Group 8 Project ITR0

29/01/2024

—

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: As an insurance user/customer, I'd like to handle my insurance and claims digitally using a modern system, so the process of managing my insurance can be done easily and efficiently in a user-friendly manner.**

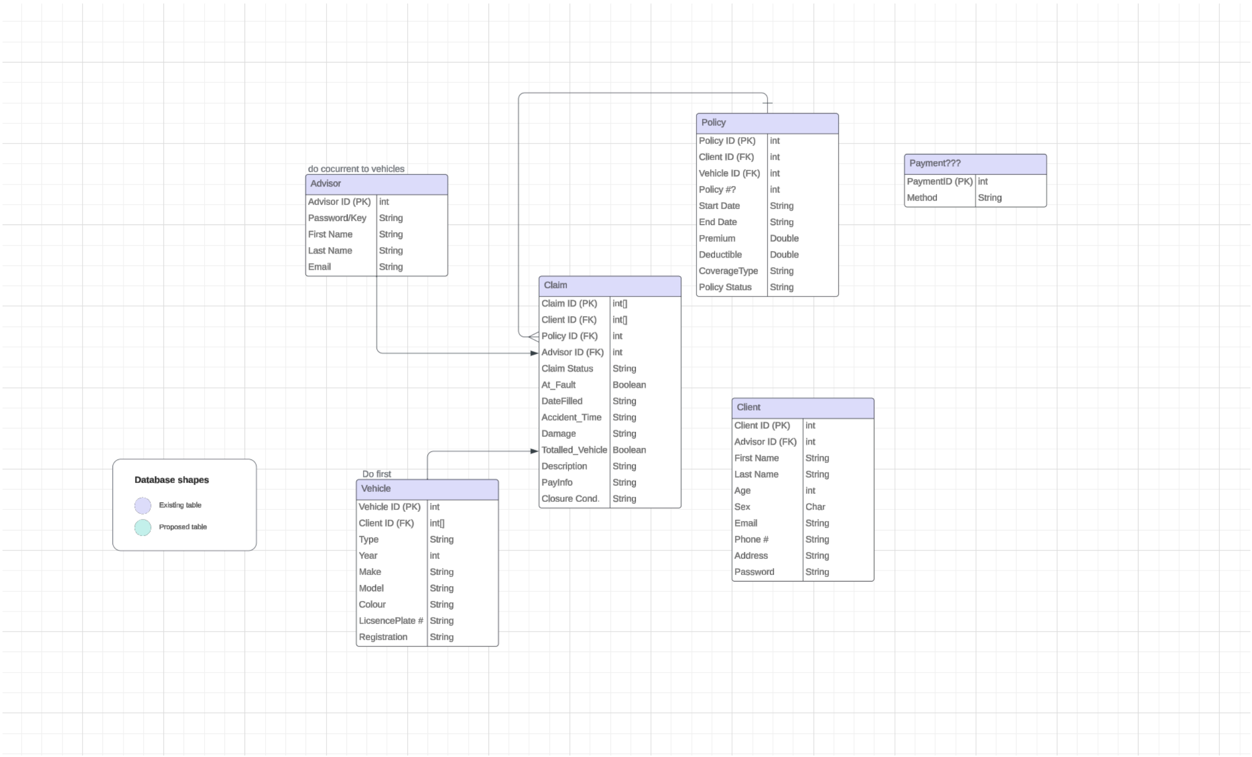
- **User 2: As an insurance executive, I want to enhance customer experience through the use of some customer-centric focused/user experience enhancing systems to improve transparency and communication with our clients and what we produce**

- **User 3: As an insurance advisor, I want a system that manages and maintains our internal business flow, including claims advisors and**

actuarial processes, and that is up to date with regulatory standards and company updates.

- User Stories
-

The Planning Map



do cocurrent to vehicles

Advisor	
Advisor ID (PK)	int
Password/Key	String
First Name	String
Last Name	String
Email	String

Policy	
Policy ID (PK)	int
Client ID (FK)	int
Vehicle ID (FK)	int
Policy #?	int
Start Date	String
End Date	String
Premium	Double
Deductible	Double
CoverageType	String
Policy Status	String

Payment???	
PaymentID (PK)	int
Method	String

Claim	
Claim ID (PK)	int[]
Client ID (FK)	int[]
Policy ID (FK)	int
Advisor ID (FK)	int
Claim Status	String
At_Fault	Boolean
DateFilled	String
Accident_Time	String
Damage	String
Totalled_Vehicle	Boolean
Description	String
PayInfo	String
Closure Cond.	String

Client	
Client ID (PK)	int
Advisor ID (FK)	int
First Name	String
Last Name	String
Age	int
Sex	Char
Email	String
Phone #	String
Address	String
Password	String

Do first

Vehicle	
Vehicle ID (PK)	int
Client ID (FK)	int[]
Type	String
Year	int
Make	String
Model	String
Colour	String
LicsencePlate #	String
Registration	String

- Database schema

Clients	
client_ID (primary key)	
advisor_ID (foreign key)	// does every client have an assigned advisor?

vehicle_ID (foreign key)	// not sure about this one bc maybe clients can have multiple vehicles // Perhaps we can allow clients to add more vehicles
first_name	
last_name	
phone_number	
email	
address	

Vehicles	
vehicle_ID (primary key)	
client_ID (foreign key)	
vehicle_type	// what type of vehicles? // Refers to shape of car e.g sedan
vehicle_make	// Car manufacturers e.g Toyota
vehicle_model	// Specific car model from manufacturer e.g Corolla
vehicle_colour	

Advisors	
advisor_ID (primary key)	
first_name	
last_name	

Claims	
claim_ID (primary key)	
client_ID (foreign key)	
advisor_ID (foreign key)	
claim_status	// What kind of status conditions?
at_fault	//true, false, 5050
accident_time	// date and time
totaled_vehicle	// true or false
payment_info	// should clients have payment info?
closure_feedback	

Notes	
note_ID (primary key)	
claim_ID (foreign key)	
advisor_ID (foreign key)	
note_content	
note_timestamp	

- ## Java Classes

Claim (who files the claim? Is it the advisor after the client requests? Or is it the client alone?)

-

ClaimDAO

- Handles connection between Claim and the database
- Claims can be created, edited or closed through this

Who registers a new client? Is it the advisor?

Client

- Add a counter for when a client is registered the number is incremented and assigned as clientID (if it is random then there is a chance that the IDs will be duplicate)
- firstName, lastName, phoneNumber, email…
- Getters and setters for each one
- One advisor object from Advisor

ClientDAO

- Handles connection between Client and the database
- Registers clients, returns specific client by ID, etc

Vehicle

- Type, make, model, year etc
- Has an owner of type Client

VehicleDAO

- Handles connection between Vehicle and database
- It should be able to search an specific vehicle through its owner ID (clientID)

Advisor

-

  Note (only an advisor can create a note)

  - Note is attached to an object of type Claim