Wayne Zhu
Part 4 Note

Fist of all, the ML algorithm of Part 3 has been updated on BrightSpace. The original version's R file turns out to be empty unfortunately.

All the code for this project can be found here:
https://github.com/JinweiZhu/DatabaseFinalProjectInsurance/tree/main/GetCovidInsuranceQuote

The use case of the frontend has already been explained in the previous part of the project. But I will summarize it here and highlight a few points because I added the backend database to my flow chart. You can see the picture of workflow on next page.

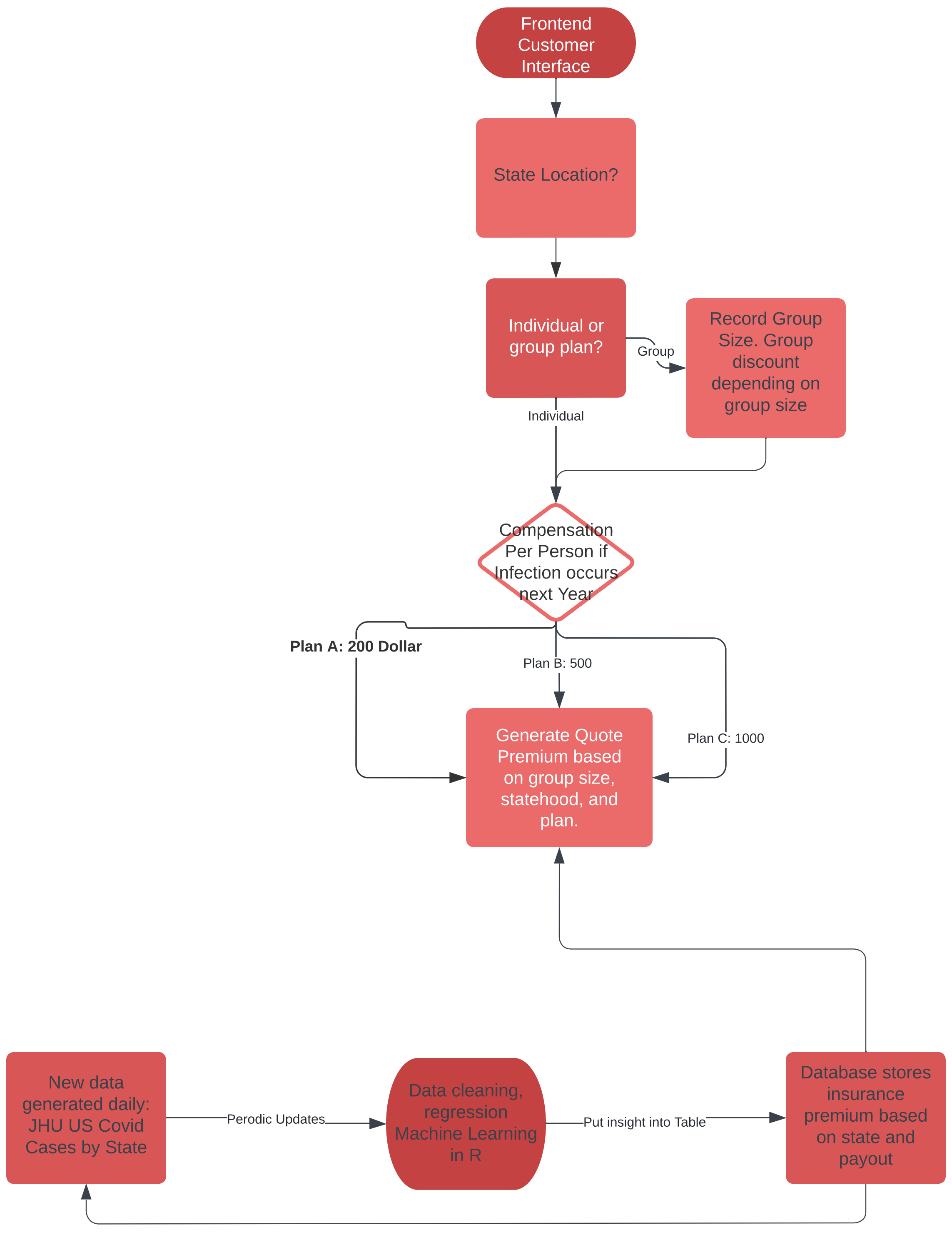
1. Customer will input their state, and the number of people they want to purchase the plan for, and how much payout per person infected they expect.

2. Then, recall from last part that ProductPlan based on statehood and payout amount is already stored in the database. So we can just query the premium from ProductPlan.

