SINGAPORE POLYTECHNIC
SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

(ET0706) Object Oriented Programming

Project Title: Seamless Logistic Transport

Student Name: Abirami Baskaran
Admission No: P2241801

Class: DCPE/FT/3A/02

Contents

Project Overview

This Project endeavors to develop a JavaFX Application to tailored to the needs of a logistics transportation company, mainly focused on land transportation companies.
The application establishes a user-friendly interface for administrative work and simple tasks related to their role in the company.
By incorporating a robust database system, data and information is seamlessly stored and facilitates execution of CRUD (Create Read Update Delete) operations.
Additionally, through the incorporation of an API, the program furnishes comprehensive travel details such as distance, duration, fuel consumption and fuel combustion.

System Requirements

The application is set up in IntelliJ IDEA 2022.3.2 (Community Edition).
It preferably requires:

- OpenJDK version 19.0.2
- JFoenix version 8.0.10
- MongoDB version 4.10.0 (Also requires connection to authorized wifi)
- JSON version 20210307
- API key from Google Cloud Services (Place Auto Fill & Distance Matrix)

Key Features

Streamlined Parcel Management System

This application provides a user-friendly interface for coordinators to effortlessly input parcel details and assign tasks to haulers. This efficient process ensures smooth logistics management within the delivery company.

Hassle-Free Employee Management

Administrators can conveniently add new employees to the system, facilitating seamless onboarding and maintaining an up-to-date employee database.

Integrated Database Functionality

Incorporates a robust database system, ensuring secure storage and retrieval of critical information related to parcels, employees, and task assignments.

Integration with Google API

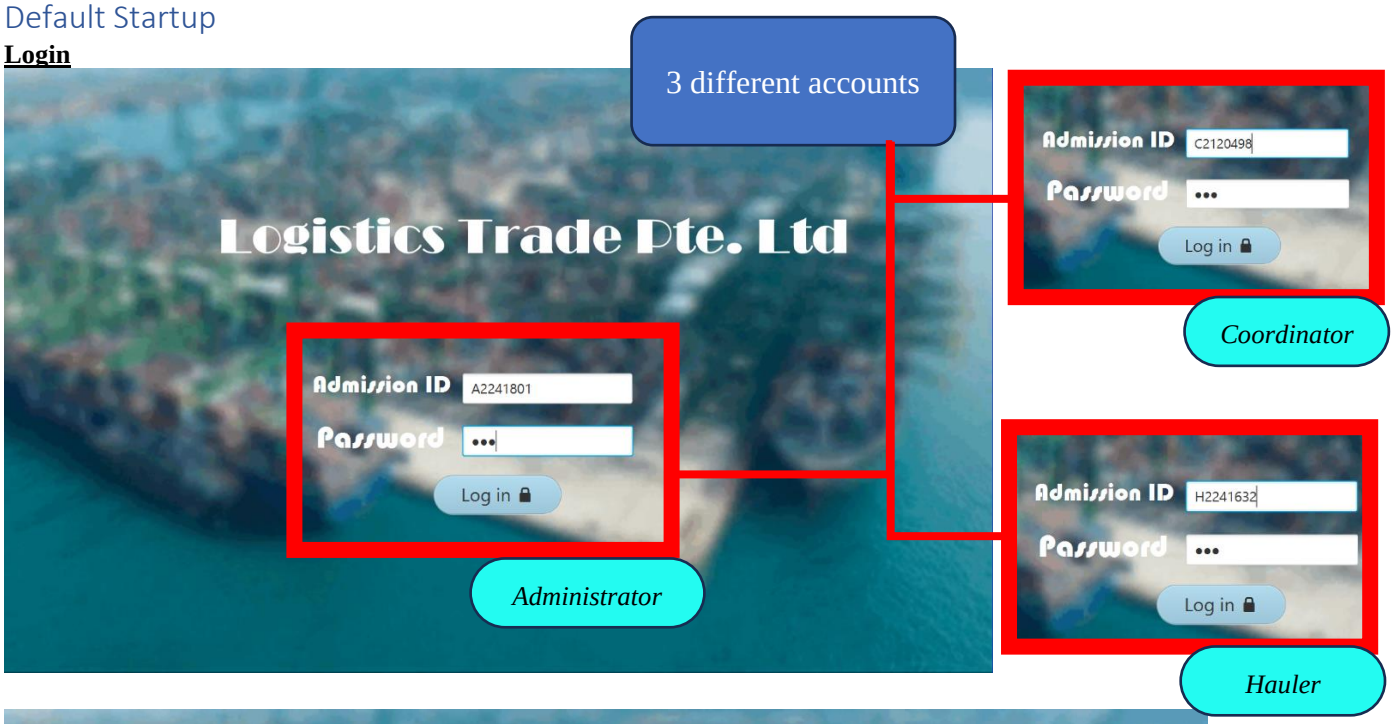
By utilizing a Google API key, the application enables automated address autofill and provides accurate duration calculations for specific locations enhancing efficiency and accuracy in delivery planning.

