IQRA UNIVERSITY IU

We aspired to design a mobile application that can provide people the opportunity to preview their rooms/walls/ceilings, etc., with a certain color before applying in reality .Through our application the user can click photos of their rooms/walls/ceilings, etc., and change their colors virtually and preview them.

Pre-Color Visualizer for Smart Phone

FINAL YEAR PROJECT REPORT

NAVEED
FARHAN-UL-HAQ KHAN

ABSTRACT:

Every so often people are confronted with the problem of deciding which colors to choose in the direction to paint their walls, doors and ceilings. They are profound to know how their rooms will look like after applying a certain color. Traditionally, there is only one technique to solve this problem, is to actually apply the paint which is an expensive and often irreversible solution.

To overcome this problem, we aspired to design a mobile application that can provide people the opportunity to preview their rooms/walls/ceilings, etc., with a certain color before applying in reality .Through our application the user can click photos of their rooms/walls/ceilings, etc., and change their colors virtually and preview them. This will contribute them a good estimate about the final look of their house.

This app will be convenient for many paint, home renovating professionals, and home owners.

DECLARATION:

This declaration is to clarify that all of the submitted contents of this project are original in its figure, excluding those, which have been admitted specifically in the references. All the work process involved is from our own idea and creativity. All contents of this project have been submitted as a part of partial fulfillment. I hereby declare that this project is the work of our own excluded for the references document and summaries that have been acknowledge.

Checked and Approved by

_____

Dr. Aarij Mahmood Hussaan

Project Supervisor

Faculty of Engineering and Technology

IQRA UNIVERSITY

April 2015

ACKNOWLEGEMENT:

First we would like to thank Allah Almighty who helps us to far-reaching this work which is one of his blessings on us and we like to express my gratitude and appreciation to all those who contributed in the possibility to complete this report.

A special thank and heart full gratitude to our final year project supervisor **Dr. Aarij Mahmood Hussaan** whose assistance, stimulating suggestions and encouragement, helped us to complete our project.

Special appreciations goes to my Co-Member, **Farhan-ul-haq Khan** who helps me to assemble the information and contributed his suggestion and his full effort in accomplishing the goal as well as his reinforcement to maintain our progress in track.

Deepest thanks and gratefulness to our parents, family, and others for their cooperation, inspiration, constructive suggestion and full of support for the report completion, from the beginning till the end.

Also thanks to all of our friends and everyone, those have been contributed by supporting our effort and help us all through the final year project advancement till it is fully completed.

MEMBERS:

NAVEED - 14921

FARHAN-UL-HAQ KHAN - 14815

Table of Contents

1. INTRODUCTION:

This project use the android platform to build an application where a customer can have prospect to preview their rooms/walls/ceilings, etc., with a certain color before applying in reality. This application allow you to upload a photo of the room you plan to paint, then apply different paint colors to the walls with a simple drag and drop action. Implementation of this model transpires by using android phone and desktop computer.

Color is the place where our brain and universe meet.

By: Paul Klee

Paint cooler ideas to unleash your creativity with a world of colors for your room or house.

1.1. SCOPE:

This App can be used by Home owners, Color shops, under construction buildings because of time-saving and productivity benefits. This apps work on android platform and you can run it on any android device.

It is also very effective for Paint industry for showing the customers the effects of their paint on the houses in real-time. This application can also be used as color visualizer.

1.2. OBJECTIVES AND GOALS:

- To allow homeowners to pre-visualize the effects of applying different colors on their walls, doors, ceilings, etc.
- To achieve the aforementioned goal using a mobile application objectives.
- To be able to identify different objects automatically using static images.
- To be able to change colors of the identified objects.
- To change colors keeping into account the light variations.

1.3. FEATURES:

Some of the features of this application are for user interface and easy to handle the application.

- Detection of different objects (walls, doors, ceilings, etc.) in an automatic manner.
- Ability to choose colors from a variety of different option and apply them to the detected objects.
- User can draw area and fill them with different colors.
- Ability to apply colors according to the light intensity for more realistic visualization.

1.4. OVERVIEW OF THE MODEL:

This is a visualizer plan or road map; it helps to determine how a person can use this application easily and effectively, which zone he desire to renovate just click a picture of it and segmented the zone according to the choice, formerly select a specific color for that area and decide after preview result. It's a good tool to help a customer to select a desirable result according to the thoughts and ideas.

