NAME: SANTOS OMONDI OKELLO

REG NO: 19/05036

SUP: MR COLLINS ONDIEK

DISEASE PREDICTION WEB BASED SYSTEM

KCA UNIVERSITY BSC.IT

TWO SEMESTER PROJECT

CHAPTER ONE

Background

Use of computing in the field of medicine can be seen from the early 1950s. However, the first applications of systems using AI in medicine can only be seen during the 1970s through expert systems such as INTERNIST-I, MYCIN, ONCOSIN. The application of artificial intelligence in medicine was mostly limited in Kenya before 2000. An international conference was organized on September 2010 in Nairobi County to provide a clear view how to help patients with the act of first Aid across the world, BLIZZ health care was concerned with the disease prediction system that would bring a brighter change to the remote future.

Relevance of the project

The major problem with using AI for the diagnosis of disease is the lack of data for training predictive models. Though there is vast amount of data including mammograms, genetic tests, and medical records,

they are not open to the people who can make use of them for research. The project tries to cover up and identify various way that patients can be help with necessarily going to the hospital.

Problem statement

The primary goal is to develop a prediction engine which will allow the users to check whether they have diseases like malaria, tuberculosis, typhoid, diabetes or heart disease et cetera sitting at home when feeling sick. The user don't need visit the doctor unless he or she has a strong disease that required physical checkup, for further treatment. The prediction engine requires a large dataset and efficient machine learning algorithms to predict the presence of the disease. Pre-processing the dataset to train the machine learning models, removing redundant, null, or invalid data for optimal performance of the prediction engine

Objectives

The primary of this project is to predict the disease from the given symptoms create and monitors a health profile of every individuals patients

In order to predict disease several factors has been consider such as body mass index, cholesterol level, blood sugar, blood pressure and so on.

It also recommend necessary precautionary measures required to treat the predicted disease

Diseases that can be predicted using machine learning are simple cart, naïve Bayes, svm and random forest are used for prediction and analyze the diabetes data

The secondary aim is to develop a web application that allows users to predict heart disease, malaria, tuberculosis, diabetes et cetera utilizing the prediction engine

To implement the IT in real world problems.

To help general practice doctors, nurses, nursing students and to assist the eye patients as first aid diagnosis

Scope of the project

The disease diagnosis system will permit end-users to predict disease like malaria, tuberculosis, typhoid, heart disease et cetera

Growth of AI systems

Artificial Intelligence is one of the hottest topics today. The revenue for cognitive and artificial intelligence systems is expected to hit $12.5 billion

Regression method fall within the category of supervised ML, They help to predict of explain a particular numerical value based on a set of prior data. For example predicting the disease based on previous disease result data inserted

Availability of doctors and chat bot

Other than disease diagnosis, artificial intelligence can be used to streamline and optimize the clinical process. There is only one doctor for over 1600 patients in Kenya. AI health assistants can help in covering large part of clinical and outpatient services freeing up doctor's time to attend more critical cases. Chat bot like "SH chat bot" can assist patients by understanding what disease to cure' symptoms and suggest easy-to-understand medical information about their condition

Internet of things (IOT), Healthcare and machine learning

Increasing use of Internet of Things has promising benefits in healthcare.

Dynamically collecting patient data using remote sensors can help in early detection of health problems and aid in preventive care

CHAPTER TWO

Literature Review

The following chapters give an overview of the various methodologies used by various authors for disease prediction using machine learning methodologies. We can observe that there is fine comparison made between 5 major machine learning algorithms whether they are able to predict the presence of the disease with a greater accuracy, achieving optimal performance. The research efforts presented by the authors in the following papers are focused in developing and evaluating a web-based tool for disease prediction

Author: Priyanka Sonar, Prof. K. JayaMalini

Published In: Proceedings of the Third International Conference on Computing

Methodologies and Communication (ICCMC 2019)

The authors have used Machine Learning approaches to predict diabetes [1]

Diabetes is one of lethal diseases in the world. It is additional an inventor of various varieties of disorders for example: coronary failure, blindness, urinary organ diseases etc. In such a case the patient is required to visit a diagnostic centre, to get their reports after consultation. Due to every time they must invest their time and currency. But with the growth of Machine Learning methods we have got the flexibility to search out an answer to the current issue, we have got advanced system mistreatment information processing that has the ability to forecast whether the patient has polygenic illness or not. Furthermore, forecasting

the sickness initially ends up in providing the patients before it begins vital. Information withdrawal has the flexibility to remove unseen data

Authors: Samrat Kumar Dey, Ashraf Hossain and Md. Mahbubur Rahman

Published In: 2018 21st International Conference of Computer and Information Technology (ICCIT)

The authors design and develop a web application to predict diabetes [2]

Diabetes is caused due to the excessive amount of sugar condensed into the blood. Currently, it is considered as one of the lethal diseases in the world. People all around

CHAPTER THREE

Methodology

RESEARCH METHODOLOGY

QUALITATIVE METHOD

Interviewing

I did interview in northern areas of kenya and found out that people cannot access hospitals as needed for help, This encourages me to provide people with system that will help them provide themselves with a clear first aid of any sort.

