

DiaSlate APPLICATION

on

Aakash Tablet

Summer Internship 2013

Submitted in fulfillment of internship project

By

Development on Aakash Platform Team

Under the Guidance of

Prof. (Dr.) D. B. Phatak



**Department of Computer Science and
Engineering,
Indian Institute of Technology, Bombay
Mumbai**

Summer Internship 2013
Project Approval Certificate
Department of Computer Science and Engineering
Indian Institute of Technology Bombay

The project entitled “**DiaSlate Application**” submitted by Mr. Amandeep Singh, Mr. Amit Kumar, Mr. Chandrashekhar Bobade, Mr. Naveen Kumar, Mr. Nishant Kumar, Miss. Vanya Kandlakunta, Mr. Vikram Kore, and Mr. Parlapalli Vinod is approved for Summer Internship 2013 programme from 9th May 2013 to 6th July 2013, at Department of Computer Science and Engineering, IIT Bombay.

Prof. Deepak B. Phatak
Dept of CSE, IITB
Principal Investigator

Mr. Parag Tiwari
Dept of CSE, IITB
Project-In-Charge

Place: IIT Bombay, Mumbai
Date: 2nd July 2013

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/ fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Amandeep Singh

Amit Kumar

Chandrashekhar Bobade

Naveen Kumar

Nishant Kumar

Vanya Kandlakunta

Vikram Kore

Vinod Parlapalli

Acknowledgement

We, the summer interns of Development on Aakash Platform, consider it our privilege to undertake the project of '**DiaSlate**' Android Application. The success and final outcome of this project required the assistance of many people without whom we could not have materialized our vision for the application.

We are deeply indebted to Dr. **D.B.Phatak** for giving us the opportunity to be a part of this esteemed project, for giving us inspiration to pursue the project and guiding us in this endeavor. He has been a constant source of motivation and encouragement for us.

Our profound sense of gratitude is due to our Project Manager **Mr. Parag Tiwari** for constant encouragement and valuable guidance. We thank him for all the initiative and zeal he has filled us with throughout the course of this project.

Furthermore, we heartily thank our mentors **Mr. Ninad Chilap** and **Mr. Arun Nair** with whom the exchange of knowledge and skills helped in making this an enriching experience for us all.

We extend a warm thank you to **Mr. Dilip Sable** and **Mr. Rahul Kharat** for making our stay here as summer interns comfortable and for all their administrative help. Finally, we would also like to thank the support staff at the institute who have directly or indirectly helped us in the completion of the project.

Team Members:

1. Amandeep Singh

2. Amit Kumar

3. Chandrashekhar Bobade

4. Naveen Kumar

5. Nishant Kumar

6. Vanya Kandlakunta

7. Vikram Kore

8. Vinod Parlapalli

Abstract

The **DiaSlate** application true to its name is developed as a digital equivalent of a real time slate. It is an amalgamation of UML **Diagrams** and **Slate**. **DiaSlate** is an Android Application that implements an intelligent canvas interface that aids learning, work and fun. It has 2 parts, one emulating the slate and the second a user-friendly way of drawing UML diagrams. It aims to replace the handheld slate and chalk. The application caters to all age groups.

The SLATE:

Apart from the standard drawing and erasing, the user can make use of the additional functionalities provided. Features include changing the color, size of the brush, changing the color of the background, addition of dynamic pages et al. It also gives the user freedom to save and share what has been drawn. Drawings can be saved as an image (PNG format) or a PDF based on requirement.

The Unified Modeling Language Diagrams:

This enables the user to draw UML diagrams with ease. Its functionalities include explicit buttons for each shape (rectangle, line, circle etc.). In addition, they can be saved as images which can be edited, implemented and shared. This part of the application is still a prototype and requires further enhancement for full-fledged functioning.

The **DiaSlate** application is developed to make the classrooms more interactive and learning fun. Targeted users contain people of all age groups who wish to learn, work or just have fun. It is an open source application designed with a single purpose – education in a low resource environment.

Contents

1. Introduction.....	1
1.a. Purpose.....	1
1.b. Scope.....	1
2. Software Requirement Specification.....	2
2.1 Introduction.....	3
2.2 Overall Description.....	5
2.2.a. Product Perspective.....	5
2.2.b. Product Functions.....	5
2.2.c. User Characteristics.....	5
2.2.d. Constraints.....	5
2.2.e. Assumptions & Dependencies.....	6
3. Software Design Description.....	7
3.1 External Interface Requirements.....	8
3.1.a User Interfaces.....	8
3.1.b Hardware Interfaces.....	8
3.1.c Software Interfaces.....	9
3.2 Functional Requirements.....	9
3.2.1.Slate.....	9
3.2.1.1 Brush/Pencil.....	9
3.2.1.2 Eraser.....	9
3.2.1.3 Save.....	9
3.2.1.4 Add page.....	9
3.2.1.5 Delete Page.....	9
3.2.1.6 Background Colour.....	10
3.2.1.7 Full Screen.....	10
3.2.1.8 Refresh.....	10
3.2.2 Uml.....	10
3.2.2.1 Shapes.....	10
3.2.2.2 Move.....	10
3.2.2.3 Text.....	10
3.2.2.4 Delete.....	10
3.2.2.5 Background Colour.....	10
3.2.2.6 Save.....	11

3.2.2.7 Edit.....	11
3.2.2.8 View.....	11
3.3 Non Functional Requirements.....	11
3.3.1 Performance Requirements.....	11
3.3.1.1 Maintainability.....	11
3.3.1.2 Reliability.....	11
3.3.2 Safety Requirements.....	11
3.3.3 Security Requirements.....	11
3.3.4 Software Quality Attributes.....	12
3.4 Behavioral Requirements.....	14
4. User Manual	30
4.1. SLATE.....	33
4.2. UML.....	52
5. Summary and Conclusion.....	65
6. References.....	66

List of Figures

3.1 Use Case Diagram for DiaSlate Application.....	13
3.2 Use Case For Slate.....	14
3.3 Use Case For Uml.....	15
3.4 Class Diagram For Slate.....	16
3.5 Class Diagram For Uml.....	17
3.6 Data Flow Diagram For DiaSlate Application.....	18
3.7 Data Flow Diagram For Slate.....	19
3.8 Data Flow Diagram For Uml.....	20
3.9 Sequence Diagrams:	21
3.9.1 Drawing in Slate	21
3.9.2 Addition and Deletion of pages in slate.....	22
3.9.3 Drawing Shapes in Uml.....	23
3.9.4 Addition of text in Uml.....	24
3.9.5 Viewing existing image in Uml.....	25
3.9.6 Editing existing image in Uml.....	26
3.9.7 Changing Background in Uml.....	27
3.10 Activity diagram For Slate.....	28
3.11 Activity diagram For Uml.....	29
4.1 Screenshot of Main Page.	32
4.2 Screenshot of Components of Slate.....	33
4.3 Screenshot of Pencil/Brush.....	34
4.3.1 Demonstration of Pencil/Brush.....	34
4.3.2 Drawing with default Pencil/Brush width.....	34
4.3.3 Long Press of Pencil/Brush.....	35
4.3.4 Changing Width of Pencil/Brush.....	35
4.3.5. Drawing with new Pencil/Brush width.....	35
4.4 Screenshots of Eraser.....	36
4.4.1 Writing Letters.....	36
4.4.2 Erasing Using Eraser.....	36
4.4.3 Rewriting Letter 'A'.....	37
4.4.4 Black portion of Canvas to be arranged.....	37
4.4.5 Erasing with Default Width.....	37
4.4.6 Long Press of Eraser.....	38
4.4.7 Changing width of eraser.....	38
4.4.8 Erasing with new set eraser width.....	38

4.5 Screenshot of Canvas.....	39
4.6 Screenshots of pencil/brush colour.....	39
4.6.1 Selecting colour from colour Picker.....	39
4.6.2 Drawing with selected colour.....	40
4.7 Screenshots of background colour.....	40
4.7.1 Selecting colour using color picker.....	40
4.7.2 Selected colour applied to background.....	41
4.8 Screenshots of full screen Button.....	41
4.8.1 Drawing in Default Mode.....	41
4.8.2 Changing to full screen mode.....	42
4.8.3 Drawing in full screen mode.....	42
4.9 Screenshots of Refresh.....	43
4.9.1 Drawing on the Canvas.....	43
4.9.2 Refreshing canvas.....	43
4.10 Screenshots of adding pages.....	44
4.10.1 Drawing on single page.....	44
4.10.2 Adding a page.....	44
4.10.3 Adding another page.....	45
4.11 Screenshots of deleting pages:	45
4.11.1 Deleting a page.....	45
4.11.2 Pages that remain after deletion.....	46
4.12 Screenshot of demonstration of previous page.....	46
4.13 Screenshot of demonstration of next page.....	47
4.13.1 Clicking on next page.....	47
4.13.2 Displaying next page.....	47
4.14 Screenshot of page counter.....	48
4.15 Screenshot of page Scrollbar.....	48
4.16 Screenshots for demonstration of save.....	49
4.16.1 Save menu.....	49
4.16.2 Naming the file.....	49
4.16.3 Typing name for the file.....	50
4.16.4 Saving 2 pages.....	50
4.16.5 Naming the pages to be saved.....	51
4.16.6 Images Saved.....	51
4.17 Screenshot of a UML main page.....	52
4.18 Screenshot of a new canvas.....	53
4.19 Screenshot of list of images for editing.....	54
4.20 Screenshot of list of images for viewing.....	54
4.21 Screenshot for save.....	55
4.22 Screenshot of drawing shapes.....	55

4.22.1 Shapes menu.....	56
4.22.2 Drawing a Rectangle.....	56
4.22.3 Drawing a Circle.....	57
4.22.4 Drawing an Oval.....	57
4.22.5 Drawing a line.....	58
4.22.6 Drawing an Actor.....	59
4.23 Screenshots for move.....	59
4.23.1 Initial Position of shape.....	59
4.23.2 Final Position of shape.....	60
4.24 Screenshots of Text.....	60
4.24.1 Drawing a shape/entity.....	60
4.24.2 Selecting entity to be named.....	61
4.24.3 Entering name for entity.....	61
4.24.4 Labeled Entity.....	62
4.25 Screenshots for Delete.....	62
4.25.1 Selecting shape for Deletion.....	62
4.25.2 Canvas after deletion of shape.....	63
4.26 Screenshots for background colour.....	63
4.26.1 Selecting a colour from background colour menu.....	63
4.26.2 Canvas with new colour selected.....	64

1. Introduction:

1. a. Purpose:

The **DiaSlate** application is an Android Application that implements an intelligent canvas interface that aids learning of alphabets etc. It aims at not only learning through drawing but also for fun. The UML Diagram part of the application is aimed at creating a user-friendly interface for drawing UML diagrams. It is well endowed with features for enabling the same. Catering to all age groups, this application is open source and preferably used on tablets.

1. b. Scope:

The dual feature of this application enables it to be used by children and adults alike. Users can implement both UML diagrams and text to give an elaborate description of any topic. In addition to ease of use, it is eco-friendly and timesaving. This application is unique owing to its dual feature property. It not only gives comfort but also saves paper.

Software Requirement Specification

2.1 Introduction:

The document aims at defining the overall software requirements for DiaSlate Application. Efforts have been made to define the requirements exhaustively and accurately. The final product contains only those features/functionalities mentioned in this document.

Assumption for any additional functionality/feature should not be made by any of the parties involved in developing /testing/implementing the product. In case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/or product will be produced.

Document Purpose

The purpose of this document is to present a detailed description of the DiaSlate Application. It will explain the purpose and features of the application and what the application will do. This document is intended for developers, testers and users.

Product Scope

“DiaSlate Application” is an Open source software product. The main goal of this Android application is to learn, work and enjoy. DiaSlate Application is going to be freely available for anyone to download. It is intended to be used on mobiles and tablets that run on Android only.

Intended Audience

Our application’s target audience includes:

- Children
- School Teachers
- Professors in Universities
- Anybody who enjoys drawing

Glossary:

Term	Definition
Android	Linux based operating system
Android Canvas	A drawing surface that handles compositing of the actual bits against a Bitmap or Surface object.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Unified Modelling Language(UML)	Programming Language used for object oriented software development
UML Diagram	It is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence.
Android 2D	The Android framework APIs provides a set 2D drawing APIs that allow you to render your own custom graphics onto a canvas or to modify existing Views to customize their look and feel

References:

- developer.android.com
- IEEE STD 830-1998. (Revision of. IEEE STD 830-1993). IEEE Recommended Practice for Software Requirements.

2.2. Overall Description:

2.2. a. Product Perspective:

This product is developed on the lines of pre-existing applications (not open source) that provide the functionality of drawing. Furthermore, the feature of drawing UML diagrams has been inculcated.

2.2. b. Product Functions:

The **DiaSlate** contains both SLATE and UML. However, they cannot be used concurrently. The application output is primarily for the Aakash tablet but can be downloaded onto any Android device.

2.2. c. User Characteristics:

The intended users will be children (ages 5-15) and Professors. Users are not expected to have a very high level of technical expertise to use the application .Although; prior knowledge about the UML Diagrams will prove to be beneficent.

2.2. d. Constraints:

The user will not be able to run both parts of the application simultaneously. The product is developed only for devices that support Android and cannot be used on any other platform.

2.2. e. Assumptions and Dependencies:

There are several assumptions at the time of this document's drafting that are still being explored. We have assumed that user interaction with the application will only be through touch. The application will be available only to devices that run on Android operating system.

Software Design **Description**

3. Software Design Description:

3.1. External Interface Requirements:

The application is intended to be a stand-alone, single-user system. The application will run on any Android mobile device or an Android emulator. No further hardware devices or interfaces will be required.

3. 1.a. User Interfaces:

- Inputs:
The user select from number of available options.
- Outputs:
The application responds to the input by displaying the request in the screen.
- Operating Systems :
Android
- There will be a 'front page' where three buttons will be displayed.
On selecting:
 - Slate Button:** SLATE application will be launched.
 - UML Button:** UML application will be launched.
 - Exit Button:** Application is closed.
 - Information Button:** Information about the application will be displayed.

3.1. b. Hardware Interfaces:

- Any android operating system supported device.
- micro SD-Card to view the lecture video

3.1. c. Software Interfaces:

- Eclipse IDE shall be used as development environment for implementing the modules
- Designing of modules and diagrams is done in UML using MS word and Rational Rose.

3.2. Functional requirements

3.2.1 Slate

3.2.2 UML

Slate

3.2.1.1. Brush/Pencil:

The user can choose the color and width of the brush and draw, paint or write with flexibility over screen usage.

3.2.1.2. Eraser:

The user can remove what is not required using the eraser. Further, the size of the eraser's width can be adjusted as per the user's whim.

3.2.1.3. Save:

The user can save the drawing as either an image (PNG format) or a PDF. Furthermore, in case of multiple pages, the user can save all of them at the same time.

3.2.1.4. Add page:

The application allows the user to add pages and draw on them simultaneously.

3.2.1.5. Delete page:

The user can delete any added pages in case it needs to be discarded.

3.2.1.6. Background colour:

The colour of the canvas can be changed using this option. A colour picker is launched from which the user can select the desired colour. In case of multiple pages, this colour is applied to all the pages present.

3.2.1.7. Full Screen:

The application can be changed to full screen mode for increasing the comfort of drawing.

3.2.1.8. Refresh:

Using this button the user can clear the entire screen and start over.

UML

3.2.2.1. Shapes:

The user can choose from a number of shapes like Circle, Rectangle, Ellipse, Line and Actor to be used for drawing the UML diagrams.

3.2.2.2. Move:

The shape can be dragged and dropped to any part of the screen.

3.2.2.3. Text:

The user can write text inside the shapes/entities for the UML diagrams.

3.2.2.4. Delete:

The user can delete any shape that is not required.

3.2.2.5. Background colour:

The colour of the background can be changed.

3.2.2.6. Save:

The diagram can be saved as an image (PNG format).

3.2.2.7. Edit:

Old diagrams saved as images can be edited.

3.2.2.8. View:

Old images saved can be viewed.

3.3 Non-functional requirements

3.3.1 Performance Requirements:

The application will need to provide a responsive interface. There should be minimal delay in retrieving any content or associated detail.

3.3.1.1. Maintainability: Software needs to be upgraded if required in future.

3.3.1.2. Reliability: System must be reliable and data should persist even after suffering some system crashes or booting of some Android devices

3.3.2 Safety Requirements

There is no safety requirements associated with this application.

3.3.3 Security Requirements

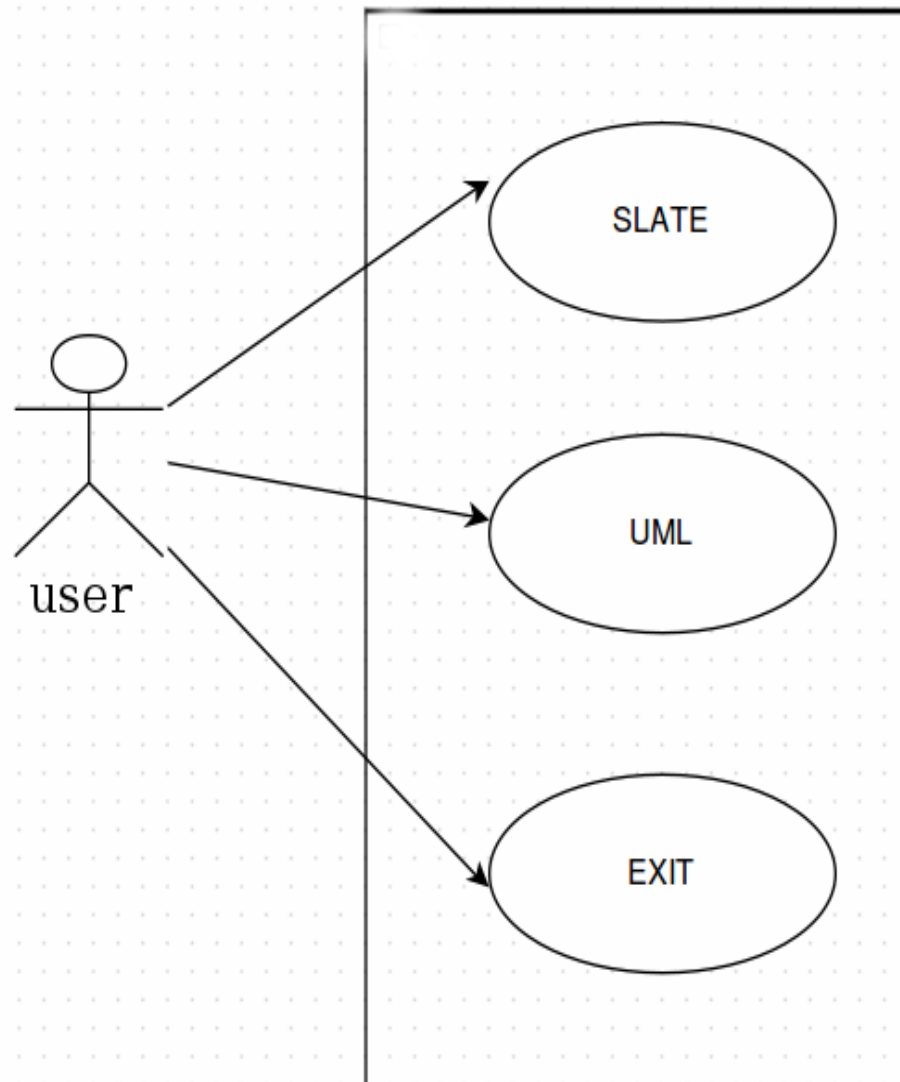
There is no security requirements associated with this application.

3.3.4 Software Quality Attributes

The software is planned to be robust and complete, in order to attract new users, while also providing a usable interface that is clutter free and easy to use.

