

# ***“Digital White Board”***

**A Project Report Submitted to  
Rajiv Gandhi Proudhyogiki Vishwavidyalaya**



**Towards Partial Fulfillment for the Award of  
Bachelor of Engineering in *Computer Science Engineering***

***Submitted by:***

**Omansh Kalra(0827CS181137)  
Rahul Choudhary(0827CS181161)**

***Guided by:***

**Ms. Priyanka Jangde  
Associate Professor**



***Department of Computer Science Engineering  
Acropolis Institute of Technology & Research, Indore  
Jan - June 2022***

# EXAMINER APPROVAL

The Project entitled "***Digital White Board***" submitted by **Omansh Kalra (0827CS181137)**, **Rahul Choudhary(0827CS181161)** has been examined and is hereby approved towards partial fulfillment for the award of ***Bachelor of Engineering degree in Computer Science Engineering*** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

**(Internal Examiner)**

**Date:**

**(External Examiner)**

**Date:**

# GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled “***Digital White Board***” submitted by **Omansh Kalra (0827CS181137), Rahul Choudhary (0827CS181161)** is a satisfactory account of the bonafide work done under the supervision of ***Ms. Priyanka Jangde, Associate Professor***, is recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science Engineering) degree by Rajiv Gandhi Pradyogiki Vishwavidhyalaya, Bhopal.

**(Project Guide)**

**(Project Coordinator)**

# STUDENTS UNDERTAKING

This is to certify that project entitled ***“Digital White Board”*** has developed by us under the supervision of ***Ms. Priyanka Jangde, Associate Professor***. The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

**Omansh Kalra (0827CS181137)**

**Rahul Choudhary (0827CS181161)**

# Acknowledgement

---

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Ms. Priyanka Jangde, Associate Professor**, Professor, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr. Kamal Kumar Sethi**, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CS Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr. S. C. Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to our parent and family members who have always loved and supported us unconditionally. To all of them, we want to say “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

**Omansh Kalra (0827CS181137), Rahul Choudhary(0827CS181161)**

# Executive Summary

---

## ***Digital White Board***

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Computer Science Engineering branch under the sagacious guidance and vigilant supervision of ***Ms. Priyanka Jangde, Associate Professor.***

This Project is developed keeping in mind the effect of the pandemic situation on our educational system. As Everything is going online so we decide to make this project which can be used by teachers to teach their students online easily. “Digital White Board” is a real time application that allows user to draw/ write important points during a meeting and can also be used for teaching purposes while it helps the teacher to explain the topic in detail while taking the classes online. It has an intuitive interface that allows user to share ideas in real time.

*“Where the vision is one  
year, cultivate flowers;*

*Where the vision is ten years,  
cultivate trees;*

*Where the vision is eternity,  
cultivate people.”*

*- Oriental Saying*

# List of Figures

---

Figure 2-1 : Google Jam Board.....7

Figure 2-2 : Zoom White Board.....7

Figure 2-3 : Amazon White Board.....7

Figure 3-1 : Class Diagram.....11

Figure 4-1 : JSON.....13

Figure 4-2 : Screenshot 1.....15

Figure 4-3: Screenshot 2.....16

Figure 4-4 : Screenshot 3.....16

Figure 4-5 : Screenshot 4.....17



# Table of Contents

---

<b>CHAPTER 1 .</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	Overview .....	1
1.2	Background and Motivation .....	2
1.3	Problem Statement and Objectives .....	2
1.4	Scope of the Project .....	3
1.5	Team Organization .....	4
1.6	Report Structure .....	4
<b>CHAPTER 2 .</b>	<b>REVIEW OF LITERATURE .....</b>	<b>6</b>
2.1	Preliminary Investigation .....	6
2.1.1	Current System .....	6
2.2	Limitations of Current System .....	8
2.3	Requirement Identification and Analysis for Project .....	8
2.3.1	Conclusion .....	8
<b>CHAPTER 3 .</b>	<b>PROPOSED SYSTEM .....</b>	<b>9</b>
3.1	The Proposal .....	9
3.2	Benefits of the Proposed System.....	9
3.3	Block Diagram .....	9
3.4	Feasibility Study .....	10
3.4.1	Technical .....	10
3.4.2	Economical.....	10
3.4.3	Operational .....	10
3.5	Design Representation.....	11
3.5.1	Data Flow Diagrams .....	20
3.6	Deployment Requirements .....	11
3.6.1	Hardware .....	11
3.6.2	Software.....	11

<b>CHAPTER 4 .</b>	<b>IMPLEMENTATION .....</b>	<b>12</b>
<b>CHAPTER 5 .</b>	<b>CONCLUSION.....</b>	<b>19</b>
5.1	Conclusion .....	19
5.2	Limitations of the Work .....	19
5.3	Suggestion and Recommendations for Future Work .....	20
<b>BIBLIOGRAPHY .....</b>		<b>21</b>
<b>GUIDE INTERACTION SHEET.....</b>		<b>23</b>
<b>SOURCE CODE.....</b>		<b>24</b>

# Chapter 1 . Introduction

## Introduction

---

Digital White Board is a desktop application build on Qt 5.14.2 using C++. This application will provide various features like drawing shapes (such as a line, rectangle , circle) , a freehand tool for writing , grouping/ungrouping of shapes drawn , inserting images, clearing the canvas ,save/open files that are drawn on the canvas. It can used in a variety of settings, including classrooms at all levels of education, in corporate board rooms and work groups, in training rooms for professional sports coaching, in broadcasting studios, and others.

### 1.1 Overview

Our Digital White Board provides whiteboard takes you a step beyond basic ideation by helping you organize and synthesize information. Use assisted grouping to gather similar ideas together or use tags to label and organize different items based on whatever identifier you choose. With containers for shapes and sticky notes, you can ensure that your ideas are as clearly visualized on your whiteboard as they are in your head. It doesn't matter whether you're working individually or as part of a team—an online digital education tool can help you jot down and organize your ideas faster. With templates, freehand drawing, sticky notes, and other intuitive features, you can easily participate in a variety of creative brainstorming activities. Our infinite canvas lets you and your collaborators spread out over different areas of the board so everyone has enough space to present their ideas. Our digital whiteboard takes you a step beyond basic ideation by helping you organize and synthesize information. Use assisted grouping to gather similar ideas together or use tags to label and organize different items based on whatever identifier you choose. With containers for shapes and sticky notes, you can ensure that your ideas are as clearly visualized on your whiteboard as they are in your head.

## **1.2 Background and Motivation**

The best web based learning objectives is significant in executing web based learning as it would help in making extraordinary outcomes on the off chance that one realizes how to utilize it. The critical achievement of powerful web based learning isn't just to define objectives yet to lay out the correct objectives. Subsequently, it is essential to comprehend the sorts of various objectives and its extraordinary contrasts on the grounds that every objective has its own motivations.

Also, if web based learning is thought of and set up without the great objectives as a main priority, it will be an exercise in futility and cash as the issues should be tended to again from different viewpoints. Hence, to lay out the great objectives all along, one should know the various types of objectives dependent on the circumstance and the learning circumstance. To lay out the great objectives, one can follow a basic technique which is made to comprehend the more profound circumstance on internet learning.

## **1.3 Problem Statement and Objectives**

It doesn't matter whether you're working individually or as part of a team— an online digital education tool can help you jot down and organize your ideas faster. With templates, freehand drawing, sticky notes, and other intuitive features, you can easily participate in a variety of creative brainstorming activities. Our infinite canvas lets you and your collaborators spread out over different areas of the board so everyone has enough space to present their ideas.

- There will be a virtual white board available on which user can write and draw with the help of free hand tool.
- There will be a feature to draw shapes such as circle, rectangle and line.
- There will be a feature by which user can select the drawing components.
- There will be a feature of drag and drop by which user can move components.
- There will be a feature of resizing components.
- There will be a feature of inserting images on virtual board and also have feature of clearing the virtual white board canvas.

## **1.4 Scope of the Project**

As the world is moving forward towards digitalization so keeping the education as 1st priority for students as well as the faculties. So, keeping all the scenarios in mind the currently available digital education tools are somehow difficult to use for normal people. Since the pandemic happened everything is going online, including online classes for students and there are very limited number of resources available for the teachers to teach students digitally without a physical board so for this we are creating a digital education tool which is much easier as compared to other existing tools so that teachers can use this tool for teaching students digitally.

Our digital whiteboard takes you a step beyond basic ideation by helping you organize and synthesize information. Use assisted grouping to gather similar ideas together or use tags to label and organize different items based on whatever identifier you choose. With containers for shapes and sticky notes, you can ensure that your ideas are as clearly visualized on your whiteboard as they are in your head.

## **1.5 Team Organization**

- **Rahul Choudhary:**

Along with doing preliminary investigation and understanding the limitations of current system, I studied about the topic and its scope and surveyed various research papers related to the available education tool. Some part of implementation logic for the project objective and coding of internal functionalities is also done by me.

Documentation is also a part of the work done by me in this project.

- **Omansh Kalra:**

I investigated and found the right technology and studied in deep about it. For the implementation of the project. Some part of implementation logic for the project objective and coding of internal functionalities is also done by me.

Documentation is also a part of the work done by me in this pr

## **1.6 Report Structure**

Whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discuss the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

# **Chapter 2 . Review of Literature**

## **Review of Literature**

---

Lesson study is one of the most adopted models of teacher professional development. However, as education has become increasingly digital, this study aims to investigate the use of digital tools to support teacher professional development in lesson studies. While the lesson study model is typically based on the premise that teachers prepare and observe a lesson at a school, the reviewed research suggests that digital tools open new ways to conduct lesson studies. Six themes on the use of digital tools to support teacher professional development in lesson studies are identified: analyzing videos from the teachers' classrooms, analyzing external video resources, fictional animations as a complement to videos, structured digital lesson study work, hybrid teacher collaboration and digital teacher collaboration. Opportunities for further research are suggested. The identified themes can inspire practice on how to use digital tools to support teacher professional development in lesson studies. Little attention has been paid to the use of digital tools to support teacher professional development in lesson studies.

