**Real Time Open Board Application**

**A PROJECT REPORT**

**Submitted By**

**AKASH SINGH**

**University Roll No- 2100290140013**

**ABHISHEK GAUTAM**

**University Roll No- 2100290140005**

**AKSHIT**

**University Roll No- 2100290140019**

**ISHIKA**

**University Roll No- 2100290140072**

**Submitted in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATIONS**

**Under the Supervision of**

**MR. ANKIT VERMA**

### ASSISTANT PROFESSOR

**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(FEB-2023)**

**CERTIFICATE**

Certified that **Akash Singh (University Roll No.- 2100290140013)** have carried out the project work having “**Realtime Open Board Applications**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)**, Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Akash Singh**

**University Roll No. 2100290140013**

**Abhishek Gautam**

**University Roll No- 2100290140005**

**Akshit**

**University Roll No- 2100290140019**

**Ishika**

**University Roll No- 2100290140072**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 14/11/2022

**Mr. Ankit Verma**

**Assistant Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Arun Kumar Tripathi**

**Head, Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**ABSTRACT**

Open Board is an open source cross-platform teaching software for interactive whiteboard designed primarily for use in schools and universities. It can be used both with interactive whiteboards or in a dual-screen setup with a pen-tablet display and a beamer.

It provides a lightweight, interactive interface so that it can be used easily on all types devices.

User can choose more than one item to make anything. The user will send and upload the documentation to system or user.

Open board system, as described above, can lead to error free, secure, reliable and fast application. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help to user in better utilization of resources. The user can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

**ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my project supervisor, **Mr. Ankit Verma** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Akash Singh (2100290140013)**

**Abhishek Gautam (2100290140005)**

**Akshit (2100290140018)**

**Ishika (2100290140072)**

**TABLE OF CONTENTS**

|  |
| --- |
| **Page No.** |
| * Certificate 2 |
| * Abstract 3 |
| * Acknowledgement 4 |
| * Table of Contents 5 * Project Description 6 * Functional Requirements 7 * Non- functional Requirements 8 * Feasibility Study 9 * SDLC Model 10 * ER Diagram 12 * Use Case Diagram 15 * Activity Diagram 16 * Sequence Diagram 17 * Collaboration Diagram 18 * State Chart Diagram 19 * Component Diagram 20 * Deployment Diagram 21 * Snapshots 22 * Report 24 * Source Code 25 * Conclusion 55 * References and Bibliography 56 |

**PROJECT DESCRIPTION**

Open Board is an open source cross-platform teaching software for interactive whiteboard designed primarily for use in schools and universities. It can be used both with interactive whiteboards or in a dual-screen setup with a pen-tablet display and a beamer.

Interest is growing fast across Switzerland but also in Europe and North America. A Swiss law non-profit organisation should be created. This organisation would promote Open Board and its open source development and foster collaboration between interested parties. This in respect of the core principles which led to the creation of this fork, that is the refocusing on the work of a teacher in a classroom, privileging the ease of use.

The project comprises of basic open board functionalities such as, writing, erasing, downloading the canvas, adding notes (with minimizing it, closing it, dragging it around on the page), uploading the image (features same as of notes), undo and redo actions.

Added Realtime drawing functionality using Socket.io by connecting to server using Express.js. Using green button, we can minimize the notes and by clicking red button, notes can be closed. Similarly for uploading images. Realtime drawing can be achieved by the people using same link at same time.

Open Board comes with a built-in library of sounds, videos, images, shapes and animations, allowing teachers to create interactive content and increase student engagement. Schools and universities can upload and store course-specific documents within the platform, ensuring quick access to content for teachers and students. It provides a web browser, which lets tutors make annotations and share webpages from a centralized platform. Additionally, users can access various tools such as protractors, calculator, clock, stopwatch, world maps and sticky notes to improve the learning experience.

Open Board facilitates integration with third-party applications, including Google Maps and Graphs. It is available for free and support is extended via documentation, discussion forum and email.

**FUNCTIONAL REQUIREMENTS**

**Teaching and Learning System**

* 1. Navigate the Board menu
  2. Select an item from the Function from Navigation Menu
  3. Customized options for selected Function
  4. Remove Function/ remove all Function from current Board

**NON-FUNCTIONAL REQUIREMENTS**

**Portability**

System running on one platform can easily be converted to run on another platform.

**Reliability**

The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended.

**Availability**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs.

**Performance**

How fast does it need to operate.

**Supportability**

Is support provided in-house or is remote accessibility for external resources required.

**FEASIBILITY STUDY**

After studying and analyzing all the existing and requires functionalities of the system, the next task is to do the feasibility study for the project. Feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

**Economical Feasibility**

For the economic feasibility, Economic analysis or cost/benefits analysis is most frequently used technique the effectiveness of a proposed system. it is a procedure to determine the benefits and saving those are expected from the proposes system and compare them with cost. If the benefits outweigh the costs, a decision is taken to design and implement the system. otherwise, further justification or alternative in proposed system will have to be made if it is to have a chance of being approved this is ongoing effort that improves in accuracy at each phase of a system life cycle

**Technical Feasibility**

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionalities to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of front end and backend platform.

**Operational Feasibility**

No doubt the technically growing world needs more enhancement in technology, this application is very user friendly and all inputs to be taken all self-explanatory even to a layman. As far our study is concerned, the clients will be comfortable and happy as the system has cut down their loads and bring the young generation to the same virtual world they are growing drastically.

**SOFTWARE DEVELOPMENT LIFE CYCLE**

**WATERFALL MODEL**

The waterfall model is a well-known structured methodology for software

development. The whole process of system development is divided into distinct

phases. The model has been introduced in 1970s. Every phase has a unique output.

