# Group 8 Project ITR3

31/03/2024

Jiaye Chen, Joshua Zuker, Tejae Whitley, Kyle Williamson, Cyrus Yang, and Julianna Silva

LE/EECS 2311 X | Software Development Project
York University

# Table of Contents

## Our Ideas

### Time Consuming but Unique:

- ~~Game~~
    - ~~2D top down game (Zelda, Metal Gear)~~

### Simple, but Effective:

- ~~Music player~~
    - ~~Play music~~
- ~~Recipe Management System~~
    - ~~Input recipes, recipe info, categorise recipes, …~~
- ~~Soundboard~~

### Balanced:

- Claims Database
    - Input claim number, get information on claim, and edit claim information
- ~~Dental Patient Record~~
    - ~~Contains different patients, the condition of their teeth, their insurance status~~
- ~~Movie Review (Database?)~~
    - ~~Search Movies, Leave Reviews,~~
- ~~Personal Finance App~~
    - ~~Takes real financial info, UI interface, Shows visual data~~

## Vision Statement

The claims database is used to hold the personal information of an insured individual, as well as the information regarding their claim. This can include whether the vehicle has been totaled, who was at fault for the accident, and the payment information. The database will also have a toolbar which contains a variety of utilities an advisor can use. This project will be done from scratch.

This database also contains a ‘notes’ section of the toolbar that will allow the claims advisor to add or remove information pertaining to the claim and the client. In order to access a particular claim, the advisor must use a corresponding claim number.

If a client requires a claim be opened, the advisor can open a new claim by going to the ‘open claim’ section of the toolbar. This will allow them to input information regarding the claim that they have at the moment, such as the make and model of the client’s vehicle, the time of the accident, etc. As time goes on this information can be updated.

When the client has been satisfied, and has been paid, the advisor may close the claim. This can be done by going to the ‘close claim’ section of the toolbar. The advisor must note the conditions in which the claim closed, such as how the client felt, and will be able to close the claim afterwards.

This database will be an improvement to the current system, as it will be more simplified compared to the one in use. It will also include a more modern UI. This system will be successful if users of the older system prefer it to the current version.

Our customer will be a claims advisor who specialises in bodily injury claims.

## User Stories

- ### Big User Stories
- **User 1(Customer): As a customer, I want a comprehensive and user-friendly online claims processing system that allows me to quickly and conveniently file a claim at any time without the need to visit an office or speak with an agent. This system should support the upload of pictures or video footage of my damaged vehicle, providing me with the option to submit additional evidence for my claim.**

  **Additionally, I would like the ability to track the progress of my claim, allowing me to monitor each step of the process in real-time through a clear and modern dashboard upon logging in. The login experience and account creation process should be modern, easy, and straightforward,**

**ensuring that I can sign up for an account without unnecessary complications or excessive information requirements.**

**Furthermore, the system should enable me to easily edit my account information, such as my phone number and email, to ensure accuracy. In case I decide not to use the service anymore, I would like the option to delete my account hassle-free. Overall, the goal is to provide a seamless, efficient, and user-centric online claims experience.**

- **User 2(Insurer): As an insurer, I need an intuitively designed dashboard within AutoClaim that consolidates all submitted claims, allowing me to efficiently organize my workload. This dashboard should provide access to related photos, videos, and documents, offering additional context for accurate claim assessment. Furthermore, I want the ability to update claim statuses, facilitating timely notifications to customers.**

  **To begin claims adjustments, I seek a straightforward account signup process within autoclaims. Additionally, I desire access to past claims history and records for future reference, ensuring a comprehensive overview of the insurer's interaction with claims over time.**

  **User 3(Administrator): As an administrator, I need comprehensive control over user accounts and permissions to ensure that staff members have appropriate access to resources and information necessary for their roles. This includes:**

  **Implementing password masking on the login page to safeguard users' important information.**
  **Facilitating easy user type switching on the login page for seamless access.**
  **Accessing and managing all customer, advisor, claims, and policies data for effective administration.**
  **It's essential to maintain strict control over user roles within the AutoClaim platform. The Administrative Code feature provides a robust security measure by requiring a unique administrative code during the registration process. This code serves as a safeguard against unauthorized access and**

