Database Systems Project Part IV
End-to-End Solution Integration and Data-Driven / Database Programming

Team Members:

Pu Wang	N17330908
Tianle Yang	N17553267
Yin Wang	N11864326

CATALOG

1. Project Background and Objectives

1.1 Background

The project is centered around a database initiative for an insurance company, focusing on chronic disease data to develop pricing strategies and insurance products. Initially, the project leveraged data from traditional relational databases, widely used in the insurance sector for reporting and analytics. The primary challenge was the replication of metadata across various parts of the enterprise. To address this, the project aimed to establish an Enterprise Data Architecture (EDA) to streamline the integration of existing and new data sources. This initiative involves three key components: Modeling, creating an Operational Data Store (ODS), and developing a roadmap for application integration with the ODS. The project evolved to include unstructured data collection for enhanced decision-making, leading to a hybrid data management approach that combines structured and unstructured data.

1.2 Objectives

The primary objective of this project is to develop a sophisticated database system that effectively integrates chronic disease data for the insurance company. This system aims to refine insurance pricing models and product offerings based on a deep analysis of structured and unstructured data. The end goal is to create an automated, seamless operational system that not only updates the OLTP/ODS relational database with new insights but also adapts dynamically as new data is incorporated. This includes the continuous retraining of machine learning models to ensure accurate and relevant analytics, ultimately enhancing the company's competitive edge in offering tailored insurance solutions.

1.3 Project Overview

Our Holiday Insurance's Project is an innovative end-to-end database solution focused on integrating chronic disease data into insurance pricing and product strategies. Key features include the design and implementation of data-driven business use cases, such as facilitating customer access to insurance quotes and product selection, heavily reliant on machine learning insights. The project also involves developing a program module that adapts to changes in unstructured data, necessitating periodic retraining of the ML model. Advanced database connectivity and ORM frameworks ensure seamless integration into the OLTP/ODS system. The project concludes with a comprehensive documentation of a reference architecture, encompassing business strategies, application design, data governance, and infrastructure, all adhering to the Insurance company's operational and ethical standards.

2. The Creation of An Enterprise Data Architecture （EDA）

2.1 Business Use Cases

Our project is designed for an insurance company, utilizing machine learning (ML) to process chronic disease data and build models to assist the insurance company in pricing its various insurance plans (Plan A - Premium Coverage, Plan B - Standard Coverage, Plan C - Basic Coverage). Here are several business use cases related to this project and how they apply to a data-driven workflow database application:

1) Insurance Quoting System:

Use Case Description: Customers provide personal health information and a history of chronic illnesses through surveys to obtain quotes for different insurance plans. Machine learning models analyze this data, predict the customer's risk level, and recommend suitable coverage plans and pricing based on this risk level.

Relevance and Applicability: This system can provide customized insurance quotes based on the customer's individual health condition, thereby enhancing the attractiveness and competitiveness of insurance products. The data-driven approach allows the insurance company to assess risk more accurately and set reasonable premiums.

2) Product Selection Process:

Our project for the insurance company leverages advanced machine learning (ML) algorithms to analyze detailed chronic disease data. This sophisticated analysis is key to developing a dynamic model that assists the insurance company in structuring its various insurance plans. We offer three distinct coverage options: Plan A (Premium Coverage), Plan B (Standard Coverage), and Plan C (Basic Coverage), each designed to cater to different customer needs and budgets.

Customers engage in this process by submitting their personal health information and a detailed history of any chronic illnesses through an intuitive survey. This critical data forms the foundation upon which our ML models operate.

Once the customer's data is collected and processed, our system initiates an interactive product selection journey. Here, the ML models analyze the customer's health information in conjunction with their preferences. Based on this analysis, the system generates personalized insurance plan recommendations, guiding customers to the plan that best aligns with their health needs and financial constraints.

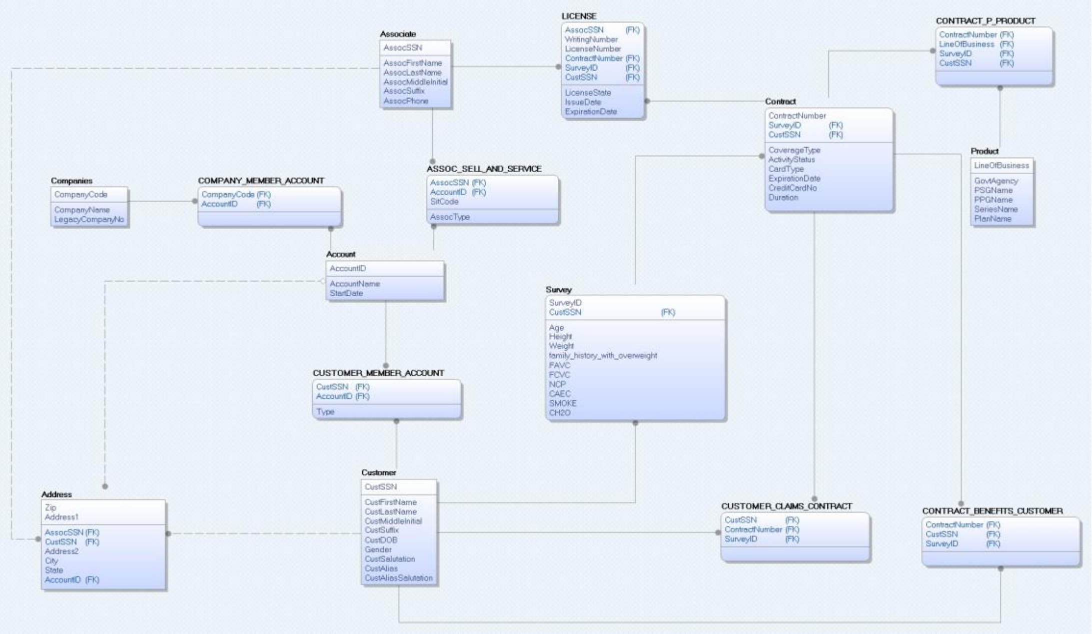
This interactive and personalized selection process significantly enhances the customer experience. It simplifies the decision-making process for customers, ensuring that they are matched with an insurance plan that is both suitable and affordable. For the insurance company, this approach not only fosters a deeper understanding of customer needs but also aids in the refinement and optimization of insurance product offerings.