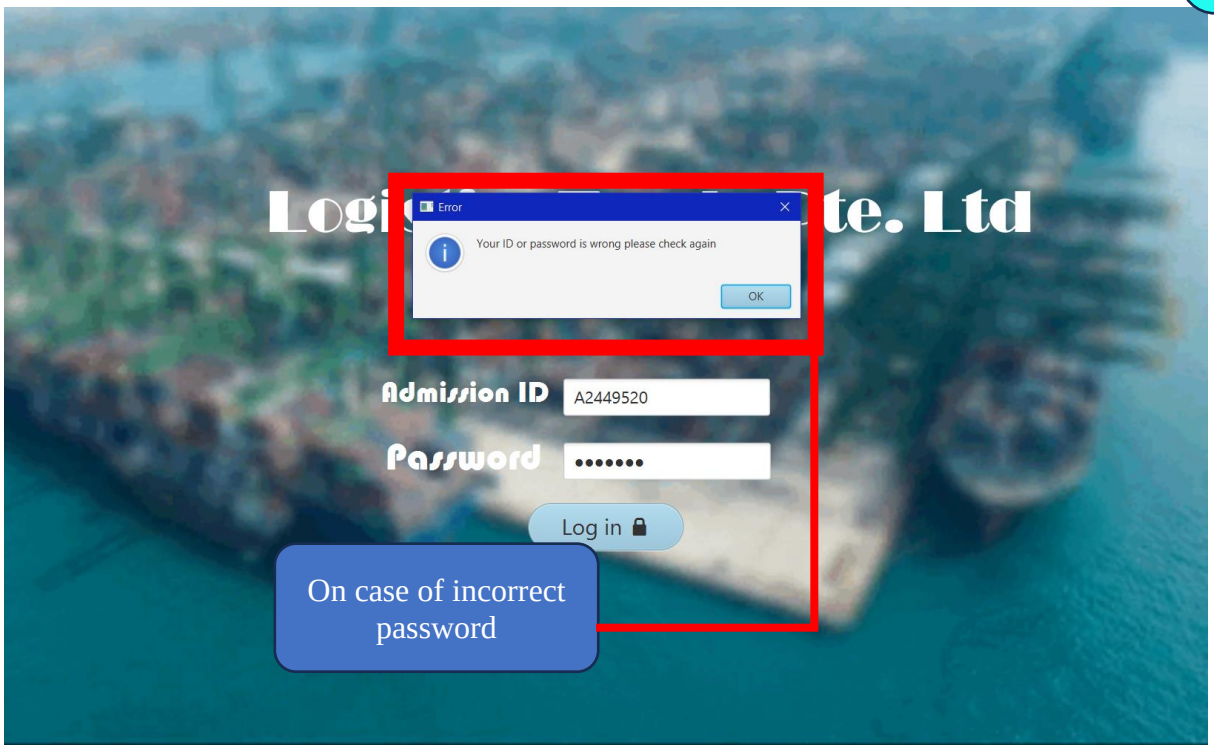
Together, these compelling features make the JavaFX application an indispensable tool for the logistic delivery company, streamlining operations, enhancing communication, and optimizing overall efficiency.