### **2.1 Preliminary Investigation**

#### **2.1.1 Current System**

There are several tools available such as paint, google meet and zoom for digital education which contains pen with different colors and sizes can insert images and shapes but they also comes with some disadvantages such as these tools does not contains subject specific teaching assistance.



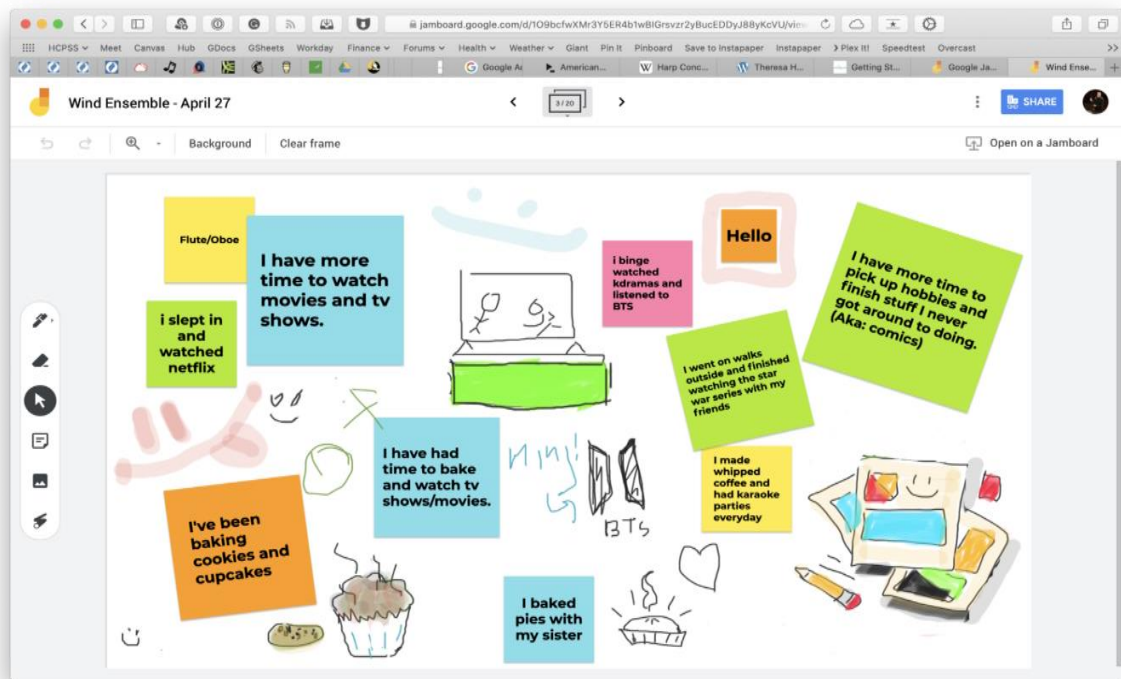


Fig 2-1: Google Jam Board



Fig 2-2: Zoom Whiteboard

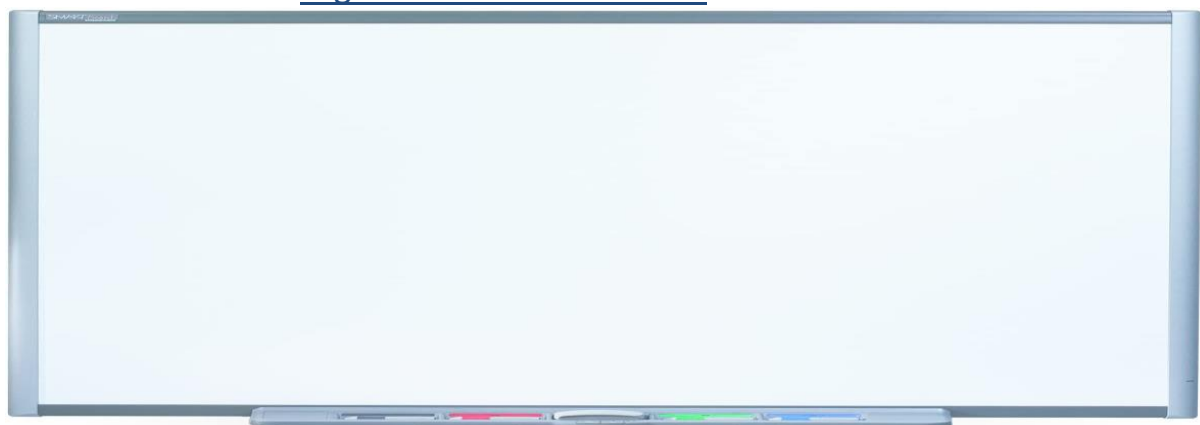


Fig 2-3: Amazon White Board

## **2.2 Limitations of Current System**

- This study and research depends on essential information gathered from understudy and instructors by overview strategy. As a considerable lot of the educators and understudy outfitted, they require data their circumstance and condition during web based learning. The gathered information would be exposed to review inclination.
- Covid-19 also affected the study system.
- Responses submitted by the respondents are not appropriate due to unwilling of respondents to fill up the questionnaires.
- Same responses submitted by the respondents.
- Respondents are not able to understand the questions lack of reliability and authenticity of the responses.
- Recently due to COVID – 19 surge cases, personal visits were less.
- Teacher were very less involved.
- No or very less computer base knowledge in primary school taught to students.
- Lack of infrastructure and hardware facilities which hamper the reliability of online learning.

## **2.3 Conclusion**

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. The research papers related to Digital White Board and how education has been transformed before and after the pandemic.

# Chapter 3 . Proposed System

## Proposed System

---

### 3.1 The Proposal

As COVID-19 is continuing to impact people around the world, thousands of schools and teachers are using Whiteboard as a key tool to provide the best remote learning experience and to help students stay engaged during remote learning. With everyone staying and working from home these days, due to the COVID-19 lockdowns, the amount of online meetings and video conferences have skyrocketed. Many schools and design companies require systems that allow for collaborative sessions. Major problem with the available online whiteboards is that most of the boards are locked behind a money vault or most of the features of the whiteboard are needed to be purchased, which makes it difficult for small organizations to afford the board. Hence making it difficult for virtual interactions and idea sharing. Our whiteboard strives to solve these problems. Taking the magical simplicity of an analog whiteboard and adding interactive, collaborative technology, our Whiteboard gives the whole class a new space to engage, ideate, and create in real time. It will a free to use online whiteboard that anyone can use without paying huge sums of money. As the world is moving forward towards digitalization so keeping the education as 1st priority for students as well as the faculties. So, keeping all the scenarios in mind the currently available digital white boards are somehow difficult to use for normal people. Since the pandemic happened everything is going online, including online classes for students and there are very limited number of resources available for the teachers to teach students digitally without a physical board so for this we are creating a digital white board which is much easier as compared to other existing tools so that teachers can use this tool for teaching students digitally.

### 3.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by this system:

- **Economic :** The proposed system is economic as there is not at all requirement for subscriptions and all.

- It is developed keeping in mind that it can be used for a particular subject also.
- User friendly.
- Easy to use.

### 3.3 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

#### 3.3.1 Technical

For creating Digital White Board , there is a need to create GUI for users. For this, the kind of Tool used must be the one that is user friendly and accurately. The Tool used in this is Qt, Qt is used for developing graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms. Most GUI programs created with Qt have a native-looking interface, in which case Qt is classified as a *widget toolkit*. Non-GUI programs can also be developed, such as command-line tools and consoles for servers. An example of such a non-GUI program using Qt is the Cutelyst web framework., making the system technically feasible. Qt supports various compilers, including the GCC C++ compiler, the Visual Studio suite. Qt also provides Qt Quick, that includes a declarative scripting language called QML that allows using JavaScript to provide the logic. With Qt Quick, rapid application development for mobile devices became possible, while logic can still be written with native code as well to achieve the best possible performance.

#### 3.3.2 Economical

For Digital White Board, there is a need of a High Configuration Machine for better and accurate performance .

