions based on his/her role |
| Customer | Yêu cầu của tôi | List of one’s requests |
| Customer | Đơn hàng của tôi | List of one’s orders |
| Sales Staff | Sales Dashboard | Contains all pages for sales staff |
| Sales Staff, Admin | Danh sách yêu cầu | List of customers’ requests |
| Sales Staff, Admin, Manager | Danh sách đơn đặt hàng | List of customers’ orders |
| Sales Staff | Tạo đơn đặt hàng | Create a new order based on information from customer’s request |
| Sales Staff, Manager, Admin | Chi tiết đơn hàng | Detail of the customer’s order |
| Manager | Manager Dashboard | Contains all pages for manager |
| Manager | Danh sách đơn đặt hàng chờ phê duyệt | List of customers’ orders waiting to be approved |
| Design Staff | Design Dashboard | Contains all pages for design staff |
| Production Staff | Production Dashboard | Contains all pages for production staff |
| Admin | Admin Dashboard | View income statistics |

* 1. Screen flow



* 1. Functional Requirements

1. Requests Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Actors | Function Name | Purpose | Data Requirements | Data Validation | Business Rules |
| 1 | Customer | Send Requests | Send requests to the sales staff |  |  | BR-12  BR-13 |
| 2 | Sales Staff | View requests | View the list of requests made by customers | * Request id * Customer id * Request date * Approved by staff * Active | Request id   * Type: int   Customer id   * Type: int   Request date   * Type: datetime   Approved by staff   * Type: bit   Active   * Type: bit | BR-16 |

1. Orders Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Actors | Function Name | Purpose | Data Requirements | Data Validation | Business Rules |
| 1 | Sales Staff | Create orders | Create new orders based on customer’s request | * Order id * Create date * Request id * Customer id * Process id * State id * Active | Order id   * Type: int   Create date   * Type: datetime   Request id   * Type: int   Customer id   * Type: int   Process id   * Type: int   State id   * Type: int   Active   * Type: bit | BR-15  BR-16  BR-17  BR-18 |
| 2 | Sales Staff, Manager, Admin | View all orders | View the list of orders | * Order id * Create date * Customer id * Process id * State id * Active | Order id   * Type: int   Create date   * Type: datetime   Customer id   * Type: int   Process id   * Type: int   State id   * Type: int   Active   * Type: bit |  |
| 3 | Sales Staff, Design Staff | View orders approved by customer | View the list of orders approved by the customer | * Order id * Create date * Customer id * Process id * State id * Active | Order id   * Type: int   Create date   * Type: datetime   Customer id   * Type: int   Process id   * Type: int   State id   * Type: int   Active   * Type: bit |  |
| 4 | Sales Staff, Production Staff | View orders with approved design | View the list of orders with designs approved by customer | * Order id * Create date * Customer id * Process id * State id * Active | Order id   * Type: int   Create date   * Type: datetime   Customer id   * Type: int   Process id   * Type: int   State id   * Type: int   Active   * Type: bit |  |
| 5 | Sales Staff, Design Staff, Production Staff | Update orders process | Update the process of the orders based on their roles | * Order id * Process id * State id * Active | Order id   * Type: int   Process id   * Type: int   State id   * Type: int   Active   * Type: bit |  |

1. Warranties Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Actors | Function Name | Purpose | Data Requirements | Data Validation | Business Rules |
| 1 | Sales Staff | Create warranties | Create new warranties | * Warranty id * Product id * User id * Delivery date * Warranty time | Warranty id   * Type: int   Product id   * Type: int   User id   * Type: int   Delivery date   * Type: date   Warranty time   * Type: varchar * Max length = 20 |  |
| 2 | Sales Staff | View warranties | View the list of warranties | * Warranty id * Product id * User id * Delivery date * Warranty time | Warranty id   * Type: int   Product id   * Type: int   User id   * Type: int   Delivery date   * Type: date   Warranty time   * Type: varchar * Max length = 20 |  |

1. Authenticate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Actors | Function Name | Purpose | Data Requirements | Data Validation | Business Rules |
| 1 | Customer, Sales Staff, Design Staff, Production Staff, Manager, Admin | Login | Login to website | * Email * Password | Email   * Type: varchar * Max length = 100   Password   * Type: varchar * Max length = 255 | BR-01 |

1. Register

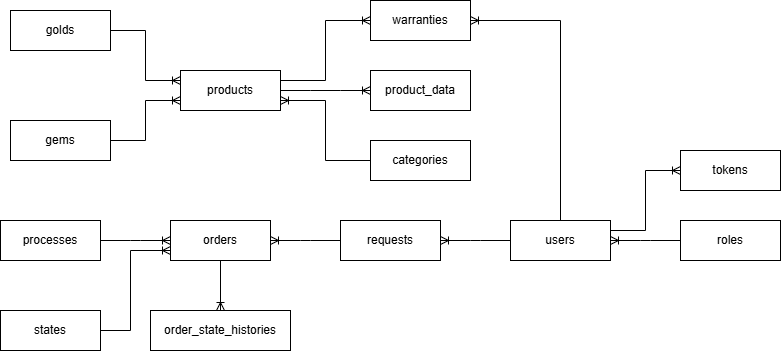
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Actors | Function Name | Purpose | Data Requirements | Data Validation | Business Rules |
| 1 | Guest | Register | Register account | * Email * Phone number * Password * Confirm Password | Full name   * Type: varchar * Max length = 100   Phone number   * Type: varchar * Max length = 100   Email   * Type: varchar * Max length = 100   Password   * Type: varchar * Max length = 255 | BR-02  BR-03  BR-04  BR-08  BR-09  BR-10  BR-11 |

