**Software** **Requirements**

**Specification** For

**Insurance Customer Behaviour Prediction System**

23BD1A6643

23BD1A664D

23BD1A6652

23BD1A6658

23BD1A665K

**Table of Contents**

1. Introduction
   * 1.1 Purpose
   * 1.2 Document Conventions
   * 1.3 Intended Audience and Reading Suggestions
   * 1.4 Product Scope
   * 1.5 References
2. Overall Description
   * 2.1 Product Perspective
   * 2.2 Product Functions
   * 2.3 User Classes and Characteristics
   * 2.4 Operating Environment
   * 2.5 Design and Implementation Constraints
   * 2.6 User Documentation
   * 2.7 Assumptions and Dependencies
3. External Interface Requirements
   * 3.1 User Interfaces
   * 3.2 Hardware Interfaces
   * 3.3 Software Interfaces
   * 3.4 Communication Interfaces
4. System Features
   * 4.1 Customer Classification
   * 4.2 Government Policy Listing
   * 4.3 Q&A System for Policy Recommendations
5. Other Nonfunctional Requirements
   * 5.1 Performance Requirements
   * 5.2 Safety Requirements
   * 5.3 Security Requirements
   * 5.4 Software Quality Attributes
6. Other Requirements

Appendices

* Appendix A: Glossary
* Appendix B: Analysis Models
* Appendix C: To Be Determined (TBD) List

**1. Introduction**

**1.1 Purpose**

This document provides a detailed Software Requirements Specification (SRS) for a web-based system that predicts **customer behaviour for insurance companies**. The system identifies prominent customers using **Artificial Neural Networks (ANN)** . It supports insurers by classifying customers, listing government insurance policies, and offering a **Q&A tool** for policy recommendations.

**1.2 Document Conventions**

* **Bold text:** Important terms.
* *Italicized text:* Emphasis.
* Monospace text: Code snippets and API references.

**1.3 Intended Audience and Reading Suggestions**

This document is intended for:

* **Developers** – To guide implementation and system design.
* **Data Scientists** – To refine deep learning models.
* **Insurance Companies** – To integrate predictive analytics into customer profiling.
* **End Users** – To explore policies and receive recommendations.

**1.4 Product Scope**

The system will:

* **Predict customer prominence** using deep learning models (ANN & RNN).
* **Recommendations of policies** including govt. and private policies
* **Provide an interactive Q&A system** for policy recommendations.
* **Deployment**  in local system by cloning git repository.

**1.5 References**

* CRISP-DM Methodology
* React Documentation
* Flask & Node.js Frameworks
* MongoDB
* Deep Learning Models (ANN)

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**2. Overall Description**

**2.1 Product Perspective**

The system is a **web-based AI-powered platform** that enhances customer classification and policy recommendations for insurance companies.

**2.2 Product Functions**

* **Customer Classification:** Uses deep learning (ANN ) to classify customers as "Prominent" or "Non-Prominent."
* **Policy recommendations:** Displays **government and private medical insurance policies** based on user eligibility.
* **Chatbot:** An **interactive AI rule based** chatbot which give basic information about our website

**2.3 User Classes and Characteristics**

* **Insurance Companies:** Require detailed customer classification for better policy targeting.
* **Customers:** Need policy recommendations tailored to their needs.
* **Government Agencies:** Use customer classification data for policy planning.

**2.4 Operating Environment**

* **Frontend:** ReactJS with Tailwind CSS.
* **Backend:** Flask (for AI models) & Node.js (for scalability).
* **Database:** MongoDB for customer & policy data.
* **AI Models:** ANN for customer classification.
* **Deployment:** Local systems by cloning git repository.

**2.5 Design and Implementation Constraints**

* GPU required for deep learning model inference.
* Compliance with **data privacy** and **security regulations**.

**2.6 User Documentation**

* User Guide
* API Documentation

**2.7 Assumptions and Dependencies**

* Users provide structured customer data (age, income, etc.).
* No Internet connection required for local hosting in MongoDb.

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**3. External Interface Requirements**

**3.1 User Interfaces**

* **Dashboard:** Displays customer insights.
* **Upload Page:** Allows insurers to submit customer data.
* **Policy recommendations:** Based on factors like income, age.
* **Q&A System:** Asking questions through forms for dl model check prominence.
* **Chatbot:** a rule based chatbot for basic information of website

**3.2 Hardware Interfaces**

* Runs on **PCs by set up environments with required stack**.

