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**

|  |  |
| --- | --- |
|  | |
| **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. Ton That Hoang Minh |

1

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

# Project Introduction

## Overview

### Project Information

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

### Project Team

1. ***Supervisors***

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Role** | **Email** | **Mobile** |
| Ton That Hoang Minh | Lecturer | [minhtth5@fe.edu.vn](mailto:minhtth5@fe.edu.vn) |  |

Table 1 - Supervisors

1. ***Team Members***

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

Table 2 - Team Members

* 1. Product Background

2

## Project Scope & Limitations

### Major Features

* **Guest:**

+ FE-01: Register account

* **Customer:**

+ FE-02: Sent request

+ FE-03: View order detail

+ FE-04: View order process

+ FE-05: Decide on the order

+ FE-06: Cancel order

* **Sales Staff:**

+ FE-07: View request

+ FE-08: View all order detail

+ FE-09: View order process

+ FE-10: Edit customer order

+ FE-11: Cancel order

+ FE-12: Submit order's price to manager

+ FE-13: Export receipt and warranty

* **Manager:**

+ FE-14: View all order detail

+ FE-15: View order process

+ FE-16: Approve or decline order's price

* **Design Staff:**

+ FE-17: View 'approved' order detail

+ FE-18: Upload 3D design for customer order

+ FE-19: Approve or decline order's price

3

* **Production Staff:**

+ FE-20: View 'approved 3D design' order detail

+ FE-21: Report finished order

* **Admin:**

+ FE-09: View all order detail

+ FE-10: View order process

+ FE-11: View dashboard

# Project Management Plan

## Overview

### Scope & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort (man-days)** |
| ***1*** | ***Requirement analysis*** |  | ***10*** |
| 1.1 | Define background and context of product | Complex | 7 |
| 1.2 | Find problems of users | Medium | 2 |
| 1.3 | Existed system analysis | Simple | 1 |
| ***2*** | ***Technology training*** |  | ***12*** |
| 2.1 | Learn Spring Boot framework | Complex | 6 |
| 2.2 | Learn ReactJS framework | Complex | 6 |
| ***3*** | **Workspace setup** |  | ***3*** |
| 3.1 | Setup resources storage workspace | Medium | 2 |
| 3.2 | Setup resources and version control | Medium | 1 |
| ***4*** | **Document writing** |  | ***17*** |
| 4.1 | Write project introduction | Medium | 2 |
| 4.2 | Write project management plan | Medium | 3 |
| 4.3 | Write software requirement specification | Complex | 6 |
| 4.4 | Write software design description | Complex | 6 |
| ***5*** | **Interface design** |  | ***10*** |

|  |  |  |  |
| --- | --- | --- | --- |
| 5.1 | Design web application interface | Complex | 10 |
| ***6*** | ***Implementation*** |  | **14** |
| 6.1 | Implement authentication/authorization  features | Complex | 2 |
| 6.2 | Implement customer features | Complex | 3 |
| 6.3 | Implement sales staff features | Complex | 4 |
| 6.4 | Implement manager features | Medium | 1 |
| 6.5 | Implement design staff features | Medium | 1 |
| 6.6 | Implement production staff features | Medium | 1 |
| 6.7 | Implement admin features | Medium | 2 |
| ***7*** | ***Testing*** |  |  |
| 7.1 | Unit Test | Complex |  |
| 7.2 | Integration Test | Complex |  |
| 7.3 | System Test | Complex |  |
| ***8*** | ***Closing*** |  |  |
| 8.1 | Maintenance | Medium |  |
| 8.2 | Prepare slides for defense | Medium |  |

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

Table 3 - Scope & Estimation

4

### 1.2 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. |

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.

5

+ 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: Each 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.

