Project Report

For

Deliverable 2

iGo

Submitted By: TEAM R

Anant Bir Singh
Prabhjot Singh
Piyush Singla
Parth Sonani
Shivam Dipak Soni

Submitted to

Prof. Pankaj Kamthan

SOEN 6461 - Software Design Methodologies

Concordia University, Montreal, QC

Contents

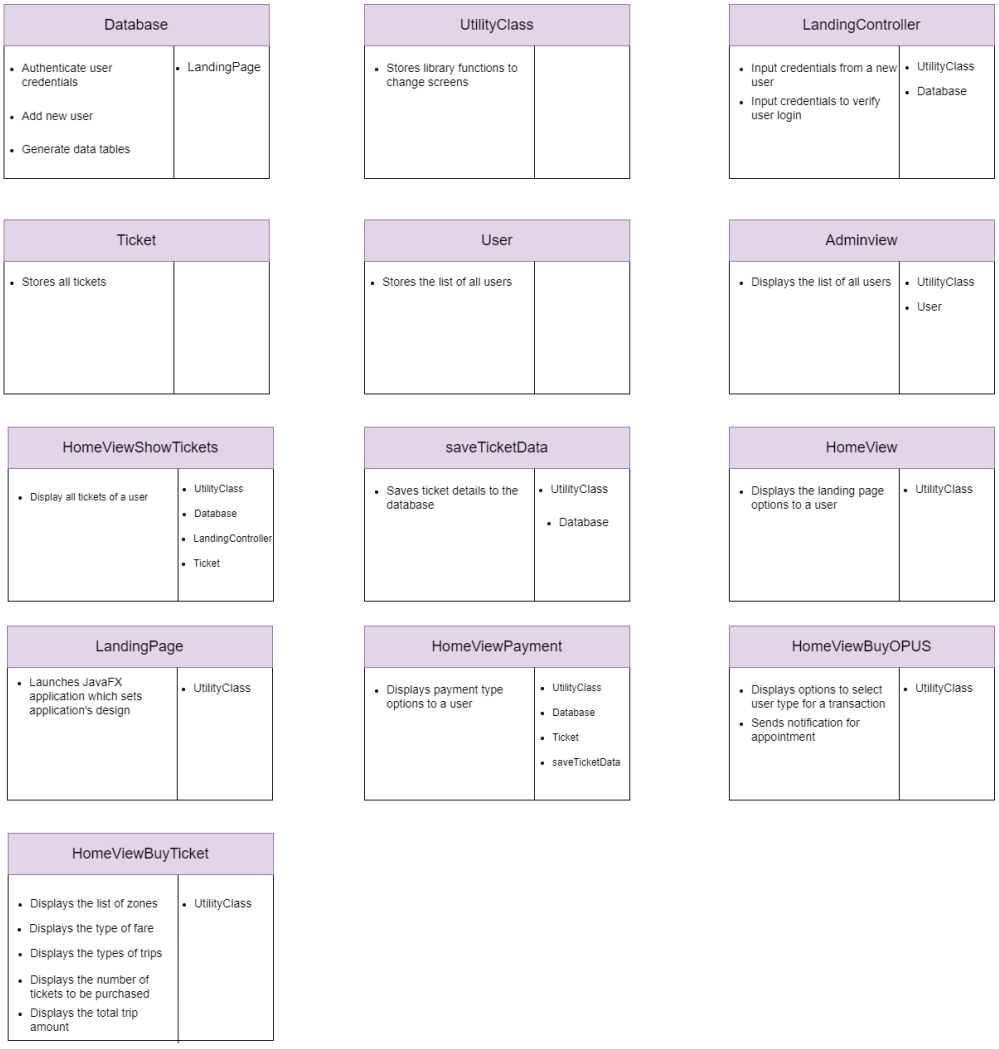
1 High level solution domain model

1.1 CRC Model for iGo

A Class Responsibility Collaborator (CRC) model is a collection of standard index cards that have been divided into three sections, as

- A class represents a collection of similar objects.
- a responsibility is something that a class knows or does.
- a collaborator is another class that a class interacts with to fulfill its responsibilities

Below are the CRC cards for the classes which relate to iGo.



CRC Card Model of iGo

2 Low level solution domain model

2.1 Structural Design Model

Structural design model refers to a type of model that represents the static structure of a system, including the components, their relationships, and the constraints between them. It is used to specify the architecture of a system and to ensure that the components are properly integrated and interact with each other as intended.

UML class diagram is a type of structural design model that represents the static structure of a system in terms of its classes, their attributes, methods, and relationships. Class diagrams are commonly used in object-oriented programming to represent the classes that make up a system and their relationships to each other.

Below given diagram is the UML class diagram of iGo which consists of all the classes and the relations between them.

