

Adi Dinner - 302658711

Benny Stan - 305117020

Pattern Design - 61988 - Final Project

Company Insurance

Table of Contents

[2 About the Application 2](#_Toc13775449)

[3 Application User Manual 2](#_Toc13775450)

[3.1 User 2](#_Toc13775451)

[3.1.1 Login 2](#_Toc13775452)

[3.1.1.1 Fail to login 3](#_Toc13775453)

[3.1.1.2 Loader – Successful login establish (Green circles) 3](#_Toc13775454)

[3.1.2 Settings 3](#_Toc13775455)

[3.2 Client 4](#_Toc13775456)

[3.2.1 Home 4](#_Toc13775457)

[3.2.2 Overview 4](#_Toc13775458)

[3.2.3 Insurance 5](#_Toc13775459)

[3.2.4 Claims 5](#_Toc13775460)

[*3.3* Customer Service 6](#_Toc13775461)

[3.3.1 Home 6](#_Toc13775462)

[3.3.2 Overview 6](#_Toc13775463)

[3.3.3 Claims 7](#_Toc13775464)

[3.4 Salesman 7](#_Toc13775465)

[3.4.1 Home 7](#_Toc13775466)

[3.4.2 Overview 8](#_Toc13775467)

[3.4.3 New Insurance 8](#_Toc13775468)

[3.4.3.1 Fail 9](#_Toc13775469)

[3.4.4 New Insurance & Client 9](#_Toc13775470)

[3.4.4.1 Fail 10](#_Toc13775471)

[3.4.5 Customer 10](#_Toc13775472)

[4 UML Diagrams 0](#_Toc13775473)

[5 Design Patterns 0](#_Toc13775474)

[5.1 MCV 0](#_Toc13775475)

[5.2 Singleton 0](#_Toc13775476)

[5.3 Façade 0](#_Toc13775477)

[5.4 Factory 0](#_Toc13775478)

# About the Application

Application is handling any insurance that the company is offers (Car, House, Loss of Job and Life). The application is for any user that have username and password as he need to login with it. Each different of user have deafferent menu that can access. Moreover, all the data is store on a Azure SQL exclude the XML configuration file that handle the insurance type. This mean that if company will add a new insurance there is no need to any GUI or code. Duo to this approach there is no need to develop anything for the new insurance.

Each **User** actions are saving in the log file with the time and date and the class that the action occurred.

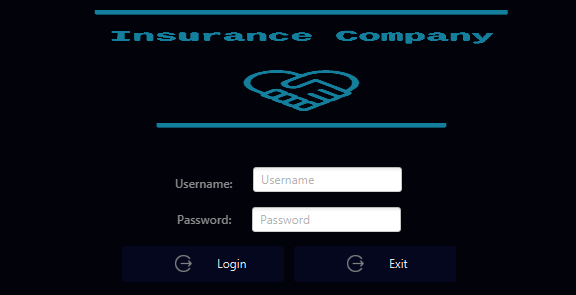
Example: 17.06.2019 - 10:31:01 -- app.Controllers.Login Yosi Dodo Sign out.

# Application User Manual

## User

### Login

For login into the application user (Id) and password needed. Each user (Client, Salesman and Customer Service) need to use his own authorization information. If the information is correct the user will enter to his menu. But, if any of the field is missing or incorrect the application will notify the user.



To login to the application – If we didn’t change the password in the settings.