3.4 Behavioral Requirements

Use Case Diagrams



Use case diagram for Application

Fig 3.1. Use case diagram for DiaSlate Application

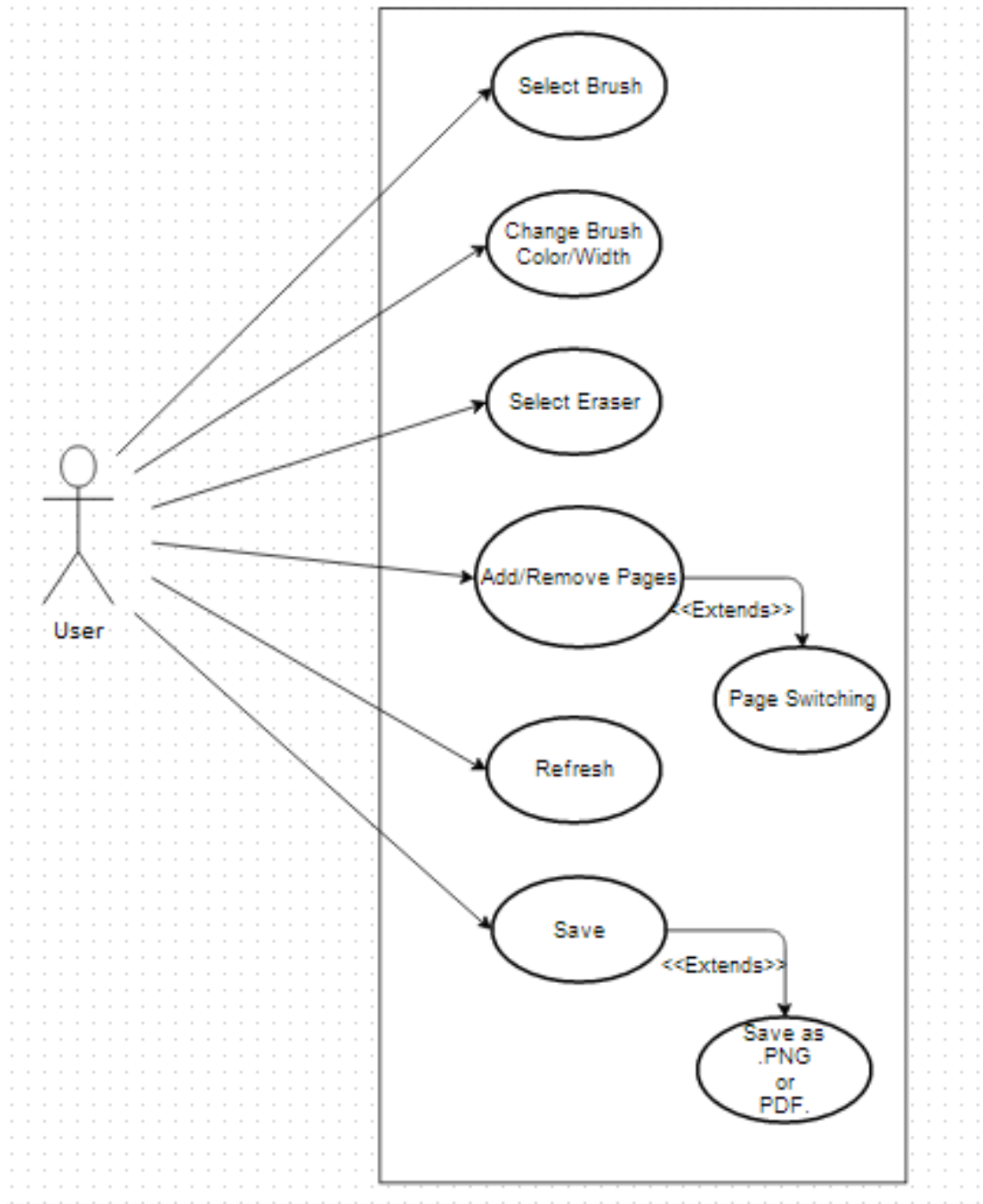


Fig 3.2. Use Case for SLATE

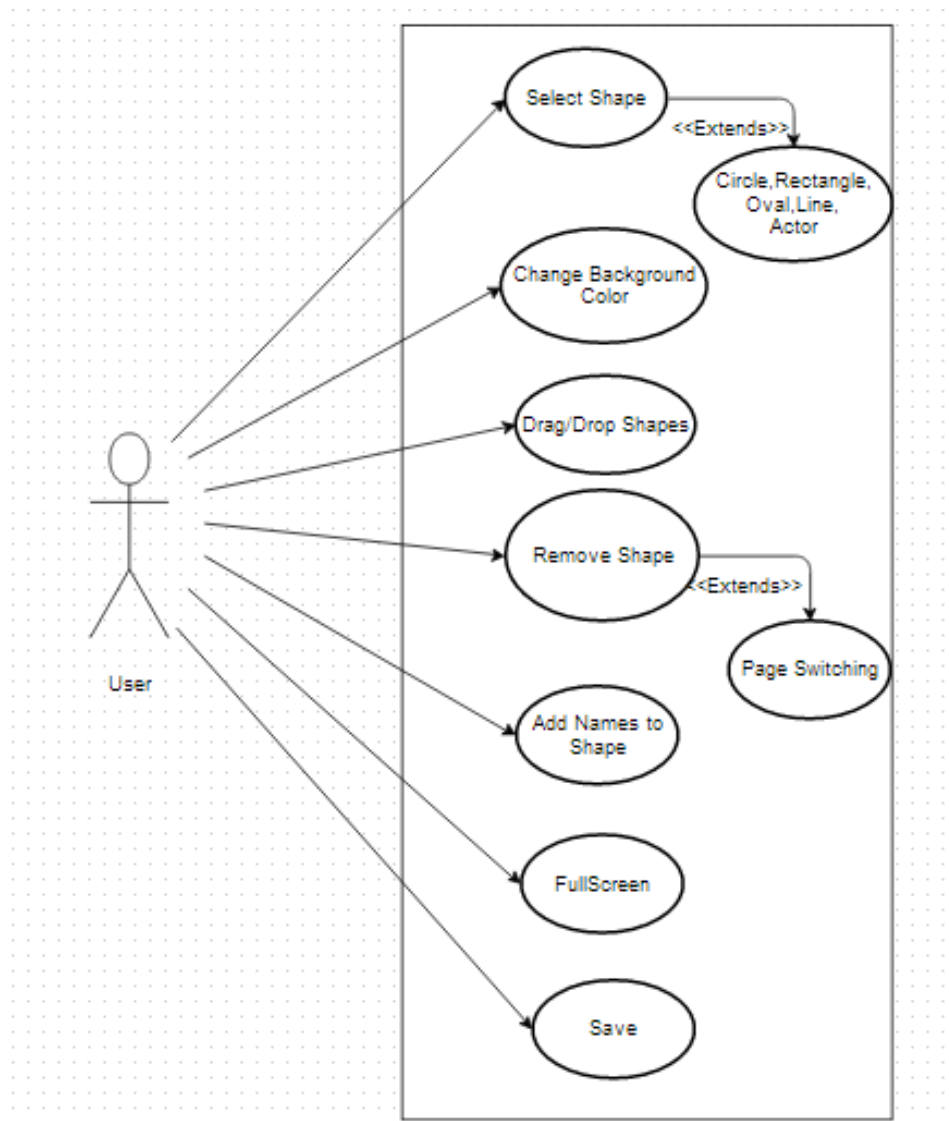


Fig 3.3. Use Case Diagram for UML

Class Diagrams

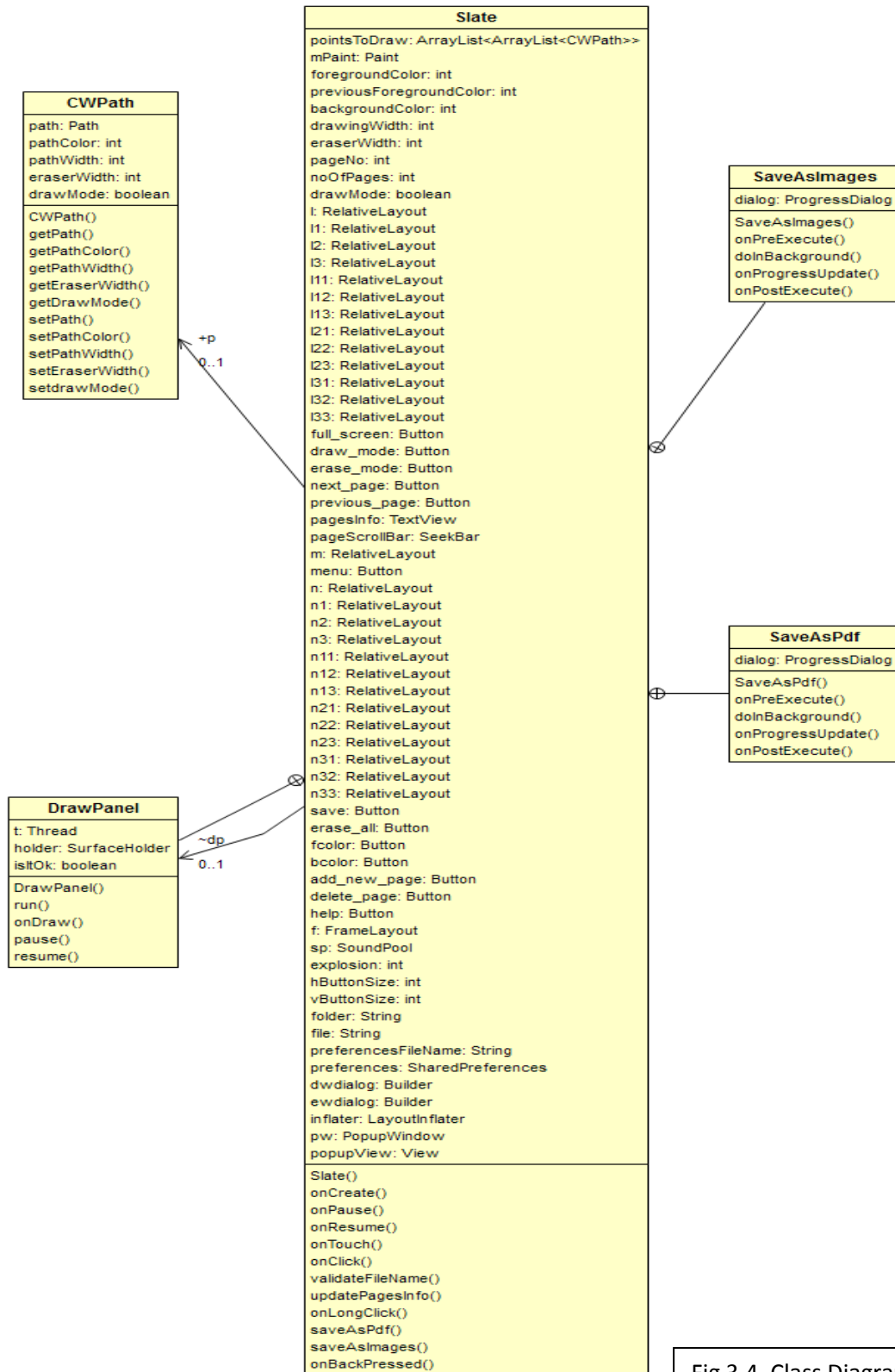


Fig 3.4. Class Diagram for Slate

DATA FLOW DIAGRAMS

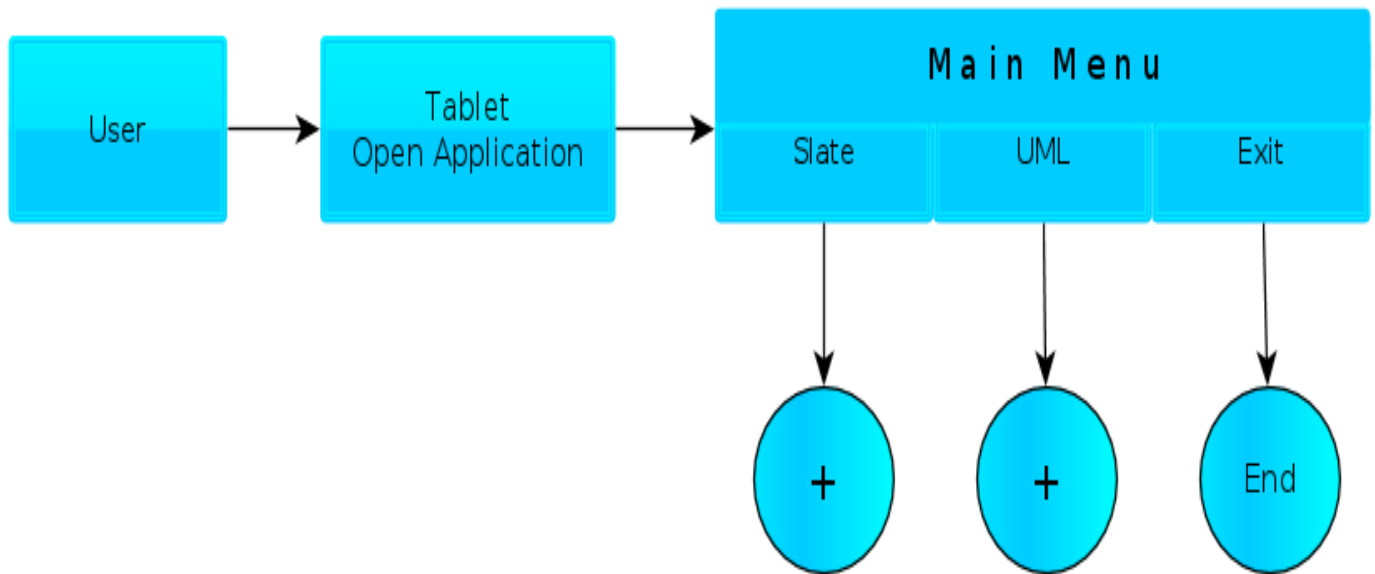


Fig 3.6. Data Flow Diagram for DiaSlate Application

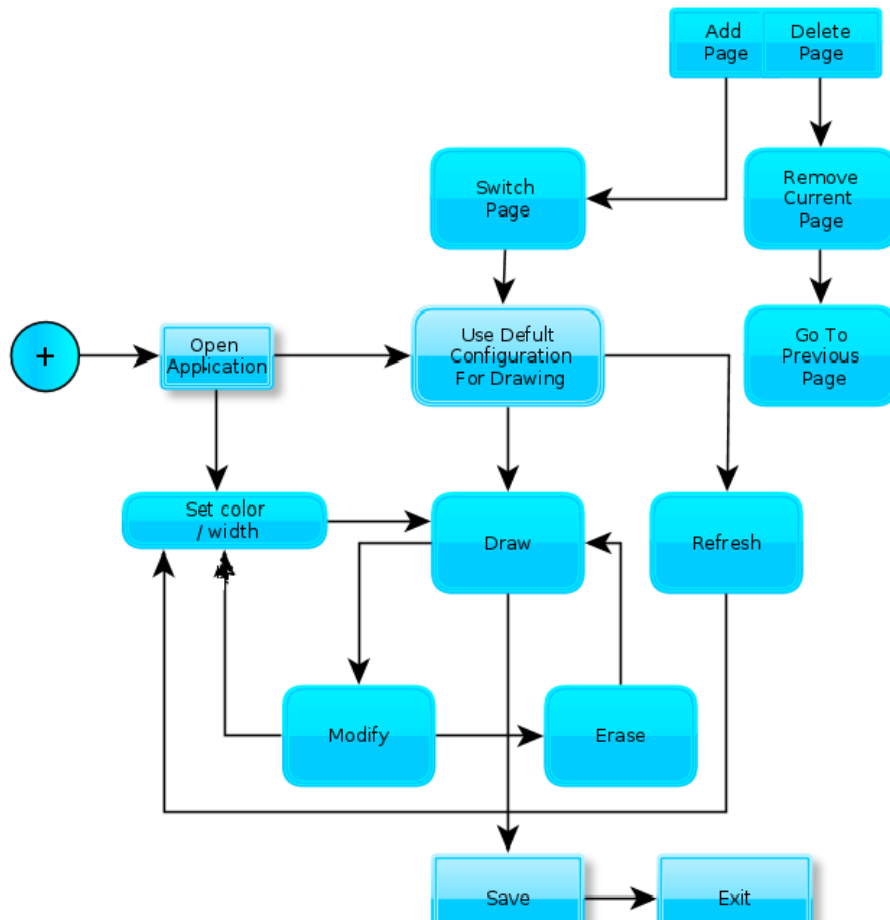


Fig 3.7. Data Flow Diagram for Slate

Sequence diagram for Slate:

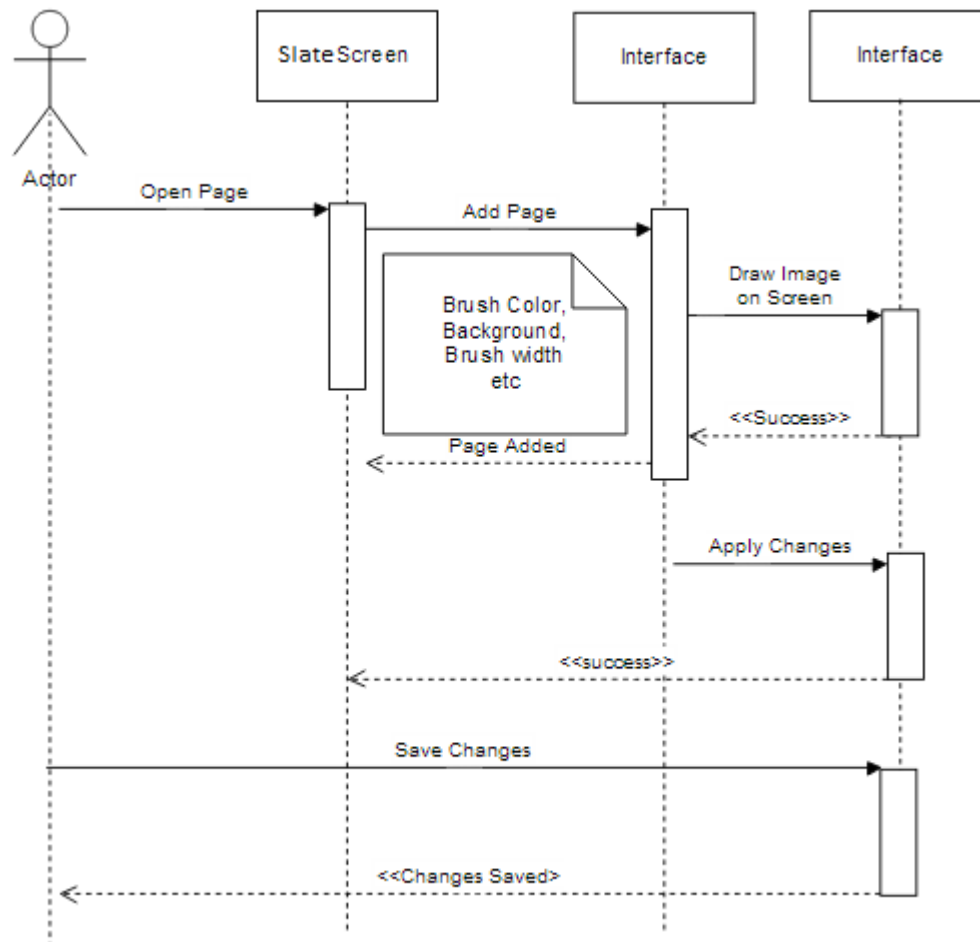


Fig 3.9.1 Sequence Diagram for drawing in Slate

1.B.Slate Add/Delete:

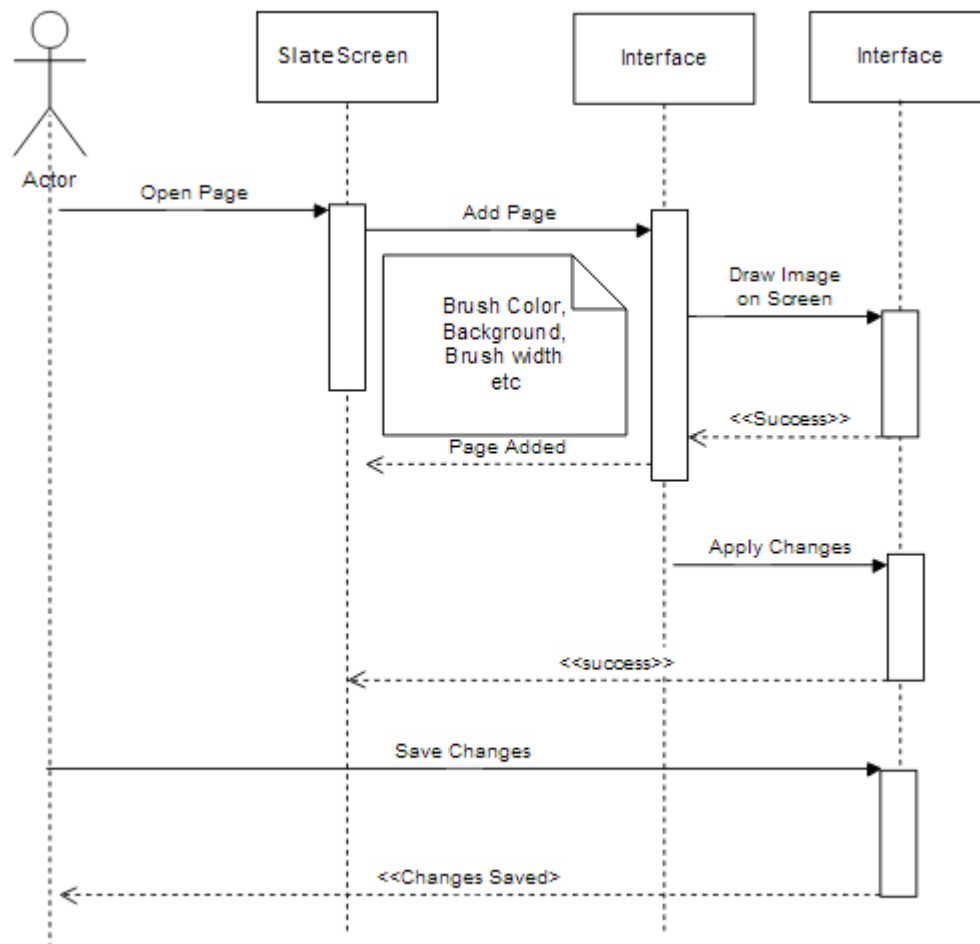


Fig 3.9.2 Sequence Diagram for drawing in Slate with addition and deletion of pages

2.A. UML Adding Shape:

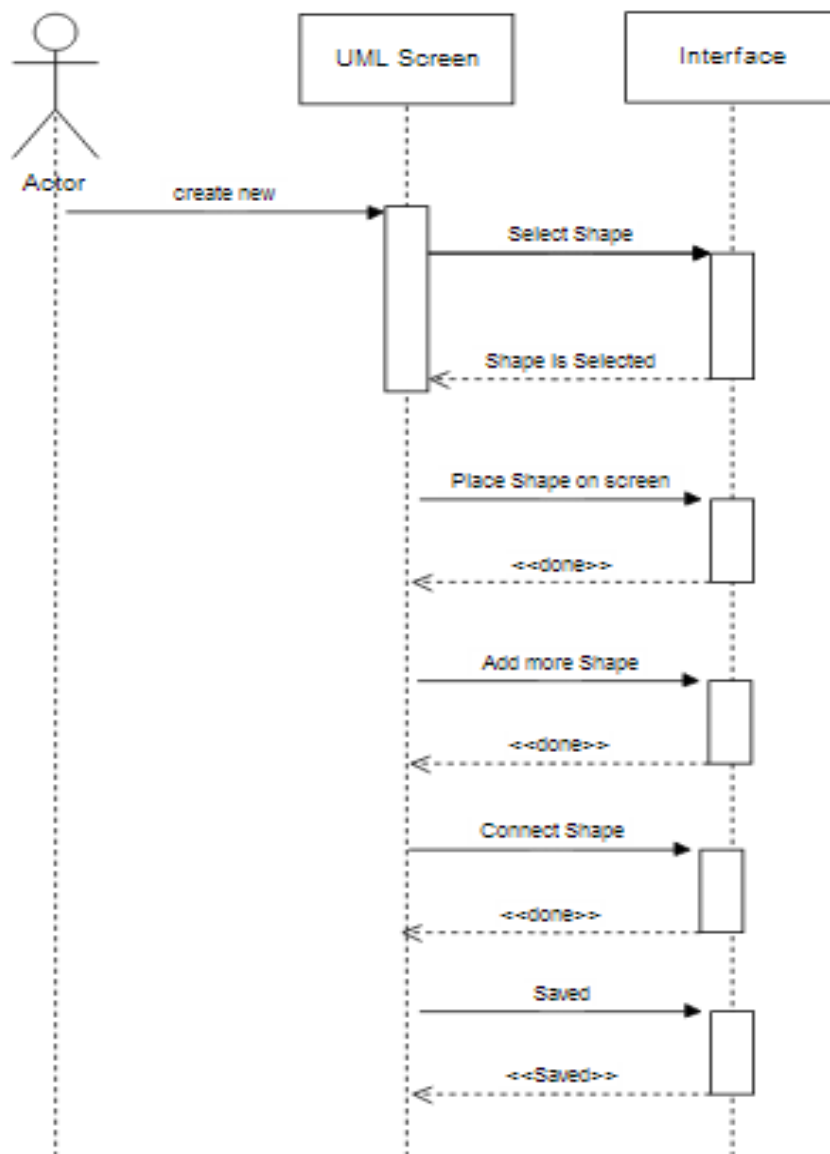


Fig 3.9.3. Sequence Diagram for drawing shapes in UML

2.B. UML Add text/Name:

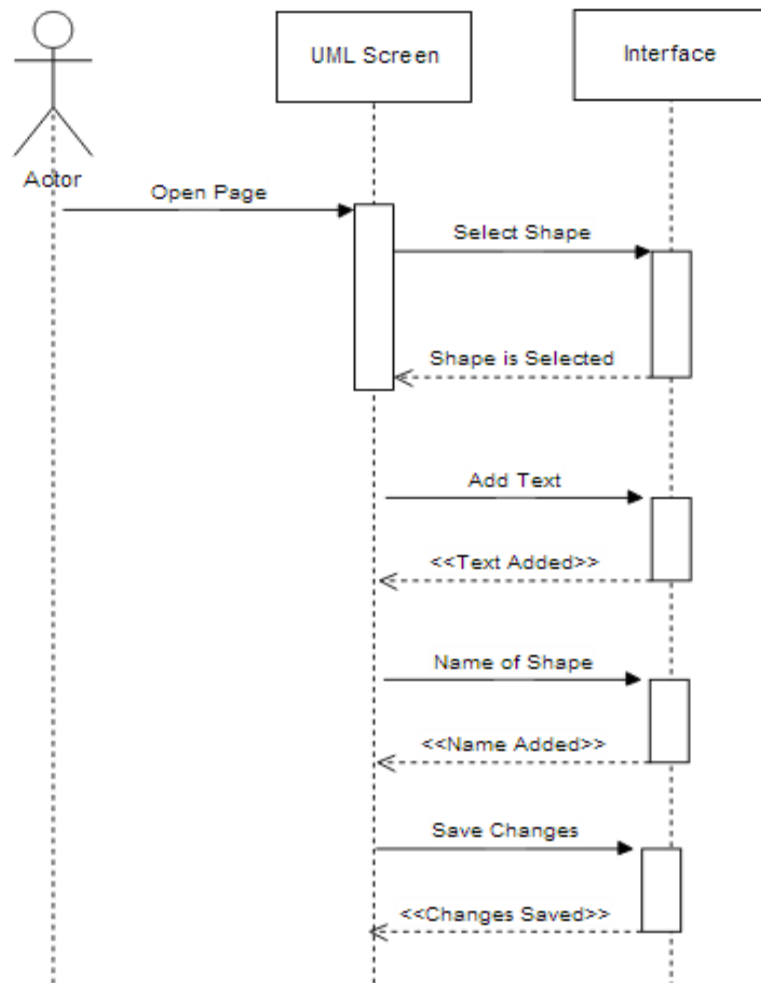


Fig 3.9.4. Sequence Diagram for adding text in UML

2.C.UML View Existing:

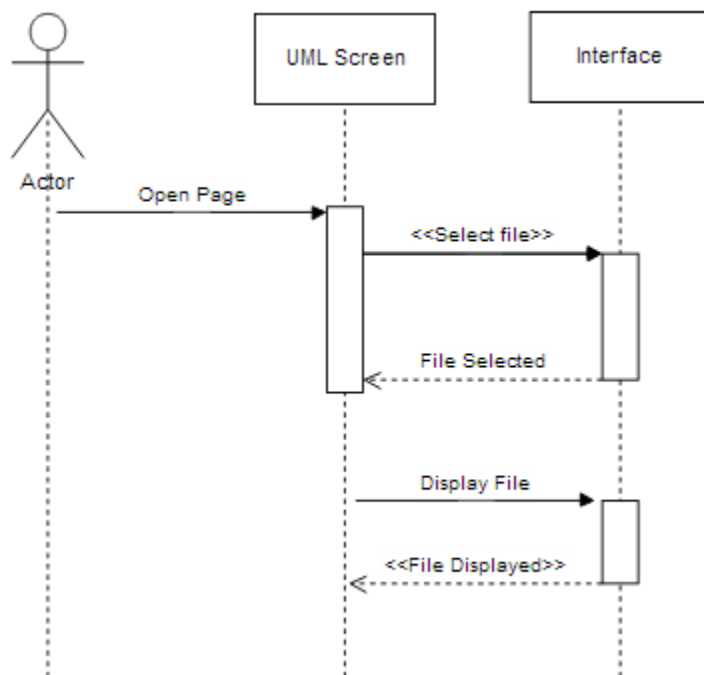


Fig 3.9.5. Sequence Diagram for viewing existing image in UML

2.D.UML Edit Existing:

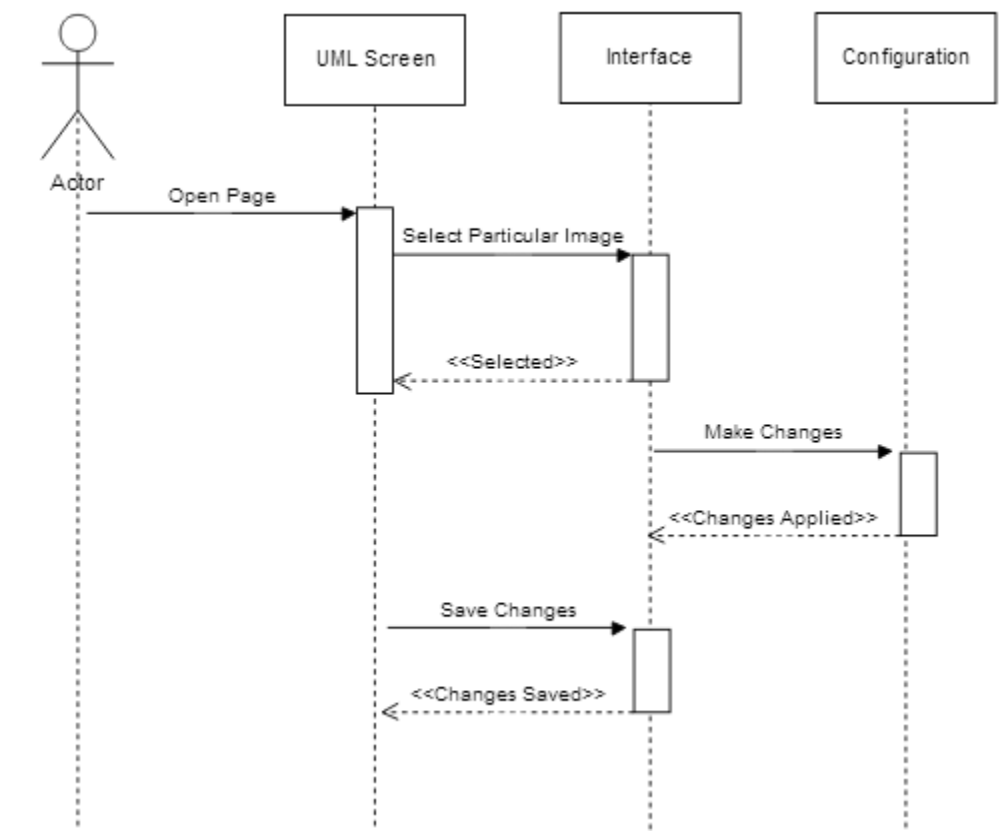


Fig 3.9.6. Sequence Diagram for editing existing image in UML

2.E.UML Change Background Color:

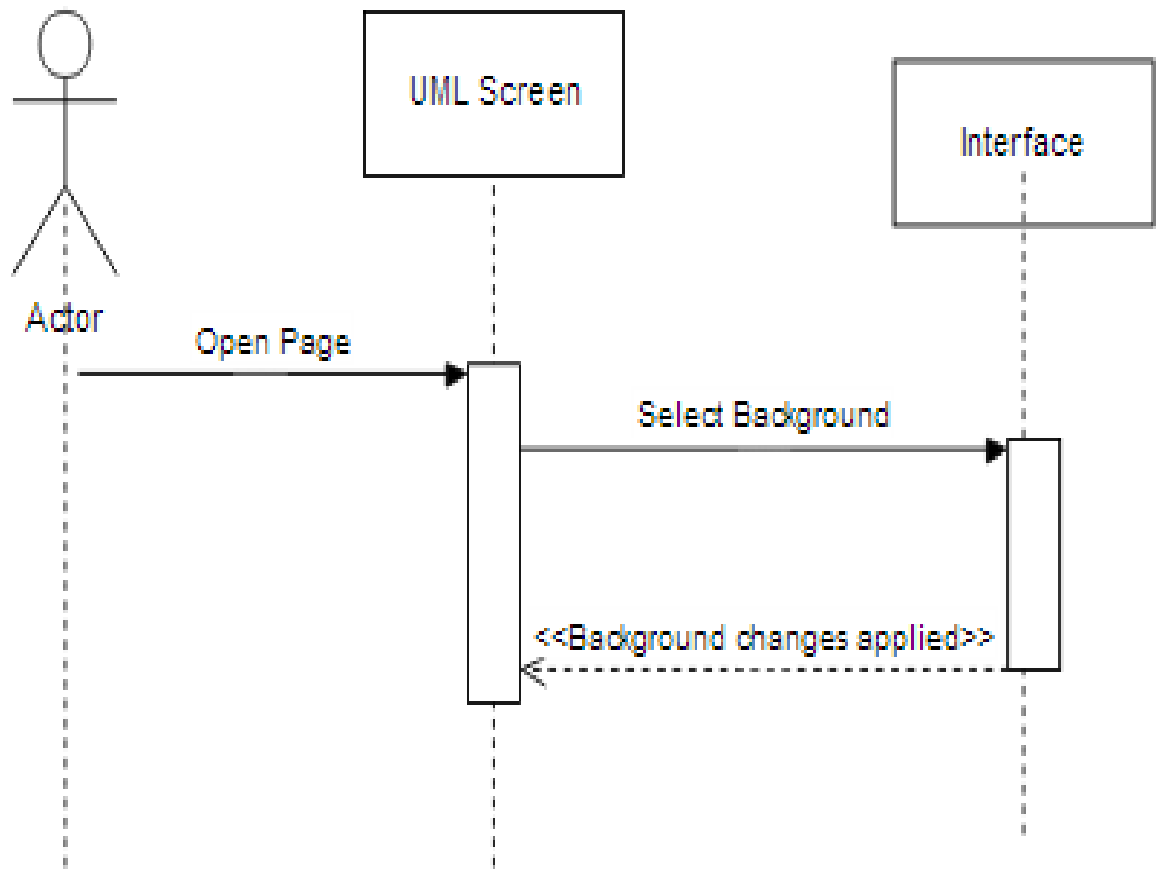


Fig 3.9.7. Sequence Diagram for changing background colour in UML

Activity Diagram for Slate:

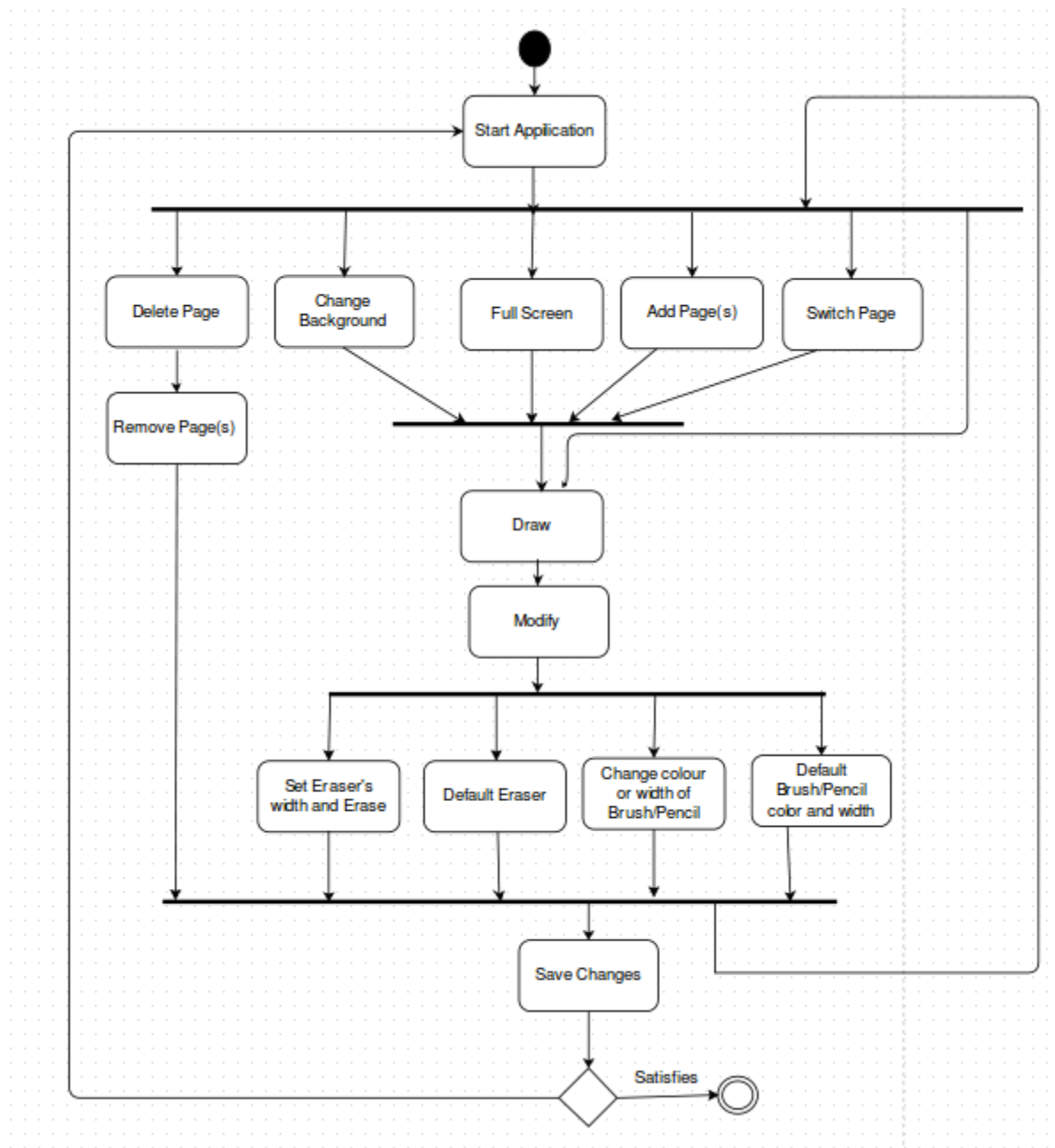


Fig 3.10. Activity Diagram for SLATE

Activity Diagram for UML:

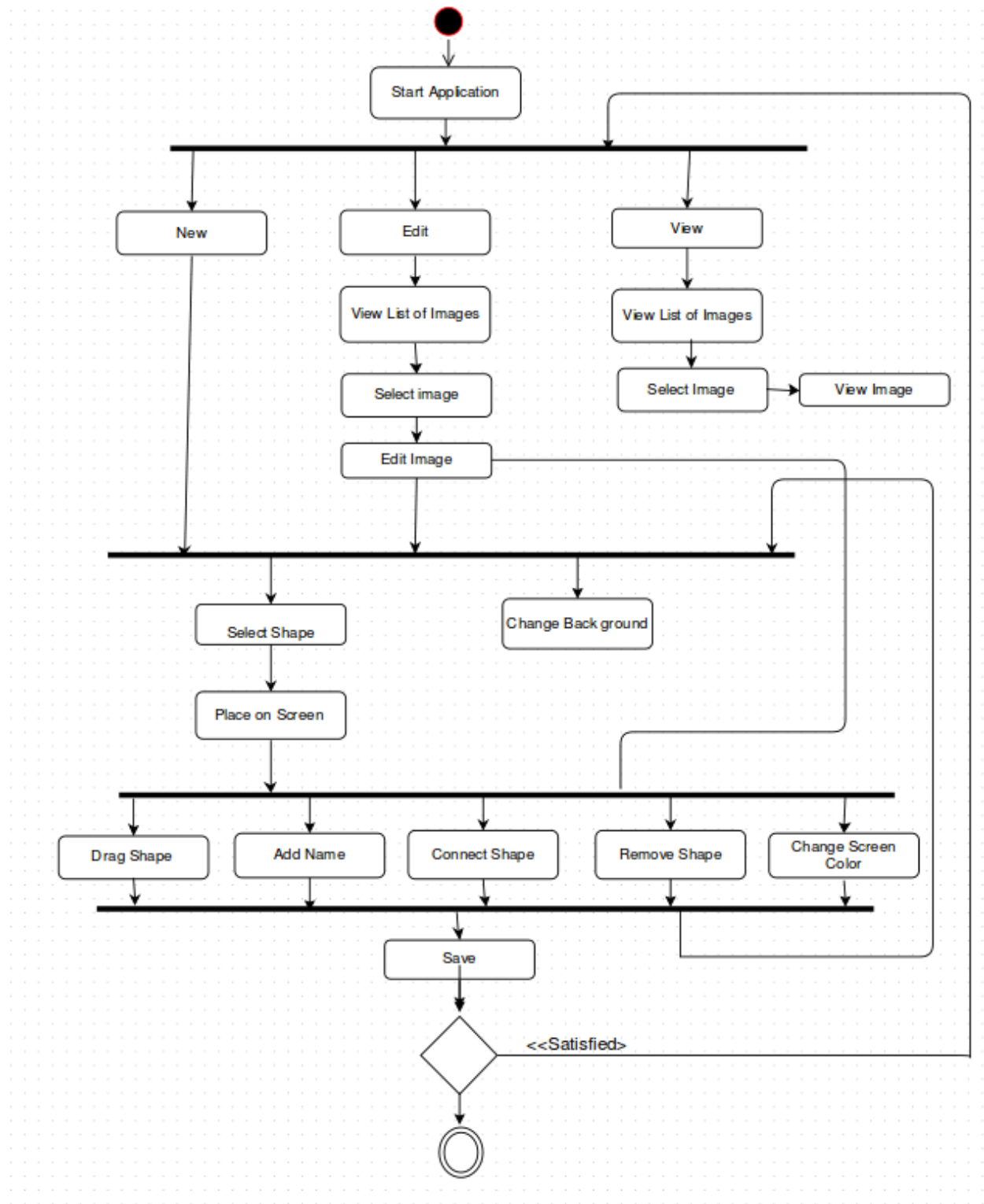


Fig 3.11. Activity Diagram for UML

User Manual

4. User Manual for DiaSlate Application:

Introduction:

The **DiaSlate** application true to its name is developed as a digital equivalent of a real time slate. **DiaSlate** is an Android Application that implements an intelligent canvas interface that aids learning of alphabets etc. It has 2 parts, one emulating the slate and the second a user-friendly way of drawing UML diagrams. It aims to replace the handheld slate and chalk. The application caters to all age groups.

The SLATE:

Apart from the standard drawing and erasing, the user can make use of the additional functionalities provided. Features include changing the color, size of the brush, changing the color of the background, adding text, addition of dynamic pages et al. It also gives the user freedom to save and share what has been drawn.

The Unified Modeling Language Diagrams:

This enables the user to draw UML diagrams with ease. Its functionalities include explicit buttons for each shape (rectangle, line, circle etc.). In addition, they can be saved as images which can be edited, implemented and shared.

The **DiaSlate** application is developed to make the classrooms more interactive and learning fun. Furthermore, the children can take this knowledge home and share it with their parents, educating them in the process. It aims at empowering every household in rural India with education through the Aakash Tablet.

It is an open source application designed with a single purpose – education in a low resource environment.

This manual will guide you through the usage of the application in two parts:

1. Slate
2. UML



Main Page

Fig 4.1

SLATE:

The components of the slate are:

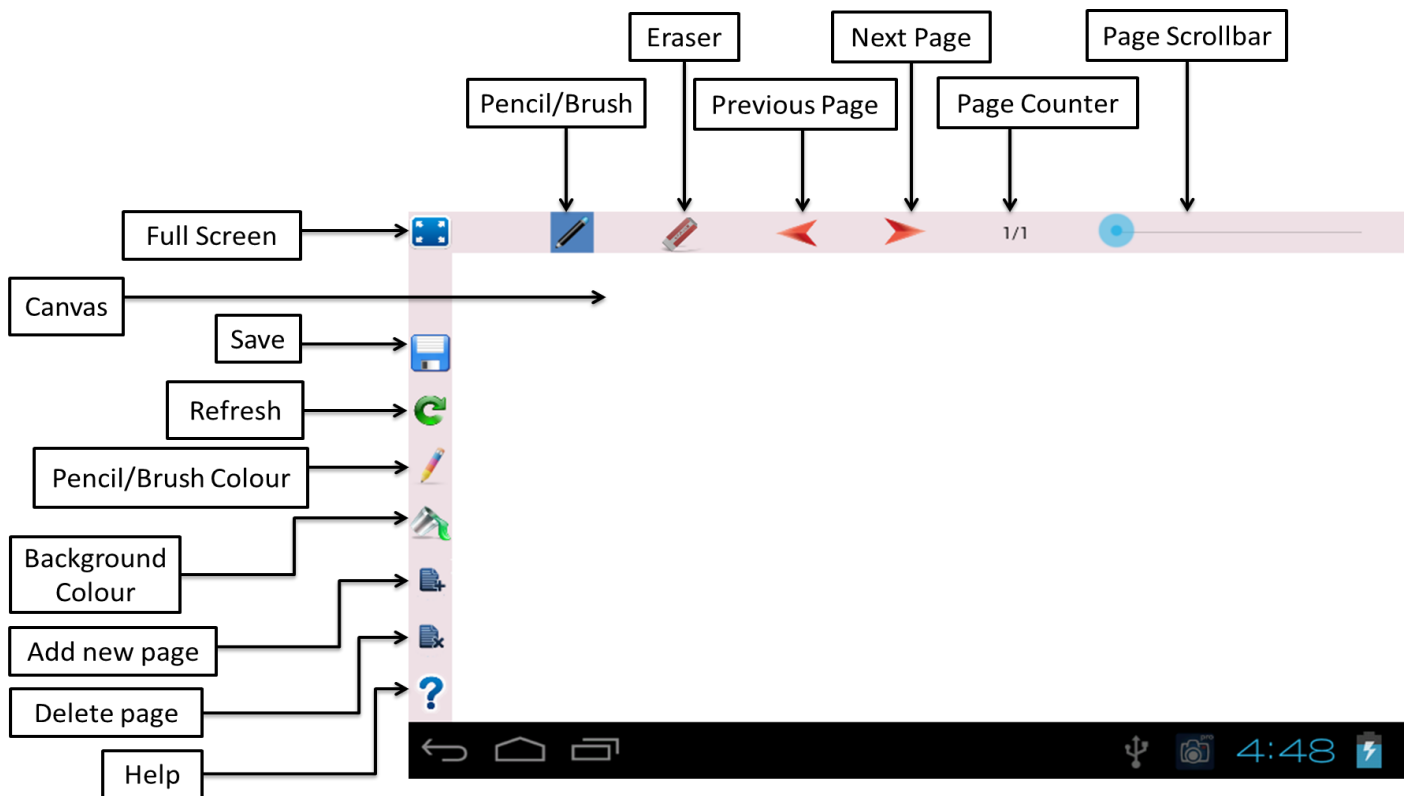


Fig 4.2

1. Brush/Pencil
2. Eraser
3. Canvas
4. Pencil/Brush Colour
5. Background Colour
6. Full Screen
7. Refresh
8. Add new page
9. Delete page
10. Previous Page
11. Next Page
12. Page Counter
13. Page Scrollbar
14. Save
15. Help

1. Brush/ Pencil:

This is used to write/draw freehand on the entire canvas. When the user clicks on this icon, the application is in drawing mode. This enables the user to unleash his creativity and draw. We can also change the width of the brush/pencil. On long press of the button, a popup menu appears containing a scrollbar. Using this, the user can adjust the width as required.