UML class diagram of iGo

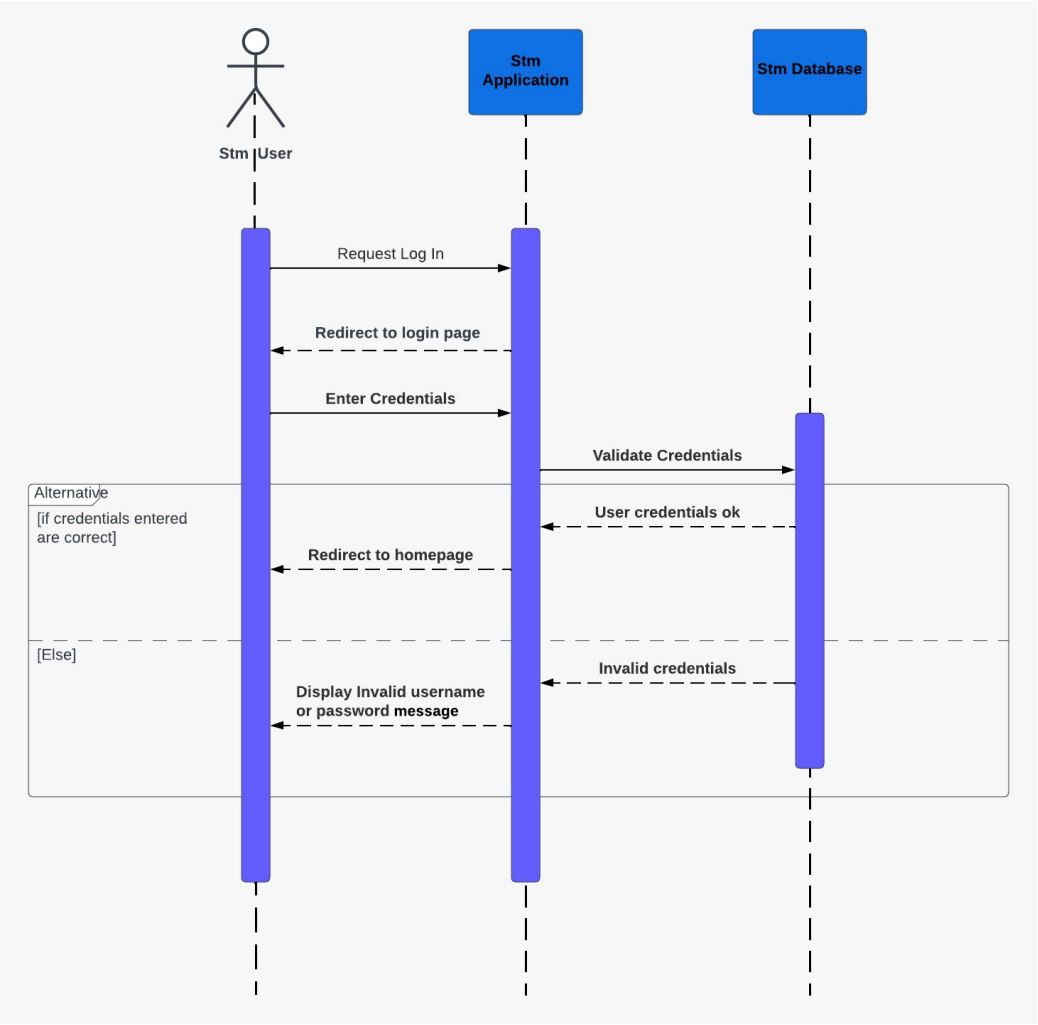
2.2 Behavioral Design Model

Behavioral design model refers to a type of model that represents the dynamic behavior of a system, including the interactions between its components and the flow of data and control between them. It is used to specify the behavior of a system and to ensure that it meets the desired functional requirements.

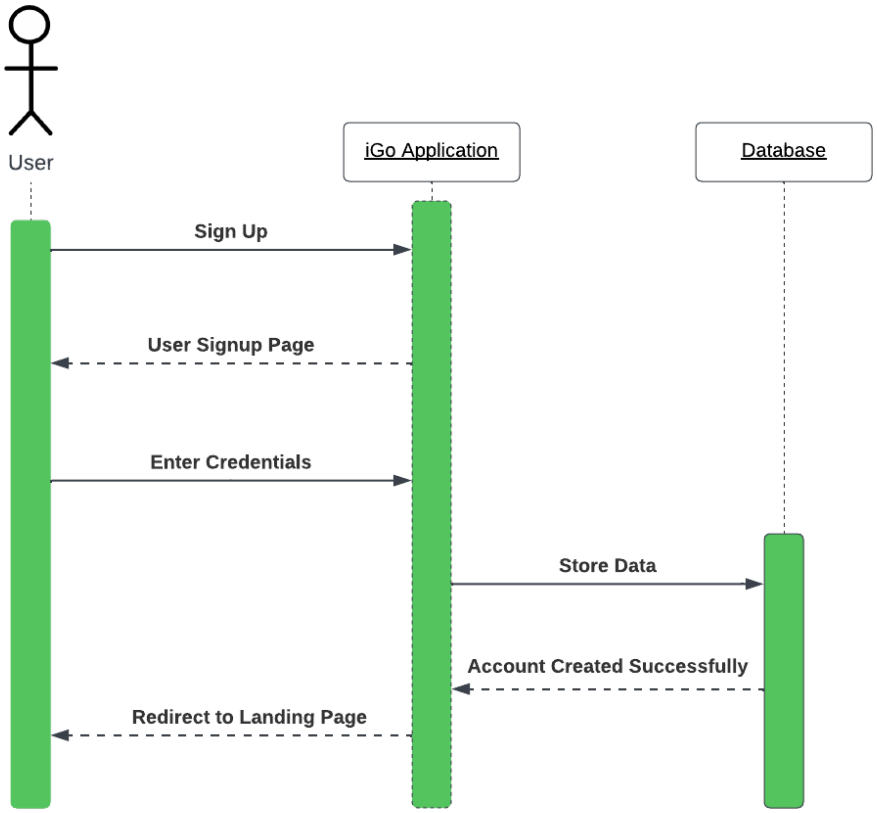
Sequence diagrams are a type of behavioral design model that represent the interactions between the objects or components of a system over time.