**ensures that only designated individuals can register as advisors or administrators.**

- ## User Stories

**User 1(Customer):**

- As a customer, in order to quickly file a claim at any time without having to visit an office or speak with an agent, I would like to do the whole process remotely. (**Cyrus)**
- As a customer, I would like to be able to track the progress of my claim so I can see what step of the process my claim is at. **(Julianna)**
- As a customer, I would like to have a modern and easy login experience. **(Klye)**
- As a customer, I would like to sign up for an account easily, to avoid too much complicated and useless information. (**Tejae**)
- As a customer, I expect a clear and modern dashboard that provides a real-time overview of my claim when I'm logged in. (**Tejae & Kyle**)
- As a customer, I would like to edit my account information, like phone number, and email. **(Jaye)**
- As a customer, I'd like to delete the account when I'm not using it. **(Jaye)**

**User 2(Insurer):**

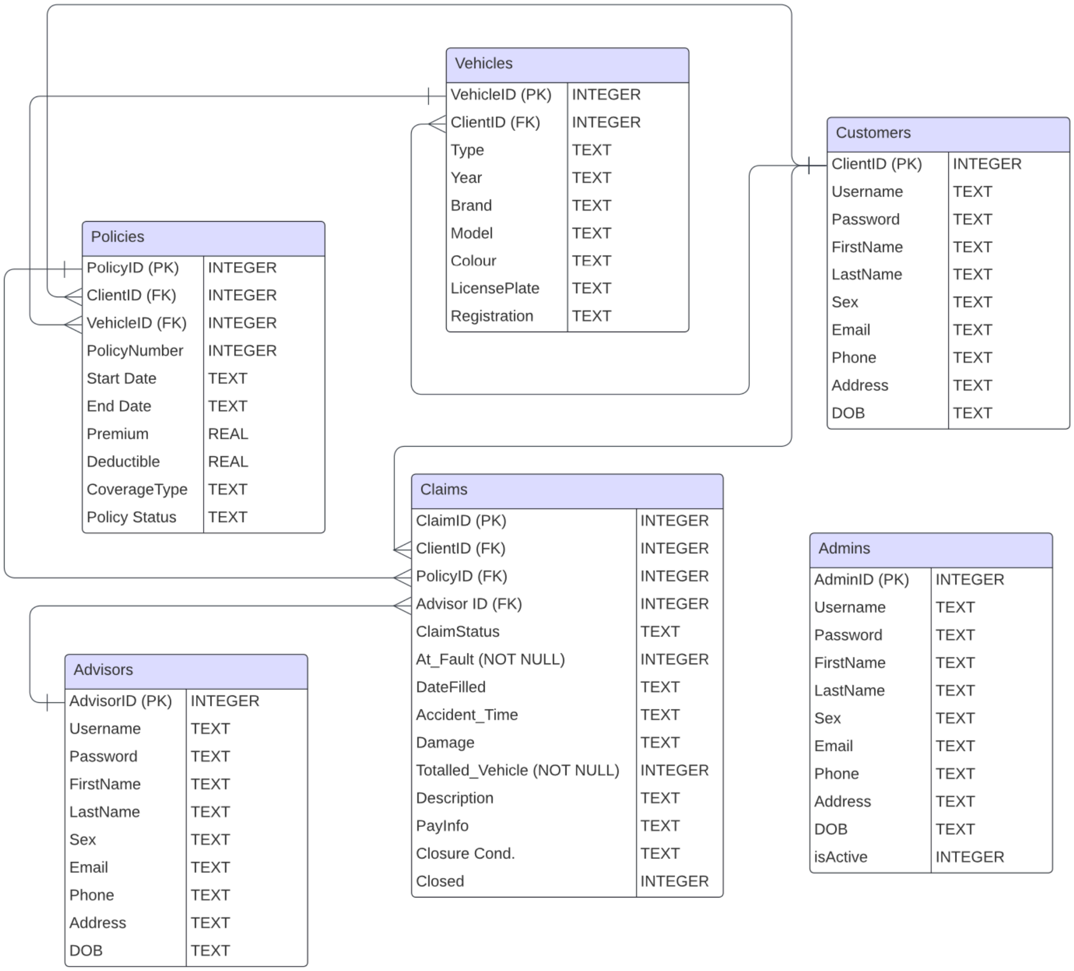
- As an insurer, I want to be able to access all the submitted claims through an intuitively designed dashboard, so I am able to organize my workload well. (**Joshua**)
- As an insurer, I would like to be able to view all the related photos, videos, and documents, so that I can have more context to assess the claim.**(Kyle)**
- As the insurer, I want to be able to update the status of the claim, so that I can notify the customer. (**Joshua)**
- As the insurer, I want to be able to view all claims in a list.(**Joshua)**
- As an insurer, I'd like to access past claims history and records for future references. **(Julianna)**