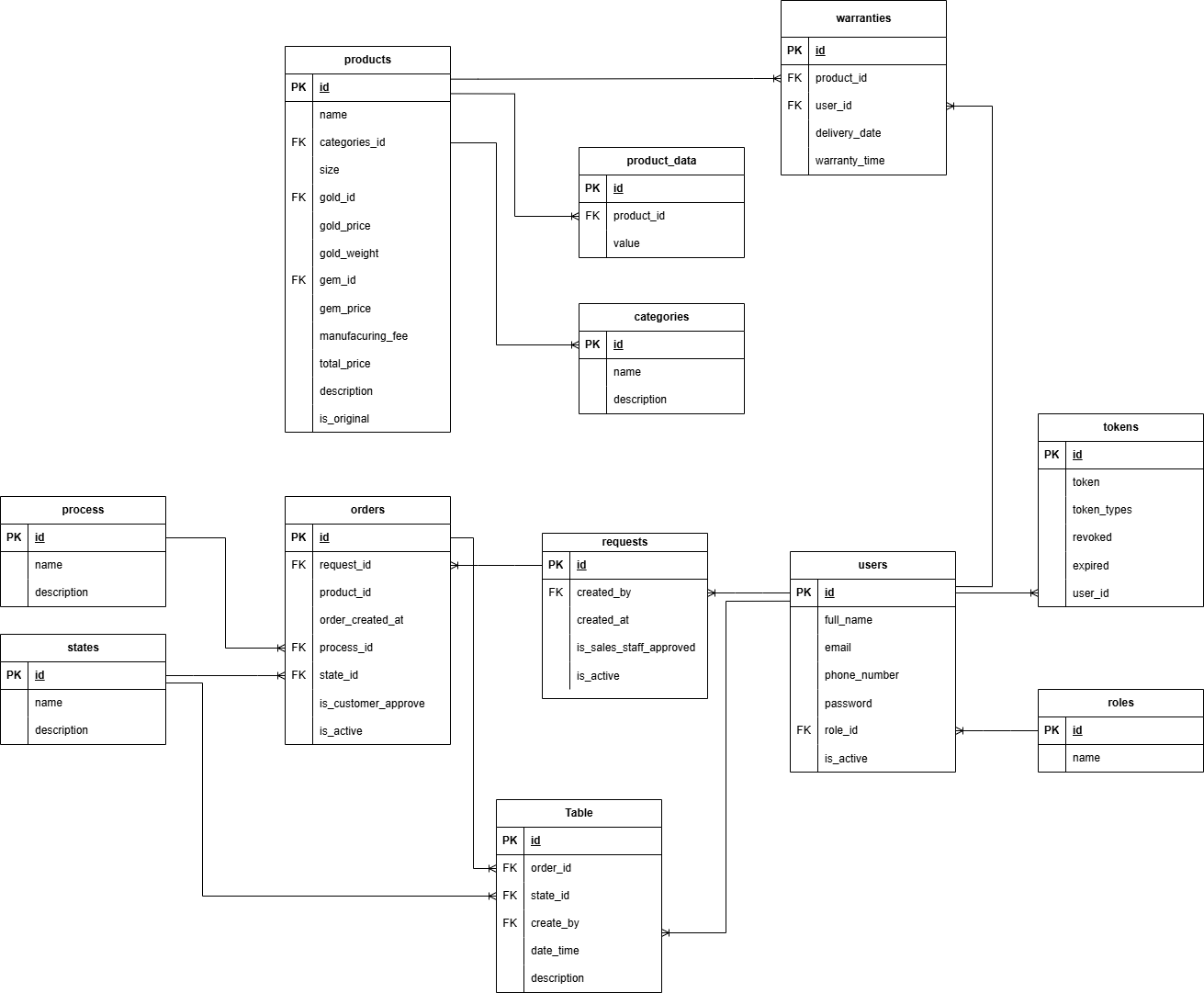
## Screen Authorization

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Screen | Guest | Customer | Sales Staff | Manager | Design Staff | Production Staff | Admin |
| Homepage | X | X | X | X | X | X | X |
| Tin tức | X | X | X | X | X | X | X |
| Chi tiết tin tức | X | X | X | X | X | X | X |
| Gia công | X | X | X | X | X | X | X |
| Trang sức | X | X | X | X | X | X | X |
| Giá vàng | X | X | X | X | X | X | X |
| Về chúng tôi | X | X | X | X | X | X | X |
| Đăng nhập | X | X | X | X | X | X | X |
| Đăng ký | X | X | X | X | X | X | X |
| Đặt lại mật khẩu | X | X | X | X | X | X | X |
| Đổi mật khẩu | X | X | X | X | X | X | X |
| Đăng xuất |  | X | X | X | X | X | X |
| Hồ sơ của tôi |  | X | X | X | X | X | X |
| Đơn hàng của tôi |  | X |  |  |  |  |  |
| Yêu cầu của tôi |  | X |  |  |  |  |  |
| Sales Dashboard |  |  | X |  |  |  |  |
| Danh sách yêu cầu |  |  | X |  |  |  |  |
| Danh sách đơn đặt hàng |  |  | X | X |  |  |  |
| Tạo đơn đặt hàng |  |  | X |  |  |  |  |
| Manager Dashboard |  |  |  | X |  |  |  |
| Design Dashboard |  |  |  |  | X |  |  |
| Production Dashboard |  |  |  |  |  | X |  |
| Admin Dashboard |  |  |  |  |  |  | X |

## Entity Relationship Diagram

* + 1. Conceptual ERD



* + 1. Logical ERD

|  |  |  |
| --- | --- | --- |
| Entity name | Attributes | Description |
| tokens | id {PK} | A unique identifier for each token |
| token | The actual token string used for authentication |
| token\_types | The type or category of the token |
| revoked | A boolean indicating whether the token has been revoked |
| expired | A boolean indicating whether the token has expired |
| user\_id {FK} | A foreign key linking the token to a specific user |
| roles | id {PK} | A unique identifier for each role |
| name | The name of the role |
| users | id {PK} | A unique identifier for each user |
| full\_name | The full name of the user |
| email | The email address of the user |
| phone\_number | The phone number of the user |
| password | The hashed password of the user |
| role\_id {FK} | A foreign key linking the user to a specific role |
| is\_active | A boolean indicating whether the user's account is active |
| requests | id {PK} | A unique identifier for each request |
| created\_by | The identifier of the user who created the request |
| created\_at | The date and time when the request was created |
| is\_sales\_staff\_approved | A boolean indicating whether the request has been approved by the sales staff |
| is\_active | A boolean indicating whether the request is active |
| orders | id {PK} | A unique identifier for each order |
| request\_id {FK} | A foreign key linking the order to a specific request |
| product\_id {FK} | A foreign key linking the order to a specific product |
| order\_created\_at | The date and time when the order was created |
| process\_id {FK} | A foreign key linking the order to a specific process |
| state\_id {FK} | A foreign key linking the order to a specific state |
| is\_customer\_approved | A boolean indicating whether the order has been approved by the customer |
| is\_active | A boolean indicating whether the order is active |
| products | id {PK} | A unique identifier for each product |
| name | The name of the product |
| description | A description of the product |
| categories\_id {FK} | A foreign key linking the product to a specific category |
| size | The size of the product |
| gold\_id {FK} | A foreign key linking the product to a specific gold record |
| gem\_id {FK} | A foreign key linking the product to a specific gem record |
| manufacturing\_fee | The manufacturing fee of the product |
| total\_price | The total price of the product |
| is\_original | A boolean indicating whether the product is original |
| warranties | id {PK} | A unique identifier for each warranty |
| warranty\_time | The warranty time of the product |
| description | A description of the warranty |
| states | id {PK} | A unique identifier for each state |
| name | The name of the state |
| description | A description of the state |
| order\_states\_histories | id {PK} | A unique identifier for each order state history entry |
| order\_id {FK} | A foreign key linking the history entry to a specific order |
| state\_ id {FK} | A foreign key linking the history entry to a specific state |
| updated\_at | The date and time the order state was updated |
| updated\_by | The identifier of the user who updated the order state (can be null) |