## 3.4 CLASS DIAGRAM

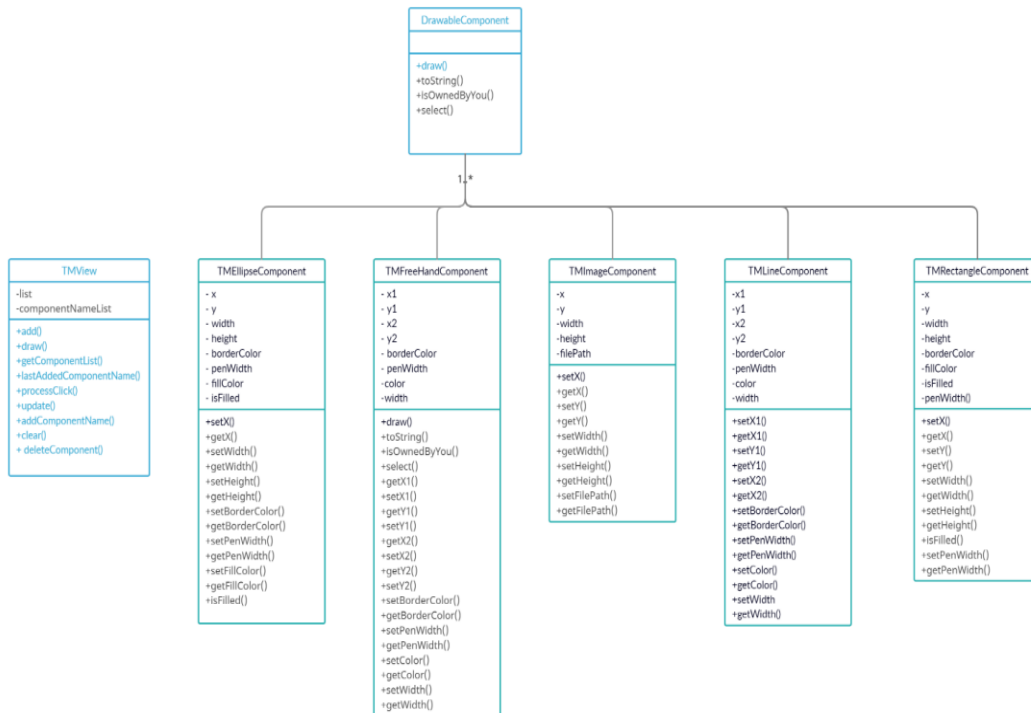


Fig 3-1: Class Diagram

## 3.5 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below :

### 3.5.1 Hardware

- 32-bit, x86 Processing system
- Windows 7 or later operating system
- High processing computer system without GPU or with GPU(high performance)

### 3.5.2 Software

- Qt
- Mingw g++

# Chapter 4. Implementation

## Implementation

---

For Creating Digital White Board we have used C++ programming language and Qt. C++ is a cross-platform language that can be used to create high-performance applications. C++ was developed by Bjarne Stroustrup, as an extension to the C language. C++ gives programmers a high level of control over system resources and memory. The language was updated 3 major times in 2011, 2014, and 2017 to C++11, C++14, and C++17. Qt is used for developing graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms. Most GUI programs created with Qt have a native-looking interface, in which case Qt is classified as a widget toolkit. Non-GUI programs can also be developed, such as command-line tools and consoles for servers. An example of such a non-GUI program using Qt is the Cutelyst web framework.

### 4.1 Tools Used

#### 4.1.1 Qt

Qt is a cross-platform application development framework for desktop, embedded and mobile. Supported Platforms include Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS and others.

Qt is *not* a programming language on its own. It is a framework written in C++. A pre processor, the MOC (Meta-Object Compiler), is used to extend the C++ language with features like signals and slots. Before the compilation step, the MOC parses the source files written in Qt-extended C++ and generates standard compliant C++ sources from them. Thus the framework itself and applications/libraries using it can be compiled by any standard compliant C++ compiler like Clang, GCC, ICC, MinGW and MSVC.

### 4.1.2 JSON

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa). You'll come across it quite often, so in this article we give you all you need to work with JSON using JavaScript, including parsing JSON so you can access data within it, and creating JSON.

---

```
{
  "info"           : info,
  "images"         : [image],
  "annotations"    : [annotation],
  "licenses"       : [license],
}

info{
  "year"           : int,
  "version"        : str,
  "description"    : str,
  "contributor"    : str,
  "url"            : str,
  "date_created"   : datetime,
}

image{
  "id"             : int,
  "width"          : int,
  "height"         : int,
  "file_name"      : str,
  "license"        : int,
  "flickr_url"     : str,
  "coco_url"       : str,
  "date_captured"  : datetime,
}
```

[Fig 4-1: JSON](#)

## 4.2 Language Used

C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.

C++ is regarded as a **middle-level** language, as it comprises a combination of both high-level and low-level language features.

C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.

C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

**Note** – A programming language is said to use static typing when type checking is performed during compile-time as opposed to run-time.

### Object-Oriented Programming

C++ fully supports object-oriented programming, including the four pillars of object-oriented development –

- Encapsulation
- Data hiding
- Inheritance
- Polymorphism

### Standard Libraries

Standard C++ consists of three important parts –

- The core language giving all the building blocks including variables, data types and literals, etc.
- The C++ Standard Library giving a rich set of functions manipulating files, strings, etc.
- The Standard Template Library (STL) giving a rich set of methods manipulating data structures, etc.

### Learning C++

The most important thing while learning C++ is to focus on concepts.

The purpose of learning a programming language is to become a better programmer; that is, to become more effective at designing and implementing new systems and at maintaining old ones.

C++ supports a variety of programming styles. You can write in the style of Fortran, C, Smalltalk, etc., in any language. Each style can achieve its aims effectively while maintaining runtime and space efficiency.



## Use of C++

C++ is used by hundreds of thousands of programmers in essentially every application domain.

C++ is being highly used to write device drivers and other software that rely on direct manipulation of hardware under realtime constraints.

C++ is widely used for teaching and research because it is clean enough for successful teaching of basic concepts.

Anyone who has used either an Apple Macintosh or a PC running Windows has indirectly used C++ because the primary user interfaces of these systems are written in C++.

## 4.3 Screenshots

The Following are the screenshots of the result of the project :

