## Classes where connection is implemented:

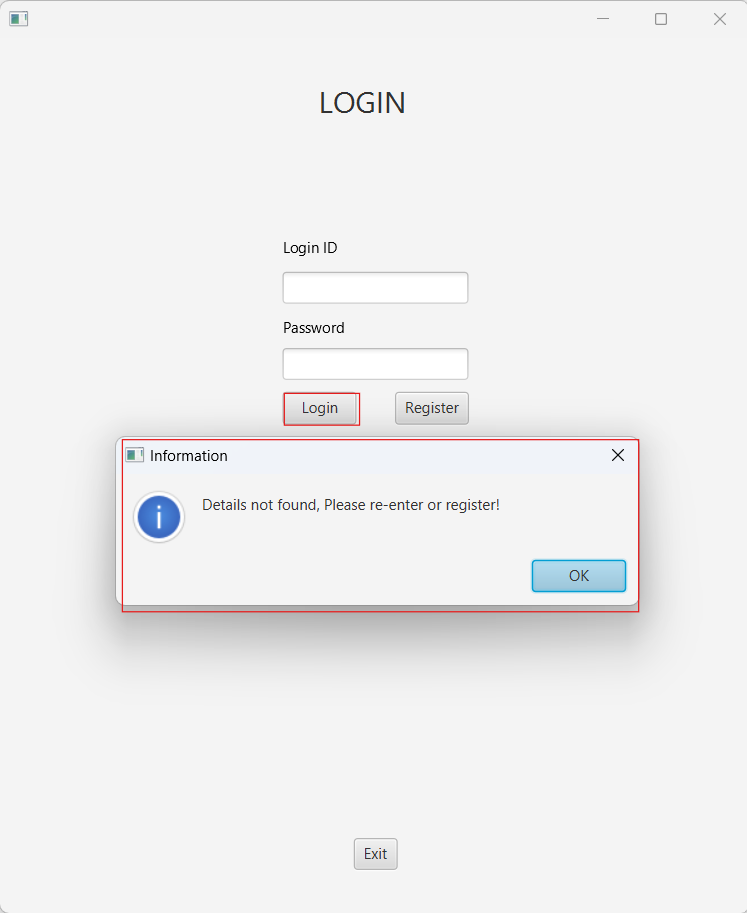
The password and root name are implemented in ‘App Class’.

The connection is implemented in following classes:

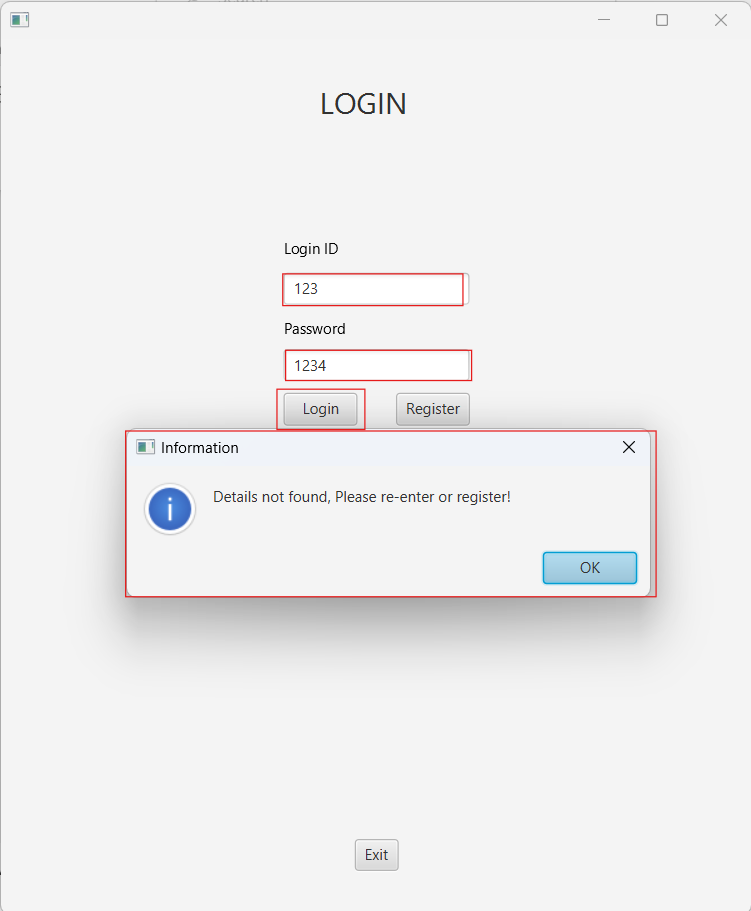
1. CustomerModel
2. PropertyModel
3. JobModel

## Testing Login Page:

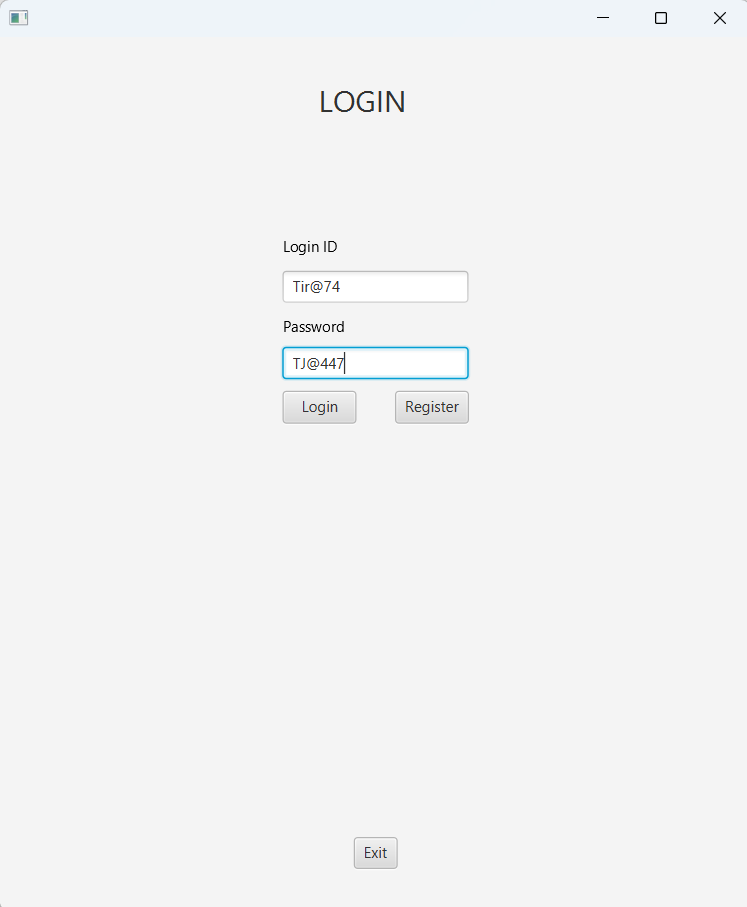
When no data is provided, and login is clicked:



When wrong credentials are provided:



When provided right credentials:



A screenshot of a computer

Description automatically generated

When clicked on Register button:

The user clicked ‘Register’ button on the login page and it lend the user on the below page:

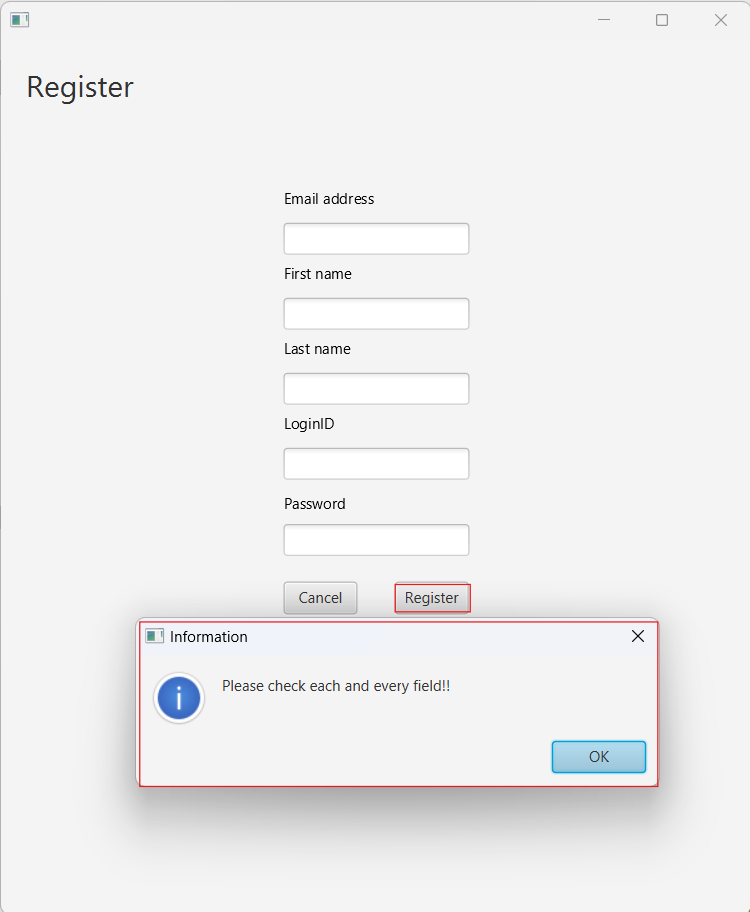
A screenshot of a computer

Description automatically generated

On pressing ‘Exit’ button, the system will successfully close.