## Business Rules

|  |  |
| --- | --- |
| ID | Rule Definition |
| BR-01 | If the account is correct, system redirect users to corresponding home page:  - User logs in as role “admin”, the system will display admin Dashboard page  - User logs in as role “manager”, the system will display manager dashboard page  - User logs in as role “sales staff”, the system will display sales staff dashboard page  - User logs in as role “design staff”, the system will display design staff dashboard page  - User logs in as role “production staff”, the system will display production staff dashboard page  - User logs in as role “customer”, the system will display homepage |
| BR-02 | Email must be unique for user |
| BR-03 | Phone number must be 10 digits |
| BR-04 | Email must be in standard email format Ex: name@gmail.com |
| BR-05 | Account profile can be updated by only it's owner |
| BR-06 | Customer can only update their profile |
| BR-07 | Customer can only change their password |
| BR-08 | Confirm password obligatory like password |
| BR-09 | Username mustn’t be empty |
| BR-10 | Password length must have at least 6 characters |
| BR-11 | Password must have 1 normal character, 1 capitalized character, 1 number character and 1 special character |
| BR-12 | Customer must create account before sending request |
| BR-13 | Customer can only send production request in order production page |
| BR-14 | Customer can’t cancel order after accepting price from sales staff |
| BR-15 | An order must only contain a single product |
| BR-16 | Only sales staff can view customers’ requests |
| BR-17 | Only sales staff can create orders |
| BR-18 | Sales staff can only create new order once a request is approved |
| BR-19 | All products have 6 months of warranty time |

## Application Message List

|  |  |  |  |
| --- | --- | --- | --- |
| Pages | Message Type | Context | Content |
| Login | Toast info | form valid | Đang xử lý đăng nhập... |
| Toast success | email - password correct | Đăng nhập thành công |
| Toast warning | invalid email-password | Đăng nhập thất bại! |
| Toast error | lost connection to BE | Lỗi kết nối tới máy chủ |
| Input error | email empty | Email không được để trống |
| email not valid one | Email không hợp lệ |
| password empty | Mật khẩu không được để trống |
| password length less than 6 | Mật khẩu phải có tối thiểu 6 ký tự |
| Input valid | email valid | Email hợp lệ |
| password valid | Mật khẩu hợp lệ |
| Signup | Toast info | form valid | Đang xử lý đăng ký... |
| Toast success | email not signup yet | Đăng ký thành công |
| Toast warning | email already used | Đăng ký không thành công! |
| Toast error | lost connection to BE | Lỗi kết nối tới máy chủ |
| Input error | fullname empty | Họ và tên không được để trống |
| fullname not valid one | Họ và tên không hợp lệ |
| phone empty | Số điện thoại không được để trống |
| phone not valid one | Số điện thoại không hợp lệ |
| email empty | Email không được để trống |
| email not valid one | Email không hợp lệ |
| password empty | Mật khẩu không được để trống |
| password length less than 6 | Mật khẩu phải có tối thiểu 6 ký tự |
| password\_confirm empty | Mật khẩu nhập lại không được để trống |
| password\_confirm not match | Mật khẩu nhập lại không khớp |
| checkbox agrement empty | Vui lòng đồng ý với điều khoản sử dụng và chính sách bảo mật |
| Input valid | fullname valid | Họ và tên hợp lệ |
| phone valid | Số điện thoại hợp lệ |
| address | Địa chỉ hợp lệ |
| email valid | Email hợp lệ |
| password valid | Mật khẩu hợp lệ |
| password\_confirm valid | Mật khẩu nhập lại hợp lệ |
| Landing | Toast info | if login success | Chào mừng [username] đến với cửa hàng của chúng tôi |
| Dashboard | Toast info | is token authorized | Đang lấy thông tin |
| Toast success | success | Lấy thông tin thành công |
| Toast error | lost connection to BE | Lỗi kết nối tới máy chủ |

# Software Design Description

## System Design

System Architecture Overview

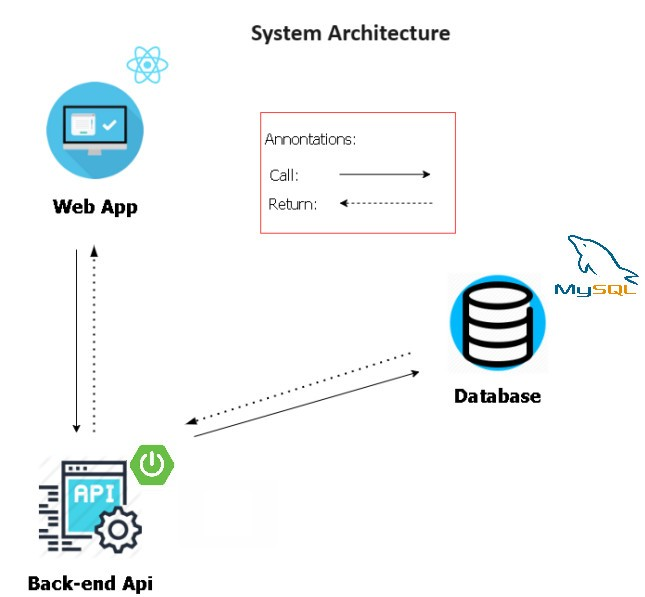


Figure 8 - System architecture

-Luxuria consists of two components, the Spring Boot web server provides APIs, ReactJS is used for the web application.

