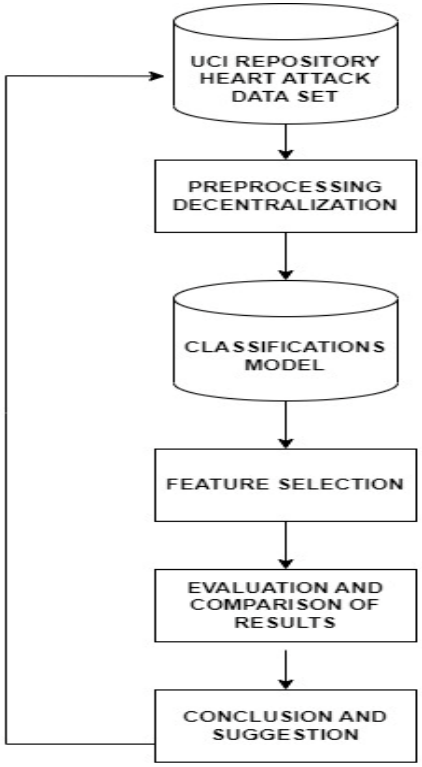
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	15-oct-2022
Team ID	PNT2022TMID21580
Project Name	Visualizing and predicting heart disease with an interactive dashboard

PROPOSED MODEL:



TECHNICAL ARCHITECTURE:

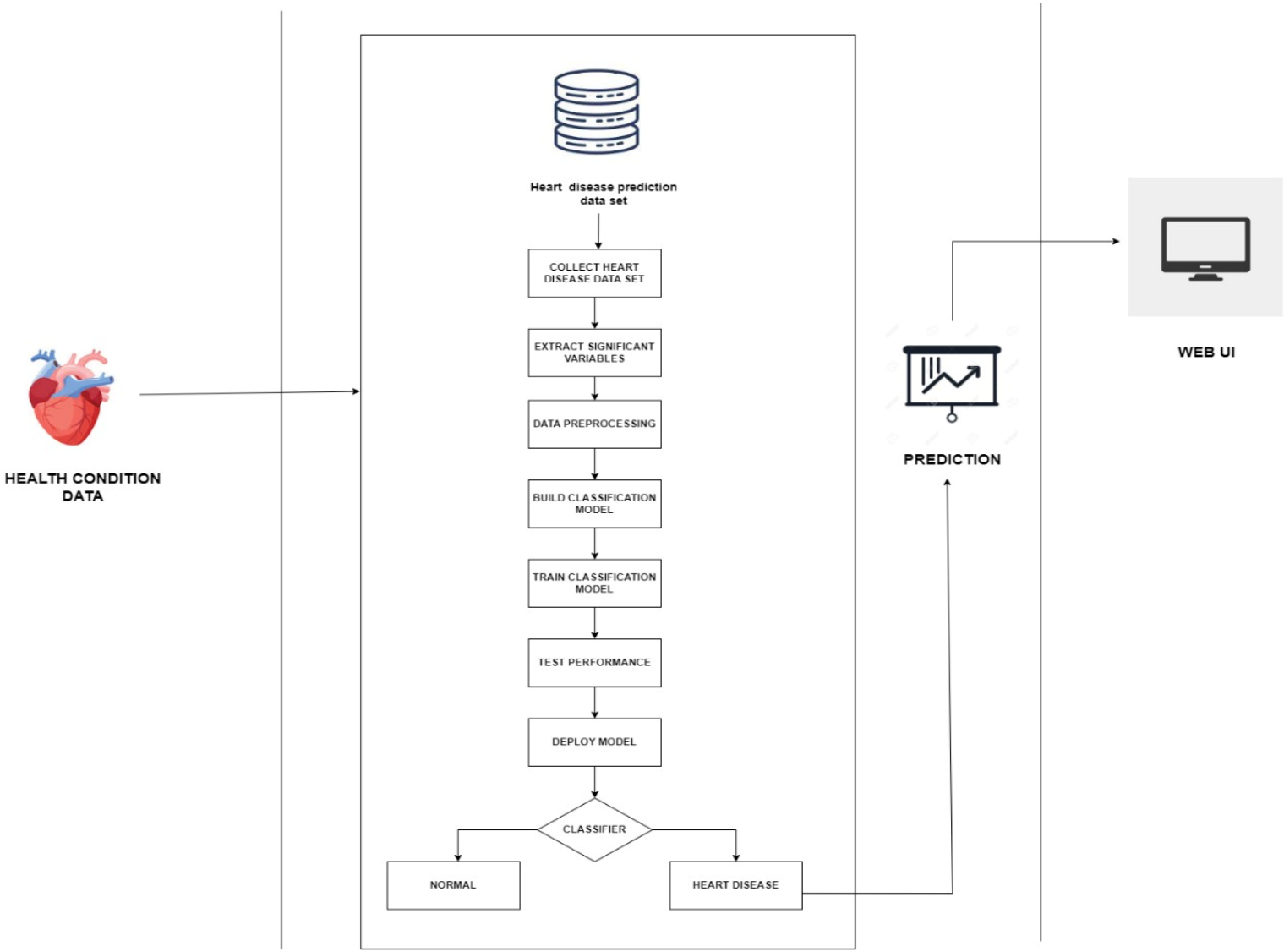


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with the application through the web UI	HTML, CSS, JavaScript, Angular Js
2.	Application Logic-1	Logic for registration feature of the application	Python
3.	Application Logic-2	Logic for a login feature of the application	Python
4.	Application Logic-3	Logic for one click signup through google and Facebook	Google and Facebook API's
5.	Database	Contains information of registered users, user's health related data	MySQL
6.	Cloud Database	Storing data on cloud	IBM DB2.
7.	File Storage	File storage requirements	Local Filesystem
8.	External API-1	one clicks signup feature can be implemented using API's	Google API and Facebook API.
9.	Machine Learning Model	The main objective of heart prediction system is to discover and extract hidden knowledge associated with diseases from a historical heart data set and to identify whether a patient is diagnosed with heart disease or not. Heart disease prediction system aims to exploit data mining techniques on medical data set to assist in the prediction of the heart diseases	Predictive and Classification model.
10.	Infrastructure (Server / Cloud)	Application Deployment: on Local System. Local Server Configuration: Flask inbuilt web server	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Python microservice web frame work for web development
2.	Security Implementations	Flask-Security allows you to quickly add common security mechanisms to your Flask application. They include Session based authentication, Role management, Password hashing, Basic HTTP authentication, Token based authentication, Token based account activation, Token based password recovery / resetting, User registration, Login tracking, JSON/Ajax Support	Flask Security modules.
3.	Scalable Architecture	Flask is highly scalable. It can process a high number of requests each day, as it gets popular. Flask modularizes the code so that developers can segregate them as independent chunks and use them as the code base grows.	Flask
4.	Availability	As the application is developed with Flask it can handle large number of requests so our application will be available to large number of users and process many numbers of requests.	Flask
5.	Performance	Integrated support for unit testing, RESTful request dispatching., Uses Jinja templating, Support for secure cookies (client-side sessions)	Flask

References:

https://www.analyticsvidhya.com/blog/2020/09/web-application/

https://app.diagrams.net/

https://app.creately.com/

https://www.researchgate.net/publication/331589020_Heart_Disease_Prediction_System

https://towardsdatascience.com/an-end-to-end-machine-learning-project-heart-failure-prediction-part-2-4518d2596421