## Testing Register Page:

If no details are provided:

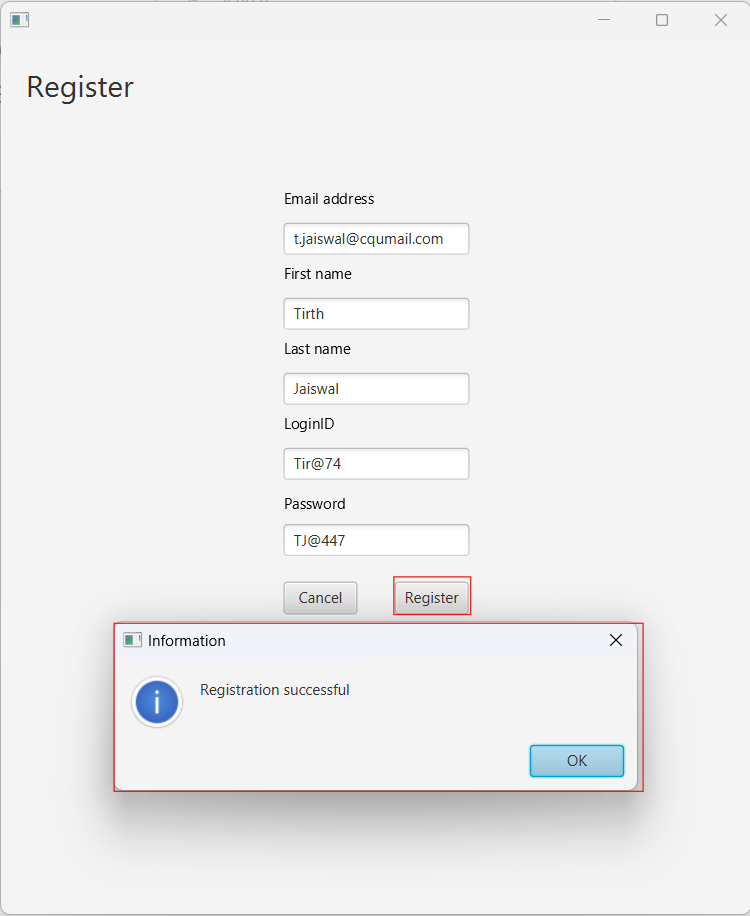


If any detail is left to be filled:

A screenshot of a computer

Description automatically generated

If all details are provided and correct:



When pressed on ‘Cancel’ button:

A screenshot of a login screen

Description automatically generated

# Testing MainMenu:

A screenshot of a computer

Description automatically generated

When clicked on ‘Customers’ button, the FXML will switch to customer search fxml:

A screenshot of a computer

Description automatically generated

When clicked on ‘Properties’ button, the FXML will switch to Property search fxml:

A screenshot of a computer

Description automatically generated

When clicked on ‘Jobs’ button, the FXML will switch to Job search fxml:

A screenshot of a computer

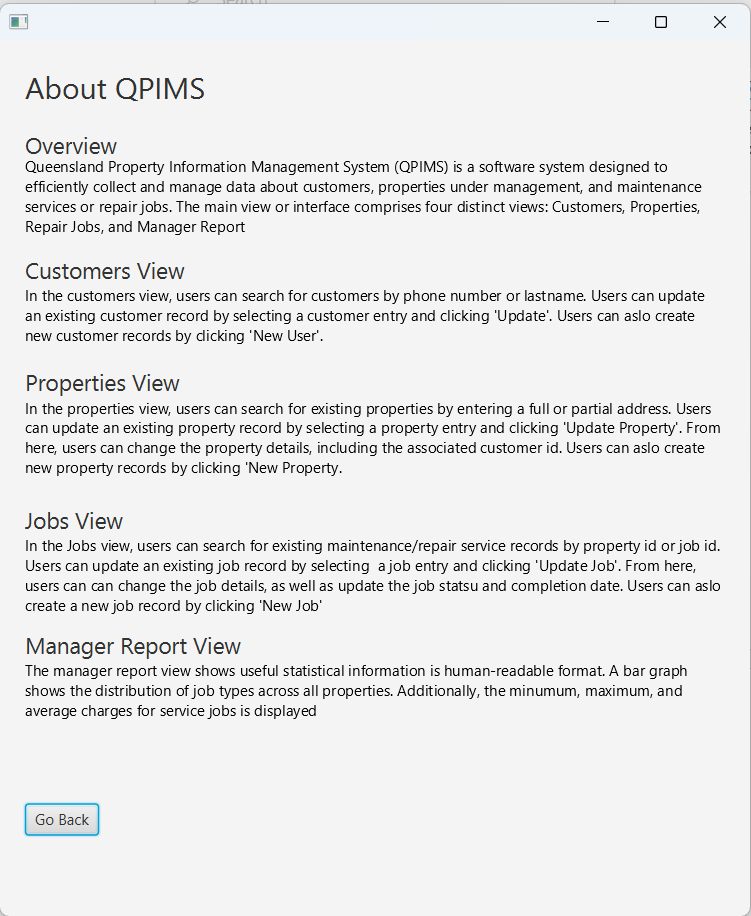
Description automatically generated

When clicked on ‘Report’ button, the FXML will switch to report fxml:

A screenshot of a computer

Description automatically generated

When clicked on ‘About’ button, the FXML will switch to about fxml:



# Testing Customer:

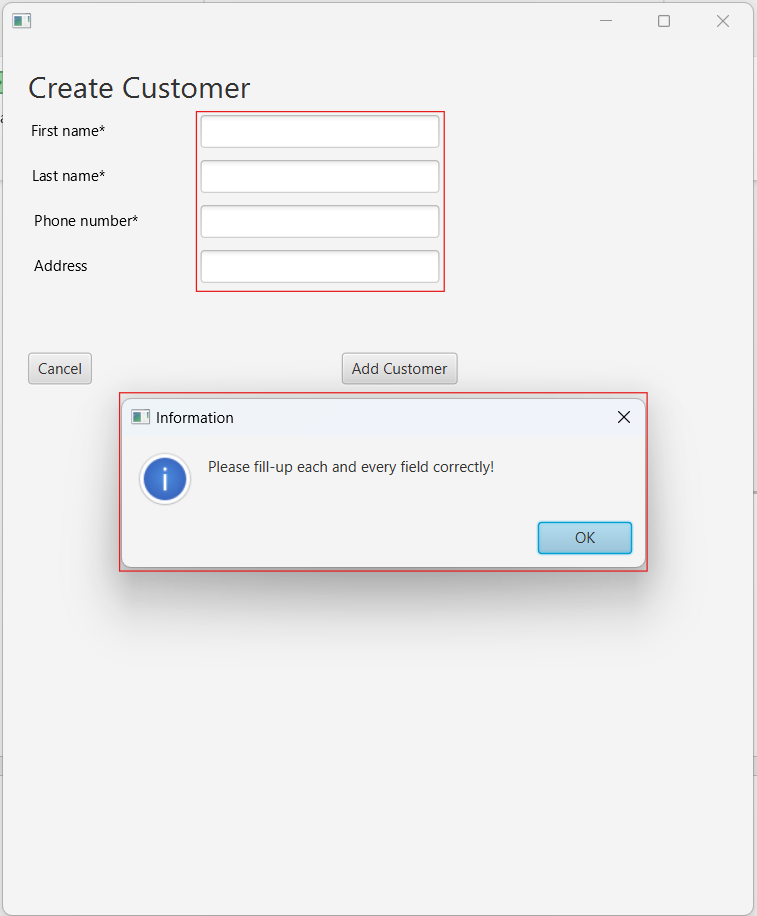
## Testing create customer:

When clicked on ‘New’ button, the FXML will be changed to Create Customer where user can create customer:

A screenshot of a computer

Description automatically generated

In Creating customer, providing no details will result in following outcome:



If there is one or more customer details are missed, in this I have missed the First Name but in reality, it can be any details, the outcome will be:

A screenshot of a computer

Description automatically generated

If all the details are correctly provided, the outcome expected will be:

A screenshot of a computer

Description automatically generated

When clicked on the back button, the create customer FXML will be switched to search customer FXML:

A screenshot of a computer

Description automatically generated

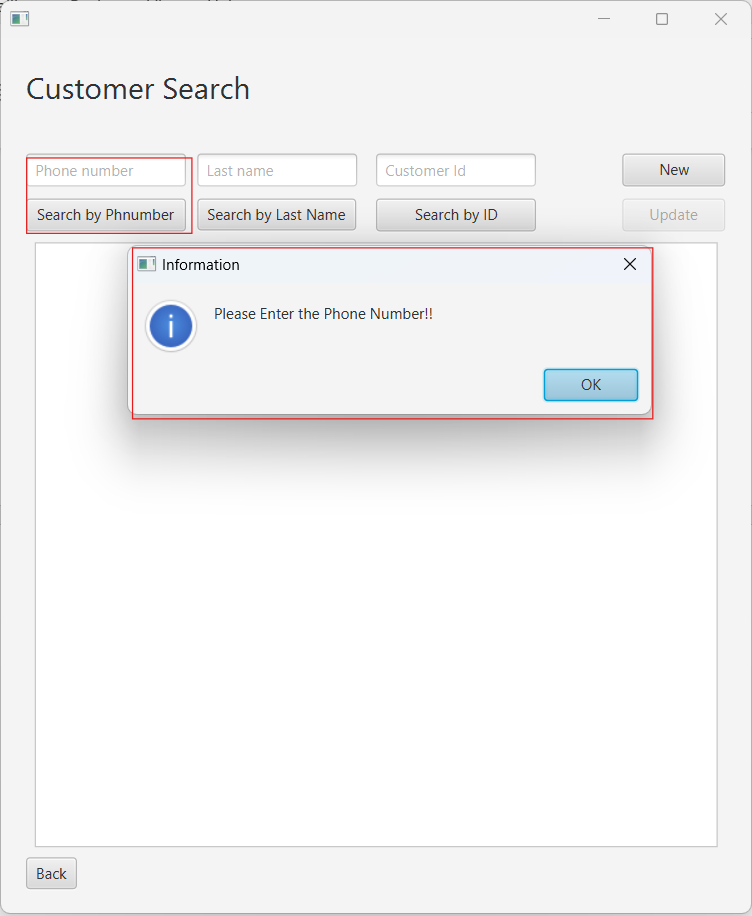
A screenshot of a computer

Description automatically generated

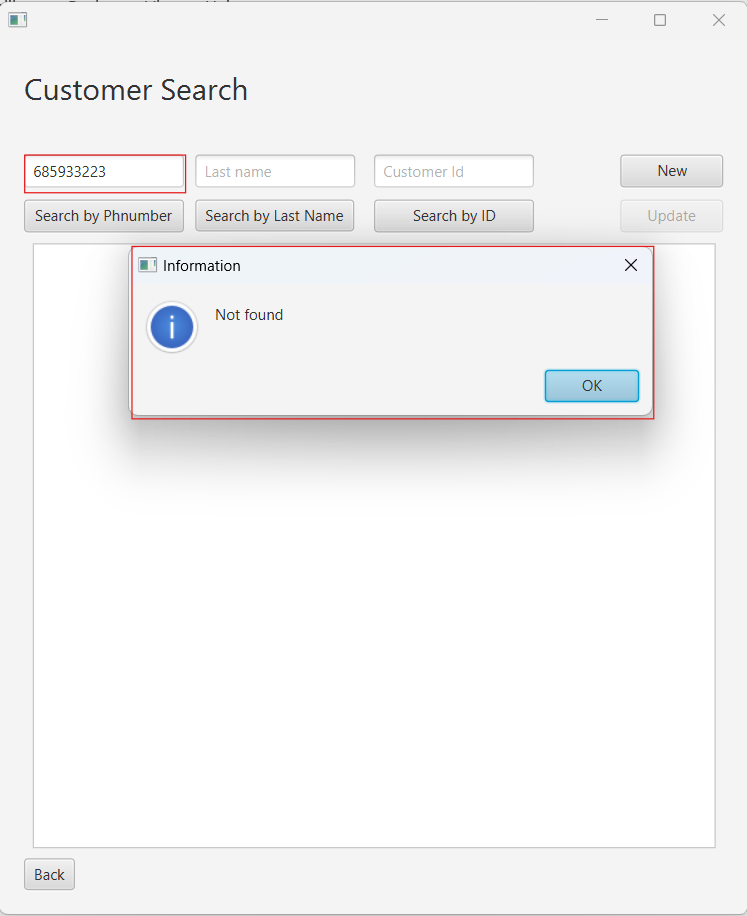
## Testing search customer:

### Testing search using phone number:

While searching using the phone number, if no phone number is provided, following output is expected:



If incorrect phone number is provided which does not exist in the system, the output is expected to be:



If the input format is used any other than integers, the output expected to be is:

A screenshot of a computer

Description automatically generated

Finally, if the input is correct:

A screenshot of a computer

Description automatically generated

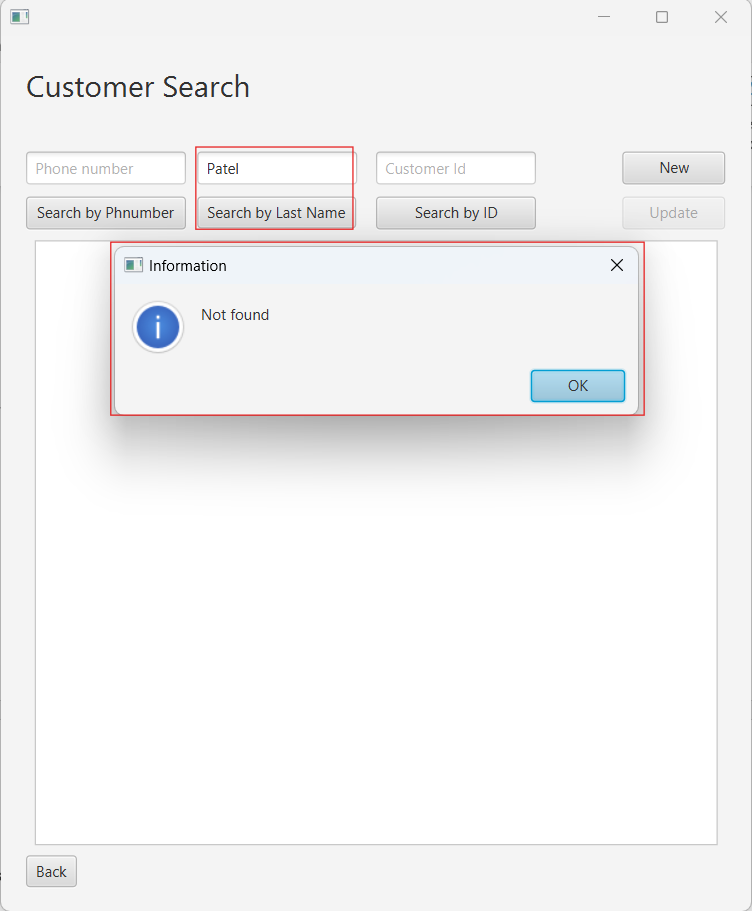
### Testing search using last name:

While searching using the last name, if no input is provided, the output is expected to be:

A screenshot of a computer

Description automatically generated

If the incorrect last name is entered which is not present in the database, the expected outcome is:



Finally, if the correct information is provided as an input for last name:

A screenshot of a computer

Description automatically generated

### Testing search using customer id:

While searching with customer id, if the customer id is not provided:

A screenshot of a computer

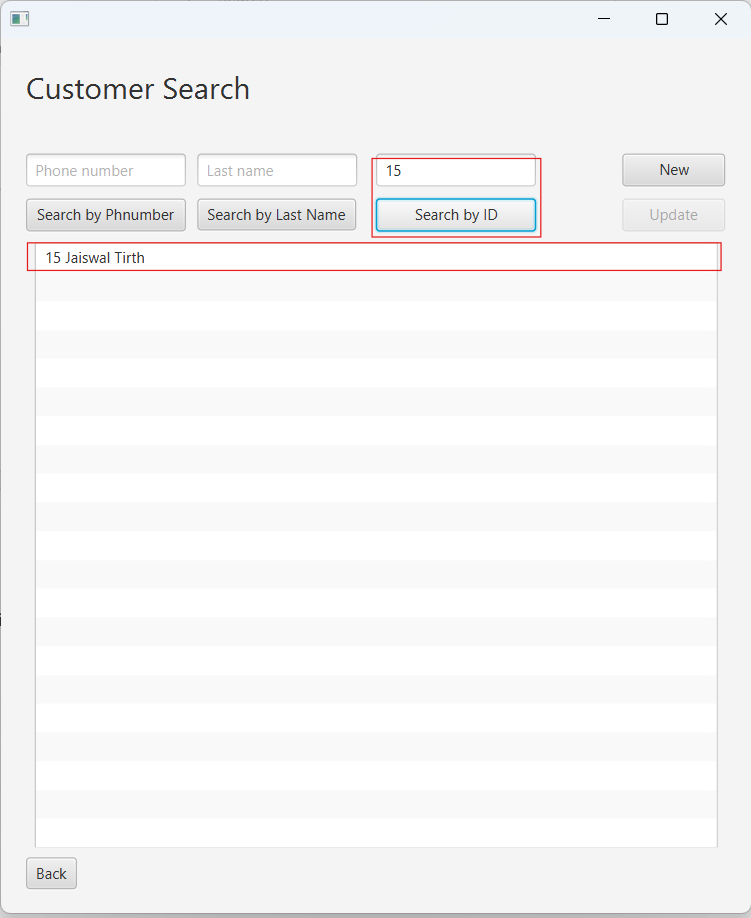
Description automatically generated

If incorrect customer id is provided:

A screenshot of a computer

Description automatically generated

Finally, if the correct and existing customer id is provided as the input:



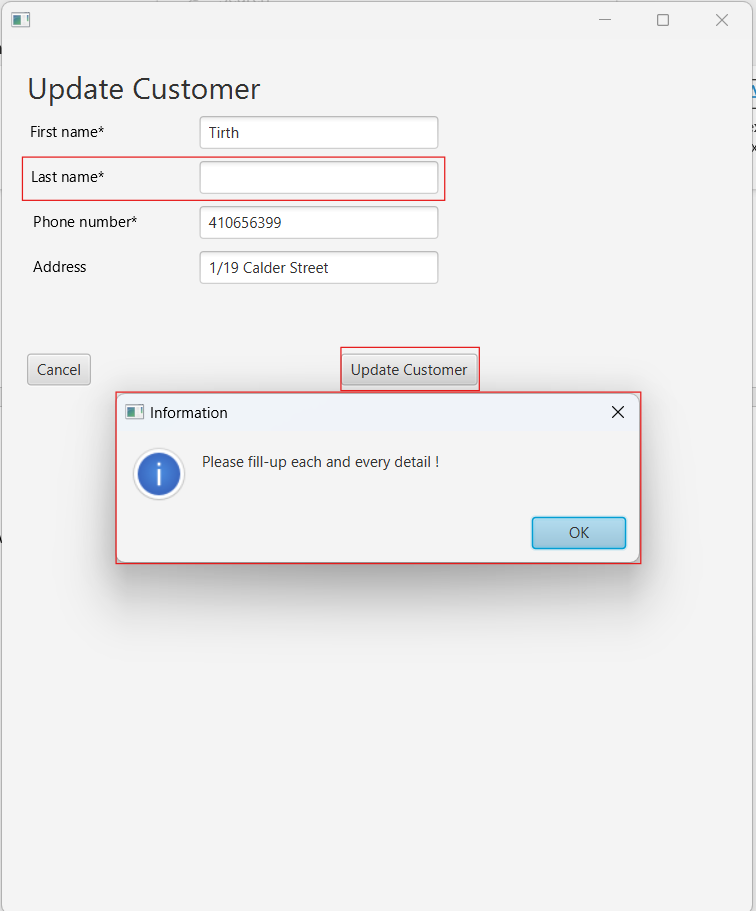
## Testing update customer:

When a customer entry from the list view is selected, the update button gets activated, customer can click on update the customer details by clicking on the update button:

**A screenshot of a computer

Description automatically generated**

When a detail is missing while updating the customer details, following outcome is expected:



When all the details are entered, the customer details will be update in database:

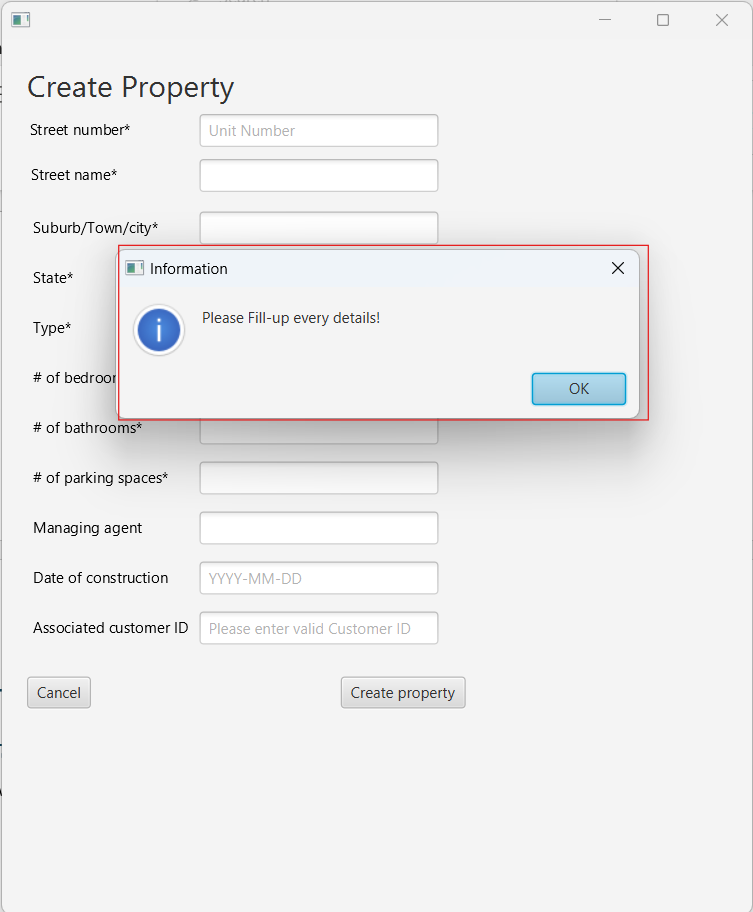
A screenshot of a computer

Description automatically generated

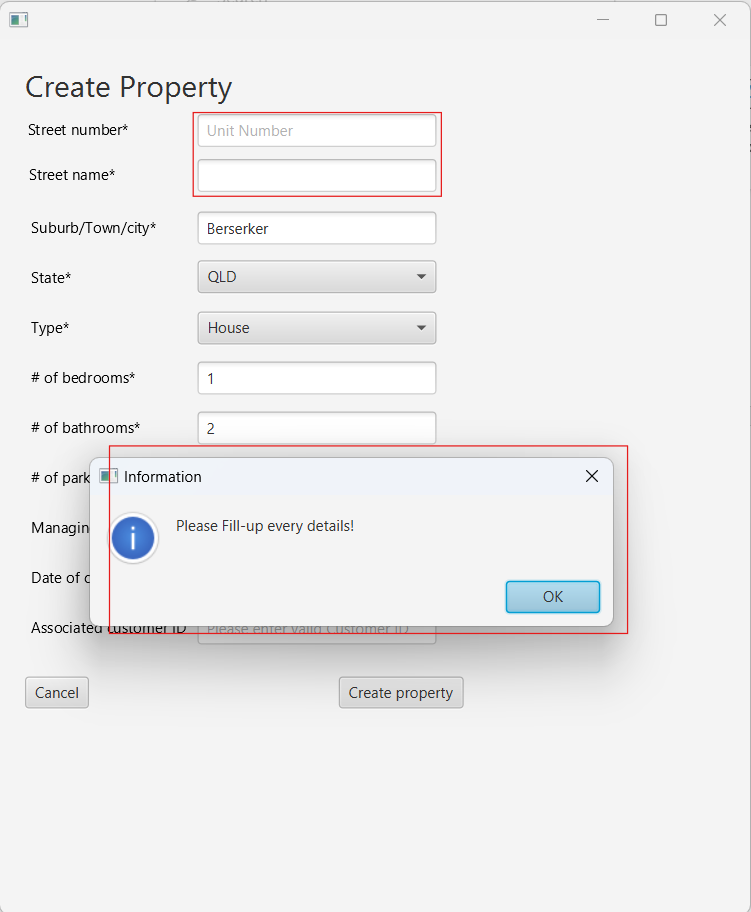
# Testing Properties:

## Testing create property:

While creating a property, if no details are provided, the outcome expected is:



If one or more details are missed, the outcome expected to be is:

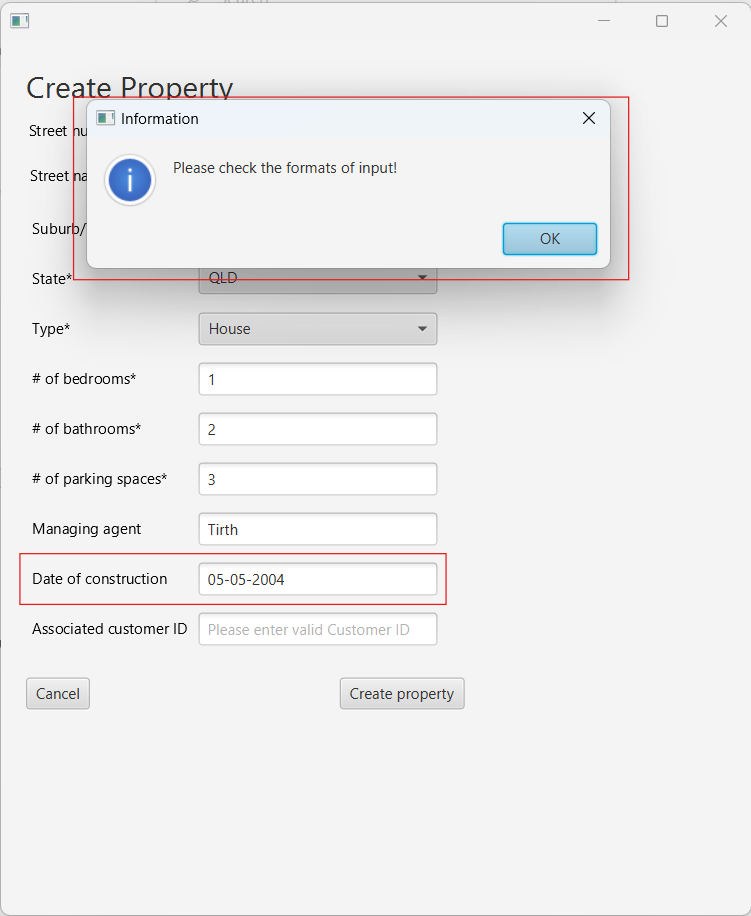


If the format of input wrong, i.e. at the places where integer is required and string is provided, the outcome expected to be is:

A screenshot of a computer

Description automatically generated

If while creating a property, if the date format is not according to ‘yyyy-mm-dd’, the following outcome is expected:

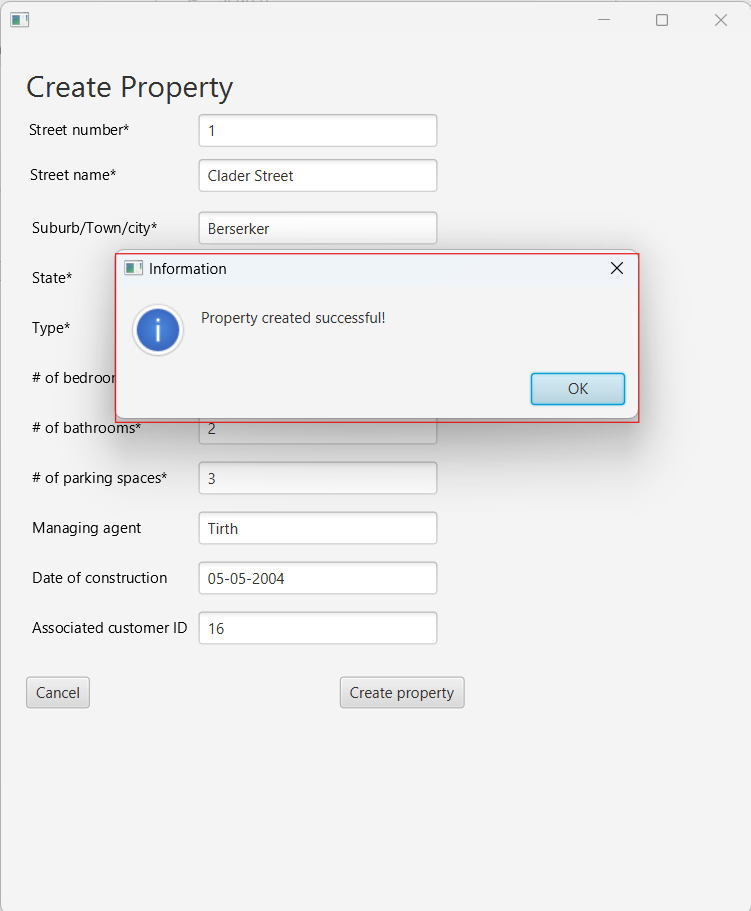


If the customer ID provided is not in the database, following outcome is expected:

A screenshot of a computer

Description automatically generated

If all the details are valid and customer ID is existing, following outcome is expected:



## Testing search property:

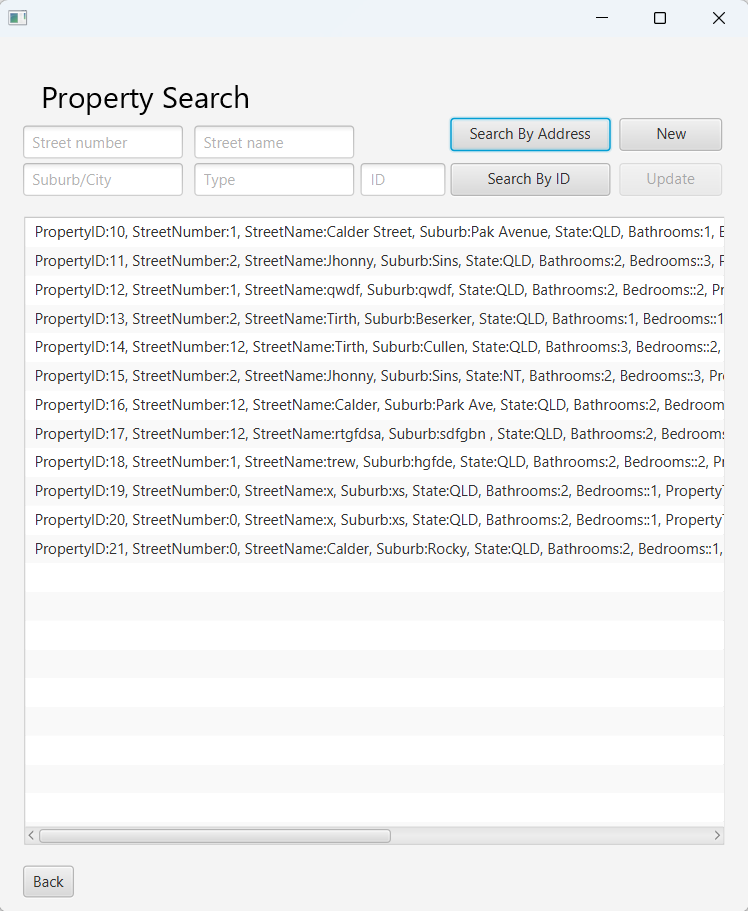
### Testing search property using ‘Search by Address’:

The search by address works on inputs provided in text fields:

1. Street Number
2. Street Name
3. Suburb
4. Type

If anyone of the text input is provided, the search is going to take place with that one input but if more than one input is provided, the search is going to take place with combination of those inputs.