**User 3(Administrator):**

- As an administrator, I want to control user accounts and permissions to make sure staff members have access to the right resources and information for their jobs. **(Tejae)**

- As an administrator, I want my users' important information, such as passwords, will not be leaked, so I hope to use * instead of password characters on the login page. (**Cyrus)**
- As an administrator, I want to easily switch user types on the login page so that every user can log in correctly. **(Kyle & Jaye)**
- As an administrator, I want accessibility to view all customers, advisors, claims, and policies. **(Julianna)**
- As an administrator, I need an administrative code to ensure customers could not register as an advisor or an admin.**(Jaye & Cyrus)**
- As an administrator, I would like to add an age restriction when people want to sign up for an account to avoid children from using our system. (**Cyrus)**

# The Planning Map



| Vehicles | |
|---|---|
| VehicleID (PK) | INTEGER |
| ClientID (FK) | INTEGER |
| Type | TEXT |
| Year | TEXT |
| Brand | TEXT |
| Model | TEXT |
| Colour | TEXT |
| LicensePlate | TEXT |
| Registration | TEXT |

| Customers | |
|---|---|
| ClientID (PK) | INTEGER |
| Username | TEXT |
| Password | TEXT |
| FirstName | TEXT |
| LastName | TEXT |
| Sex | TEXT |
| Email | TEXT |
| Phone | TEXT |
| Address | TEXT |
| DOB | TEXT |

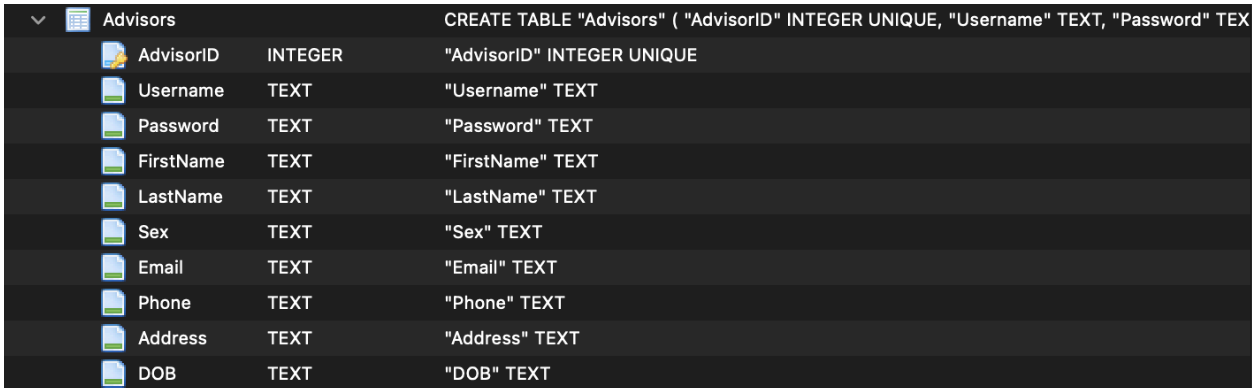
| Policies | |
|---|---|
| PolicyID (PK) | INTEGER |
| ClientID (FK) | INTEGER |
| VehicleID (FK) | INTEGER |
| PolicyNumber | INTEGER |
| Start Date | TEXT |
| End Date | TEXT |
| Premium | REAL |
| Deductible | REAL |
| CoverageType | TEXT |
| Policy Status | TEXT |

| Claims | |
|---|---|
| ClaimID (PK) | INTEGER |
| ClientID (FK) | INTEGER |
| PolicyID (FK) | INTEGER |
| Advisor ID (FK) | INTEGER |
| ClaimStatus | TEXT |
| At_Fault (NOT NULL) | INTEGER |
| DateFilled | TEXT |
| Accident_Time | TEXT |
| Damage | TEXT |
| Totalled_Vehicle (NOT NULL) | INTEGER |
| Description | TEXT |
| PayInfo | TEXT |
| Closure Cond. | TEXT |
| Closed | INTEGER |