**3.3 Software Interfaces**

* **Flask API** for backend services.
* **MongoDB Atlas** for customer and policy storage.

**3.4 Communication Interfaces**

* **REST API endpoints** for AI model interaction.
* **Secure authentication** using OAuth/JWT.

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**4. System Features**

**4.1 Customer Classification**

* Predicts customer prominence using **ANN** models.
* **Input:** age,bmi,smoker,pre\_existing\_conditions,dependents,hospital\_visits\_last\_year,chronic\_disease,physical\_activity\_level,alcohol\_consumption,gender,income,health\_score.
* **Output:** Classification as "Prominent" or "Non-Prominent."

**4.2 Q&A System for Policy Recommendations**

* Provides **interactive, Q&A** to help users find the best policy.
* **Response Time:** ~15-20 seconds per query.

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* Customer classification response time: **<1 min**.
* Should handle datasets **up to 1 million records**.

**5.2 Safety Requirements**

* Prevent **data loss** by **encrypting documents** before processing.
* Implement **fail-safe mechanisms** in case of system failure.

**5.3 Security Requirements**

* Secure authentication using **OAuth/JWT**.
* **Role-based access control** for different user groups.

**5.4 Software Quality Attributes**

* **Usability:** User-friendly interface.
* **Scalability:** Supports increasing user loads.
* **Maintainability:** Modular and well-documented codebase.

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Appendices**

**Appendix A: Glossary**

* **ANN (Artificial Neural Network):** AI model mimicking human brain functions.
* **CRISP-DM:** Data Science methodology used for this project.
* **MongoDB Atlas:** Cloud-based NoSQL database for storing data.