It was the first SDLC model to be used widely. So that, sometime it is referred to

Waterfall by SDLC. The waterfall model is used when the system requirements are

well known, technology is understood and the system is a new version of an existing

product (Dennis, Wixom and Roth, 2012).

Mainly there are six phases in Waterfall model. If there is a problem faced in any

phase of the cycle, the system goes to the previous phase. The phases of Waterfall

method is:



**Requirements Gathering & Analysis:**

In this Phase, all possible requirements of the system are captured and documented in a requirement specification doc.

**System Design:**

The requirements documented in previous phase are studied in this phase and the system design is prepared.

**Implementation:**

With inputs from system design, the system is developed in several unites. Then the units are tested.

**Integration & Testing:**

The units of the program developed in previous phase are integrated into a system. Then the whole system is tested.

**Deployment of the system:**

When the all kind of testing is done, the product is deployed in the customer environment.

**Maintenance:**

There are some issues which are found in the client environment. Patches are released to fix those issues.

**ER DIAGRAM**

Entity is represented by the rectangle shape. The entity will be our database table of Online Food Ordering System later on.

Attribute is represented by the oval shape. This will be the columns or fields of each table in the Online Food Ordering System.

Relationship is represented by diamond shape. This will determine the relationships among entities. This is usually in a form of primary key to foreign key connection.

We will follow the 3 basic rules in creating the ER Diagram.

* Identify all the entities.
* Identify the relationship between entities and
* Add meaningful attributes to our entities.

**Diagram

Description automatically generated**

**USE CASE DIAGRAM**

Use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

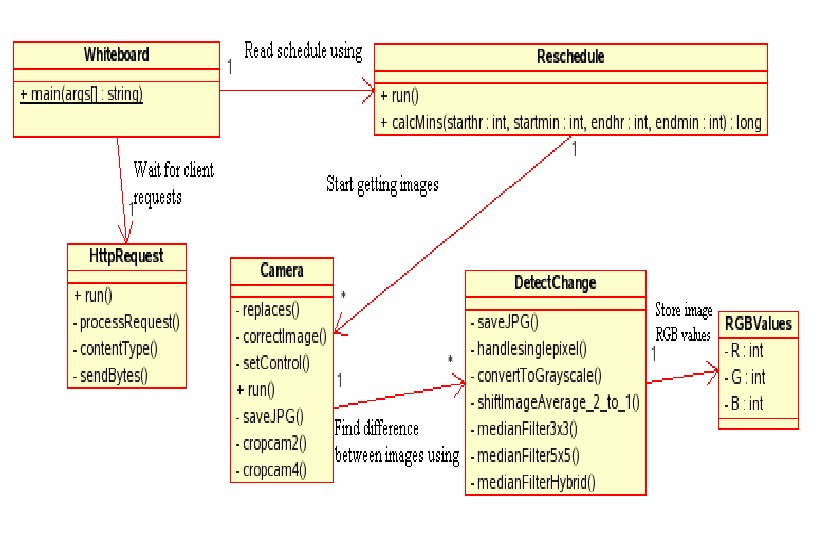
Purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

**CLASS DIAGRAM**

An Class Diagram is a **behavioural diagram**. It depicts the **behaviour of a system**. Its primary use is to **depict the dynamic aspects of a system**. The **dynamic aspect of a system** specifies how the **system operates to attain its function**.

It is basically a flowchart to represent the flow from one activity to another activity. Activity Diagrams are not exactly flowcharts as they have some additional capabilities including branching, parallel flow, etc.

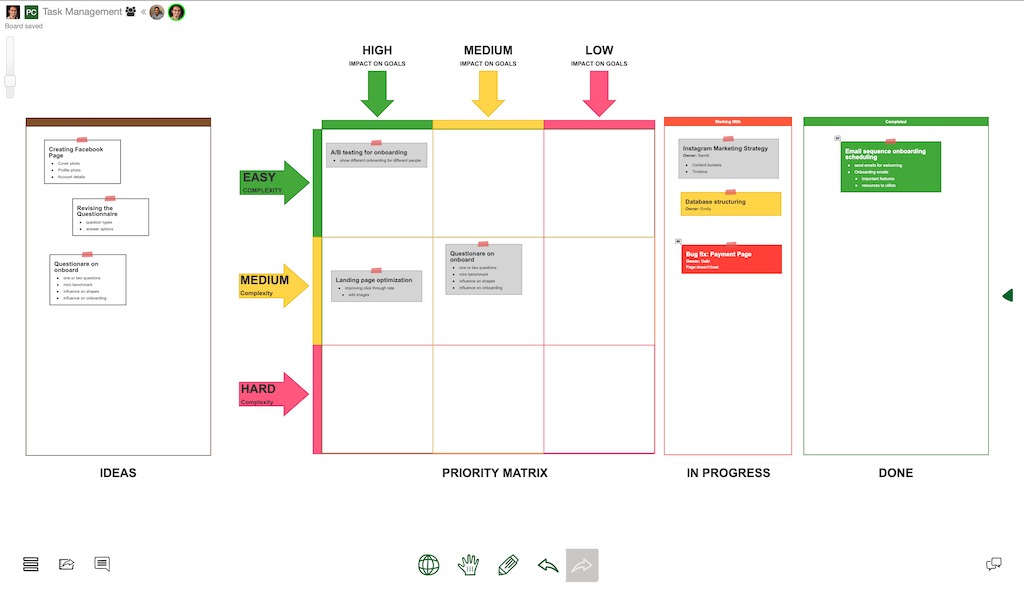


**SEQUENCE DIAGRAM**

The design shows the detailed illustration of events sequenced and happens in Realtime Openwhite . This designed sequence diagram is able to show programmers and readers about the sequence of messages between the actor and the objects.