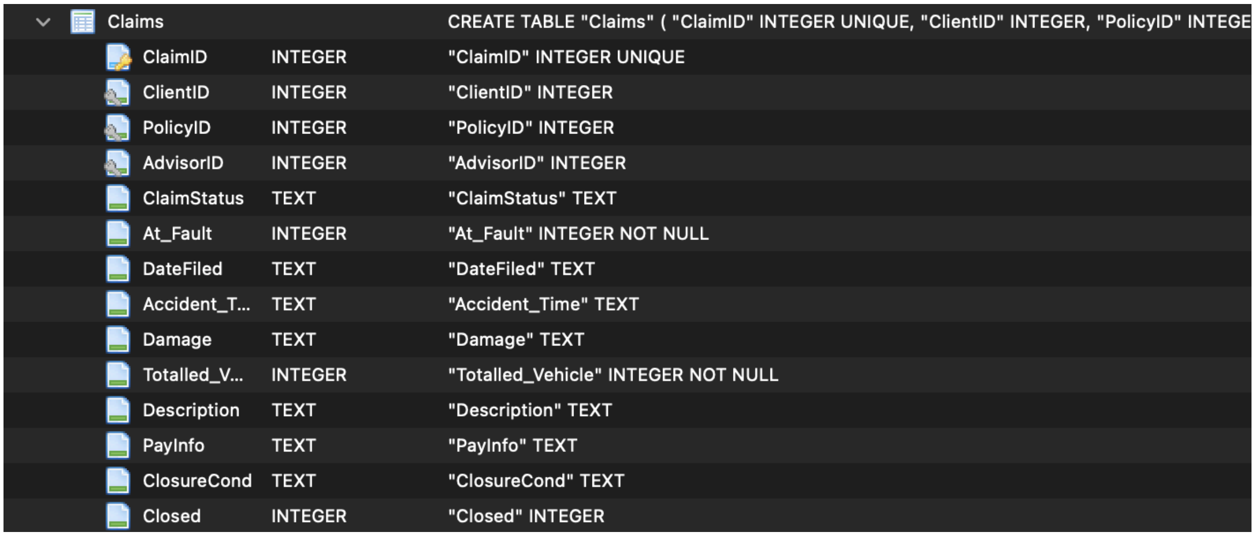
| Admins | |
|---|---|
| AdminID (PK) | INTEGER |
| Username | TEXT |
| Password | TEXT |
| FirstName | TEXT |
| LastName | TEXT |
| Sex | TEXT |
| Email | TEXT |
| Phone | TEXT |
| Address | TEXT |
| DOB | TEXT |
| isActive | INTEGER |

| Advisors | |
|---|---|
| AdvisorID (PK) | INTEGER |
| Username | TEXT |
| Password | TEXT |
| FirstName | TEXT |
| LastName | TEXT |
| Sex | TEXT |
| Email | TEXT |
| Phone | TEXT |
| Address | TEXT |
| DOB | TEXT |