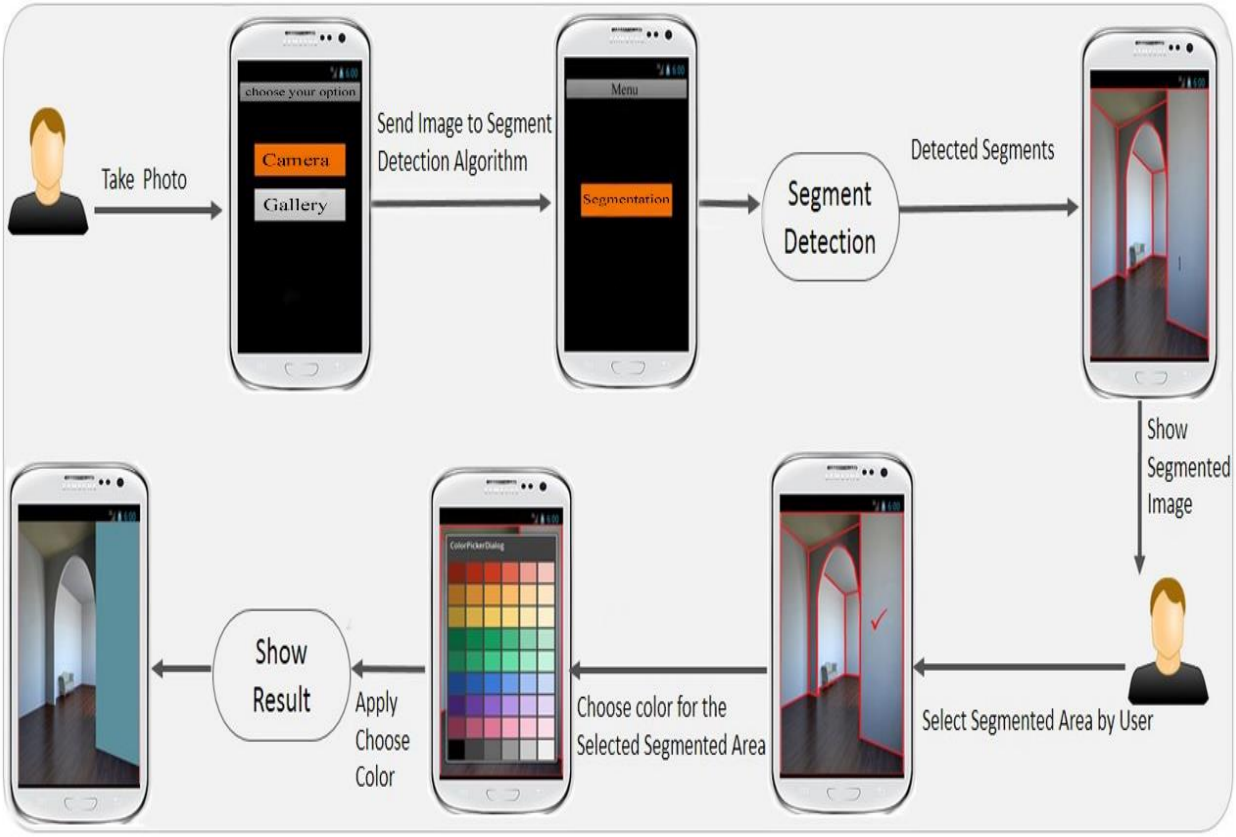


Figure 1: overview of the model

2. PROJECT DESCRIPTIONS:

2.1. ENLIGHTENMENT OF THE MODEL:

This representation and rules are for expressing the conceptual and implementation architectures of this application. Application software cannot run on itself but is dependent on system software to execute.

I. User will either take a photo from his/her smartphone or select an image from the phone's gallery.
II. The image is then sent to our detection engine
III. The detection engine will detect the different objects from the image
IV. The detected segments will be shown to the user
V. The user will select a segment and a color from the provided color pallet
VI. Finally, our app will apply the chosen color of the selected segment taking into consideration the light variations.
VII. And the user will have the same image but with the colors s/he wanted.

2.2. TECHNOLOGY USED:

Multiple technologies are used to develop this application and create its user interface and other functions of this application.

2.2.1. ECLIPSE (KEPLER 4.3)

Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse is used to develop applications. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system. Its runtime system is based on Equinox, an implementation of the OSGi core framework specification.

Eclipse implements the graphical control elements of the Java toolkit called SWT, whereas most Java applications use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's

user interface also uses an intermediate graphical user interface layer called JFace, which simplifies the construction of applications based on SWT.

2.2.2. JAVA SDK (SDK VERSION 8):

Java SDK is used to create and modify java programs. It is a development environment for building applications, applets, and components using the Java programming language.

It includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

2.2.3. ANDRIOD SDK(SDK 22.04)

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android Software Development Kit (SDK). It includes a comprehensive set of development tools. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

2.2.4. OPENCV 2.4.6:

OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

2.2.5. OPENCV (2.4.6) FOR ANDROID:

OpenCV4Android SDK package enables development of Android applications with use of OpenCV library.

2.2.6. ANDROID COLOR PIKER lIBRARY

This is a small library for your application to enable the users to select an arbitrary color. (See references for more details)

2.3. USER INTERFACE DESIGN:

User interface designs are illustrated by some screen shots for proper demonstration step by step.

