

1. Choose a project from an internship portal and try to write a HLD and LLD based on the sample given in your portal for a respective project.

High-Level Design (HLD):

1. System Overview:

- The credit default prediction system aims to predict the probability of credit default based on credit card owner's characteristics and payment history.

2. System Components:

- Data Collection: Gather credit card owner's data, including characteristics and payment history.
- Data Preprocessing: Perform data cleaning, feature engineering, and data transformation tasks to prepare the data for analysis.
- Feature Selection: Identify relevant features for credit default prediction.
- Model Building: Train and evaluate different machine learning algorithms to find the best model for prediction.
- Model Deployment: Host the trained model on a suitable platform for real-time credit default prediction.
- User Interface: Develop a user-friendly interface to input customer information and receive credit default predictions.

3. System Workflow:

- Data Collection: Retrieve credit card owner's data from relevant sources (e.g., databases, APIs).
- Data Preprocessing: Clean the data by handling missing values, outliers, and data inconsistencies. Perform feature engineering tasks, such as creating new features or transforming existing ones.
- Feature Selection: Use statistical analysis or machine learning techniques to select the most informative features for credit default prediction.
- Model Building: Train and evaluate multiple machine learning models, such as logistic regression, decision trees, random forests, or gradient boosting, using suitable evaluation metrics (e.g., accuracy, precision, recall).
- Model Deployment: Host the selected model on a cloud platform (e.g., AWS, Azure, Google Cloud) or edge device, ensuring scalability, availability, and security.
- User Interface: Develop a web-based or application-based interface to collect user inputs (credit card owner's characteristics and payment history) and display credit default predictions.

Low-Level Design (LLD):

1. Data Preprocessing:

- Handle missing values: Impute missing values using techniques like mean, median, or advanced imputation methods.
- Outlier detection: Identify and handle outliers using statistical methods or anomaly detection algorithms.
- Data normalization/standardization: Scale numerical features to a common range.
- Categorical variable encoding: Encode categorical variables using techniques like one-hot encoding or label encoding.
- Feature engineering: Create new features from existing ones or transform features based on domain knowledge.

2. Model Building:

- Split dataset: Divide the dataset into training and testing sets.
- Model selection: Implement and evaluate various machine learning algorithms suitable for classification tasks.
- Hyperparameter tuning: Optimize model performance by tuning hyperparameters using techniques like grid search or random search.
- Model evaluation: Assess model performance using appropriate evaluation metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

3. Model Deployment:

- Export trained model: Save the trained model in a serialized format for deployment.
- Model serving: Deploy the model on a suitable platform, considering factors like scalability, latency, and resource requirements.
- API development: Expose the model as an API endpoint to accept credit card owner's data and return credit default predictions.
- Integration with user interface: Connect the user interface with the model API for real-time prediction.

4. User Interface:

- Design and develop an intuitive user interface to collect credit card owner's characteristics and payment history.
- Validate and sanitize user inputs to ensure data integrity and security.
- Communicate with the model API to send user inputs and receive credit default predictions.

- Display prediction results to the user in a clear and understandable format.