In summary, this product selection process exemplifies a strategic integration of data-driven insights and customer-centric design. It not only elevates the quality of service provided by the insurance company but also ensures more precise, tailored insurance solutions for customers. This, in turn, enhances customer satisfaction and positions the insurance company competitively in the market through improved risk management and product customization.

2.2 Design Documentation:

In this section, we delve into the intricate design aspects of our data-driven workflow-based database application, specifically crafted for the insurance sector. This documentation employs modeling notation to meticulously outline and illustrate the business processes involved. The focus is on providing a comprehensive understanding of the architectural design, highlighting how various components interact and contribute to the overall functionality of the application.

2.2.1 Diagrams of System Components: Entity-Relationship Diagram (ERD)



2.2.2 Description of System Components: Entity-Relationship Diagram (ERD)

This diagram illustrates the database architecture of our insurance company, including multiple interconnected tables and their relationships. Below is a description of each table and their relationships:

A. Entities:

1) Companies: Contains attributes like CompanyCode, CompanyName, and LegacyCompanyNo.
2) Company_Member_Account: Acts as a junction table between Companies and Account, including attributes CompanyCode (Foreign Key - FK) and AccountID (FK).

3) Account: Holds details of accounts with attributes AccountID, AccountName, and StartDate.
4) Customer_Member_Account: Another junction table, this time between Account and Customer, containing CustSSN (FK), AccountID (FK), and Type.
5) Customer: Represents the customer information with attributes like CustSSN, CustFirstName, CustLastName, CustMiddleInitial, among others.
6) Address: Contains address details like Zip, Address1, Address2, City, State, and also has foreign keys AssocSSN and CustSSN indicating association with Associate and Customer, respectively.
7) Associate: Stores associate's details including AssocSSN, AssocFirstName, AssocLastName, etc.
8) Assoc_Sell_And_Service: A junction table connecting Associate with Account, including AssocSSN (FK), AccountID (FK), SICode, and AssocType.
9) License: Tied to Associate, it includes AssocSSN (FK), WingNumber, LicenseNumber, SurveyID (FK), etc.
10) Survey: Contains health-related attributes like SurveyID, Height, Weight, Family_history_with_overweight, and others.
11) Contract: Includes contract details such as ContractNumber, SurveyID (FK), CoverageType, ActivityStatus, etc.
12) Product: Holds product-related information like GovAgency, FSGName, PSGName, ServiceName, PlanName.
13) Customer_Claims_Contract: A junction table connecting Customer with Contract, including CustSSN (FK), ContractNumber (FK), and SurveyID (FK).
14) Contract_P_Product: Links Contract with Product and includes ContractNumber (FK), LineOfBusiness (FK), SurveyID (FK).
15) Contract_Benefits_Customer: Another junction table that connects Contract with Customer including ContractNumber (FK), CustSSN (FK), and SurveyID (FK).

B. Relationships:

1) Companies to Company_Member_Account: One-to-many, as one company can have multiple member accounts.
2) Account to Company_Member_Account and Customer_Member_Account: One-to-many for both, indicating that an account can be associated with multiple companies and customers.
3) Customer to Customer_Member_Account: One-to-many, a customer can have multiple member accounts.
4) Associate to Address, Assoc_Sell_And_Service, and License: One-to-many for all, an associate can have multiple addresses, sales/services, and licenses.
5) Survey to License, Contract, Customer_Claims_Contract, and Contract_P_Product: One-to-many, as a survey can be associated with multiple licenses, contracts, and products.
6) Contract to Customer_Claims_Contract, Contract_P_Product, and Contract_Benefits_Customer: One-to-many, a contract can have multiple claims, products, and benefits associated with it.