Client

Username: 012012012

Password: 1234

Customer Service

Username: 012345678

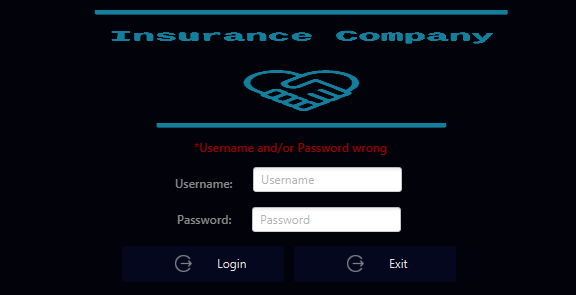
Password: 1234

Salesman

Username: 123456789

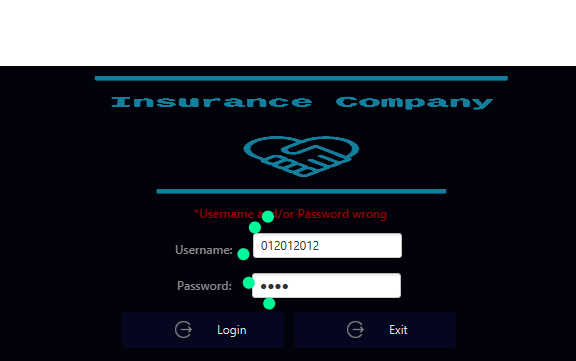
Password: 1234

#### Fail to login



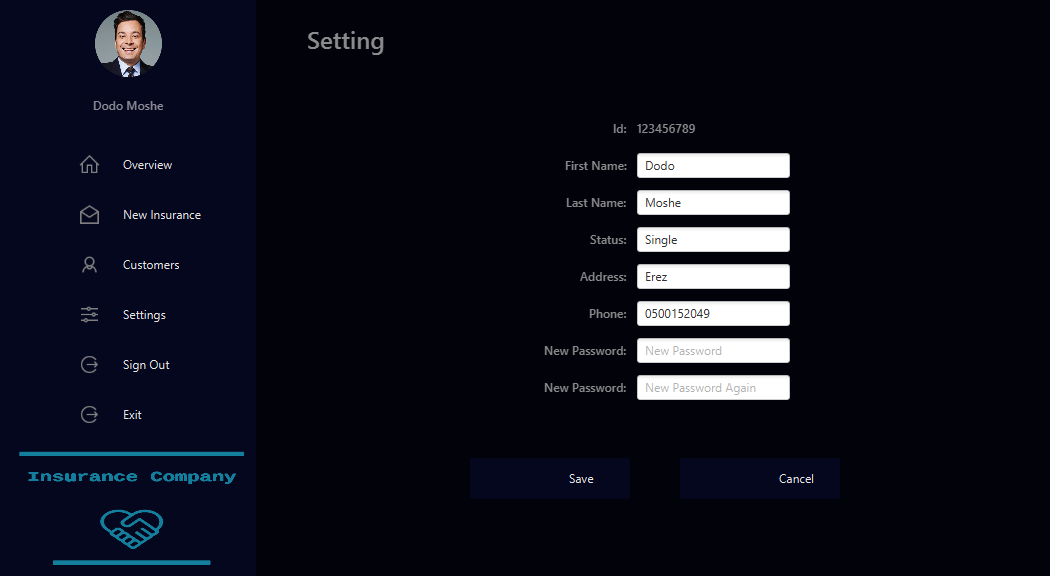
#### Loader – Successful login establish (Green circles)

When the application load there are green circles that present to the user to notification for processing and loading all the information that the user will need. This mean only one load for whole user GUI.



### Settings

Default settings for any user that want to change his information exclude his Id.



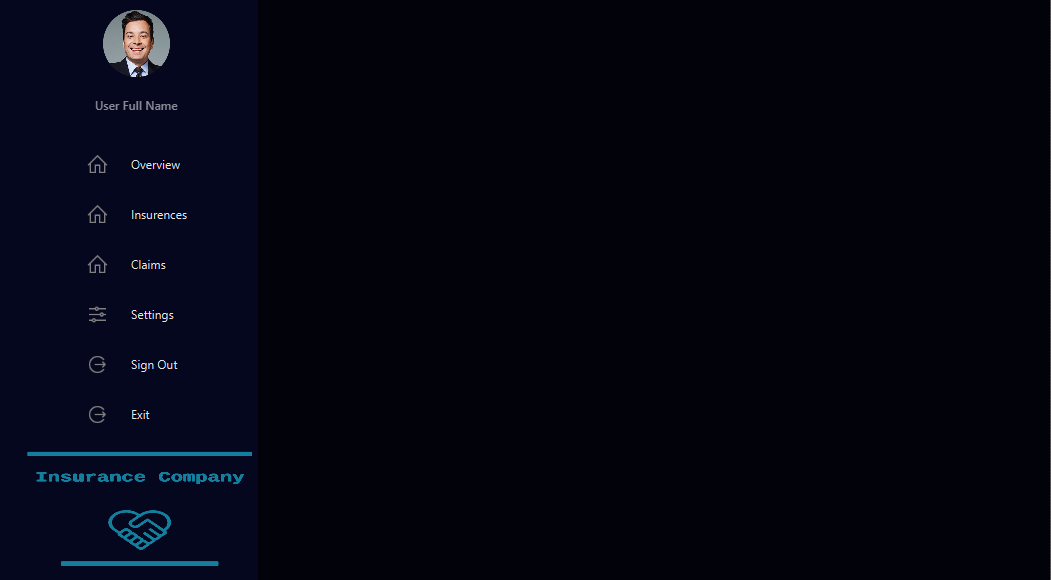
On successful saving a message will popup to notify the user that the changes saved and updated.