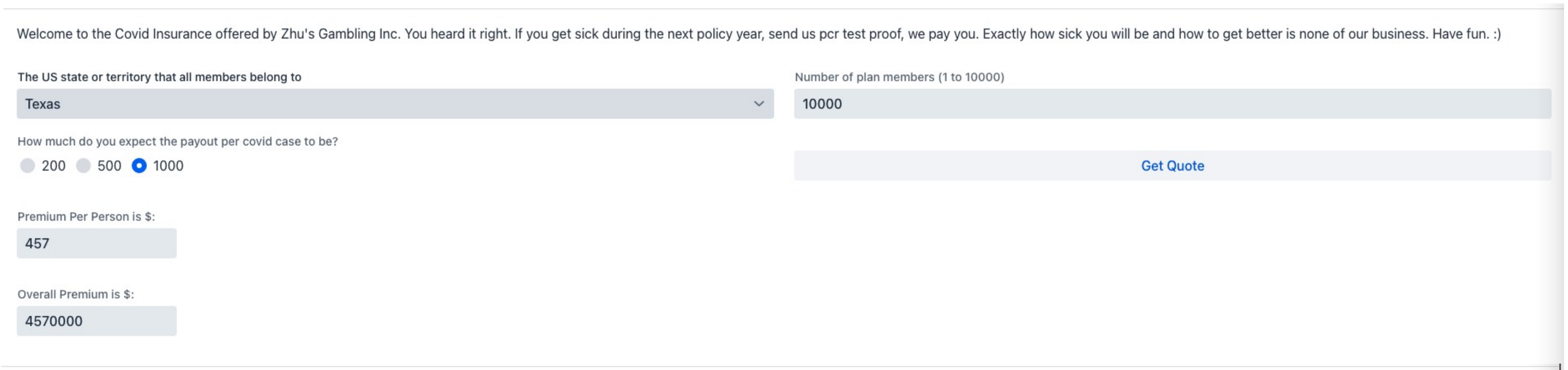
3. Then, we calculate ActualPremiumPerPerson based on number of memebers' discount.

4. We then output the total cost to the user as well. This will just be ActualPremiumPerPerson multiplied by the NumOfMemebrs.

Business Use Case Diagram for Accident Insurance Company for Covid-19 Infection


Frontend Customer Interface
State Location?
Individual or group plan?
Group
Record Group Size. Group discount depending on group size
Individual
Compensation Per Person if Infection occurs next Year
Plan A: 200 Dollar
Plan B: 500
Plan C: 1000
Generate Quote Premium based on group size, statehood, and plan.
New data generated daily: JHU US Covid Cases by State
Perodic Updates
Data cleaning, regression Machine Learning in R
Put insight into Table
Database stores insurance premium based on state and payout

I use Spring, Vaadin, Java, JDBC, and MySQL to implement the application that connects the backend to the front end. The customer enter, state name, number of members, and expected payout. Premium will be pulled from database and printed. See screenshot below.



Welcome to the Covid Insurance offered by Zhu's Gambling Inc. You heard it right. If you get sick during the next policy year, send us pcr test proof, we pay you. Exactly how sick you will be and how to get better is none of our business. Have fun. :)

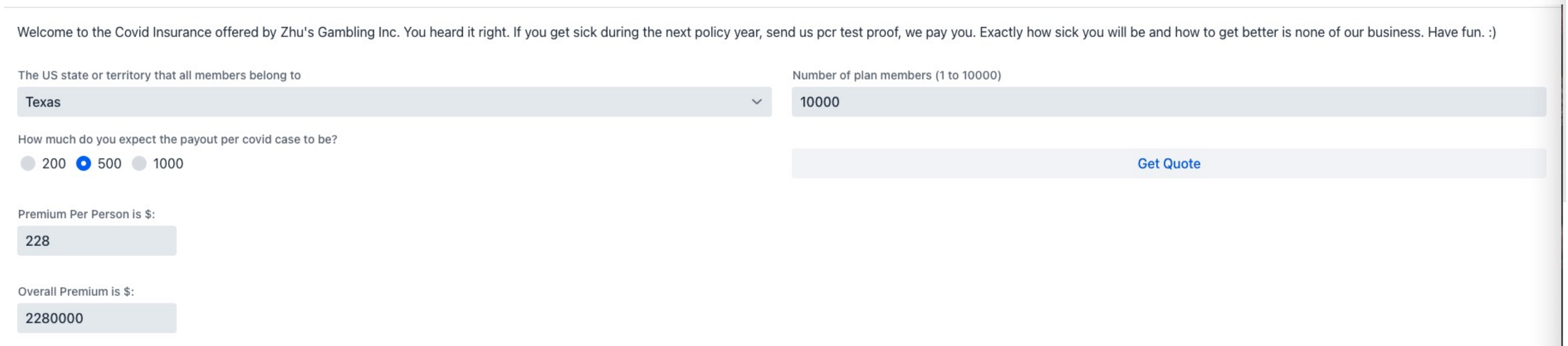
The US state or territory that all members belong to: Texas

