DAIRY APP DATA FLOW, CONTEXT AND DOCUMENTATION

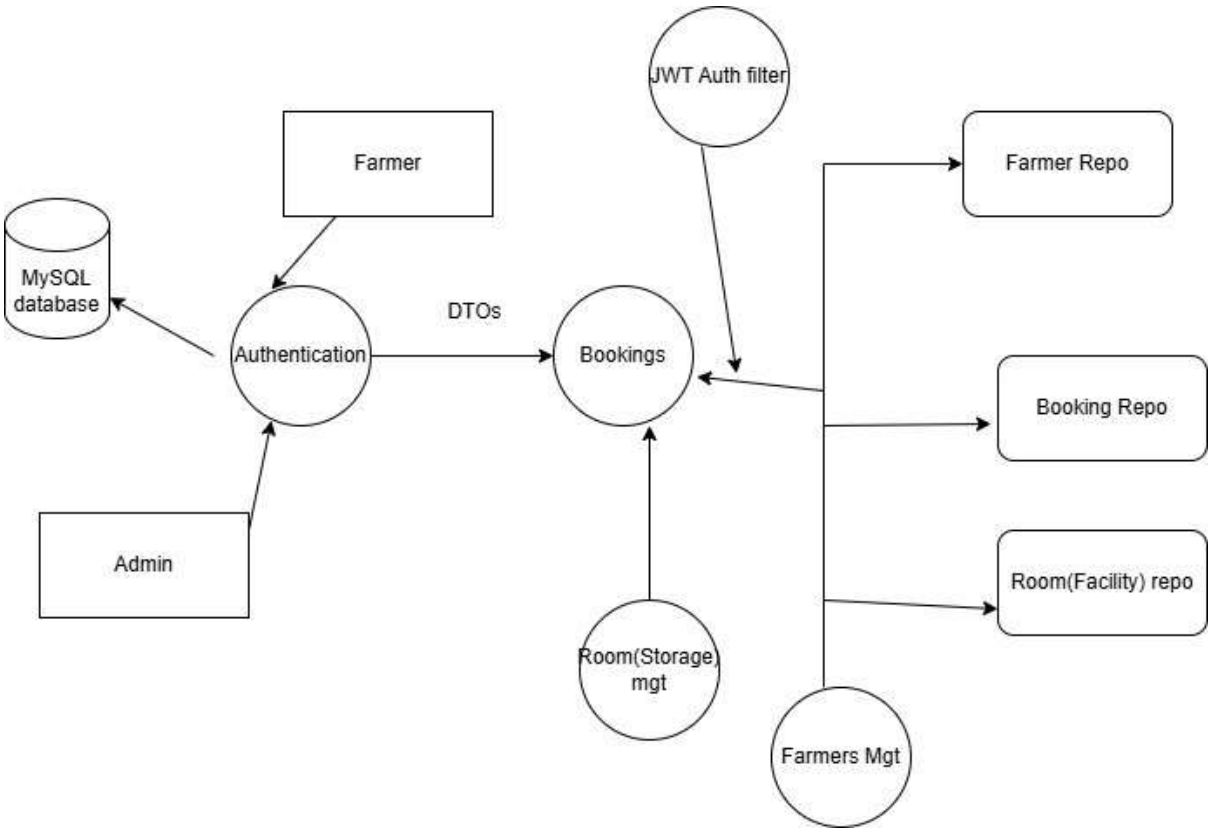
Dekow Mohamed **SCT212-0085/2023**

Wanjala Ian Mandila **SCT212-0162/2022**

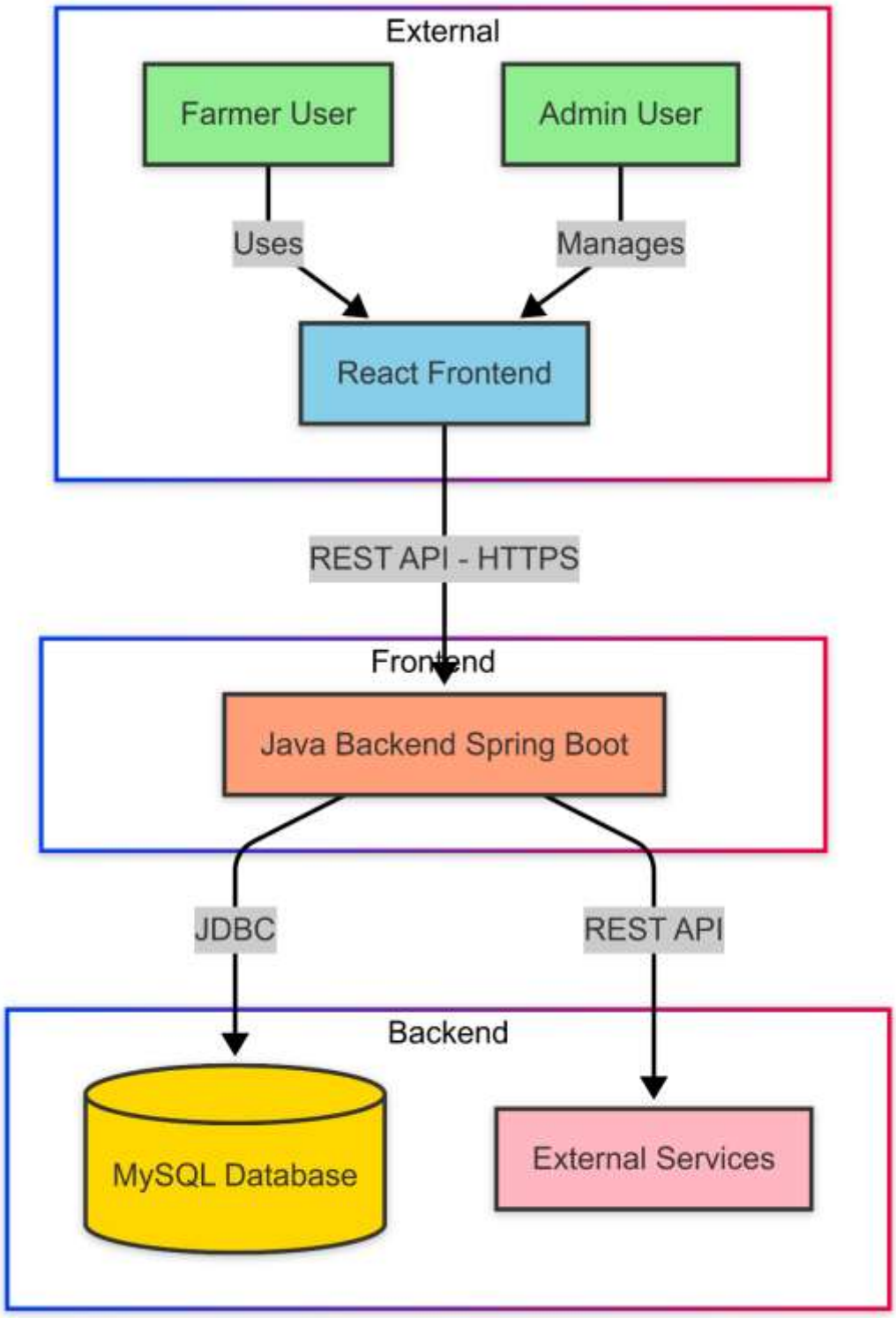
Meshack Ochieng' **SCT212-0078/2023**

GitHub link: https://github.com/d3q0w/Dairy_App

Data Flow Description



CONTEXT DIAGRAM



DESIGN APPROACH

We focused on an Object Oriented Design (OOD) methodology, focusing on modularity, encapsulation, and clear relationship between modules:

- First, we identified the relevant Objects, extracted core entities for the system requirements. Example: Farmer, User Authentication and Authentication Service among other key Objects.
- Secondly, we worked on Class Definition by defining classes with encapsulated properties.
- From the Data Flow Diagram (DTO) and the Context Diagram (CD), relationships between Objects were defined, from checking on dependencies, composition and inheritance. For example: Authentication Service and Farmer collaborate through interfaces.
- Then, we defined Methods of the system to enforce behavior and logic. Example: validating of credentials like passwords and usernames.
- Once the design was completed, we went over to Implement the system by coding the Backend with Java, MySQL for Database and React JS for the Frontend and API in use is RESTful.

To ensure seamless working of the application, regular debugging and testing were done to achieve the final outcome.

WORKFLOW OF THE APPLICATION

1.USER INTERACTION

The process starts with the farmer who is the admin and so initiates actions.

The actions involve logging in, creating booking requests, managing farm details, etc.

2.AUTHENTICATION

The farmer's interactions are routed through the Authentication component.

This component verifies the user's identity (likely using credentials stored in the MySQL database).

Successful authentication allows the user to proceed.

3.DATA TRANSFER OBJECTS(DTOs)

After authentication, data is packaged into Data Transfer Objects (DTOs). This is a practice used for transferring data between different layers of an application, ensuring only necessary data is exchanged and in a standardized format.