2.3 Machine Learning Integration

2.3.1 Description of the Machine Learning Model Used

Our project leverages a dynamic machine learning strategy to provide actionable insights for an insurance company's chronic disease management and pricing policies. Utilizing customer health data from surveys, we have developed models that determine risk levels and inform the pricing of three tiered insurance plans: Advanced, Standard, and Basic Protection Plans.

Further enhancing our capability, we have integrated these models with Microsoft Azure Cloud, leveraging its scalable computing resources, advanced analytics, and AI services. This integration allows for the secure processing of large datasets and ensures our solutions adhere to industry standards. Employing Azure positions our client at the vanguard of digital innovation in healthcare management, with a robust infrastructure that supports both current and future analytics demands.

The key components of the machine learning architecture include:

a. **Predictive Models:** Regression algorithms forecast potential healthcare costs and utilization.
b. **Risk Stratification:** Classification algorithms, like decision trees, segment customers into risk brackets.
c. **Anomaly Detection:** Methods such as isolation forests identify outliers in health data, crucial for risk management and fraud detection.

2.3.2 Development Process of the Data-Driven Program Module

The development of our machine learning module follows a structured process:

1) Data Preprocessing and Cleaning: Ensuring data quality by addressing missing values and outliers.
2) Exploratory Data Analysis (EDA): Detailed analysis, as seen in the notebook, where health indicators are visualized to understand their distribution and impact on chronic disease.
3) Feature Selection and Engineering: Identifying and creating predictive features that capture the nuances of health risks.
4) Model Training and Validation: Systematic training of models with robust validation techniques to ensure accuracy and reliability.
5) Hyperparameter Optimization: Employing grid search and other optimization strategies to fine-tune model performance.

2.3.3 Integration Techniques with the Overall Application

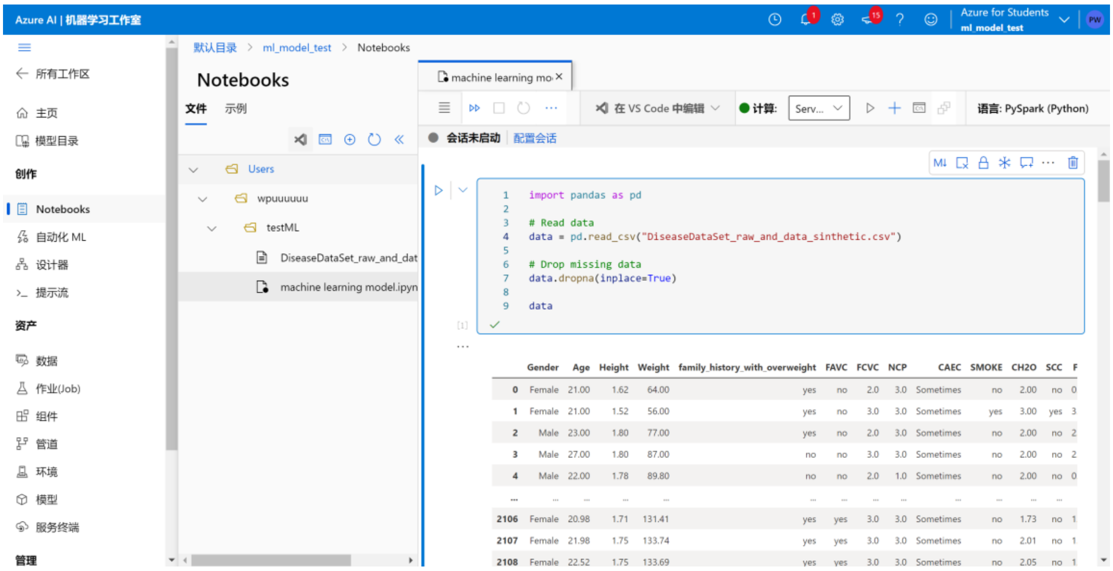
The integration of machine learning models with the insurance company's application is particularly sophisticated due to the incorporation of Microsoft Azure Cloud services.

1) Integration Process with Microsoft Azure Cloud:

The integration process was meticulously designed to leverage the full suite of Microsoft Azure Cloud services, thereby enhancing the technical capabilities of our machine learning models and ensuring seamless deployment within the insurance company's application ecosystem.

Cloud Infrastructure Setup: We initiated the process by setting up a secure and scalable cloud infrastructure on Azure, utilizing Azure Virtual Machines for computation and Azure Blob Storage for managing the vast datasets in a cost-effective manner.

