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

Date	16 October 2022
Team ID	PNT2022TMID32893
Project Name	Project - Personal Expense Tracker
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

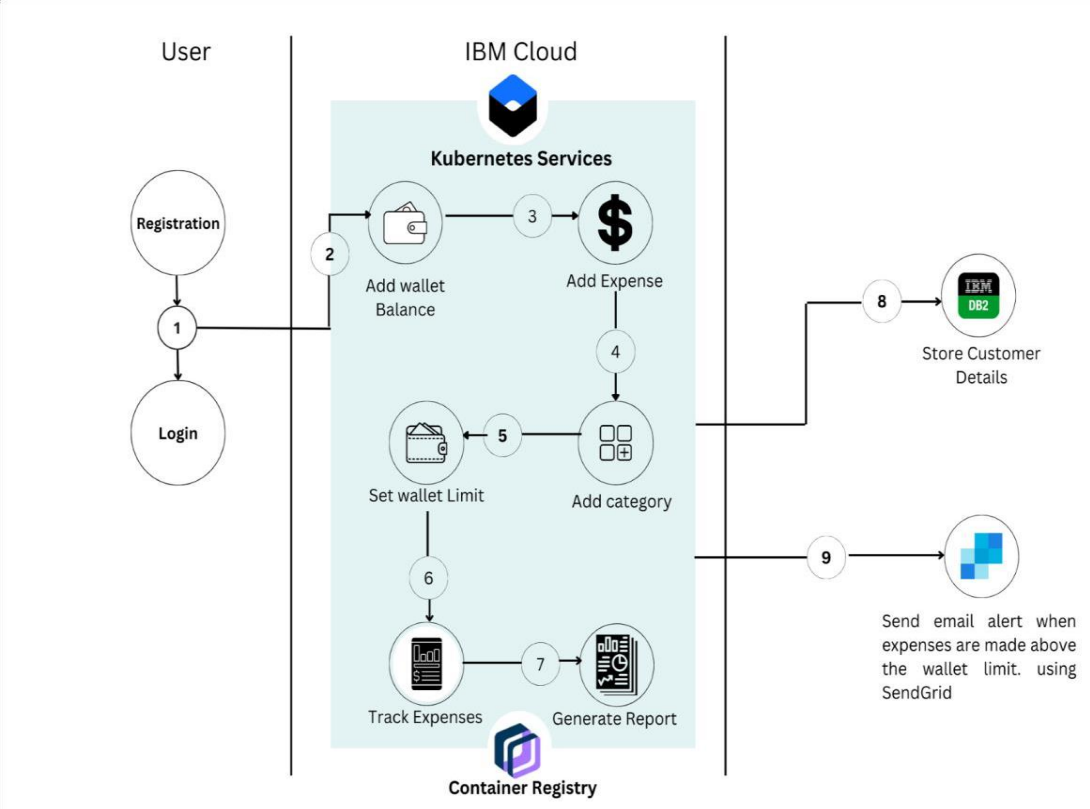

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chat bot etc.	HTML, CSS, JavaScript
2.	Registration and Login	To develop the application	Python, Docker
3.	Wallet Dashboard	IBM Cloud Kubernetes Service provides a native Kubernetes experience that is secure and easy to use. This tool is used to load-balance, scale, and monitor the containers.	IBM Cloud Kubernetes Services
4.	Tracking of Expenses.	IBM Container Registry enables to store and distribute Docker images in a managed, private registry.	IBM Cloud Container Registry
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	To send email alerts when the expenses are made above the wallet limit.	SendGrid

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask is an open source framework written in Python.	Flask
2.	Security Implementations	The user accounts are configured to only allow access from users with specific privileges.	IBM DB2
3.	Scalable Architecture	Three-tier architecture- user server, application server and cloud server.	Python, IBM Cloud Services
4.	Availability	Kubernetes services, the crudest form of load balancing traffic. The most basic type of load balancing is load distribution.The Docker load balancer runs on every node and can load balance requests across any of the containers on any of the hosts in the cluster.	Kubernetes and Docker
5.	Performance	Can handle a large number of requests per second.	IBM Container Registry.

References:

https://ieeexplore.ieee.org/document/9250969

https://www.cerdonis.tech/blogs/build-an-expense-tracking-app/

https://bootcamp.uxdesign.cc/expense-manager-application-ui-ux-case-study-the-dignitas-a6714900c60f