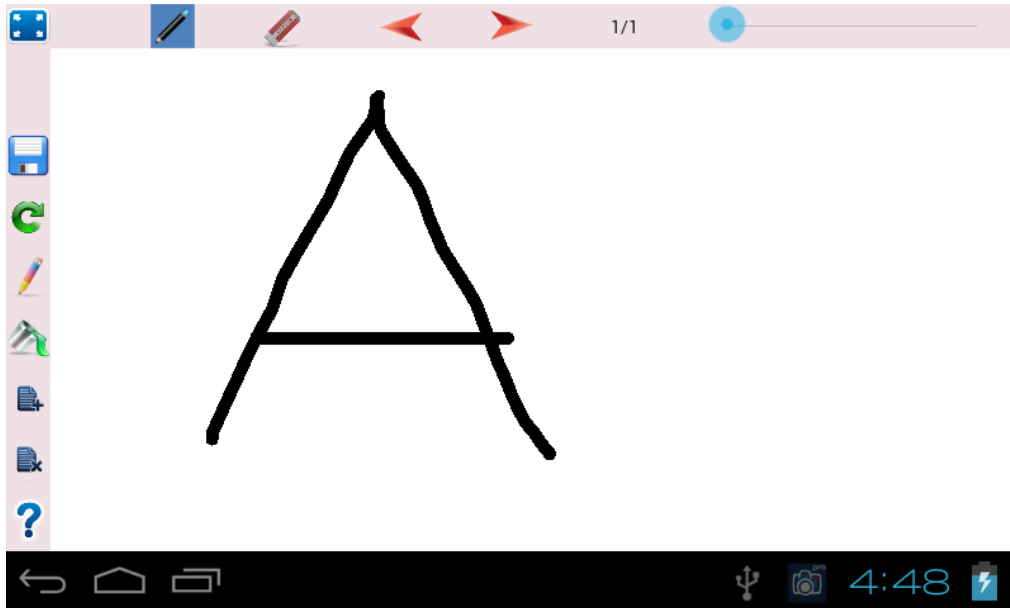


Fig 4.3.1. The user draws letter 'A'

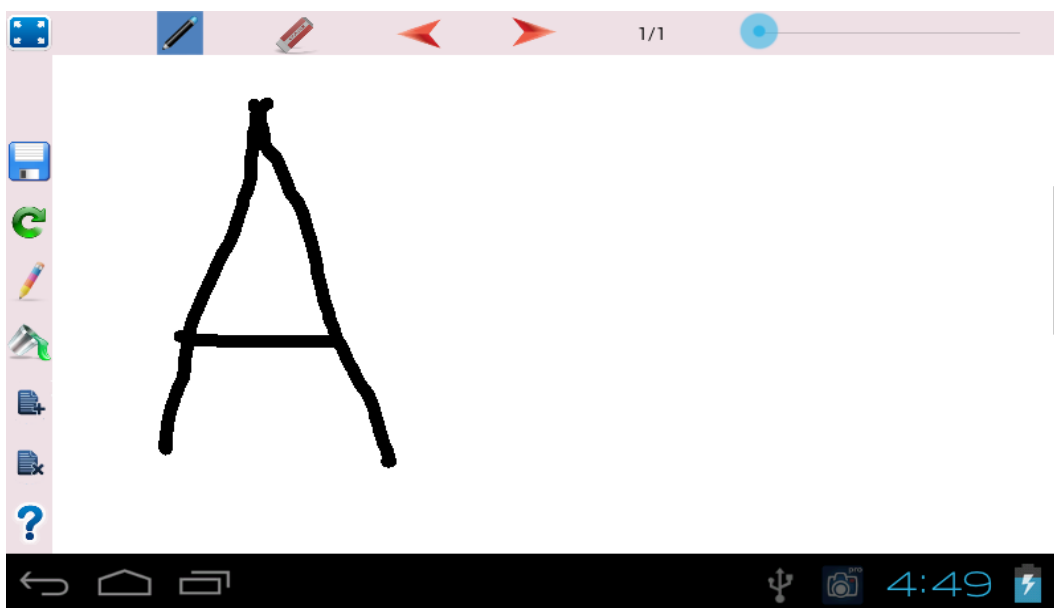


Fig 4.3.2. The user draws an 'A' with default

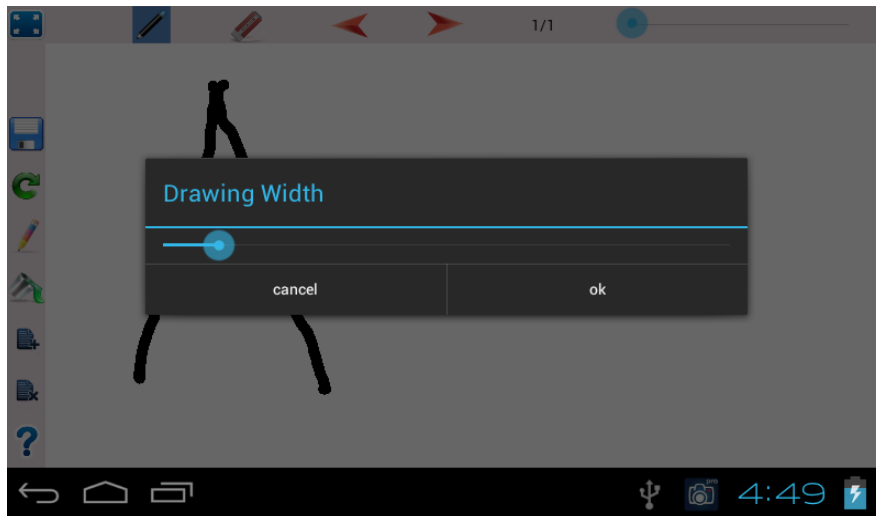


Fig 4.3.3. On long press of the pencil/brush icon, a popup menu appears indicating the drawing width being used.

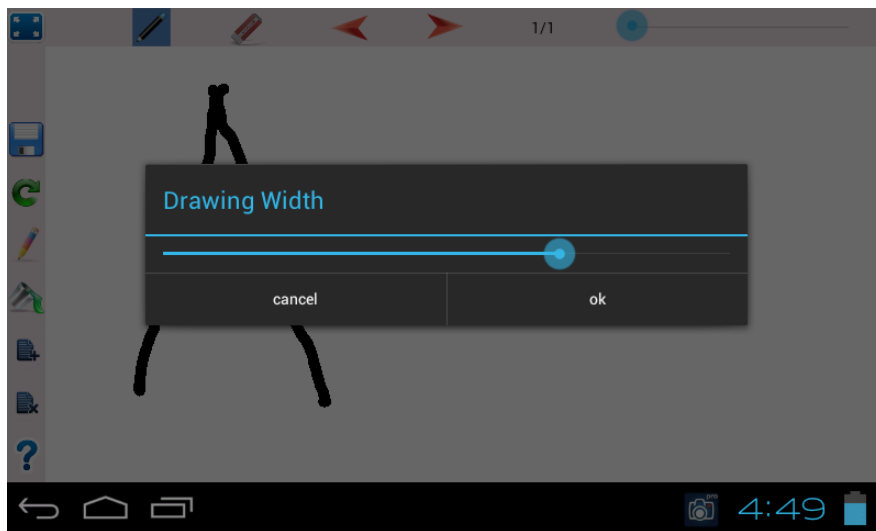


Fig 4.3.4. The user changes the width of the brush/pencil as required.

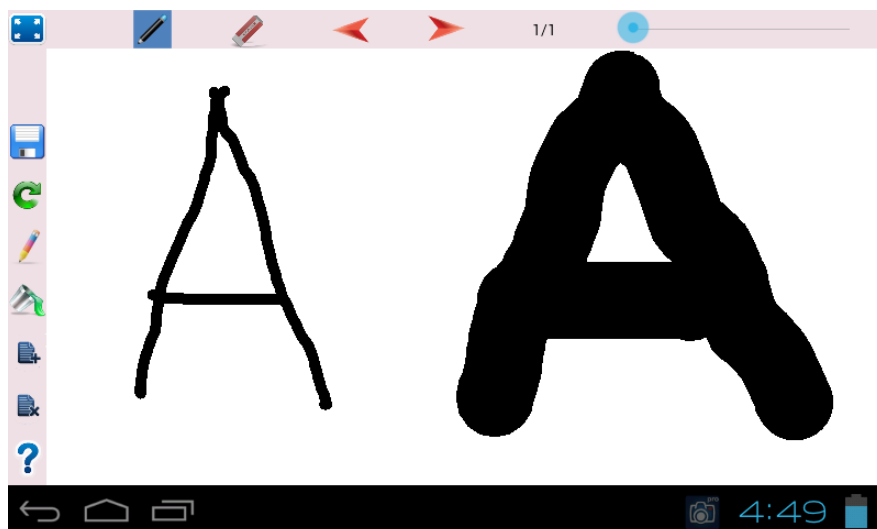


Fig 4.3.5. Now, the user writes the letter 'A' with the new width selected.

2. Eraser:

Since all of us are humans, we are bound to make mistakes. When the user makes any mistake, the eraser can be used to rectify it. By clicking on the eraser icon, the application enters erase mode. Using the eraser, the user can rub out anything unnecessary. We can also change the width of the brush/pencil. On long press of the button, a popup menu appears containing a scrollbar. Using this, the user can adjust the width as required.



Fig 4.4.1. The user writes the letter 'A' using brush/pencil



Fig 4.4.2. She erases it using the eraser.

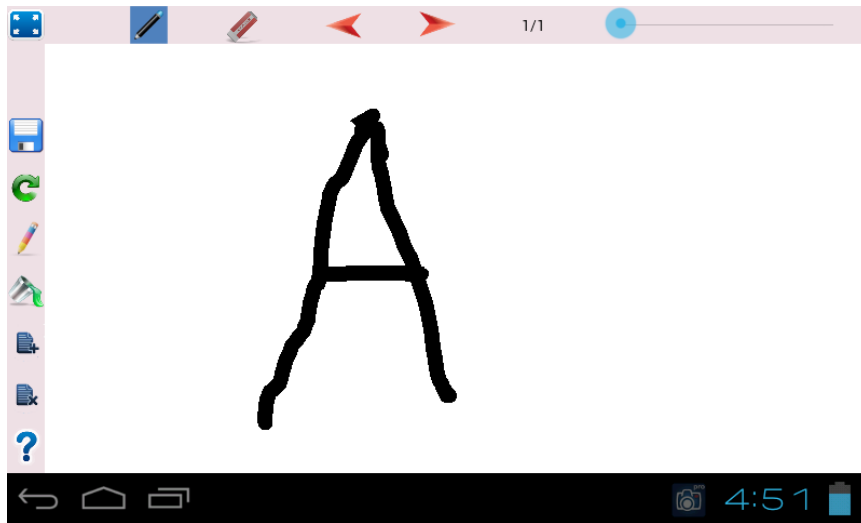


Fig 4.4.3. The user then rewrites the letter 'A' better.

Demonstration of setting eraser width.

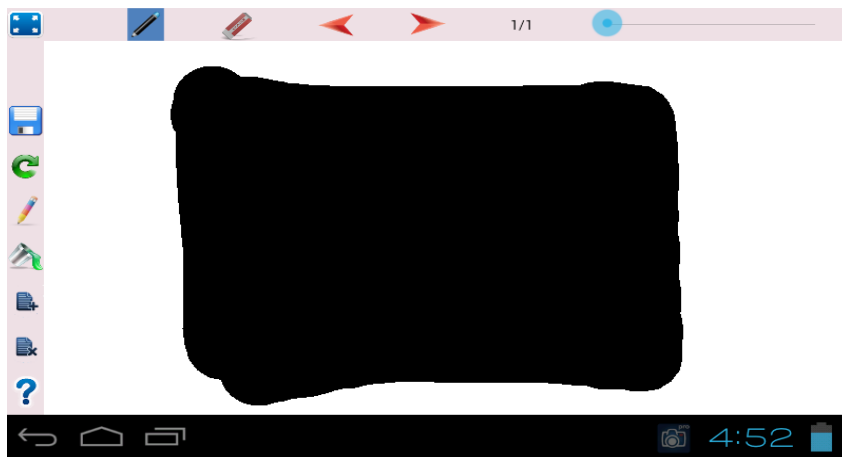


Fig 4.4.4. Initially, there is a black portion drawn by the user on the canvas.

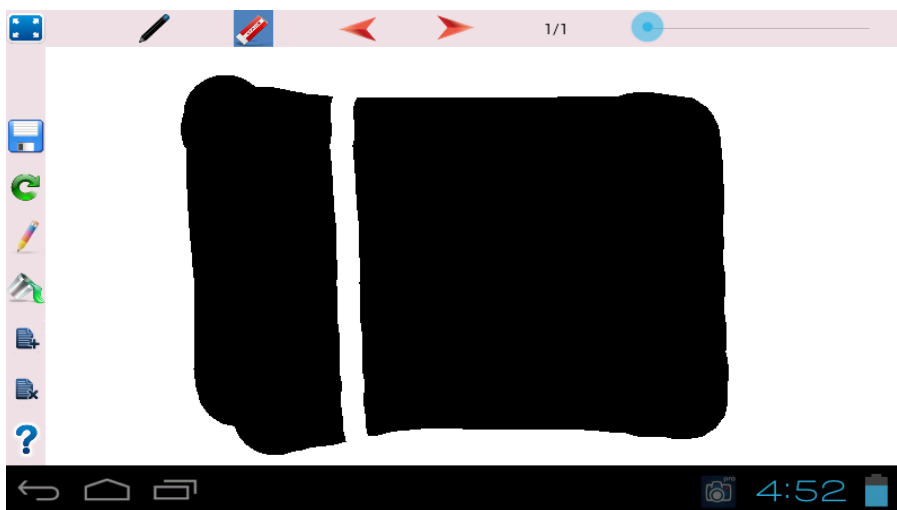


Fig 4.4.5. The user clicks on the eraser button and erases a part of the drawing.

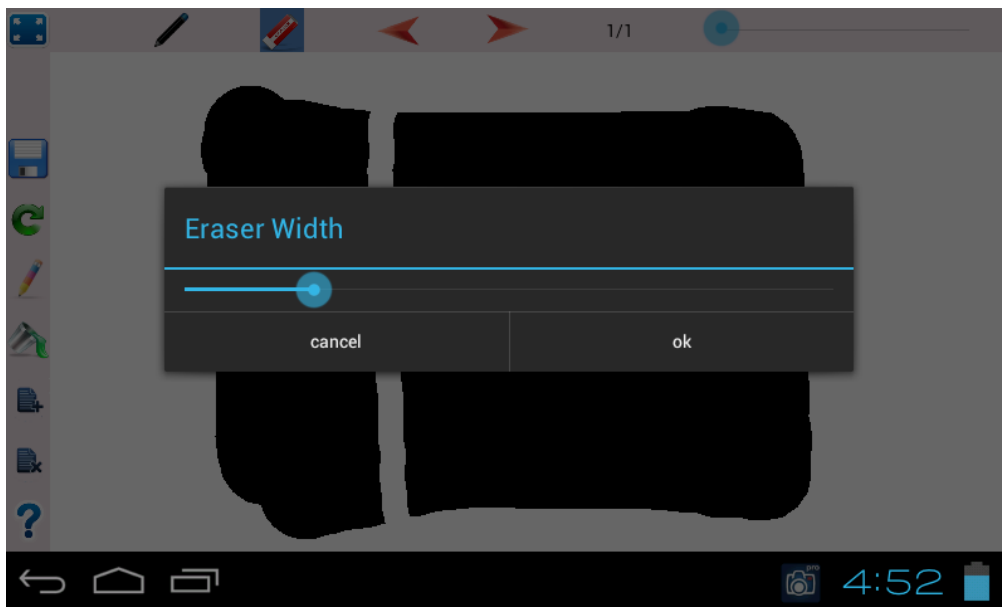


Fig 4.4.6. On long press of the eraser button, a popup menu appears displaying the width of the eraser being used currently.

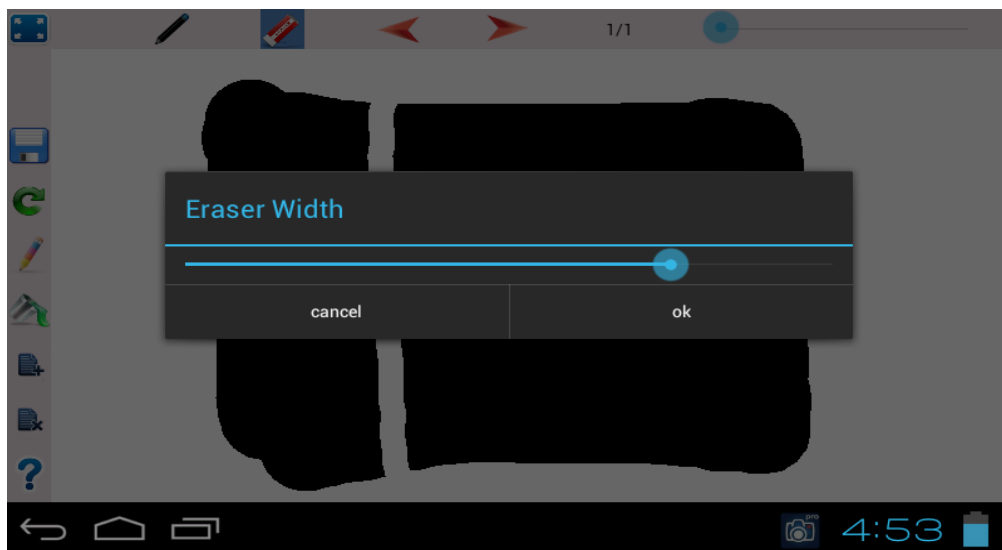


Fig 4.4.7. The user makes use of the scrollbar to change the width of the eraser.

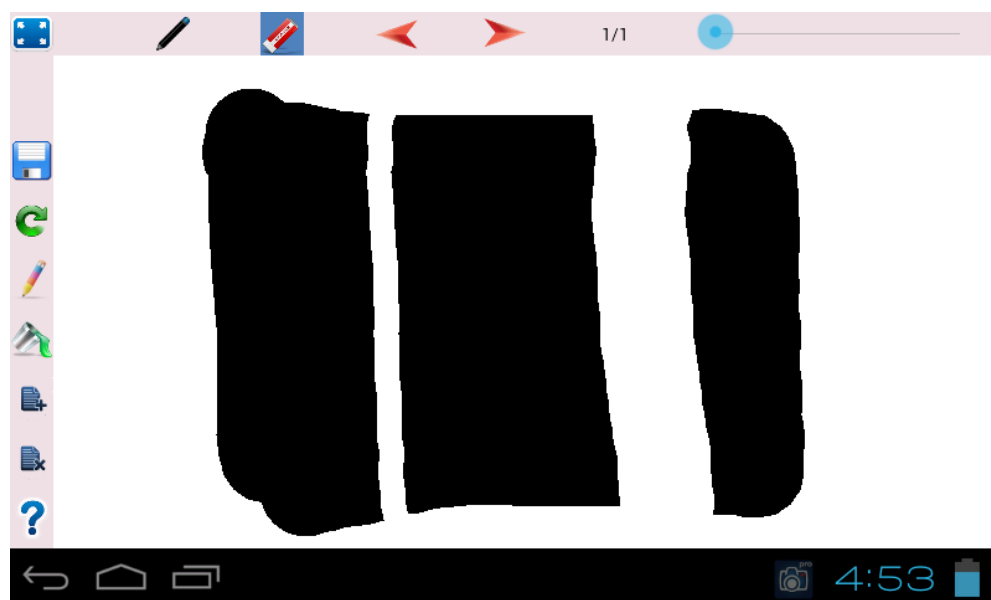


Fig 4.4.8. Now the user erases with the newly selected eraser width.

3. Canvas:

This is the surface on which the user draws.

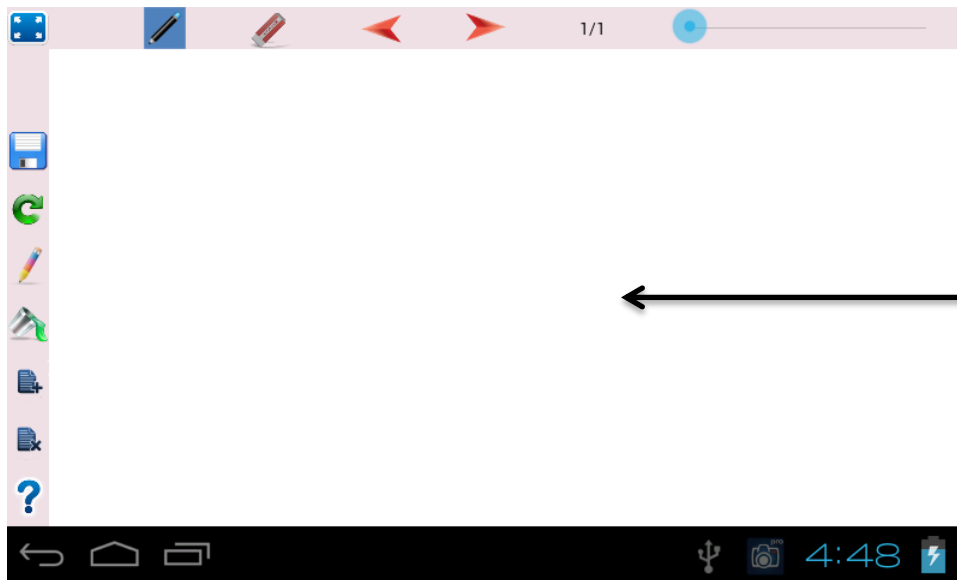


Fig 4.5. Canvas (Area that the user can use to implement

4. Pencil/Brush Colour:

The user can change the colour of the brush/pencil by making use of this button. It launches a colour picker on selection from which the user can select the shade he requires.

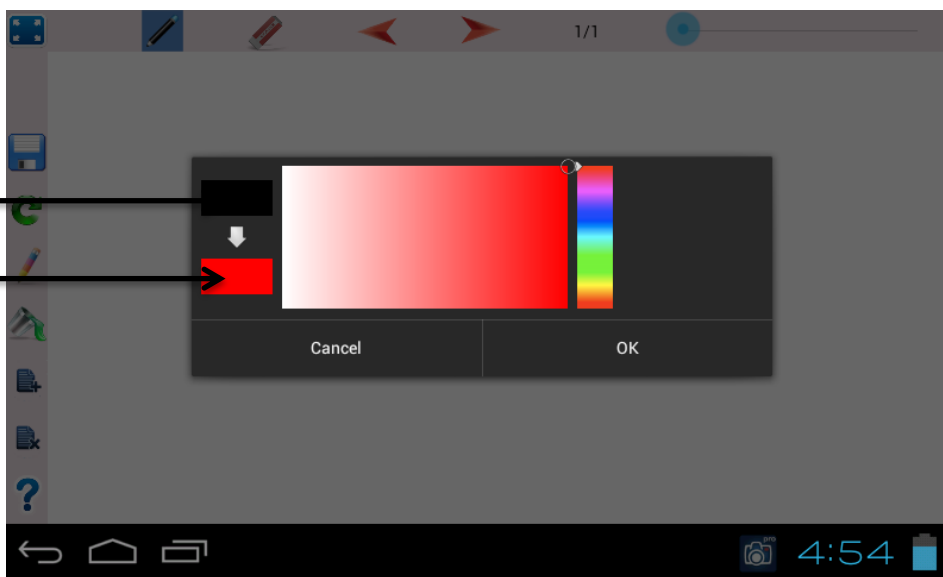


Fig 4.6.1.

- After selecting the brush/pencil colour icon, colour picker is launched.
- The colour of the brush/pencil is presently black. User selects red.

