Document Version Control: 1

**Abstract:**

This Low-Level Design Document elaborates on the technical details of the Adult Census Income Prediction project, including scope, constraints, technical specifications, technology stack, proposed solution, workflows, and exceptional scenarios.

**1. Introduction:**

1.1 Why this Low-Level Design Document?

This document provides an in-depth description of the technical aspects of the Adult Census Income Prediction project, including workflows, technology stack, and handling exceptional scenarios.

**1.2 Scope:**

The document covers the detailed design considerations for the implementation of the project's technical components.

**1.3 Constraints:**

N/A

**1.3 Risks:**

The primary risk is that the chosen machine learning algorithm may not perform adequately on the provided dataset, leading to inaccurate predictions.

**1.4 Out of Scope:**

The document does not cover the specifics of data collection or the selection of the machine learning algorithm, which were covered in the High-Level Design.

**2. Technical Specifications:**

2.1 Predicting Disease:

The project's core task is predicting an individual's income level based on demographic features.

**2.2 Logging:**

Implement logging mechanisms to capture relevant information during data preprocessing, model training, and deployment.

**2.3 Database:**

Utilize a relational database to store and manage the trained model and relevant metadata.

**3. Technology Stack:**

- Programming Languages: Python

- Machine Learning Libraries: Scikit-learn, TensorFlow

- Data Preprocessing: Pandas, NumPy

- Database: SQLite

**4. Proposed Solution:**

The project involves developing a machine learning model that uses demographic features to predict an individual's income level. The model will be trained on preprocessed data and deployed in a production environment.

**5. Model Training/Validation Workflow:**

- Load the preprocessed data.

- Split data into training and validation sets.

- Choose a machine learning algorithm (e.g., Random Forest).

- Train the model using the training data.

- Validate the model's performance using the validation set.

**6. User I/O Workflow:**

- Collect demographic information from the user.

- Preprocess the user input.

- Use the trained model to predict the income level.

- Provide the prediction result to the user.

**7. Exceptional Scenarios:**

- If user input contains missing or invalid data:

- Provide appropriate feedback and request valid input.

- If the trained model's performance is below a certain threshold:

- Log the issue and notify administrators for investigation.

- If the database encounters errors during model storage:

- Implement error handling to ensure data integrity.