Number of plan members (1 to 10000): 10000

How much do you expect the payout per covid case to be? 200 500 [✓ 1000]

Premium Per Person is $: 457

Overall Premium is $: 4570000



Welcome to the Covid Insurance offered by Zhu's Gambling Inc. You heard it right. If you get sick during the next policy year, send us pcr test proof, we pay you. Exactly how sick you will be and how to get better is none of our business. Have fun. :)

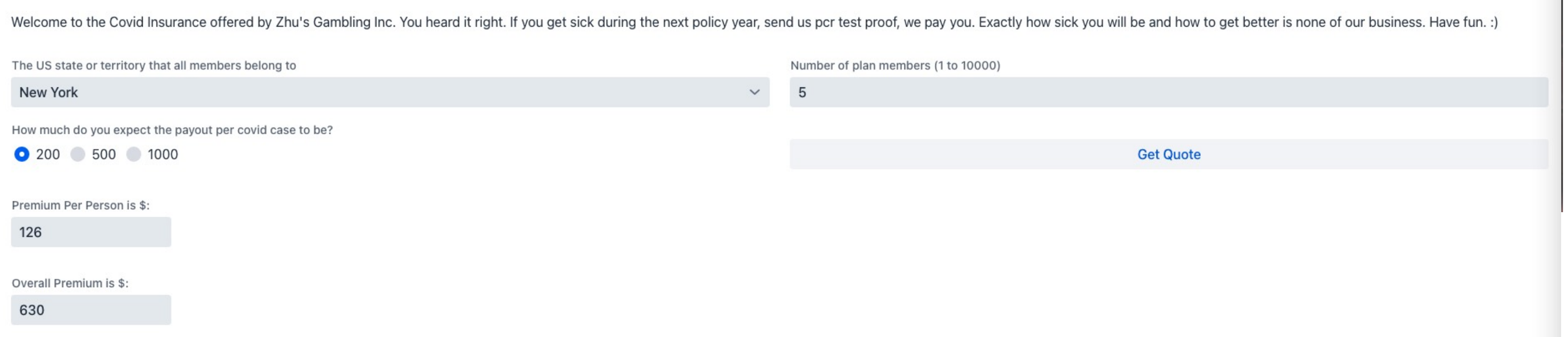
The US state or territory that all members belong to: Texas

Number of plan members (1 to 10000): 10000

How much do you expect the payout per covid case to be? 200 [✓ 500] 1000

Premium Per Person is $: 228

Overall Premium is $: 2280000



Welcome to the Covid Insurance offered by Zhu's Gambling Inc. You heard it right. If you get sick during the next policy year, send us pcr test proof, we pay you. Exactly how sick you will be and how to get better is none of our business. Have fun. :)

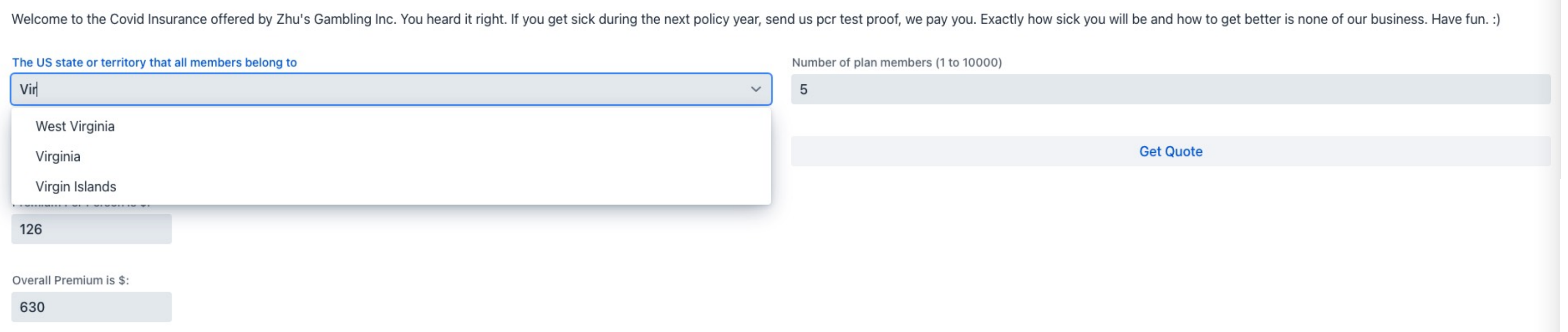
The US state or territory that all members belong to: New York