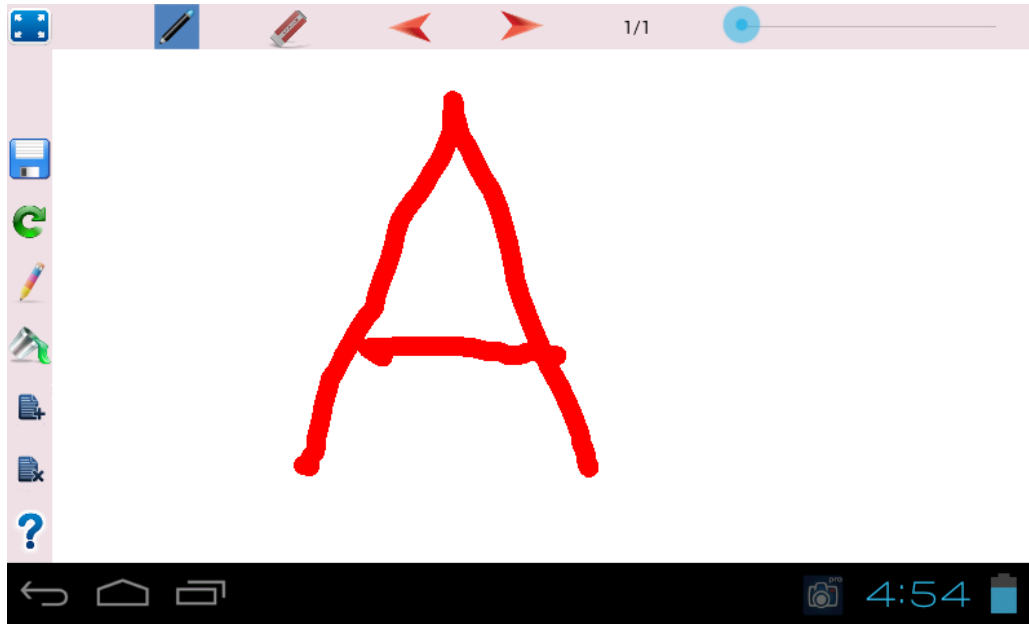


Fig 4.6.2. The user writes the letter 'A' in red (chosen colour).

5. Background Colour:

If the user wishes to change the colour of the background, then this is selected. Like the brush/pencil colour, the selection of this launches a colour picker. The user can choose the colour he wants to apply to the background and continue drawing.

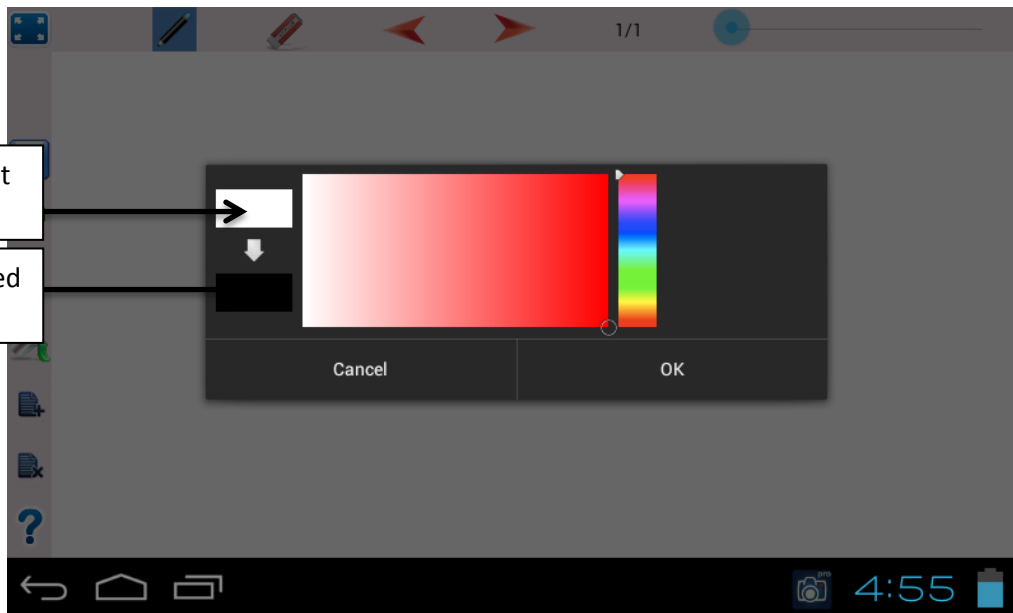


Fig 4.7.1.

- After selecting the background colour icon, colour picker is launched.
- The colour of the background is presently white. User selects black.

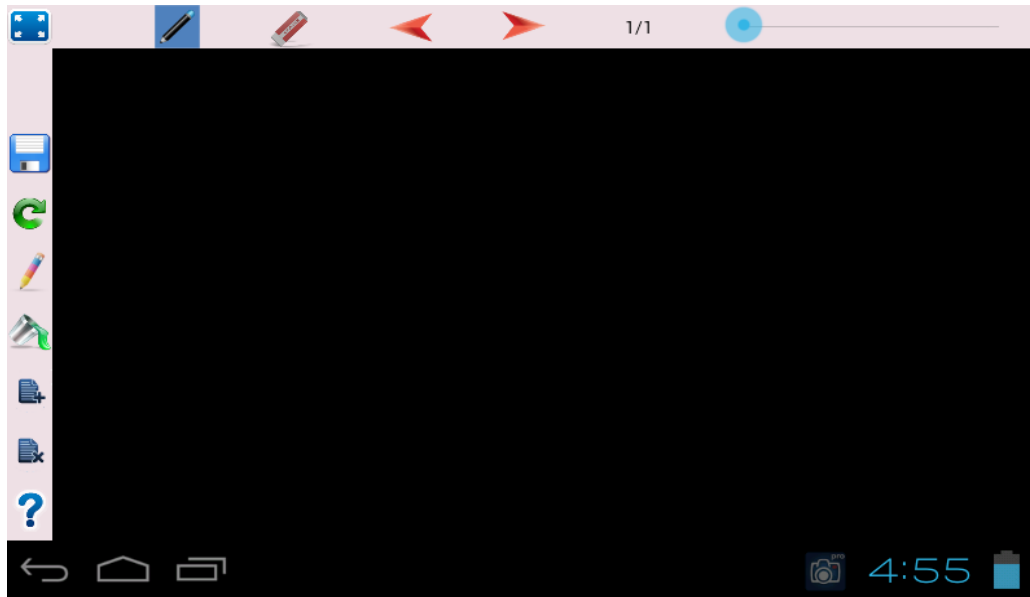


Fig 4.7.2. The background colour is now black (selected colour).

6. Full Screen:

Sometimes the icons on the screen might obstruct what is drawn on the canvas. The user can view the entire image without the icons using the Full Screen button. When in Full Screen mode, the user can continue drawing without any hindrance. On clicking the home button in Full Screen mode, the user can go back to the home screen where all the options are present.

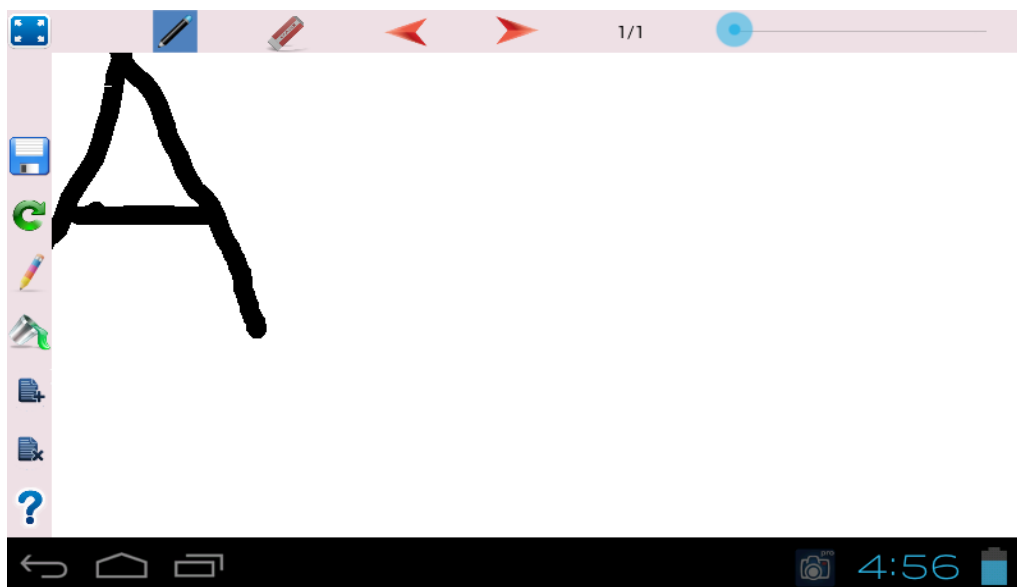


Fig 4.8.1. The user draws in normal mode. The icons are obstructing the drawing.

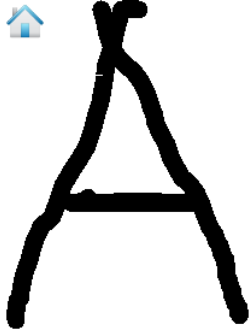


Fig 4.8.2. Full Screen button is clicked and the canvas covers the entire screen making it easier for the user to draw.



Home Button



Fig 4.8.3.

- The user can continue drawing in Full Screen mode.
- Home button can be clicked to exit Full Screen.

7. Refresh:

This button is used to clear the screen. It erases all that is drawn on the canvas. If the user wants to start over, he can make use of this button.

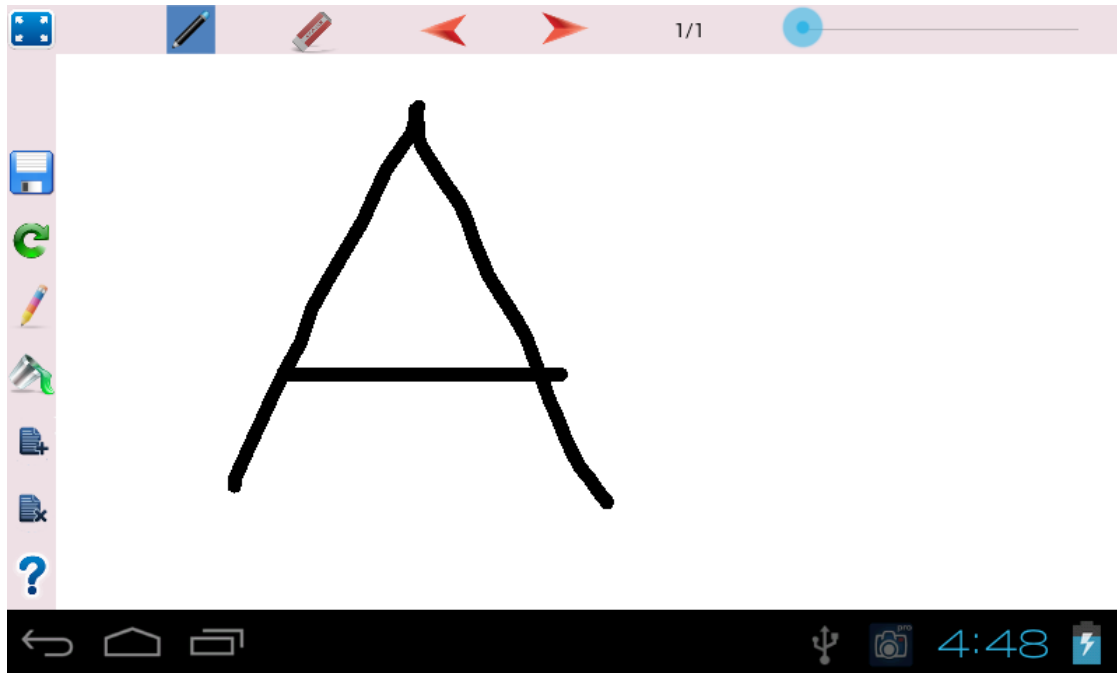


Fig 4.9.1.The user writes the letter 'A' on the canvas.

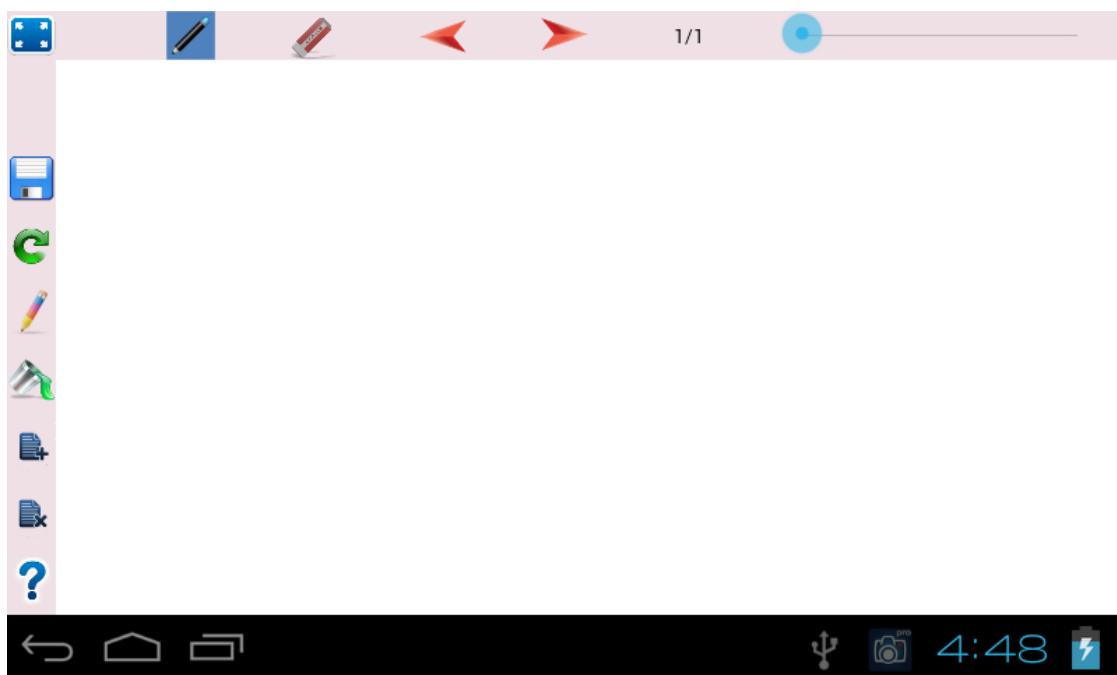


Fig 4.9.2. After clicking the Refresh button, the entire screen

8. Add new page:

This enables the user to add pages to the existing canvas. She can work on two or more pages simultaneously. When background colour is set or changed, it is applied to all the pages automatically.

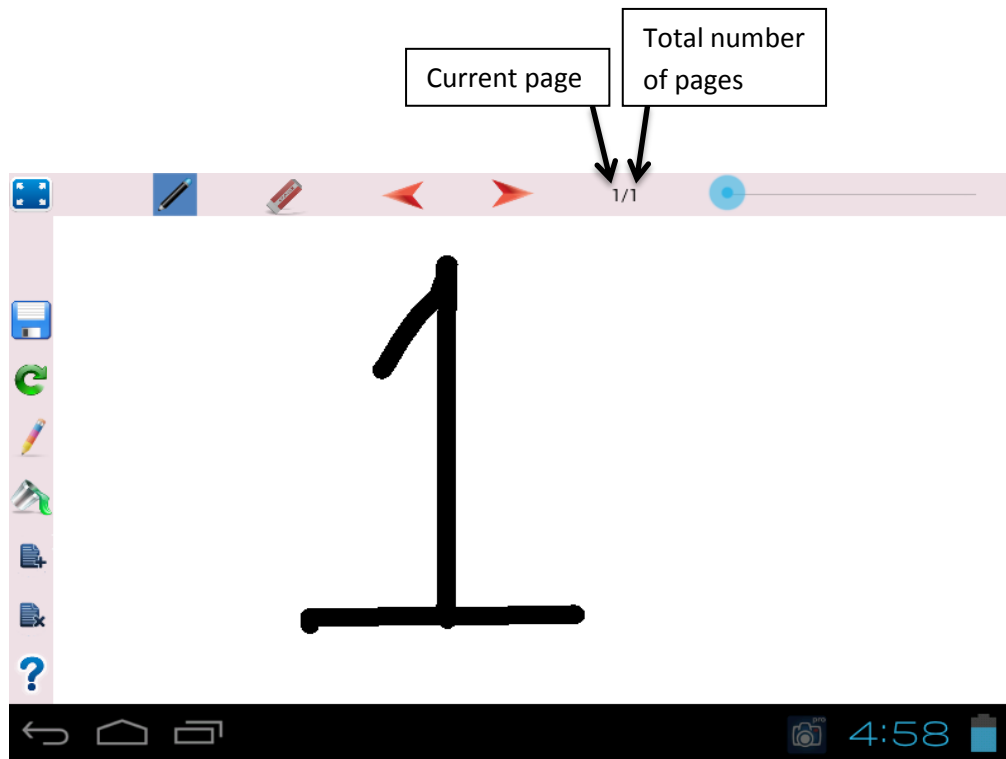


Fig 4.10.1. The user is currently on page number 1



Fig 4.10.2. The user adds another page by clicking on the Add new page button.

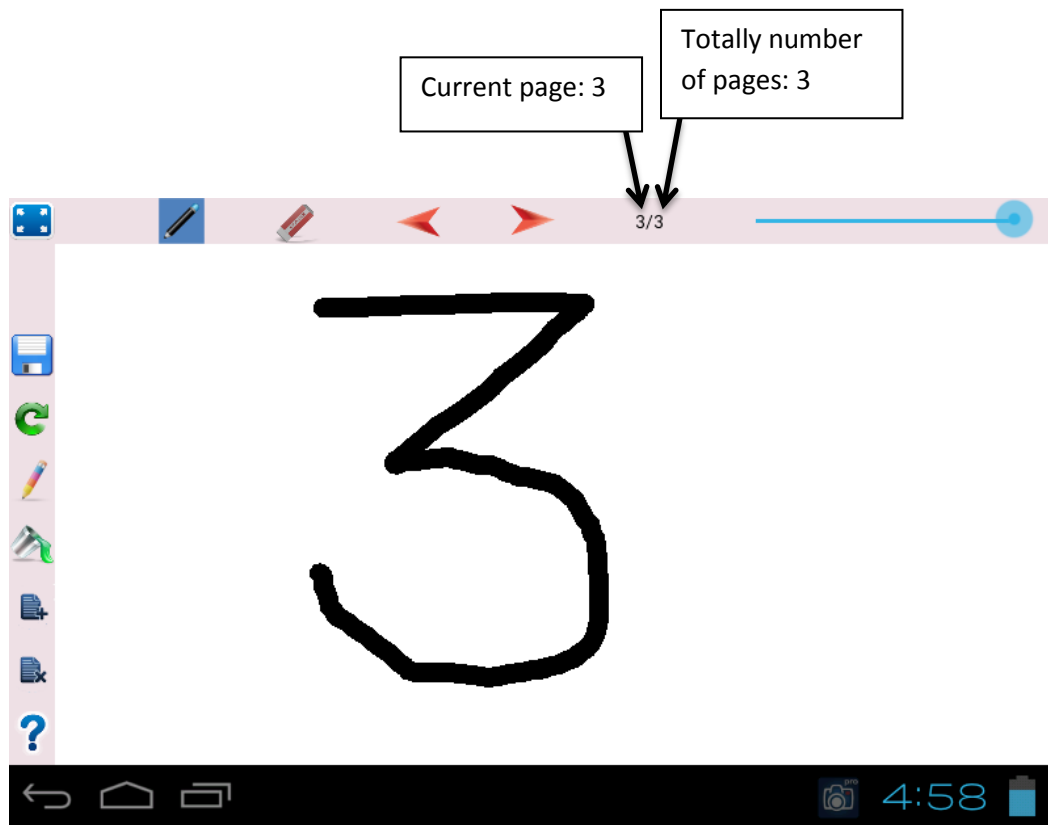


Fig 4.10.3. The user adds a third page by clicking on the Add new page button.

9. Delete page:

To delete a page, the user must go to that page and then click on the Delete page button. If the user wants to delete the pages in sequence then he can go to any page and click the Delete page button. In this case, the pages will begin getting deleted starting from the last page.

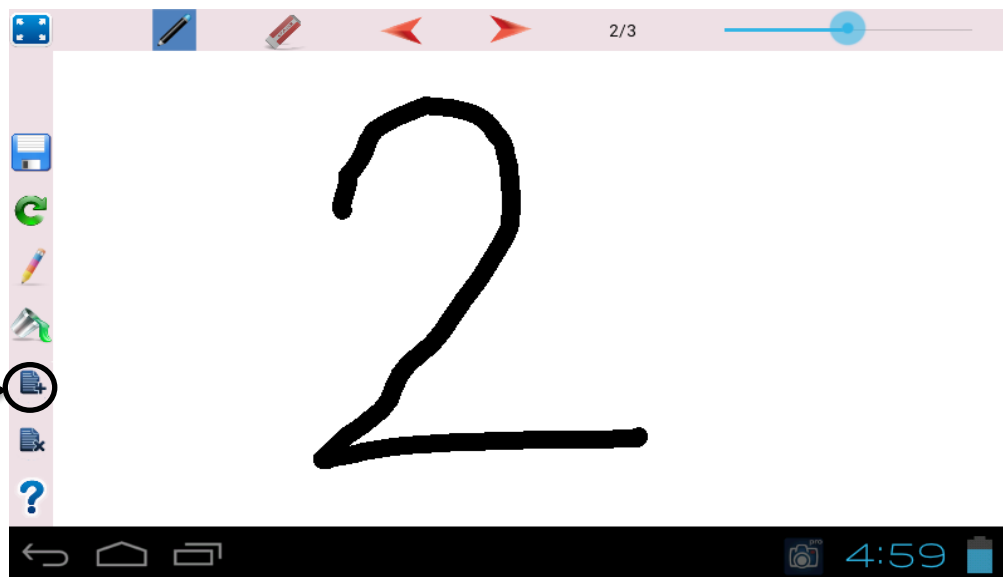


Fig 4.11.1. To delete page 2, the user goes to page 2 and then clicks on the Delete page button.

Delete page button is clicked.



Fig 4.11.2. The user now has pages 1 and 3 after

10. Previous Page:

The user can go to the previous page using this button.

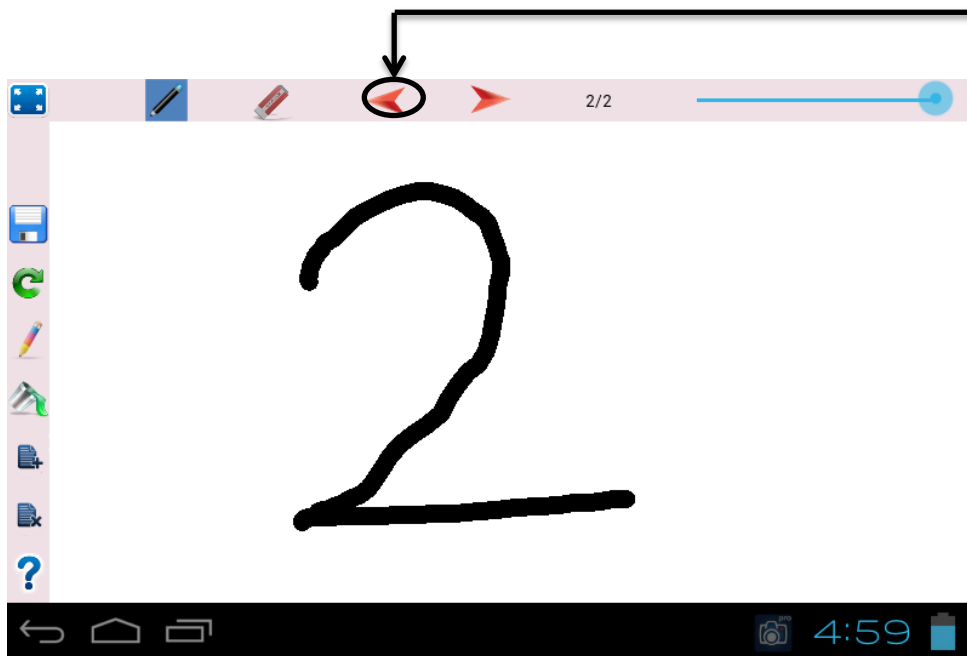


Fig 4.12. The user clicks on this button to go to page 1.