If no input is provided, all the result stored in database are expected to be displayed on list view:



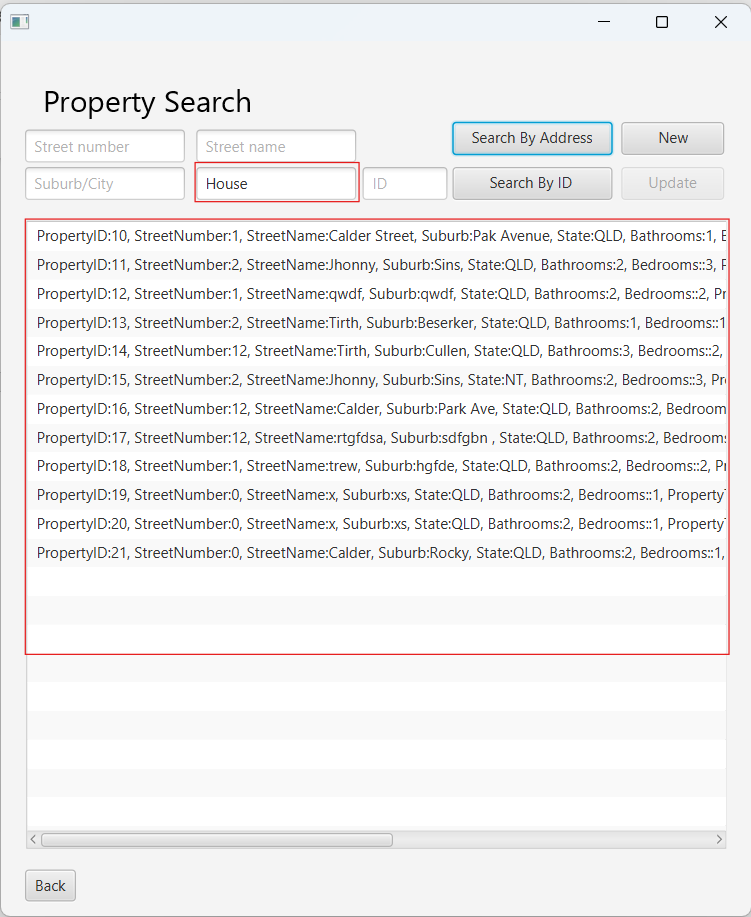
If combination of input is provided, the search will be happening with help of that combination.

For example, lets say, street name Tirth and type House is provided, is the result matching with this input fields will be displayed:

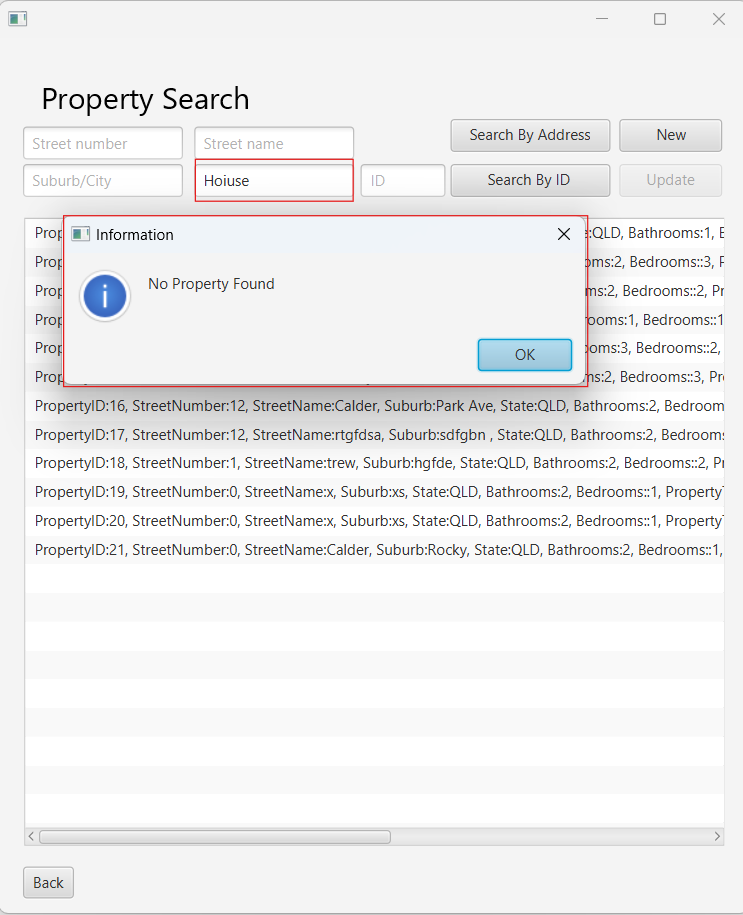
A screenshot of a search box

Description automatically generated

If single search input is provided, the following outcome is expected:



If incorrect input is provided, which does not match any result stored in the database, the following outcome is expected:



### Testing search property with property Id:

If the ID provided is not an integer, following outcome is expected:

A screenshot of a computer

Description automatically generated

If incorrect property Id is provided:

A screenshot of a computer

Description automatically generated

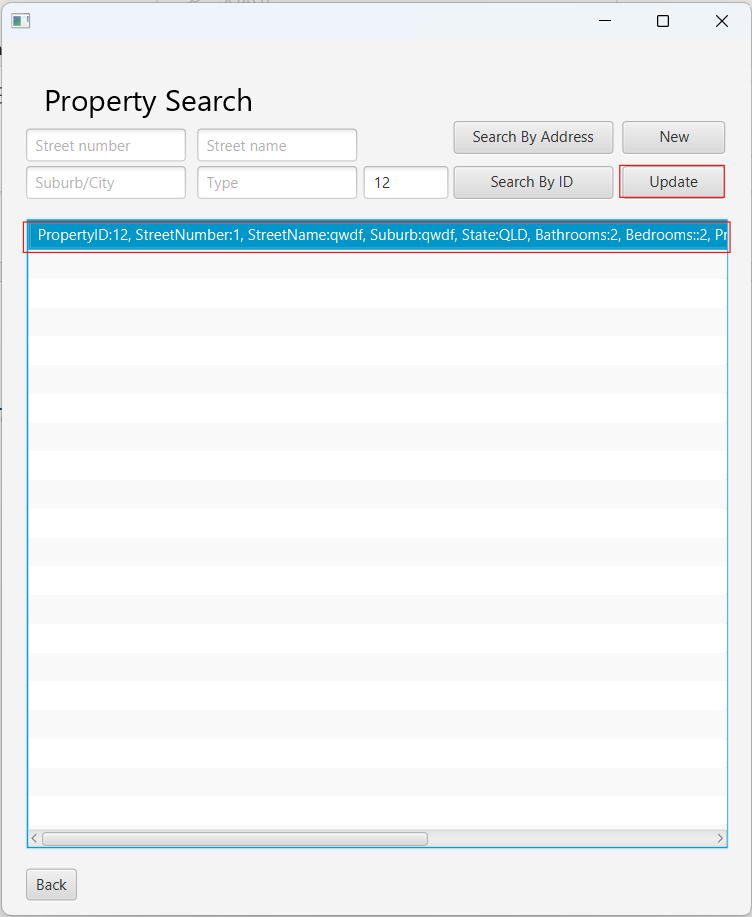
If correct property Id is provided:

A screenshot of a search box

Description automatically generated

## Testing update property:

If the property entry from the listview is selected, the update button will become active with will help in updating the selected property:



While updating, if any field is left to be filled, the following outcome is expected:

A screenshot of a computer

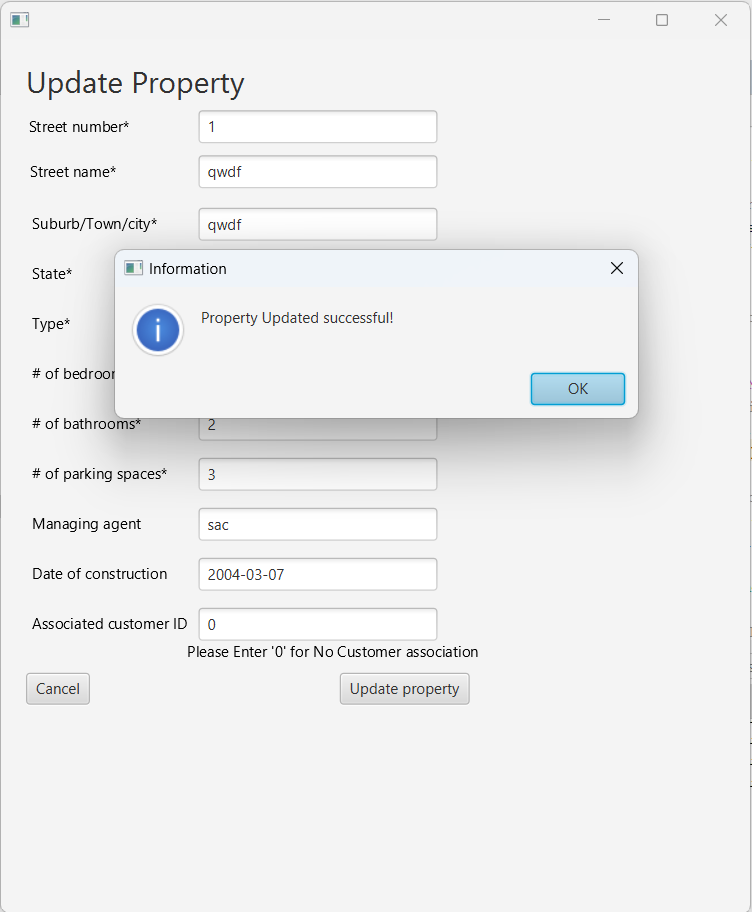
Description automatically generated

If the format of the fields are incorrect, the following outcome is expected:

A screenshot of a computer

Description automatically generated

If all details are correct and provided, the property will be updated:



# Testing Job class:

## Testing create Property:

Create page can be accessed with by clicking on ‘New Booking’ button on the search page:

A screenshot of a computer

Description automatically generated

While making a booking, if property Id is not provided, following outcome is expected:

A screenshot of a computer

Description automatically generated

StartDate is not provided or provided in a wrong format:

A screenshot of a computer

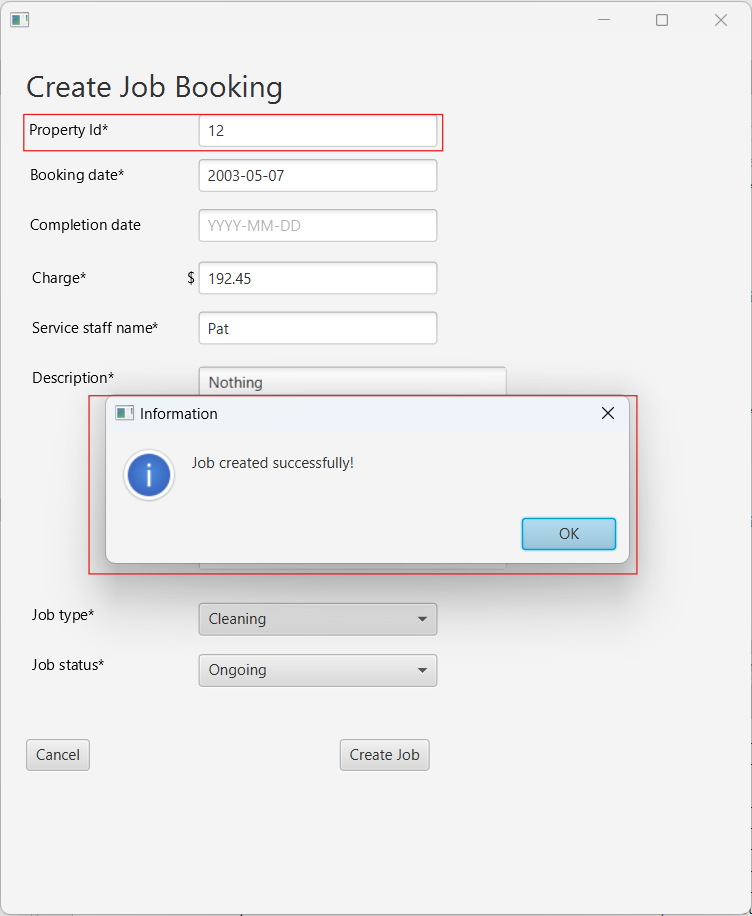
Description automatically generated

If any other data is not provided:

A screenshot of a computer

Description automatically generated

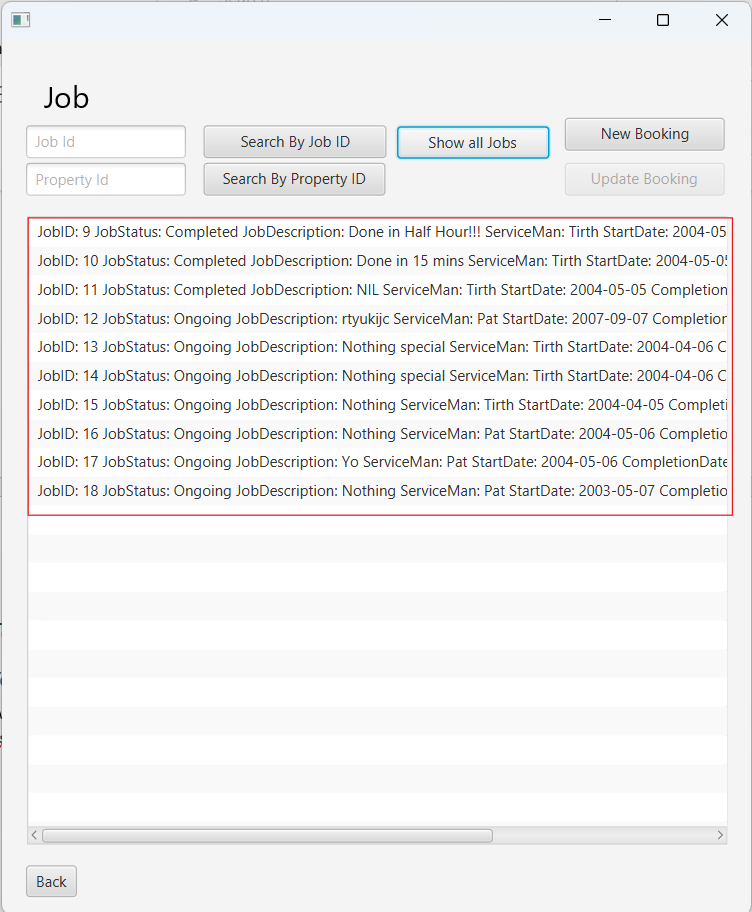
If all the details are provided and valid, the job will be created:



## Testing search Job:

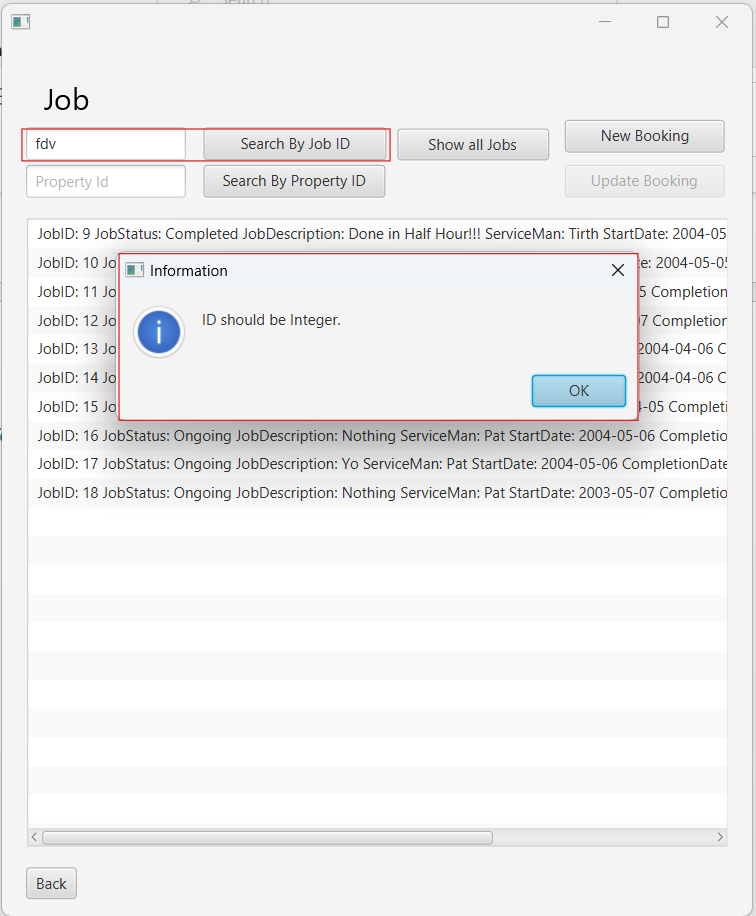
### Testing the ‘show all jobs’