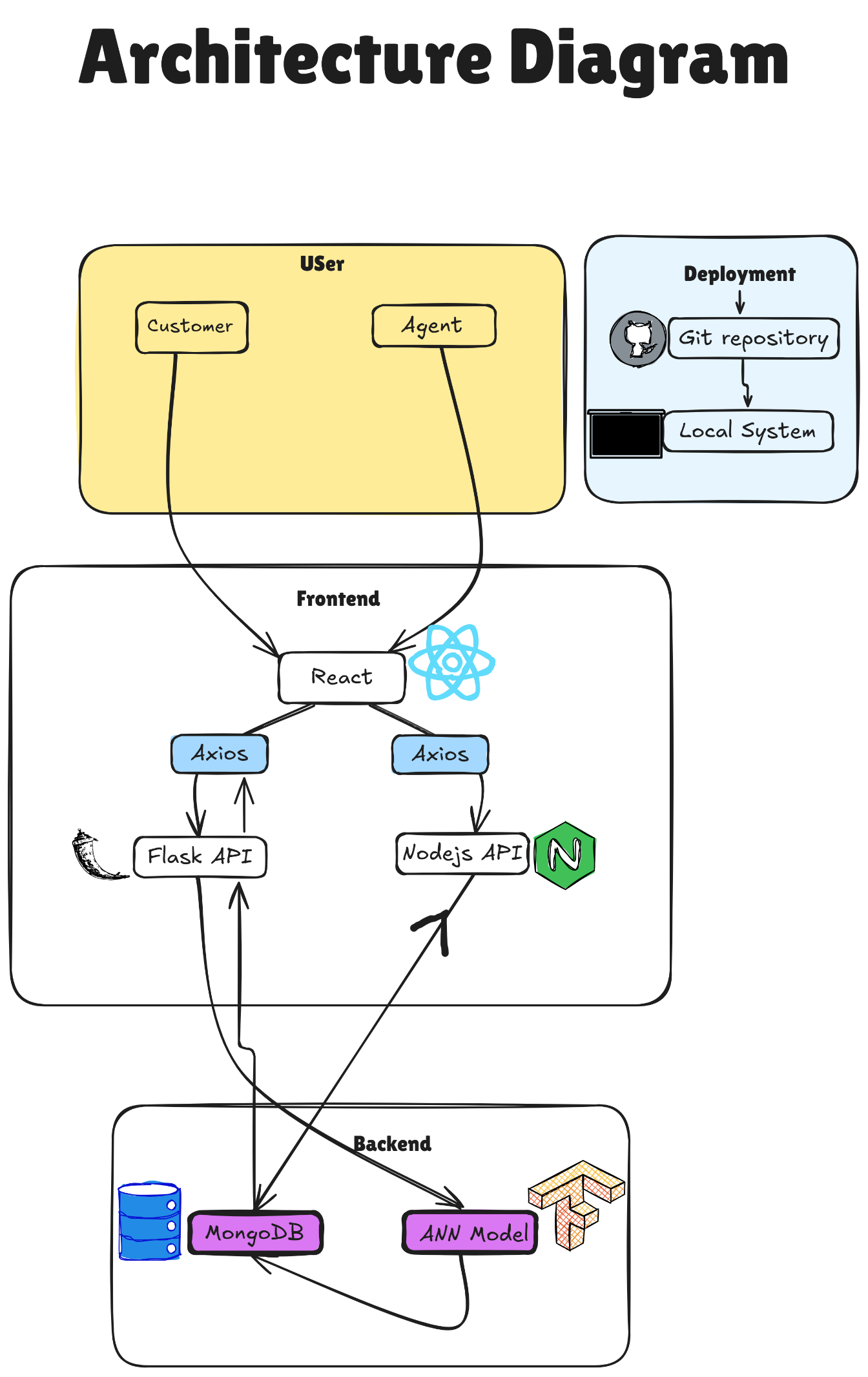
**Appendix B: Analysis Models**

* **Data Flow Diagram (DFD)**: Shows system data movement.
* **ER Diagram:** Defines database entities (Users, Policies, Queries).

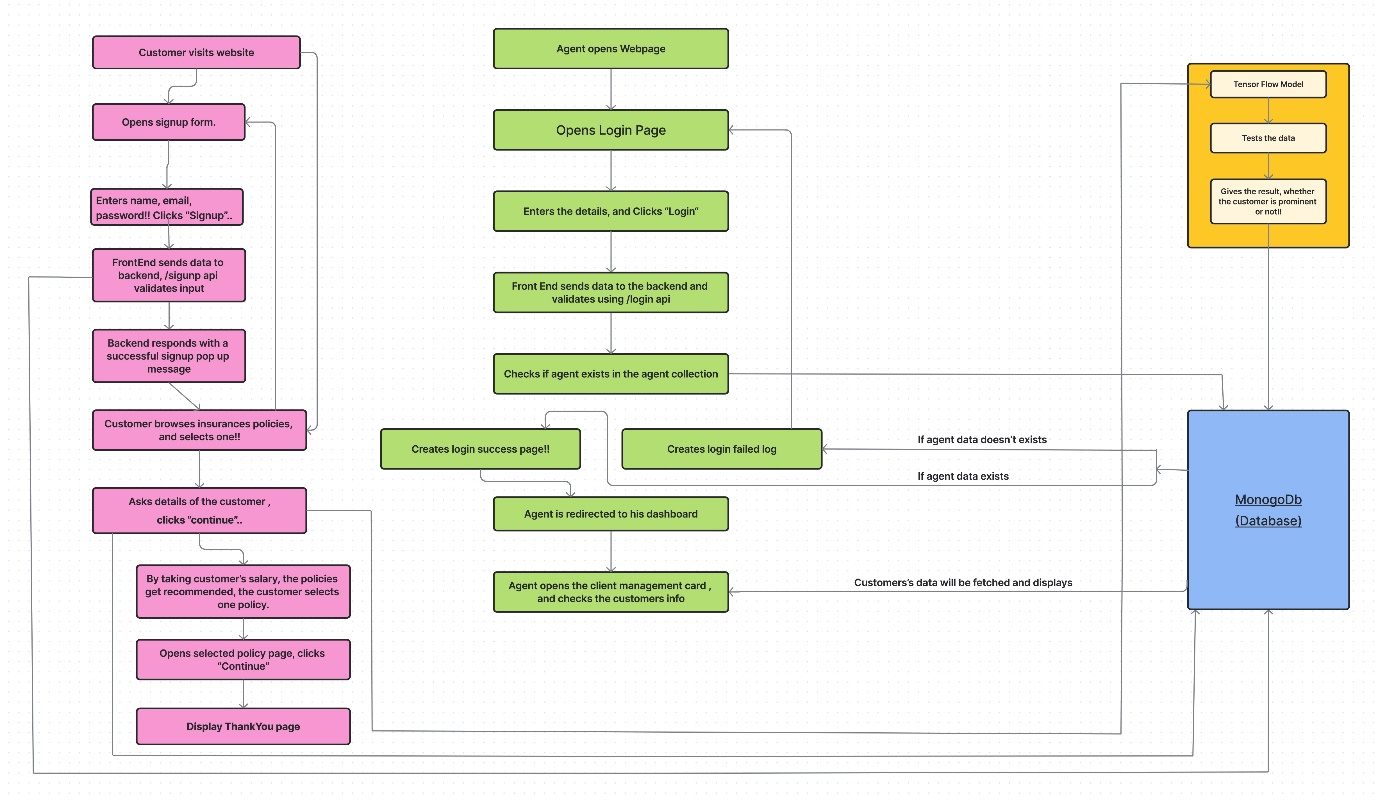
**Appendix C: TBD List**

* API rate limits.
* User feedback mechanisms.