11. Next Page:

The user can go to the next page by using this button.

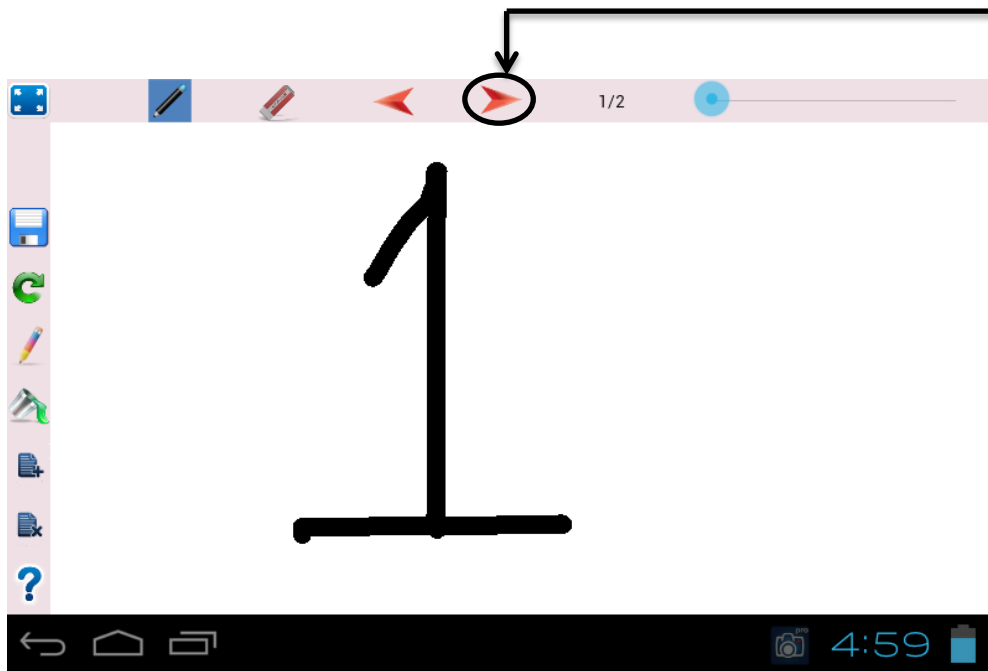


Fig 4.13.1. The user clicks on Next Page button to go to page 2.

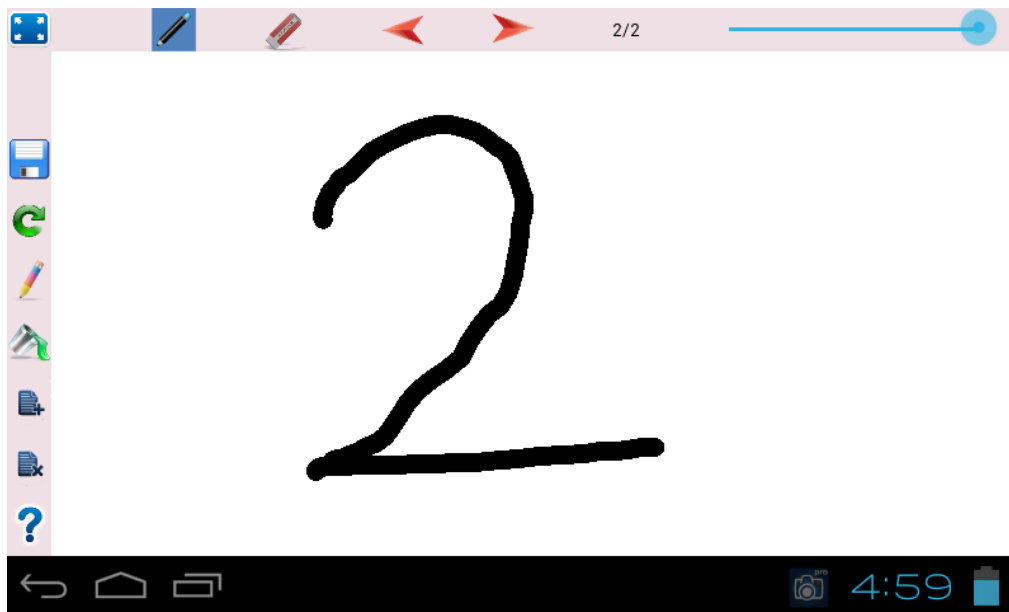


Fig 4.13.2. The user is now on page 2.

12. Page Counter:

Whenever pages are added or deleted, the Page Counter keeps a track of it. It displays the pages in the format 'current page/total number of pages'. If the user wants to see which page he is on and how many pages are there, he can refer to this counter.

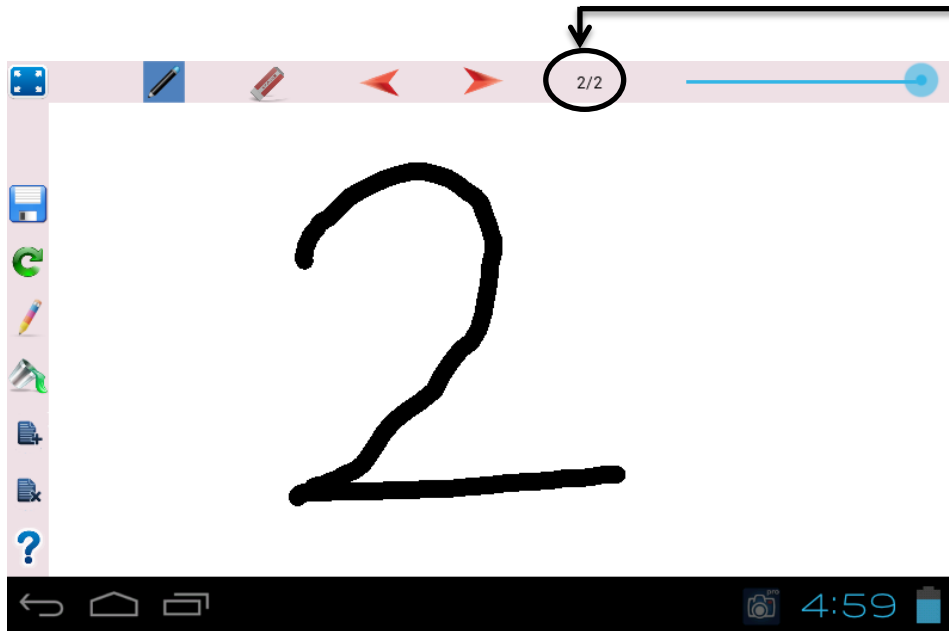


Fig 4.14. The Page Counter shows that there are 2 pages in total and the user is presently working on the 2nd page.

13. Page Scrollbar:

Page Scrollbar is another way of scrolling through the pages present. Just scroll across the scrollbar to the page that has to be viewed.

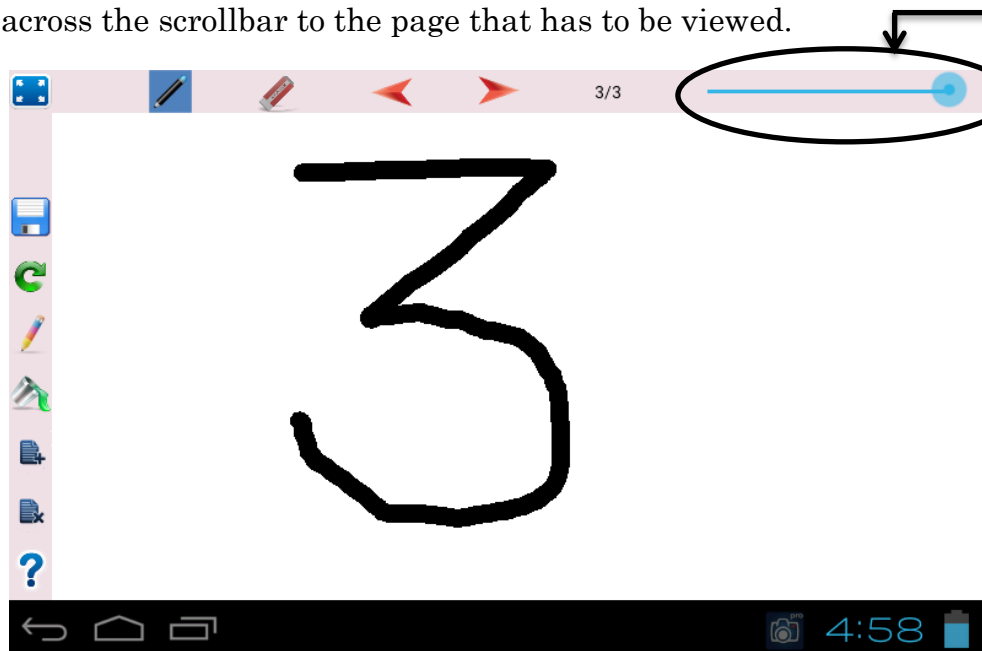


Fig 4.15. The user has scrolled to page 3 by dragging the Page Scrollbar until it came to page 3.

14. Save:

This enables the user to save any page as a pdf or image. By clicking on the save button, the user can store the drawing as an image or pdf. In case of presence of multiple pages, on clicking Save, all the pages get saved.

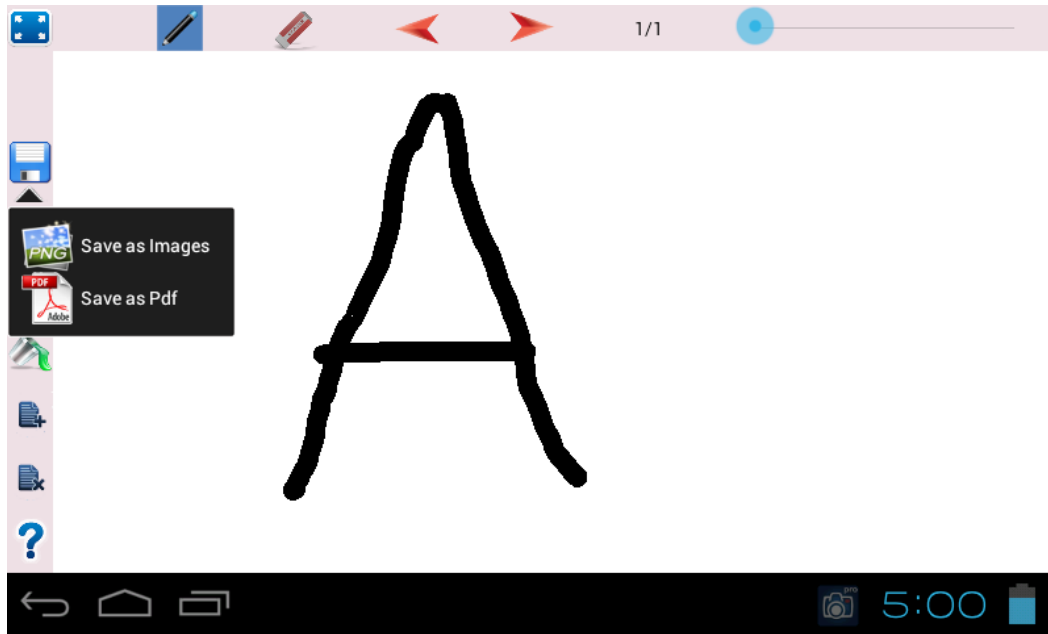


Fig 4.16.1. When the save button is clicked, a menu appears. The user can save his drawing as an image or a pdf.

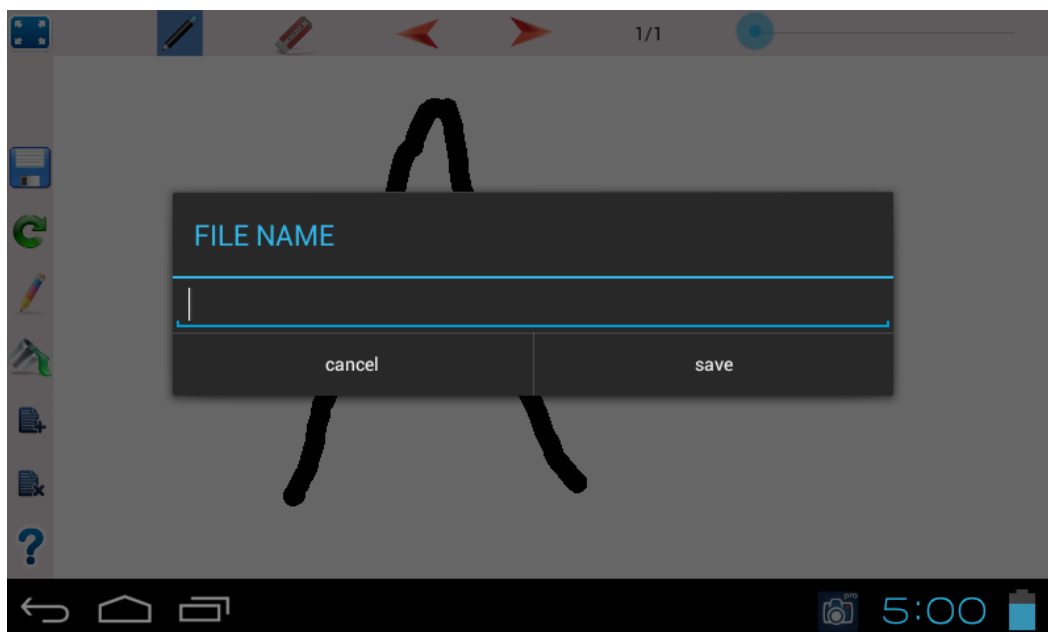


Fig 4.16.2. The user is asked to enter a name for the file to be saved as.

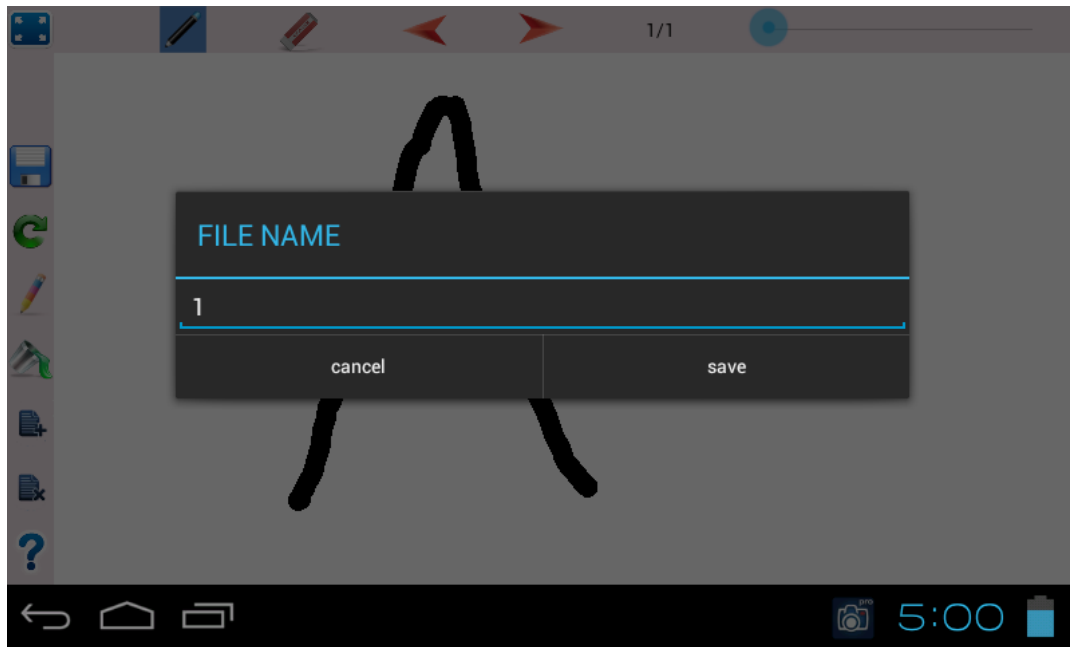


Fig 4.16.3. The user enters a name for the image. Here, the name is '1'.

Saving multiple pages.

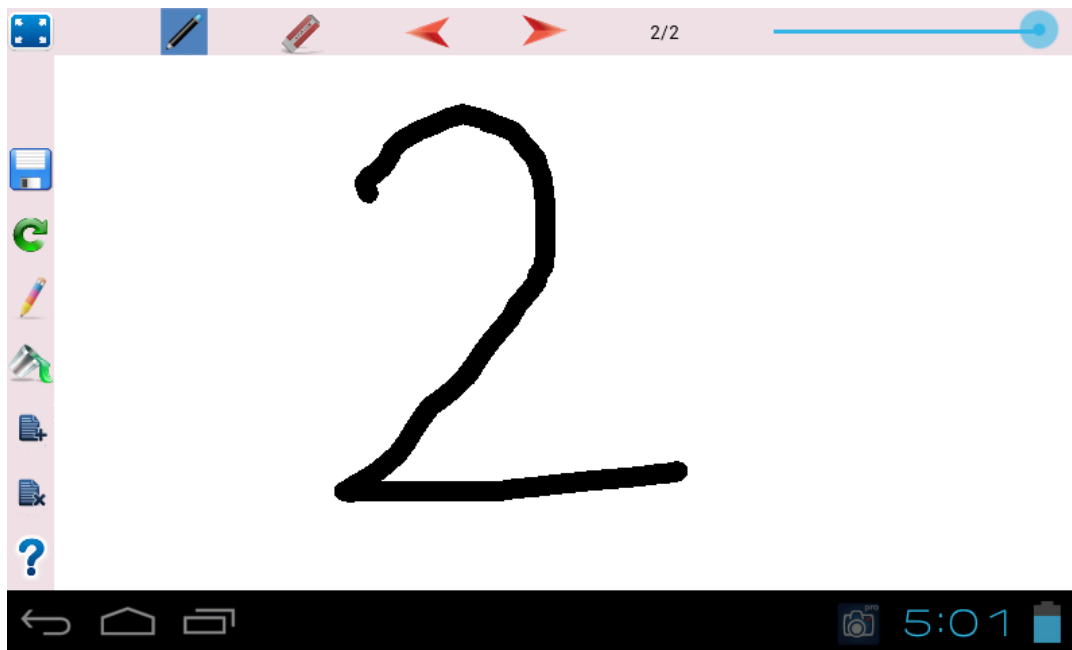


Fig 4.16.4. There are 2 pages

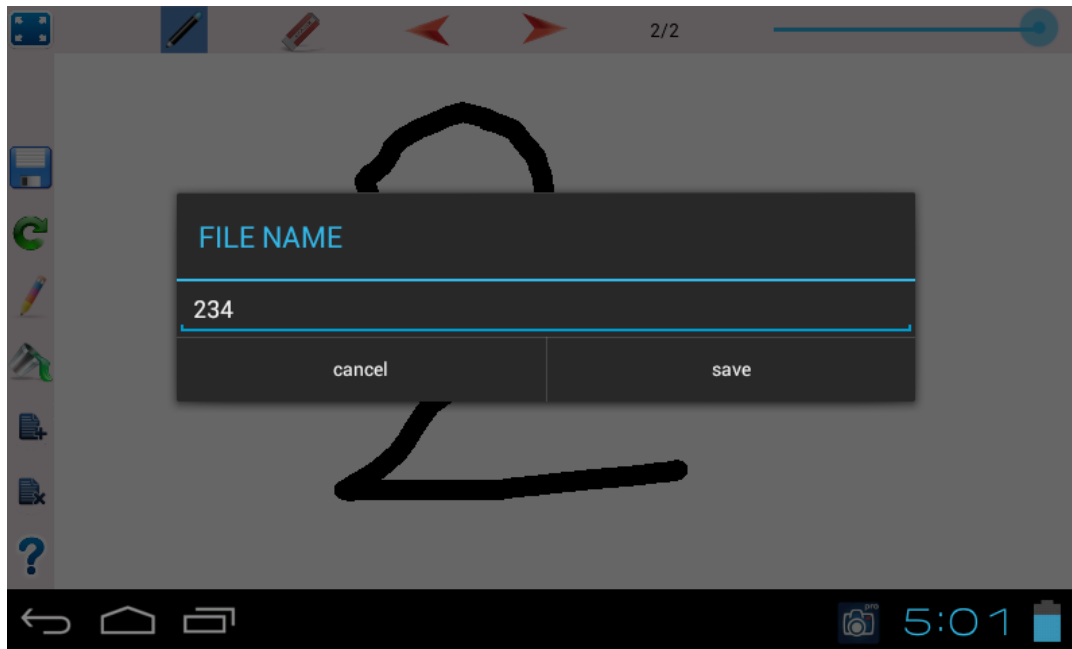


Fig 4.16.5. The user clicks on the save button to save all the pages. She names it '234'.



Fig 4.16.6. Both the pages are saved simultaneously. They are labeled as '2341' and '2342' automatically.

15. Help:

This will guide the user on how to use the application.



UML:

The components of the UML are:

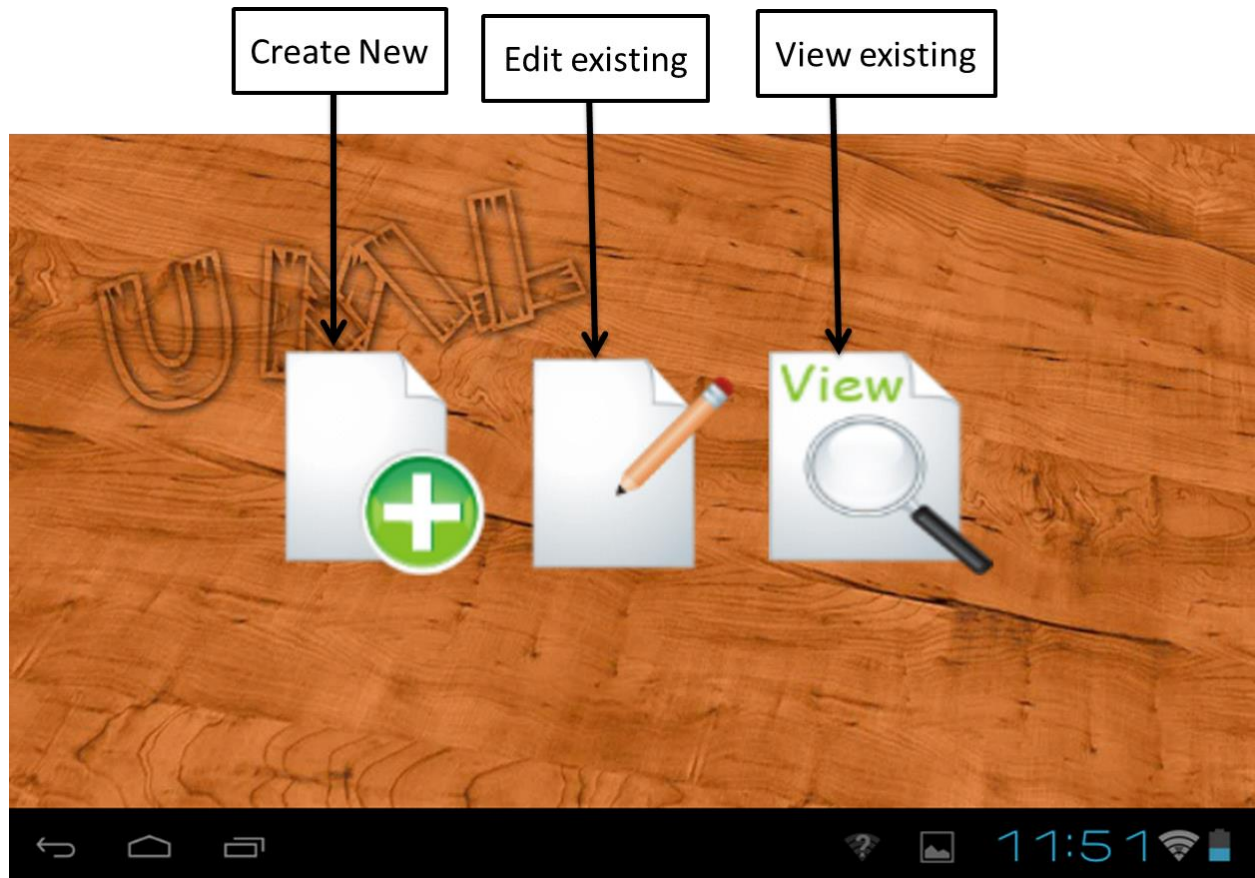


Fig 4.17.

