Software Requirements Specification

for

CHRONIC KIDNEY DISEASE MODEL

Version 1.0 approved

Prepared by RAJIV GUPTA

Reg No : 18BCE2203

VIT Vellore

27TH AUGUST 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
INITIAL CKD MODEL	27 AUG 2020	INITIAL DOCUMENTATION	1.0

1. Introduction

1.1 Purpose

This project will help doctors and other users to predict whether the patient has a kidney failure. The CKD model is made using python and machine learning to detect the kidney failure.

1.2 Document Conventions

DS = Dataset
UI = User Interface
CKD = Chronic Kidney Disease
ANN = Artificial Neural Network

1.3 Intended Audience and Reading Suggestions

This document is meant to be read by anyone who is working on this project as well as for the people who would like to refer this project for future work.

1.4 Product Scope

Chronic kidney disease (CKD) is common, and associated with increased risk of cardiovascular disease and end-stage renal disease, which are potentially preventable through early identification and treatment of individuals at risk, this product will develop models that can help in the prediction of the disease at an early stage so that sufficient health facilities could be provided to the patients.

Features:

- Commenting and interacting between other users.
- Uploading files, documents, datasets and relevant information.
- Validation of expert users and normal users.

Milestones:

- Designing CKD models and relationships.
- Designing the UI/UX and its webpage layout

Deliverables:

- Basic API for interaction.
- Fully fledged model with many features.

Technical Requirements:

- Responsive UI
- Should be a scalable model that is able to handle a large dataset.
- Should facilitate to interaction with users.

1.5 References

1. https://thesai.org/Downloads/Volume10No8/Paper_1Detection_of_Chronic_Kidney_Disease.pdf

2. https://medium.com/@randerson112358/chronic-kidney-disease-prediction-detection-using-machine-learning-29cc7e3eba96

3. https://wonderfulengineering.com/this-artificial-kidney-eliminates-the-need-for-kidney-ialysis/?fbclid=IwAR19Olftqa9vr0HEhwl51c89OPN6RGsxPNDDJB2rzhqdKkIVGYYXeGCF1Ds

2. Overall Description

2.1 Product Perspective

The proposed project operates on three stages: pre-processing, implementation and conclusion. The pre-processing stage is the primary process since the database may contain redundant and noise data. By examining the data, different processes take place such as data cleaning, filling missing values, removing excessive data because the missing values and excessive data degrade the performance. For the implementation stage we will work with three most common neural network algorithms (Probabilistic Neural Networks, Multilayer Perceptron algorithm, Support vector machine algorithm) and visualization tools to map the results, in conclusion stage by analysing the graphs various results could be drawn.

2.2 Product Functions

- Predict kidney failure.
- Show a comparison study between all the patients.
- Shows how accurate the model is.
- Graphical visualization of the model and the results.

2.3 User Classes and Characteristics

- Scientists
- Engineers
- Doctors
- Entrepreneurs
- Innovative/Curious folk
- Students

2.4 Operating Environment

This product will be launched for Window, Linux and Mac operating systems. The basic hardware required to support these operating systems are (4GB ram,4core processor), (1GB ram, 2core processor) and 2GB ram,4 core processor) respectively.

2.5 Design and Implementation Constraints

- The algorithms used should be able to train the model properly to increase the accuracy of model.
- The dataset used should be extensive enough to train the model properly.
- Data Cleaning should be done properly to get proper results.

2.6 User Documentation

Please refer section 3.1 for user interfaces.

2.7 Assumptions and Dependencies

It is assumed that the hardware designed will work correctly with the third-party operating system Windows 10 and other Operating Systems and the developed software.

3. External Interface Requirements

3.1 User Interfaces

If the User is not logged in through proper channel, then the screen will appear for which the user can be authenticated and then directed to the main page.
The main page will contain the users feed or relevant posts or projects on which the user can comment and interact. The main page will also contain links to the logout page, documentation page FAQ page.
The user will also be able to request help from one of the experts in the site.
The interface works using machine learning libraries and runs on Jupyter Notebook.

