# Problem Statement

**PROBLEM  
BACKGROUND**

**RELEVANCE**

**OBJECTIVES**

***Key Functionalities -***

*1. Login functionality*

*2. User role*

*3. Product*

*4. Categories*

*5. Searching and filtration of product (by cost, ratings, category)*

*6. Pagination*

*7. Add to cart*

*8. List of selected products (Add, update, delete product from cart), checkout,*

*9. Order History*

*10. Receipt of checkout*

# Functional Requirements

**Login Module Details**

For login below details are to be captured:

* User email
* Password

**Product Module Details**

To list a product the below details are to be captured:

* Product name
* Category ID
* Product ID
* Price
* Discount on Product
* Discount Price
* Description

**Tracking Module Details**

* Order ID

**Products / Search**

* Search Term
* Category / Filter
* Pagination

**User Profile**

* User ID
* User Name
* Password
* First Name
* Last Name
* Address
* Contact Number
* Role

### **Use Case Diagram**

## **<<Add more screens based on your case study>>**

**Homepage – Contact etc. in footer**

**Login Screen**

This is the default screen when application loads, post registration user can log-in. The log-in screen shows options for registration and forgot/ change password.

**Registration**

**Shipping Tracking**

**Categories List**

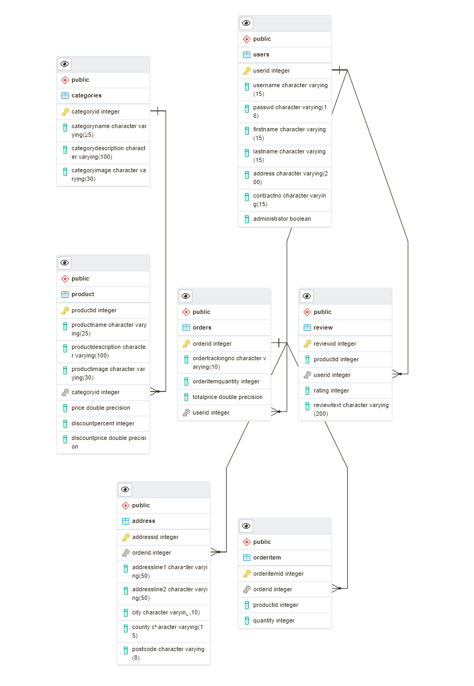
**Products / Search**

**User Profile**

**Checkout**

**Invoice**

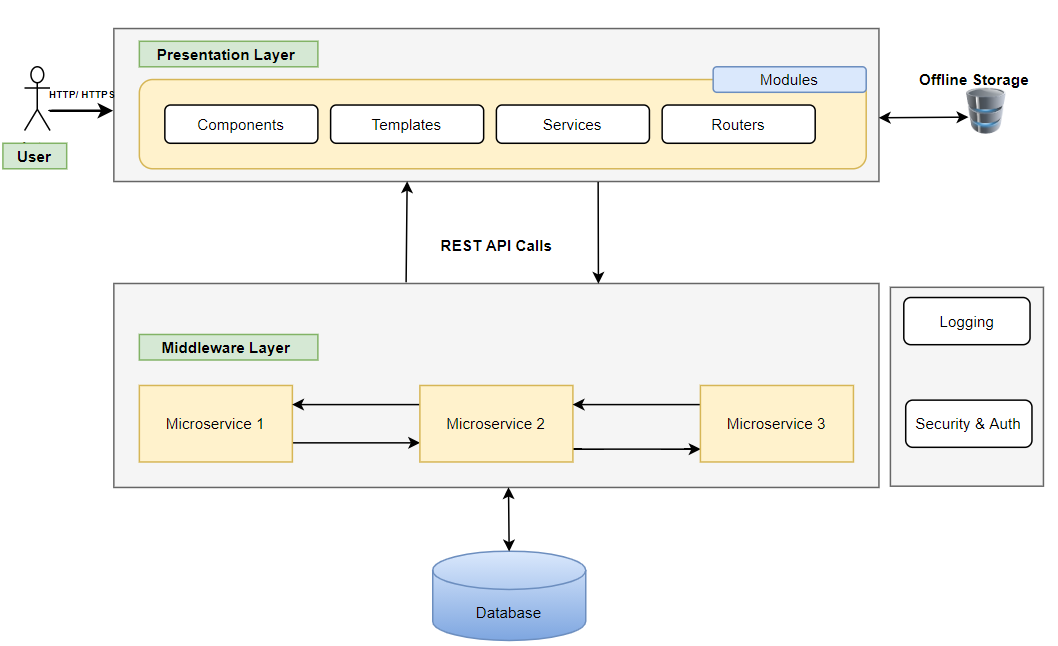
**Database Design**



# Non-Functional Requirements

* The application UI should be supported on Google Chrome and Microsoft Edge
* All the coding guidelines and code quality standards to be followed
* Proper database schema’s to be created
* Follow best practices for databases
* Unit Testing code coverage should be 80%
* Unit Testing for UI is optional
* The application should following proper naming conventions for both front-end as well as backend codes
* The application should be completed with 80% working functionalities to be accepted for final presentation

# Application Architecture



# Component Description

|  |  |  |  |
| --- | --- | --- | --- |
| # | Component | Description | Technology Stack |
| 1 | Presentation Layer | User Interface (UI) will be developed using HTML5, CSS3, Bootstrap, and Angular 10 JavaScript Framework. These application components will be responsible for rendering User visualization elements, UI processing, data binding, event wiring and command dispatching. | * HTML5, CSS3 * Angular 10+ * Bootstrap or Angular Material for Responsive Design |
| 2 | Offline Storage | Browser has ability to store the data offline | * Optional (Browser-based or offline storage) |