Front-End User Interface Design

Default Startup

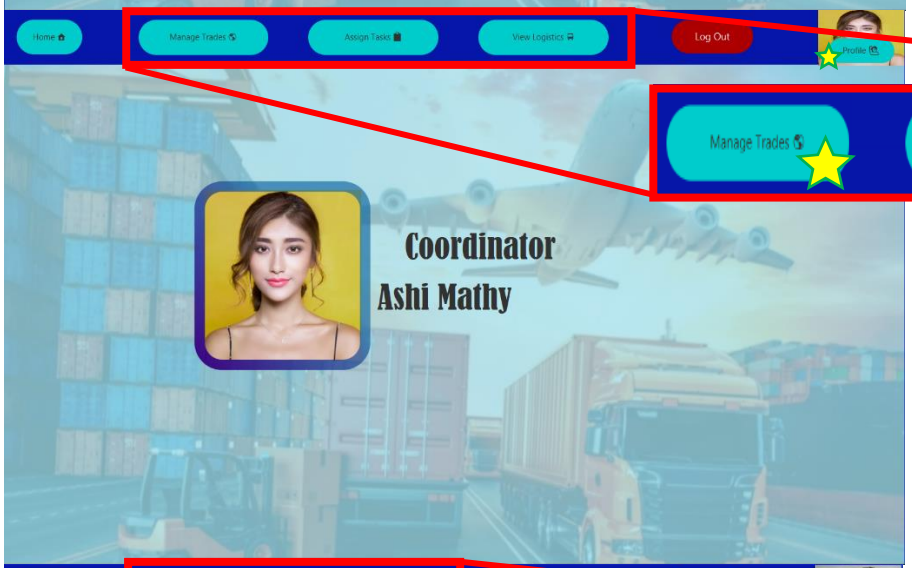
Login



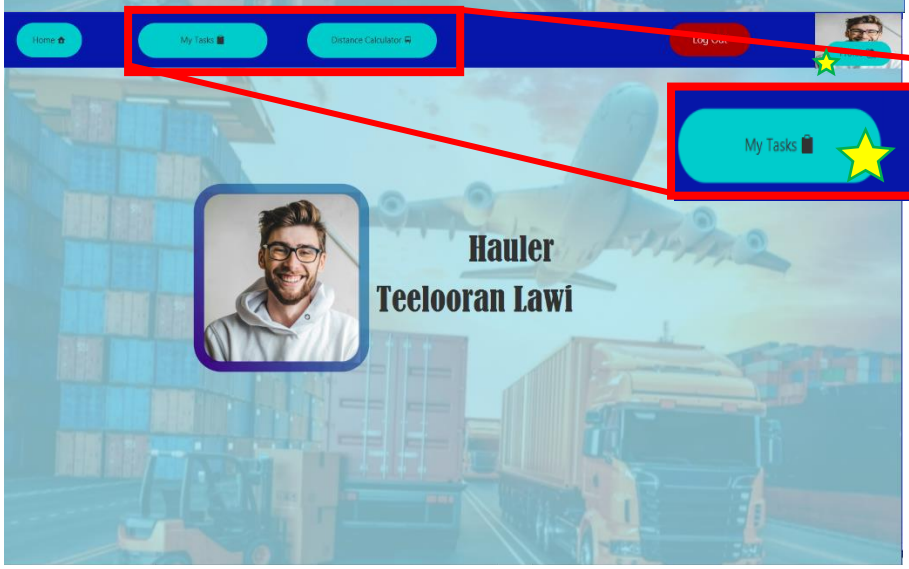


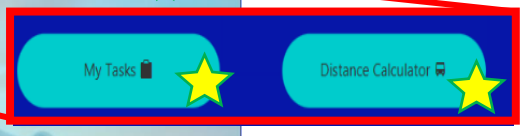
Home

Please click on the stars to navigate to respective pages



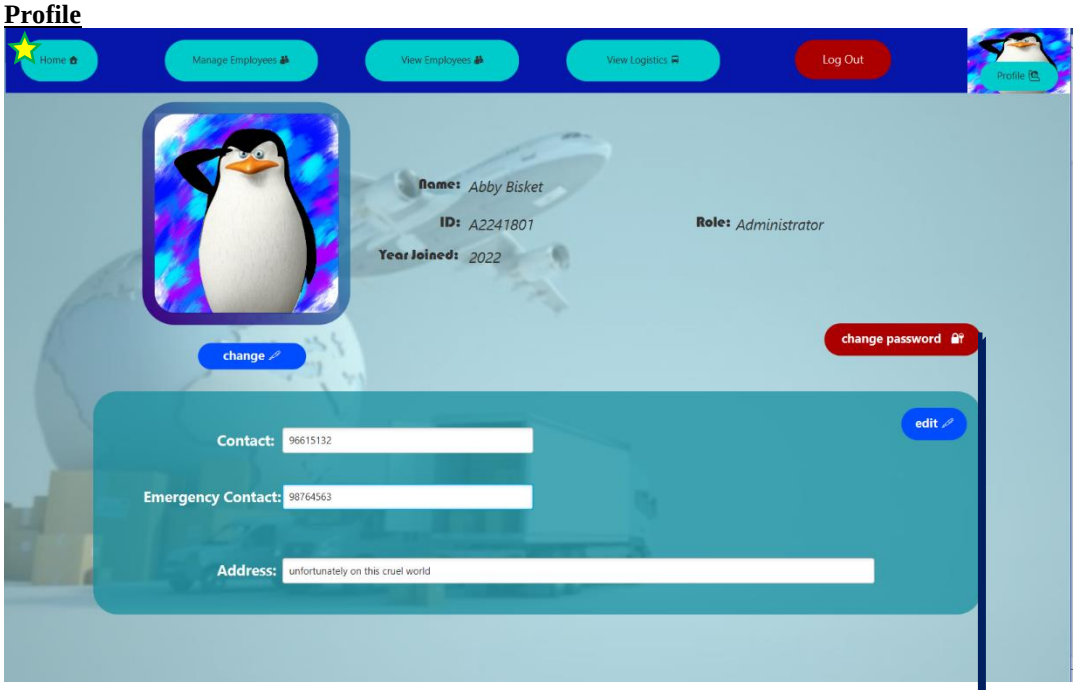
Please click on the stars to navigate to respective pages



