PROJECT REPORT

CONTENTS

1. INTRODUCTION

1.1 Project Overview

Chronic Kidney Disease(CKD) arises when there is a gradual loss of kidney function. The main function of the kidney is to filtrate the blood in our body, the gradual loss of kidney function can cause the waste to deposit in our body. CKD is the one of the prevalent diseases in the world that threatens the lifespan of a person if it is not diagnosed in its early stages. CKD does not show any symptoms at all or in some cases few symptoms in the early stage, so it is difficult for people to detect the disease in the early stages. Thus most of it remains undiagnosed until advanced stage leading to delayed treatment. We propose a solution wherein if required data is available, this problem can be easily solved by running a machine learning model which can predict the presence of CKD in the early stages.

1.2 Purpose

The purpose of this project is predict the presence or absence of chronic kidney disease using the machine learning model

2. LITERATURE SURVEY

2.1 Existing problem

Normally people tend to go to hospital when they are experiencing any symptoms, but chronic kidney disease does not show any symptoms at all or in some cases few symptoms in early stages. So it is difficult to detect in the early stage and leaving it undiagnosed may lead to an advanced stage which is hard to fatal situations or death. Traditional methods used to determine chronic kidney disease are blood test to detect glomerular filtrate rate(GFR) and urine test to determine albumin.

[1] The paper proposes an unsupervised framework to detect chronic kidney diseases. It compares the results of four unsupervised algorithms namely K-Means clustering, DB-Scan, Isolation Forest, and Autoencoder to determine the efficiency of each. Feature extraction is performed prior to processing the data to reduce the high dimensionality. The proposed method claims to achieve 99.3% accuracy and concludes that highest accuracy was given by K-Means algorithm compared to others.

[2] The paper proposes a deep neural network approach to detect chronic kidney disease. It uses Deep Belief Network (DBN) with modifications and Softmax as activation function. The deep

network consists of 6 layers with 24,13,28,8,4,2 nodes respectively, the first and last layer being the input and output nodes. The paper claims to achieve a 98.5% accuracy.

[3] Chronic kidney disease is one of the severe problems with a high mortality rate. Early detection will help to cure the disease or even reverse the disease. Machine learning will provide an efficient solution for the early diagnosis of the disease. In this paper, they proposed machine learning methodologies for the early detection of chronic disease. They overcome the problem of missing values in the dataset by KNN imputation. They have used six machine learning models like support vector machine, k-nearest neighbor, logistic regression, random forest, feed forward neural network, naive bayes classifier. Among these models, the random forest model has the highest accuracy, thus they produced an integrated model which combines random forest and logistic regression to achieve better accuracy.

[4] Chronic kidney disease(CKD) is a disease that does not show any symptoms at all or few symptoms in the early stage, so it is difficult to predict and prevent the disease and if it is left unpredicted it could cause fatal damages to the human body. As machine learning is best in its analysis and prediction, it could provide a solution to this problem. In this paper, they downloaded the dataset from UCI repository. It includes 400 instances and 25 features. Out of these 25 features, only 14 are used to build the model. Here they used two machine learning models namely SVM and Decision Tree for prediction. Decision tree approach has an accuracy of 91.75% and the SVM approach has an accuracy of 96.75%

[5] Chronic Kidney Disease(CKD) has become a chronic disease with a steady growth rate among the population. It's crucial to develop effective methods for early prediction for CKD. We can utilize machine learning models for this. The steps involved in this process include data collection and preprocessing. This paper makes use of a missing handling method such as collaborative filtering and attribute selection. Then we have our feature selection following which the models are selected and trained. 11 models were tested out of which extra tree classifier and random forest classifier has the highest accuracy and minimal bias.This paper also highlights the importance of incorporating domain knowledge in both medicine and computer science when using Machine learning for early prediction of CKD

[6] Chronic kidney disease is one of the prevalent diseases in the world. As it does not reveal any symptoms in the early stage, most of the cases remain undiagnosed until the advanced stage. In this paper, they reviewed the existing machine learning models to diagnose chronic kidney disease and they proposed a method based on Extreme Gradient Boost(XGBoost) that satisfies three feature selection techniques for better and precise diagnosis.

2.2 References

[1] L. Antony et al., "A Comprehensive Unsupervised Framework for Chronic Kidney Disease Prediction," in IEEE Access, vol. 9, pp. 126481-126501, 2021, doi: 10.1109/ACCESS.2021.3109168.

[2] S. M. M. Elkholy, A. Rezk and A. A. E. F. Saleh, "Early Prediction of Chronic Kidney Disease Using Deep Belief Network," in IEEE Access, vol. 9, pp. 135542-135549, 2021, doi: 10.1109/ACCESS.2021.3114306.

[3] J. Qin, L. Chen, Y. Liu, C. Liu, C. Feng and B. Chen, "A Machine Learning Methodology for Diagnosing Chronic Kidney Disease," in IEEE Access, vol. 8, pp. 20991-21002, 2020, doi: 10.1109/ACCESS.2019.2963053.