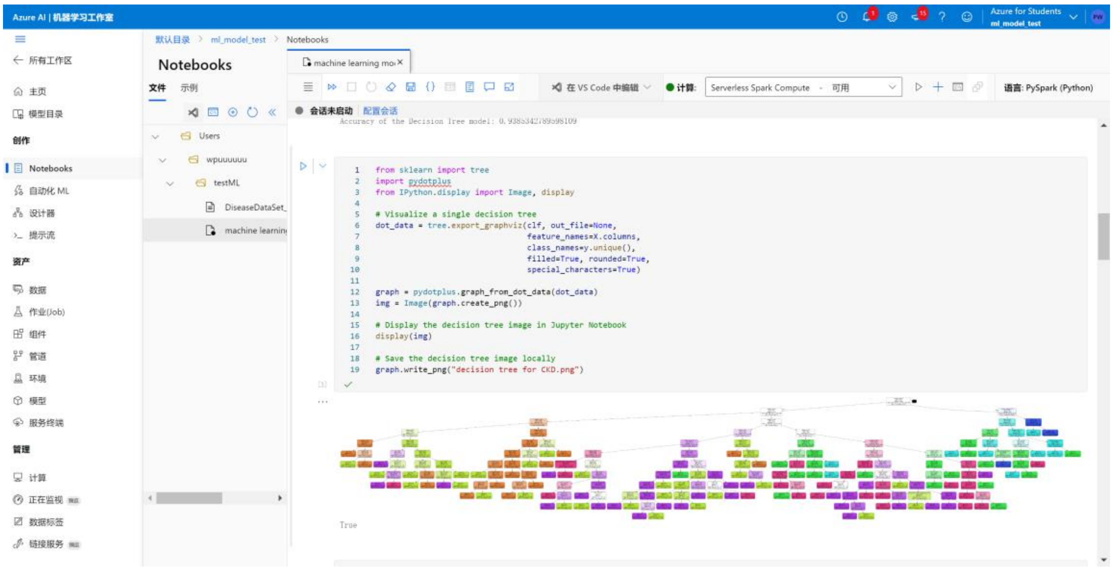
Data Ingestion and Management: Utilizing Azure Data Factory, we automated the ingestion of survey data into Azure Data Lake Storage, which offers a highly scalable and secure repository for big data analytics.



```
import pandas as pd

# Read data
data = pd.read_csv("DiseaseDataSet_raw_and_data_sinthetic.csv")

# Drop missing data
data.dropna(inplace=True)

data
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	F
0	Female	21.00	1.62	64.00	yes	no	2.0	3.0	Sometimes	no	2.00	no	0
1	Female	21.00	1.52	56.00	yes	no	3.0	3.0	Sometimes	yes	3.00	yes	3
2	Male	23.00	1.80	77.00	yes	no	2.0	3.0	Sometimes	no	2.00	no	2
3	Male	27.00	1.80	87.00	no	no	3.0	3.0	Sometimes	no	2.00	no	2
4	Male	22.00	1.78	89.80	no	no	2.0	1.0	Sometimes	no	2.00	no	0
...	...	...	...	...	...	...	...	...	...	...	...	...	
2106	Female	20.98	1.71	131.41	yes	yes	3.0	3.0	Sometimes	no	1.73	no	1
2107	Female	21.98	1.75	133.74	yes	yes	3.0	3.0	Sometimes	no	2.01	no	1
2108	Female	22.52	1.75	133.69	yes	yes	3.0	3.0	Sometimes	no	2.05	no	1

Model Development and Training: Azure Machine Learning Workspaces provided a collaborative environment for developing and training our models. We utilized Azure ML’s automated machine learning capabilities to identify the best models and hyperparameters.



```
Accuracy of the Decision Tree model: 0.9385342789598109
```

```python
from sklearn import tree
import pydotplus
from IPython.display import Image, display

# Visualize a single decision tree
dot_data = tree.export_graphviz(clf, out_file=None,
                                feature_names=X.columns,
                                class_names=y.unique(),
                                filled=True, rounded=True,
                                special_characters=True)

graph = pydotplus.graph_from_dot_data(dot_data)
img = Image(graph.create_png())