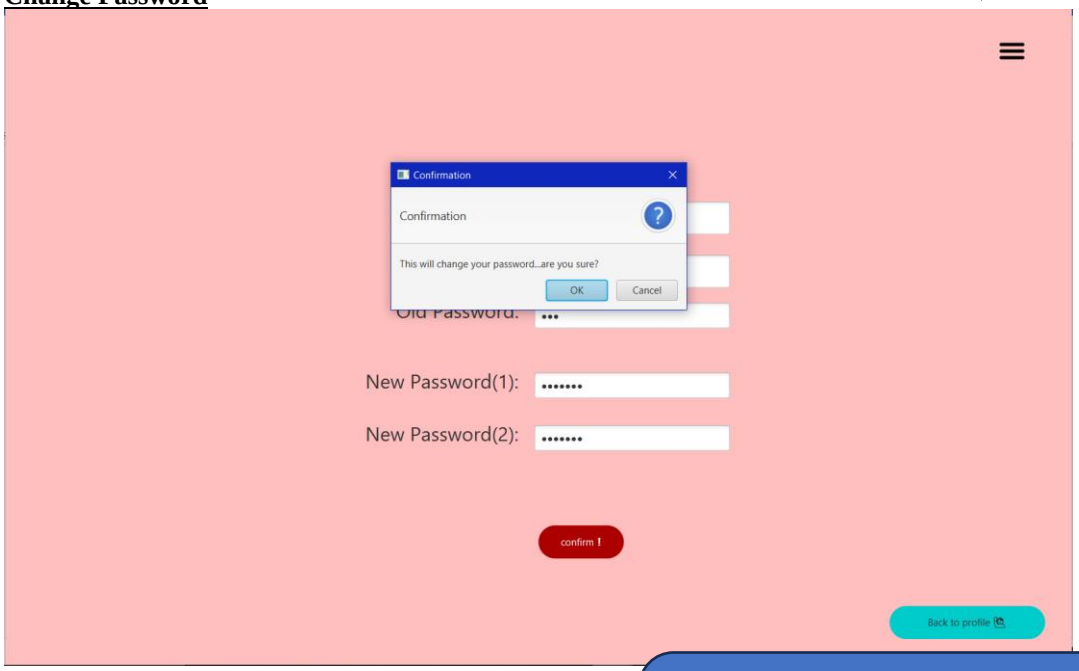


Please click on the stars to navigate to respective pages

Profile



Change Password



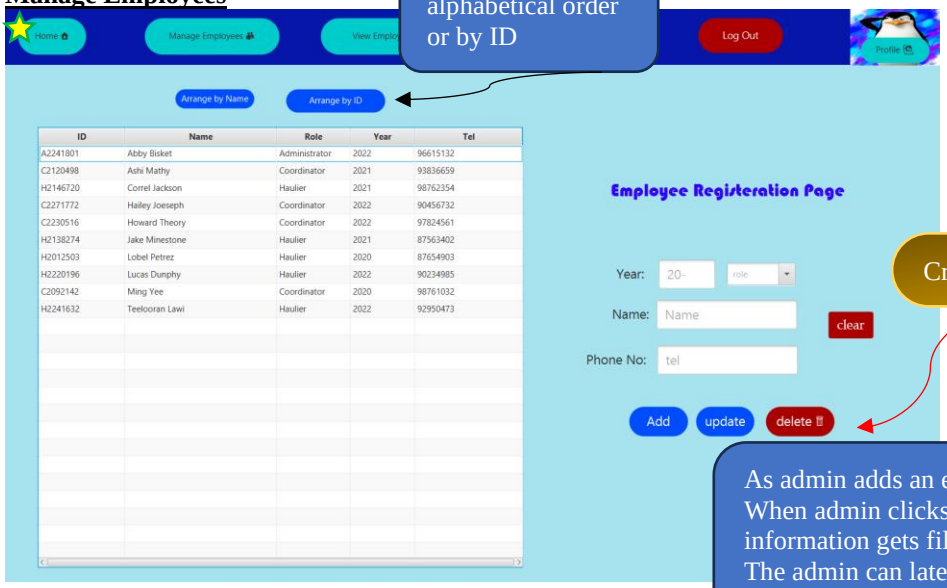
Before allowing user to enter password…

- there'll be credentials check,
- ensures password is correctly typed
- asks user for confirmation before officially changing

Administrator Site

Manage Employees

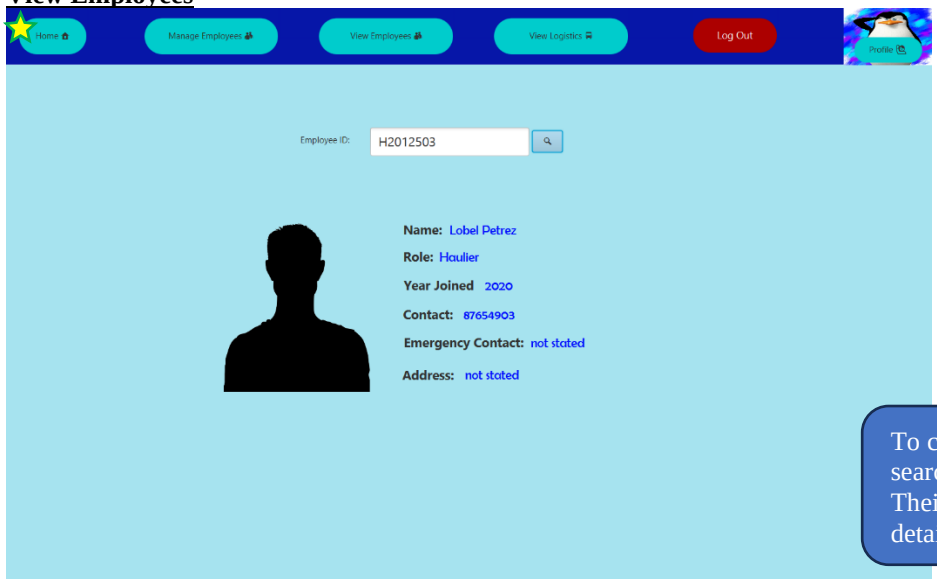
To view the list in alphabetical order or by ID