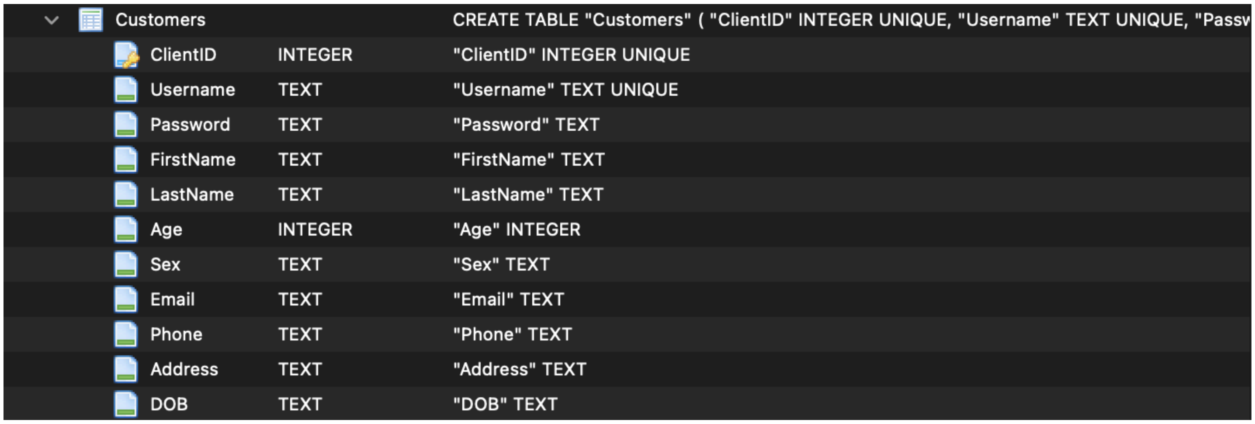
- ## Database schema



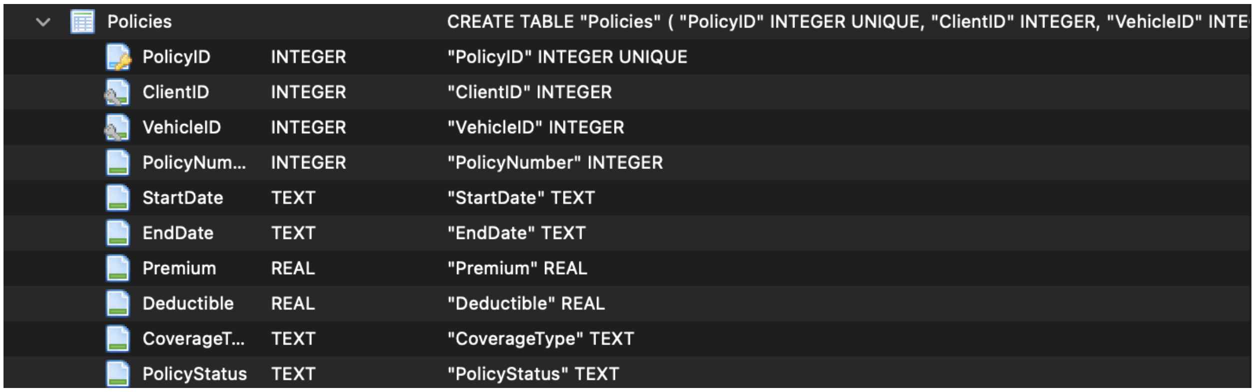
| | | |
|---|---|---|
| Advisors | | CREATE TABLE "Advisors" ( "AdvisorID" INTEGER UNIQUE, "Username" TEXT, "Password" TEX |
| AdvisorID | INTEGER | "AdvisorID" INTEGER UNIQUE |
| Username | TEXT | "Username" TEXT |
| Password | TEXT | "Password" TEXT |
| FirstName | TEXT | "FirstName" TEXT |
| LastName | TEXT | "LastName" TEXT |
| Sex | TEXT | "Sex" TEXT |
| Email | TEXT | "Email" TEXT |
| Phone | TEXT | "Phone" TEXT |
| Address | TEXT | "Address" TEXT |
| DOB | TEXT | "DOB" TEXT |



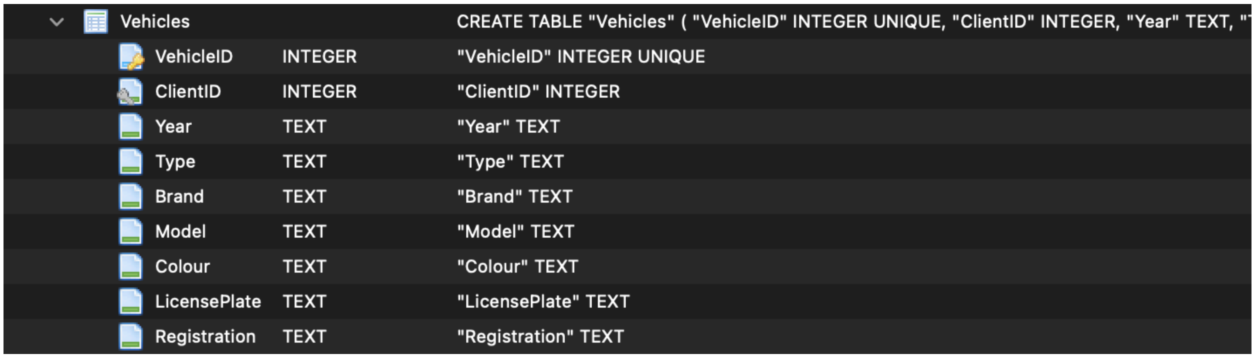
| | | |
|---|---|---|
| Claims | | CREATE TABLE "Claims" ( "ClaimID" INTEGER UNIQUE, "ClientID" INTEGER, "PolicyID" INTEGE |
| ClaimID | INTEGER | "ClaimID" INTEGER UNIQUE |
| ClientID | INTEGER | "ClientID" INTEGER |
| PolicyID | INTEGER | "PolicyID" INTEGER |
| AdvisorID | INTEGER | "AdvisorID" INTEGER |
| ClaimStatus | TEXT | "ClaimStatus" TEXT |
| At_Fault | INTEGER | "At_Fault" INTEGER NOT NULL |
| DateFiled | TEXT | "DateFiled" TEXT |
| Accident_T... | TEXT | "Accident_Time" TEXT |
| Damage | TEXT | "Damage" TEXT |
| Totalled_V... | INTEGER | "Totalled_Vehicle" INTEGER NOT NULL |
| Description | TEXT | "Description" TEXT |
| PayInfo | TEXT | "PayInfo" TEXT |
| ClosureCond | TEXT | "ClosureCond" TEXT |
| Closed | INTEGER | "Closed" INTEGER |