3.2 Hardware Interfaces

The website can be accessed through any of computers with a proper internet connection. Preferred configuration is i7 intel processor, 8gb Ram, 2 Gb Graphics card.

3.3 Software Interfaces

The website can be accessed through any of the common browsers such as mozilla Firefox, Google Chrome, Microsoft edge, Safari etc. Additional softwares Anaconda, Python, Jupyter Notebook also had to be installed for the model to work.

3.4 Communications Interfaces

This software will make use of HTTP connections for website access and will use SMTP for email related functionality. It will also use Jupyter Notebook and various machine learning libraries to work properly.

4. System Features

4.1 Upload Project Content

4.1.1 Description and Priority

This allows the users to predict whether the patient is suffering from kidney failure or not using various attributes and medical terms.

Priority:
This is the main priority of the whole project and crucial to implement.

4.1.2 Stimulus/Response Sequences

1) User inputs the dataset
2) User enters his requirements.
3) User enters related documents and clicks run.

4.1.3 Functional Requirements

1) The system should provide the option of an upload to upload the dataset.
2) User needs to be authenticated before uploading.
3) The system should support all file type uploads relevant to the project.
4) There should be a UI for writing descriptions in the website itself.

4.2 Expert help

4.1.1 Description and Priority

Experts are people who are well versed in the subjects and can help the other users on the platform.

Priority:
This is a medium priority feature.

4.1.2 Stimulus/Response Sequences

1) User presses the help button
2) User browses all experts presently available to help.
3) User can then interact with the expert and ask for help.

4.1.3 Functional Requirements

1) The system should provide the option of a help button.
2) Experts have to be certified and verified before becoming an authenticated expert on the website.
3) The system should support all communicating factors such as file sharing or image sharing.
4) There should be a UI for interacting with experts in the website itself.

5. Other Non-functional Requirements

5.1 Performance Requirements

The website should be fully functional even when scaled to 1000's of users. The latency periods for retrieving data should be under 1ms per 1000 queries. 48 threads should be running on the Jupyter Notebook server.
The model should be properly fitted and accurate to avoid large size that increases latency.

5.2 Safety Requirements

There are no safety compromises in this project.

5.3 Security Requirements

Sufficient measures should be taken to avoid the following security attacks:

- Cross site request Forgery
- XML scripting
- SQL injections
- Dataset checking

5.4 Software Quality Attributes

This project aims to be easy to use for end users and also aims to attract experts on to the site. For an overall better experience.

The main focus is to increase the accuracy of the model to get better results.