[4] Tekale, S., Shingavi, P., Wandhekar, S., & Chatorikar, A. (2018). Prediction of chronic kidney disease using machine learning algorithm. International Journal of Advanced Research in Computer and Communication Engineering, 7(10), 92-96.

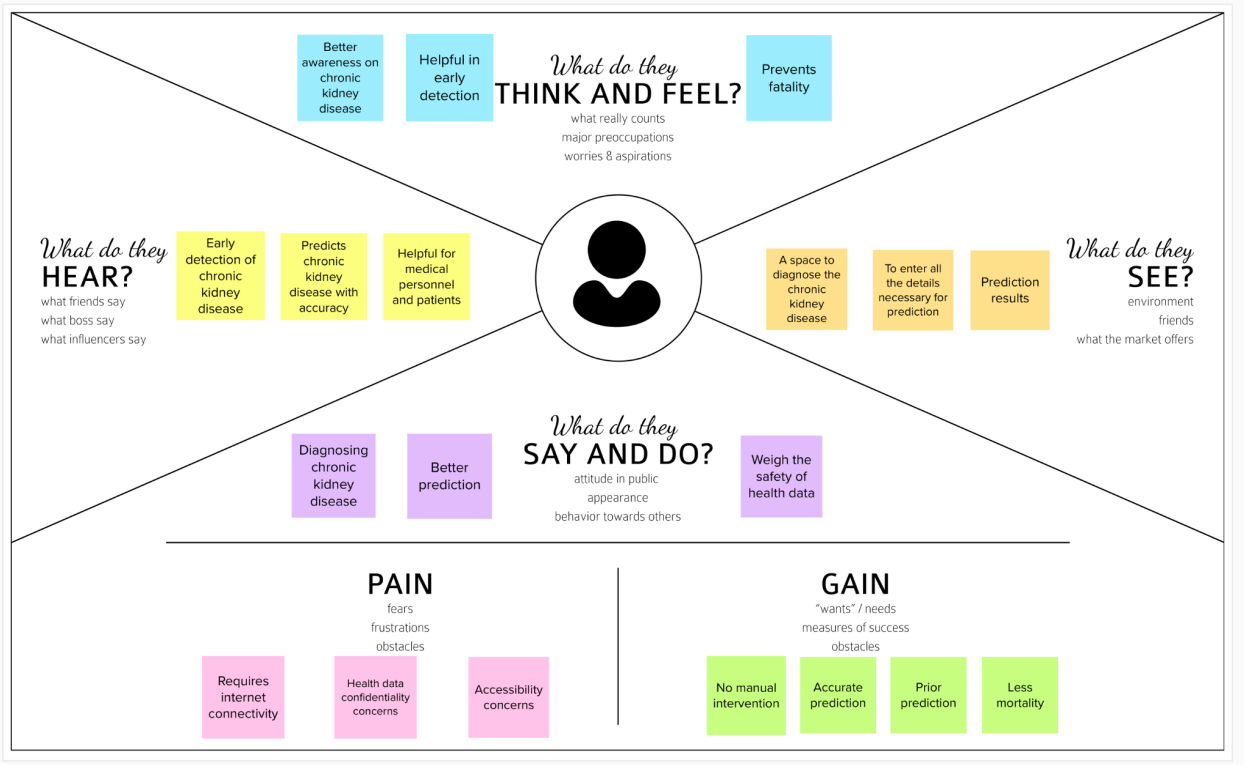
[5] I. U. Ekanayake and D. Herath, "Chronic Kidney Disease Prediction Using Machine Learning Methods," 2020 Moratuwa Engineering Research Conference (MERCon), 2020, pp. 260-265, doi: 10.1109/MERCon50084.2020.9185249.

[6] A. Ogunleye and Q. -G. Wang, "Enhanced XGBoost-Based Automatic Diagnosis System for Chronic Kidney Disease," 2018 IEEE 14th International Conference on Control and Automation (ICCA), 2018, pp. 805-810, doi: 10.1109/ICCA.2018.8444167