6

### Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| Spring Boot | Thái, Lương | 1 week | Mandatory |
| ReactJS | Vũ, Long | 1 week | Mandatory |
| GitHub | All member | 4 days | Mandatory |

Table 5 - Training plan

## Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint #** | **Sprint objective** | **Duration** | **Deliverable** |
| 1 | Project Plan Document, SRS Document | 6 days | Project Planning, Software Requirement |
| 2 | Design Document | 5 days | Architecture Design, Basic Design, Detail Design and Database Design |
| 3 | Authentication Module, User Module, Role Module | 14 days | Code & System test cases |

Table 6 - Project deliverables

7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Responsibility** | **Trịnh Công Thái** | **Phạm Nguyên Vũ** | **Lê Vũ Đức Lương** | **Chu Phan Nhật Long** |
| Backend for system | x |  | x |  |
| Front end web |  | x |  | x |
| Other | x | x | x | x |

## Responsibility Assignments

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 | * Raise opinions, | Always | Face to |

8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | members | problems   * Ask for help from other members * Report working status to leader |  | Face, Messenger, Discord, Zalo |

Table 8 - Project Communications

### 5.2 External Interface

**a. FU Contacts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person**  **(Name, position)** | **Contact address (Email, telephone)** | **Responsibility** |
| Advisor | Ton That Hoang 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. We create documents on Discord based on the document template, then divide the document writing work for members based on their predefined responsibilities. Each member can see the results of the other members' work instantly.

9

### 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 |
| **Source Version Control** | GitHub |
| **Project Management Tool** | Zalo |
| **Framework** | ReactJS (Frontend), Spring Boot (Backend) |
| **Programming Languages** | Java |
| **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.

10

### 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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

* + 1. 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.