Number of plan members (1 to 10000): 5

How much do you expect the payout per covid case to be? [✓ 200] 500 1000

Premium Per Person is $: 126

Overall Premium is $: 630



Welcome to the Covid Insurance offered by Zhu's Gambling Inc. You heard it right. If you get sick during the next policy year, send us pcr test proof, we pay you. Exactly how sick you will be and how to get better is none of our business. Have fun. :)

The US state or territory that all members belong to: Vir

Number of plan members (1 to 10000): 5

126

Overall Premium is $: 630

The data analytics for insight is done in R, and we can easily export our R output and import it into MySQL database. This can happen regularly and it won't affect the frontend.

We use JDBC as database connectivity framework and MySQL API in Java's Spring framework to send query to MySQL and send back response to my application. That part of the code is in ApplicationToDatabase.java in my project.

For the frontend, we use vaadin UI interface to call ApplicationToDatabase's method. Vaadin allows UI to be designed completely in Java.

This ensures the flow from frontend UI entry to actual query output is automatic and error free.

My ProductPlan table is denormalized such that ProductPlanId and PayoutPerPerson is partially dependent. This way, it's clear that we have 3 types of plans $200 plan 1, $500 plan 2 and $1000 plan 3.

It's also possible to denormalize further. For convenience purposes, it's possible to store total premium or the corresponding discount for group in IndividualOrGroupAccount in the future. We normalize it earlier by removing those fields to control for duplication as well.

For query optimization, MySQL uses external sorting to do Group by or order by. However, we are only doing select queries. So I change my search query to use primary index query, which MySQL already implements.

There are unfortunately, duplicate FKs in my model, but it's time-consuming to remove and they don't affect the outcome.

I also used ORM to convert a row of ProductPlan into a ProductPlan object in java using MySQL API and JDBC API. Because Java is already an object oriented language, doing so allows me to write my code in Java almost entirely, and makes the code cleaner and more manageable.

Complete Reference Architecture for this project (revised from previous version). Try to keep this as concise as possible.

Principles:
IT solutions will align with business strategy.
People first, technologies second (as in Agile).
All IT assets will have identified business and IT owners (ownership is important for motivation)

Pyramids of Knowledge:
Browse through big unstructured or uncleaned data —data
Identify useful data and turned them into structured data—information
Use machine learning, like regression, to derive insight—knowledge
Use the insight to predict a profitable pricing for the insurance—wisdom

Framework:
Responsibility of the 6 domains as listed.
Business: Business research into pricing, competition and payout. They also search and identify various data sources for new variants, new vaccines and etc. and inform the tech people. This is not a health insurance, so profit maximization is the purpose.
People: Offer the best customer service and resolve conflicts between teams.
Application: Adopt the agile methodology to incorporating customer feedback into the UI and Java program.
Information: Manage the backend database and debug systemic issues. Migrate the backend infrastructure to public cloud in the future if necessary in order to support big data.
Technical: Process the data that Business identified and rerun the Machine Learning analytics regularly based on business research.
Process: Integrate the systems and test run and push to production. Brainstorming ways to bring DevOps into the other domains.

Method:
Plan: We follow Agile methodology, daily standup meeting to corroborate information. We have weekly meeting of all the domains also.
Deliver: We follow Agile methodology and introduce sprints, small patches, revision, and adjustment based on customer's needs. We ensure IT serves Business needs.
Operate: We follow DevOps principles to integrate development and production. We create an open culture where anyone feel free to speak their ideas. It's an idea meritocracy.

Governance:
The RA will be made clear to everyone on the team.
Data quality management: Technical team should keep in frequent contact with the Business domain in order to either process or remove the unstructured datasets from the database.
Prevention of data losses and leakage: Public cloud should be used and shared among the team as unstructured data get larger.
Management of data lifecycle: The relational database should be updated regularly based on the newly available big data.
Ownership: Although we work as a team, each person takes ownership of each particular story and remain accountable for code and data maintainance in the long run.
Violation of any rules will be met with initial conversation, subsequent warning, and if—unresolved or serious enough—eventual dismissal.