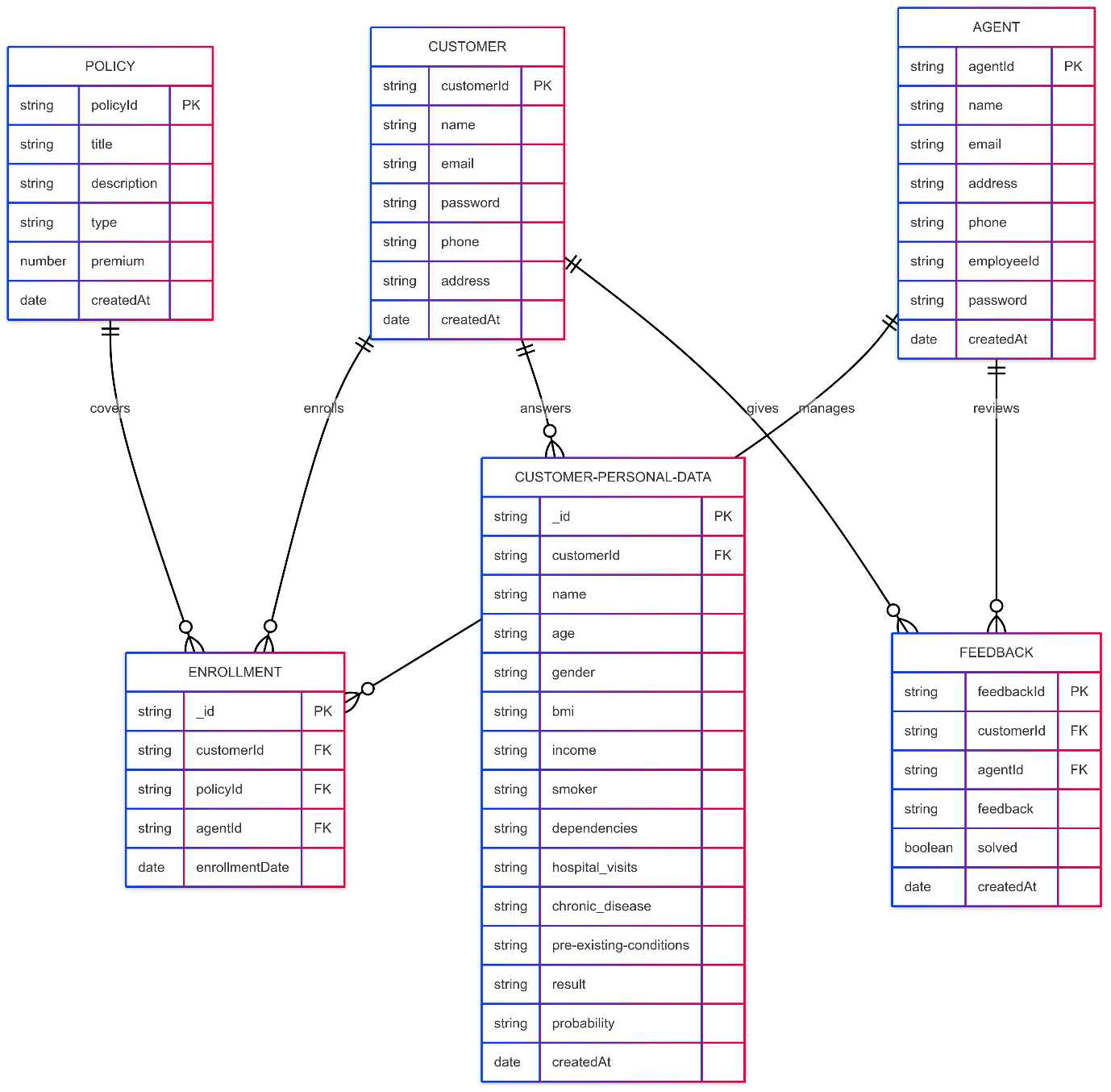
Architecture Diagram:



Work Flow:



**Database Diagram:**



**Class Diagram:**