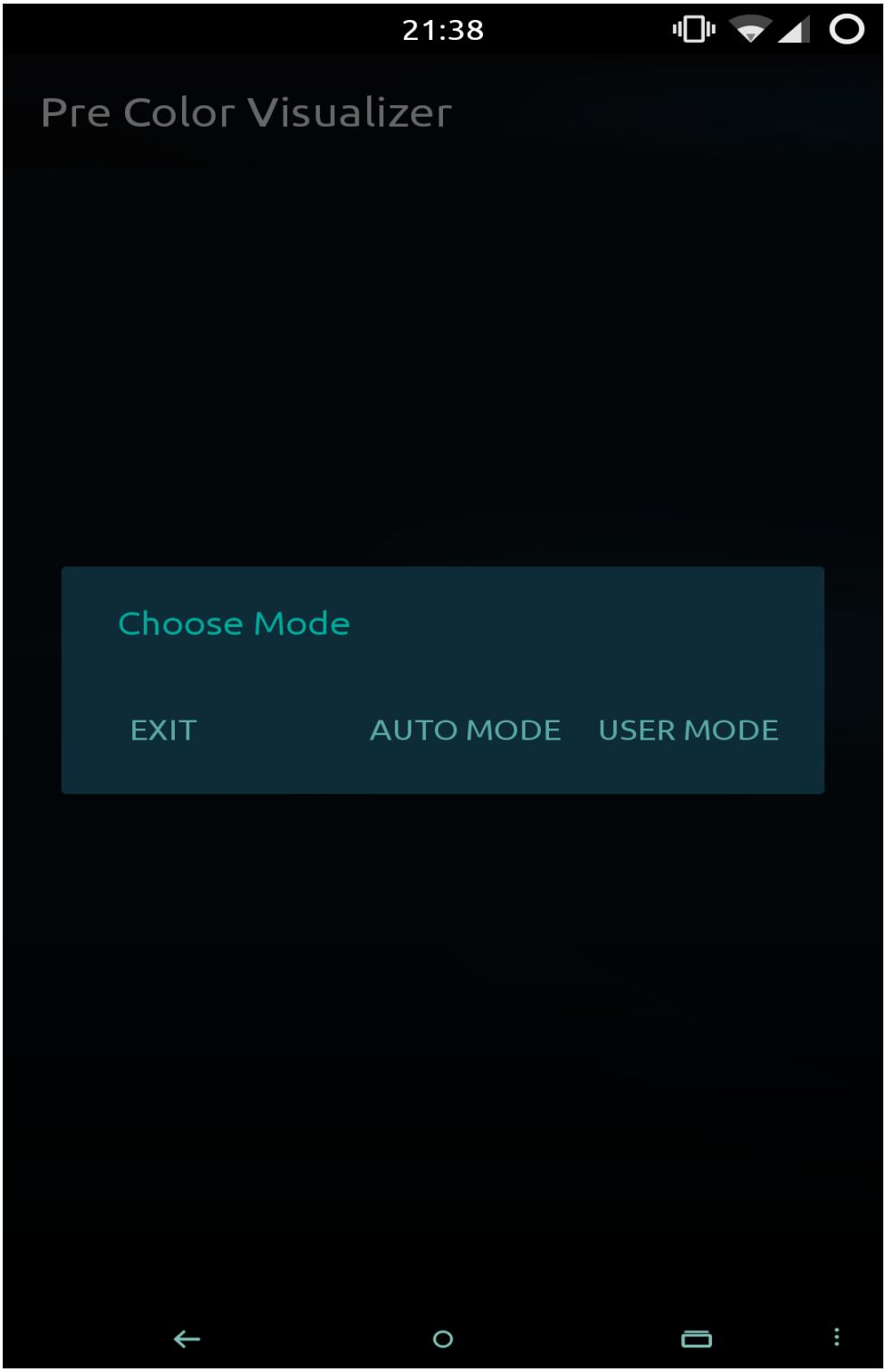


Figure 2: application mode selection screen

Figure 3: Auto mode screen

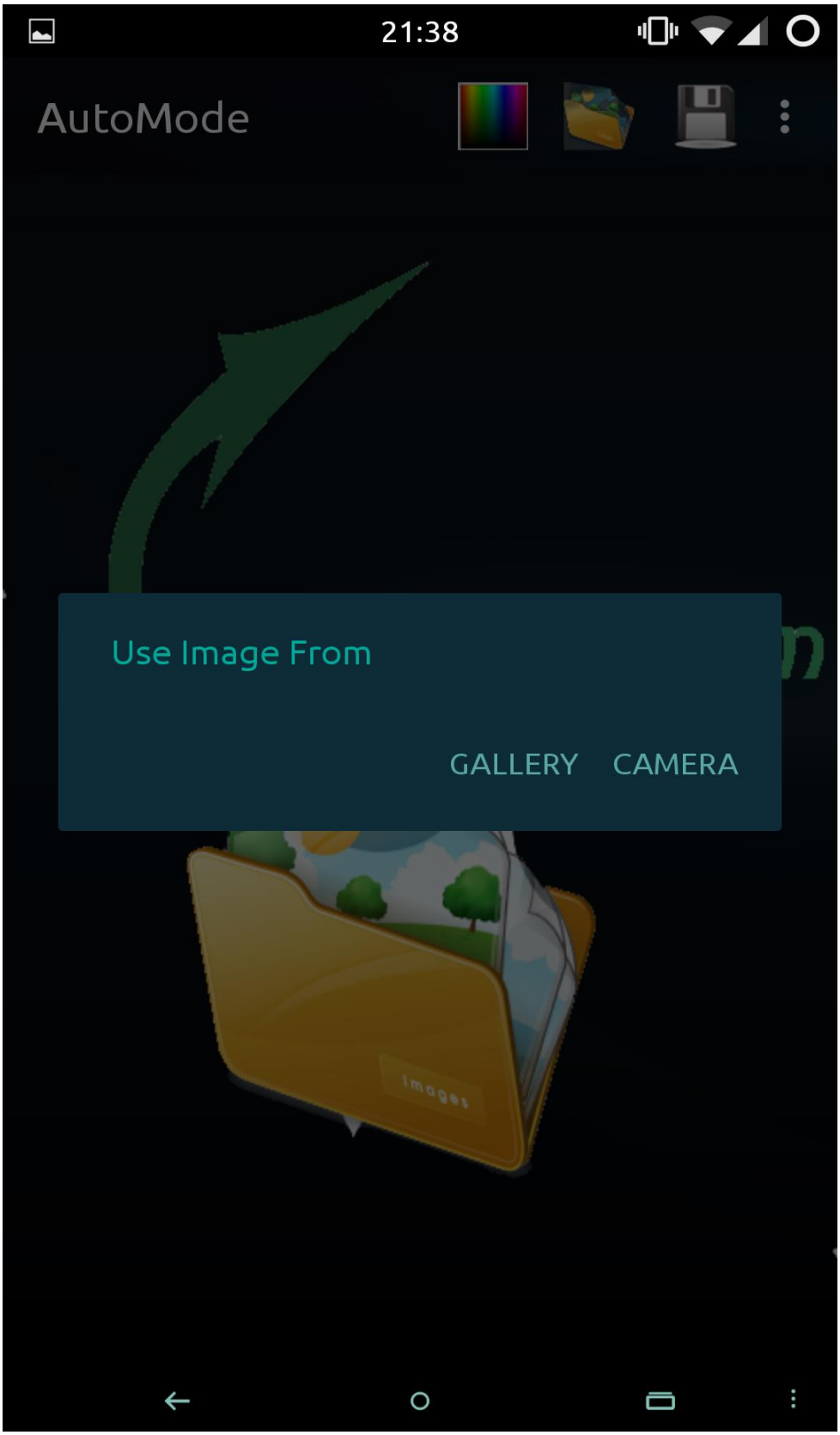


Figure 4: image source selection popup dialog

Figure 5: selected/captured image loaded and segmented