# Display the decision tree image in Jupyter Notebook
display(img)

# Save the decision tree image locally
graph.write_png("decision tree for CKD.png")
```

Real-time Analytics: Integration with Azure Stream Analytics enabled us to perform real-time analytics on data streams for immediate risk assessment and dynamic pricing adjustments.

2) **Advantages of Cloud Integration:**

This cloud integration offers several advantages:

Scalability and Flexibility: Azure's scalable infrastructure allows our machine learning models to handle fluctuating volumes of data effortlessly, providing the flexibility to grow with the company's needs.

Data Lake and Big Data Platforms: Azure's data lake services compile and store vast amounts of structured and unstructured data, while Big Data platforms like Azure HDInsight enable efficient processing and analysis.

Real-time Analytics: By leveraging Azure Stream Analytics, the system can perform real-time data analytics, which is critical for dynamic pricing models and risk assessment.

Security and Compliance: Azure provides robust security features that comply with industry standards, ensuring the integrity and privacy of the sensitive health data being processed.

API Management: Azure API Management facilitates secure and efficient API exposure of our machine learning models to the application layer, enabling seamless communication and data exchange.

By utilizing Microsoft Azure Cloud, we ensure that our machine learning solutions are not only integrated with cutting-edge technology but also positioned for future advancements and innovations in the insurance sector.

2.4 Workflow-Based Application Implementation

2.4.1 System design and implementation

The backend of the Holiday Insurance Company system is developed in Java, utilizing the Spring Boot framework alongside the MyBatis-Plus ORM architecture. The frontend is crafted using Unity. Communication between the frontend and backend adheres to RESTful APIs. The database is managed through Azure SQL Database. Below is a detailed breakdown of the implementation:

The backend utilizes the Spring Boot architecture, configured via YAML files. It connects to the Azure cloud database using the JDBC SQL Server Driver, ensuring reliable and efficient data processing. The backend architecture is divided into four layers: Entity, Mapper, Service, and Controller.

Entity Layer: This layer represents the domain models that are mapped to the database tables.

Mapper Layer: Using the MyBatis-Plus framework, this layer handles data persistence operations (CRUD), mapping the database relations to entities within the application.

Service Layer: This layer focuses on implementing methods and encapsulates the business logic of the application.

Controller Layer: Responsible for managing the flow of the application, this layer handles business process control and communicates with the frontend using RESTful APIs.

The use of the ORM framework simplifies database interactions, allowing developers to work with data using object-oriented methods, enhancing development efficiency and database access security by preventing SQL injection attacks. The RESTful API in the Controller layer ensures a separated and efficient communication between frontend and backend.

The machine learning method used in this project is decision trees. The decision tree reads the user's data through the Survey panel, uploads it to the cloud-trained model of Microsoft Azure AI Studio for processing, and returns the classification results to determine the user's risk of chronic diseases in real time and recommend an insurance plan.

Customer survey data serves as model input, including numerical and textual data. When training the model, encode all text categorical variables to numeric in order to facilitate input to the model. In the end, the prediction accuracy reached 93.8%, which is a credible prediction model that can be applied to this project.