1. Create New
2. Edit existing
3. View existing
4. Save
5. Shapes
6. Move
7. Text
8. Delete
9. Background colour

1. Create New:

If the user wishes to create a new diagram, then this button is clicked. On clicking, a blank canvas appears.

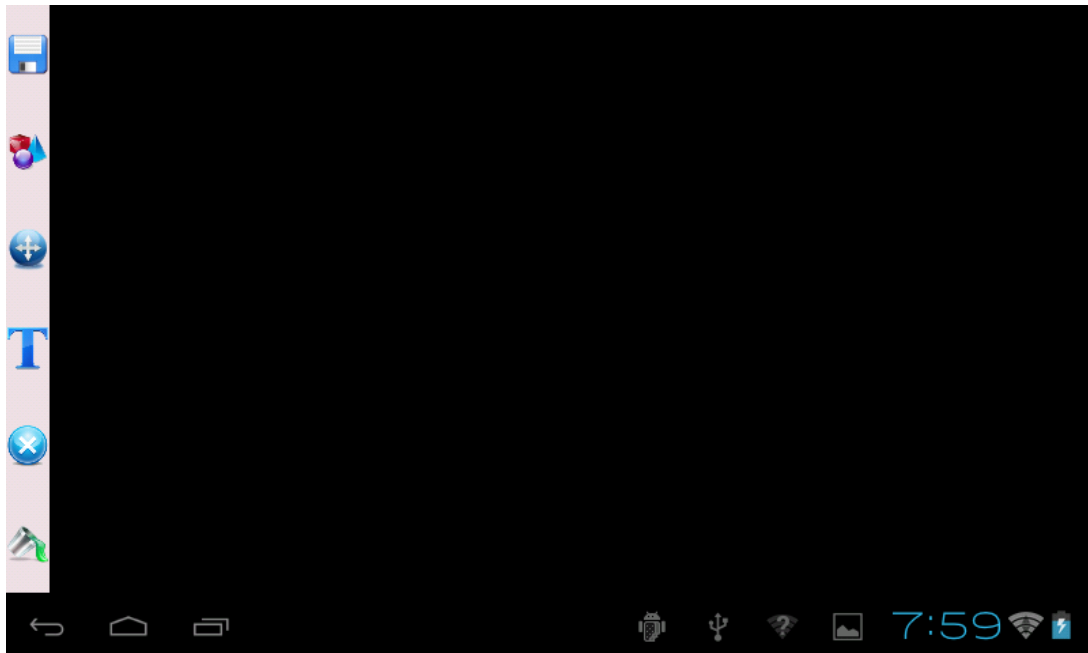


Fig 4.18.
New canvas.

2. Edit existing:

On clicking this button, a list of previously saved images is displayed. The user can select from this which one is to be edited.



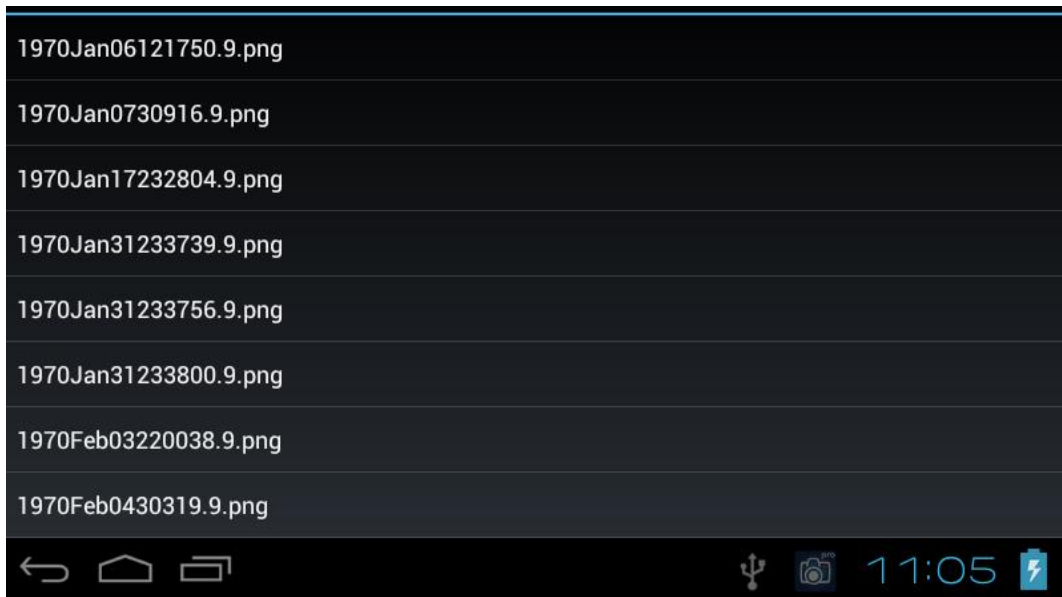


Fig 4.19. On clicking the Edit button, the list of previously saved images is displayed for the user to select from.

3. View existing:

On clicking this button, a list of previously saved images is displayed. The user can select any image to be viewed only. Images cannot be edited here.

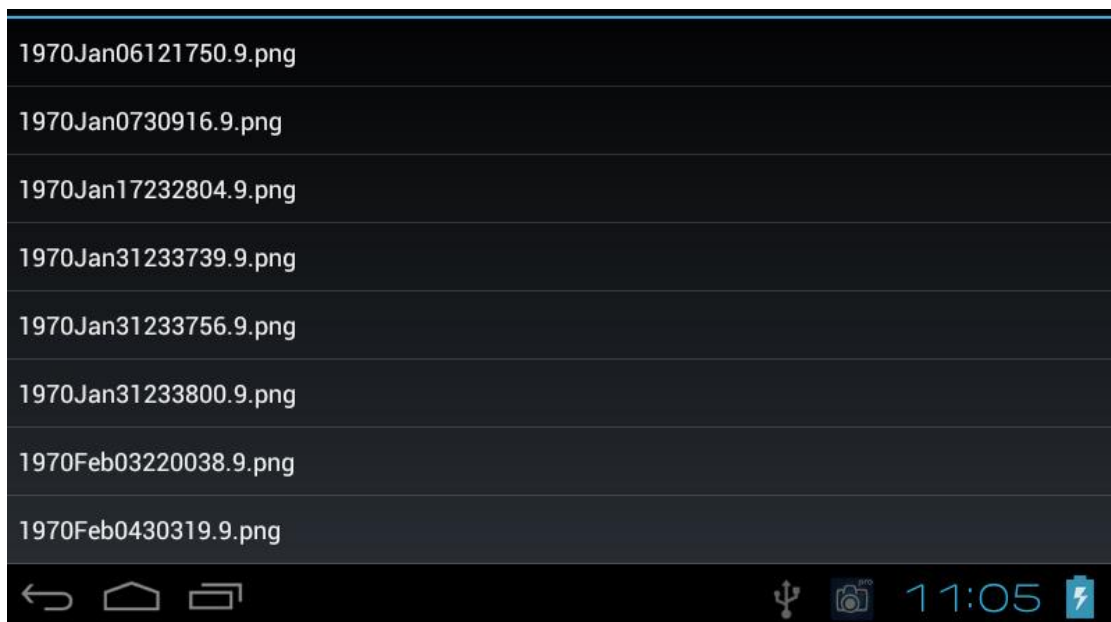


Fig 4.20. The list of images that can be viewed is displayed.

4. Save:

The Save button enables the user to save the drawing as an image (PNG format). The drawing is named automatically using the date when the drawing was made.

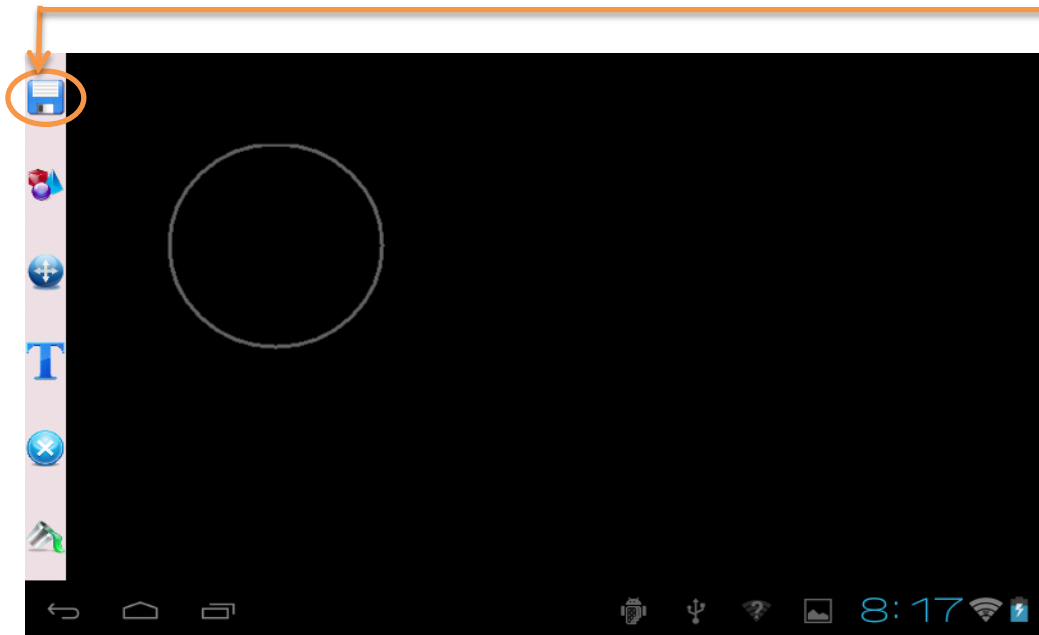


Fig 4.21. On clicking this, the drawing is saved.

5. Shapes:

Using this option, the user can draw shapes:

5.1. Rectangle

5.2. Circle

5.3. Oval

5.4. Line

5.5. Actor



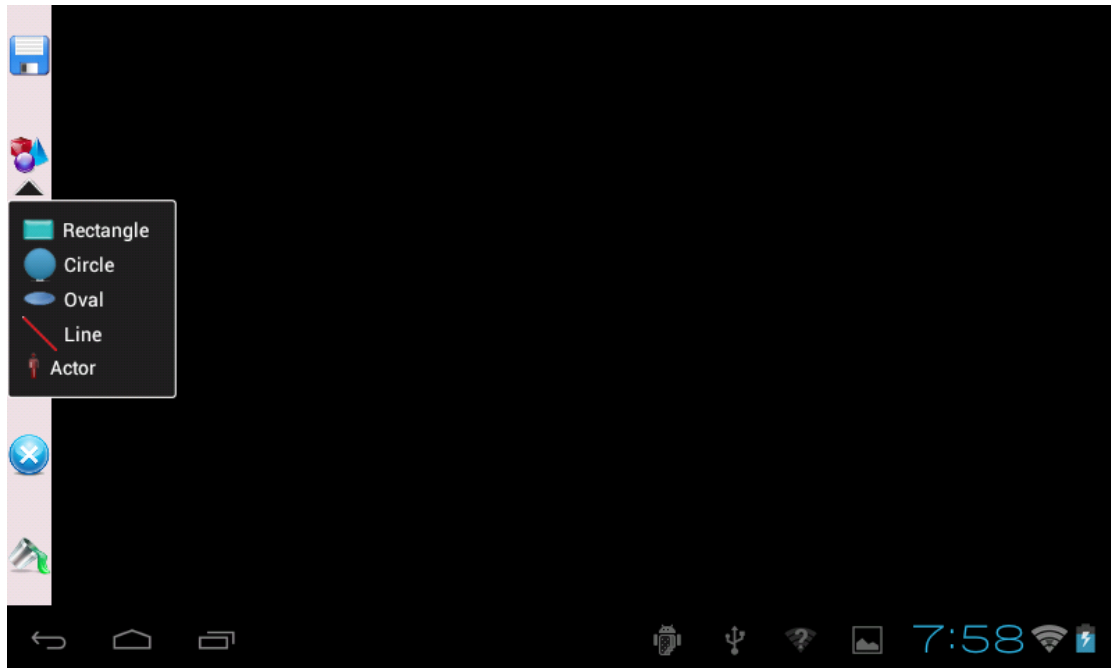


Fig 4.22.1.
Shapes
menu

5.1. Rectangle:

The user can draw the required size of the rectangle by tracing its diagonal on the canvas. The application calculates the length and breadth of the rectangle based on the diagonal and draws the rectangle.

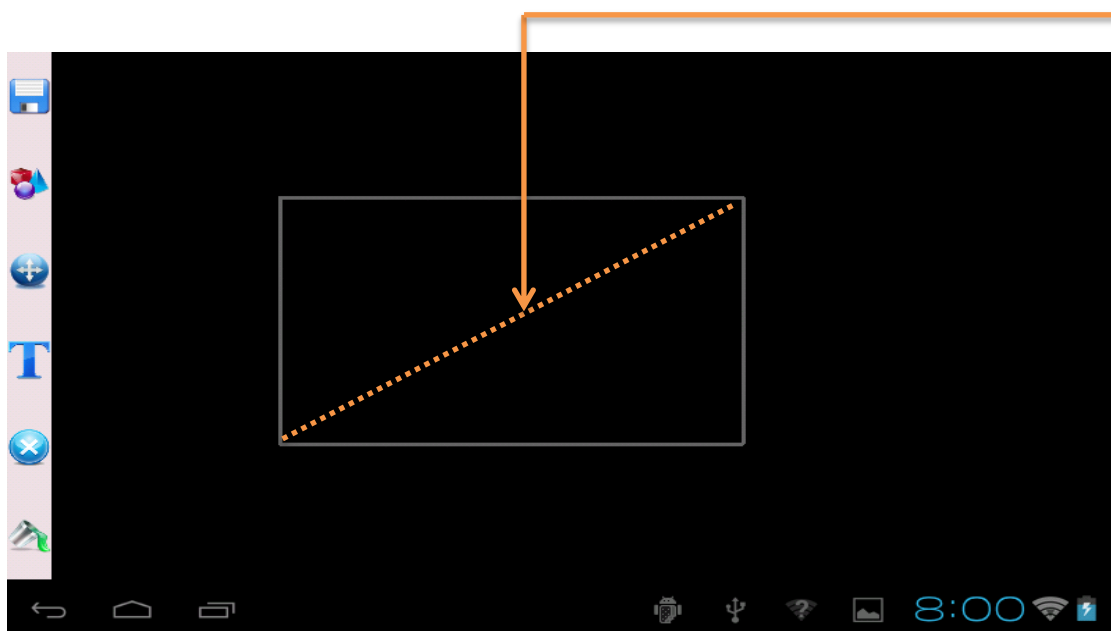
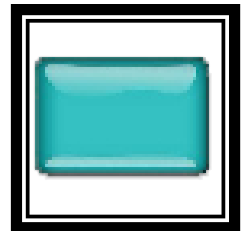


Fig 4.22.2. The
dotted line
traces the finger
movement of
the user.
DiaSlate
calculates the
length and
breadth of the
rectangle
accordingly and
draws it.

5.2. Circle:

The user can draw a circle by tracing the diameter on the canvas. When the user moves his finger along the diameter of the circle required, the application calculates and constructs the circle.

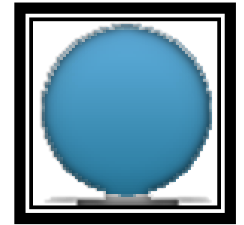
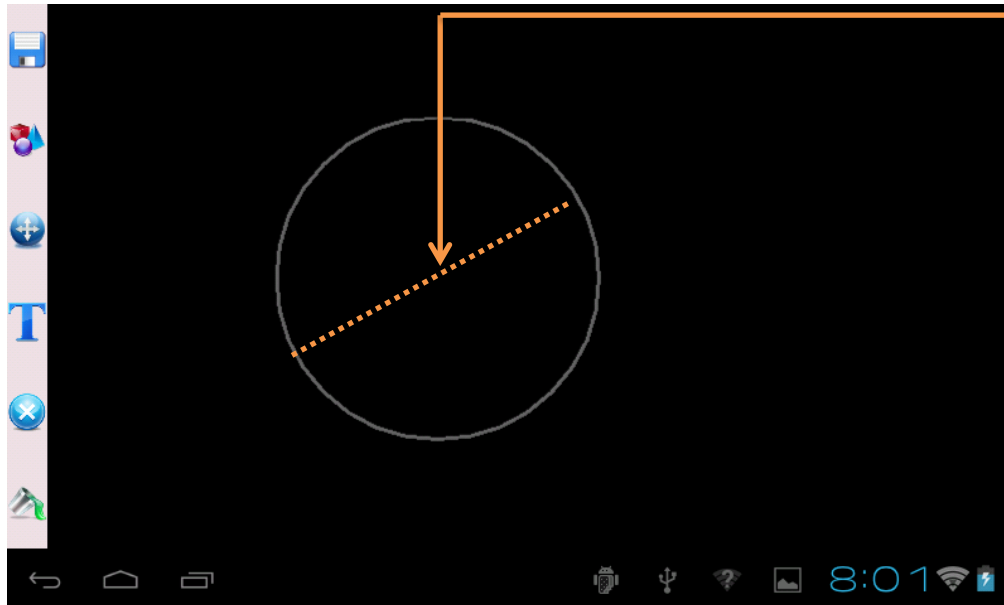


Fig 4.22.3. The dotted line depicts the path traced by the user's finger. It is taken as the diameter of the circle to be constructed by the application.

5.3. Oval:

Drawing an oval is similar to drawing a circle. But here, the longer diameter should be traced by the user. The length traced by the user is taken as the longer diameter by the application and the oval is calculated and drawn accordingly.

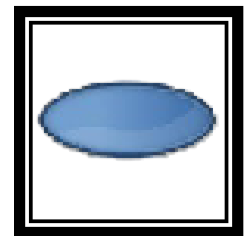
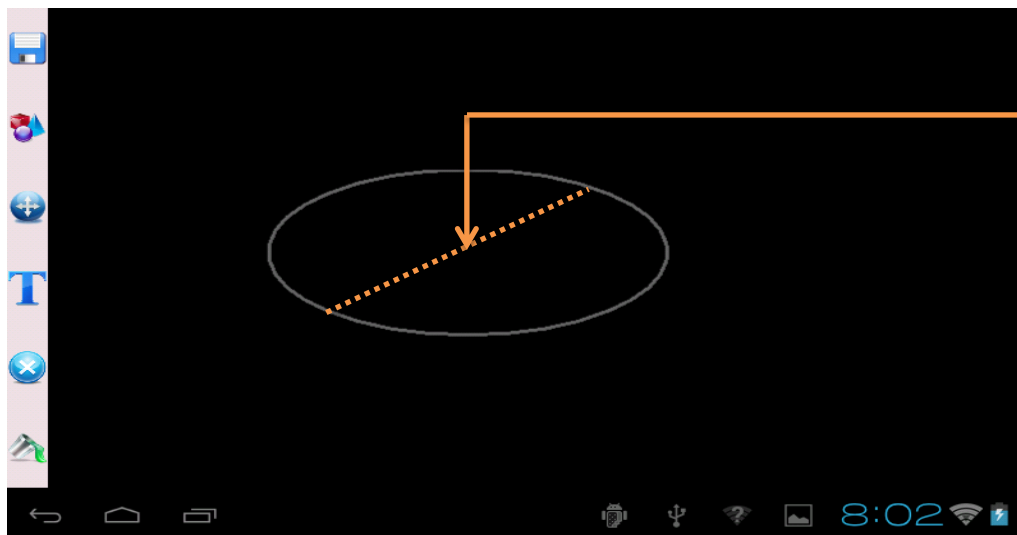


Fig 4.22.4. The dotted line represents the path of the user's finger on the canvas. DiaSlate takes this input as the longer diameter and constructs the oval/ellipse accordingly.

5.4. Line:

To draw a line, the user must trace it from starting point to the end point.

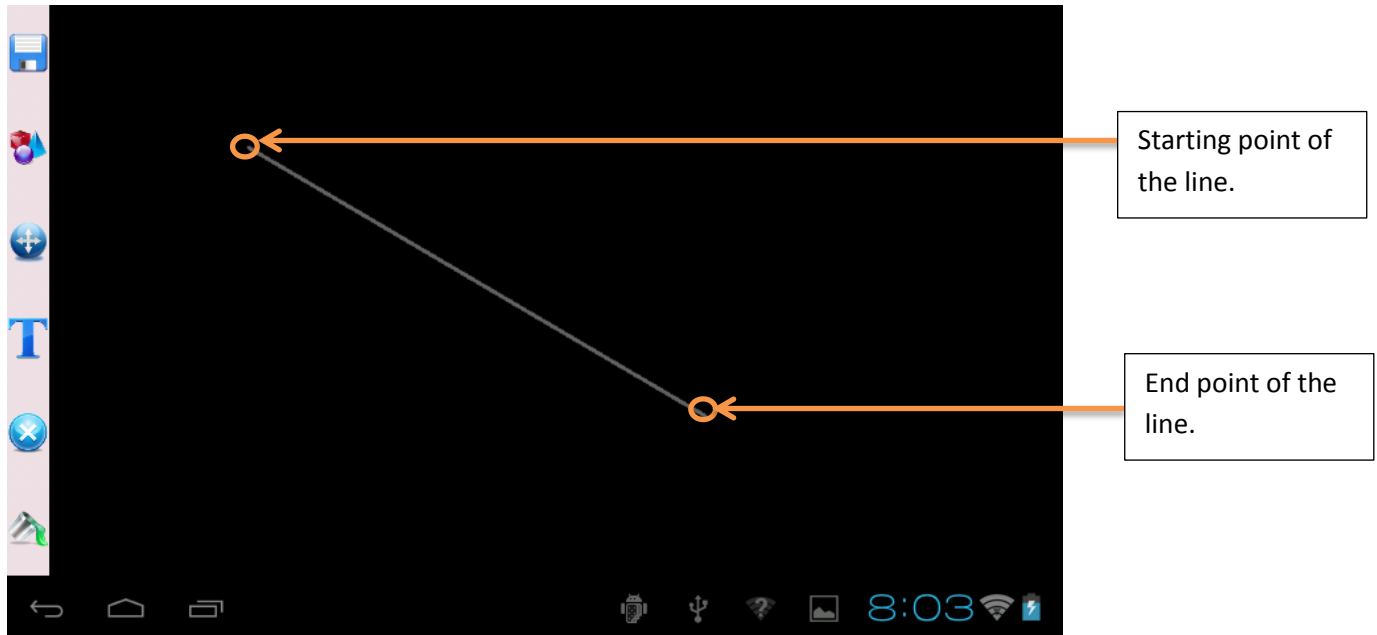
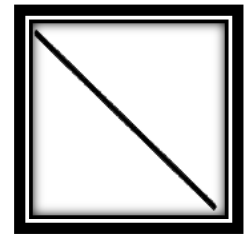
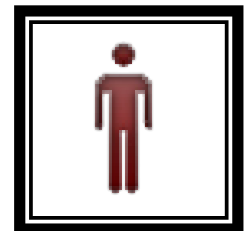


Fig 4.22.5.