Observation

This is a clear show that my system and ideological perspective should exist in human daily life. Most of us have problem of disease help depending on geographical location the patient is located, but I believe my system will have a positive impact to all

Questionnaire

I did some few questionnairing and found out that people luck a digitized system that detect disease and give a possible solution to cure the disease. Example. John Kamau said that It would be better if the system exist to help people figure out what is happening to them because not all people have knowledge of what to do when sick. Documentation of first aid help may help alot

DEVELOPMENT METHODOLOGY

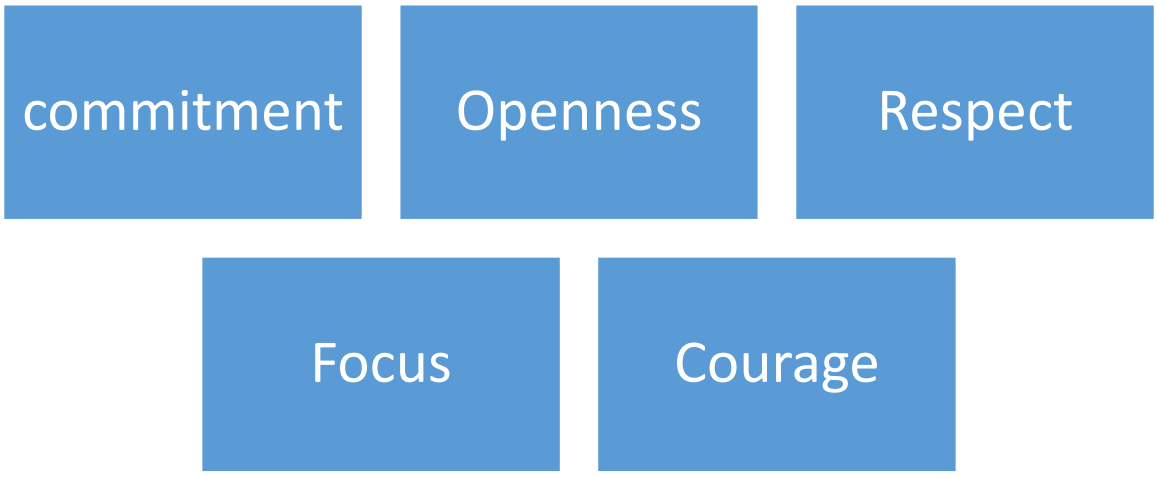
AGILE METHODOLOGY

Agile is a process by which a team can manage a project by breaking it up into several stages and involving constant collaboration with stakeholders and continuous improvement and iteration at every stage. It promotes continuous iteration of development and testing throughout the software development life cycle of the project. Both development and testing activities are concurrent

AGILE SCRUM METHODOLOGY

SCRUM is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment. Scrum encourages learner to learn through experiences, self-organize while working on a problem. Each iteraction consists of two to four sprints, where the goal of each is to build the most important features first and come out with a potentially deliverables product.

