**Problem Statement:**

To develop an application to Digitize the Railway Crossing Status for the Public. Government aims to communicate the status of a Railway Crossing to the public well in advance i.e., a railway crossing is open, closed etc.,

**Proposed Solution:**

James tech pvt. Ltd. proposes a Java based application that has:

1. Separate user menu for general public and admin users.
2. User authentication system for public and admins
3. Data import and export utility for backups.

Using the application, the admin users will be able to:

1. Add new railway crossing
2. Delete existing railway crossing
3. Search for railway crossing in data base
4. Update the status of the railway crossing
5. Export data to database
6. Import data from the database

Same application can be used by the general public to perform following actions:

1. Create new account
2. Login to existing account
3. Fetch the details of existing crossings
4. Check status of crossing at given point of time
5. Search for a specific railway crossing
6. Sort timings of given railway crossing

**Concepts to be used in the project**

1. Java – for coding the backend logic
2. Object Oriented Programming – to simulate real world entities like users, railway crossings, etc.
3. Various data structures like hash maps, Lists, etc. – to store data during runtime
4. Collections frameworks – to implement various data structures
5. File IO – to import and export data from text files
6. Unified Modelling Language (UML) – to visualize various components used in project.

**Project File Structure**

| Railway Crossing Status

|

\---src

\---com

\---amazon

\---atlas22

\---railwaycrossingapp

| GovernmentApp.java

| PublicApp.java

| RailwayCrossingApp.java

|

+---controller

| RailwayCrossingController.java

| UserController.java

|

+---db

| DAO.java

| DB.java

| DBService.java

|

\---model

RailwayCrossing.java

User.java

**UML: Class Diagram for the project**

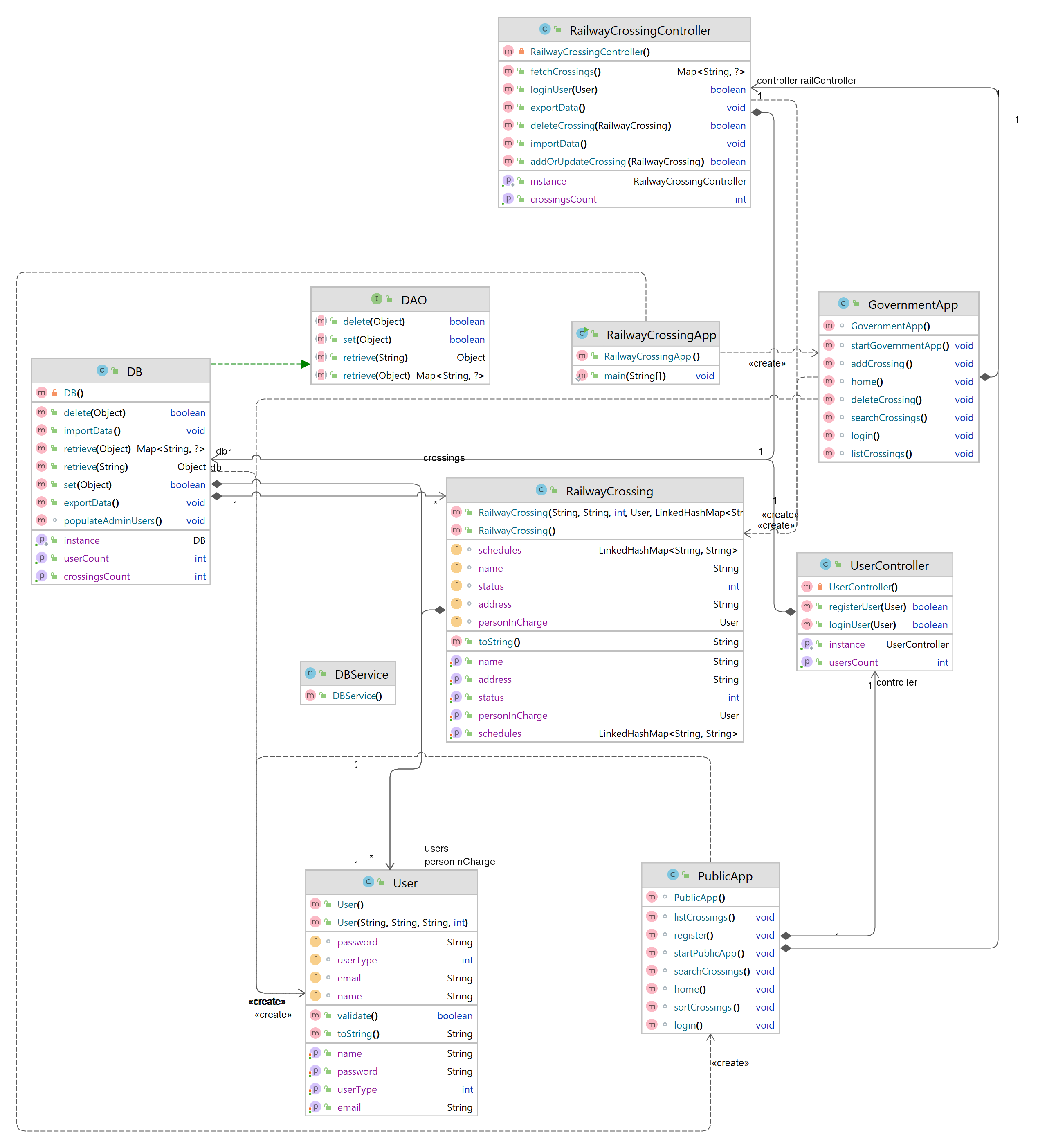
****

Figure 1: Class Diagram

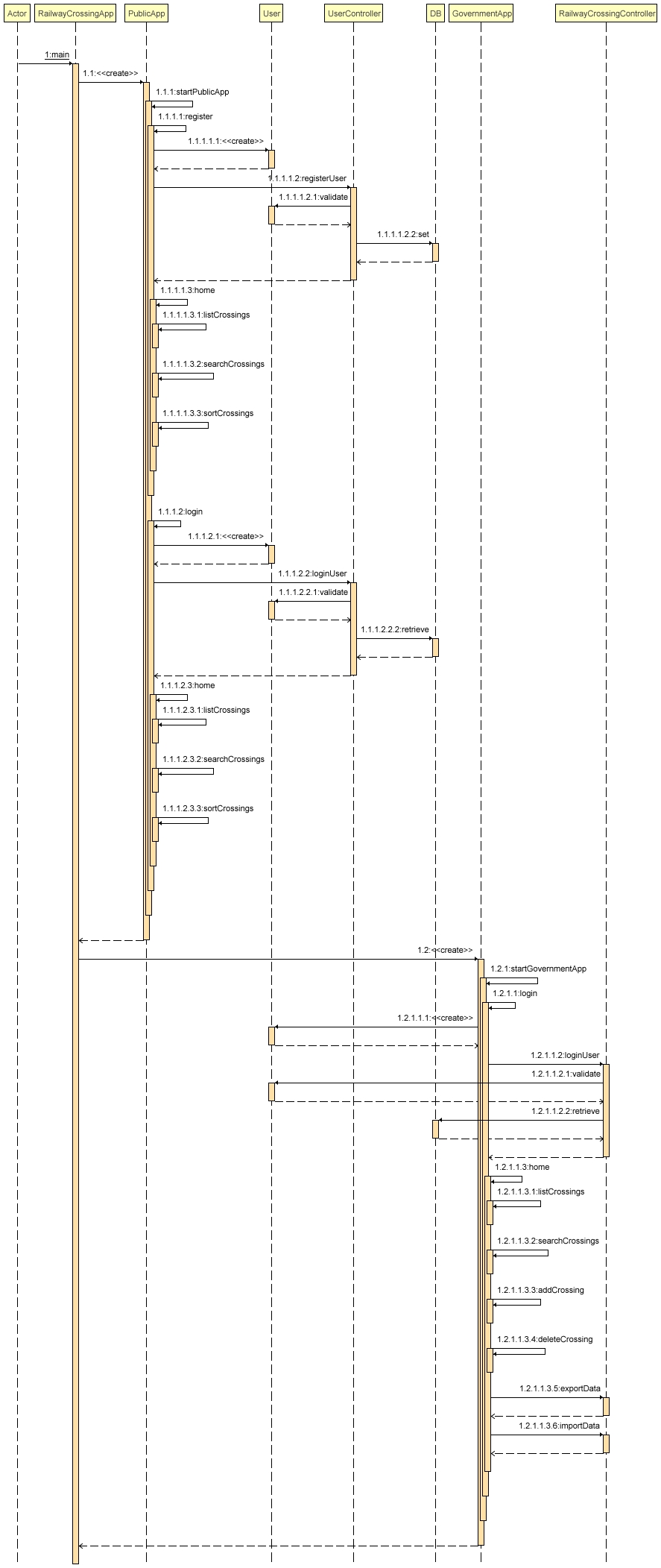
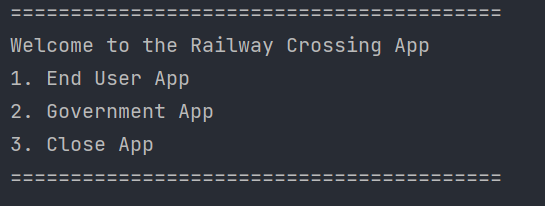
**UML: Sequence Diagram**