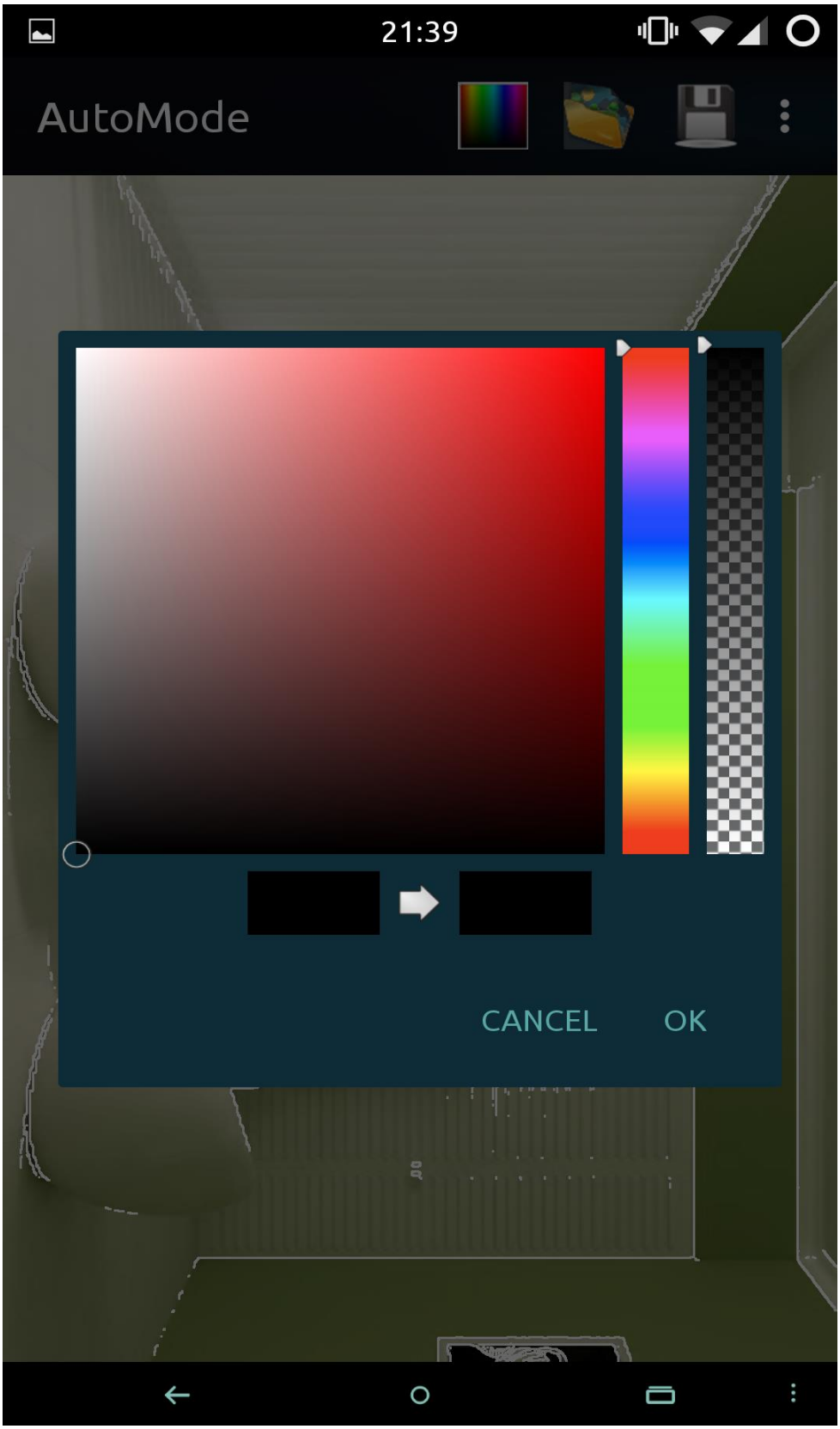


Figure 6: color chooser popup

Figure 7: another segmented image

Figure 8: color applied to a segment

Figure 9: color applied to ceiling segment

Figure 10: options menu, top right