## Client

The client cannot create a new insurance in the application only see his information regard to his insurances. But the client can create a claim on an exist insurance.

### Home

Home is split to two parts, the first part is the menu that present to the user, Client, and the second part is where the page that will present to the client when clicking on the menu’s buttons.



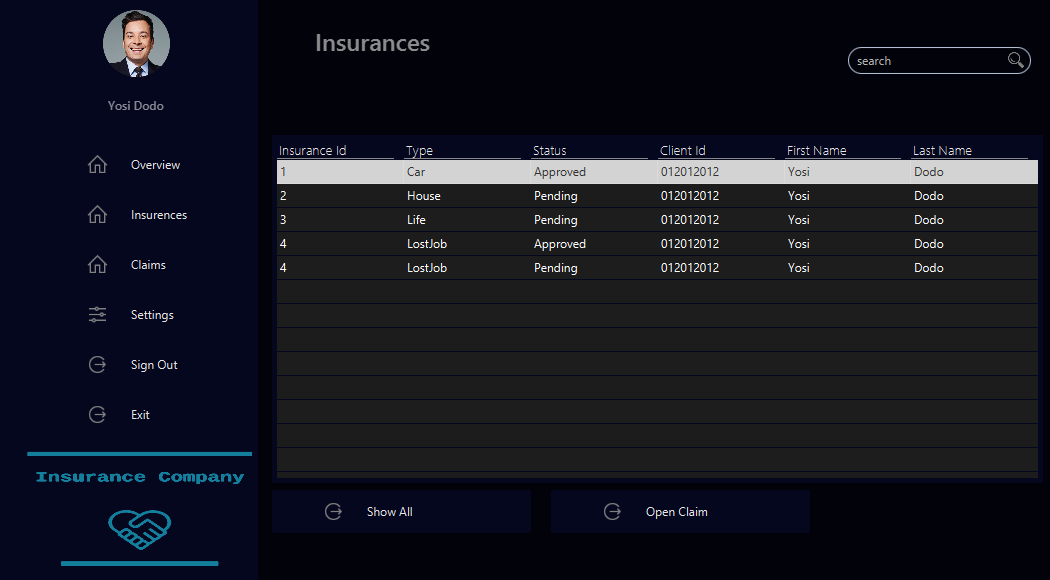
### Overview

Overview present the information regard to the login client. It splits to three sections. The first section presents the personal information, the second section present the total insurances and claims by numbers, and the third section is present with chart pi for better visualization.



### Insurance

Present all the insurances that the specific client has. If the client wants to open a new claim on an exist insurance, it can be done by clicking the Open Claim.