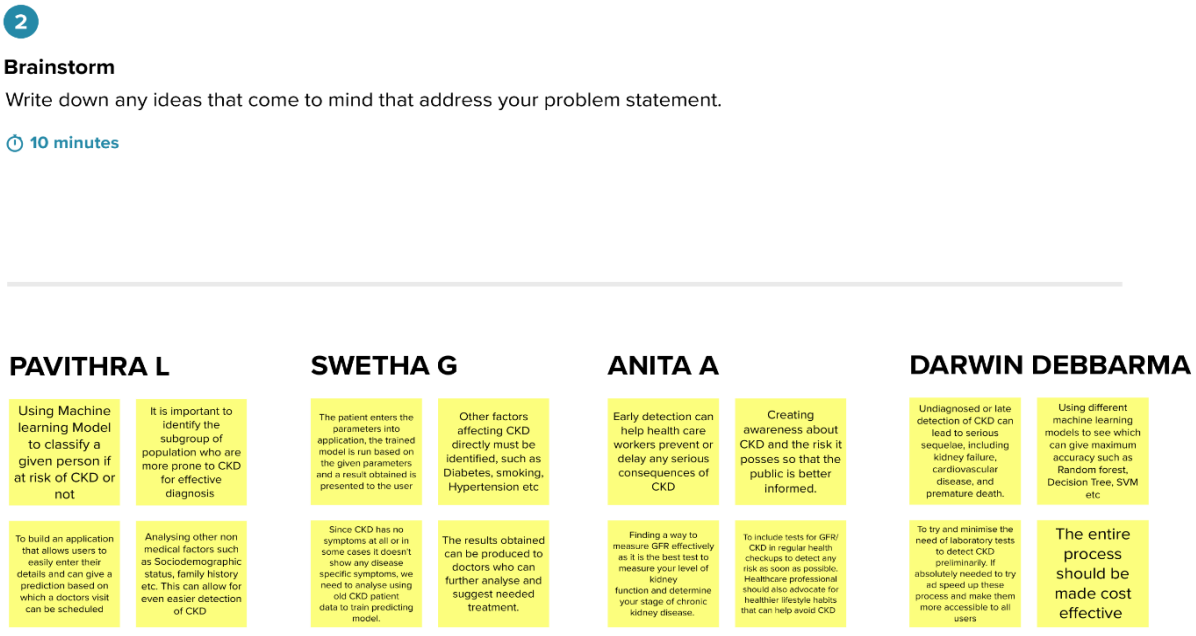
2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