Figure 11: after clearing segment edges

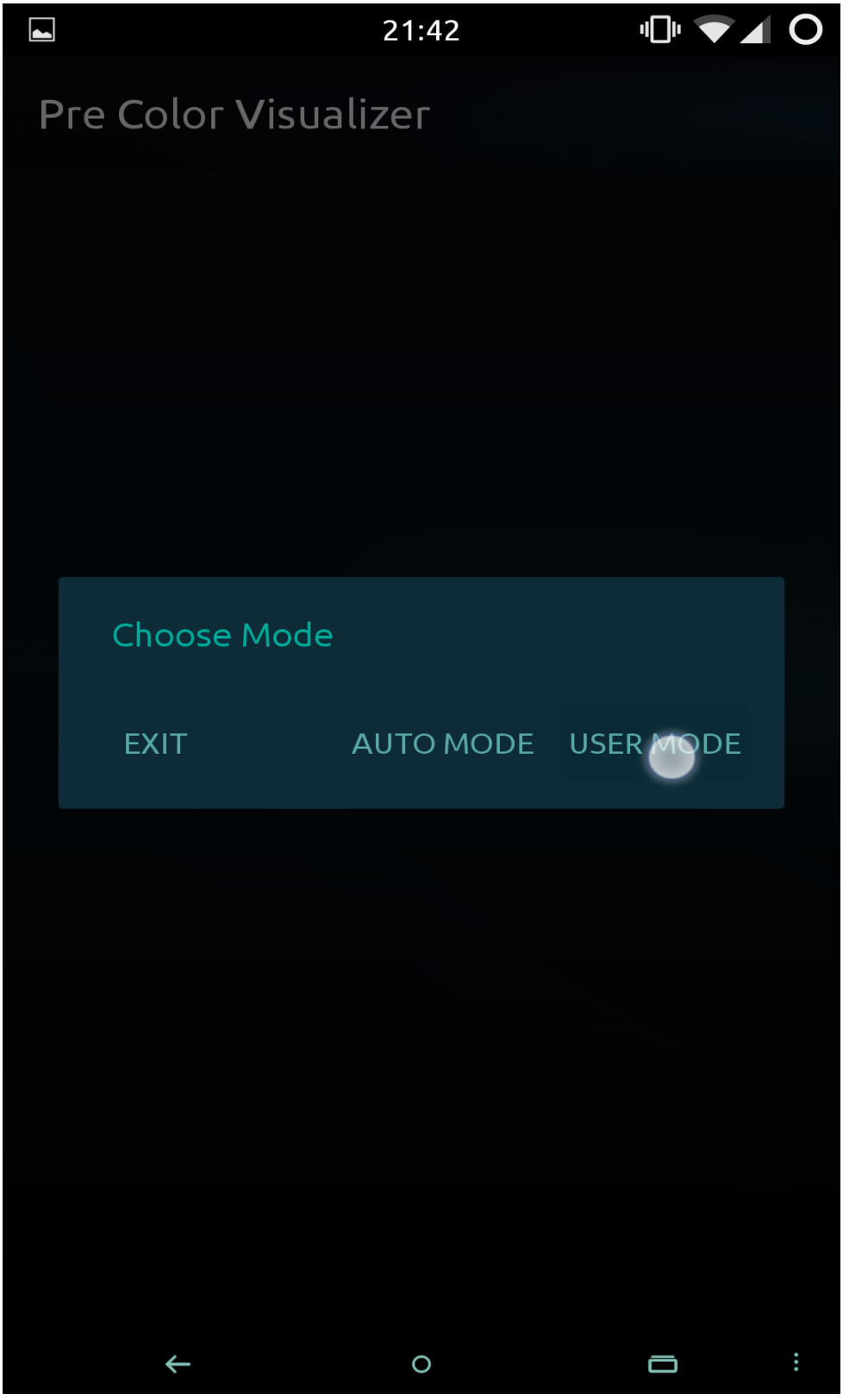


Figure 12: selecting user mode

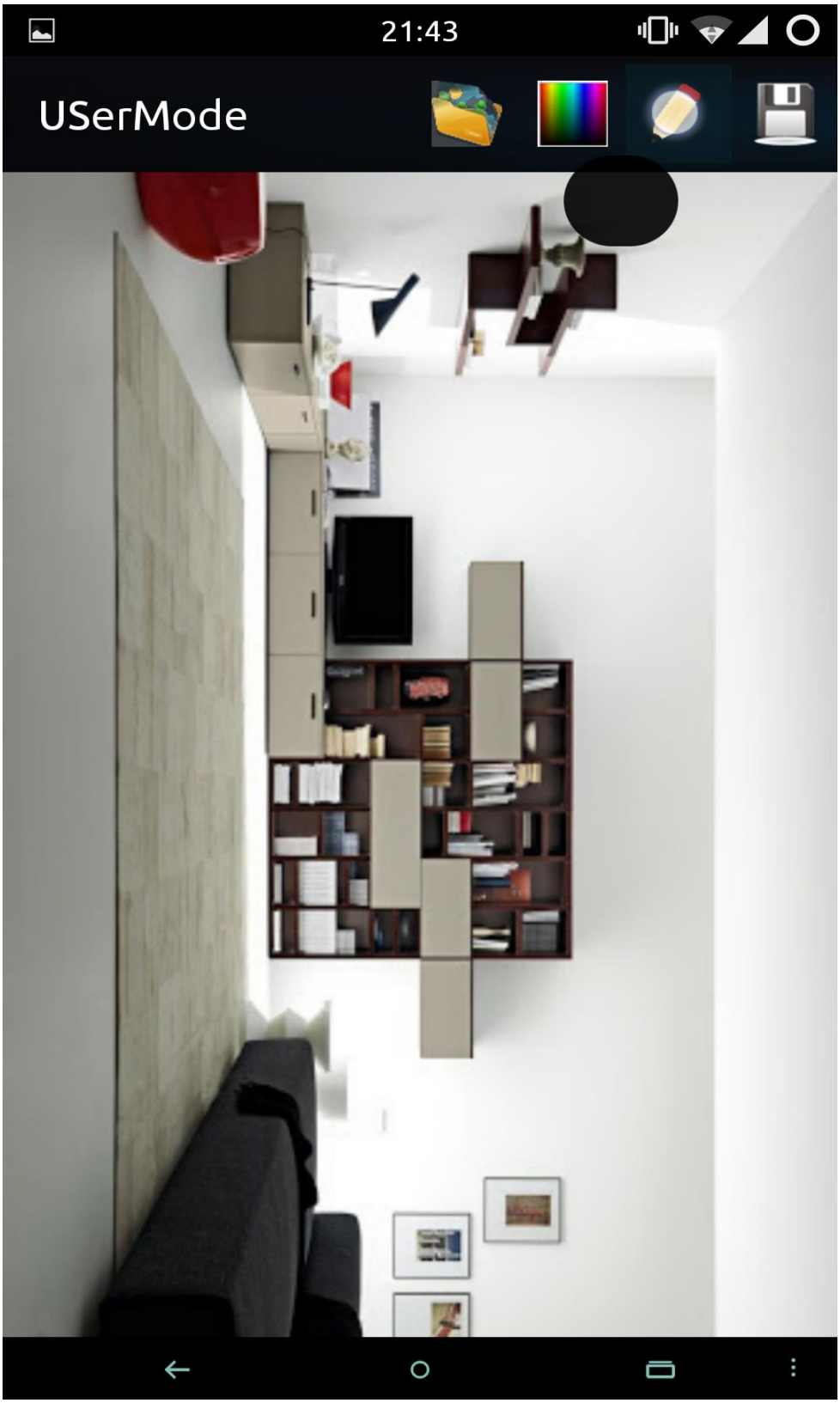


Figure 13: selecting brush to draw segment

Figure 14: drawn segment can be seen on the ceiling