2.4.2 Workflow implementation based on the design

The workflow would follow these steps:

1)User Interface Design:

Develop a user-friendly interface where users can purchase, cancel, and view insurance plans.

Implement secure login and personal data management systems.

2)Insurance Management System:

Create a database to store and manage different insurance plans, user details, and transaction records.

Develop functionalities for purchasing and cancelling insurance, along with updating user profiles and plan details.

3)Big Data Model Integration:

Establish a robust backend system to maintain and analyze chronic illness data.

Continuously feed the model with up-to-date healthcare data and research findings to analyze trends and risk factors associated with chronic diseases.

4)Real-Time Analysis and Adjustment:

Use the big data model to perform real-time analysis of the collected data.

Implement algorithms to adjust insurance pricing, policies, and offerings based on the analysis. Ensure that adjustments are transparent and communicated effectively to users.

5)Testing and Quality Assurance:

Rigorously test the system for functionality, security, and performance issues.

Collect user feedback for continuous improvement and ensure compliance with healthcare regulations and data protection laws.

6)Deployment and Maintenance:

Deploy the platform while ensuring scalability and reliability.

Set up a maintenance schedule for regular updates, security checks, and model retraining to adapt to new chronic illness trends.

2.5 Application Documentation

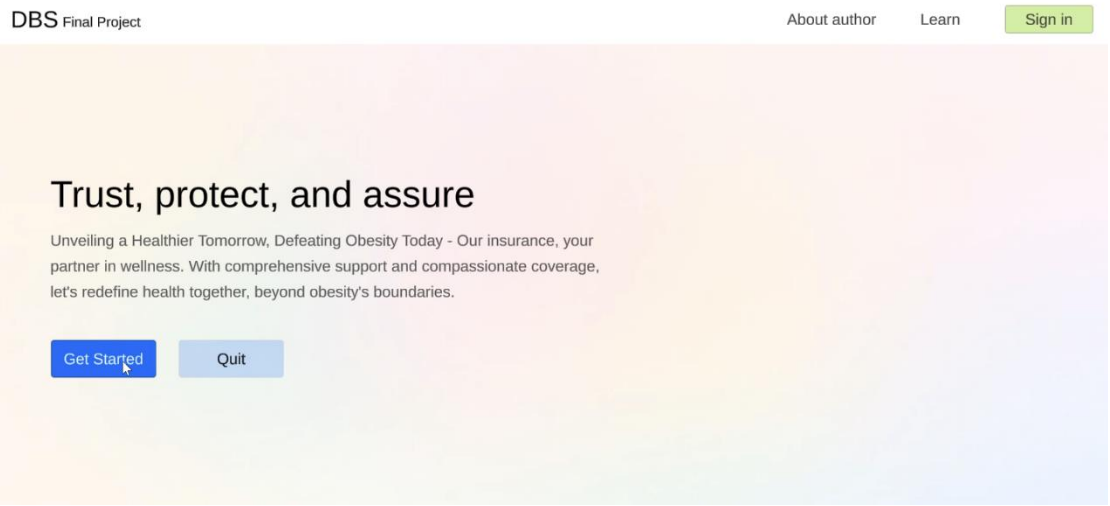
2.5.1 Detailed documentation of the application.

The application documentation has been enriched with details specific to the Unity-created front-end interface. It provides a comprehensive guide on navigating the application, with a particular focus on the interactive components designed to engage users in chronic disease management and insurance plan selection. The Unity front-end is meticulously connected to the Spring Boot back-end, ensuring a seamless user experience and real-time data processing.

2.5.2 Screenshots demonstrating the application functionality.

This includes a series of screenshots illustrating the application’s user interface:

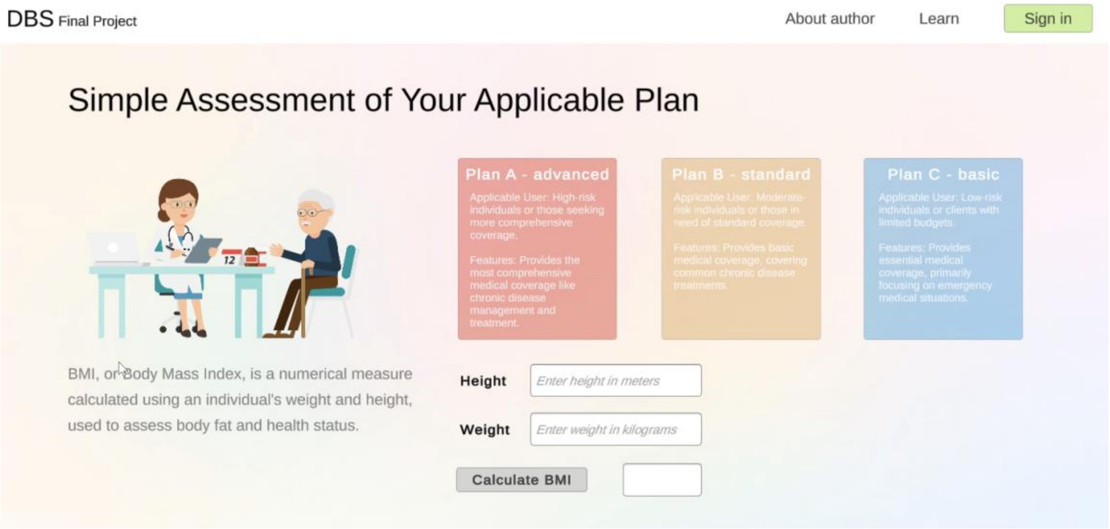
1) **Home Screen:** Showcases the welcoming splash screen with options to 'Get Started' or 'Quit,' emphasizing the application's commitment to health management.



Trust, protect, and assure

Unveiling a Healthier Tomorrow, Defeating Obesity Today - Our insurance, your partner in wellness. With comprehensive support and compassionate coverage, let's redefine health together, beyond obesity's boundaries.

2) **Plan Selection:** Depicts the assessment interface where users can evaluate which insurance plan suits them best, based on their health data.



Simple Assessment of Your Applicable Plan

Plan A - advanced

Applicable User: High-risk individuals or those seeking more comprehensive coverage.

Features: Provides the most comprehensive medical coverage like chronic disease management and treatment.

Plan B - standard

Applicable User: Moderate-risk individuals or those in need of standard coverage.

Features: Provides basic medical coverage, covering common chronic disease treatments.

Plan C - basic

Applicable User: Low-risk individuals or clients with limited budgets.

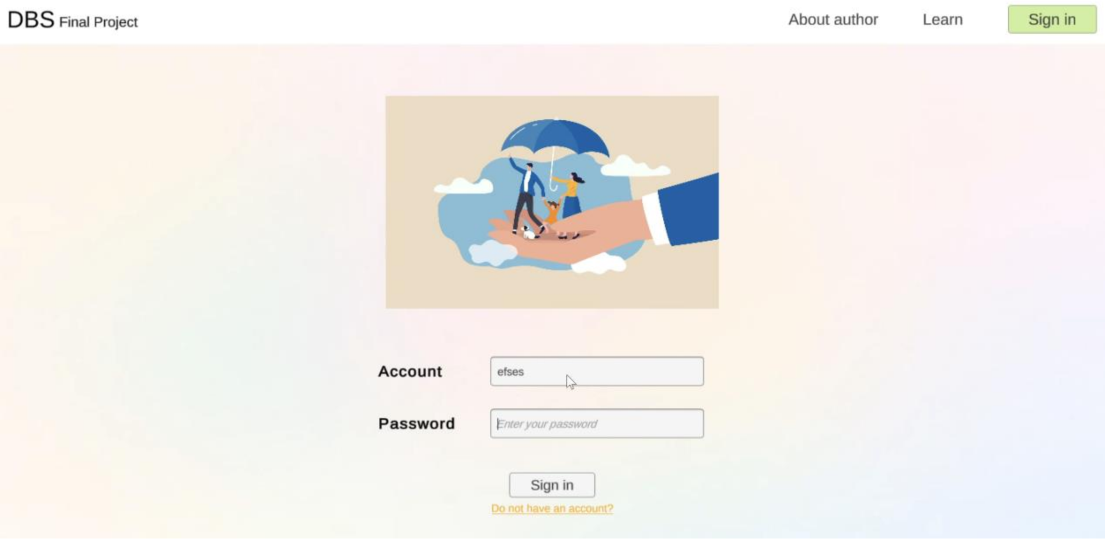
Features: Provides essential medical coverage, primarily focusing on emergency medical situations.

BMI, or Body Mass Index, is a numerical measure calculated using an individual's weight and height, used to assess body fat and health status.