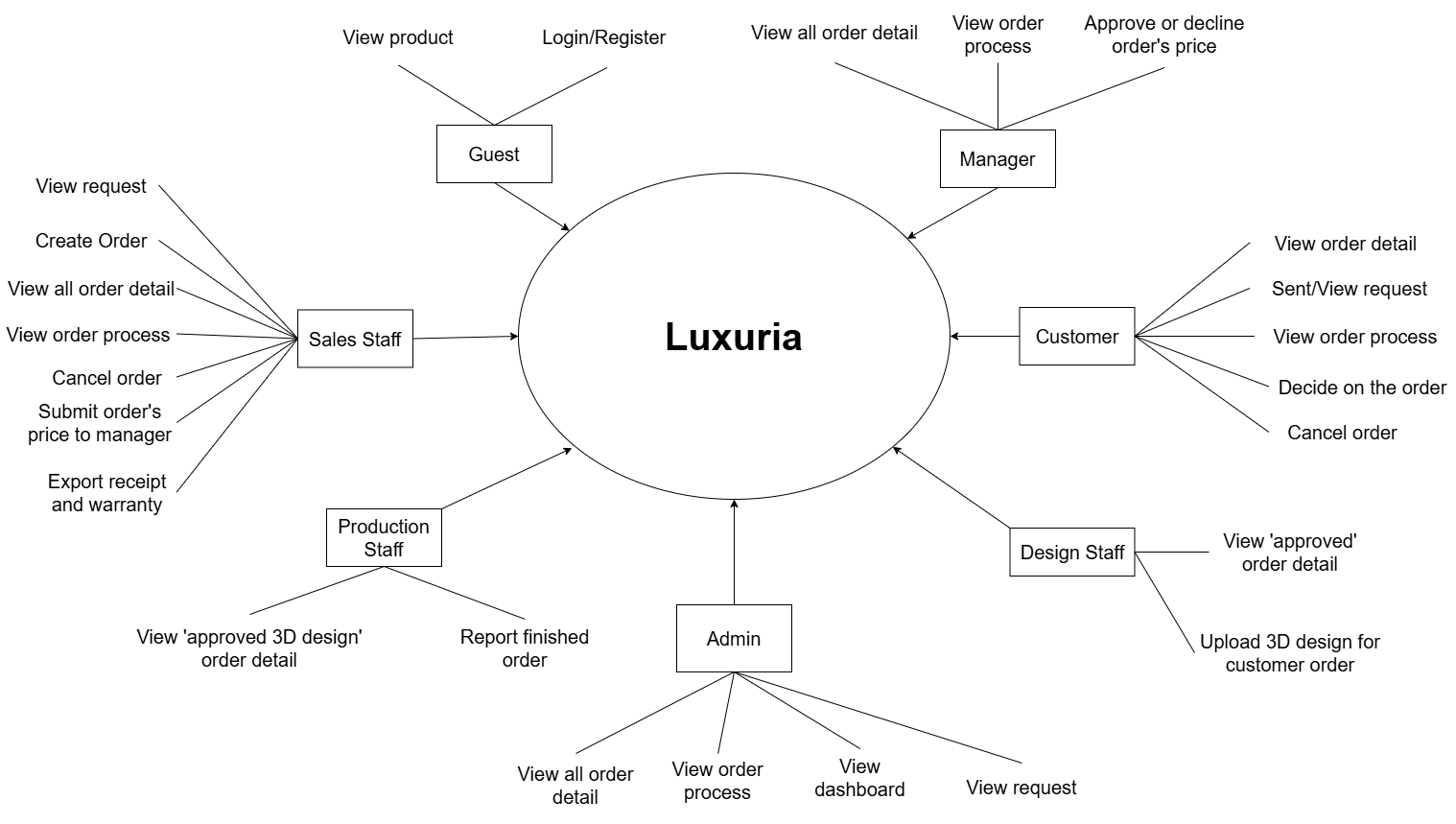


Figure 2 - Context diagram

## 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

12

### Use Cases

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

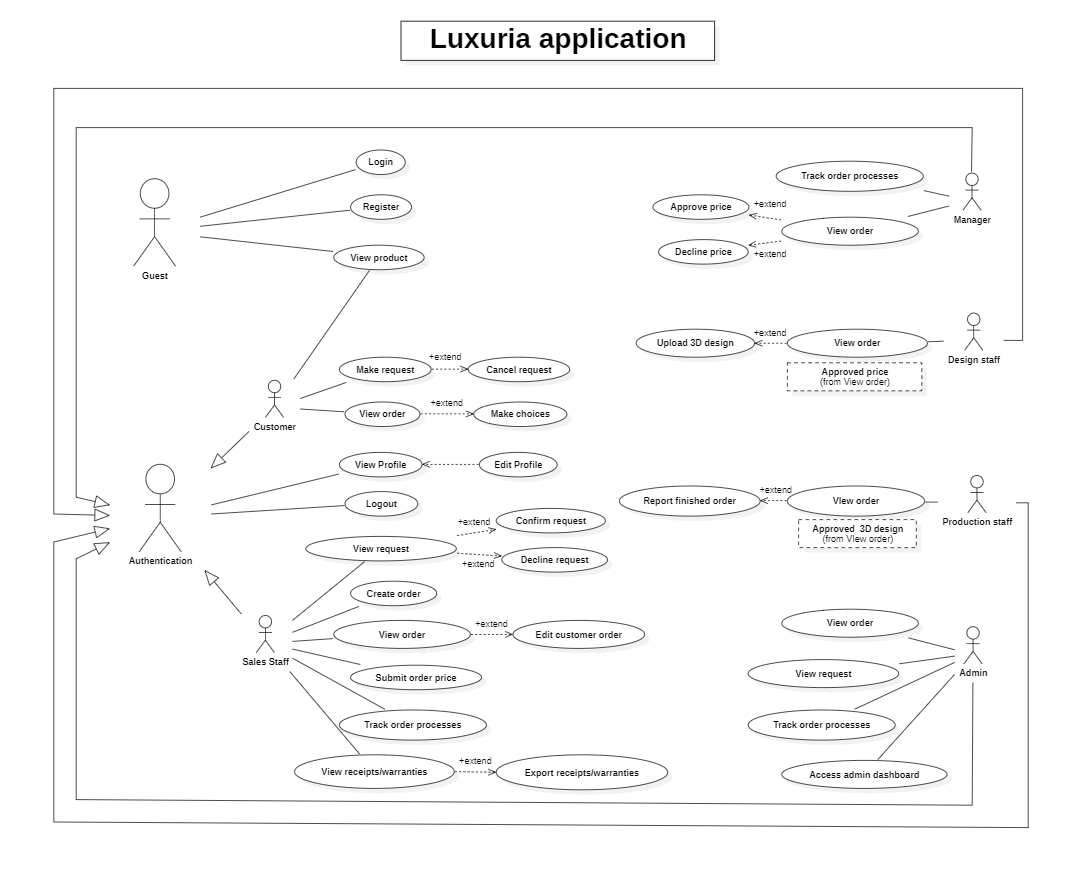