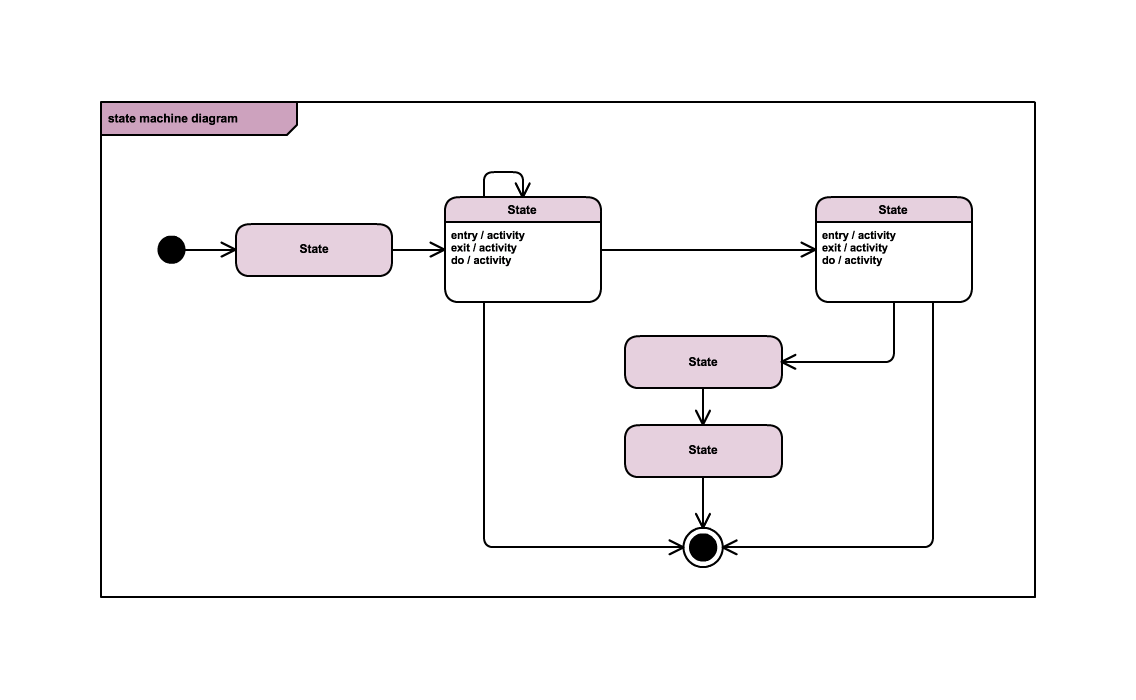
As you can see through the illustration, the conditions and interactions are emphasized. These interactions are essential for the Realtime Openwhite System development.  
  
The series of messages are shown and labeled to guide you in building the System. You can modify the design if you have more ideas. You can also add more features to this design and use it as your project blueprint.

**COLLABORATION DIAGRAM**



The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

**STATE CHART DIAGRAM**



The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

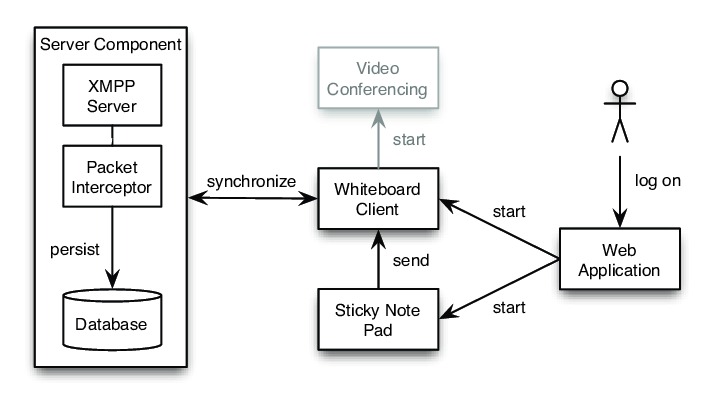
Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most