Create Read Update Delete (CRUD)

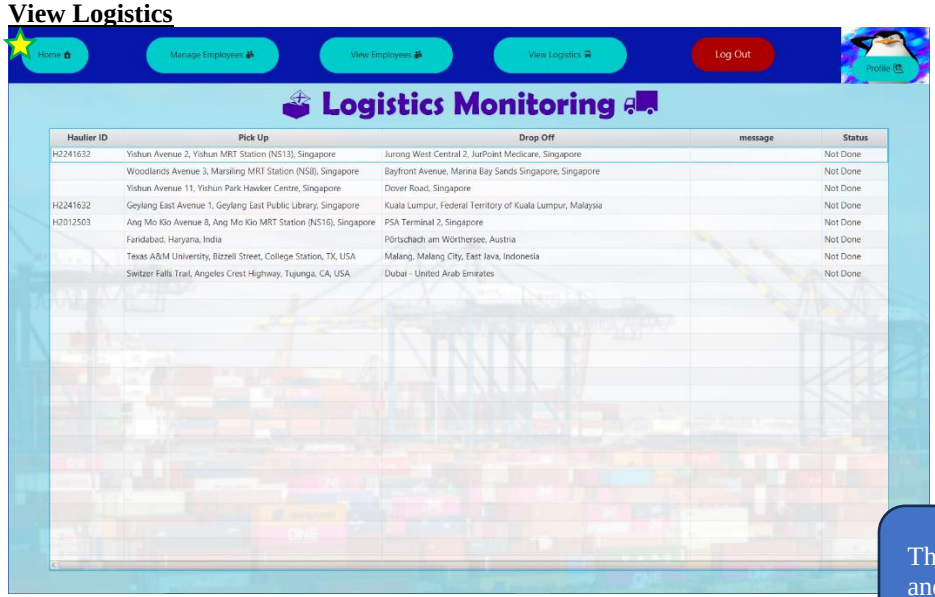
As admin adds an employee, it gets updated in the list. When admin clicks on an employee in the list, the information gets filled in the text fields.
The admin can later update information or delete information

View Employees



To check for an employee, admin can search for them through their ID. Their profile picture and personal details will be displayed.

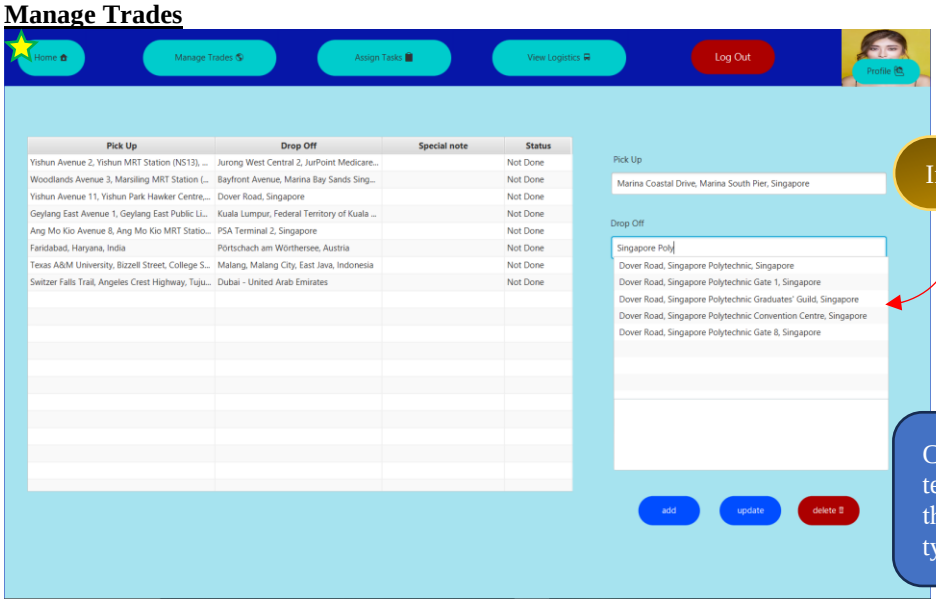
View Logistics



This page is displayed to both Admin and Coordinators. This is an overview of all logistics.