### Claims

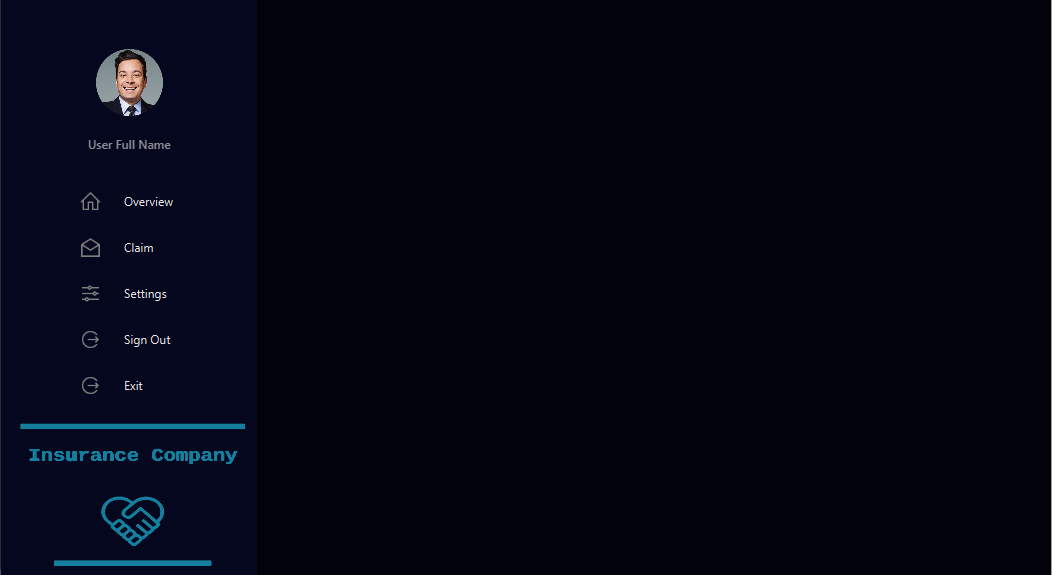
Present all the claims for a specific client.



## Customer Service

### Home

Home is split to two parts, the first part is the menu that present to the user, Customer Service, and the second part is where the page that will present to the client when clicking on the menu’s buttons.



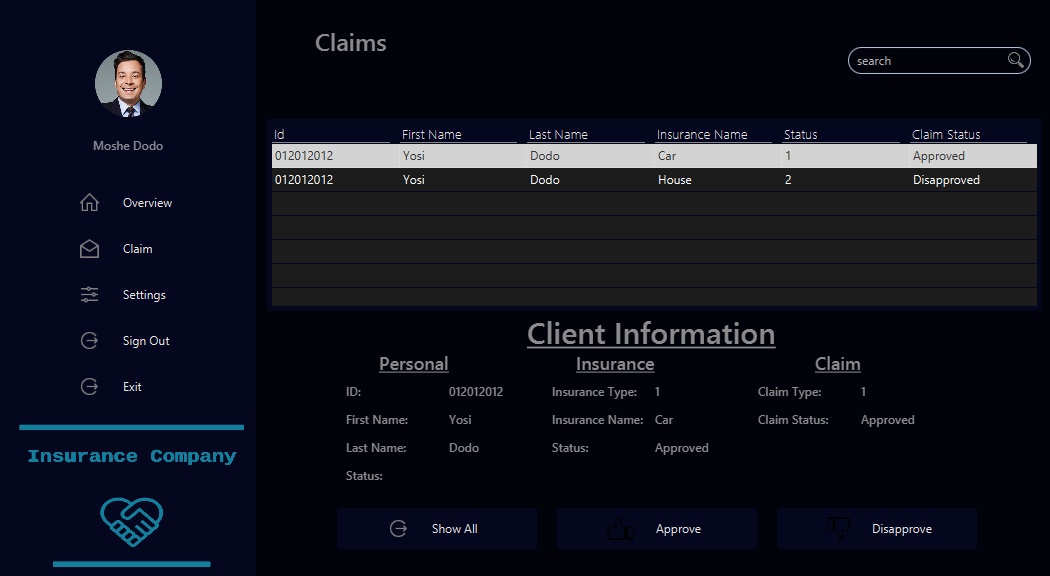
### Overview

Overview present the information regard to the login client. It splits to two sections. The first section presents the total claims by numbers, and the second section is present with chart pi for better visualization.



### Claims

The Customer Service have the option to Approved or Disapproved a specific claim. A claim can change at any time as the company decides.