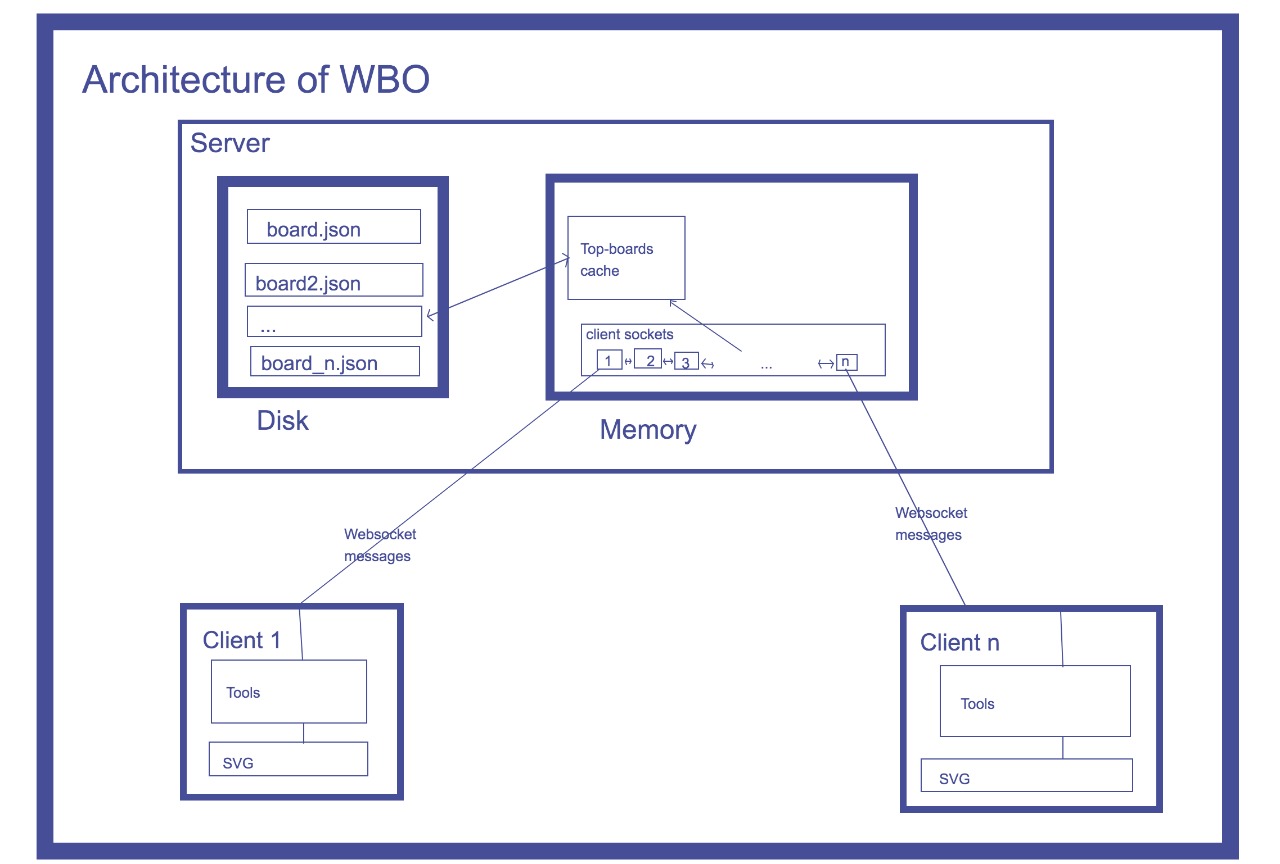
**COMPONENT DIAGRAM**

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.



**DEPLOYMENT DIAGRAM**



Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.

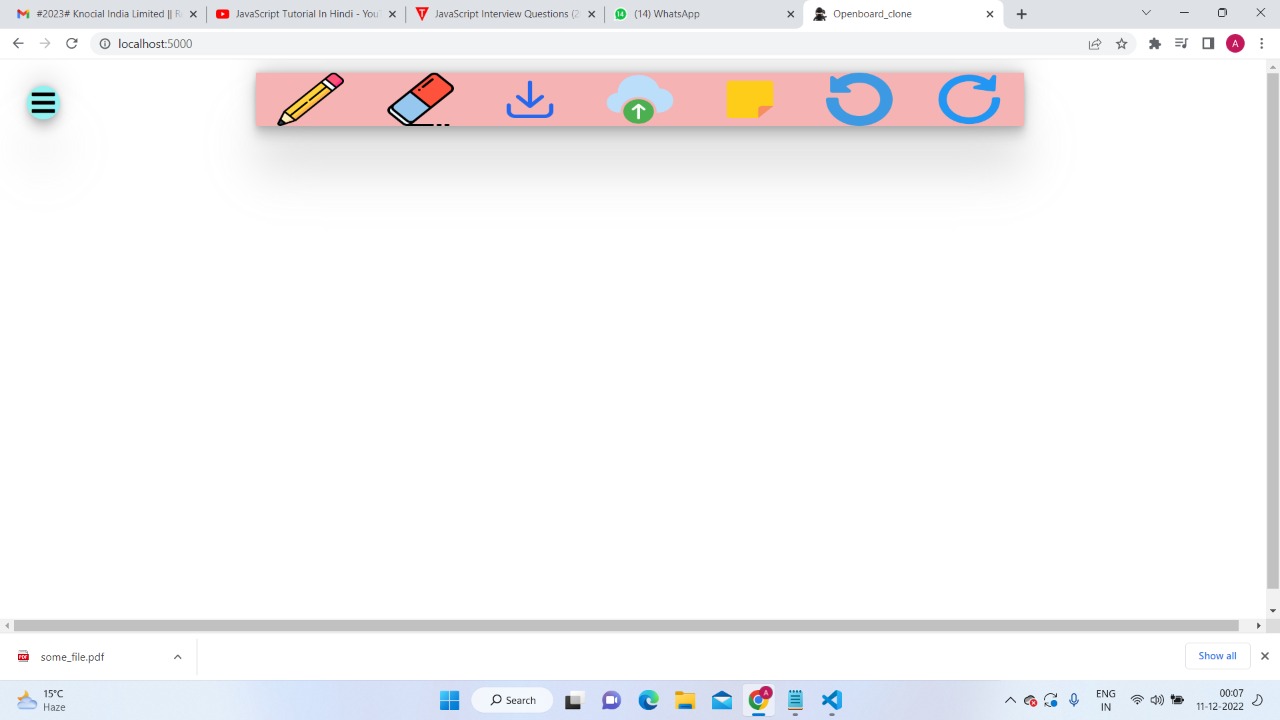
**Interface Design User**

User Interface Design is concerned with a dialogue between a user and the computer.