They show the sequence of events that occur in a particular scenario or use case, including the messages that are passed between the objects and the order in which they are processed.

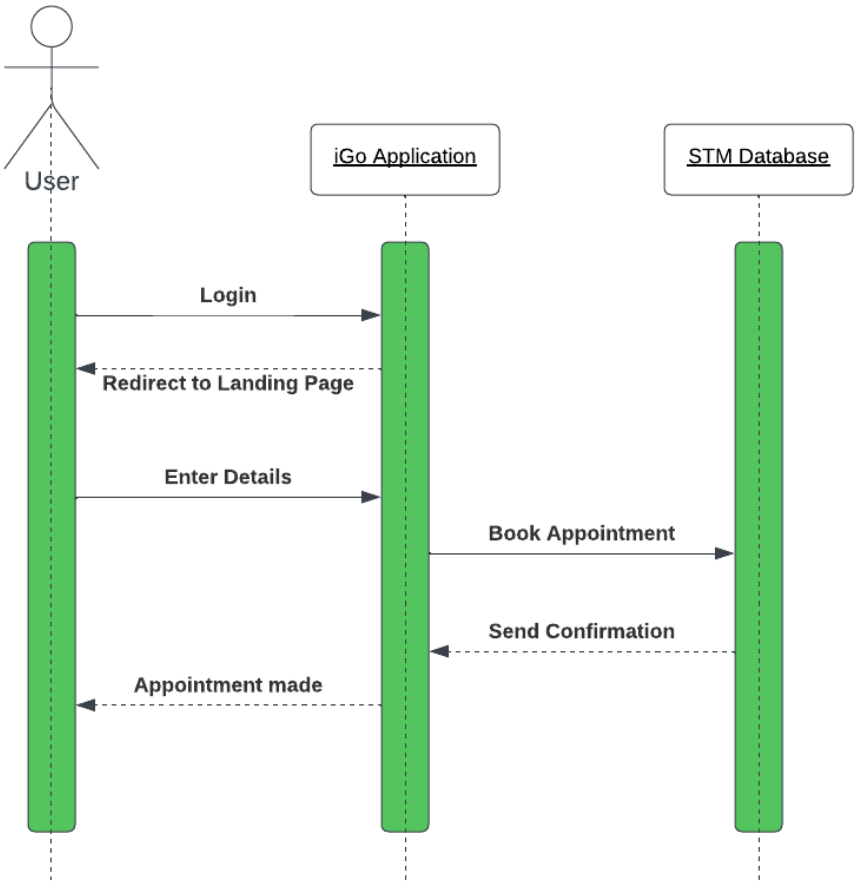
The following sections describes the 5 sequence diagrams which are the processes followed in the iGo.