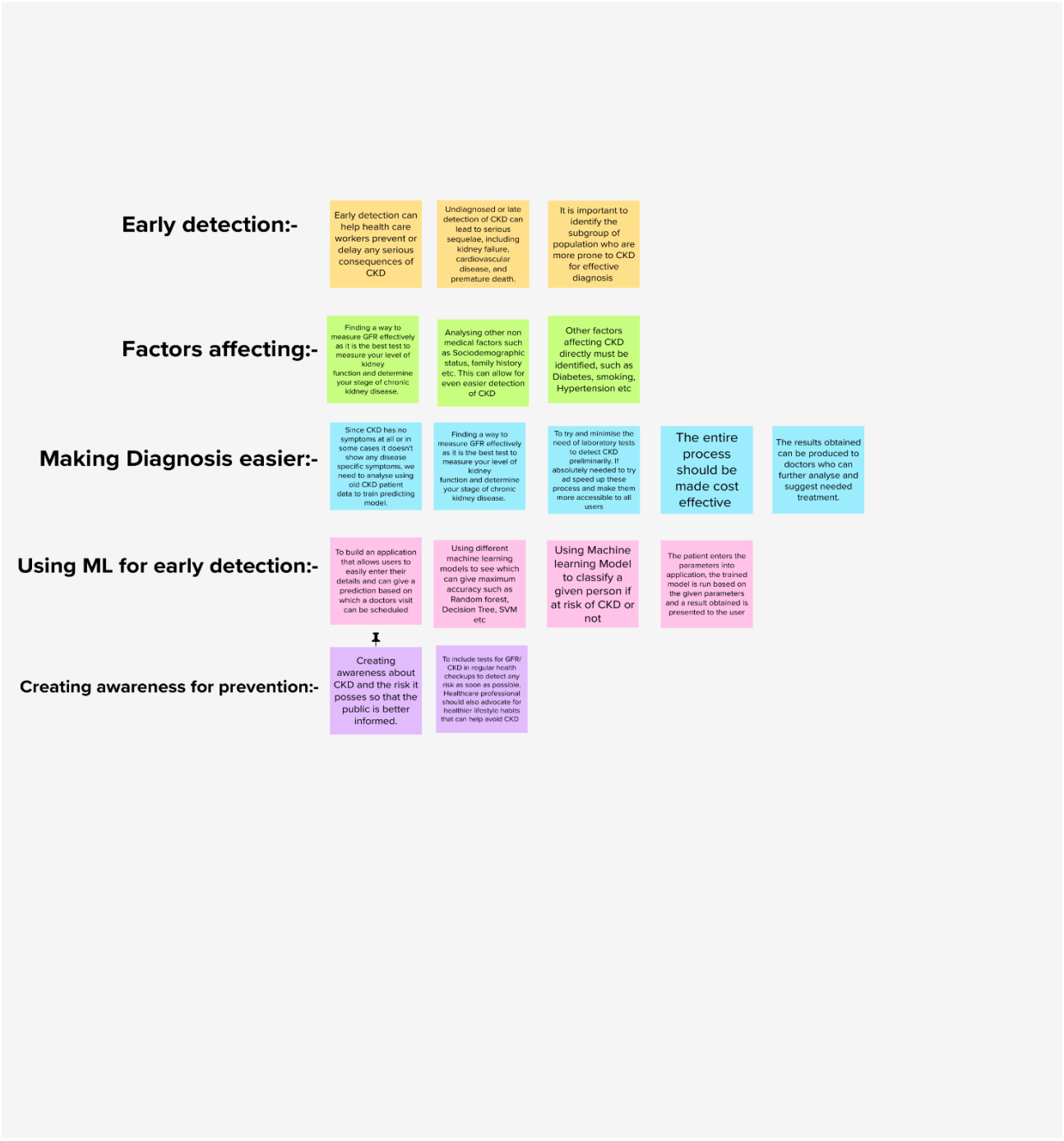


3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes



4

Prioritize

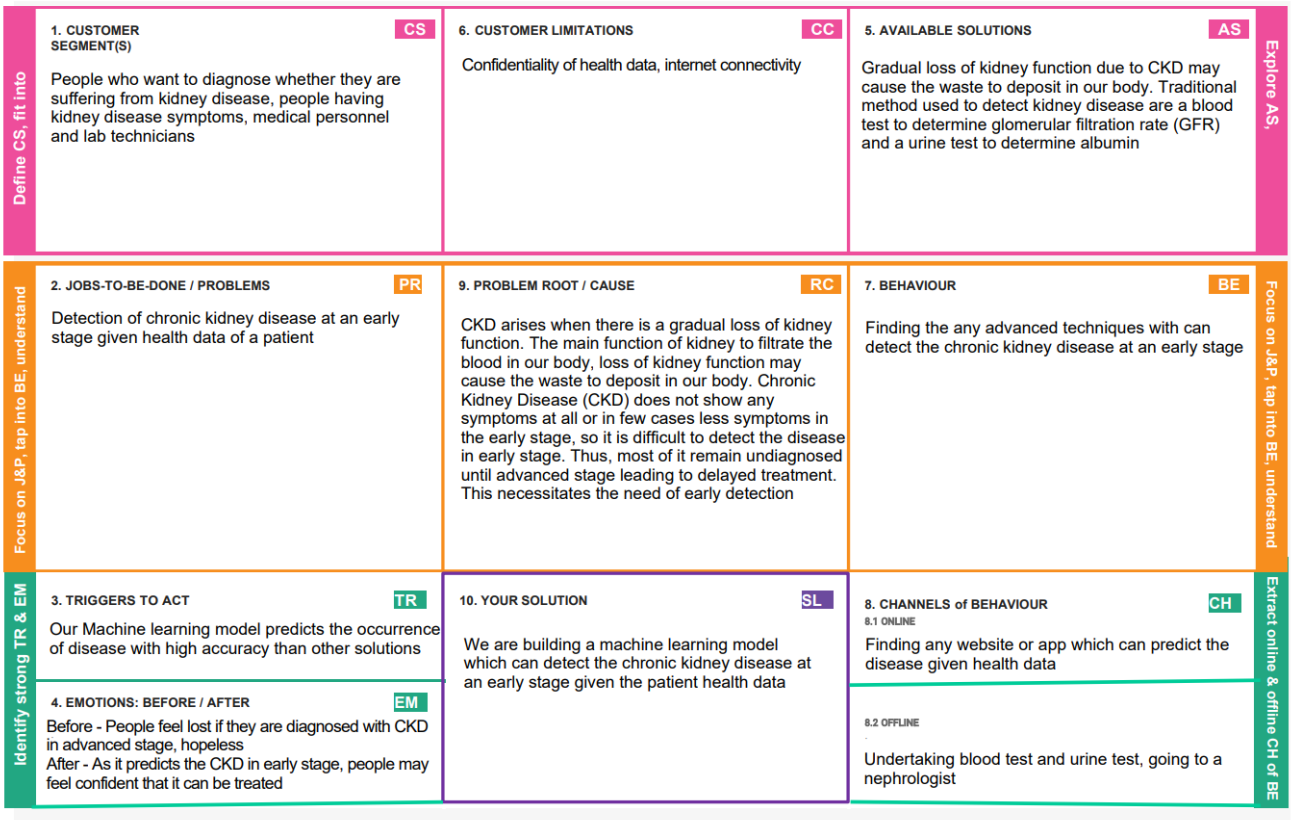
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop an application to detect chronic kidney diseases in its early stages with patient health data.
2.	Idea / Solution description	Given the required patient data, we propose a solution wherein a machine learning model is leveraged to diagnose CKD at the earliest.
3.	Novelty / Uniqueness	Convenient user interface for first time users. K-nearest neighbours classification for better accuracy.
4.	Social Impact / Customer Satisfaction	Users can be aware of the presence of CKD at the earliest and receive appropriate medical treatment
5.	Business Model (Revenue Model)	Transaction based model, where the user of the product can be casual users or hospitals and laboratories.
6.	Scalability of the Solution	The developer can scale the solution as per needs and extend functionality to detect other diseases.

3.4 Problem solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	Home Page	Icon and brief Description of what the application does.
FR-2	Dashboard -CKD Prediction	Parameters required to make prediction must be entered
FR-3	Dashboard-Results	Generate Report with Results based on the prediction made using entered parameters

4.2 Non functional requirements

Following are the non-functional requirements of the proposed solution.