The DTOs are sent to the Bookings component.

4.BOOKINGS

The Bookings component is the central hub for handling booking-related operations. It receives the DTOs and processes them.

This component interacts with various repositories to manage bookings, farmer information, and room/facility availability.

5.JWT AUTH FILTER

Before the Bookings component processes the request, it goes through the JWT Auth Filter.

This filter verifies the user's authorization using a JSON Web Token (JWT), ensuring they have the necessary permissions to perform the requested actions.

6.REPOSITORIES

The Bookings component interacts with three main repositories:

- Farmer Repository: Manages farmer-related data (e.g., farmer details, farm information).
- Booking Repository: Handles booking-related data (e.g., booking requests, booking status).
- Room (Facility) Repository: Manages room/facility availability and information (e.g., storage capacity, room details).

7.ROOM (STORAGE) MANAGEMENT

The Room (Storage) management component is responsible for managing room or storage facilities. It interacts with the Room (Facility) Repository to handle storage capacity, availability, and other related functionalities.

This component is connected to the Bookings component, suggesting that bookings are tied to specific rooms or storage facilities.

8.FARMERS MANAGEMENT

The Farmers Management component handles farmer-related management tasks. It likely interacts with the Farmer Repository to manage farmer profiles, farm details, and other farmer-related data.

It is also connected to the Bookings component, indicating that farmer information is relevant to booking processes.

9.MySQL DATABASE

The MySQL database serves as the persistent storage for the application. It stores user credentials, farmer information, booking details, room/facility data, and other relevant information.

The Authentication component directly interacts with the database for user authentication.