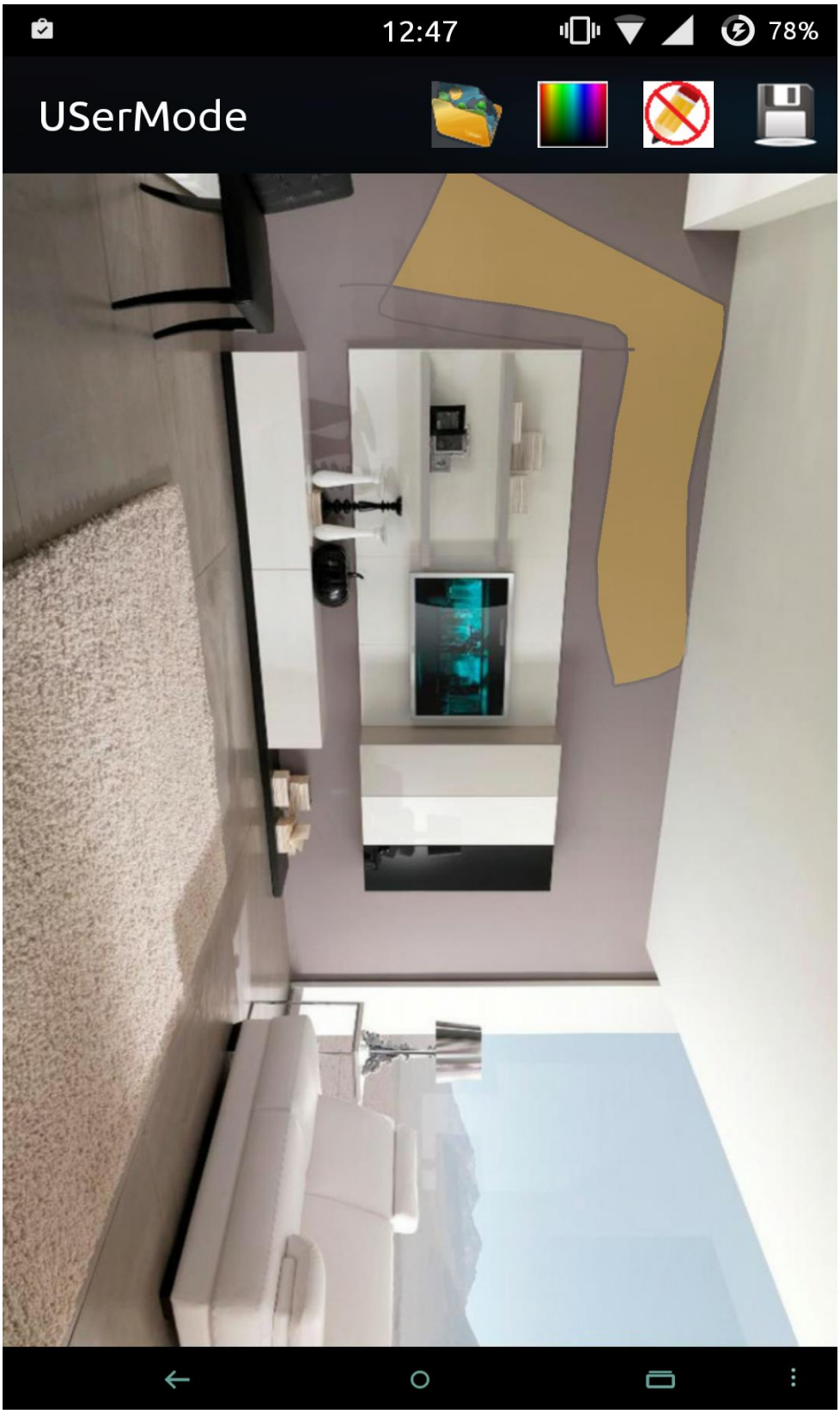


Figure 15: user drawn segment is filled with selected color

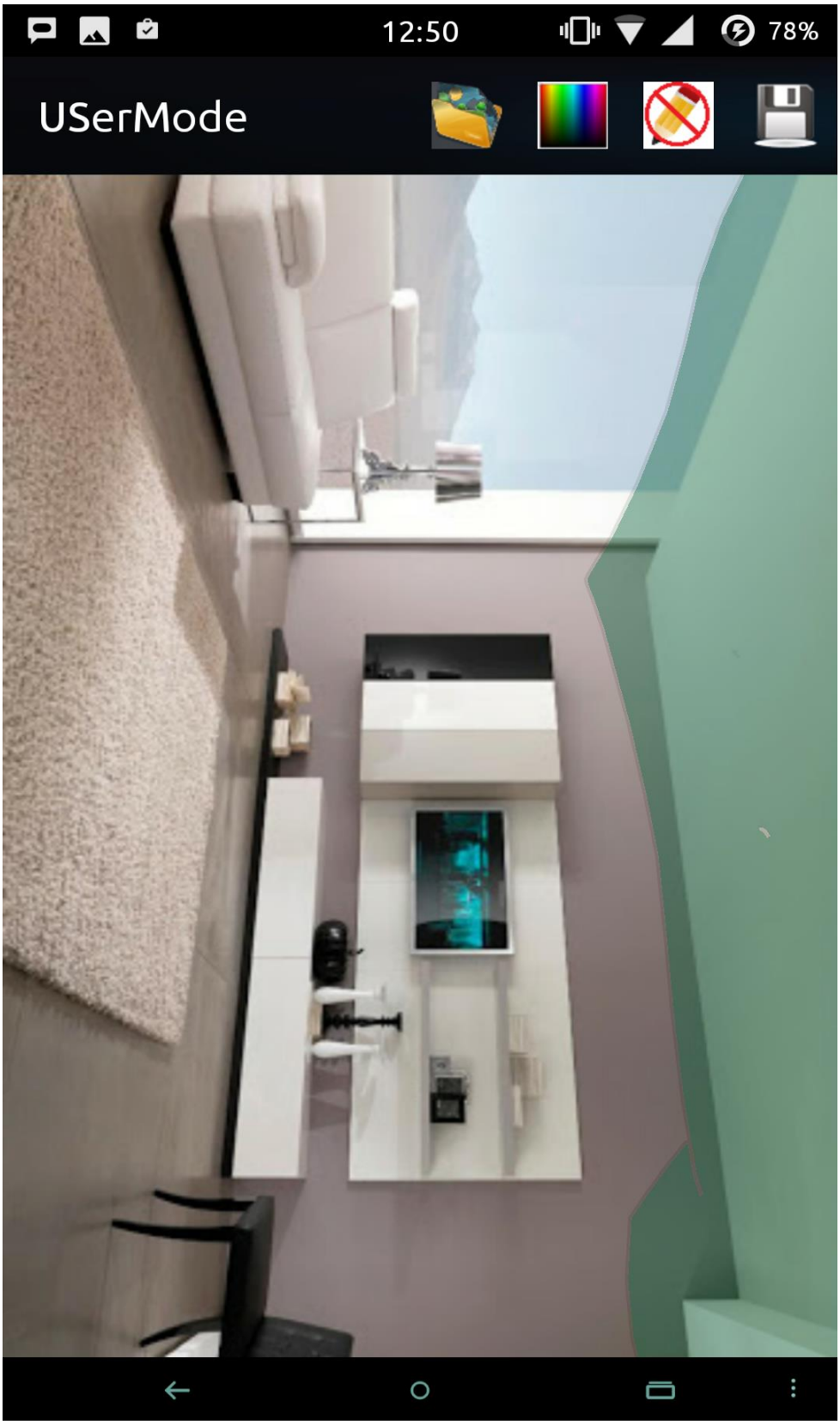


Figure 16: another user drawn segment example image

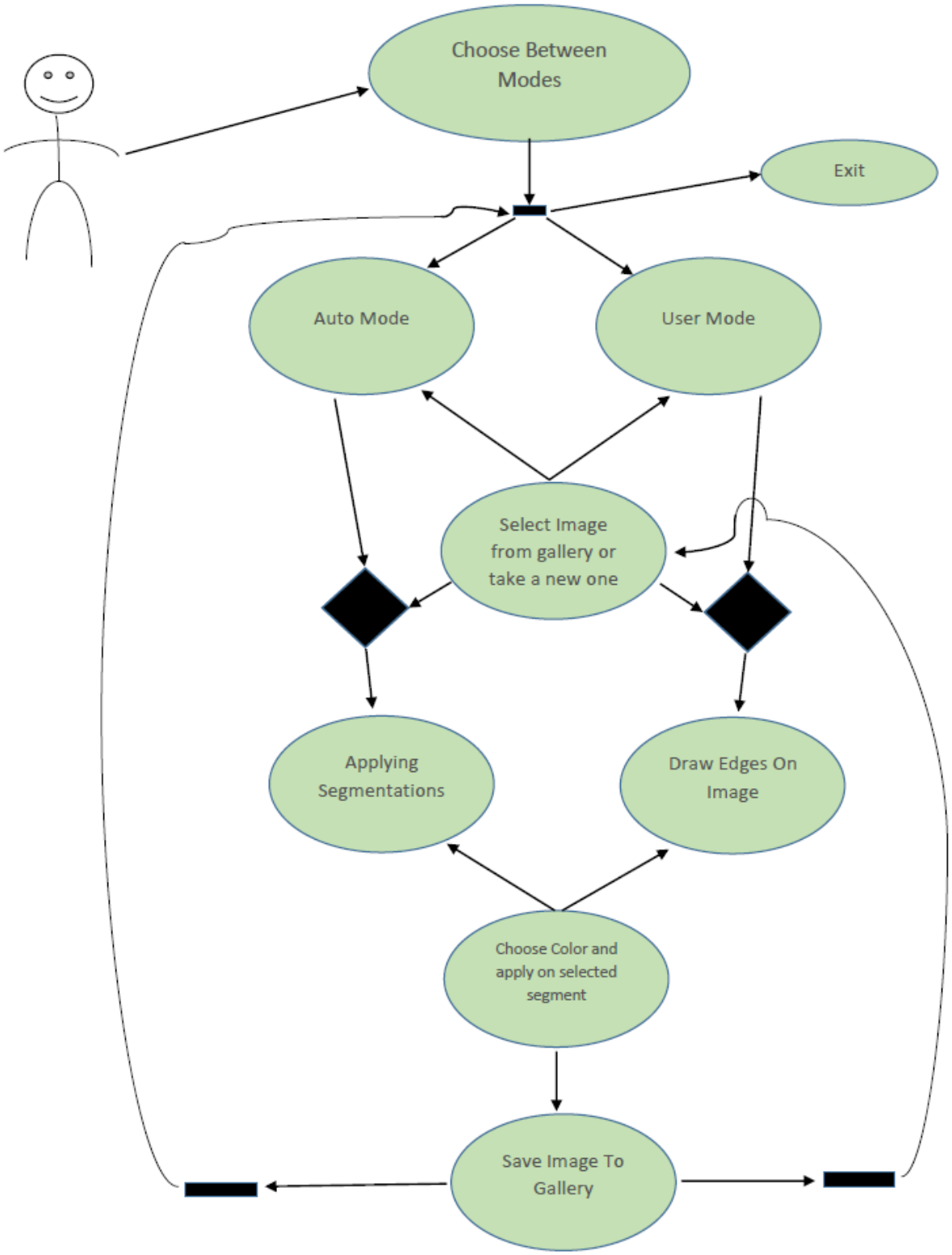
3. IMPLEMENTATION

3.1. DELIVERABLE ITEMS

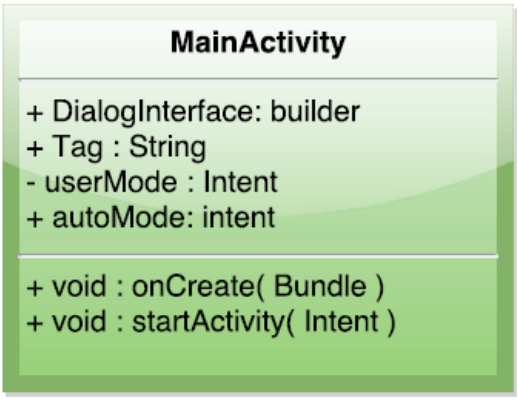
- Project Report
- Software Requirement Specification
- UML
  - Use Case Diagrams
  - Class Diagrams
  - Sequence Diagrams
  - Activity Diagram
- DFDs
- Software (Package file)