SCRUM VALUES



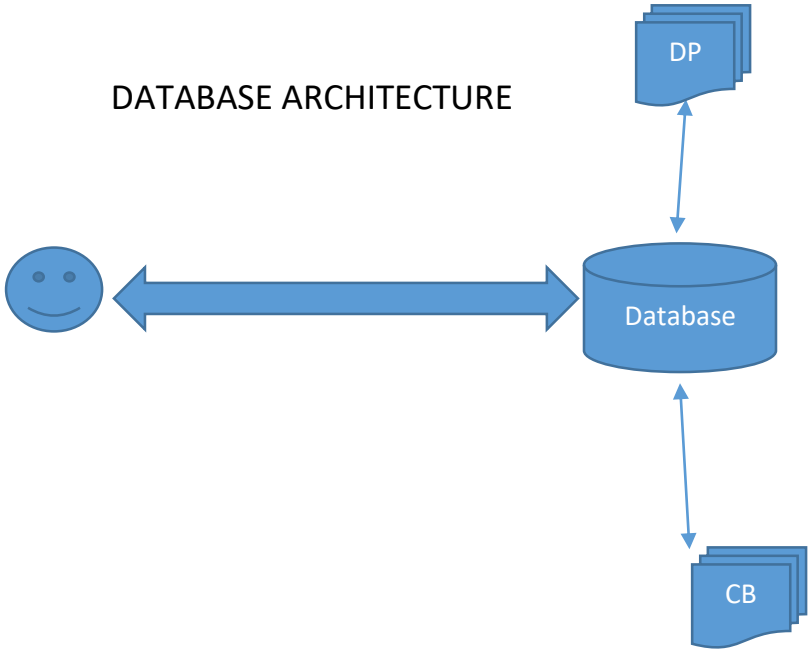
THE MAIN ARTEFACT

Product Backlog is the master list of work that needs to get done maintained by the product owner or product manager.

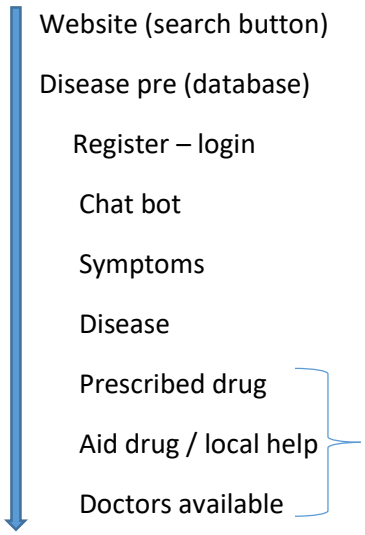
Sprint Backlog is the list of items, user stories, or bug fixes, selected by the development team for implementation in the current sprint cycle.

Increment (or Sprint Goal) is the usable end-product from a sprint

DATABASE ARCHITECTURE



System overview



SOFTWARE REQUIREMENT SPECIFICATIONS

FUNCTIONAL RECQUIREMENTS

Other system features include:

DISTRIBUTED DATABASE Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as network

CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as front-end), and the server is the DBMS (also known as the back-end). A client/server system is a distributed system in which, Some sites are client sites and others are server sites. All the data resides at the server sites. All application execute at the client sites

NON FUNCTIONAL RECQUIREMENT

PERFOMANCES RECQUIREMENTS

The steps involved to perform the implementation of garage database are as listed below

E-R DIAGRAM The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is the used to organize data as a relation, normalizing relation and finally obtaining a relation database.

ENTITIES: Which specify distinct real-world items in an application PROPERTIES: Which specify properties of an entity and relationships

RELATIONSHIP: Which connect entities and represent meaningful dependencies between them:
NORMALIZATION The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

SOFTWARE DESIGN AND IMPLEMENTATION

Software to be Implemented Implementation plan

The software to be Implemented here will be a disease prediction system. This system will be web based, that is able to be accessed from any device that has internet connection. Name of the website: **Hivedesign.de**

Model Assumptions Here I outline of model design of the approach I will take in the software implementation life cycle that clearly shows the implementation objectives of the system. This is to guide to define the test coverage and testing scope to clear the picture of the project at any instance, to ensure a implementation activity is not missed.

Choose algorithm (NAÏVE BAYES)

Naive Bayes is a type of machine learning algorithm that is based on Bayes' theorem, which states that the probability of a hypothesis (such as a disease diagnosis) given some evidence (such as patient data) is proportional to the prior probability of the hypothesis and the likelihood of the evidence given the hypothesis. In the context of disease prediction, Naive Bayes can be used to predict the probability of a disease diagnosis based on patient data, such as demographic information, medical history, and laboratory test results. The algorithm assumes that the features in the patient data are independent of each other, which is known as the "naive" assumption. Naive Bayes can be a useful algorithm for disease prediction in certain scenarios, such as when the dataset is small or the relationship between the features is not well understood. It is also relatively fast and simple to implement, making it a good choice for applications with limited computational resources. However, it is important to keep in mind that the performance of Naive Bayes can be impacted by the validity of the "naive" assumption, and it may not be the best choice for datasets with complex relationships between the features. The development team may need to evaluate multiple algorithms to determine which one is best suited for the system's goals and objectives.

Train model Involves the following steps

▪ Prepare the data: The first step is to gather and preprocess the patient data that will be used to train the model. This may involve cleaning the data, imputing missing values, and normalizing the features

. ▪ Establish the prior probabilities: The next step is to establish the prior probabilities of each disease class in the training data. This can be done by counting the number of instances of each class in the training data and dividing by the total number of instances.