Figure 3 - <Use case overview> Luxuria Application

13

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

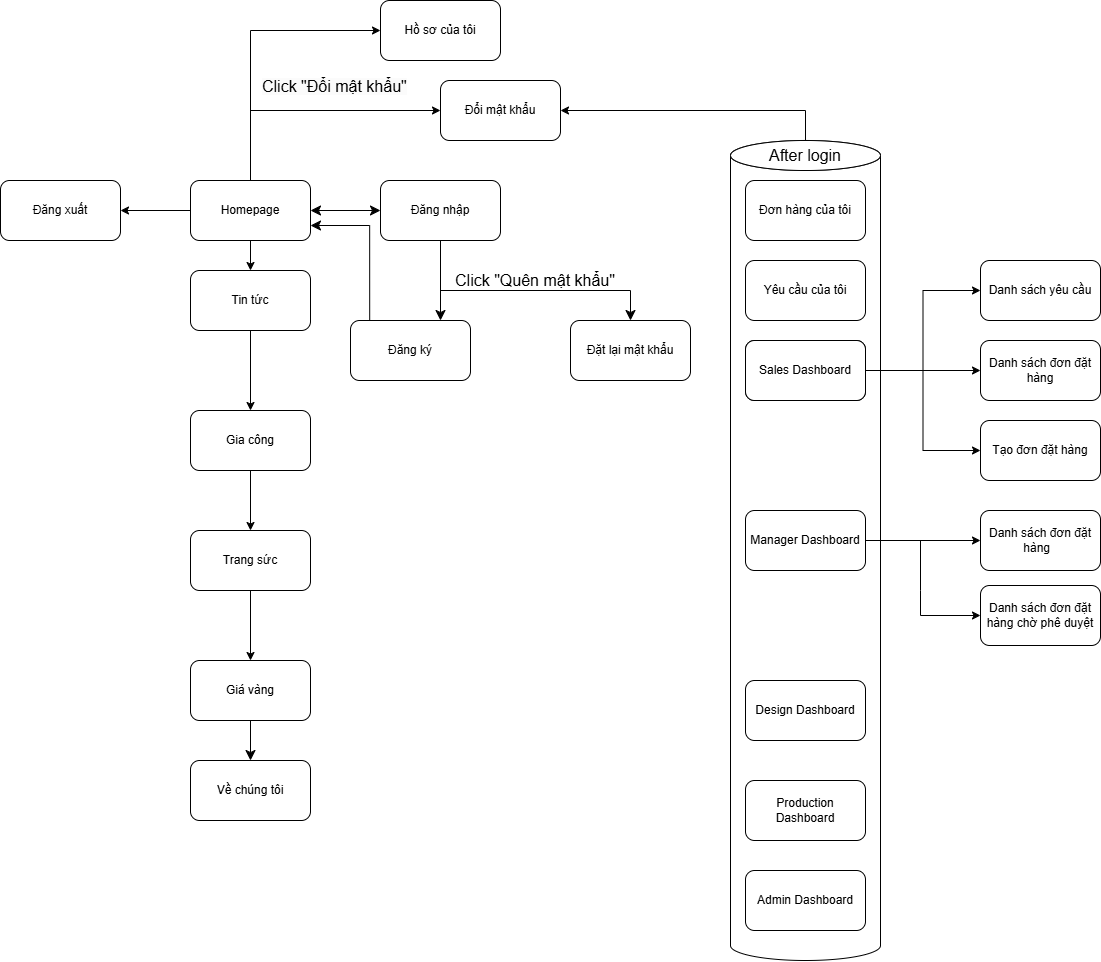
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Description** |
| UC01 | Register | Guest | Guest register for account |
| UC02 | Login | Guest | Unauthenticated user login to Luxuria to verify role, and give them access to more features |
| UC03 | Logout | All actors 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 |