## Package Diagram

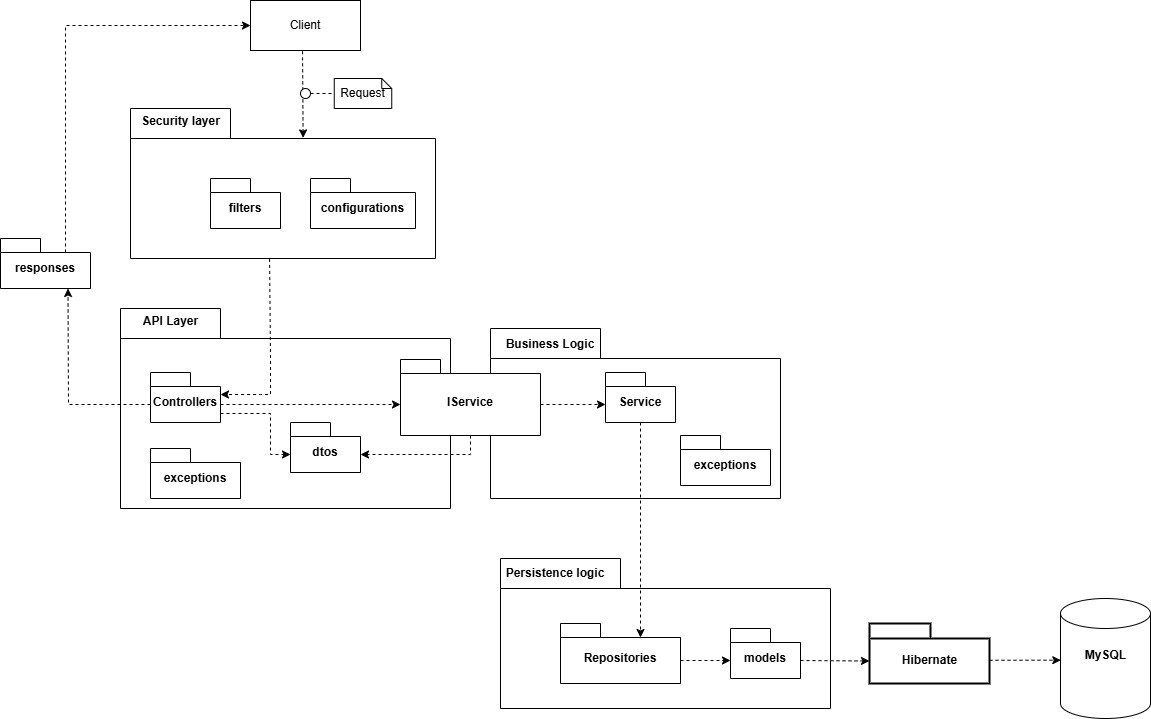


Figure 11 – Package Diagram

## Database Design

A screenshot of a computer

Description automatically generated

Figure 12- MySQL Database Design

***Table Descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | roles | Describe roles in the system |
| 02 | users | Describe users in the system |
| 03 | tokens | Describe tokens in the system |
| 04 | requests | Describe requests in the system |
| 05 | orders | Describe orders in the system |
| 06 | processes | Describe processes in the system |
| 07 | states | Describe states in the system |
| 08 | products | Describe products in the system |
| 09 | product\_data | Describe products’ data in the system |
| 10 | categories | Describe categories in the system |
| 11 | gold | Describe gold in the system |
| 12 | gems | Describe gems in the system |
| 13 | warranties | Describe warranties in the system |
| 14 | order\_state\_histories | Describe states history in the system |
| 15 | gold\_prices | Describe gold price in the system |

Table 37 - Table Data Descriptions

***Attribute Data Dictionary***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Domain** | **Null** |
| roles | id {PK} | Unique identifier of role | int | No |
| name | Name of role | varchar(10) | No |
| tokens | id {PK} | Unique identifier of token | int | No |
| token | Name of token | varchar(255) | No |
| token\_type | Type of token | varchar(50) | No |
| revoked | Verify if token is revoked or not | bit | No |
| expired | Verify if token is expired or not | bit | No |
| user\_id {FK} | Foreign key, unique identifier of user | int | No |
| users | id {PK} | Unique identifier of user | int | No |
| full\_name | Full name of user | varchar(100) |  |
| email | Email of user | varchar(100) | No |
| phone\_number | Phone number of user | varchar(20) | No |
| password | Password of user | varchar(255) | Yes |
| role\_id {FK} | Foreign key, unique identifier of role | int | No |
| is\_active | Verify if user is active or not | bit | No |
| requests | id {PK} | Unique identifier of request | int | No |
| created\_by {FK} | Foreign key, unique identifier of user | int | No |
| created\_at | Request create date | datetime | No |
| is\_sales\_staff\_approved | Verify if request is approved by sales staff or not | bit | No |
| is\_active | Verify if request is active or not | bit | Yes |
| orders | id {PK} | Unique identifier of order | int | No |
| request\_id {FK} | Foreign key, unique identifier of request | int | No |
| product\_id {FK} | Foreign key, unique identifier of product | int | No |
| order\_created\_at | Order create date | datetime | No |
| process\_id {FK} | Foreign key, unique identifier of process | int | No |
| state\_id {FK} | Foreign key, unique identifier of state | int | No |
| is\_customer\_approved | Verify if order is approved by customer or not | bit | No |
| is\_active | Verify if order is active or not | bit | No |
| processes | id {PK} | Unique identifier of process | int | No |
| name | Name of process | varchar(100) | No |
| description | Description of process | varchar(255) | No |
| states | id {PK} | Unique identifier of state | int | No |
| name | Name of state | varchar(100) | No |
| description | Description of state | varchar(255) | No |
| categories | id {PK} | Unique identifier of category | int | No |
| name | Name of category | varchar(100) | Yes |
| description | Description of category | varchar(255) | Yes |
| product\_data | id {PK} | Unique identifier of product data | int | No |
| product\_id {FK} | Foreign key, unique identifier of product | int | No |
| value | Value of product data | varchar(255) | No |
| gems | id {PK} | Unique identifier of gem | int | No |
| name | Name of gem | varchar(100) | No |
| description | Description of gem | varchar(255) | Yes |
| gold | id {PK} | Unique identifier of gold | int | No |
| name | Name of gold | varchar(100) | No |
| description | Description of gold | varchar(255) | Yes |
| products | id {PK} | Unique identifier of product | int | No |
| name | Name of product | varchar(200) | No |
| category\_id {FK} | Foreign key, unique identifier of category | int | No |
| size | Size of product | float | No |
| gold\_id {FK} | Foreign key, unique identifier of gold | int | No |
| gold\_price | Price of gold for product | float | Yes |
| gold\_weight | Weight of gold for product | float | No |
| gem\_id {FK} | Foreign key, unique identifier of gem | int | No |
| gem\_price | Gem of price for product | float | No |
| manufacturing\_fee | Manufacturing fee of product | float | No |
| total\_price | Total price of product | float | No |
| description | Description of product | varchar(255) | Yes |
| is\_original | Verify if product is original or not | bit | No |
| warranties | id {PK} | Unique identifier of warranty | int | No |
| product\_id {FK} | Foreign key. Unique identifier of product | int | No |
| user\_id {FK} | Foreign key. Unique identifier of user | int | No |
| delivery\_date | Date of delivery | date | No |
| warranty\_time | Time until warranty expires | varchar(50) | No |
| order\_state\_histories | id {PK} | Primary key. Unique identifier for each order state history. | int | No |
| order\_id {FK} | Foreign key. Unique identifier of order | int | No |
| state\_id {FK} | Foreign key. Unique identifier of state | int | No |
| create\_by {FK} | Foreign key. Unique identifier of user that create state | int | No |
| date\_time | Timestamp of when this state change occurred for the order | datetime | No |
| description | Provide additional context about the state | varchar(255) | Yes |