Coordinator Site

Manage Trades



Implementation of API

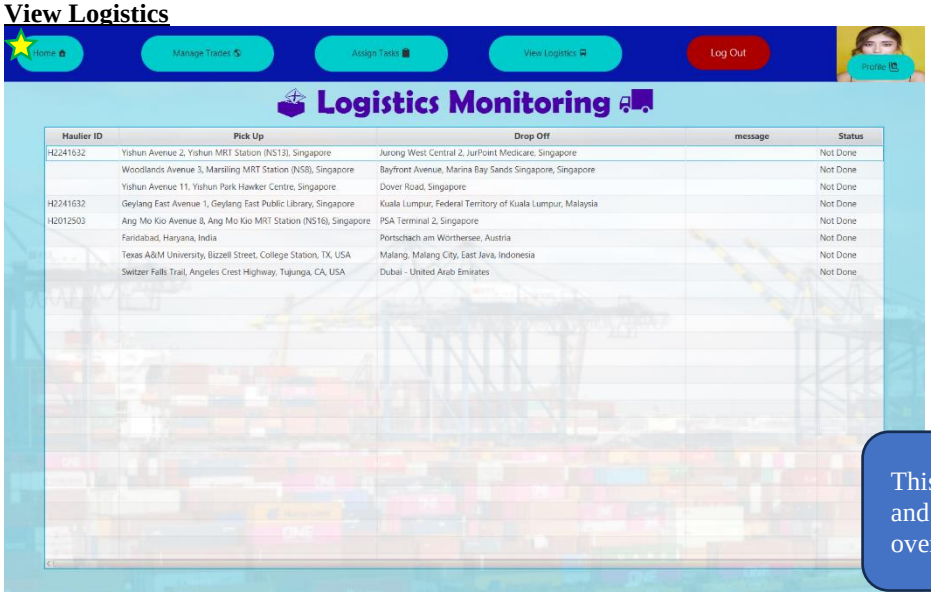
CRUD operation is used here. The text fields have auto fill features so the user doesn't have to manually type all the locations

Assign Tasks

When the coordinator clicks on a trade on the right, it fills up the label as shown.
The user can later select a hauler *(in the dropdown list)* to take on the task.

If the user wants to remove any, or update any, they are able to do so. This action will update both tasks accordingly

View Logistics



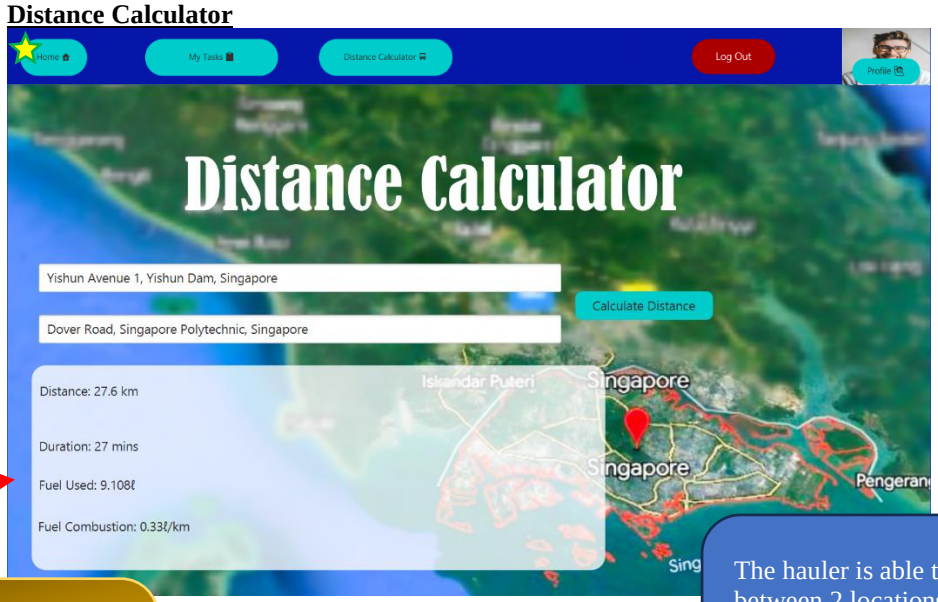
This page is displayed to both Admin and Coordinators. This is an overview of all logistics.

Hauler Site

My Tasks

The hauler can view their tasks for the day. The are able to set the status of their task to let the coordinators updated of their progress.

Distance Calculator



Implementation of API

The hauler is able to check the distance and duration between 2 locations.

There is calculations done to provide approximate fuel used and fuel combustion for the respective trip

Back-End

The application has 2 super classes, 2 domain classes and 12 controller classes.

The super classes are:

1) Application
   It incorporates functions to create a new pages, convert base64 strings to image formats, and pop open alert boxes. This class is integrated so that the features mentioned could be executed in a single line, enhancing code flow for the programmer.
2) ConnectionToMongoDB
   This class is integrated so that connection to database is only needed to be done once. The class also has functions such as to read from specific column, add data and update data in certain collections. These functions are added so that executing these features can be done in a single line in the controller classes.