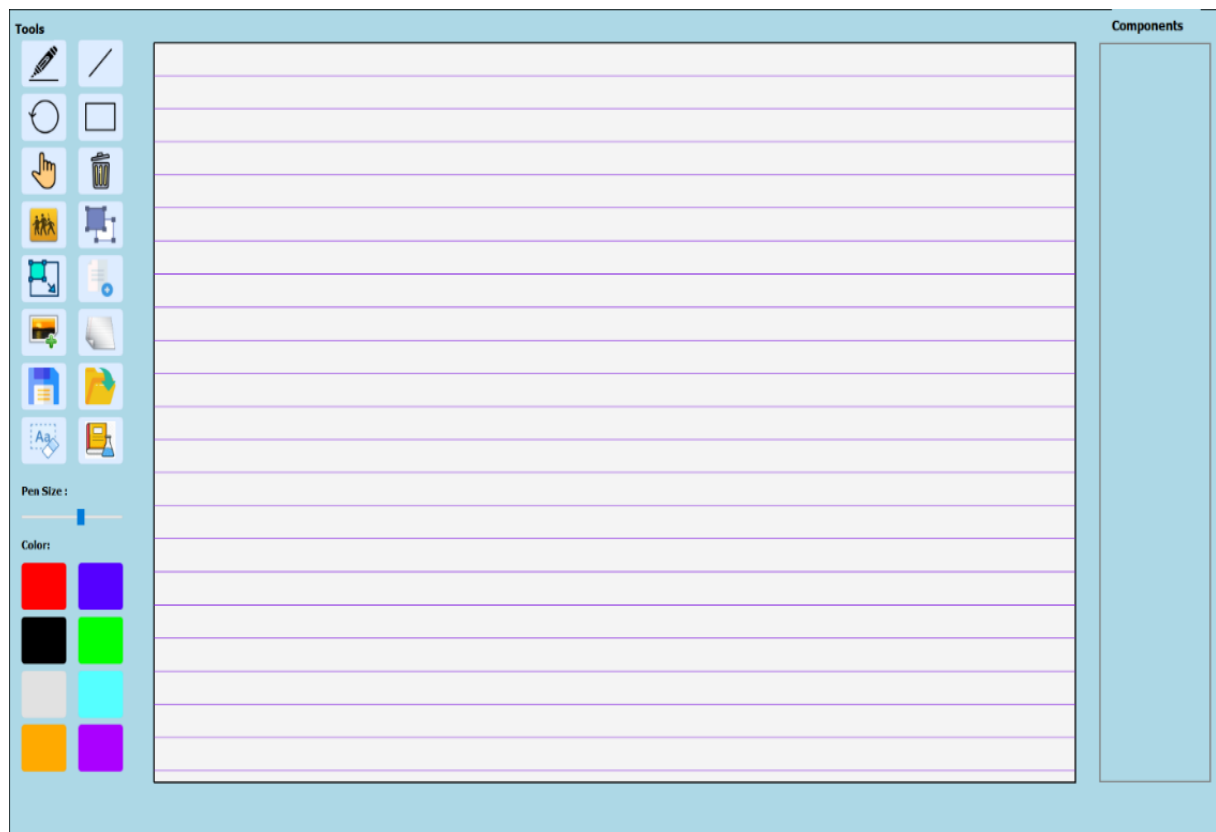


Fig 4-2: Screenshot 1

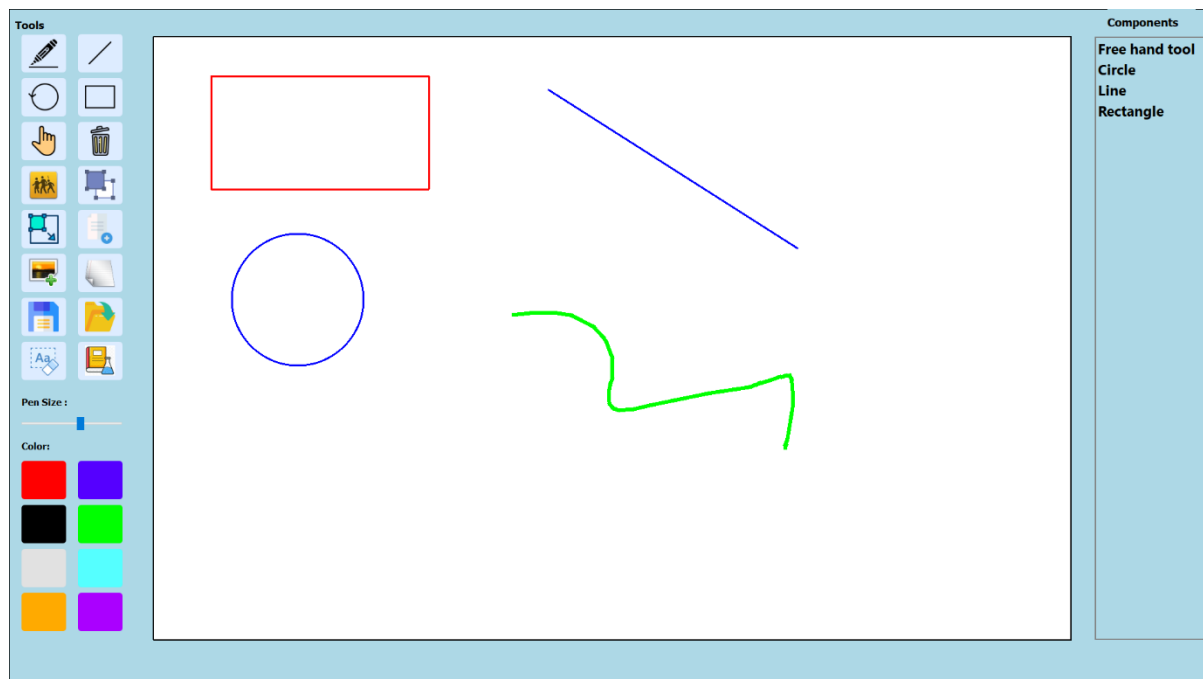


Fig 4-3: Screenshot 2

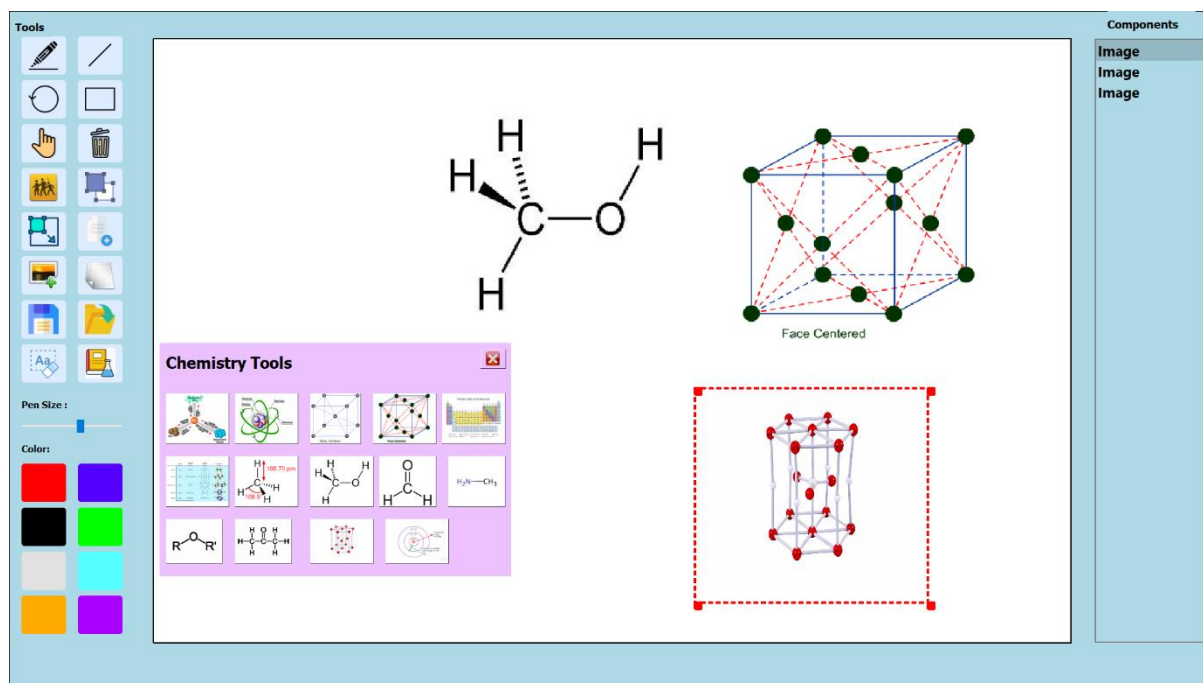


Fig 4-4: Screenshot 3

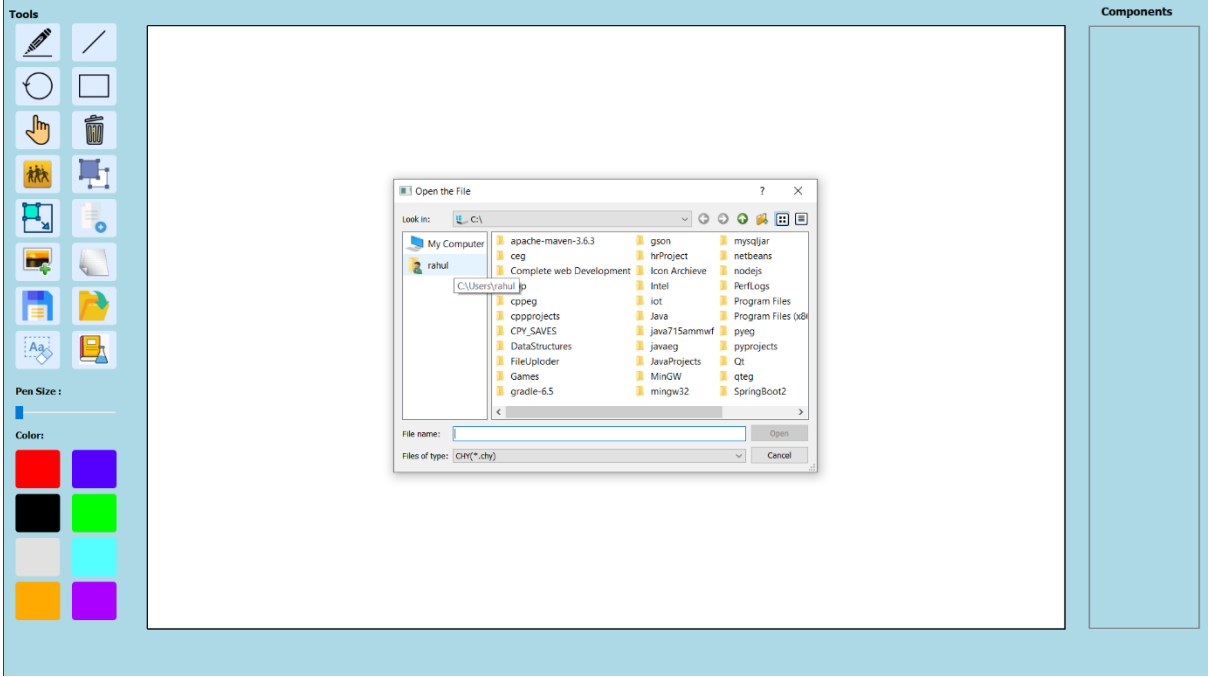


Fig 4-5: Screenshot 4

## **4.4 Testing**

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

### **4.4.1 Strategy Used**

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

The testing method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

# **Chapter 5. Conclusion**

## **Conclusion**

---

### **5.1 Conclusion**

Online Whiteboard is an application that allows user to draw/ write important points during a meeting and can also be used for teaching purposes while it helps the teacher to explain the topic in detail while taking the classes online. It can be used in meeting and corporate discussions to provide a streamlined work management. The whiteboard can be used in development cycles in firms to share and collaborate, to make mind maps and increase efficiency. The uses of an online whiteboard are as limited as your imagination. Learning tools and technology allow students to grow effective self-directed learning skills. The students are capable of analyzing what they require to learn to search and use online resources. Digital learning enlarges their efficiency and productivity. Furthermore, to appealing students, digital learning tools, and technology sharpen critical thinking skills which are the basis for the growth of systematic reasoning. Students also develop positive feelings of exploitation by mastering new knowledge and skills using digitized learning tools, offering them the confidence they require to want to learn even more new things. One of the major advantages of digital learning is it expanded learning opportunities for students. The digital learning is extending learning opportunities so the students can grasp the fact that learning occurs various times and in several places, and digital learning will allow this varied learning opportunity. Students learn many new things from digital tools and technology.

### **5.2 Limitations of the Work**

Now coming to the end of the presentation, the digital white board developed is much more handy , easier to use for almost everyone , and for nowadays it has become a necessity for the society as it requires less time and less hard work to become knowledgeable. But every good thing comes

with a limitation and in this application it is screen recording, but we will overcome this in a very short period of time.

### **5.3 Suggestion and Recommendations for Future Work**

- Currently it does not have any recording feature , we will try to add it .
- We will try to add screen sharing option in our tool.

# **Bibliography**

- [1] Prasetya Didik Dwi, and Muhammad Ashar. "Design of Interactive Whiteboard to Support E-Learning." [www.Atlantis-Press.com](http://www.Atlantis-Press.com), Atlantis Press, 1 Nov. 2017, [www.atlantis-press.com/proceedings/icovet-17/2588641](http://www.atlantis-press.com/proceedings/icovet-17/2588641)
- [2] Ringe, Swati & Kedia, Rishabh & Poddar, Anuj & Patel, Sahil. (2015). HTML5 Based Virtual Whiteboard for Real Time Interaction. *Procedia Computer Science*. 49. 170-177. 10.1016/j.procs.2015.04.241.
- [3] Lant, Carol Le, and Michael J. Lawson. "Interactive Whiteboard Use and Student Engagement." *Publishing Higher Degree Research*, 2016, pp. 33–42, 10.1007/978-94-6300-672-9\_4.
- [4] American Council on Education. (1995). *Computers, technology, and people with disabilities*. Washington, DC: American Council on Education.
- [5] Anderson, R.E., and Ronnkvist, A. (1999). *The presence of computers in American schools*. Irvine, CA: Center for Research on Information Technology and Organizations, University of California, Irvine.
- [6] Allen, N., Kline, D., and Zelenal, C. (1997). *The NAEP 1994 technical report* (NCES 97-897). U.S. Department of Education. Washington, DC: National Center for Education Statistics.