Height *Enter height in meters*

Weight *Enter weight in kilograms*

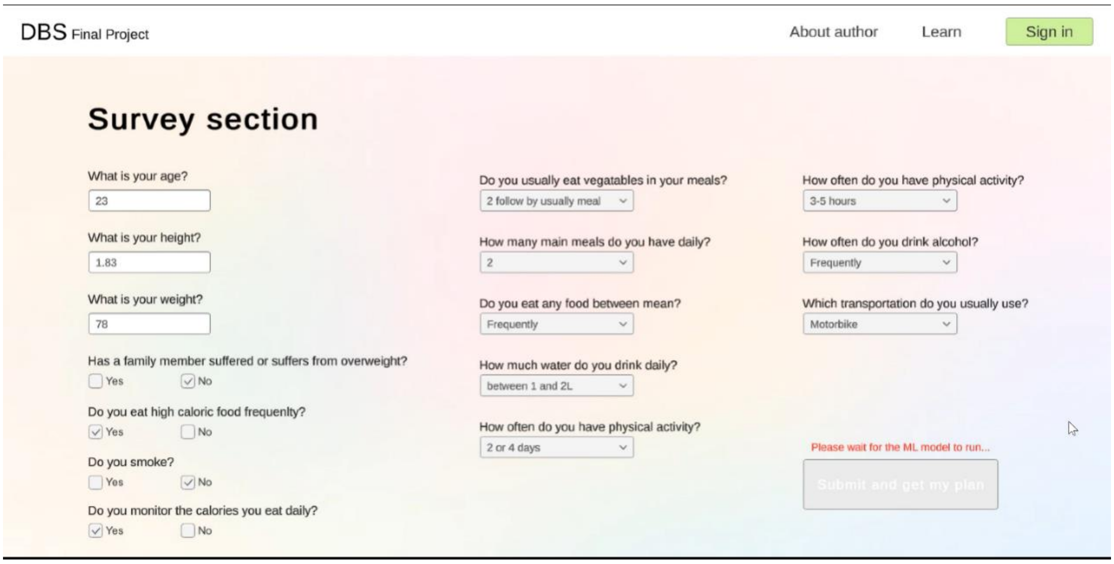
3) **User Authentication:** Displays the secure sign-in page, assuring users of their data privacy.



Account: efses

Password: Enter your password

4) **Survey Module:** Exhibits the comprehensive survey section where users input their health information to receive personalized insurance plan suggestions.



Survey section

What is your age? 23

What is your height? 1.83

What is your weight? 78

Has a family member suffered or suffers from overweight? □ Yes ☑ No

Do you eat high caloric food frequenlty? ☑ Yes □ No

Do you smoke? □ Yes ☑ No

Do you monitor the calories you eat daily? ☑ Yes □ No

Do you usually eat vegatables in your meals? 2 follow by usually meal

How many main meals do you have daily? 2

Do you eat any food between mean? Frequently

How much water do you drink daily? between 1 and 2L

How often do you have physical activity? 2 or 4 days

How often do you have physical activity? 3-5 hours

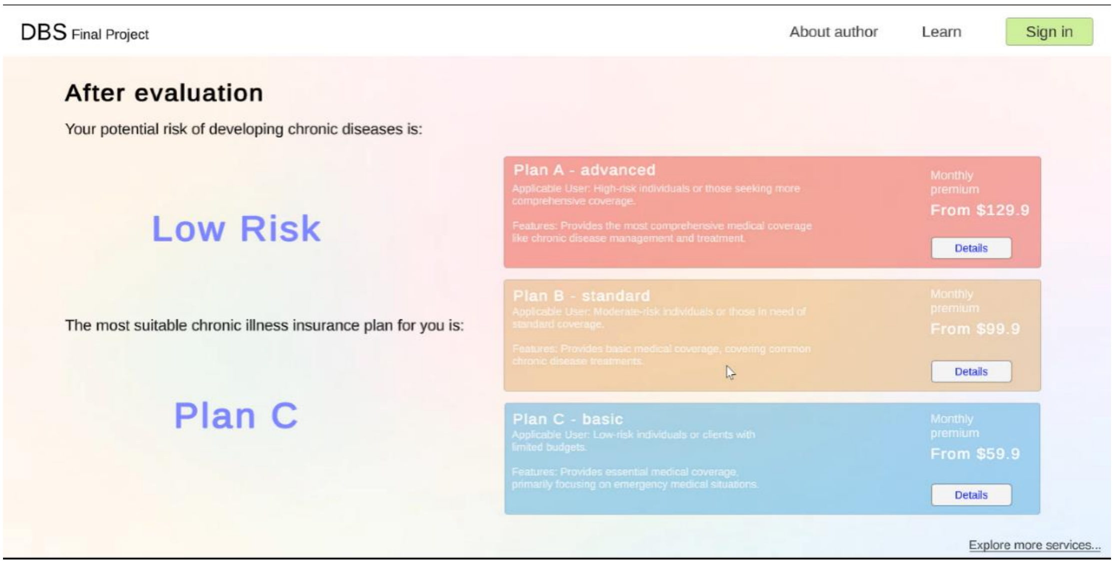
How often do you drink alcohol? Frequently

Which transportation do you usually use? Motorbike

Please wait for the ML model to run...

5) **Insurance Plan Recommendation:** The interface presents a succinct BMI calculator as the first step in assessing a user’s health profile. Following the survey input, which collects comprehensive health information, the system intelligently recommends the most suitable among three distinct insurance plans. Users are guided to Plan A - Premium Coverage for extensive medical needs, Plan B - Standard Coverage for balanced care, or Plan C - Basic Coverage for essential health services. The recommendation is personalized,

taking into account the individual's unique health data and survey responses to ensure an optimal match with their insurance requirements.



After evaluation

Your potential risk of developing chronic diseases is:

Low Risk

The most suitable chronic illness insurance plan for you is:

Plan C

Plan A - advanced

Applicable User: High-risk individuals or those seeking more comprehensive coverage.

Features: Provides the most comprehensive medical coverage like chronic disease management and treatment.

Monthly premium

From $129.9

Plan B - standard

Applicable User: Moderate-risk individuals or those in need of standard coverage.

Features: Provides basic medical coverage, covering common chronic disease treatments.

Monthly premium

From $99.9

Plan C - basic

Applicable User: Low-risk individuals or clients with limited budgets.

Features: Provides essential medical coverage, primarily focusing on emergency medical situations.

Monthly premium

From $59.9

2.5.3 Description of how the solution meets project requirements

We use the Unity front-end, Spring Boot back-end, and IntelliJ IDEA as our IDE to bring the project's objectives to life. The solution is designed to align closely with the project's objectives.

Chronic Disease Management: Users engage with an intuitive and interactive interface to input and manage their health data, which is then processed by the back-end for real-time insights.

Pricing Policies: The application includes a decision-support tool that allows users to understand the different insurance plans and select the one that aligns with their health needs and financial capabilities.

User Experience: The front-end is crafted for ease of use, with clear calls to action, simple navigation, and a comforting color scheme that makes the process of health data management and insurance plan selection stress-free.

2.5.4 Database query optimization techniques and results

This part of the documentation discusses the optimization of database queries, including:

Indexing: Implementation of strategic indexing to expedite query performance.