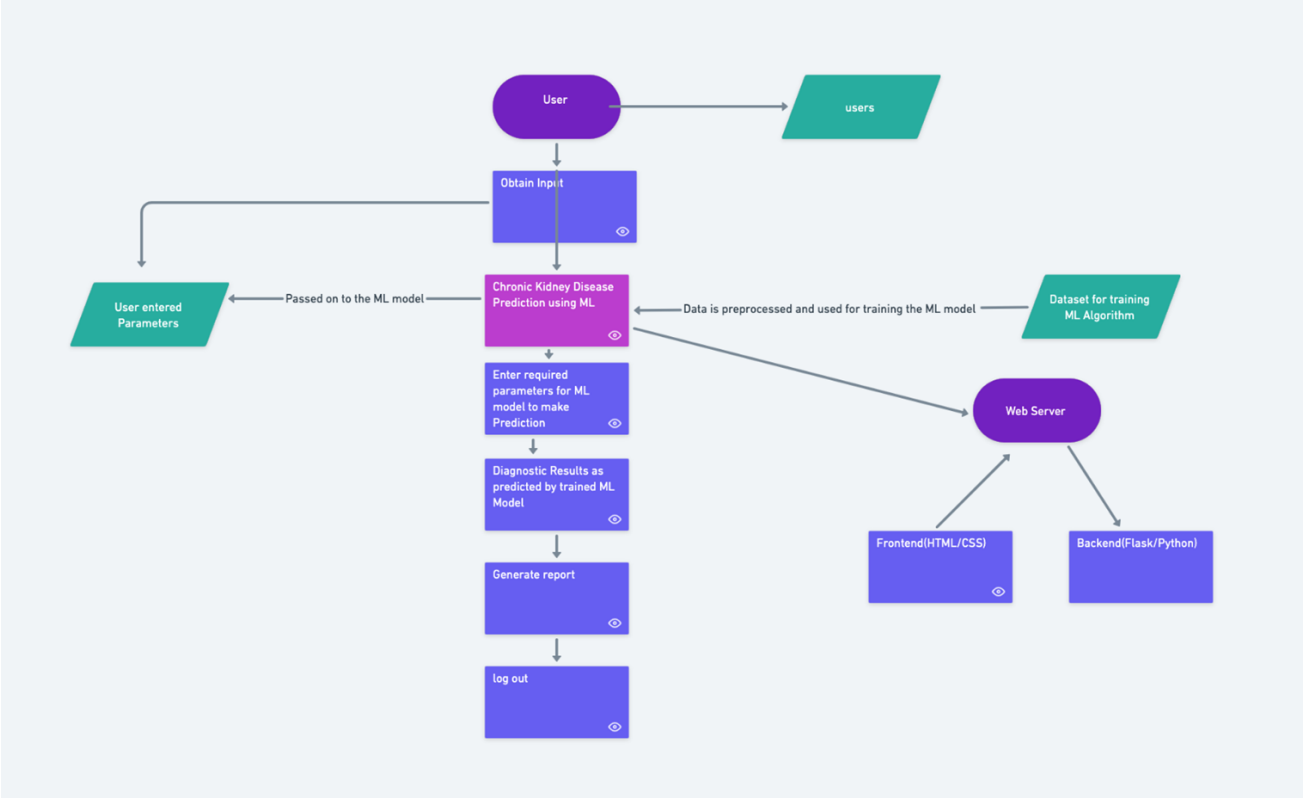
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	The UI must be intuitive and should not require much technical knowledge to traverse through the application. This ensures all the features of the application are used to their best capability

NFR-2	Security	Access to sensitive information such as personal and medical details must be protected and stored in a secure database
NFR-3	Reliability	The accuracy of the prediction of the Machine Learning model should be sufficiently high so that the results obtained are trustworthy. The probability of failure or failure rate can be calculated to assess the reliability of the system
NFR-4	Performance	The ease of interactivity and reducing the load time of prediction/diagnosis can help with increased performance
NFR-5	Availability	The application can be accessed by users with an internet connection from anywhere at any time
NFR-6	Scalability	The website should be able to handle influx or reduced traffic at any given point with low latency