## Salesman

### Home

Home is split to two parts, the first part is the menu that present to the user, Client, and the second part is where the page that will present to the client when clicking on the menu’s buttons.



### Overview

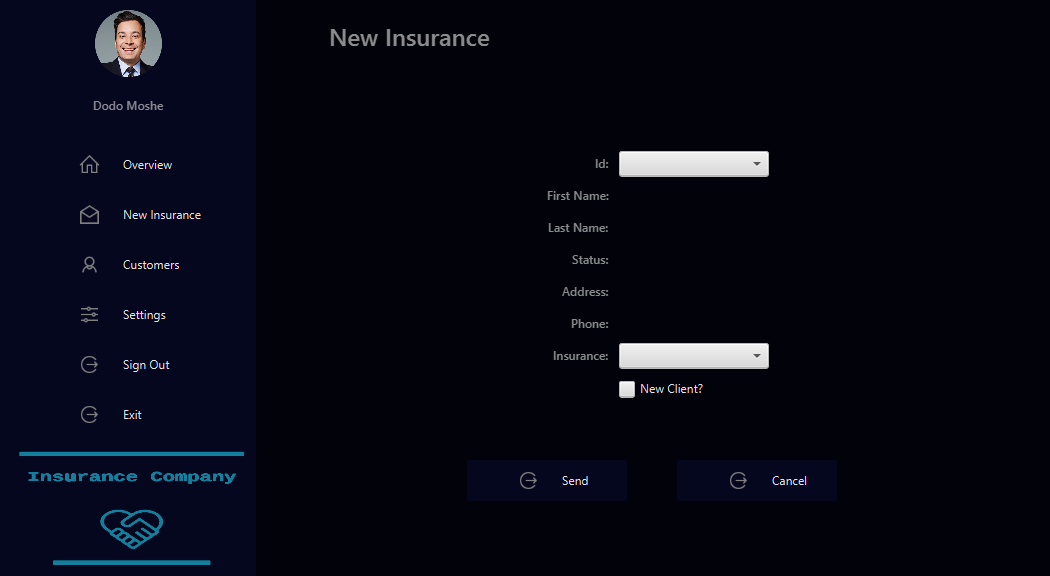
Overview present the information regard to the login client. It splits to two sections. The first section presents the total insurances by numbers, and the second section is present with chart pi for better visualization.



### New Insurance

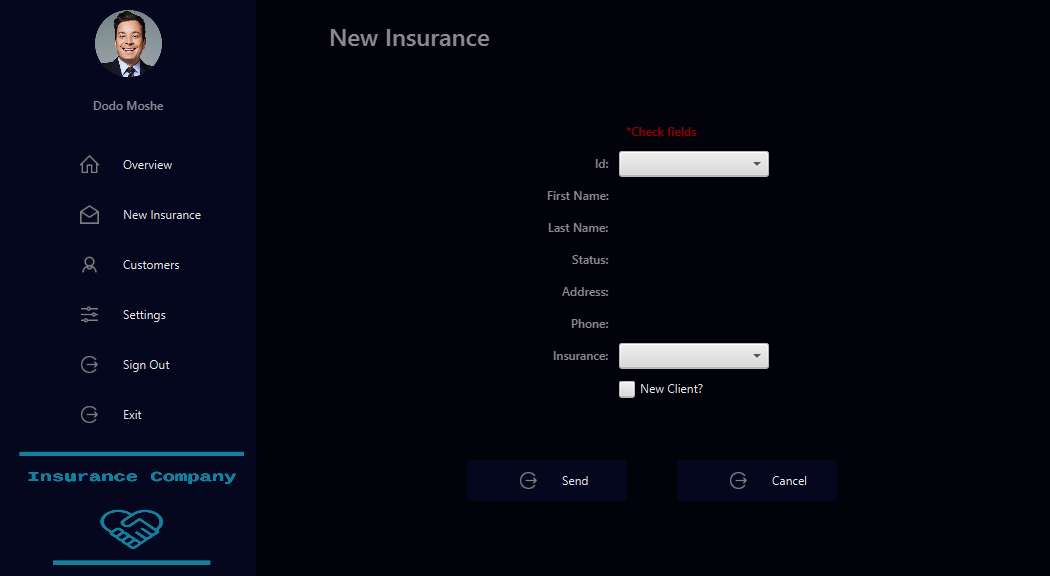
The Salesman have two options: one is to add a new insurance to an exist client, second is to create a new client with a new insurance. He cannot create both of them together.