Table 38 - Attribute Data Description

## 4. Detailed Design

* 1. Login Feature
     1. Class diagram

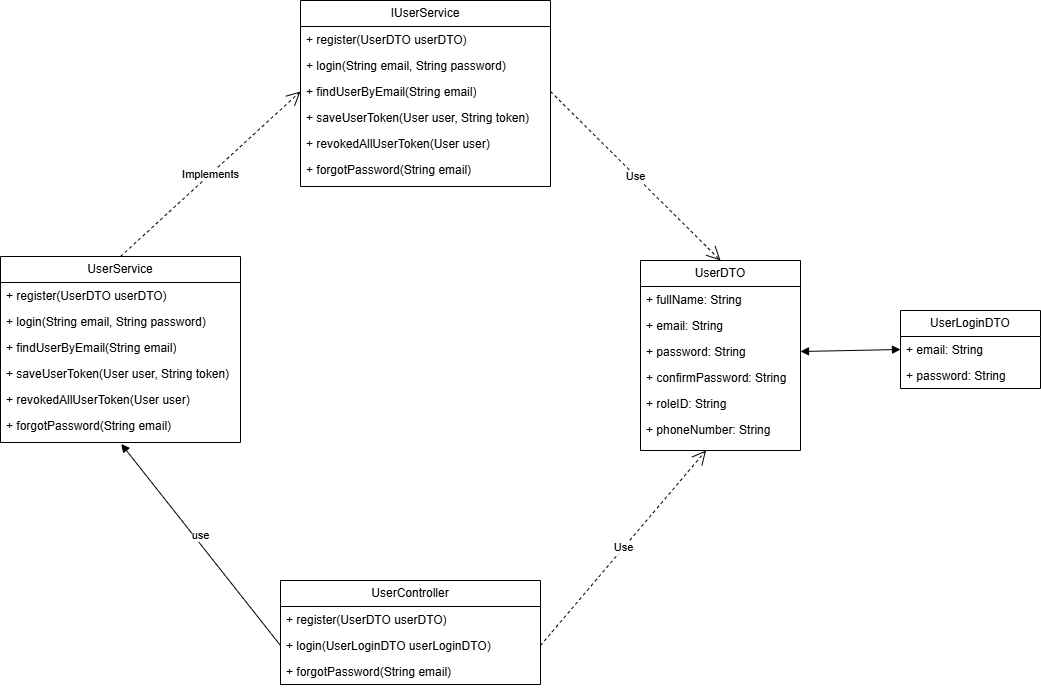


Figure … : Login Feature

* + 1. Class Specification

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 1 | register(UserDTO userDTO) | handles register to system |
| 2 | login(UserLoginDTO userLoginDTO) | handles login to system |
| 3 | forgotPassword(String email) | handles help user login to system when forgot password |

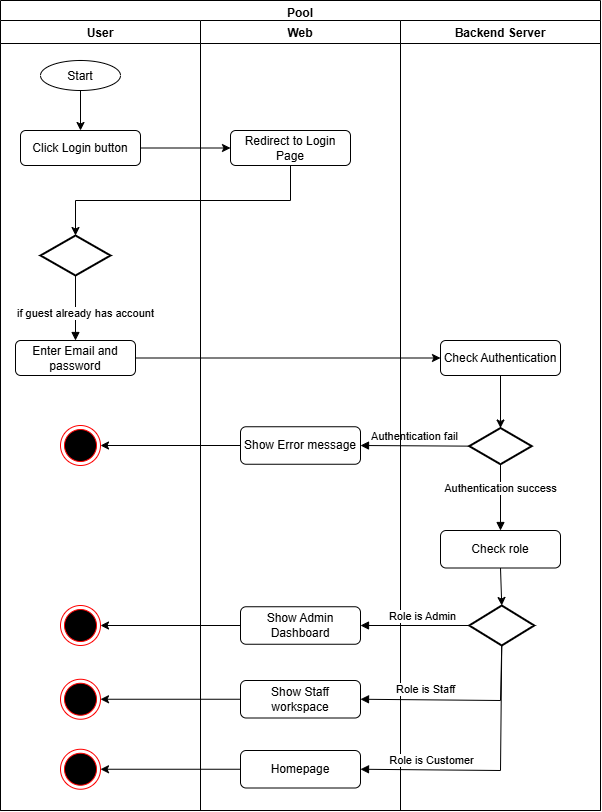
* + 1. Login Activity Diagram

Figure 12 - Login Activity Diagram

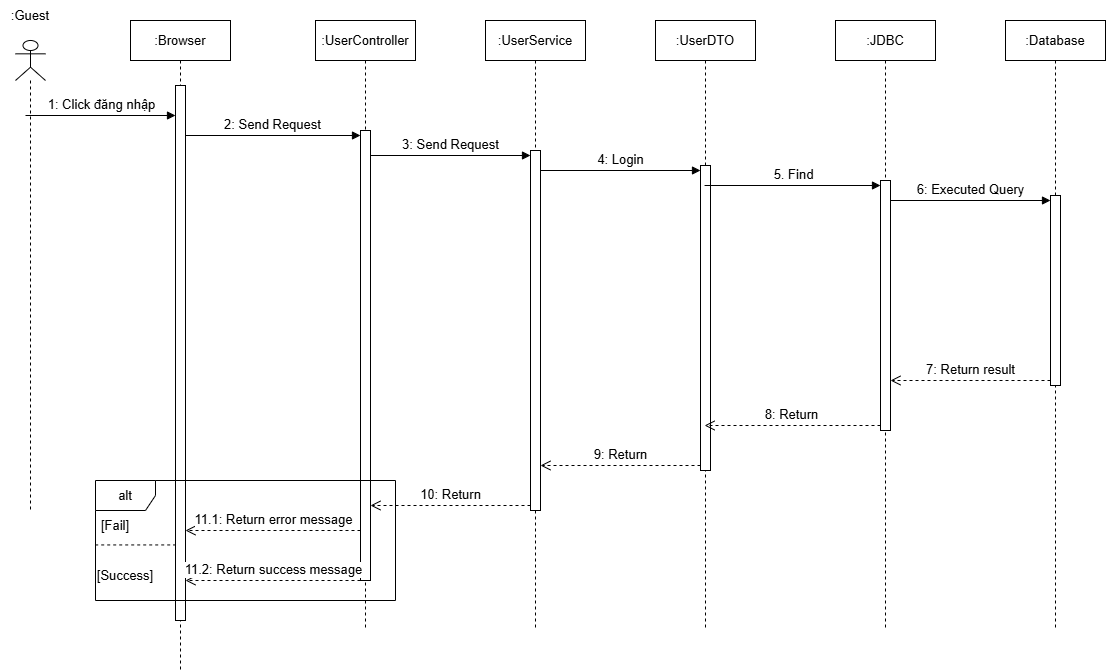
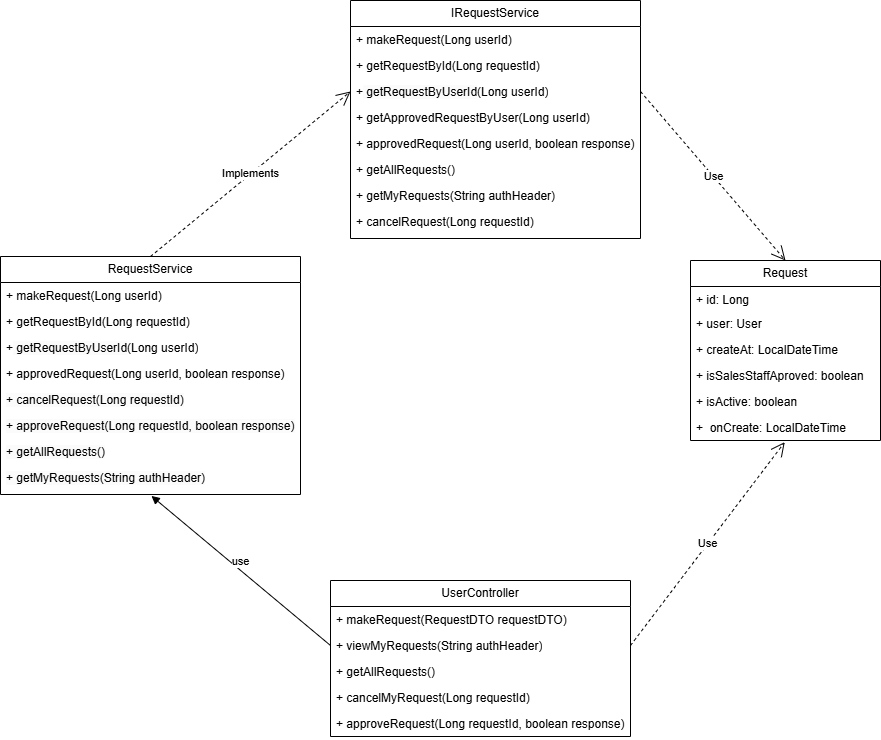
* + 1. Login Sequence Diagram