5. PROJECT DESIGN

5.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored



5.2 Solution & Technical Architecture

Technical architecture

Diagram 1

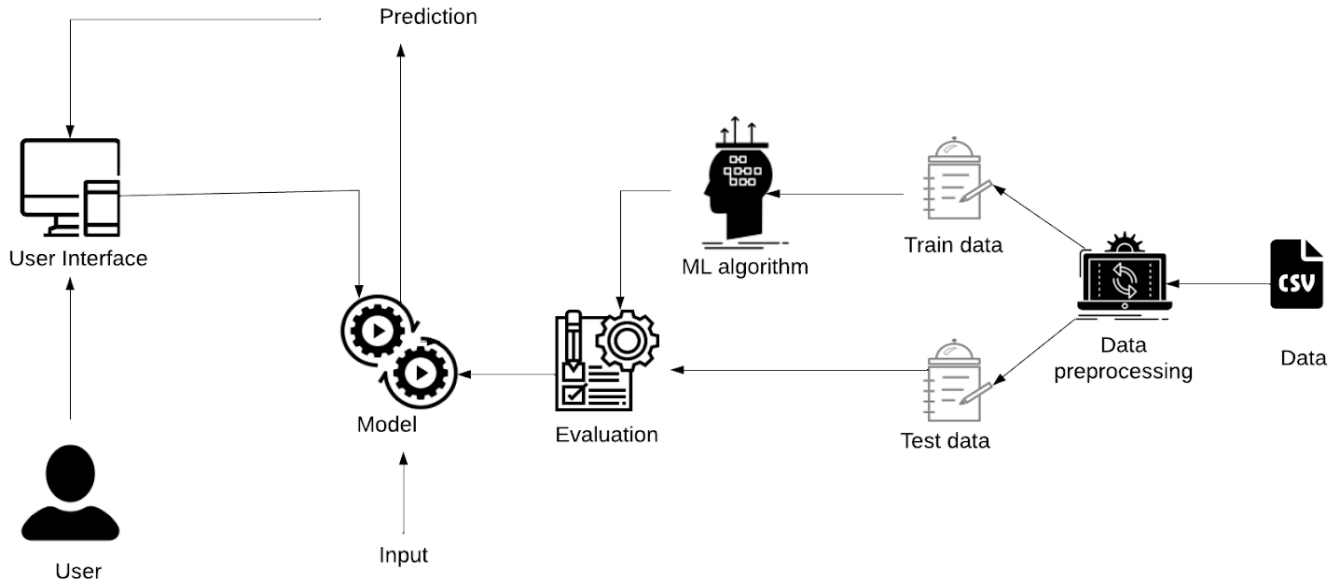


Diagram 2

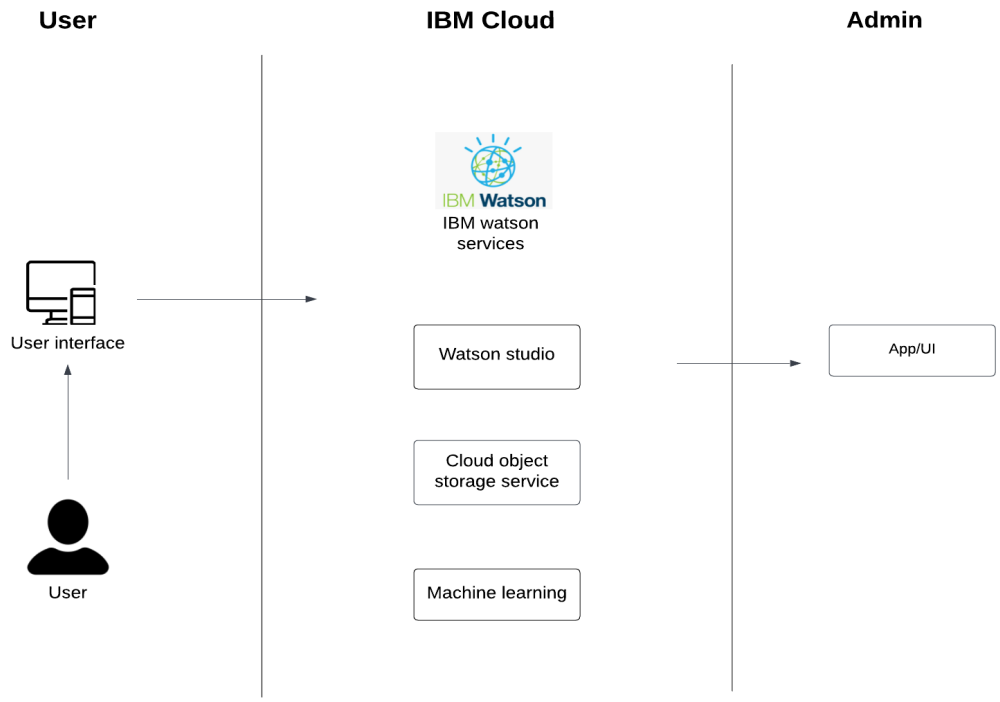


***Table-1**: Components & Technologies:*

S.No	Component	Description	Technology
1	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript.
2	Data preprocessing	Cleaning the dataset (Handling the missing values etc)	Java / Python
3	Splitting the dataset	Splitting the dataset into train and test data	Java / Python
4	Test the model	Testing the model using test data	Java / Python
5	Evaluation	Evaluating the built model (accuracy, confusion matrix)	Java / Python
6	Machine Learning Model	The ML model which takes the input parameter given by the user and predict the result	IBM watson Machine learning service
7	Infrastructure (Server / Cloud)	Application Deployment on Cloud	IBM watson services (Cloud object storage service, watson studio, machine learning)

***Table-2**: Application Characteristics:*

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Flask mico web framework used for developing web application
2	Scalable Architecture	The website should be able to handle influx or reduced traffic at any given point	
2	Availability	The application can be accessed by users with an internet connection from anywhere at any time (hostin it in cloud)	IBM cloud