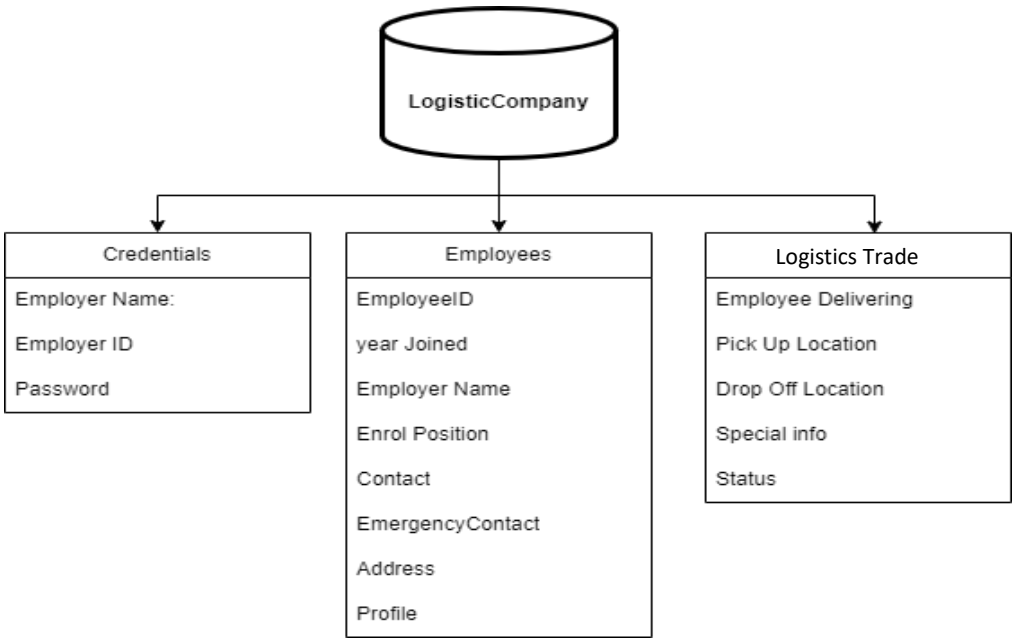
The domain classes are:

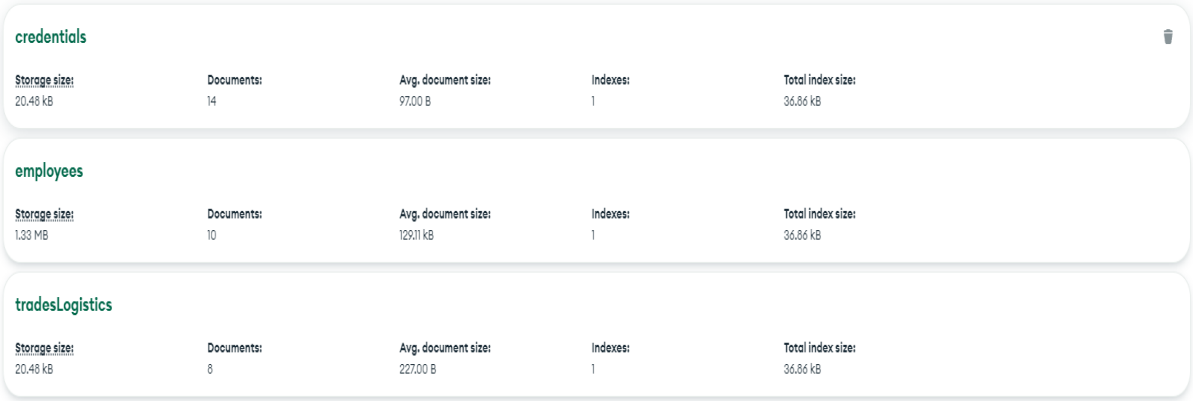
1) Employees
   This domain class is accessed by 'EmployeesController' where an observable list of instances of this class is created. This domain class encapsulates variables that store employee information, facilitating not only additions on but also updates or deletions.
2) Trades
   This domain class is accessed by 'Trades Contoller', 'AssignTasksController' and 'ViewTradesOnlyController' and '*HaulerViewTasks*'. This domain class encapsulates information of logistic trades such as the '*haulerID*', '*pickUp*', '*dropOff*', '*message*' and lastly '*status*'. The 4 controller classes create an observable list of instances of this domain class.

The controller classes are:

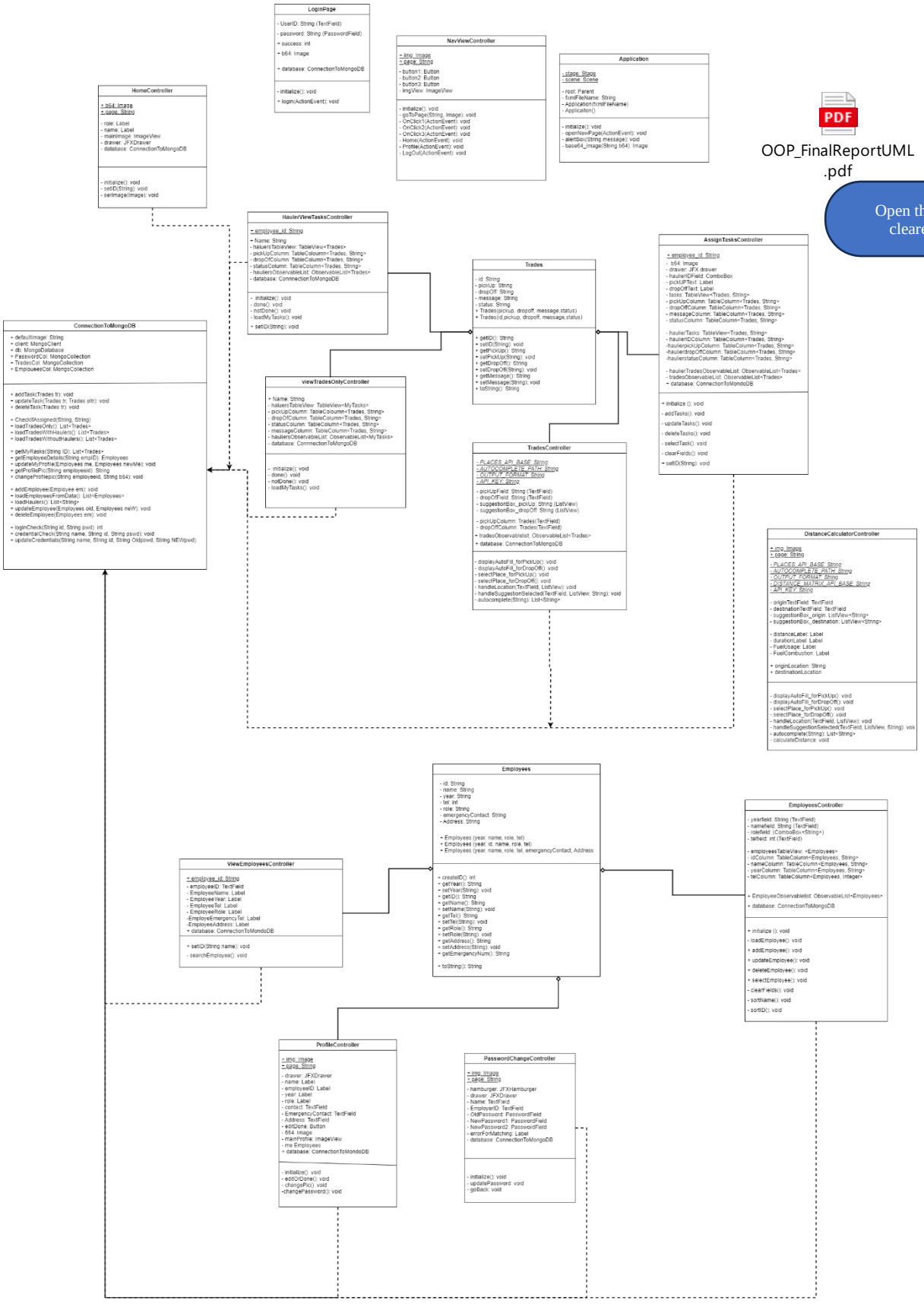
1) EmployeesController
2) TradesController
3) AssignTasksController
4) ViewTradesOnlyController
5) HaulerViewTasks
6) DistanceCalculatorController
   This controller accesses the place auto-fill API to ease typing in the text field by auto completing user sentences and the distance matric API to provide distance and duration details.
7) LoginPageController
   This class accesses the database to ensure if the user id and password typed by the user is correct to allow access to assigned page.
8) NavViewController
   This class arranges the navigation bar according to the user. Options displayed and actions taken varies depending on the user. Hence this controller takes charge of these features.
9) ProfileController
   This class access the database to display personal information of the user. These information could also be edited.

Database architecture





Object Class Design



OOP_FinalReportUML.pdf

Open the pdf for clearer view

Conclusion

I was pleased to realize I managed to create all the features I mentioned in the concept paper and was also able to add additional features on the final project. Embarking on this project has been a profound learning experience that vividly demonstrated Object-Oriented Programming (OOP) concepts. By integrating a database and effectively implementing CRUD operations, I gained valuable insights into data management and persistence. Adding and understanding the distance matric API and the Place auto-fill API not only showcased the power of API utilization but also enriched my experience. Creating this application honed my skills in creating a well organized and maintainable code. Overall, this endeavor underscored the significance of structured programming, efficient data handling, and strategic API integration, enhancing my proficiency as a developer and problem solver.

Resource

- Flowchart and Object Class Diagram
  - https://app.diagrams.net/
  - https://www.youtube.com/watch?v=UI6lqHOVHic
- MongoDB database
  - https://www.baeldung.com/java-mongodb
  - https://www.tutorialspoint.com/mongodb/mongodb_java.htm
- Google Distance Matrix API & Google Place Auto Fill API
  - https://developers.google.com/maps/documentation/places/android-sdk/autocomplete-tutorial
  - https://console.cloud.google.com
  - https://youtu.be/dFhKVJcmADo
- Basics of JavaFX
  - https://youtu.be/9XJicRt_FaI
- CSS
  - https://youtu.be/o-lAsVuskKI