Figure 13 - Login Sequence Diagram

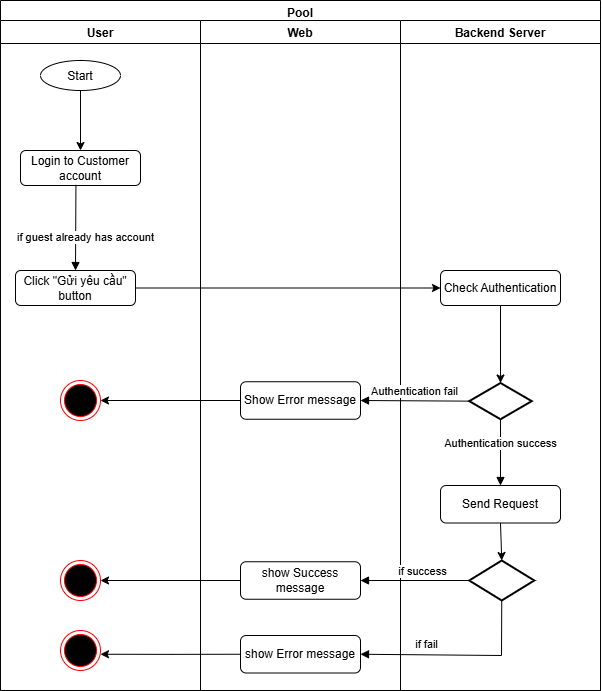
* 1. Request Management Feature
     1. Class Diagram



* + 1. Class Specification

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 1 | makeRequest(RequestDTO requestDTO) | Handles register to system |
| 2 | viewMyRequets(String authHeader) | Handles login to system |
| 3 | getAllRequest() | Handles help user login to system when forgot password |
| 4 | cancelMyRequest(Long requestId) | Handles cancel request for user |
| 5 | approveRequest(Long requestId, Boolean response) | Handles approve user’s request for Customer |