New insurance only can be done when the salesman chooses an exist client from the bombo box (label -> Id). Then, need to choose the insurance that the client asked for. If the client agrees the salesman will save the new insurance into the database by clicking Send. The new insurance will add to the database. Any new insurance gets a default status “Pending”. Therefore, the salesman will need to enter to the Customer menu and there to approved or disapproved as the company decides.



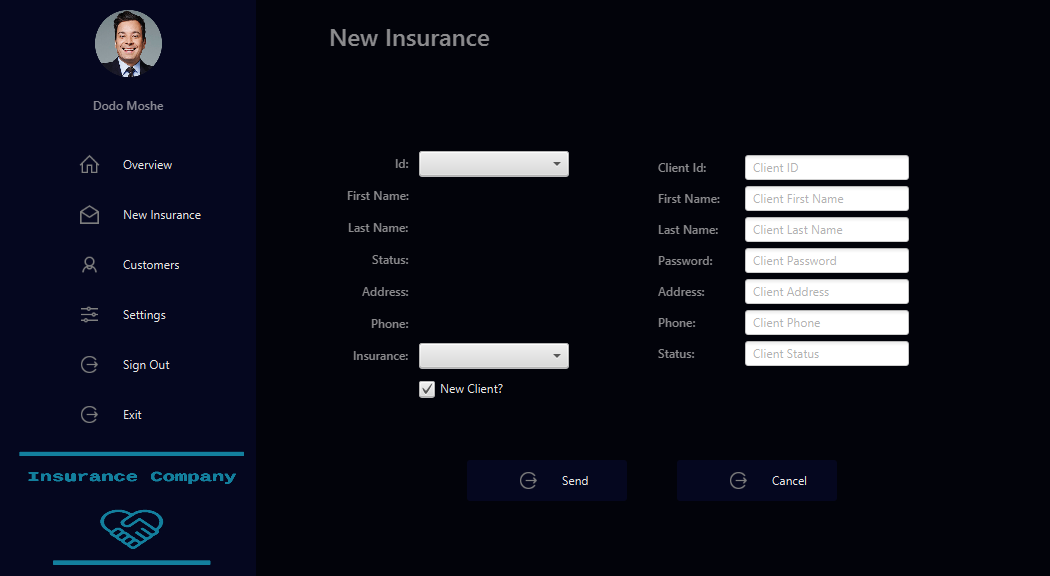
#### Fail

If the salesman will try to send an insurance without filling all the requirement fields, it will failed and will notify the salesman for missing fields.