6. Other Requirements

Cloud storage for faster access to data and easy to maintain server

Appendix A: Glossary

Appendix B: Analysis Models

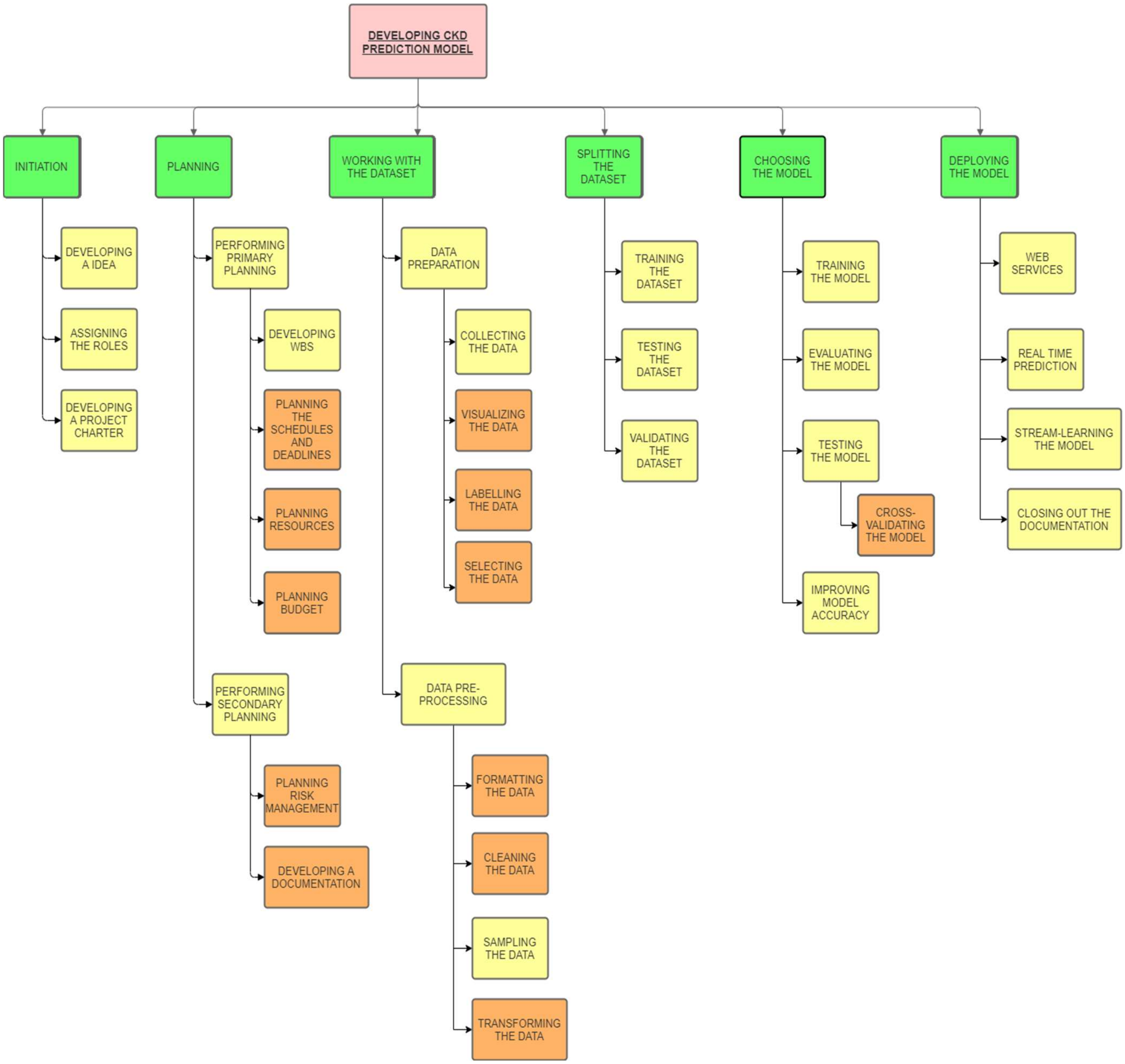
Refer the Graphical representations below.

