Chapter 9

Implementation

**Introduction:**

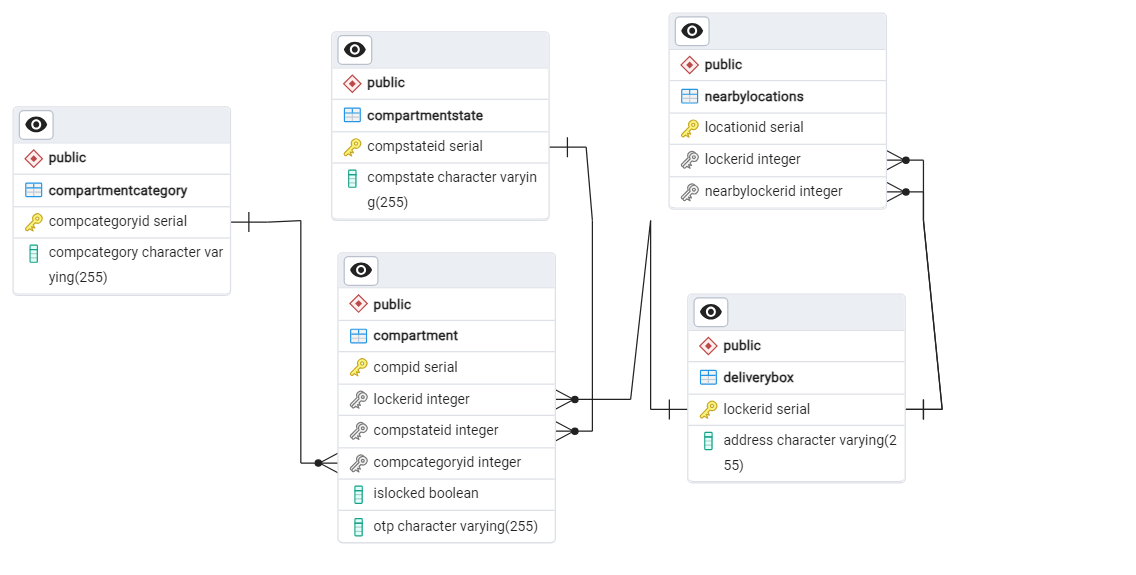
After thorough study of delivery process of multiple companies, we have analyzed all their requirements and found some common requirements that are crucial for last mile delivery process. Due to the dynamic nature of our project in which our requirements are based upon some dependencies that are meant to be resolved after the achievement of some initial milestones. So, first we need to achieve those milestones that start with the availability of Smart Locker (Smart Delivery Box).

**Construction of Smart Locker’s Emulator:**

After meetings with multiple sellers and manufacturers of smart lockers, the core functionality and their implementation has been studied through their provided documentation of smart locker.

1. **Smart Locker database**

According to existing parcel lockers, those who are based on the modern architecture have been selected to be followed. Database schema has all those entities that are followed by existing parcel lockers for state management. The design of schema has been made carefully by prioritizing the extendibility and integrability to high. The higher the extendibility, the more likely it will be for SPs in future to get extended as it will be integrated into SP’s existing system.



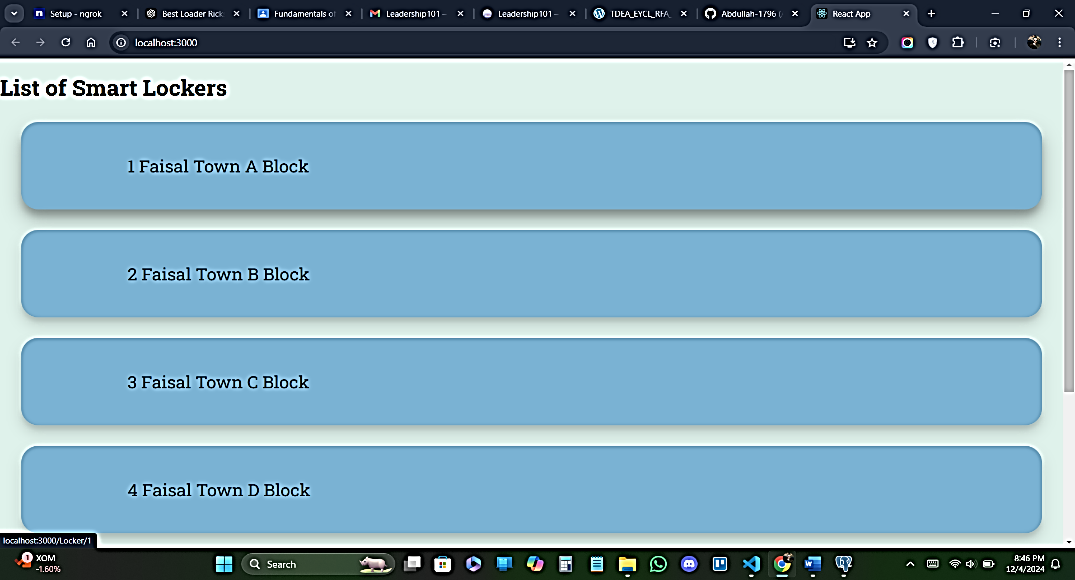
1. **Smart Locker Emulator**

Discussion is still going on with the client about purchasing of smart locker. So, to continue with development, an emulator of smart locker has been developed that will react same as the real smart locker.

* 1. **Smart Locker UI**

The front-end of smart locker has been developed using React.js. The user will be able to put and receive parcel (hypothetically) by using the screen of emulator.

The following image shows the list of installed lockers:



The following images show compartments of different categories (Small, Medium, Large) in a single delivery box.





Each compartment shows all of its states, the interaction screen is under development and is in final touching.

* 1. **Smart Locker Backend for State Management**

Backend functionality has been implemented in Node.js and Express.js. MERN stack has been incorporated for the development of Smart Locker.

1. **Smart Locker APIs**

APIs have been developed to manage all the states of smart locker. They are built in Express.js.

**Implementation of last mile delivery process:**

The last mile delivery process starts after scheduling of parcel’s delivery at last checkpoint/warehouse of SP. The rider takes the parcel from warehouse and delivers it to the receiver (process of usual delivery). Now with smart delivery, rider will place parcel in smart locker selected by the receiver and receiver will take it from the smart locker.

1. **OTP generation**

When the receiver selects the parcel locker, OTP will be generated for that specific compartment and will be sent to the rider. The state of compartment is changed to **Reserved**.

1. **Parcel check in & check out**

The rider uses the screen of locker to enter OTP, after verification the compartment associated with that OTP will be opened. After closure of compartment’s door, the state of parcel will be changed as well. The state of compartment is changed to **Acquire** if parcel placed in or **Empty** if taken out.

1. **Security Measures**

* **Receiver or rider leaves the compartment opened**

To tackle this problem, we are implementing a check that whether anyone is around the compartment or not. If not, it will enable the timer after which the compartment will get closed automatically.