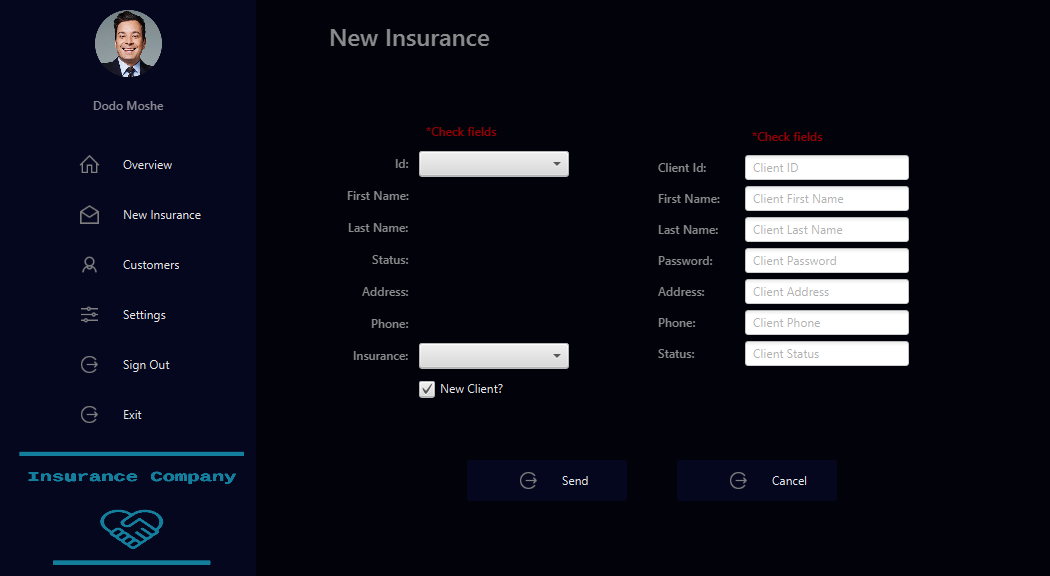
### New Insurance & Client

When the salesman wants to create a new client with insurance, the salesman needs to click the choose box (New Client?) then new client fields will appears and will need to fill the requirement fields and the insurance type. Only then clicking the button Send.



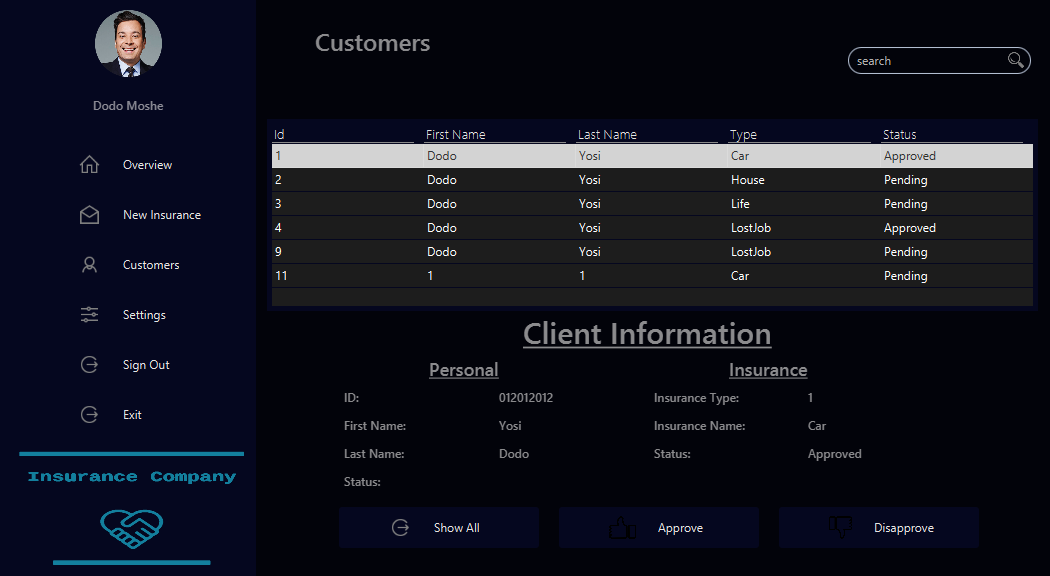
#### Fail

For any reason the salesman will not fill the fields the application will notify the user.