[7] Ballator, Nada. (1997). The NAEP guide: A description of the content and methods of the 1994 and 1996 Assessments (NCES 97-586). U.S. Department of Education. Washington, DC: National Center for Education Statistics.

[8] Ball, D. L. (1990). "Reflections and deflections of policy: The case of Carol Turner." *Educational Evaluation and Policy Analysis*, 12(3), 263-275.

[9] Bare, J., and Meek, A. (1998). Internet access in public schools (NCES 98-031). U.S. Department of Education. Washington, DC: National Center for Education Statistics.



## Guide Interaction Sheet

---

Date	Discussion	Action Plan
27/09/2021	Discussed about the title of the Project	Digital White Board was Decided as project title.
29/09/2021	Discussion on the technology to be used for Digital White Board	C++and QT other tools were finalized
07/10/2021	Discussion of the creation of synopsis of the project	Gathering of information for synopsis creation
12/03/2022	Suggestions on how to do a literature survey and preliminary investigation on the topic	Many research papers were read , understood and their abstract were to be written.
01/04/2022	Discussion on the implementation of the project	Using QT and other tools, we decided to implement UI Part.

# Source Code

```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QIcon>
#include <QPixmap>
#include <QPainter>
#include <QFont>
#include <QtMath>
#include <QMessageBox>
#include <QJsonValue>
#include <QJsonObject>
#include <QJsonDocument>
#include <QFile>
#include <QTextStream>
#include <QFileDialog>
#include <QDebug>
#include <QJsonArray>
QListWidgetItem *item1;
bool deleteComponentFlag=false;
bool createLineFlag=false;
bool resizingComponent=false;
bool insertImage=false;
int penWidth;
QString colorOfBorder="";
QColor *borderColor;
bool redColorFlag=false;
bool blueColorFlag=false;
bool blackColorFlag=false;
bool greenColorFlag=false;
bool greyColorFlag=false;
bool lightBlueColorFlag=false;
bool orangeColorFlag=false;
bool violetColorFlag=false;
bool controlKeyIsPressed=false;
bool clearComponentFlag=false;
QList<DrawableComponent *> listForSelection;
QList<DrawableComponent *> groupList;
int posx1=0;
int posx2=0;
int posX=0;
int posY=0;
int i=0;
int j=0;
int t=0;
int u=0;
int groupSelect=0;
int row=0;
int column=0;
int select=-1;
int previousX;
int previousY;
int posX2;
int posY2;
int X1;
int Y1;
int MainWindow::oldX=-1;

```

```

int MainWindow::oldY=-1;
int MainWindow::currentX=0;
int MainWindow::currentY=0;
int var=0;
TMView tmView;
DrawableComponent *d;
QModelIndex index;
QList<DrawableComponent *> list;
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setUpUi(this);
    setWindowTitle("Thinking machines");
    this->ui->mainFrame->setVisible(false);
    ui->listWidget->
>setSelectionMode(QAbstractItemView::SelectionMode::SingleSelection);
    ui->listWidget->setSelectionBehavior(QAbstractItemView::SelectRows);
    QFont font;
    font.setBold(true);
    // ui->viewToolBox->setFont(font);
    model=new QStringListModel(this);
    //this->setStyleSheet("background-color:black");
    ui->horizontalSlider->setRange(0,10);
    QIcon freeHandIcon(QPixmap(":/images/pencil_icon.png"));
    this->ui->freeHandTool->setIcon(freeHandIcon);
    this->ui->freeHandTool->setToolTip(QString("Free Hand Tool"));
    this->ui->freeHandTool->setIconSize(QSize(48,48));
    QIcon lineIcon(QPixmap(":/images/line_icon.png"));
    this->ui->goToLineInfoPanel->setIcon(lineIcon);
    this->ui->goToLineInfoPanel->setIconSize(QSize(48,48));
    this->ui->goToLineInfoPanel->setToolTip("Line");
    QIcon circleIcon(QPixmap(":/images/circle_icon1.png"));
    this->ui->goToEllipseInfoPanel->setIcon(circleIcon);
    this->ui->goToEllipseInfoPanel->setToolTip("Circle");
    this->ui->goToEllipseInfoPanel->setIconSize(QSize(48,48));
    QIcon rectangleIcon(QPixmap(":/images/rectangle_icon.png"));
    this->ui->goToRectangleInfoPanel->setIcon(rectangleIcon);
    this->ui->goToRectangleInfoPanel->setIconSize(QSize(48,48));
    this->ui->goToRectangleInfoPanel->setToolTip("Rectangle");
    QIcon pickIcon(QPixmap(":/images/pick_tool_icon.png"));
    this->ui->pickTool->setIcon(pickIcon);
    this->ui->pickTool->setIconSize(QSize(48,48));
    this->ui->pickTool->setToolTip("Pick tool");
    QIcon deleteIcon(QPixmap(":/images/delete_icon.png"));
    this->ui->deleteButton->setIcon(deleteIcon);
    this->ui->deleteButton->setIconSize(QSize(48,48));
    this->ui->deleteButton->setToolTip("Delete");
    QIcon groupIcon(QPixmap(":/images/group_icon.png"));
    this->ui->groupButton->setIcon(groupIcon);
    this->ui->groupButton->setIconSize(QSize(48,48));
    this->ui->groupButton->setToolTip("Group");
    QIcon ungroupIcon(QPixmap(":/images/ungroup_icon.png"));
    this->ui->ungroupButton->setIcon(ungroupIcon);
    this->ui->ungroupButton->setIconSize(QSize(48,48));
    this->ui->ungroupButton->setToolTip("Ungroup");
    QIcon saveIcon(QPixmap(":/images/save.png"));
    this->ui->saveButton->setIcon(saveIcon);
    this->ui->saveButton->setToolTip("Save");
    this->ui->saveButton->setIconSize(QSize(48,48));
    QIcon openIcon(QPixmap(":/images/open.png"));
    this->ui->openButton->setIcon(openIcon);
    this->ui->openButton->setIconSize(QSize(48,48));
    this->ui->openButton->setToolTip("Open");
    QIcon resizeIcon(QPixmap(":/images/resize_icon.png"));
    this->ui->resizePushButton->setIcon(resizeIcon);
    this->ui->resizePushButton->setIconSize(QSize(48,48));
    this->ui->resizePushButton->setToolTip("Resize components");
    QIcon addNewPageIcon(":/images/add_new_icon.png");

```

```

this->ui->addNewPage->setIcon(addNewPageIcon);
this->ui->addNewPage->setIconSize(QSize(48,48));
this->ui->addNewPage->setToolTip("Add new page");
QIcon insertImageIcon(":/images/insert_image_icon.png");
this->ui->insertImage->setIcon(insertImageIcon);
this->ui->insertImage->setIconSize(QSize(48,48));
this->ui->insertImage->setToolTip("Insert image");
QIcon createLineIcon(":/images/page-icon.png");
this->ui->createLine->setIcon(createLineIcon);
this->ui->createLine->setIconSize(QSize(48,48));
this->ui->createLine->setToolTip("Insert page");
QIcon clearCanvasIcon(QPixmap(":/images/clear_icon.png"));
this->ui->clearCanvas->setIcon(clearCanvasIcon);
this->ui->clearCanvas->setToolTip("Clear canvas");
this->ui->clearCanvas->setIconSize(QSize(48,48));
QIcon chemistryBookIcon(QPixmap(":/images/chemistry_icon.png"));
this->ui->chemistryBoard->setIcon(chemistryBookIcon);
this->ui->chemistryBoard->setIconSize(QSize(48,48));
this->ui->chemistryBoard->setToolTip("Chemistry tools");
}
int radius;
int p=0;
void MainWindow::paintEvent(QPaintEvent *event)
{
    QRect mainWindowRect(0,0,this->width(),this->height());
    QColor color(173, 216, 230);
    QPainter painter(this);
    painter.setBrush(color);
    painter.drawRect(mainWindowRect);
    /* QRect boardBorder(200+30,40,1492,970);
    QPainter boardBorderPainter(this);

boardBorderPainter.drawPixmap(boardBorder,QPixmap(":/images/board2.png"));
*/
    QRect boundingRect(200+30,44,1492-30,961);
    QPen boundingPen;
    boundingPen.setColor(Qt::black);
    boundingPen.setWidth(2);
    QPainter p(this);
    p.eraseRect(QRect(200+30,44,1492-30,961));
    p.setBrush(Qt::white);
    p.setPen(boundingPen);
    p.drawRect(boundingRect);
    p.setClipRect(boundingRect,Qt::ReplaceClip);
    if(createLineFlag==true)
    {
        // QPainter linePainter(this);
        QRect lineCanvasRect(200+30,44,1492-30,961);
        p.setBrush(QColor(244,244,244));
        p.drawRect(lineCanvasRect);
        int w=0;
        while(w<961)
        {
            w=w+43;
            //p.setPen(QColor(47,79,79));
            p.setPen(QColor(138,43,226));
            p.drawLine(230,w,1692,w);
        }
    }
    p.setBrush(Qt::NoBrush);
    tmView.draw(&p);

if(clearComponentFlag!=true)
{
    if(i==5)
    {
        select=tmView.processClick(currentX,currentY,&p);
        if(controlKeyIsPressed==false)
        {