3	Performance	Multiple users should be able to access the application at the same time	

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (web user)	Home page	USN 1	As a user, I can see the home page of the application	I can access the home page of the application	Medium	Sprint-3
	Dashboard	USN 2	As a user, I must lead to a page where it asks the necessary parameters required to make the prediction	I can enter the input parameters	High	Sprint-3
	Result	USN 3	As a user, I must lead to a page where I can view the result generated by the application (Prediction result – Positive/Negative)	I can view negative/ positive results produced after diagnosis	High	Sprint-3
Administrator		USN 4	As an administrator, I should collect the data and pre-process the data	I should collect the data and pre-process the data	High	Sprint-1

		USN 5	As an administrator, I should identify the most significant factors that lead to CKD based on the present trend and come up with the input parameter that should be given by the user for CKD prediction	I must identify input parameters required for CKD prediction	High	Sprint-2
		USN 6	As an administrator, I must use the most suitable ML model for detection of CKD	I should efficiently train the ML model	High	Sprint-2
		USN 7	As an administrator, I should develop a web application using Flask (Creating homepage, input page and result page using HTML and integrating ML model)	I should develop the web application	High	Sprint-3
		USN 8	As an administrator, I must ensure that the application is live and is accessible on any device with internet connectivity (Deploying on IBM cloud)	I should deploy the model on cloud	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User story number	User Story / Task	Story points	Priority	Team Members
Sprint-1		USN-4	As an administrator, I should collect the data and pre-process the data	4	High	Pavithra, Swetha
Sprint-2		USN-5	As an administrator, I should identify the most significant factors that lead to CKD based on the present trend and come up with the input parameter that should be given by the user for CKD prediction	5	High	Pavithra, Anita Priyadharshini
Sprint-2		USN-6	As an administrator, I must use the most suitable ML model for detection of CKD	4	High	Pavithra, Swetha
Sprint-3	Homepage	USN-1	As a user, I can see the homepage of the application	4	Medium	Darwin Debbarma, Swetha

Sprint-3	Dashboard	USN-2	As a user, I must lead to a page where it asks the necessary parameters required to make the prediction	7	High	Darwin Debbarma, Anita Priyadharshini
Sprint-3	Result	USN-3	As a user, I must lead to a page where I can view the result generated by the application (Prediction result – Positive/Negative)	8	High	Anita Priyadharshini, Swetha
Sprint-3		USN-7	As an administrator, I should develop a web application using Flask (Creating homepage, input page and result page using HTML and integrating ML model)	8	High	Pavithra, Swetha, Anita Priyadharsini, Darwin Debbarma
Sprint-4		USN-8	As an administrator, I must ensure that the web application is live and is accessible on any device with internet connectivity (Deploying on IBM Cloud)	7	High	Darwin Debbarma, Pavithra

6.2 Sprint delivery schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

7. CODING AND SOLUTIONING

7.1 Prediction

This helps users to detect chronic kidney disease at an early stage. The user has to enter the necessary data like blood urea level, blood glucose random, coronary artery disease, anemic, pus cells, red blood cells, diabetes mellitus, pedal edema. When the user enters these details, it will give out the results like whether they are suffering from chronic disease or not.

Code :

```
selected_columns=['red_blood_cells','pus_cells','blood_glucose_r
andom','blood_urea','pedal_edema','anemia',
'diabetesmellitus','coronary_artery_disease']
```

```
x=pd.DataFrame(data,columns=selected_columns)

y=pd.DataFrame(data,columns=['class'])

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2
,random_state=2)

lgr=LogisticRegression()

lgr.fit(x_train, y_train)

y_pred=lgr.predict(x_test)
```

8. TESTING

8.1 Test cases

Test case ID	Feature Type	Component	Test Scenario	Steps to Execute	Expected Result	Status
Home_Page_TC_001	Functional	Home Page	Verify user is able to see the Home page when user enters the site	1. Enter URL and press enter 2. Verify Home page is displayed or not	Home page should be displayed	Pass
Home_Page_TC_002	UI	Home Page	Verify the UI elements in Home page	Enter URL and press enter Verify Home page with below elements: a. Image display b. Text display c. Home Page button d. Prediction button	Application should showl below UI elements: a. Image display b. "Chronic Kidney Disease Prediction" text	Pass

					display c. Home Page button d. Prediction button	
Home_Page_TC_003	UI	Home Page	Verify user is able to navigate to the Home page	1. Enter URL and press enter 2. Verify Home page is displayed or not 3. Click Home button to navigate to Home page	Home page should be displayed	Pass
Prediction_Page_TC_001	UI	Prediction Page	Verify user to able to navigate to the Prediction page	1. Enter URL and press enter 2. Enter Home page 3. Click on Prediction button to navigate to Prediction page	Prediction page should be displayed	Pass
Prediction_Page_TC_002	Functional	Prediction Page	Verify user is able to enter details	1. Enter URL and press enter 2. Enter Home page 3. Click on Prediction button to navigate to Prediction page	Details are received	Pass