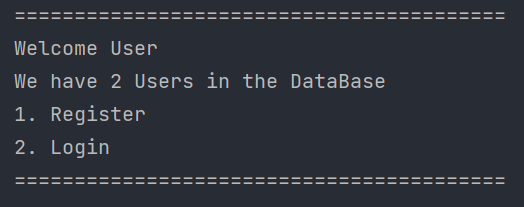
Figure 2: Sequence Diagram – RailwayCrossingApp.java

**User Manual – End User**

1. **Run *RailwayCrossingApp.java* to start the application.**

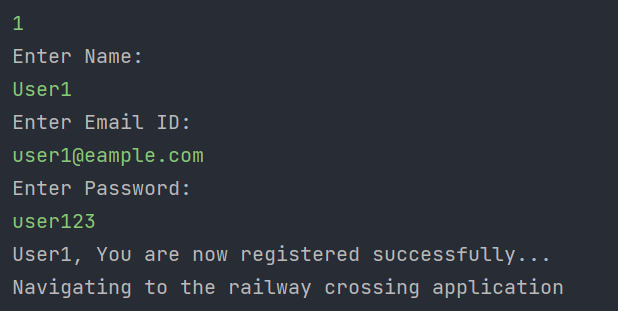
****

1. **Enter 1 to explore end user menu, 3 to close the program.**
2. **End User menu:**

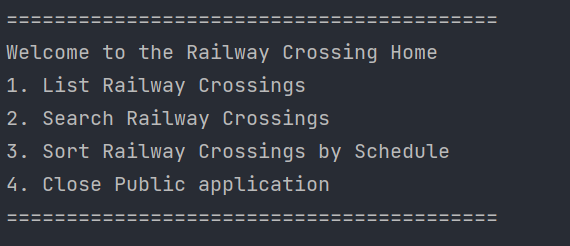
****

**Chose 1 to register new user, 2 to login to existing account.**

1. **Upon choosing 1, enter details as follows to create new account and explore additional options**