```

```

        listForSelection.clear();
        int y=0;
        while(y<listForSelection.size())
        {
            listForSelection.removeAt(y);
            y++;
        }
        if(select==(-1))
        {
            return;
        }
        DrawableComponent *d=tmView.getComponentList()[select];
        listForSelection.push_back(d);
    }
    else
    {
        if(select==(-1)) return;
        DrawableComponent *d=tmView.getComponentList()[select];
        listForSelection.push_back(d);
        int i=0;
        while(i<listForSelection.size())
        {
            listForSelection[i]->select(&p);
            i++;
        }
        qDebug()<<i;
    }

    if(groupSelect==1)
    {
        int e=0;
        while(e<groupList.size())
        {
            groupList[e]->select(&p);
            e++;
        }
    }
    this->ui->listWidget->setCurrentRow(select);
    this->ui->listWidget->currentItem();
    j=0;
    u=1;
}

if(i==1)
{
    QPen pen;
    if(colorOfBorder.compare("")==0)
    {
        pen.setColor(Qt::black);
    }
    else
    {
        pen.setColor(QColor(colorOfBorder));
    }
    pen.setWidth(this->ui->horizontalSlider->value());
    p.setPen(pen);
    p.drawLine(oldX,oldY,currentX,currentY);
}
if(i==2)
{
    QPen pen;
    if(colorOfBorder.compare("")==0)
    {
        pen.setColor(Qt::black);
    }
    else
    {
        pen.setColor(QColor(colorOfBorder));
    }
}

```

```

        pen.setWidth(this->ui->horizontalSlider->value());
        p.setPen(pen);
        p.drawLine(oldX,oldY,currentX,currentY);
    }
    if(i==3)
    {
        QPen pen;
        /*if(borderColor!=NULL)
        {
            pen.setColor(*borderColor);
        }
        else
        {
            pen.setColor(QColor(Qt::black));
        }
        pen.setWidth(this->ui->horizontalSlider->value());
        p.setPen(pen);
        */
        if(colorOfBorder.compare("")==0)
        {
            pen.setColor(Qt::black);
        }
        else
        {
            pen.setColor(QColor(colorOfBorder));
        }
        pen.setWidth(this->ui->horizontalSlider->value());
        p.setPen(pen);
        p.drawRect(oldX,oldY,currentX-oldX,currentY-oldY);
    }
    if(i==4)
    {
        QPen pen;
        if(colorOfBorder.compare("")==0)
        {
            pen.setColor(Qt::black);
        }
        else
        {
            pen.setColor(QColor(colorOfBorder));
        }
        pen.setWidth(this->ui->horizontalSlider->value());
        p.setPen(pen);
        radius= qSqrt(qPow((currentX-oldX),2)+qPow((currentY-oldY),2));
        p.drawEllipse(QPointF(oldX,oldY),radius,radius);
    }
    if(t==1)
    {
        tmView.getComponentList()[row]->select(&p);
        t=0;
    }
    if(redColorFlag==true)
    {
        borderColor=new QColor(255,0,0);
        redColorFlag=false;
    }
    if(blueColorFlag==true)
    {
        borderColor=new QColor(85, 0, 255);
        blueColorFlag=false;
    }
    if(blackColorFlag==true)
    {
        borderColor=new QColor(0,0,0);
        blackColorFlag=false;
    }
    if(greenColorFlag==true)
    {
        borderColor=new QColor(0,255,0);
    }

```

```

        greenColorFlag=false;
    }
    if(greyColorFlag==true)
    {
        borderColor=new QColor(169,169,169);
        greyColorFlag=false;
    }
    if(lightBlueColorFlag==true)
    {
        borderColor=new QColor(85,255,255);
        lightBlueColorFlag=false;
    }
    if(orangeColorFlag==true)
    {
        borderColor=new QColor(255,170,0);
        orangeColorFlag=false;
    }
    if(voiletColorFlag==true)
    {
        borderColor=new QColor(170,0,255);
        voiletColorFlag=false;
    }
}
clearComponentFlag=false;
}
void MainWindow::mousePressEvent(QMouseEvent *e)
{
    oldX=e->x();
    oldY=e->y();
    if(i==5)
    {
        if(controlKeyIsPressed==true)
        {
            select=tmView.processClick(oldX,oldY,NULL);
            if(select==(-1)) return;
            groupList.push_back(tmView.getComponentList()[select]);
        }
    }
}
void MainWindow::mouseMoveEvent(QMouseEvent *e)
{
    currentX=e->x();
    currentY=e->y();
    if(i==1)
    {
        TMFreeHandComponent *tmFreeHandComponent=new TMFreeHandComponent;
        tmFreeHandComponent->setX1(oldX);
        tmFreeHandComponent->setY1(oldY);
        tmFreeHandComponent->setX2(currentX);
        tmFreeHandComponent->setY2(currentY);
        tmFreeHandComponent->setBorderColor(colorOfBorder);
        tmFreeHandComponent->setPenWidth(this->ui->horizontalSlider->value());
        tmView.add(tmFreeHandComponent);
        QWidget::update();
        oldX=currentX;
        oldY=currentY;
    }
    if(i==2)
    {
        QWidget::update();
    }
    if(i==3)
    {
        //TMRectangleComponent *tmRectangleComponent=new
TMRectangleComponent;
        //tmRectangleComponent->setX(oldX);
        //tmRectangleComponent->setY(oldY);
        //tmRectangleComponent->setWidth(currentX-oldX);

```

```

        //tmRectangleComponent->setHeight(currentY-oldY);
        QWidget::update();
    }
    if(i==4)
    {
        QWidget::update();
    }
    if(i==5)
    {
        int diffX;
        int diffY;
        diffX=currentX-oldX;
        diffY=currentY-oldY;
        oldX=currentX;
        oldY=currentY;

        if(groupSelect==0)
        {
            if(select==(-1)) return;
            DrawableComponent *d=tmView.getComponentList()[select];
            if(d->toString().compare("Rectangle")==0)
            {
                TMRectangleComponent *t=(TMRectangleComponent *)d;
                t->setX(t->getX()+(diffX));
                t->setY(t->getY()+(diffY));
            }
            if(d->toString().compare("Circle")==0)
            {
                TMEllipseComponent *t=(TMEllipseComponent *)d;
                //t->setX(currentX+posX);
                //t->setY(currentY+posY);
                t->setX(t->getX()+(diffX));
                t->setY(t->getY()+(diffY));
            }
            if(d->toString().compare("Line")==0)
            {
                TMLineComponent *t=(TMLineComponent *)d;
                t->setX1(t->getX1()+(diffX));
                t->setY1(t->getY1()+(diffY));
                t->setX2(t->getX2()+(diffX));
                t->setY2(t->getY2()+(diffY));
            }
            if(d->toString().compare("Image")==0)
            {
                TMImageComponent *t=(TMImageComponent *)d;
                t->setX(t->getX()+(diffX));
                t->setY(t->getY()+(diffY));
            }
        }
        else
        {
            if(select==(-1)) return;
            DrawableComponent *d=tmView.getComponentList()[select];
            if(d->toString().compare("Rectangle")==0)
            {
                TMRectangleComponent *t=(TMRectangleComponent *)d;
                t->setX(t->getX()+(diffX));
                t->setY(t->getY()+(diffY));
            }
            //add kiya
            if(d->toString().compare("Circle")==0)
            {
                TMEllipseComponent *t=(TMEllipseComponent *)d;
                t->setX(t->getX()+(diffX));
                t->setY(t->getY()+(diffY));
            }
            if(d->toString().compare("Line")==0)
            {
                TMLineComponent *t=(TMLineComponent *)d;

```



```

        t->setX1 (t->getX1 ()+diffX);
        t->setY1 (t->getY1 ()+diffY);
        t->setX2 (t->getX2 ()+diffX);
        t->setY2 (t->getY2 ()+diffY);
    }
    if (d->toString().compare("Image")==0)
    {
        TMTImageComponent *t=(TMTImageComponent *)d;
        t->setX(t->getX()+(diffX));
        t->setY(t->getY()+(diffY));
    }
    //yaha tak
    DrawableComponent *m;
    int k=0;
    while(k<groupList.size())
    {
        m=groupList.at(k);
        if (d==m)
        {
            k++;
            continue;
        }
        if (m->toString().compare("Rectangle")==0)
        {
            TMRectangleComponent *t=(TMRectangleComponent *)m;
            t->setX(t->getX()+(diffX));
            t->setY(t->getY()+(diffY));
        }
        if (m->toString().compare("Circle")==0)
        {
            TMEllipseComponent *t=(TMEllipseComponent *)m;
            t->setX(t->getX()+(diffX));
            t->setY(t->getY()+(diffY));
        }
        if (m->toString().compare("Line")==0)
        {
            TMLineComponent *t=(TMLineComponent *)m;
            t->setX1 (t->getX1 ()+diffX);
            t->setY1 (t->getY1 ()+diffY);
            t->setX2 (t->getX2 ()+diffX);
            t->setY2 (t->getY2 ()+diffY);
        }
        if (m->toString().compare("Image")==0)
        {
            TMTImageComponent *t=(TMTImageComponent *)m;
            t->setX(t->getX()+(diffX));
            t->setY(t->getY()+(diffY));
        }
        k++;
    }
    QWidget::update();
}
QWidget::update();
}

if (resizingComponent==true)
{
    if (select==(-1)) return;
    int diffX=currentX-oldX;
    int diffY=currentY-oldY;
    DrawableComponent *m=tmView.getComponentList()[select];
    if (m->toString().compare("Rectangle")==0)
    {
        TMRectangleComponent *t=(TMRectangleComponent *)m;
        if (oldX==t->getX() && oldY==t->getY())
        {
            qDebug()<<"Clicked1";
            t->setX(t->getX()+(diffX));
            t->setY(t->getY()+(diffY));
        }
    }
}