When clicked on show all the jobs, all the jobs stored in the database will be displayed on the listview:

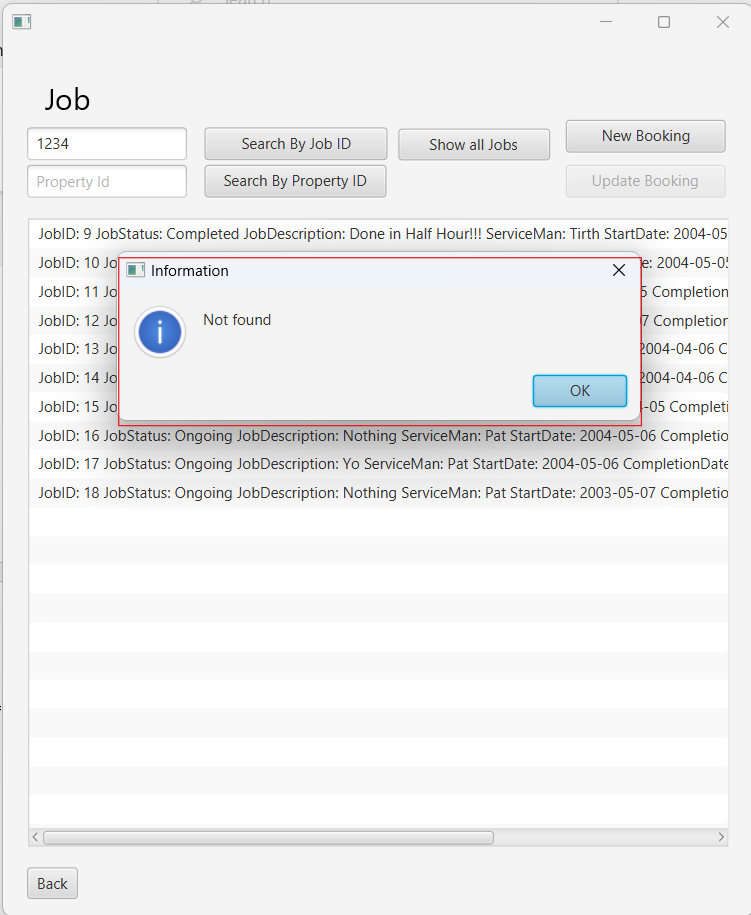


### Testing search jobs by JobID:

If the ID entered is not an integer:



If the ID provided does not exist in the database:



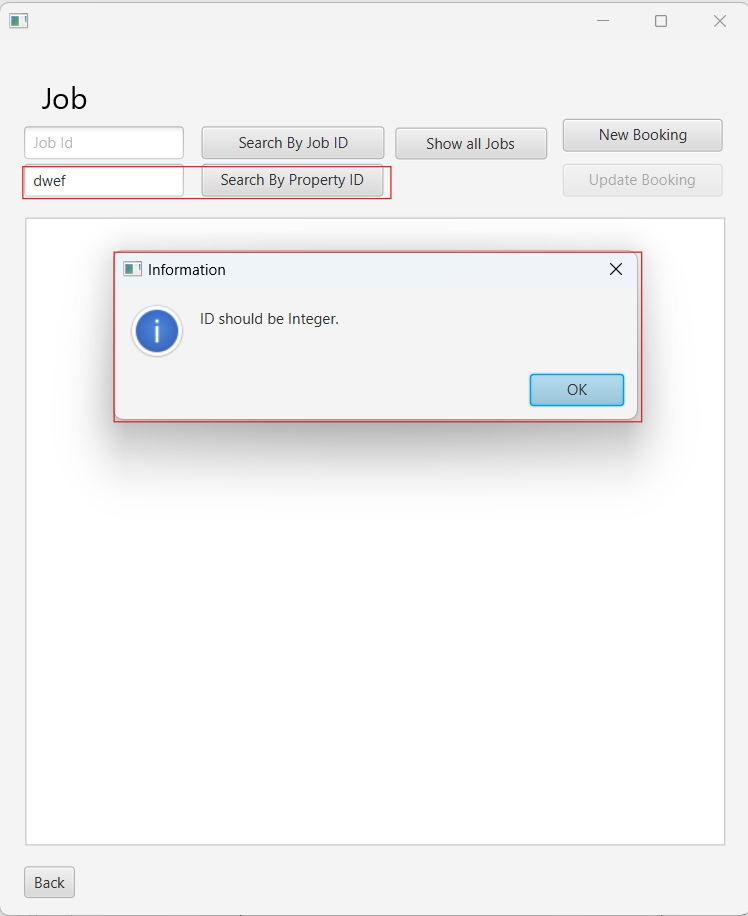
If the job ID provided is correct and exists:

A screenshot of a computer

Description automatically generated

### Testing search job with propertyid:

If the property Id provided is not in integer format:



If the property Id provided does not exists:

A screenshot of a computer

Description automatically generated

If the property Id provided is correct and exists, the job related to that property is expected to be in listview:

A screenshot of a computer

Description automatically generated

# Testing Update Job:

Jobs can be updated by selecting them from the list view and clicking on the update button:

A screenshot of a computer

Description automatically generated

If any details are changes but details do not satisfy the system requirements, some outcome like below is expected:

A screenshot of a computer

Description automatically generated

If all the details are correct, the job will be updated.

Job update function will be majorly used from changing the status of job from ongoing to completed:

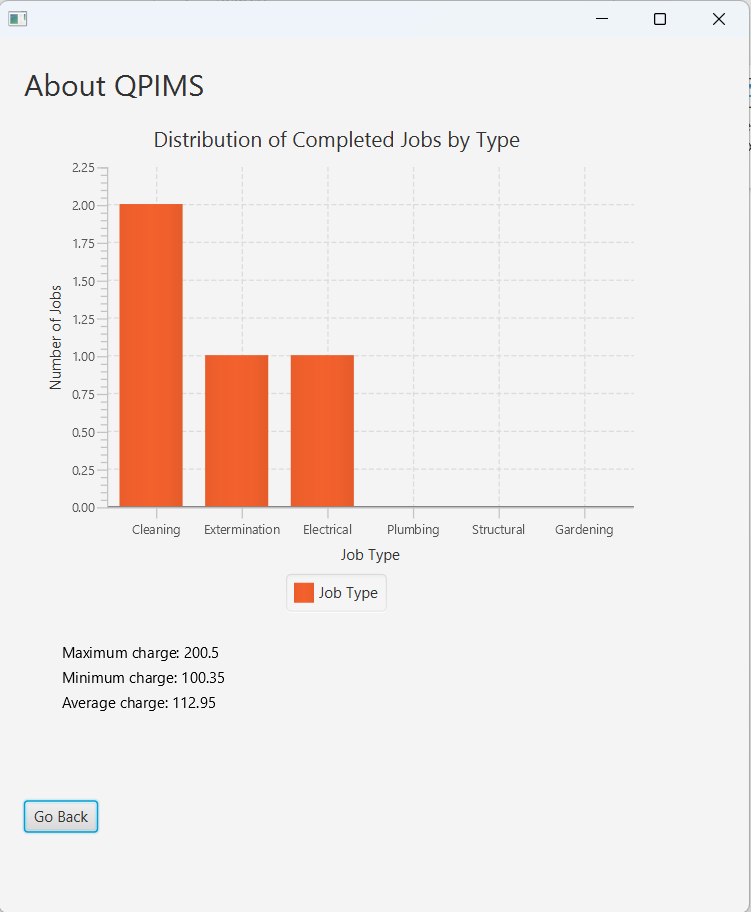
A screenshot of a computer

Description automatically generated

# Testing Report:

In this section, a bar graph is displayed which depicts the number of jobs with complete status. The x-axis on the bar graph signifies the classification of jobs like cleaning, electrician, etc. where the y-axis signifies the number of ‘completed’ jobs in that classified field.

The maximum charge, minimum charge and average charge are calculated on basis of all jobs regardless of completed, ongoing or cancelled.



# Testing About:

In this page of application, the information around different sections of the application i.e. customer, property and job are given. The about page looks like:

A screenshot of a computer

Description automatically generated