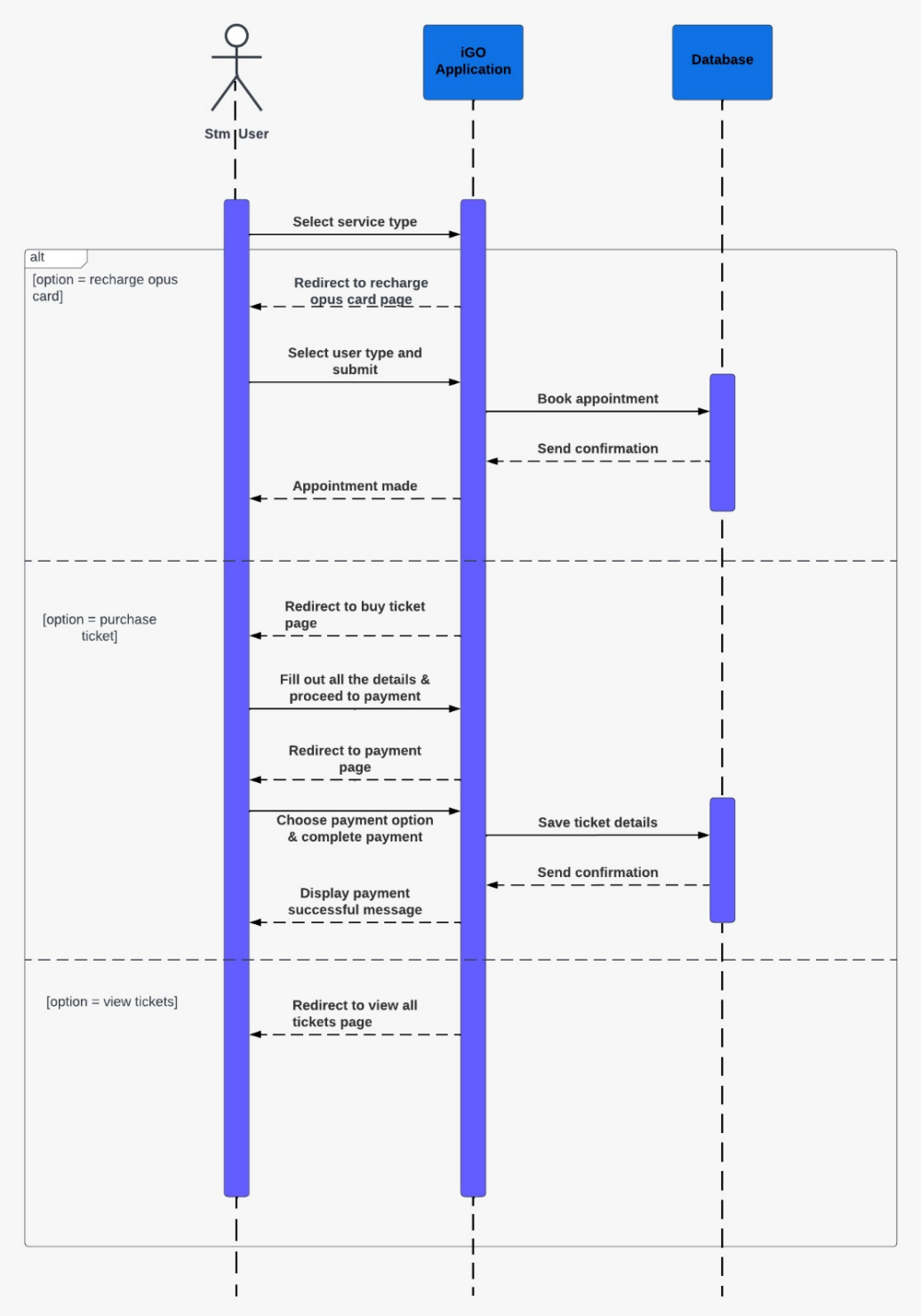
1. UML sequence diagram for login



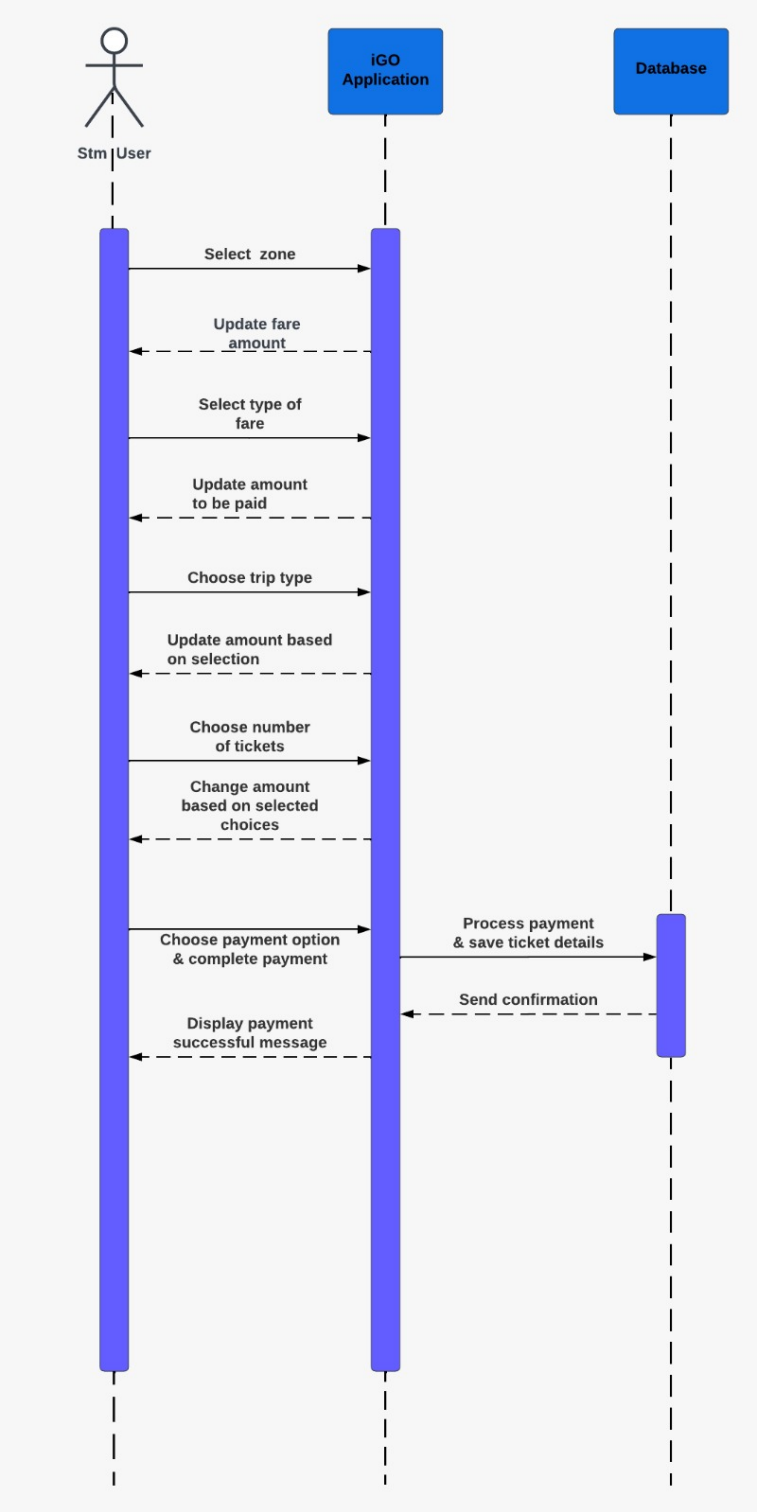
2. UML sequence diagram for signup



3. UML sequence diagram to buy new OPUS card



4. UML sequence diagram for buying new opus, selecting ticket and viewing ticket



5. UML sequence diagram for selecting the trip and payment

3 iGo - Javafx code

3.1 GitHub repository link : iGo

3.2 Programming Style and GUI

The GUI application for the iGo is developed using the java programming language. The javafx sdk is used for developing this application.