Appendix C: To Be Determined List

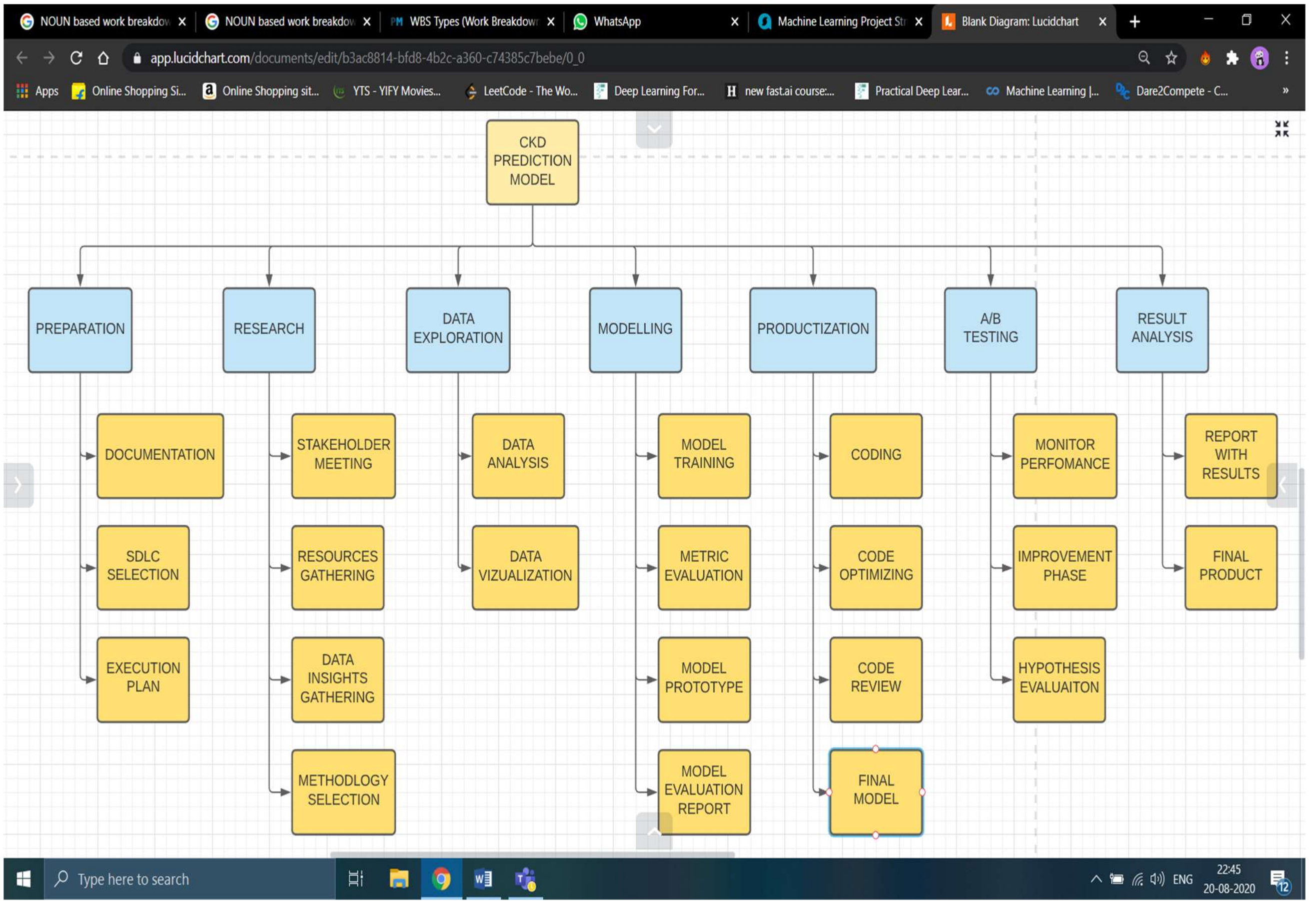
- internet assistant.
- Recommendation algorithms for users.