▪ Establish the likelihood probabilities: For each feature in the patient data, the likelihood probabilities of each feature given each disease class need to be calculated. This can be done Implementation plan 7 by counting the number of instances of each feature in the training data for each class and dividing by the total number of instances of each class.

▪ Train the model: The model is then trained using the prior and likelihood probabilities. This involves inputting new patient data into the model and using the probabilities to calculate the posterior probabilities of each disease class.

▪ Evaluate the model: Once the model has been trained, it is important to evaluate its performance on a separate dataset, such as a validation or test set. This can be done by comparing the model's predictions to the actual disease outcomes and calculating metrics such as accuracy, precision, and recall.

Evaluate the mode

l Evaluating a Naive Bayes model involves comparing its predictions to the actual disease outcomes in a separate dataset, such as a validation or test set. The following metrics can be used to evaluate the performance of the model:

▪ Accuracy: The accuracy of the model is the proportion of correct predictions made by the model. It is calculated as the number of correct predictions divided by the total number of predictions.

▪ Precision: Precision is the proportion of true positive predictions made by the model, i.e. the proportion of instances that the model correctly predicted as having the disease.

▪ Recall: Recall is the proportion of actual positive instances that the model correctly predicted, i.e. the proportion of instances with the disease that the model correctly identified.

▪ F1 Score: The F1 score is a composite metric that combines precision and recall into a single value. It is the harmonic mean of precision and recall, and is a good measure of the overall performance of the model.

TEST PLAN

Software to be tested

The software to be tested here will be a disease prediction system.

This system will be web based, that is able to be accessed from any device that has internet connection. Name of the website: **Hive design healthcare**

Test plan

Testing strategy Here I outline a high-level description of the approach I will take in the software testing life cycle that clearly shows the testing objectives of the system. This is to guide to define the test coverage and testing scope to clear the picture of the project at any instance, to ensure a test activity is not missed.

Unit testing This is the first and most important level of testing. It starts from the moment a unit of code is written. Every unit is tested for various scenarios to detect and fix bugs during early stages of software development lifecycle. For unit testing the following strategy will be used

i. Creation of test cases and test data

ii. Creation of scripts to run the test cases wherever applicable

iii. Execution of test cases once the code is ready

iv. . Fixing of the bugs of present and retesting of the code

v. Repletion of the test cycle until the confidence level is that there are no significant bugs remaining.

Tests that will be performed during unit testing include.

• Testing the user interfaces to ensure that the inputs produce the expected outputs, and that information is properly flowing into the program unit.

• White boxing testing will be used to ensure the functionalities of the system behave as they are expected to.

• Error handling paths to review if errors are handled properly by them or not.

• Independent paths are tested to see that they are properly executing their task and terminating at the end of the program

. • Local data structures are tested to inquiry if the local data within the module is stored properly or not.

Integration testing

This is the step where software modules are integrated logically and tested as a group. Here I will be integrating different components that are interrelated with the aim to exposed defects in their interactions. This is because even though unit testing is done independently defects will still exist. Integration testing is done when all modules code is completed and successfully integrated. For this testing I will follow an incremental approach this is because makes it easier to locate a fault, obtain a prototype earlier, and no time is wasted waiting for all modules to be developed like in big bang approach.

Validation testing

Validation testing is in place to determine if the existing system complies with the system requirements and performs the dedicated functions for which it is designed along with meeting the goals and needs of the organization. In this case it is to test that the disease prediction system complies with requirements and performs its functions correctly in relation to healthcare of its users i.e., business logic.

Here is where the critical functionalities of the system are tested for my case the critical functionalities of my system include correct prediction of illness, security to ensure no data is compromised, availability of the system to users, usability to ensure all level of individuals can navigate the system. Validation process will entail the following steps

i. Gather healthcare requirements for validation testing from the end user.

ii. ii. Prepare the business plan and send it for the approval to the onsite/stakeholders involved i.e the partnering hospital for my project

iii. . iii. On approval, begin to write the necessary test cases and send them for approval.

iv. iv. Once approved I begin to complete testing with the required software, environment and send the deliverables as requested by the client.

v. v. Upon approval of the deliverables, User acceptance testing is done by the client.

vi. vi. After that, the software goes for production.

High order testing

High order testing is done because unit testing and integration testing are not always sufficient to identify and locate all the vulnerabilities of a software and that's why there is a need for further testing i.e., high order tests. They are mostly done by third parties to ensure the test findings match with those of the owners of the software. High Order Test ensures the fulfilment of all sorts of requirement, criterion, at each step of the development, to produce a software product of desired quality. High order testing is of various types that include the following.