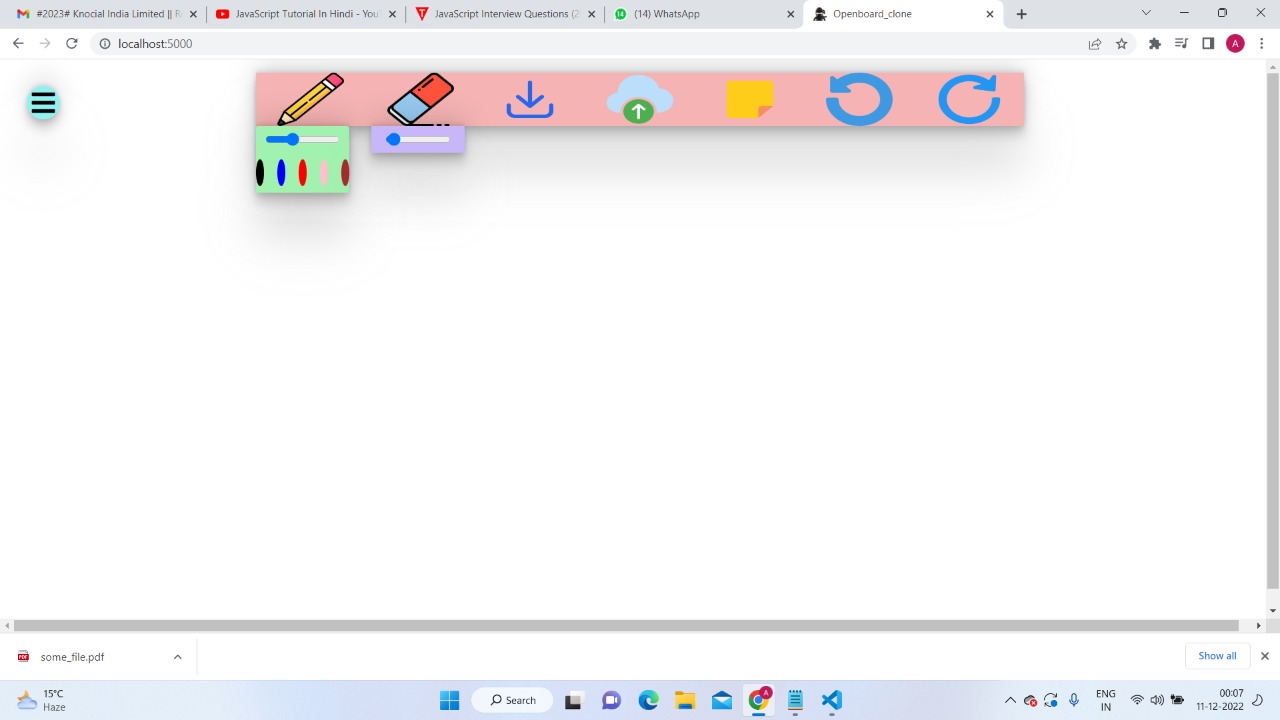
It is concerned with everything from starting the a system or logging into system into eventual presenation of desired inputs and outputs .

The following steps are various guidelines for User Interface Design:

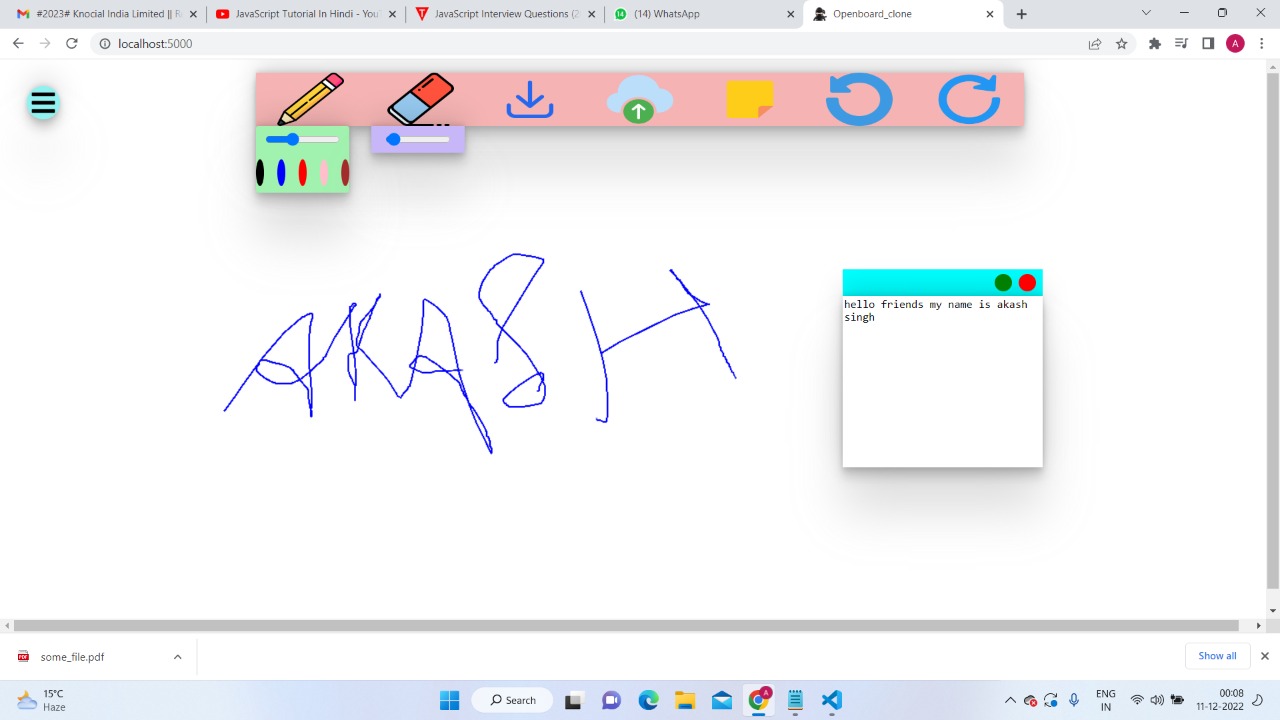
**Welcome page**



**Select an item from Navigation bar**

****

**Working on any task**

****

**REPORT**

* It generates the report on open board application.
* Provide filter reports on pencil colour, eraser and so on.
* You can easily export file ,folder on the white board.
* It generates the report on the particular work at a time.
* It generates the report on the number of notes.