Graphical Representations

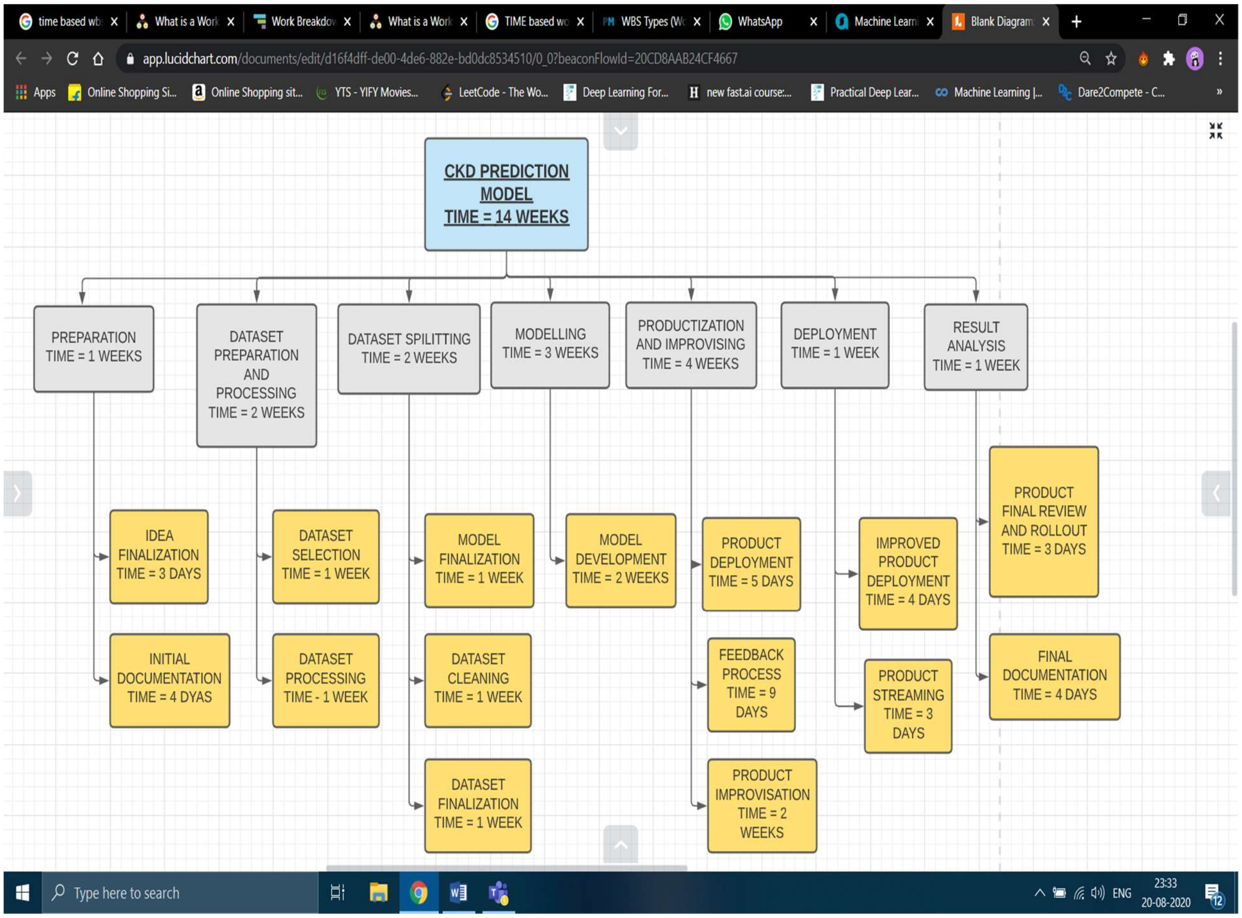
- **VERB BASED WORK BREAKDOWN STRUCTURE**



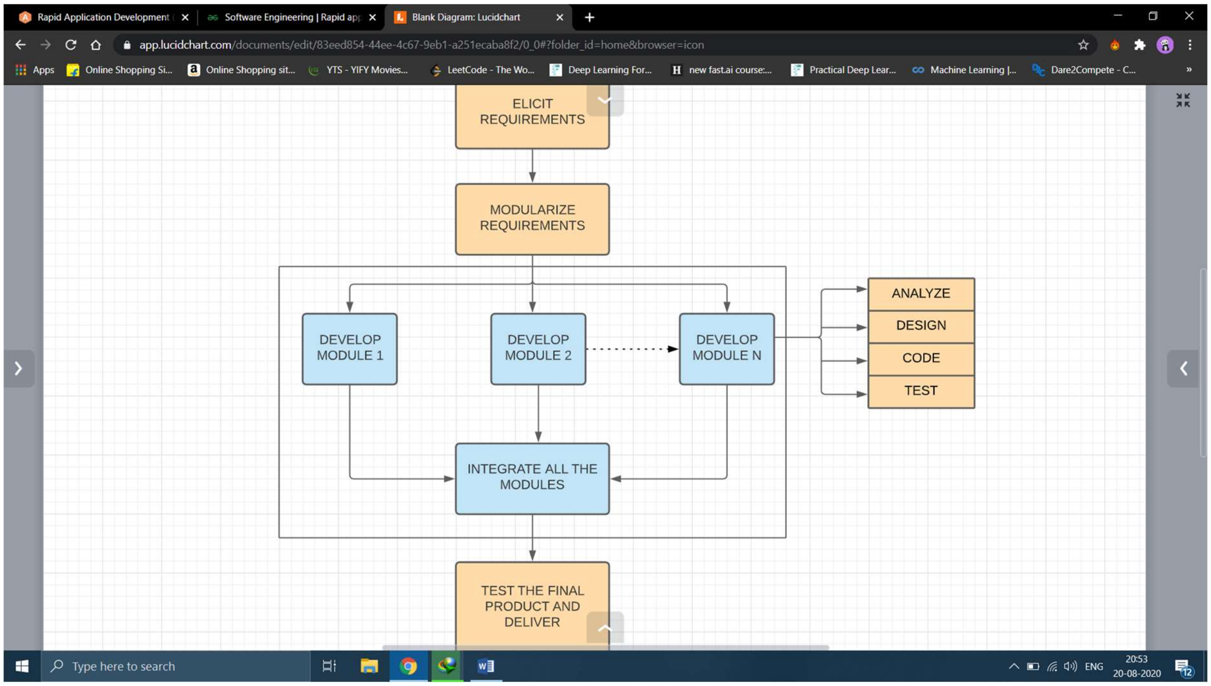
- **NOUN BASED WORK BREAKDOWN STRUCTURE**