| | | |
|---|---|---|
| Customers | | CREATE TABLE "Customers" ( "ClientID" INTEGER UNIQUE, "Username" TEXT UNIQUE, "Passw |
| ClientID | INTEGER | "ClientID" INTEGER UNIQUE |
| Username | TEXT | "Username" TEXT UNIQUE |
| Password | TEXT | "Password" TEXT |
| FirstName | TEXT | "FirstName" TEXT |
| LastName | TEXT | "LastName" TEXT |
| Age | INTEGER | "Age" INTEGER |
| Sex | TEXT | "Sex" TEXT |
| Email | TEXT | "Email" TEXT |
| Phone | TEXT | "Phone" TEXT |
| Address | TEXT | "Address" TEXT |
| DOB | TEXT | "DOB" TEXT |

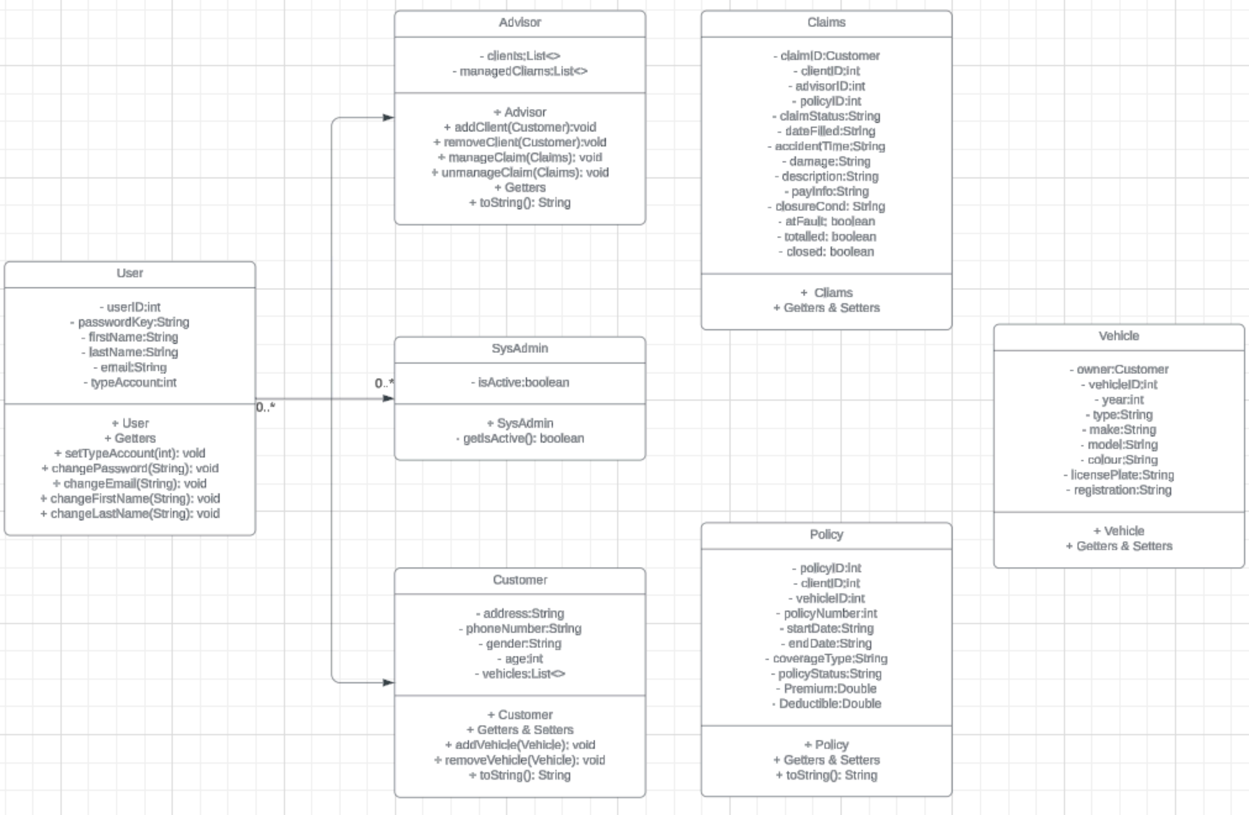


| | | |
|---|---|---|
| Policies | | CREATE TABLE "Policies" ( "PolicyID" INTEGER UNIQUE, "ClientID" INTEGER, "VehicleID" INTE |
| PolicyID | INTEGER | "PolicyID" INTEGER UNIQUE |
| ClientID | INTEGER | "ClientID" INTEGER |
| VehicleID | INTEGER | "VehicleID" INTEGER |
| PolicyNum... | INTEGER | "PolicyNumber" INTEGER |
| StartDate | TEXT | "StartDate" TEXT |
| EndDate | TEXT | "EndDate" TEXT |
| Premium | REAL | "Premium" REAL |
| Deductible | REAL | "Deductible" REAL |
| CoverageT... | TEXT | "CoverageType" TEXT |
| PolicyStatus | TEXT | "PolicyStatus" TEXT |



| | | |
|---|---|---|
| Vehicles | | CREATE TABLE "Vehicles" ( "VehicleID" INTEGER UNIQUE, "ClientID" INTEGER, "Year" TEXT, " |
| VehicleID | INTEGER | "VehicleID" INTEGER UNIQUE |
| ClientID | INTEGER | "ClientID" INTEGER |
| Year | TEXT | "Year" TEXT |
| Type | TEXT | "Type" TEXT |
| Brand | TEXT | "Brand" TEXT |
| Model | TEXT | "Model" TEXT |
| Colour | TEXT | "Colour" TEXT |
| LicensePlate | TEXT | "LicensePlate" TEXT |