**SOURCE CODE**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"

integrity="sha512-PgQMlq+nqFLV4ylk1gwUOgm6CtIIXkKwaIHp/PAIWHzig/lKZSEGKEysh0TCVbHJXCLN7WetD8TFecIky75ZfQ=="

crossorigin="anonymous" referrerpolicy="no-referrer" />

<link rel="icon" type="image/x-icon" href="./images/favicon.png">

<link rel="stylesheet" href="style.css">

<title>Openboard\_clone</title>

</head>

<body>

<div class="options-cont">

<i class="fas fa-bars"></i>

</div>

<div class="tools-cont scale-tools">

<img class="pencil" src="images/pencil.png">

<img class="eraser" src="images/eraser.png">

<img class="download" src="images/download.png">

<img class="upload" src="images/upload.png">

<img class="sticky-note" src="images/sticky-note.png">

<img class="undo" src="images/undo.png">

<img class="redo" src="images/redo.png">

</div>

<div class="pencil-tool-cont">

<div class="pencil-width-cont">

<input type="range" min="2" max="20" value="8"class="pencil-width">

</div>

<div class="pencil-color-cont">

<div class="black pencil-color"></div>

<div class="blue pencil-color"></div>

<div class="red pencil-color"></div>

</div>

</div>

<div class="eraser-tool-cont">

<input type="range" min="2" max="50" value="4" class="eraser-width">

</div>

<!-- <div class="sticky-cont">

<div class="header-cont">

<div class="minimize"></div>

<div class="remove"></div>

</div>

<div class="note-cont">

<textarea cols="40" rows="13"></textarea>

</div>

</div> -->

<canvas></canvas>

<script src="https://cdn.socket.io/4.5.0/socket.io.min.js"

integrity="sha384-7EyYLQZgWBi67fBtVxw60/OWl1kjsfrPFcaU0pp0nAh+i8FD068QogUvg85Ewy1k" crossorigin="anonymous"></script>

<script>

let socket = io.connect("http://localhost:5000");

</script>

<script src="./tools.js"></script>

<script src="./canvas.js"></script>

</body>

</html>

**Style.css**

\* {

box-sizing: border-box;

}

body {

margin:0;

padding:0;

}

.options-cont {

height: 2.5rem;

width: 2.5rem;

background-color: rgb(145, 241, 241);

position: absolute;

top: 2rem;

left: 2rem;

display: flex;

justify-content: center;

align-items: center;

border-radius: 80%;

font-size: 2rem;

box-shadow:rgba(160, 159, 159, 0.25) 0px 54px 55px, rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px, rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

}

.tools-cont {

display: flex;

height: 4rem;

width: 60vw;

background-color: rgb(245, 179, 179);

justify-content: space-around;

position: absolute;

top: 1rem;

left: 20vw;

box-shadow:rgba(128, 127, 127, 0.25) 0px 54px 55px, rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px, rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

}

.scale-tools {

animation: scale-tools;

animation-duration: 1s;

}

@keyframes scale-tools {

0%{

transform: scale(0.1);

}

100%{

transform: scale(1);

}

}

.tools-cont > \* {

display: block;

width: 5rem;

}

.pencil-tool-cont {

background-color: rgb(161, 241, 175);

box-shadow:rgba(128, 127, 127, 0.25) 0px 54px 55px, rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px, rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

height: 5rem;

width: 7rem;

position:absolute;

top: 5rem;

left:20vw;

border-radius: 3px;

display: none;

}

.pencil-width-cont {

width: 7rem;

height: 2rem;

border-radius: 5px;

display:flex;

justify-content: center;

align-items: center;

}

.pencil-width-cont input {

width: 80%;

border-radius: 5px;

}

.pencil-color-cont

{

height: calc(100% - 2rem);

display: flex;

justify-content: center;

align-items: center;

gap: 1rem;

}

.pencil-color {

width: 1.5rem;

height: 2rem;

border-radius: 50%;

gap:1rem;

}

.black {

background-color: black;

}

.blue {

background-color: blue;

}

.red {

background-color: red;

}

.eraser-tool-cont {

height: 2rem;

width: 7rem;

background-color: rgb(200, 183, 248);

box-shadow:rgba(128, 127, 127, 0.25) 0px 54px 55px, rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px, rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

display: none;

justify-content: center;

align-items: center;

position: absolute;

top:5rem;

left:29vw;

}

.eraser-tool-cont input {

width: 70%;

border-radius: 5px;

}

.sticky-cont {

height: 15rem;

width: 15rem;

position:absolute;

top:15rem;

left:10rem;

}

.header-cont {

height: 2rem;

display:flex;

justify-content: flex-end;

align-items: center;

background-color: aqua;

}

.header-cont > \* {

height:1.3rem;

width:1.3rem;

margin-right:0.5rem;

border-radius: 50%;

}

.minimize {

background-color: green;

}

.remove {

background-color: red;