* 1. 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 |  |
| 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

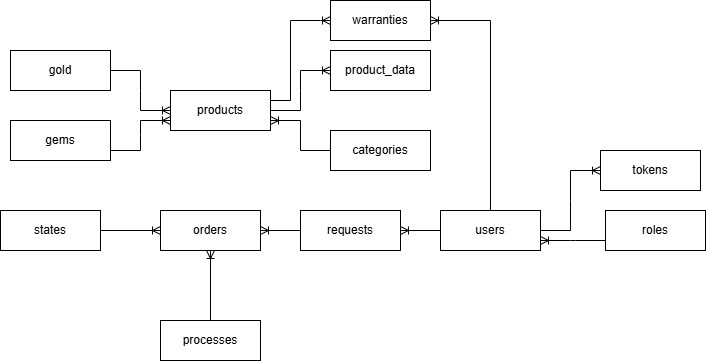


* 1. Screen Authorization

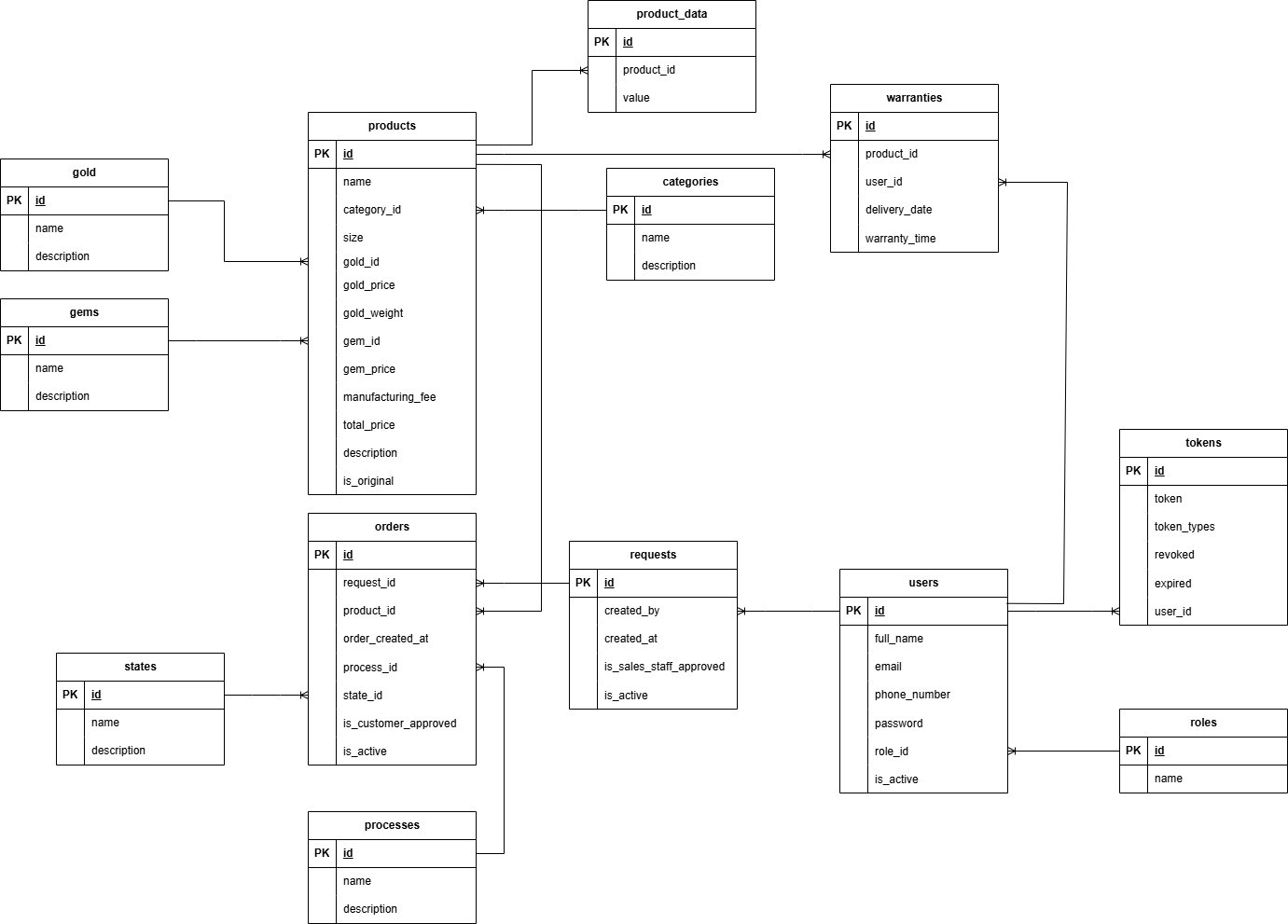
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Screen | Guest | Customer | Sales Staff | Manager | Design Staff | Production Staff | Admin |
| Homepage | X | X | X | X | X | X | X |
| Tin tức | X | X | X | X | X | X | X |
| Chi tiết tin tức | X | X | X | X | X | X | X |
| 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 |