```

```

t->setWidth(t->getWidth()-diffX);
t->setHeight(t->getHeight()-diffY);
}
if(oldX==t->getX()+t->getWidth() && t->getY()==oldY)
{
    t->setY(t->getY()+diffY);
    t->setWidth(t->getWidth()+diffX);
    t->setHeight(t->getHeight()-diffY);
}
if(oldX==t->getX() && oldY==t->getY()+t->getHeight())
{
    qDebug()<<"Clicked";
    t->setX(t->getX()+diffX);
    t->setWidth(t->getWidth()-diffX);
    t->setHeight(t->getHeight()+diffY);
}
if(oldX==t->getX()+t->getWidth() && oldY==t->getY()+t->getHeight())
{
    t->setWidth(t->getWidth()+diffX);
    t->setHeight(t->getHeight()+diffY);
}
}
if(m->toString().compare("Circle")==0)
{
    TMEllipseComponent *t=(TMEllipseComponent *)m;
    t->setWidth(t->getWidth()-diffY);
    t->setHeight(t->getWidth());
    /*if(oldX==t->getX() && oldY==(t->getY()-t->getWidth()))
    {
        t->setWidth(t->getY()-diffY);
        t->setHeight(t->getWidth());
    }*/
}
if(m->toString().compare("Line")==0)
{
    TMLineComponent *t=(TMLineComponent *)m;
    if(t->getX1()==oldX && t->getY1()==oldY)
    {
        t->setX1(t->getX1()+diffX);
        t->setY1(t->getY1()+diffY);
    }
    if(t->getX2()==oldX && t->getY2()==oldY)
    {
        t->setX2(t->getX2()+diffX);
        t->setY2(t->getY2()+diffY);
    }
}
if(m->toString().compare("Image")==0)
{
    TMImageComponent *t=(TMImageComponent *)m;
    if(oldX==t->getX() || oldY==t->getY())
    {
        qDebug()<<"Clicked1";
        t->setX(t->getX()+diffX);
        t->setY(t->getY()+diffY);
        t->setWidth(t->getWidth()-diffX);
        t->setHeight(t->getHeight()-diffY);
    }
    /* if(oldX==t->getX()+t->getWidth() && t->getY()==oldY)
    {
        t->setY(t->getY()+diffY);
        t->setWidth(t->getWidth()+diffX);
        t->setHeight(t->getHeight()-diffY);
    }
    if(oldX==t->getX() && oldY==t->getY()+t->getHeight())
    {
        qDebug()<<"Clicked";
        t->setX(t->getX()+diffX);
    }
}

```

```

        t->setWidth(t->getWidth()-diffX);
        t->setHeight(t->getHeight()+diffY);
    }
    if(oldX==t->getX()+t->getWidth() && oldY==t->getY()+t->getHeight())
    {
        t->setWidth(t->getWidth()+diffX);
        t->setHeight(t->getHeight()+diffY);
    }*/
}
oldX=currentX;
oldY=currentY;
QWidget::update();
}
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::mouseReleaseEvent(QMouseEvent *e)
{
    if(i==1)
    {
        QString name=tmView.lastAddedComponentName();
        this->ui->listWidget->insertItem(0,name);
    }
    if(i==2)
    {
        currentX=e->x();
        currentY=e->y();
        TMLineComponent *tmLineComponent=new TMLineComponent;
        tmLineComponent->setX1(oldX);
        tmLineComponent->setY1(oldY);
        tmLineComponent->setX2(currentX);
        tmLineComponent->setY2(currentY);
        tmLineComponent->setBorderColor(colorOfBorder);
        tmLineComponent->setPenWidth(this->ui->horizontalSlider->value());
        tmView.add(tmLineComponent);
        QString name=tmView.lastAddedComponentName();
        this->ui->listWidget->insertItem(0,name);
        QWidget::update();
    }
    if(i==3)
    {
        TMRectangleComponent *tmRectangleComponent=new
TMRectangleComponent;
        tmRectangleComponent->setX(oldX);
        tmRectangleComponent->setY(oldY);
        //tmRectangleComponent->setIsFilled(true);
        tmRectangleComponent->setWidth(currentX-oldX);
        tmRectangleComponent->setHeight(currentY-oldY);
        tmRectangleComponent->setBorderColor(colorOfBorder);
        tmRectangleComponent->setPenWidth(this->ui->horizontalSlider-
>value());
        //tmRectangleComponent->setFillColor(fillColor);
        tmView.add(tmRectangleComponent);
        QString name=tmView.lastAddedComponentName();
        this->ui->listWidget->insertItem(0,name);
        //this->ui->listWidget->clear();
        //this->ui->listWidget->addItems(tmView.update());
        QWidget::update();
    }
    if(i==4)
    {
        TMEllipseComponent *tmEllipseComponent=new TMEllipseComponent;
        tmEllipseComponent->setX(oldX);

```

```

        tmEllipseComponent->setY(oldY);
        tmEllipseComponent->setWidth(radius);
        tmEllipseComponent->setHeight(radius);
        tmEllipseComponent->setBorderColor(colorOfBorder);
        tmEllipseComponent->setPenWidth(this->ui->horizontalSlider-
>value());
        tmView.add(tmEllipseComponent);
        QString name=tmView.lastAddedComponentName();
        this->ui->listWidget->insertItem(0,name);
        QWidget::update();
    }

    if(i==5)
    {

        currentX=e->x();
        currentY=e->y();
        QWidget::update();
        if(u==1)
        {

            u=0;
        }

    }

}

void MainWindow::on_freeHandTool_clicked()
{
    i=1;
}

void MainWindow::on_goToLineInfoPanel_clicked()
{
    i=2;
}

void MainWindow::on_goToRectangleInfoPanel_clicked()
{
    i=3;
}

void MainWindow::on_goToEllipseInfoPanel_clicked()
{
    i=4;
}

void MainWindow::on_pickTool_clicked()
{
    i=5;
    resizingComponent=false;
}

void MainWindow::on_listWidget_itemClicked(QListWidgetItem *item)
{
    item1=item;
    i=0;
    row=this->ui->listWidget->row(item);
    t=1;
    QWidget::update();
    item->setFlags(item->flags() | Qt::ItemIsEditable);
}

void MainWindow::on_groupButton_clicked()
{
    groupSelect=1;
    i=5;
}

```

```

}

void MainWindow::on_ungroupButton_clicked()
{
    groupSelect=0;
    groupList.clear();
}

void MainWindow::on_saveButton_clicked()
{
    //QString jsonFilter="JSON(*.json)";
    QString jsonFilter="CHY(*.chy)";
    //QString fileName=QFileDialog::getSaveFileName(this,"Open the
file","", "JSON(*.json)", &jsonFilter, QFileDialog::DontUseNativeDialog);
    QString fileName=QFileDialog::getSaveFileName(this,"Open the
file","", "CHY(*.chy)", &jsonFilter, QFileDialog::DontUseNativeDialog);
    QFile output(fileName);
    if(!output.open(QIODevice::WriteOnly|QIODevice::Text))
    {
        QMessageBox::warning(this, "Warning", "Cannot open
File"+output.errorString());
    }
    QJsonDocument document;
    QList<DrawableComponent *> list;
    DrawableComponent *d;
    QByteArray data_json;
    QJsonArray listArray;
    list=tmView.getComponentList();
    int i=0;
    while(i<list.size())
    {
        d=list[i];
        QJsonObject obj;
        if(d->toString().compare("Rectangle")==0)
        {
            TMRectangleComponent *t=(TMRectangleComponent *)d;
            obj.insert("x", QJsonValue::fromVariant(t->getX()));
            obj.insert("y", QJsonValue::fromVariant(t->getY()));
            obj.insert("width", QJsonValue::fromVariant(t->getWidth()));
            obj.insert("height", QJsonValue::fromVariant(t->getHeight()));
            obj.insert("borderColor", QJsonValue::fromVariant(t-
>getBorderColor()));
            obj.insert("penWidth", QJsonValue::fromVariant(t-
>getPenWidth()));
            obj.insert("componentName", QJsonValue::fromVariant(t-
>toString()));
            listArray.push_back(obj);
        }
        if(d->toString().compare("Image")==0)
        {
            TMImageComponent *t=(TMImageComponent *)d;
            obj.insert("x", QJsonValue::fromVariant(t->getX()));
            obj.insert("y", QJsonValue::fromVariant(t->getY()));
            obj.insert("width", QJsonValue::fromVariant(t->getWidth()));
            obj.insert("height", QJsonValue::fromVariant(t->getHeight()));
            obj.insert("filepath", QJsonValue::fromVariant(t-
>getFilePath()));
            obj.insert("componentName", QJsonValue::fromVariant(t-
>toString()));
            listArray.push_back(obj);
        }
        if(d->toString().compare("Circle")==0)
        {
            TMEllipseComponent *t=(TMEllipseComponent *)d;
            obj.insert("x", QJsonValue::fromVariant(t->getX()));
            obj.insert("y", QJsonValue::fromVariant(t->getY()));
            obj.insert("width", QJsonValue::fromVariant(t->getWidth()));
            obj.insert("height", QJsonValue::fromVariant(t->getHeight()));