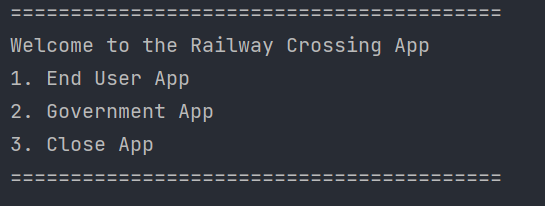
****

1. **Options for end user, chose appropriate option to begin.**

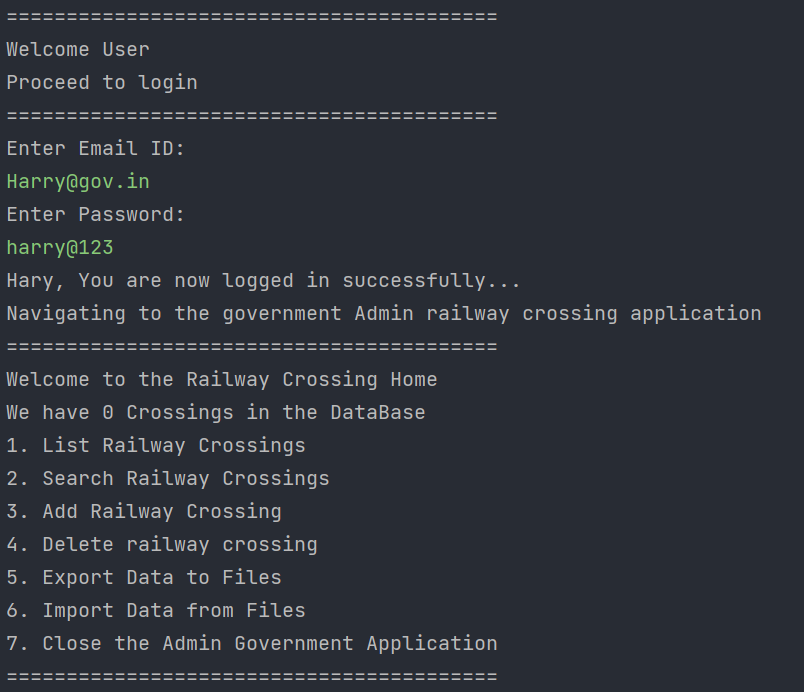
****

**User Manual – Government/Admin User**

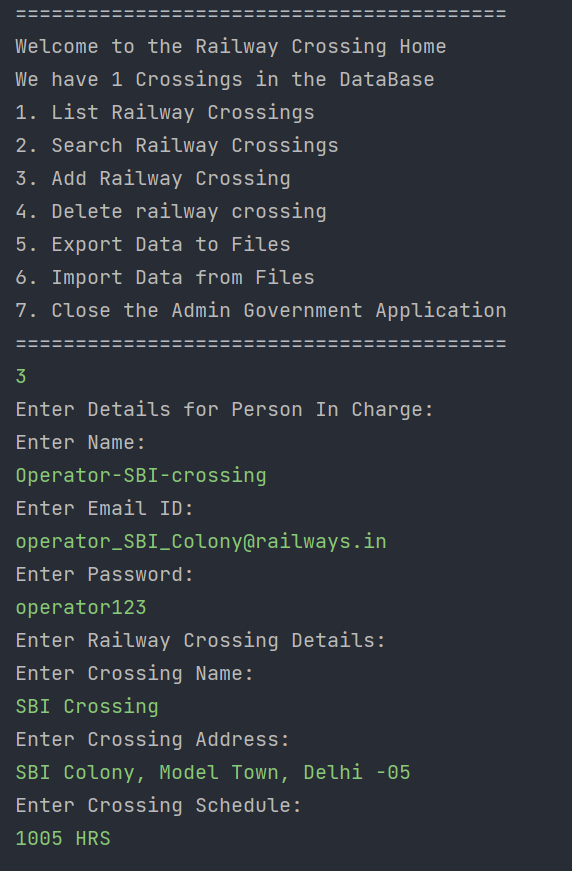
1. **Run *RailwayCrossingApp.java* to start the application.**

****

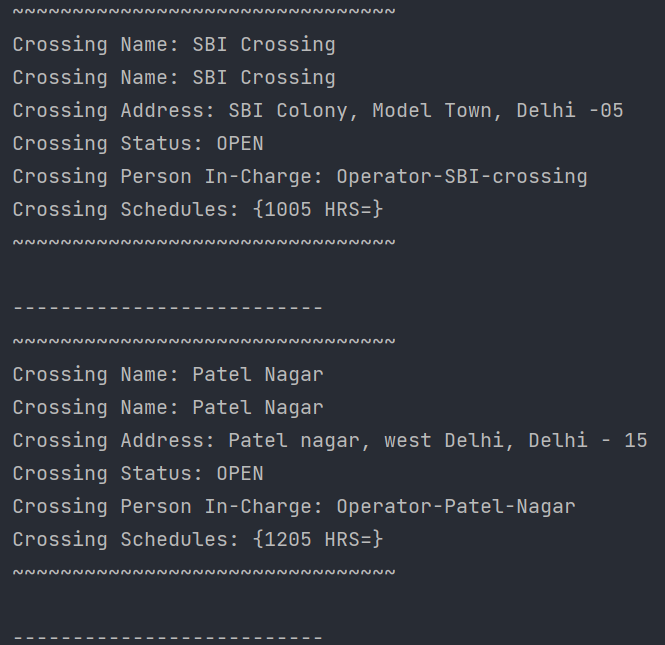
1. **Enter 2 to explore admin user menu, 3 to close the program.**
2. **Upon choosing 2, App will ask for Admin credentials. New admin accounts cannot be created from the app directly as specified in the document.**

****

1. **Select appropriate option to perform desired action.**
2. **Choose 7 to close.**
3. **Example: Adding a crossing:**

****

1. **Example: Listing all crossings**

****