| Registration | TEXT | "Registration" TEXT |

- Java Classes


Advisor
- clients:List<>
- managedCliams:List<>
+ Advisor
+ addClient(Customer):void
+ removeClient(Customer):void
+ manageClaim(Claims): void
+ unmanageClaim(Claims): void
+ Getters
+ toString(): String
Claims
- claimID:Customer
- clientID:int
- advisorID:int
- policyID:int
- claimStatus:String
- dateFilled:String
- accidentTime:String
- damage:String
- description:String
- payInfo:String
- closureCond: String
- atFault: boolean
- totalled: boolean
- closed: boolean
+ Cliams
+ Getters & Setters
User
- userID:int
- passwordKey:String
- firstName:String
- lastName:String
- email:String
- typeAccount:int
+ User
+ Getters
+ setTypeAccount(int): void
+ changePassword(String): void
+ changeEmail(String): void
+ changeFirstName(String): void
+ changeLastName(String): void
0..*
0..*
SysAdmin
- isActive:boolean
+ SysAdmin
- getIsActive(): boolean
Vehicle
- owner:Customer
- vehicleID:int
- year:int
- type:String
- make:String
- model:String
- colour:String
- licensePlate:String
- registration:String
+ Vehicle
+ Getters & Setters
Customer
- address:String
- phoneNumber:String
- gender:String
- age:int
- vehicles:List<>
+ Customer
+ Getters & Setters
+ addVehicle(Vehicle): void
+ removeVehicle(Vehicle): void
+ toString(): String
Policy
- policyID:int
- clientID:int
- vehicleID:int
- policyNumber:int
- startDate:String
- endDate:String
- coverageType:String
- policyStatus:String
- Premium:Double
- Deductible:Double
+ Policy
+ Getters & Setters
+ toString(): String