* + 1. Send Request Activity Diagram

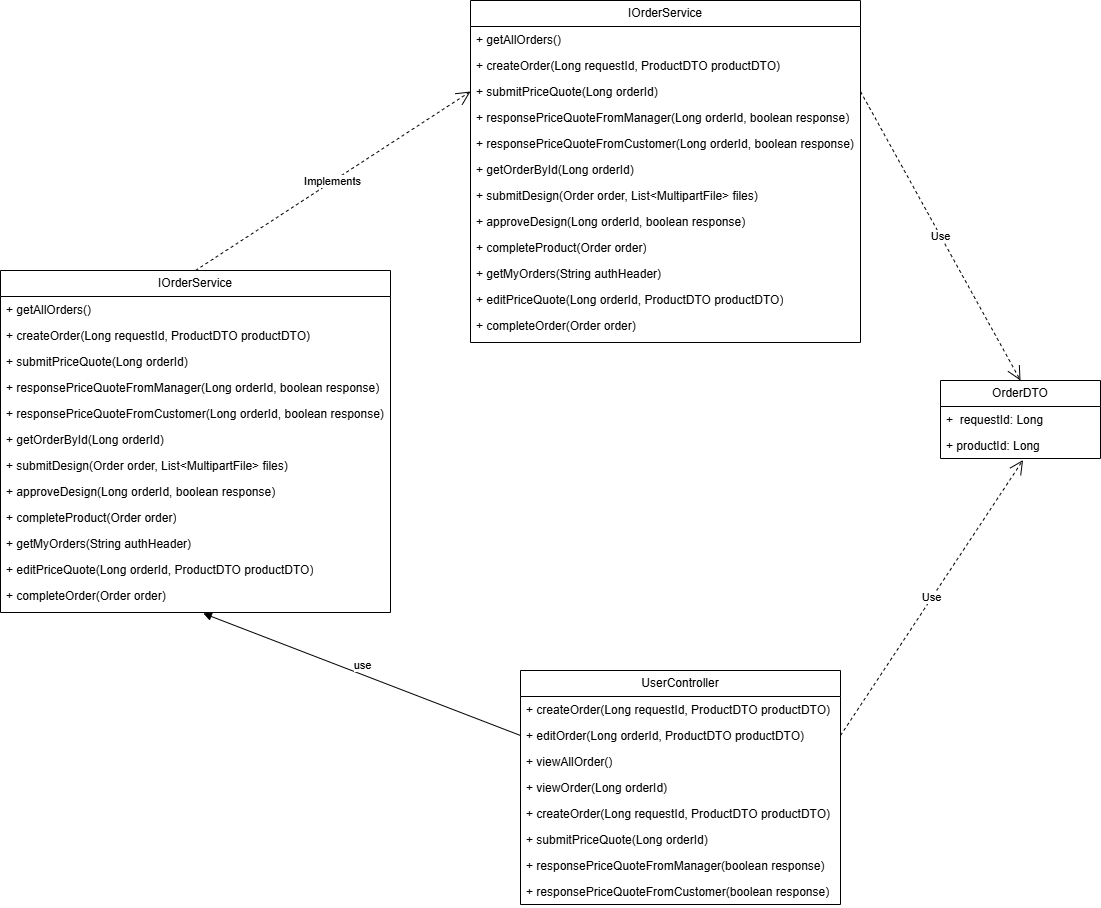


* + 1. Send request Sequence Diagram

A diagram of a diagram

Description automatically generated

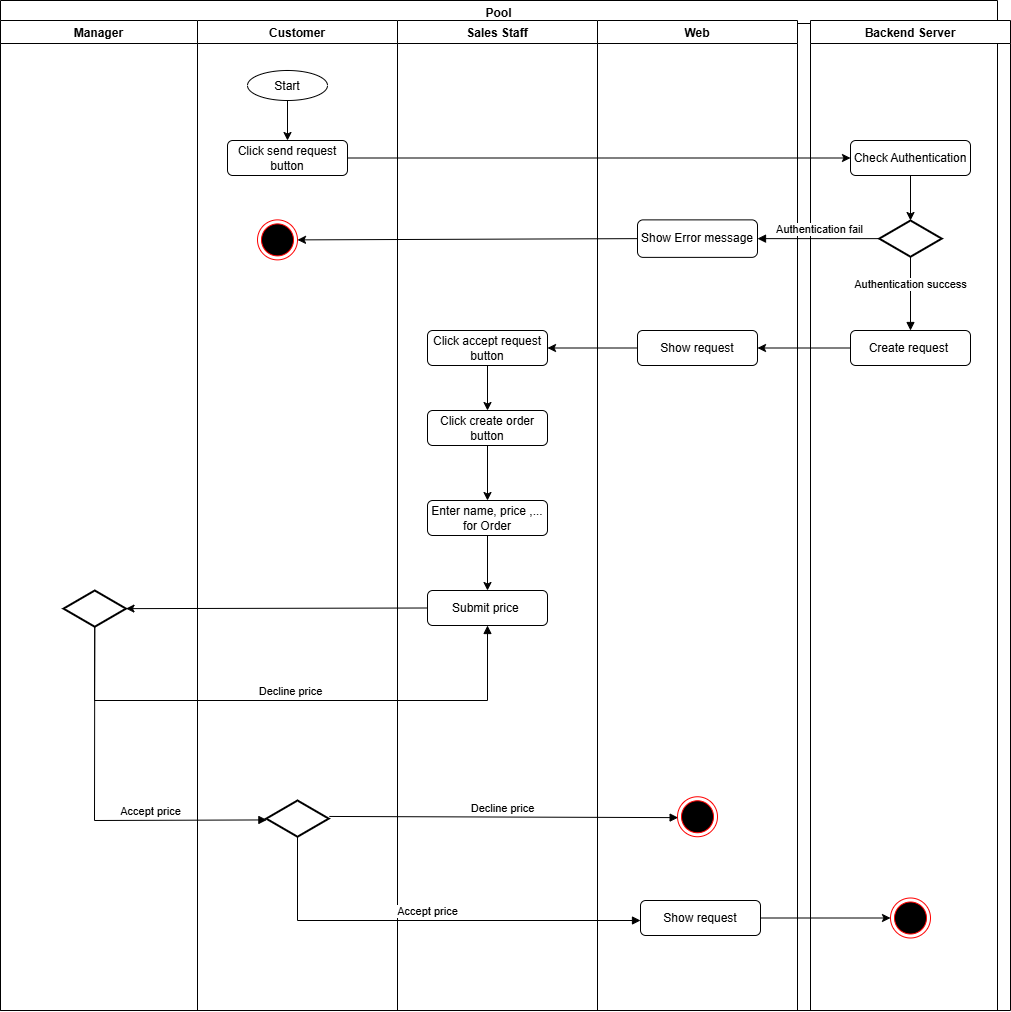
* 1. Create order feature
     1. Create Order Class Diagram



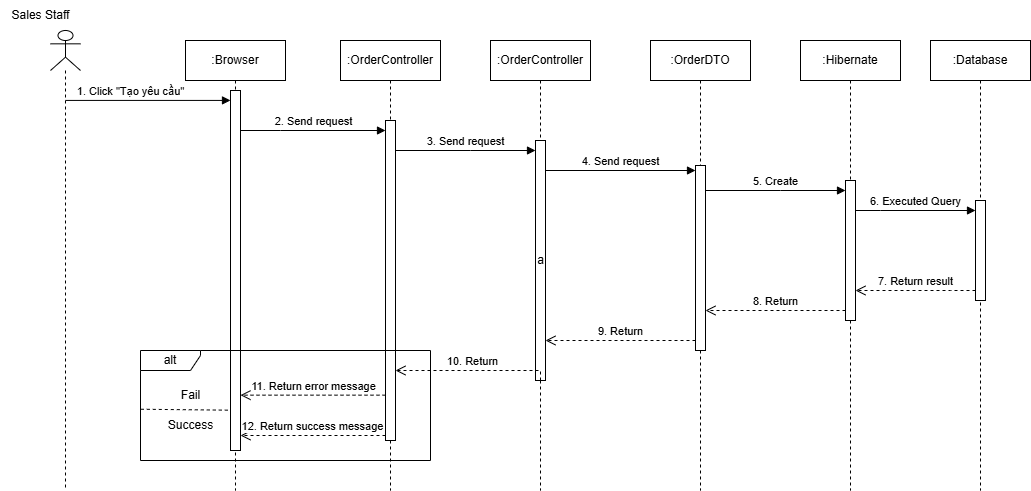
* + 1. Class Specification

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 1 | createOrder(Long requestId, ProductDTO) | Handles create order |
| 2 | editOrder(Long orderId, ProductDTO productDTO) | Handles edit order to system |
| 3 | viewAllOrder() | Handles view all order |
| 4 | viewOrder(Long orderId) | Handles cancel request for user |
| 5 | submitPriceQuote(Long orderId) | Handles submit price from Customer to Manager |
| 6 | responsePriceQuoteFromManager(boolean response) | Handles response price from Manager |
| 7 | responsePriceQuoteFromCustomer(boolean response) | Handles response prince from Customer |

* + 1. Create Order Activity Diagram



* + 1. Create Order Sequence Diagram



# Release Package & User Guides

## Deliverable Package

### Source codes & documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Items** | **Sub-Items** | **Type** | **Version** |
| **Code Package** | | | | |
| 1 | Web Module | Luxuria\_FE.zip | New | 1.0.0 |
| 2 | Web API Module | Luxuria\_BE.zip | New | 1.0.0 |
| **Database** | | | | |
| 1 | Initialise Database | Luxuria.sql | New |  |
| **Documents** | | | | |
| 1 | Final Project Report | Luxuria-FinalReport\_Full.docx | New | 1.0.0 |
| 2 | Test Cases Document |  | New |  |