The java program uses Procedural programming style with object oriented programming practice.

3.3 Program flow

The Program flow section of this report provides a brief description of the flow of control within the software system. The sequence of the program is as follow:

- The program first launches the login page.
- If the user is registered then the user can login to the application, otherwise the user can proceed to signup page.
- The user can add all the necessary details such as "First name", "Surname", "Email", "Password", "OPUS serial number"(optional), "Date of Birth" and successfully signup.
- The email is considered as the username with which the user can log in the application
- After signing up or logging in the application the user is taken to the home page where the user has 3 main options
  1. Buy new opus
     - Clicking on this option leads to the screen where the user can book appointment for buying the discounted opus card.
     - After selecting the appropriate option and clicking on the book appointment the user is prompted with a message about an email being sent to user with appointment details
  2. Buy ticket
     - Clicking on this option leads to screen where the user is given option to select the zone for the ticket (ex. A, A B,...), select fare type (ex. Regular, 65 and over,...), select trip type (ex. 1 trip, 2 trips,...), and number of tickets.
     - After the user has successfully selected the options the user can click on the proceed payment button.

- On clicking the proceed payment the user is shown option to select the payment method and payment is done.

3. View my tickets
   - On clicking this option all the purchased tickets of the logged in user can be seen by the user.

- There is one more option "Admin Panel" which shows how many users are registered on the application.

3.4 Assumptions or limitations in program

There are certain assumptions in the program which are as follow

- The database is a local database, the user cannot login remotely.
- The program cannot validate the OPUS.
- The data for the zone, fare, trip are static (if stm changes data for this we need to also update it in files)
- The payment method is assumed to be a 3rd party.

4 Testing iGo

The following section will demonstrate the working of the application iGo along with it's functionalities through the use of images.