• **System Testing** Herein the focus lies on the integrated system. The main aim behind System Testing is to completely work out and analyze the software to identify the loopholes that can cause errors. Test plan 10

• **Recovery Testing** In Recovery Testing the software is forced to fail under various scenarios and in different ways. The idea is to corroborate that the recovery process occurs as expected and that there are no deviations from it

. • **Security Testing** These days security is a very big issue. If a software is open to unauthorized and illegal access, then it is bound to cause concern. To substantiate and authenticate the fact that the software is well secured it is necessary to conduct comprehensive Security Testing.

• **Stress Testing** The software is now tested at varying frequencies, with abnormal data inputs and haphazard volume to validate its breaking point. The idea is to judge its complete functionality in the light of unpredictable and inconsistent consumer or user behavior.

• **Performance Testing** Finally, the software needs to be checked and tested for its range of performance to know its compatibility and functionality

. • **Function Testing** This testing is used to locate out the discrepancies, between the actual working and behavior and what was intended, from the user's point of view.

• **Integration Testing** Ensures the proper working of the modules, after getting integrated.

• **Acceptance Testing** It is used to verify and validate a software product, against the specified requirements and specifications, to ensure that the product is ready for the use.

Testing resources and staffing Resources that will be used include

i. Laptop
ii. VisualStudio
iii. PostgreSQL Test plan 11
iv. TestCollab
v. I will be the sole staff for the software testing.

Test metrics

Software Testing Metrics are the quantitative measures used to estimate the progress, quality, productivity, and health of the software testing process. The goal of software testing metrics is to improve the efficiency and effectiveness in the software testing process and to help make better

decisions for further testing process by providing reliable data about the testing process. A Metric defines in quantitative terms the degree to which a system, system component, or process possesses a given attribute. The types of software metrics that will be used on this project include the following

• **Process Metrics** It is used to improve the efficiency of the process in the SDLC (Software Development Life Cycle).

• **Product Metrics** It is used to tackle the quality of the software product. Test plan 12 • Project Metrics It measures the efficiency of the team working on the project along with the The life cycle for test metrics is listed below

• **Analysis** It is responsible for the identification of metrics as well as the definition.

• **Communicate** It helps in explaining the need and significance of metrics to stakeholders and testing team. It educates the testing team about the data points that need to be captured for processing the metric.

• **Evaluation** It helps in capturing the needed data. It also verifies the validity of the captured data and calculates the metric value

. • Reports It develops the report with an effective conclusion. It distributes the reports to the stakeholders, developers, and the testing teams.

Testing tools and environment

We are in an era of automation being used everywhere the increased demand for automation is trending in our software testing industry, as well. And in many software testing communities like uTest, Quora, etc. software testers are urging for various tools that can be helpful in their day□today testing activities Selenium is a good testing framework to perform web application testing across various browsers and platforms like Windows, Mac, and Linux. Selenium helps the testers to write tests in various programming languages like Java, PHP, C#, Python, Groovy, Ruby, and Perl. It offers record and playback features to write tests without learning Selenium IDE. And this makes easily adaptable for usage in my system testing. A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured.

IMPLEMENTATION STRATEGY

Data Collection and Preprocessing: Collect and preprocess the medical data that will be used to train and test the Naive Bayes model.

▪ Model Development: Develop and train the Naive Bayes model using the preprocessed data.

▪ Model Evaluation: Evaluate the performance of the trained Naive Bayes model using metrics such as accuracy, precision, recall, and the confusion matrix. Refine the model as needed to improve its performance.

▪ System Integration: Integrate the trained Naive Bayes model into the overall system architecture, including the development of a user interface.

▪ Testing and Deployment: Test the end-to-end system to ensure it meets the requirements and is working as expected. Deploy the system into a production environment.

▪ Ongoing Maintenance: Monitor the system's performance to ensure it is operating as expected. Update the model as needed to improve its accuracy and fix any bugs that arise.

▪ User Training: Provide training for users on how to use the system effectively.

▪ Launch: Launch the disease prediction system to the public

REFERENCE

Beverly G. Hope, Rosewary H. Wild, « AnExpert Support System for Service Quality Improvement», Proceedings of the TwentySeventh Annual Hawaii International Conference on System Science, 1994

Analysis and design of information systems by V.Rajaraman, 5th print, PHI, pp 113-137

Joseph Giarratano, Gary Riley (2004). Expert Systems: Principles and Programming, Fourth Edition,