- **TIME BASED WORK BREAKDOWN STRUCTURE**


time based wb
What is a Work
Work Breakdov
What is a Work
TIME based wo
WBS Types (W
WhatsApp
Machine Learni
Blank Diagram
app.lucidchart.com/documents/edit/d16f4dff-de00-4de6-882e-bd0dc8534510/0_0?beaconFlowId=20CD8AAB24CF4667
Apps
Online Shopping Si...
Online Shopping sit...
YTS - YIFY Movies...
LeetCode - The Wo...
Deep Learning For...
new fast.ai course:...
Practical Deep Lear...
Machine Learning |...
Dare2Compete - C...
CKD PREDICTION MODEL TIME = 14 WEEKS
PREPARATION TIME = 1 WEEKS
DATASET PREPARATION AND PROCESSING TIME = 2 WEEKS
DATASET SPILITTING TIME = 2 WEEKS
MODELLING TIME = 3 WEEKS
PRODUCTIZATION AND IMPROVISING TIME = 4 WEEKS
DEPLOYMENT TIME = 1 WEEK
RESULT ANALYSIS TIME = 1 WEEK
IDEA FINALIZATION TIME = 3 DAYS
INITIAL DOCUMENTATION TIME = 4 DYAS
DATASET SELECTION TIME = 1 WEEK
DATASET PROCESSING TIME - 1 WEEK
MODEL FINALIZATION TIME = 1 WEEK
DATASET CLEANING TIME = 1 WEEK
DATASET FINALIZATION TIME = 1 WEEK
MODEL DEVELOPMENT TIME = 2 WEEKS
PRODUCT DEPLOYMENT TIME = 5 DAYS
FEEDBACK PROCESS TIME = 9 DAYS
PRODUCT IMPROVISATION TIME = 2 WEEKS
IMPROVED PRODUCT DEPLOYMENT TIME = 4 DAYS
PRODUCT STREAMING TIME = 3 DAYS
PRODUCT FINAL REVIEW AND ROLLOUT TIME = 3 DAYS
FINAL DOCUMENTATION TIME = 4 DAYS
Type here to search
ENG
23:33
20-08-2020

- **SOFTWARE DEVELOPMENT LIFE CYCLE**


Rapid Application Development
Software Engineering | Rapid app
Blank Diagram: Lucidchart
app.lucidchart.com/documents/edit/83eed854-44ee-4c67-9eb1-a251ecaba8f2/0_0#?folder_id=home&browser=icon
Apps
Online Shopping Si...
Online Shopping sit...
YTS - YIFY Movies...
LeetCode - The Wo...
Deep Learning For...
new fast.ai course:...
Practical Deep Lear...
Machine Learning |...
Dare2Compete - C...
ELICIT REQUIREMENTS
MODULARIZE REQUIREMENTS
DEVELOP MODULE 1
DEVELOP MODULE 2
DEVELOP MODULE N
ANALYZE
DESIGN
CODE
TEST
INTEGRATE ALL THE MODULES
TEST THE FINAL PRODUCT AND DELIVER
Type here to search
ENG
20:53
20-08-2020