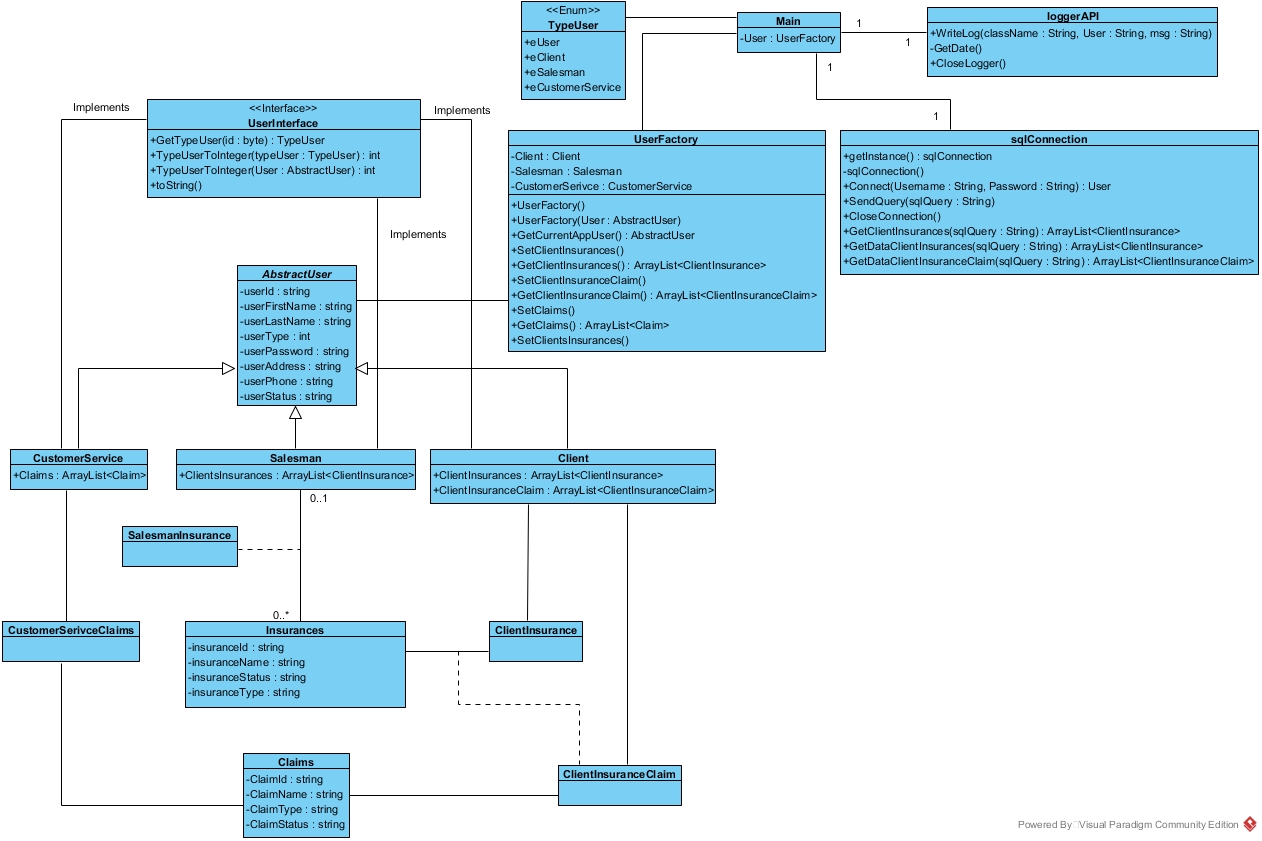


### Customer

This section the salesman will need to decide if the insurance approved or disapproved. By clicking the button, the insurance status will be updated in the table and in the database.



# UML Diagrams



# Design Patterns

## MCV

The application is using JavaFX therefor MCV is must in the development stages. The application is split to three parts. First part the is controllers where all the functionality occurs and the communication between the Model and the View.

The second part is the View that contains all the .fxml files. Each file is a view that the user of the application sees and interact with.

The third part is the Model where whole classes that contains the information of an object.

## Singleton

In the application we are using two singleton Logger and SQL communication. Each class has only one instance in the application.

To use the loogerAPI we return an instance and we cannot create another one. Therefor, all the communication between the application and the loggerAPI asks for the instance and then using the functions that API contains.

To use SQL is the same as the loggerAPI. We get an instance of the object and then we can use his functions. We cannot create another instance of the object.

## Façade

When we are using JDBC, we are using Façade design pattern.

## Factory

The users have the same default attributes as Id, Name, Address etc. But each user has a different attribute that added later. Moreover, each user uses the same functions.

Therefor, there is a UserFactory class that creates the specific user and when the user needs any information this class will return the information from the specific user.

UserInterface class has the default functions that all users use and functions that each user will need to create differently than others.

AbstractUser class has all the attributes that are default for all users like Id, Name, Address etc. Therefor, there is no need to declare and create thus function in the extended classes.

Client, Salesman and CustomerSerive are the extended class of the Abstract class that implemts the UserInterface.