Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	16-11-2023
Team ID	Team-592025
Project Name	Project - Online Payments Fraud Detection Using ML
Maximum Marks	4 Marks

Technical Architecture:

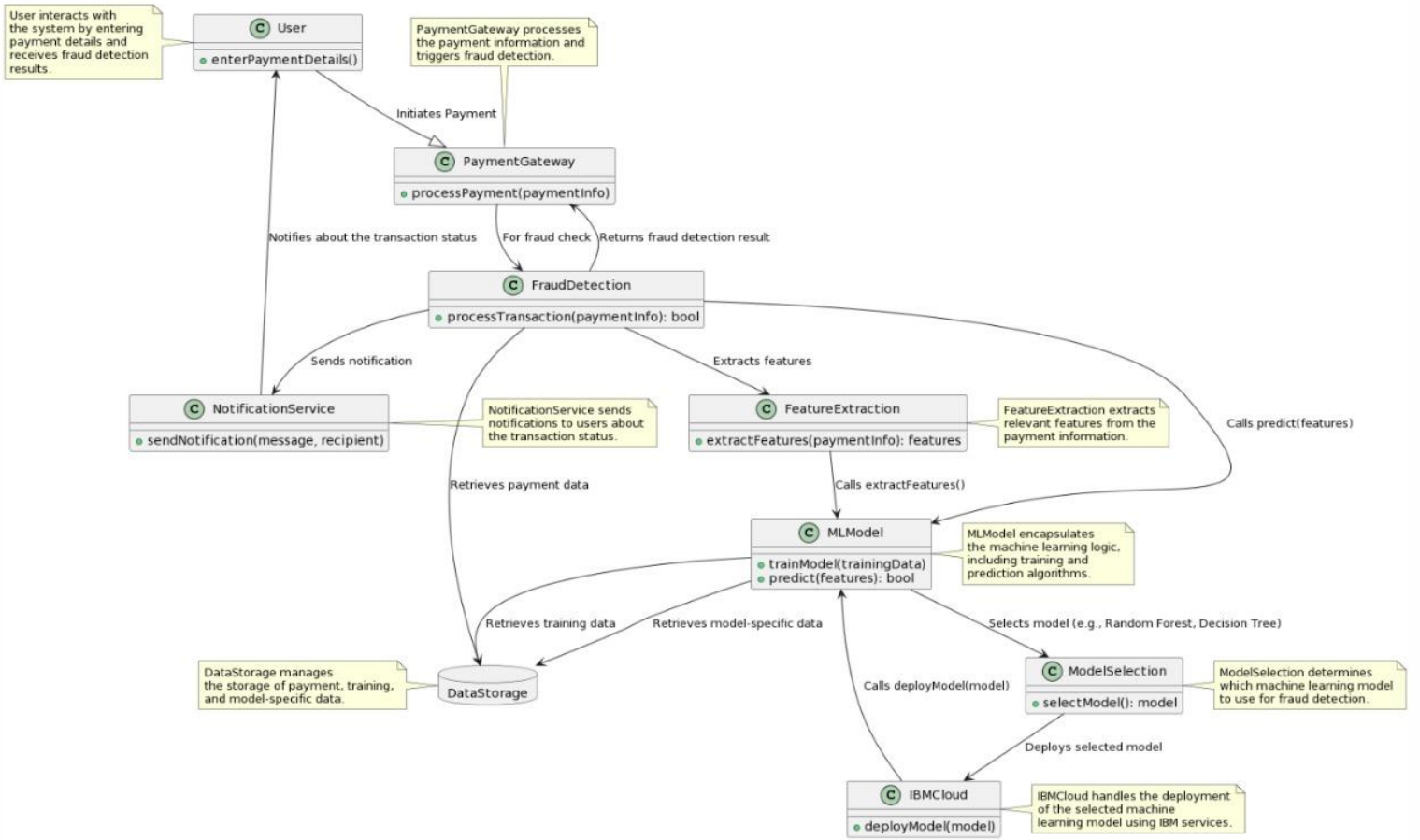

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web Application for user interaction	HTML, CSS, JavaScript / React.
2.	Application Logic-1	Backend logic for transaction processing	Python / Flask
3.	Application Logic-2	Fraud Detection Logic using Classification Algorithms	Decision Trees, Random Forest, SVM, Extra Trees, XGBoost
4.	Application Logic-3	Model Training and Testing Logic	Scikit-Learn, TensorFlow, PyTorch
5.	Database	Storage for transaction and model-related data.	MySQL or NoSQL Database
6.	Cloud Database	Cloud-based database service	IBM DB2, IBM Cloud
7.	File Storage	Storage for the best-selected model (saved in pkl format)	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Machine Learning Model	Classification Models for Fraud Detection	Decision Tree, Random Forest, SVM, Extra Tree, XGBoost.
9.	Infrastructure (Server / Cloud)	Cloud Deployment on IBM Cloud	Local, Cloud Foundry, Kubernetes
10	Integration	Flask Integration for web application	Flask
11	Model Selection	Logic for selecting the best model based on testing	Python / Scikit-Learn
12	Model Serialization	Saving the selected model in pkl format	Python / Scikit-Learn

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Python-based frameworks for machine learning and backend	Scikit-Learn, Flask
2.	Security Implementations	Encryption and Access Controls for Data and APIs	SHA-256, IAM Controls, TLS/SSL, Firewalls, OWASP
3.	Scalable Architecture	Microservices architecture for flexibility and scalability	Kubernetes, Cloud Foundry
S.No	**Characteristics**	**Description**	**Technology**
4.	Availability	High availability achieved through cloud deployment and load balancing	IBM Cloud Foundry, Kubernetes, Load Balancers
5.	Performance	Optimized performance with request handling, caching, and CDN usage.	Caching Strategies, Content Delivery Networks (CDN), Load Balancing, Efficient Algorithms

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d