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

Date	03 October 2022
Team ID	592046
Project Name	Restaurant recommendation

Technical Architecture:

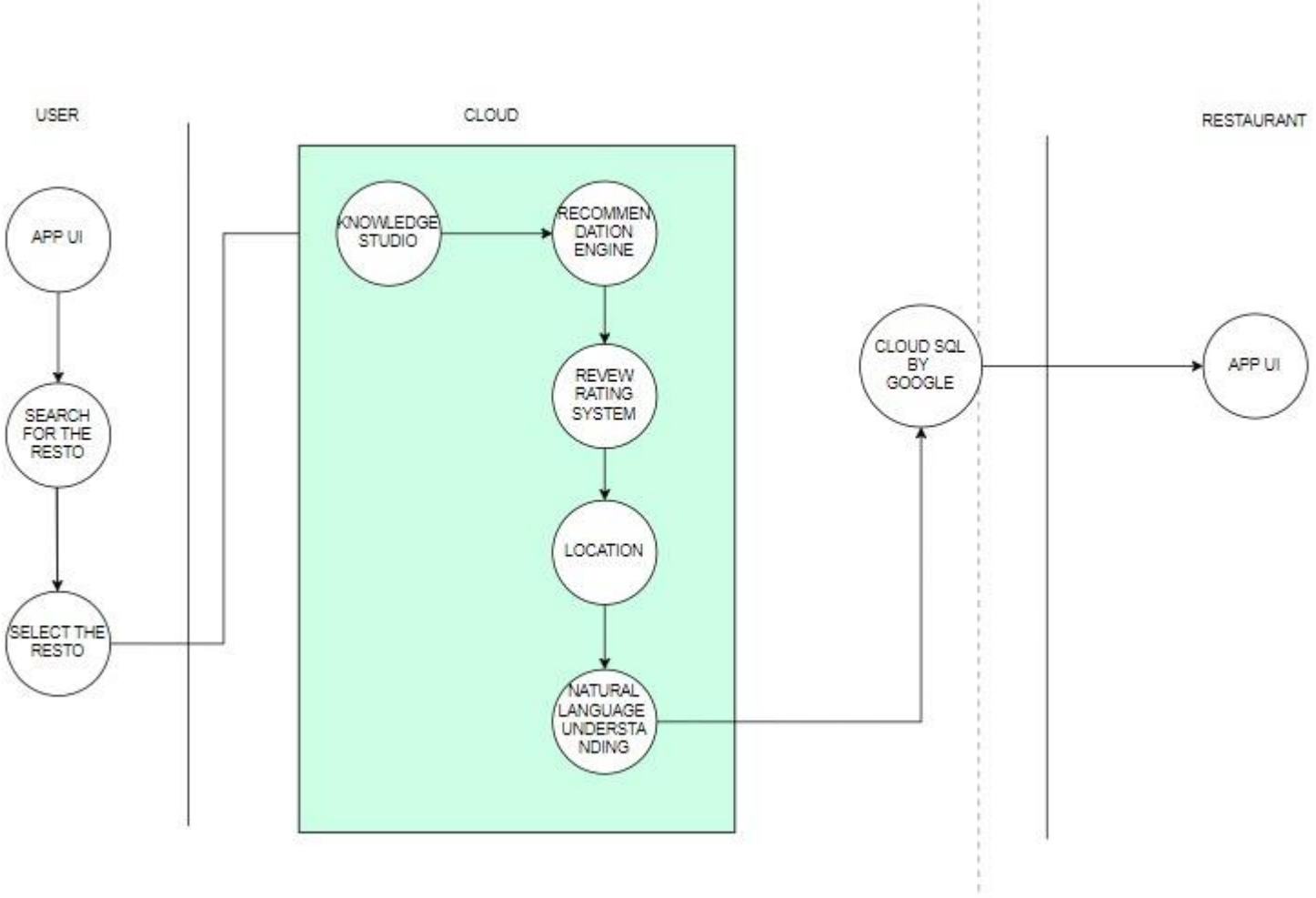

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The front-end component that users interact with to search for restaurants, view recommendations, and manage preferences.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Manages business logic, processes user requests, and communicates with the database and other services.	Express.js (Node.js), Django (Python)
3.	Application Logic-2	Manages business logic, processes user requests, and communicates with the database and other services.	Ruby on Rails (Ruby)
4.	Application Logic-3	Manages business logic, processes user requests, and communicates with the database and other services.	Flask (Python), or other server-side frameworks.
5.	Database	Stores and retrieves data related to users, restaurants, reviews, and other relevant information.	PostgreSQL, MySQL, MongoDB (for unstructured data), or other relational or NoSQL databases
6.	Cloud Database	A cloud database is a database built to run in a public or hybrid cloud environment to help organize, store, and manage data within an organization	Cloud sql by google,
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Search API	Enables efficient searching and retrieval of restaurant data.	Elasticsearch, Algolia, Solr
9.	Machine Learning Model	Collaborative filtering recommends items based on the preferences of users who have similar tastes. It can be user-based or item-based.	Collaborative filtering
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.
11.	Recommendation engine	Generates personalized restaurant recommendations based on user behavior, preferences, and other relevant factors.	ensorFlow or PyTorch

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	TensorFlow, Apache Mahout, pyTorch
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SSL/TLS, Role-Based Access Control (RBAC), OAuth or OpenID Connect
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Nginx or HAProxy, Apache Cassandra or Amazon DynamoDB, Apache Kafka or RabbitMQ.

S.No	Characteristics	Description	Technology
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used