5.5. Actor:

To draw an actor the user must click on this button and then where the actor is to be placed. The size of the actor is constant and cannot be changed.



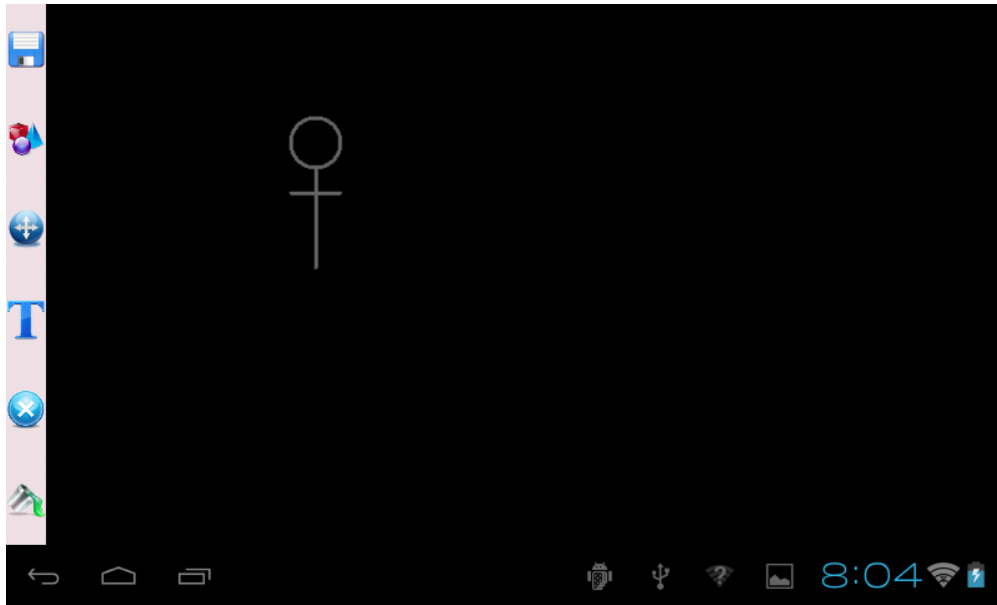


Fig 4.22.6. The user has clicked on the actor button and then on the position on the screen she wishes to place it.

6. Move:

This option enables the repositioning of the shapes that have been drawn.

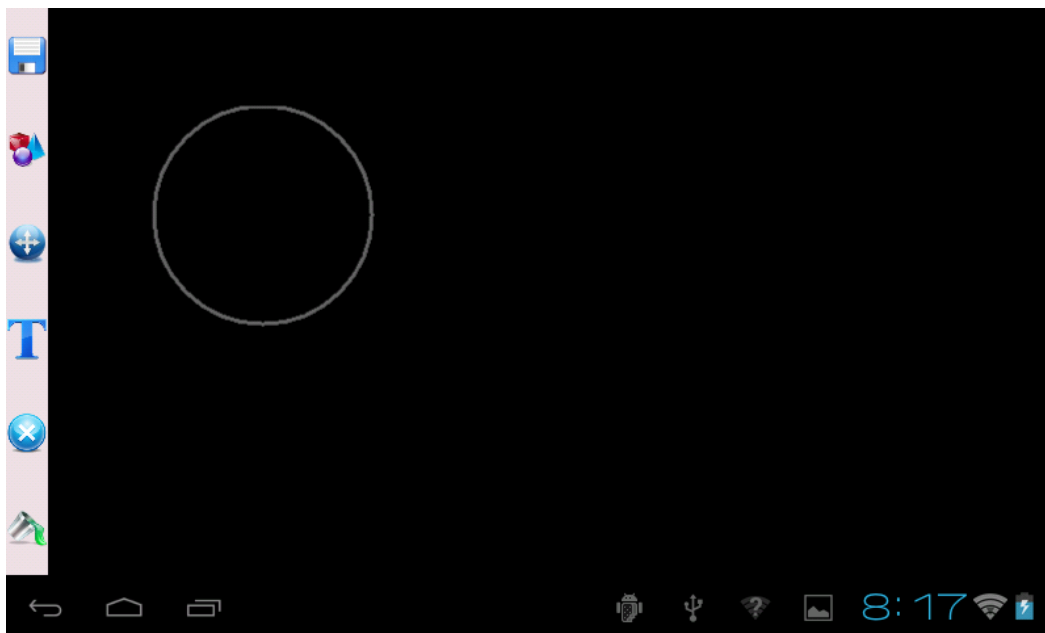


Fig 4.23.1. Initial position of the shape.

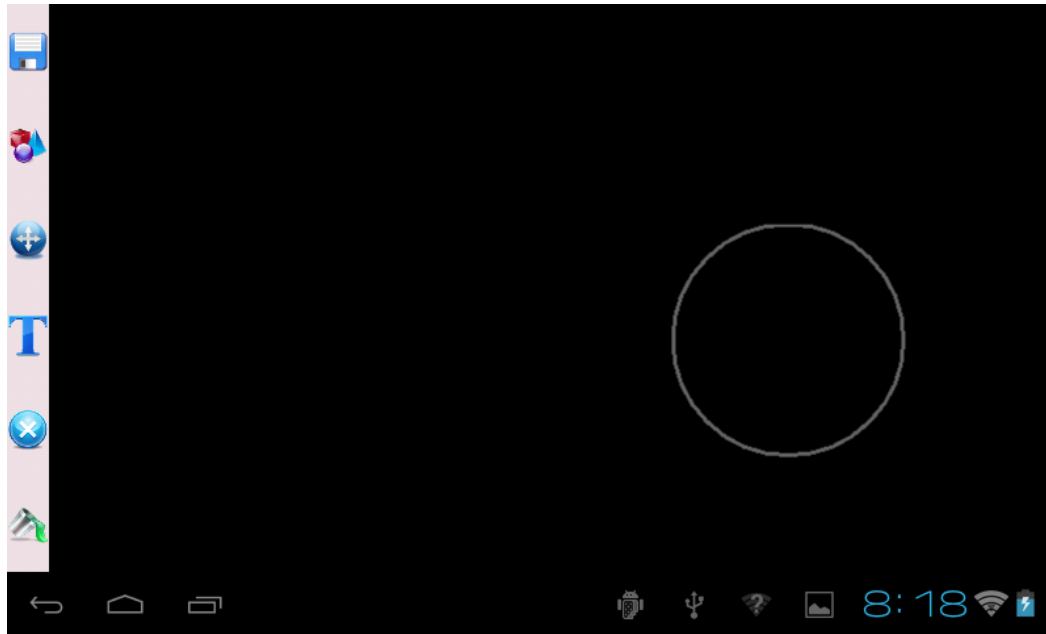


Fig 4.23.2. Position of the shape after moving it.

7. Text:

The user can add text to the shapes by using this option. First, the text button is clicked and then the shape to be labeled is selected. This generates a popup which takes input as the name.

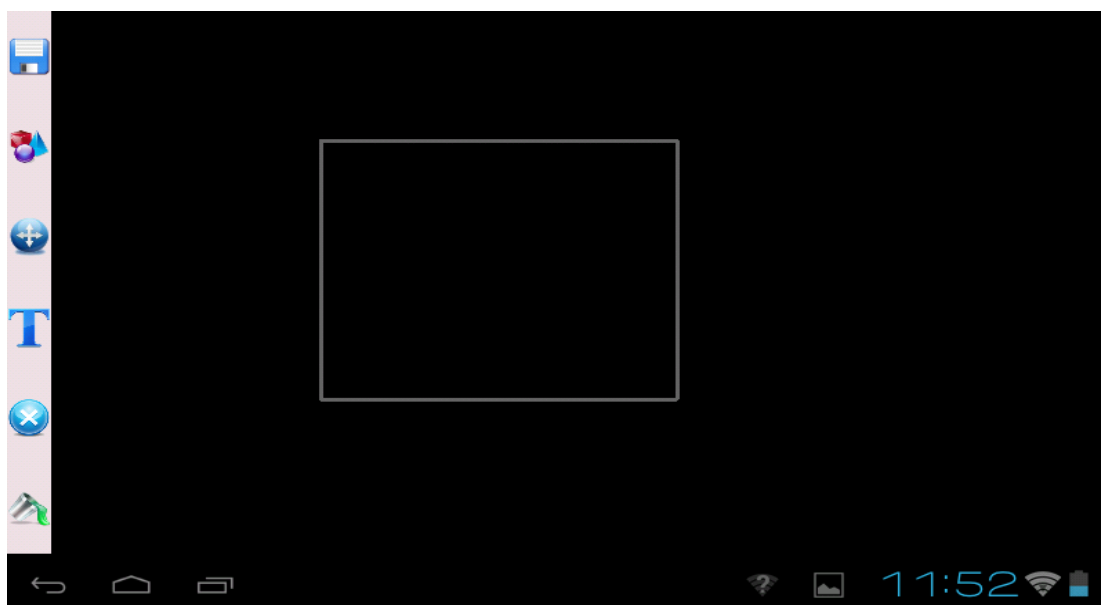


Fig 4.24.1. First, a shape/entity is drawn. Here, a rectangle is drawn.

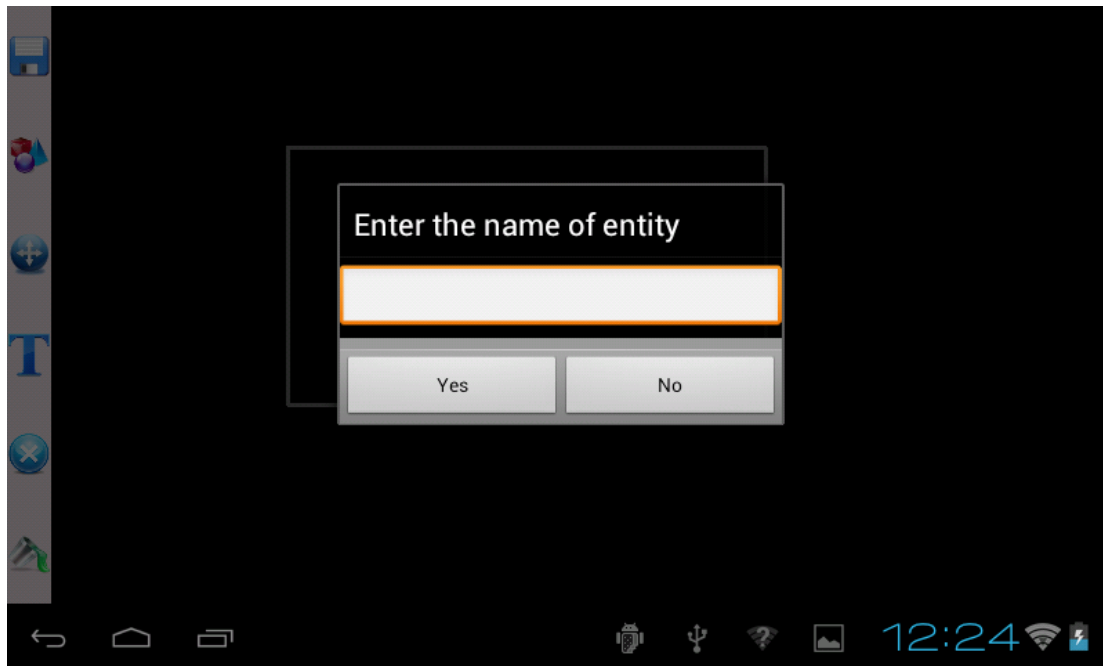


Fig 4.24.2. After clicking on the text button and selecting the shape, a popup appears. It requires a name as input.

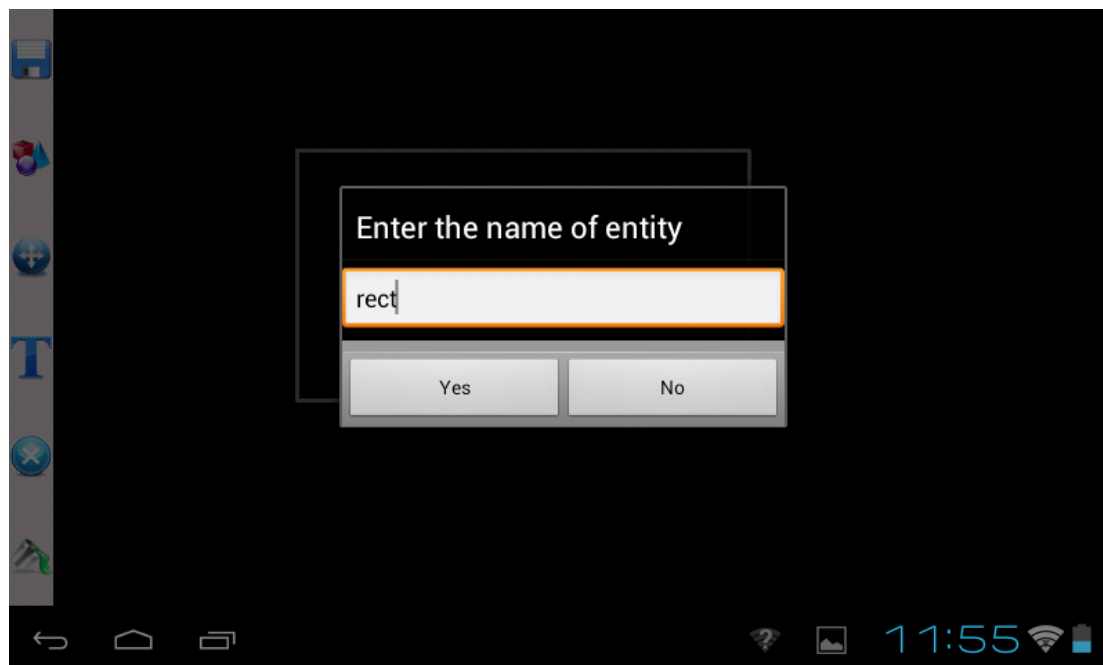


Fig 4.24.3. The user enters the required name for the entity. Here it is named 'rect'.

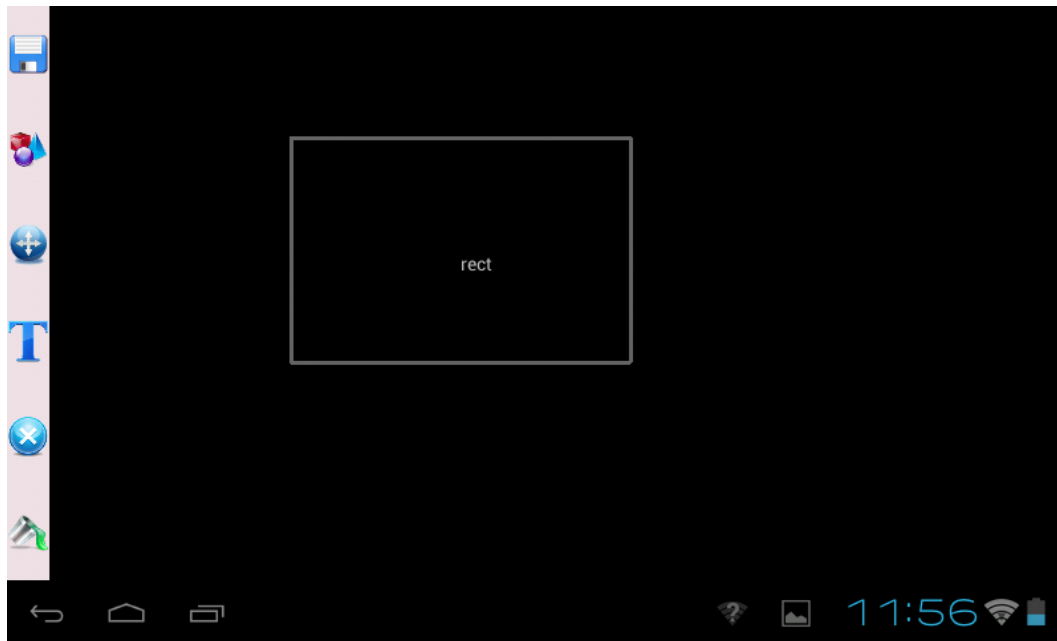


Fig 4.24.4.The entity is now labeled as 'rect'.

8. Delete:

In case the user wants to remove an unwanted shape that has been drawn, then the Delete option can be used. On clicking the delete button, the application enters delete mode and the shapes to be deleted can be selected.

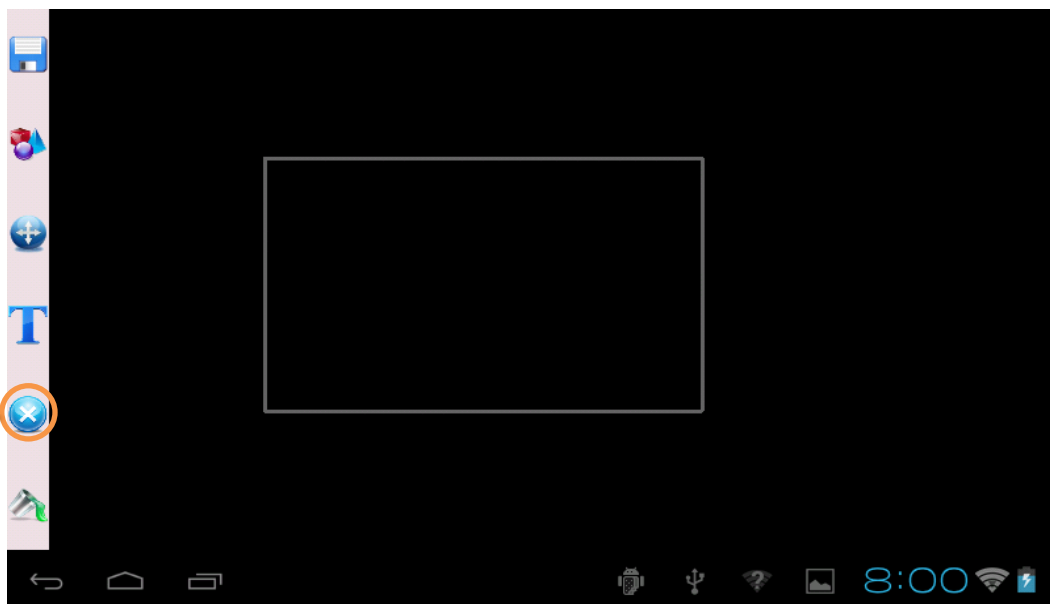


Fig 4.25.1. To delete the shape (i.e. rectangle) drawn, the user first clicks on the delete button and then touches the shape to be deleted.



Fig 4.25.2.The shape/entity is now deleted.

9. Background colour:

To change the colour of the background, this option is used. On clicking this button, a menu appears which contains a list of colours the user can change the background to. The colours are Red, Blue, Green, Yellow and Black.



Fig 4.26.1. On clicking the Background colour button, a menu displaying the list of colours you can change the background to.

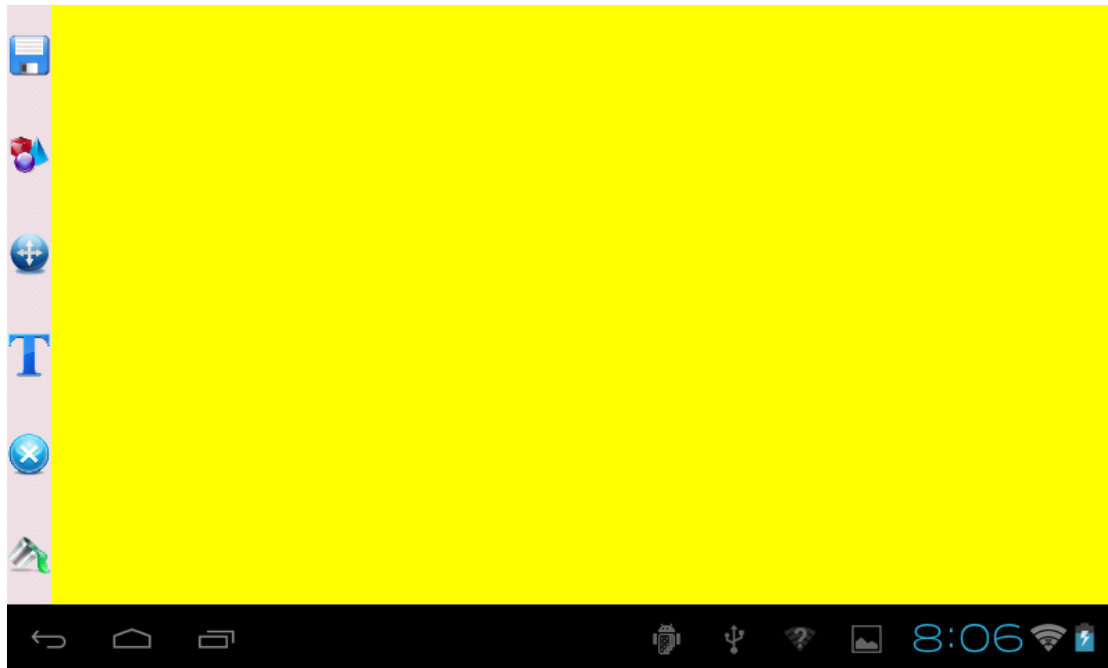


Fig 4.26.2.
Here, the user
has selected
the colour
yellow for the
background.

Summary and Conclusion

5.1. Summary:

A divide and conquer approach was taken for the completion of this application. The two parts (i.e. the Slate and the UML) of this application were worked upon separately. Firstly, a list of features was decided upon based on need and usage.

In the Slate part of the application, drawing, erasing and saving were the primary objectives. Improvising upon that the features like brush width, eraser width, brush colour, background colour, adding new pages and deleting pages were implemented. Furthermore, the drawing can be saved as either an image(png format) or a pdf.

In the UML part of the application, drawing shapes and connecting them was the primary objective. In addition, features like background colour, saving, adding text to the entities were implemented. This part of the application is still a prototype and requires further enhancement.

5.2. Future Enhancements:

- Finger tracing aided by optional follow-the-dot lines for each letter.
- Inserting of images for additional explanation.
- Voice prompt option reminds the young students which letter has been traced. This is primarily for children to learn the alphabet. The application reads out what the user has written.

References:

- [1] Android Developer Website:
<http://developer.android.com>
- [2] <http://som-itsolutions.blogspot.in/2010/12/freeware-android-paint.html>
- [3] <http://www.vogella.com/articles/Android/article.html>
- [4] <http://www.coreservlets.com/android-tutorial/>