4.1 Registering the user

1. The landing page of the application

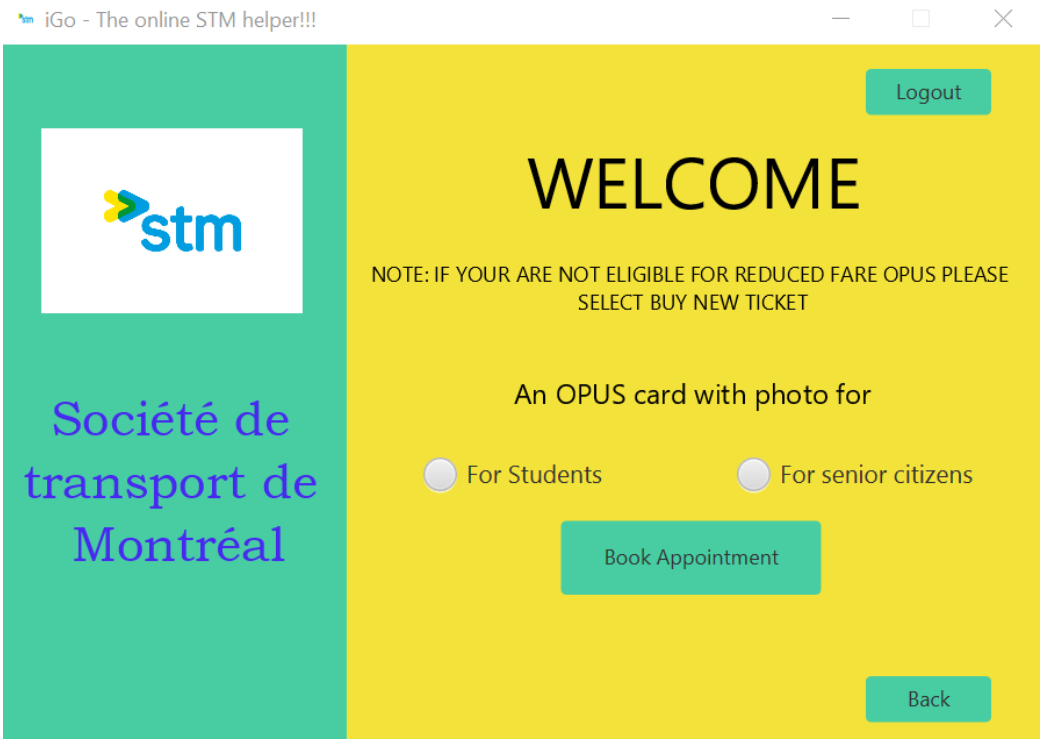
2. Signup page of the application

3. User filling up the information for signing on

4.2 User home

1. Homepage for user after signing in or logging in application

4.2.1 Buy new OPUS card

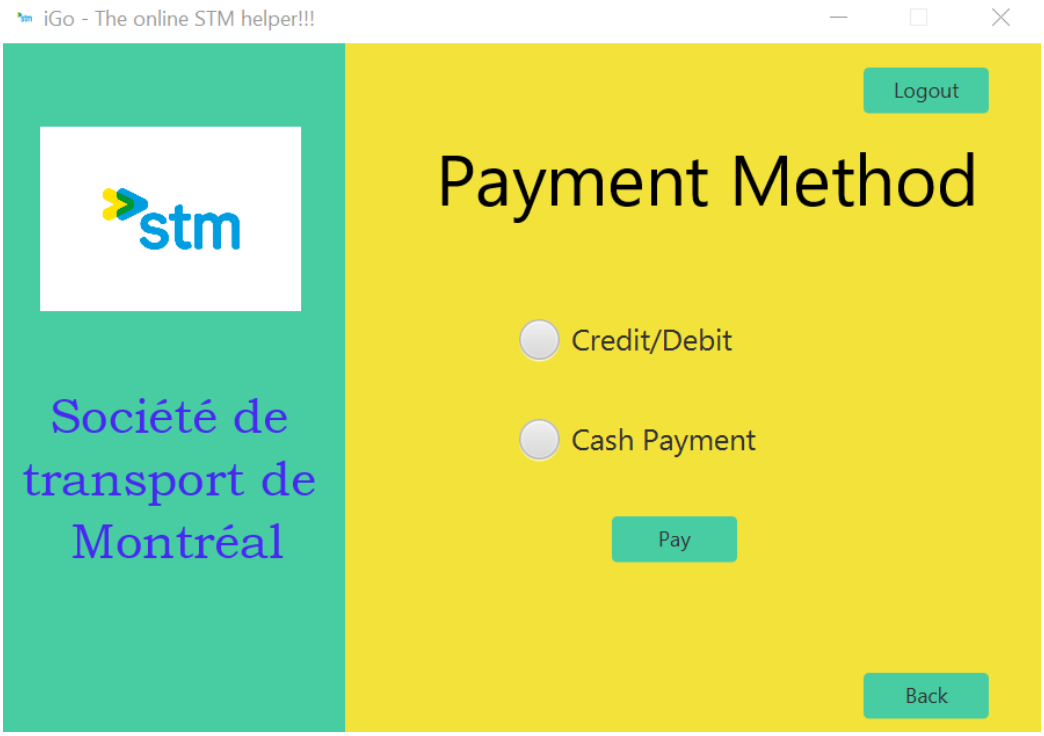


2. Option to buy a new OPUS card

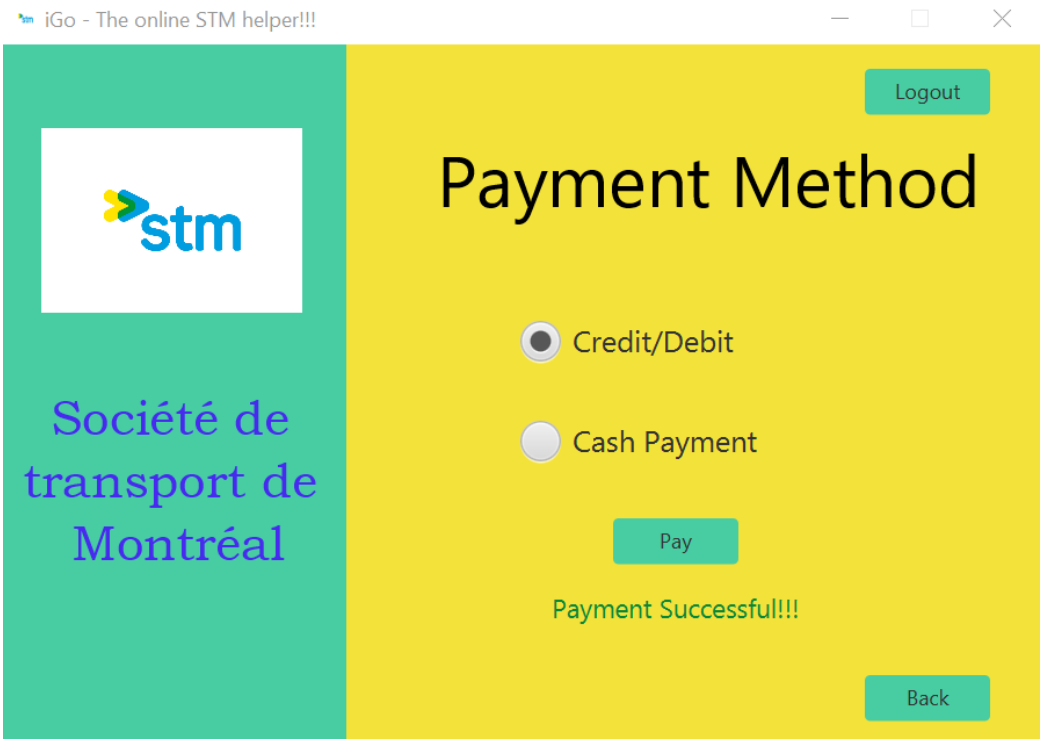
3. Notification after selecting the right option