				4. Enter details in input boxes 5. Press submit		
Prediction_Page_TC_003	Functional	Prediction Page	Verify user is able to enter details	1. Enter URL and press enter 2. Enter Home page 3. Click on Prediction button to navigate to Prediction page 4. Enter details in input boxes 5. Select Yes/No in drop down boxes 6. Press Submit button	Details are received	Pass
Result_Page_TC_001	Functional	Result Page	Verify user is able to view prediction results	1. Enter URL and press enter 2. Enter Home page 3. Click on Prediction button to navigate to Prediction Page Enter details in input boxes 5. Select Yes/No in drop down boxes 6. Press Submit button	Result page should be displayed	Pass

8.2 User Acceptance Testing

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	3	1	1	18
Fixed	14	4	2	2	12
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	23	7	3	3	30

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	13	0	0	13
Client Application	21	0	0	21
Security	2	0	0	2
Exception Reporting	1	0	0	1
Final Report Output	8	0	0	8
Version Control	1	0	0	1

9. RESULTS

9.1 Performance Metrics

The results of the Chronic Kidney disease prediction application are evaluated based on metrics to verify the correct working of the application against the business requirements.

Accuracy Score

Accuracy is the simple ratio between the number of correctly classified points to the total number of points.

Confusion Matrix

Confusion Matrix is a summary of predicted results in specific table layout that allows visualization of the performance measure of the machine learning model for a binary classification problem (2 classes) or multi-class classification problem (more than 2 classes).

Confusion Matrix composition:

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

Precision Score

Precision is the fraction of the correctly classified instances from the total classified instances. Precision is given by the formula:

$$Precision = \frac{TP}{TP+FP}$$

```
print("Accuracy score:",accuracy_score(y_test,y_pred))
confusion_mat = confusion_matrix(y_test,y_pred)
print("Confusion matrix:\n",confusion_mat)
print("Precision score: ",precision_score(y_test,y_pred))
```

```
Accuracy score: 0.925
Confusion matrix:
 [[48  6]
 [ 0 26]]
Precision score:  0.8125
```

Results

10. ADVANTAGES AND DISADVANTAGES

10.1 Merits

- The application can help diagnose Chronic Kidney Disease in its early stages.
- It has a user-friendly interface that is intuitive and easy to learn for first time users.
- The application predicts disease with a 92% accuracy.

10.2 Demerits

- The user details entered are not secure
- There is no log of user sessions

11. FUTURE SCOPE

The accuracy of the machine learning model could be improved. The existing application only predicts chronic kidney disease based on user provided data. This can be extended to predict more diseases and create awareness among people so that early diagnosis can be performed. The details entered by users can be encrypted and more secured to ensure that sensitive health data does not fall in the wrong hands.

12. CONCLUSION

While disease prediction applications provide accurate results without the help of a medical personnel, they also have the added advantage of making healthcare accessible to common users. People can now be in control of their data and be informed about their health with the swipe of a finger. These applications are resource efficient, cost effective and time conserving, making them the need of the hour. With a more accurate machine learning model, the application can raise its credibility and be of help to common users and medical health professionals. Future research can be pointed in the same direction.

13. APPENDIX

Source code

```
import numpy as np
import pandas as pd
from flask import Flask,request,render_template
import pickle as pk
import requests
import json


API_KEY = "hSkuQmck2PyDdxU8ArFuKk6fWcQgVmt3JZNVmDwRhojl"
token_response =
requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type":
'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization':
'Bearer ' + mltoken}


app=Flask(__name__)
model=pk.load(open('CKD.pkl','rb'))

@app.route('/')
def home():
    return render_template('homepage.html')
@app.route('/Prediction',methods=['POST','GET'])
def prediction():
     return render_template('indexpage.html')
@app.route('/Home',methods=['POST','GET'])
def my_home():
     return render_template('homepage.html')
@app.route('/predict',methods=['POST'])
def predict():
    input_features=[float(x) for x in request.form.values()]
```

```
    payload_scoring = {"input_data": [{"field":
[['blood_urea','blood_glucose_random','coronary_artery_disease',
'anemia','pus_cell','red_blood_cells','diabetesmellitus','pedal_
edema']], "values": [input_features]}]}

    response_scoring =
requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deploymen
ts/6e8755e6-7e54-44e2-99a6-44959f643c2f/predictions?version=2022
-11-14', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    predictions=response_scoring.json()
    pred=predictions['predictions'][0]['values'][0][0]
    if(pred==1):
        return render_template('predictionNo.html')
    else:
        return render_template('predictionYes.html')


if __name__ == '__main__':
    app.run(debug=True)
```

Github & Project Demo link

Github link : https://github.com/IBM-EPBL/IBM-Project-41652-1660643754

Demo link :

https://github.com/IBM-EPBL/IBM-Project-41652-1660643754/blob/main/Final%20Deliverables/Demo%20video.mp4