Table 54 - Source codes & documents

### Known Issues, Limitations & Restrictions

* Installers for each project haven’t been packaged
* Database must be initialized by using database scripts

## Installation Guides

### System Requirements

* + - 1. ***Hardware requirements***

|  |  |  |
| --- | --- | --- |
| **PC** | **Minimum** | **Recommend** |
| Internet connection | LAN, internet access capability | LAN, Wi-fi (16 Mbps) |
| Operating System | Windows 7 SP1/8.1/10/11/Window  Server 2016 | Window 10 or 11 |
| Processor | Intel Core i5 1.5Ghz | Intel Core i7 2.6Ghz |
| Memory | 4GB RAM | 8GB RAM up |
| Storage | 4GB | 6GB up |
| Web Browser | Chrome (v69) | Latest stable version |

Table 55 - <System Requirements> Web application

* + - 1. **Software requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **PC** | **Minimum** | **Recommend** | **Description** |
| IDE | Visual Studio Code, IntelliJ IDEA | Latest stable version | Used to edit and run code |
| Java SE Development Kit | Java SE Development Kit 18.0.0.0 up | Java SE Development Kit 18.0.2.1 | Used to run backend server |
| ReactJS(Javascript) | 18.3.0 up | 18.3.1 | Used to run frontend application |
| MySQL | MySQL 5.3 | MySQL 8.0 | Used to manage database |

### Setup Files

Luxuria.sql: Database script for system

Luxuria\_BE.zip: API for services

Luxuria\_FE.zip:

### Installation Instruction

* + - 1. ***Setup Database***

Install MySQL 8.0 by following guidelines link:

[MySQL :: Download MySQL Community Server](https://dev.mysql.com/downloads/mysql/8.0.html)

* + - 1. ***Setup Backend API***

+ Install IntelliJ IDEA by following guidelines link:

[Download IntelliJ IDEA – The Leading Java and Kotlin IDE (jetbrains.com)](https://www.jetbrains.com/idea/download/?fromIDE=&section=windows)

+ Install Java SE Development Kit by following guidelines link:

[Java Archive Downloads - Java SE 18 (oracle.com)](https://www.oracle.com/java/technologies/javase/jdk18-archive-downloads.html)

A screenshot of a computer

Description automatically generated+ Extracting: Luxuria\_BE.zip

Figure 48- Setup Backend API

+ Open Luxuria\_BE folder with IntelliJ IDEA

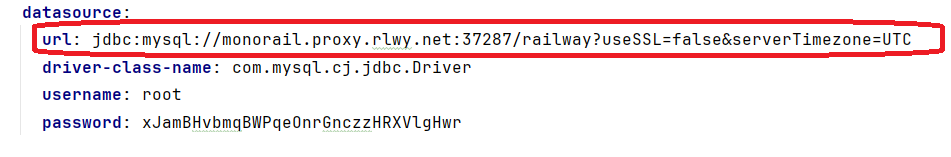
+ Config your database connection in application.yml

Figure 49 - Setup Backend API

+ Run project

A screenshot of a computer

Description automatically generated

Figure 50 - Setup Backend API

+ Test project in browser by go to url: https://localhost:44310/index.html

+ Username: admin pass 12345678

+ If having trouble when open project in IntelIJ IDEA please consider:

A screenshot of a computer

Description automatically generated

1. Reload All Maven Project

Figure 51 – Reload All Maven Project

A screenshot of a computer

Description automatically generated2. Rebuild Module

Figure 51 - Setup Backend API

* + - 1. ***Setup Management Web***

+ Install Visual Studio Code by following guidelines link: <https://thaynhuom.edu.vn/cai-visual-studio-code-lap-trinh-c-c/>

+ Install npm through Visual Studio Code or you can install npm via NodeJS <https://nodejs.org/en/>

+ Check after installation

A black background with white text

Description automatically generated

+ Extract Luxuria\_FE.zip and open folder with Visual Studio Code

+ Open terminal and enter npm i:

A black and white logo

Description automatically generated

Figure 52- Setup Management Web

+ Check after installation:



 + Enter npm i vite:

+ Change baseUrl based on your backend configuration

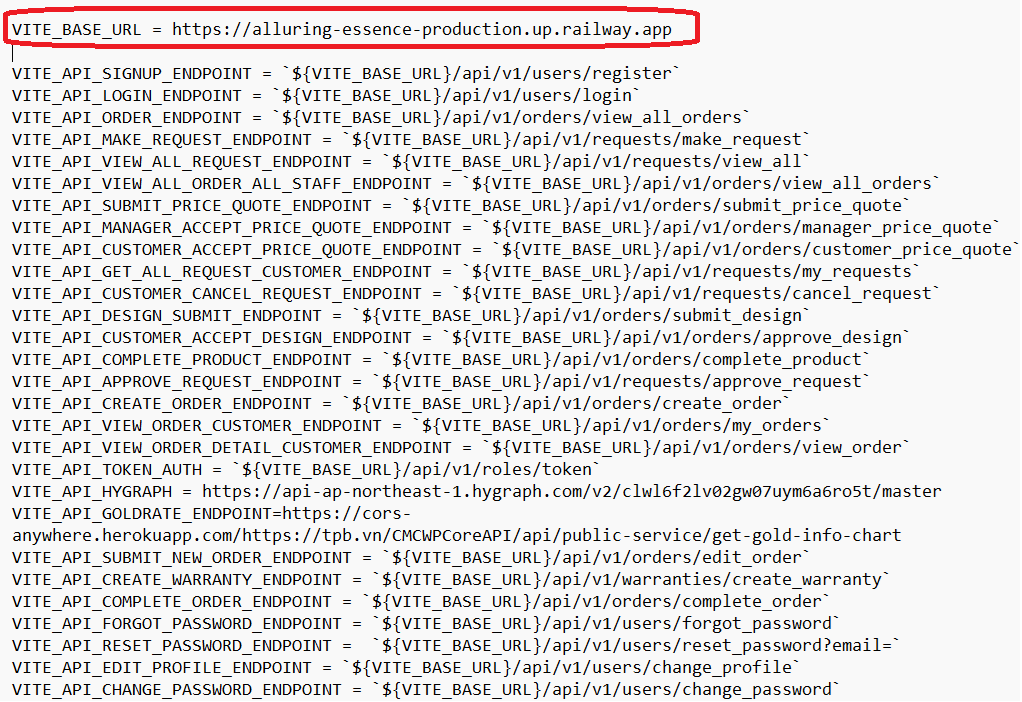


Figure 53 - Setup Management Web

+ In terminal, enter npm run dev to start



A screen shot of a computer

Description automatically generated

Figure 54 - Setup Management Web

+ Project start with http://localhost:5173/