4.2.2 Buy Ticket

4. Page to select different types of trips and amount of tickets

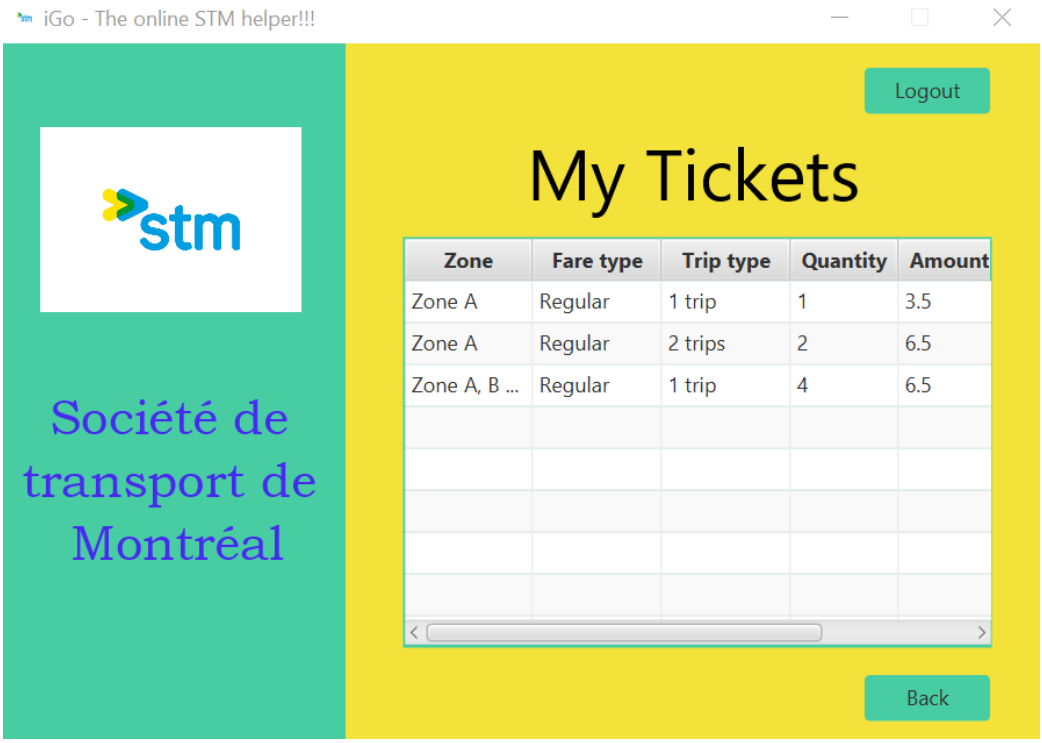


5. The payment method page for buying ticket



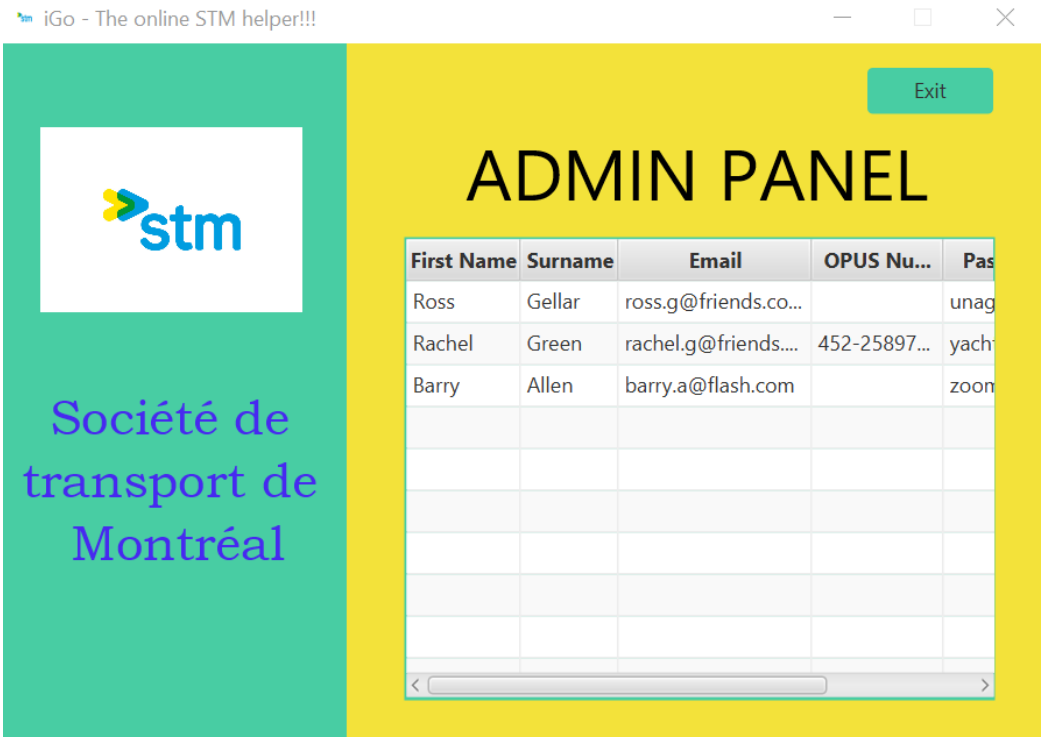
6. Payment success message

4.2.3 View my tickets



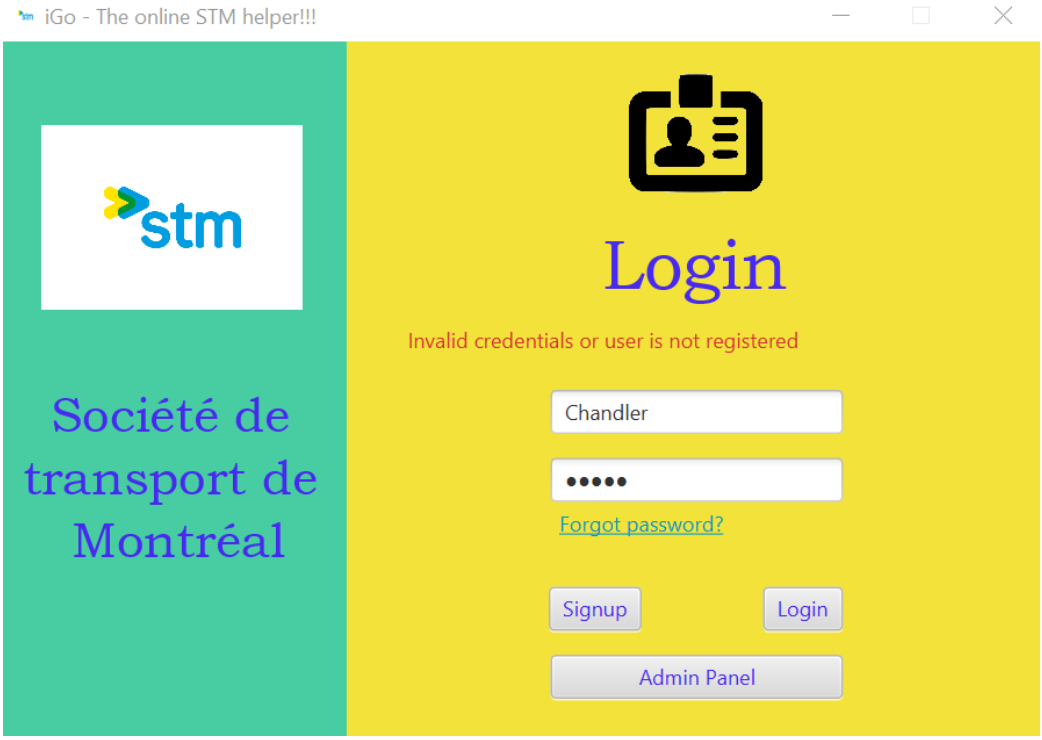
1. Page to show all the tickets purchased by the logged in user

4.3 Admin panel



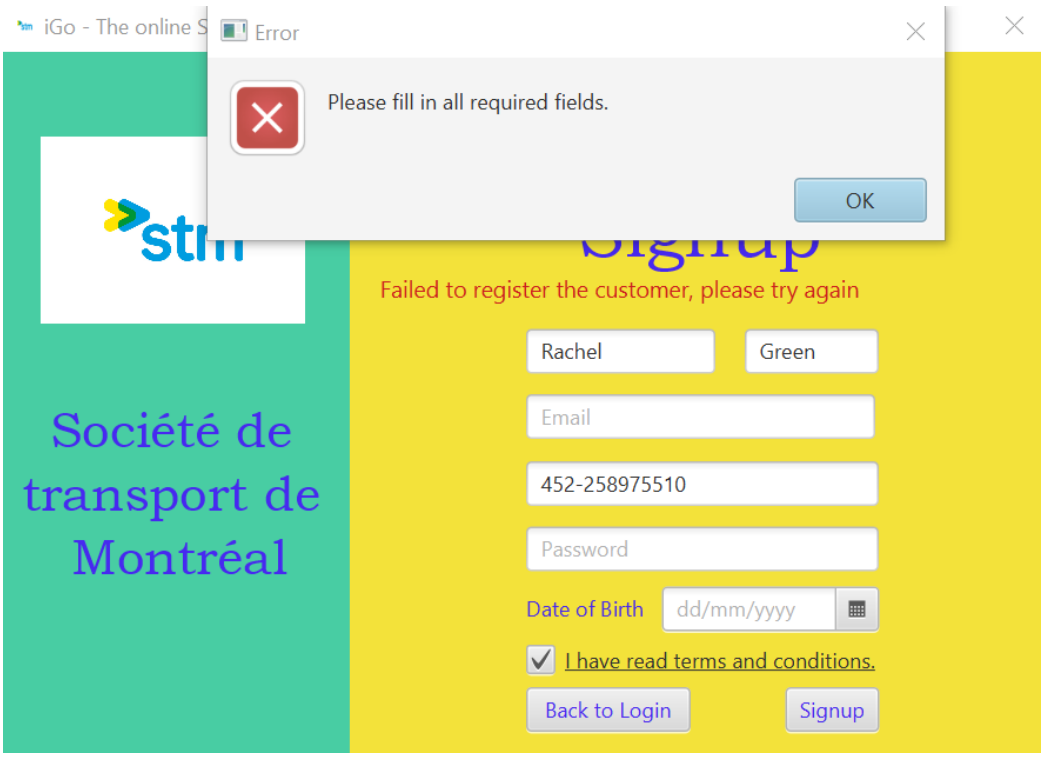
1. Admin panel to view all the available users

4.4 Invalids or Exceptions



1. Unregistered user trying to log in

2. Registered user trying to log in with invalid credentials



3. User trying to register with incomplete details

4. User trying to book appointment without selecting option

5. User trying to click pay without selecting payment method

References

- Structural Design Model
- UML Diagramming
- Sequence Diagram