* 1. Entity Relationship Diagram

6.2.1 Conceptual ERD



6.2.2 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 |

* 1. 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 | funname 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 |
| address empty | Địa chỉ không được để trống |
| address not valid one | Địa chỉ 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 |  | 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

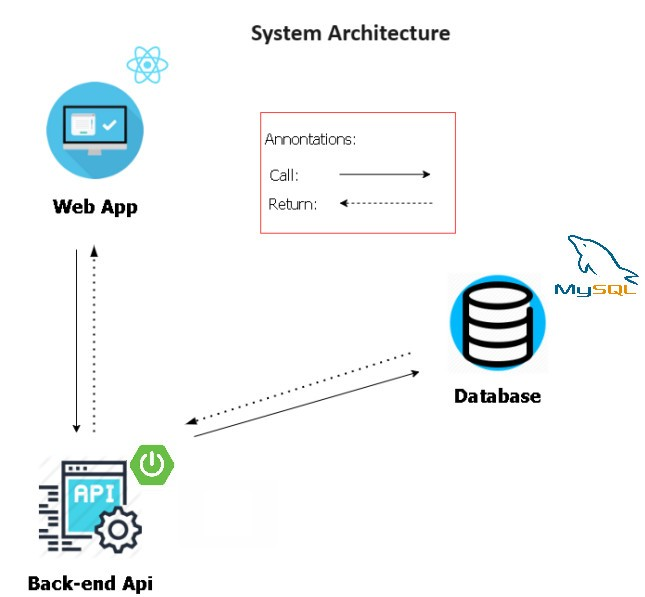
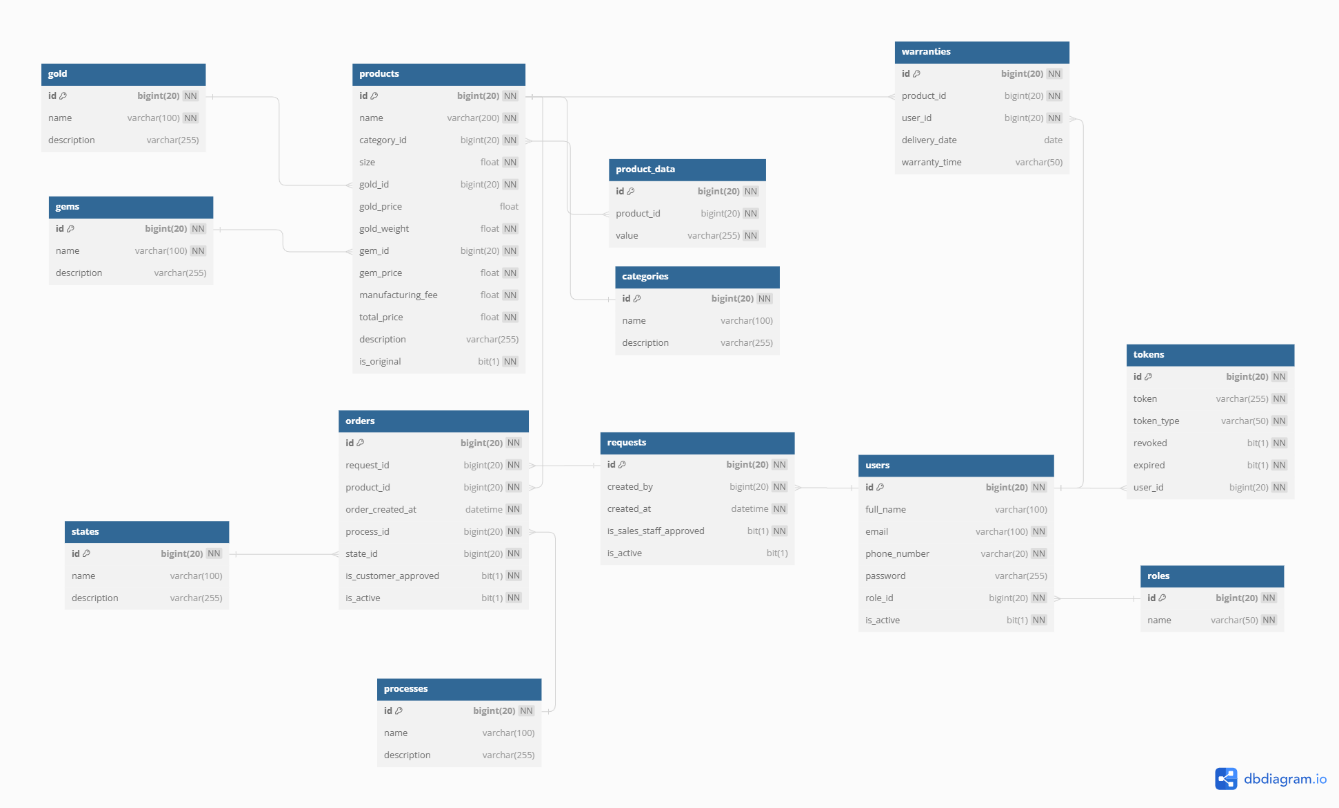


Figure 8 - System architecture

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

## Database Design

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 |

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) |  |
| description | Description of process | varchar(255) |  |
| states | id {PK} | Unique identifier of state | int | No |
| name | Name of state | varchar(100) |  |
| description | Description of state | varchar(255) |  |
| categories | id {PK} | Unique identifier of category | int | No |
| name | Name of category | varchar(100) |  |
| description | Description of category | varchar(255) |  |
| 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) |  |
| gold | id {PK} | Unique identifier of gold | int | No |
| name | Name of gold | varchar(100) | No |
| description | Description of gold | varchar(255) |  |
| 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 |

Table 38 - Attribute Data Description