| 3 | Middleware Layer | Microservices are collection of services which represent business capabilities. And are highly maintainable, testable, loosely coupled and independently deployable. | * Spring Boot * Swagger API for documentation or any other tool * Spring REST API * Spring Boot Data JPA |
| 4 | Database | Database to the relation and maintain entities data in the tables. Retrieve | * MySQL/ PostgreSQL |
| 5 | Cross Cutting | **Logging** – Useful logs can provide the developer (especially when someone has to debug/maintain someone else’s code) with tremendous help when trying to understand what the code actually does.  **Authentication** - The services would be secured by username/ password or Token based authentication. Authorization can also be performed at API service-level. | * In-built Logging APIs * Integrated Tools/ DataDOG/ LogRocket * NLOG/Log4J/Logback * Spring Security * JWT |

# Project Plan Milestones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Milestones | Deliverables | Actors | Estimated Date for Demonstration |
| 1 | M1 | * Infrastructure Setup * Project Team Finalization * Selection of Case Study * Project Flow Documentation * Git Repository Folder Structure | Trainer + Dev Team |  |
| 2 | M2 | * Design Wireframes for decided modules * Define JSON structure for UI development of Modules * Complete UI functionality for Login, Logout and landing Dashboard | Dev Team |  |
| 3 | M3 | * Develop UI Screens for other modules <<name here>> * Develop UI Screen for Admin module | Dev Team |  |
| 4 | M4 | * Design Database Schema * Create ER Diagrams * Create other objects like procedures and functions * Finalize the database tables | Dev Team |  |
| 5 | M5 | * Develop Backend REST APIs * Create REST API for Login and Registration * Integrate with UI * Generate Swagger Documentation * Integrate with UI | Dev Team |  |
| 6 | M6 | * Develop remaining backend microservices * Develop REST APIs to perform REST calls * Generate Swagger Documentation * Integrate with UI * Perform server-side validations * Add Logging and Security | Dev Team |  |
| 7 | M7 | * Integrate all Modules | Dev Team +  Trainer |  |

**Definition of Done (DOD)**

* Completed activity is demonstration ready
* All the codes are at least unit tested
* Modules should pass all the validations (UI and Business layers)
* At the end of every milestone, a demo is given to the stackholders
* Project team will have minimum 3 members and all have contributed in the development
* Every member should be presenting during demonstrations
* Few assumptions can be done while developing the project. However, it needs to be discussed with the trainer