```

```

        obj.insert("borderColor", QJsonValue::fromVariant(t-
>getBorderColor()));
        obj.insert("penWidth", QJsonValue::fromVariant(t-
>getPenWidth()));
        obj.insert("componentName", QJsonValue::fromVariant(t-
>toString()));
        listArray.push_back(obj);
    }
    if(d->toString().compare("Line")==0)
    {
        TMLineComponent *t=(TMLineComponent *)d;
        obj.insert("x1", QJsonValue::fromVariant(t->getX1()));
        obj.insert("y1", QJsonValue::fromVariant(t->getY1()));
        obj.insert("x2", QJsonValue::fromVariant(t->getX2()));
        obj.insert("y2", QJsonValue::fromVariant(t->getY2()));
        obj.insert("borderColor", QJsonValue::fromVariant(t-
>getBorderColor()));
        obj.insert("penWidth", QJsonValue::fromVariant(t-
>getPenWidth()));
        obj.insert("componentName", QJsonValue::fromVariant(t-
>toString()));
        listArray.push_back(obj);
    }
    if(d->toString().compare("Free hand tool")==0)
    {
        TMFreeHandComponent *t=(TMFreeHandComponent *)d;
        obj.insert("x1", QJsonValue::fromVariant(t->getX1()));
        obj.insert("y1", QJsonValue::fromVariant(t->getY1()));
        obj.insert("x2", QJsonValue::fromVariant(t->getX2()));
        obj.insert("y2", QJsonValue::fromVariant(t->getY2()));
        obj.insert("borderColor", QJsonValue::fromVariant(t-
>getBorderColor()));
        obj.insert("penWidth", QJsonValue::fromVariant(t-
>getPenWidth()));
        obj.insert("componentName", QJsonValue::fromVariant(t-
>toString()));
        listArray.push_back(obj);
    }
    i++;

}
document.setArray(listArray);
data_json=document.toJson();
output.write(data_json);
output.close();
}

void MainWindow::on_openButton_clicked()
{
    QString jsonFilter="CHY(*.chy)";
    QString filename=QFileDialog::getOpenFileName(this, "Open the
File", "/", jsonFilter, &jsonFilter, QFileDialog::DontUseNativeDialog);
    QFile input(filename);
    if(!input.open(QIODevice::ReadOnly|QIODevice::Text))
    {
        QMessageBox::warning(this, "warning", "Cannot open
file"+input.errorString());
    }

    QByteArray data_json=input.readAll();
    input.close();
    QJsonDocument document=QJsonDocument::fromJson(data_json);
    QJsonArray jsonArray=document.array();
    for(QJsonValue value:jsonArray)
    {
        QJsonObject obj(value.toObject());

        if(obj.value(QString("componentName")).toString().compare("Rectangle")==0)
        {

```

```

        TMRectangleComponent *t=new TMRectangleComponent;
        t->setX(obj["x"].toInt());
        t->setY(obj["y"].toInt());
        t->setWidth(obj["width"].toInt());
        t->setHeight(obj["height"].toInt());
        t->setBorderColor(obj["borderColor"].toString());
        t->setPenWidth(obj["penWidth"].toInt());
        tmView.add(t);
    }

    if(obj.value(QString("componentName")).toString().compare("Image")==0)
    {
        TMImageComponent *t=new TMImageComponent;
        t->setX(obj["x"].toInt());
        t->setY(obj["y"].toInt());
        t->setWidth(obj["width"].toInt());
        t->setHeight(obj["height"].toInt());
        t->setFilePath(obj["filepath"].toString());
        tmView.add(t);
    }

    if(obj.value(QString("componentName")).toString().compare("Circle")==0)
    {
        TMEllipseComponent *t=new TMEllipseComponent;
        t->setX(obj["x"].toInt());
        t->setY(obj["y"].toInt());
        t->setWidth(obj["width"].toInt());
        t->setHeight(obj["height"].toInt());
        t->setBorderColor(obj["borderColor"].toString());
        t->setPenWidth(obj["penWidth"].toInt());
        tmView.add(t);
    }
    if(obj.value(QString("componentName")).toString().compare("Free
hand tool")==0)
    {
        TMFreeHandComponent *t=new TMFreeHandComponent;
        t->setX1(obj["x1"].toInt());
        t->setY1(obj["y1"].toInt());
        t->setX2(obj["x2"].toInt());
        t->setY2(obj["y2"].toInt());
        t->setBorderColor(obj["borderColor"].toString());
        t->setPenWidth(obj["penWidth"].toInt());
        tmView.add(t);
        // QWidget::update();
    }

    if(obj.value(QString("componentName")).toString().compare("Line")==0)
    {
        TMLineComponent *t=new TMLineComponent;
        t->setX1(obj["x1"].toInt());
        t->setY1(obj["y1"].toInt());
        t->setX2(obj["x2"].toInt());
        t->setY2(obj["y2"].toInt());
        t->setBorderColor(obj["borderColor"].toString());
        t->setPenWidth(obj["penWidth"].toInt());
        tmView.add(t);
    }
}
QWidget::update();

for(DrawableComponent *d:tmView.getComponentList())
{
    this->ui->listWidget->insertItem(0,d->toString());
}
}

void MainWindow::keyPressEvent(QKeyEvent *event)
{
    if(event->key()==Qt::Key_Control)

```

```

        {
            controlKeyIsPressed=true;
        }
    }
void MainWindow::keyReleaseEvent(QKeyEvent *event)
{
    if(event->key()==Qt::Key_Control)
    {
        controlKeyIsPressed=false;
    }
}

void MainWindow::on_redColorButton_clicked()
{
    //redColorFlag=true;
    colorOfBorder=QString("red");
    //QWidget::update();
}

void MainWindow::on_blueColorButton_clicked()
{
    //blueColorFlag=true;
    colorOfBorder=QString("blue");
    //QWidget::update();
}

void MainWindow::on_horizontalSlider_valueChanged(int value)
{
    penWidth=value;
}

void MainWindow::on_blackColorButton_clicked()
{
    //blackColorFlag=true;
    //QWidget::update();
    colorOfBorder=QString(QColor(0,0,0).name());
}

void MainWindow::on_greenColorButton_clicked()
{
    //greenColorFlag=true;
    //QWidget::update();
    colorOfBorder=QString(QColor(0,255,0).name());
}

void MainWindow::on_greyColorButton_clicked()
{
    //greyColorFlag=true;
    //QWidget::update();
    colorOfBorder=QString(QColor(169,169,169).name());
}

void MainWindow::on_lightBlueColorButton_clicked()
{
    // lightBlueColorFlag=true;
    //QWidget::update();
    colorOfBorder=QString(QColor(85,255,255).name());
}

void MainWindow::on_orangeColorButton_clicked()
{
    //orangeColorFlag=true;
    //QWidget::update();
    colorOfBorder=QString(QColor(255,170,0).name());
}

void MainWindow::on_violetColorButton_clicked()
{

```



```

        //voiletColorFlag=true;
        //QWidget::update();
        colorOfBorder=QString(QColor(170,0,255).name());
    }

void MainWindow::on_resizePushButton_clicked()
{
    resizingComponent=true;
    i=0;
}

void MainWindow::on_insertImage_clicked()
{
    insertImage=true;
    QString imgFilter="PNG(*.png)";
    QString fileName=QFileDialog::getOpenFileName(this,"Insert
image", "/",imgFilter,&imgFilter,QFileDialog::DontUseNativeDialog);
    if(fileName=="")return;
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(fileName);
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_createLine_clicked()
{
    if(createLineFlag==false)
    {
        createLineFlag=true;
    }
    else
    {
        createLineFlag=false;
    }
    QWidget::update();
}

void MainWindow::on_deleteButton_clicked()
{
    if(select==(-1))return;
    //deleteComponentFlag=true;
    tmView.deleteComponent(select);
    QWidget::update();
    //qDebug()<<row;
    delete (this->ui->listWidget->takeItem(select));
}

void MainWindow::on_clearCanvas_clicked()
{
    clearComponentFlag=true;
    tmView.clear();
    QWidget::update();
    this->ui->listWidget->clear();
}

void MainWindow::on_chemistryBoard_clicked()
{
    this->ui->mainFrame->setVisible(true);
}

void MainWindow::on_cancelButton_clicked()
{
    this->ui->mainFrame->setVisible(false);
}

```

```

}

void MainWindow::on_mole_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/mole1.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_atomicStructure_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/atomic1.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_bodyCentered_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/bodyCentered_solid.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_faceCentered_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/faceCentered_solid.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_periodicTable_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/periodic_table.jpg");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_periodicFormula_clicked()
{

```

```

        TMImageComponent *d=new TMImageComponent;
        d->setX(440);
        d->setY(200);
        d->setWidth(371);
        d->setHeight(341);
        d->setFilePath(":/images/periodic1_imp_formula.jpg");
        tmView.add(d);
        QWidget::update();
        this->ui->listWidget->insertItem(0,d->toString());
    }

void MainWindow::on_methane_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/methane.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_methylAlcohol_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/methylAlcohol.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_formaldehyde_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/formaldehyde.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_methylAmine_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/Methylamine.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_diethylEther_clicked()
{
    TMImageComponent *d=new TMImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);

```

```

        d->setHeight(341);
        d->setFilePath(":/images/diethylEther.png");
        tmView.add(d);
        QWidget::update();
        this->ui->listWidget->insertItem(0,d->toString());
    }

void MainWindow::on_bohrModel_clicked()
{
    TmImageComponent *d=new TmImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/bohr_model.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_hexagonal_clicked()
{
    TmImageComponent *d=new TmImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/hexagonal_closed.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

void MainWindow::on_acetone_clicked()
{
    TmImageComponent *d=new TmImageComponent;
    d->setX(440);
    d->setY(200);
    d->setWidth(371);
    d->setHeight(341);
    d->setFilePath(":/images/acetone.png");
    tmView.add(d);
    QWidget::update();
    this->ui->listWidget->insertItem(0,d->toString());
}

```