}

.note-cont

{

height:calc(100%-2rem);

box-shadow:rgba(128, 127, 127, 0.25) 0px 54px 55px, rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px, rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

}

textarea {

height:100%;

width:100%;

outline:none;

border:none;

resize:none;

}

img {

height:100%;

width:100%;

}

**Tools.js**

let toolsCont = document.querySelector(".tools-cont");

let optionsCont = document.querySelector(".options-cont");

let optionsFlag = true;

let pencilToolCont = document.querySelector(".pencil-tool-cont");

let eraserToolCont = document.querySelector(".eraser-tool-cont");

let pencil = document.querySelector(".pencil");

let eraser = document.querySelector(".eraser");

let sticky = document.querySelector(".sticky-note");

let upload = document.querySelector(".upload");

let pencilFlag = false;

let eraserFlag = false;

optionsCont.addEventListener("click", function (e) {

optionsFlag = !optionsFlag;

if (optionsFlag) {

openTools();

}

else {

closeTools();

}

})

function openTools() {

let iconElem = optionsCont.children[0];

iconElem.classList.remove("fa-times");

iconElem.classList.add("fa-bars");

toolsCont.style.display = "flex";

}

function closeTools() {

let iconElem = optionsCont.children[0];

iconElem.classList.remove("fa-bars");

iconElem.classList.add("fa-times");

toolsCont.style.display = "none";

pencilToolCont.style.display = "none";

eraserToolCont.style.display = "none";

}

pencil.addEventListener("click", (e) => {

pencilFlag = !pencilFlag;

if (pencilFlag) pencilToolCont.style.display = "block";

else pencilToolCont.style.display = "none";

})

eraser.addEventListener("click", (e) => {

eraserFlag = !eraserFlag;

if (eraserFlag) eraserToolCont.style.display = "flex";

else eraserToolCont.style.display = "none";

})

upload.addEventListener("click", (e) => {

let input = document.createElement("input");

input.setAttribute("type", "file");

input.click();

input.addEventListener("change", (e) => {

let file = input.files[0];

let url = URL.createObjectURL(file);

let stickyTemplateHTML = `<div class="header-cont">

<div class="minimize"></div>

<div class="remove"></div>

</div>

<div class="note-cont">

<img src ="${url}"/>

</div>`;

createSticky(stickyTemplateHTML);

})

let stickyCont = document.createElement("div");

stickyCont.setAttribute("class", "sticky-cont");

stickyCont.innerHTML = `

<div class="header-cont">

<div class="minimize"></div>

<div class="remove"></div>

</div>

<div class="note-cont">

<textarea cols="40" rows="13" ></textarea>

</div>

`;

document.body.appendChild(stickyCont);

let minimize = stickyCont.querySelector(".minimize");

let remove = stickyCont.querySelector(".remove");

noteActions(minimize, remove, stickyCont);

stickyCont.onmousedown = function (event) {

dragAndDrop(stickyCont, event);

};

stickyCont.ondragstart = function () {

return false;

};

})

sticky.addEventListener("click", (e) => {

let stickyTemplateHTML = `

<div class="header-cont">

<div class="minimize"></div>

<div class="remove"></div>

</div>

<div class="note-cont">

<textarea cols="40" rows="13" spellcheck="false" ></textarea>

</div>

`;

createSticky(stickyTemplateHTML);

})

function createSticky(stickyTemplateHTML) {

let stickyCont = document.createElement("div");

stickyCont.setAttribute("class", "sticky-cont");

stickyCont.innerHTML = stickyTemplateHTML;

document.body.appendChild(stickyCont);

let minimize = stickyCont.querySelector(".minimize");

let remove = stickyCont.querySelector(".remove");

noteActions(minimize, remove, stickyCont);

stickyCont.onmousedown = function (event) {

dragAndDrop(stickyCont, event);

};

stickyCont.ondragstart = function () {

return false;

};

}

function noteActions(minimize, remove, stickyCont) {

remove.addEventListener("click", (e) => {

stickyCont.remove();

}

)

minimize.addEventListener("click", (e) => {

let noteCont = stickyCont.querySelector(".note-cont");

let display = getComputedStyle(noteCont).getPropertyValue("display");

if (display == "none") noteCont.style.display = "block";

else noteCont.style.display = "none";

})

}

function dragAndDrop(element, event) {

let shiftX = event.clientX - element.getBoundingClientRect().left;

let shiftY = event.clientY - element.getBoundingClientRect().top;

element.style.position = 'absolute';

element.style.zIndex = 1000;

moveAt(event.pageX, event.pageY);

// moves the ball at (pageX, pageY) coordinates

// taking initial shifts into account

function moveAt(pageX, pageY) {

element.style.left = pageX - shiftX + 'px';

element.style.top = pageY - shiftY + 'px';

}

function onMouseMove(event) {

moveAt(event.pageX, event.pageY);

}

// move the ball on mousemove

document.addEventListener('mousemove', onMouseMove);

// drop the ball, remove unneeded handlers

element.onmouseup = function () {

document.removeEventListener('mousemove', onMouseMove);

element.onmouseup = null;

};

}

**App.js**

const express = require("express");

const socket = require("socket.io");

const app = express();

app.use(express.static("public"));

let port = 5000

let server = app.listen(port, ()=> {

console.log("listening at port" + port);

})

let io = socket(server);

io.on("connection", (socket) => {

console.log("Made socket connection");

socket.on("beginPath", (data) => {

io.sockets.emit("beginPath",data);

})

socket.on("drawStroke",(data) => {

io.sockets.emit("drawStroke",data);

})

socket.on("redoUndo",(data) => {

io.sockets.emit("redoUndo", data);