Query Refactoring: Techniques used to refactor and streamline complex queries to reduce execution time.
Connection Pooling: Utilization of connection pooling in Spring Boot to minimize the overhead of establishing connections to the database.

Performance improvements are presented through before-and-after metrics, demonstrating the enhancements in data retrieval and processing times.

2.5.5 ORM Framework Optimizations

Given that Spring Boot is often paired with ORM frameworks like Hibernate, the documentation elaborates on:

Lazy vs. Eager Loading: Decision-making criteria used to determine the loading strategy, balancing immediate data availability against overall performance.
Second-Level Cache: Configuration of a second-level cache to reduce database hits for frequently accessed data.
Batch Processing: Leveraging batch processing to enhance the efficiency of bulk operations, as supported by the ORM.

Optimization results, including reduced latency and increased transaction throughput, are highlighted with empirical evidence from testing scenarios.

3. Reference Architecture (RA) Documentation

According to the IHI Triple Aim framework, our project's Reference Architecture (RA) Documentation is as follows:

Architecture and Principles: The RA aligns with the Triple Aim, focusing on a patient-centric approach that integrates business needs with health outcomes. It prioritizes a seamless application experience through a robust, secure infrastructure that supports large-scale data analytics.

Development and Delivery: Methods for planning, delivering, and operating business solutions are rooted in agile and modular design, enabling flexible service delivery and continuous improvement in healthcare management.

Governance and Data Management: Governance structures uphold data integrity and compliance, ensuring that patient information is managed ethically while supporting the strategic objectives of enhancing care and managing costs.

4. Conclusion

4.1 Summary of Key Achievements.

We have made notable progress in our project, including the successful development of a comprehensive database system that integrates chronic disease data for customized insurance pricing. This achievement was complemented by the implementation of a robust machine learning model using Microsoft Azure Cloud, which has considerably improved our data processing and analytical capabilities. Furthermore, we have established a seamless integration between the Unity-based front-end and the Spring Boot back-end, resulting in a cohesive and user-friendly experience for our users.

4.2 Reflection on the Project Outcomes

Enhanced Chronic Disease Data Management: The project outcomes reflect a significant advancement in the insurance company's ability to manage and leverage chronic disease data.

Precision in Insurance Pricing: The use of machine learning models has provided a more accurate and dynamic approach to insurance pricing, offering a competitive edge in the market.

Improved User Engagement: User engagement has been improved through an intuitive and responsive application interface, facilitating better health management for customers.

4.3 Potential Future Improvements

In this section, we outline potential improvements that can further enhance our project.

Advancing Machine Learning Algorithms: Further refinement of machine learning algorithms to encompass a broader spectrum of health data and predictive analytics.

Real-Time Health Data Integration: Expansion of the data architecture to include real-time data streams for instantaneous health monitoring and risk assessment.

Enriched Healthcare Management: Development of additional modules to cover more aspects of healthcare management, such as wellness programs and preventive care initiatives.