})

})

**Canvas.js**

let canvas = document.querySelector("canvas");

canvas.width=window.innerWidth;

canvas.height=window.innerHeight;

let pencilColorCont = document.querySelectorAll(".pencil-color");

let pencilWidthElem=document.querySelector(".pencil-width");

let eraserWidthElem=document.querySelector(".eraser-width");

let download = document.querySelector(".download");

let redo=document.querySelector(".redo");

let undo=document.querySelector(".undo");

let penColor ="red";

let eraserColor="white";

let penWidth ="pencilWidthElem.value";

let eraserWidth = "eraserWidthElem.value";

let undoRedoTracker = [];

let track = 0;

let mouseDown=false;

//API

let tool=canvas.getContext("2d");

tool.strokeStyle="penColor";

tool.lineWidth="penWidth";

canvas.addEventListener("mousedown",(e) =>

{

mouseDown=true;

let data = {

x: e.clientX,

y: e.clientY

}

socket.emit("beginPath",data);

})

canvas.addEventListener("mousemove",(e) => {

if(mouseDown)

{

let data={

x: e.clientX,

y: e.clientY,

color:eraserFlag? eraserColor:penColor,

width:eraserFlag? eraserWidth :penWidth

}

socket.emit("drawStroke",data);

}

})

canvas.addEventListener("mouseup",(e) => {

mouseDown=false;

let url= canvas.toDataURL();

undoRedoTracker.push(url);

track=undoRedoTracker.length-1;

})

undo.addEventListener("click",(e) => {

if(track > 0) track--;

let data={

trackValue: track,

undoRedoTracker

}

socket.emit("redoUndo",data);

})

redo.addEventListener("click",(e) => {

if(track < undoRedoTracker.length-1) track++;

let data={

trackValue: track,

undoRedoTracker

}

socket.emit("redoUndo",data);

})

function undoRedoCanvas(trackObj) {

track = trackObj.trackValue;

undoRedoTracker = trackObj.undoRedoTracker;

let url=undoRedoTracker[track];

let img = new Image();

img.src=url;

img.onload=(e) => {

tool.drawImage(img, 0, 0, canvas.width, canvas.height);

}

}

function beginPath(strokeObj) {

tool.beginPath();

tool.moveTo(strokeObj.x,strokeObj.y);

}

function drawStroke(strokeObj) {

tool.lineTo(strokeObj.x,strokeObj.y);

tool.stroke();

}

pencilColorCont.forEach((colorElem) => {

colorElem.addEventListener("click",(e) => {

let color= colorElem.classList[0];

penColor=color;

tool.strokeStyle=penColor;

})

})

pencilWidthElem.addEventListener("change", (e)=> {

penWidth=pencilWidthElem.value;

tool.lineWidth=penWidth;

})

eraserWidthElem.addEventListener("change", (e)=> {

eraserWidth=eraserWidthElem.value;

tool.lineWidth=eraserWidth;

})

eraser.addEventListener("click",(e) => {

if(eraserFlag) {

tool.strokeStyle=eraserColor;

tool.lineWidth=eraserWidth;

}

else {

tool.strokeStyle=penColor;

tool.lineWidth=penWidth;

}

})

download.addEventListener("click",(e) => {

let url=canvas.toDataURL();

let a = document.createElement("a");

a.href =url;

a.download="board.jpg";

a.click();

})

socket.on("beginPath", (data) => {

beginPath(data);

})

socket.on("drawStroke",(data) => {

drawStroke(data);

})

socket.on("redoUndo",(data) => {

undoRedoCanvas(data);

})

**CONCLUSION**

Our project is only a humble venture to satisfy the needs to manage their project work.Several user friendly coding have also adopted. Theis package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame word that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

**At the end it is concluded that we have made effort on following points**

* A description of the background and context of the project and its relation to word already done in the area.
* Made statement of the aims and objectives of the project.
* The description of purpose, scope, and applicability.
* We define the problem on which we are working on the project.
* We describe the requirement specifications of the system and the actions that can be done on these things.
* We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
* We included features and operations in detail, including screen layouts.
* We designed user interface and security issues related to system.

**Literature Review Chapter**

**I. Introduction**

* Brief explanation of what a whiteboard application is and its importance
* Purpose of the literature review chapter

II. Whiteboard applications: Overview and types

* Definition of a whiteboard application
* Overview of different types of whiteboard applications (e.g., desktop, web-based, mobile, collaborative)
* Brief history of whiteboard applications

**III. Use cases for whiteboard applications**

* Overview of different fields and industries where whiteboard applications are commonly used (e.g., education, business, healthcare, design)
* Examples of specific use cases (e.g., virtual brainstorming sessions, remote collaboration, online tutoring)

**IV. Features and functionality of whiteboard applications**

* Overview of common features found in whiteboard applications (e.g., drawing tools, text tools, image insertion, shape recognition)
* Discussion of more advanced features (e.g., handwriting recognition, object recognition, voice recognition)

**V. User experience and usability of whiteboard applications**

* Factors that affect the usability of whiteboard applications (e.g., interface design, ease of use, performance)
* Best practices for designing effective whiteboard applications

**VI. Research and development of whiteboard applications**

* Overview of current research and development in the field of whiteboard applications
* Discussion of emerging trends and technologies (e.g., machine learning, virtual reality)

**REFERENCES**

* <https://www.tutorialspoint.com>
* <https://www.javatpoint.com>
* <https://www.w3school.com>
* <https://www.youtube.com>
* <https://www.google.com>
* <https://www.wikepedia.com>
* <https://www.geeksforgeeks.com>