3.2. UML

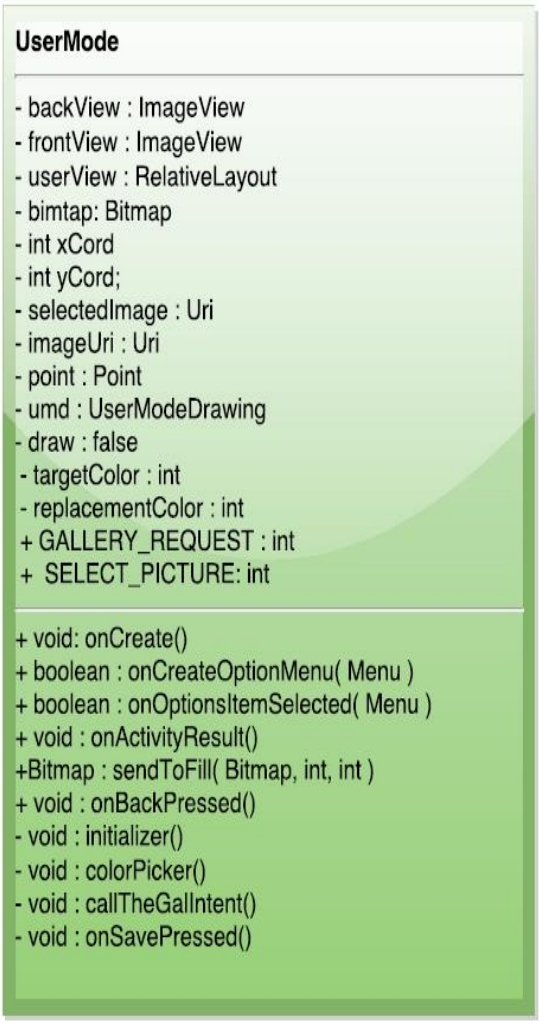
3.2.1. USE CASE



3.2.2. CLASS DIAGRAMS

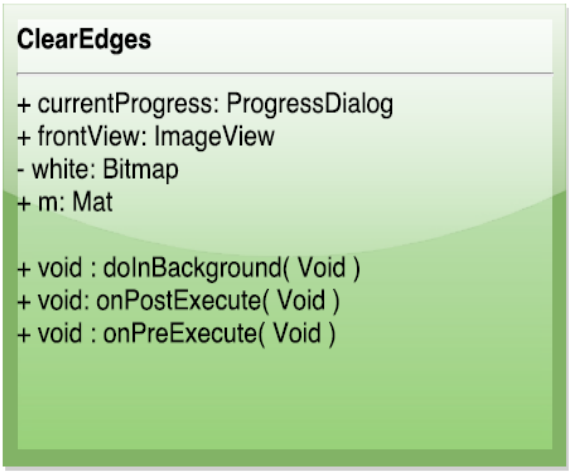

MainActivity
+ DialogInterface: builder
+ Tag : String
- userMode : Intent
+ autoMode: intent
+ void : onCreate(Bundle)
+ void : startActivity(Intent)

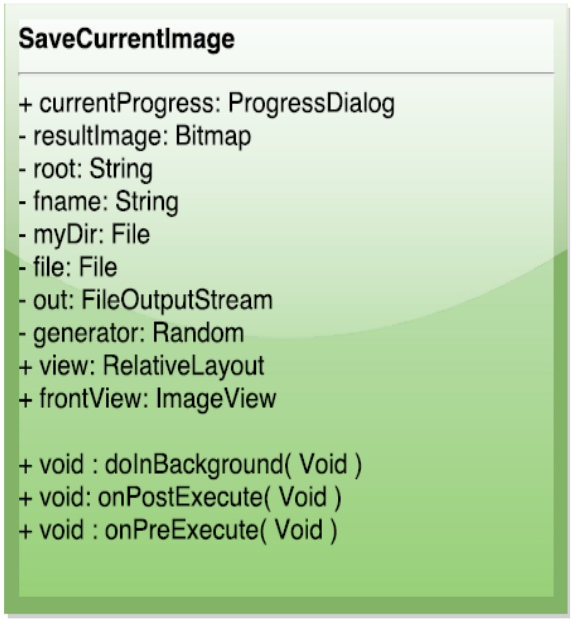

AutoMode
- backView : ImageView
- frontView : ImageView
- galleryBmp : Bitmap
- cannyBmp : Bitmap
- floodedBmp : Bitmap
- doneCanny : Bitmap
- replacementColor : int
+ GALLERY_REQUEST : int
+ SELECT_PICTURE: int
- canSaveImage: boolean
- canApplySegmentation: boolean
- canClearEdges : boolean
+ currentProgress: ProgressDialog
+ void : onCreate(Bundle)
+ void : onCreateOptionMenu(Menu)
+ boolean : onOptionsItemSelected(Menu)
+ boolean : onActivityResult(int requestCode, int resultCode, Intent data)
+Bitmap : sendToFill(Bitmap, int, int)
+ void: onBackPressed()
- void: initializer()
- Bitmap : applyLaplac(Bitmap)
- Bitmap : applyCanny(Bitmap)
- void : colorPicker()
- void : callTheGalIntent()
-void : onSavePressed()

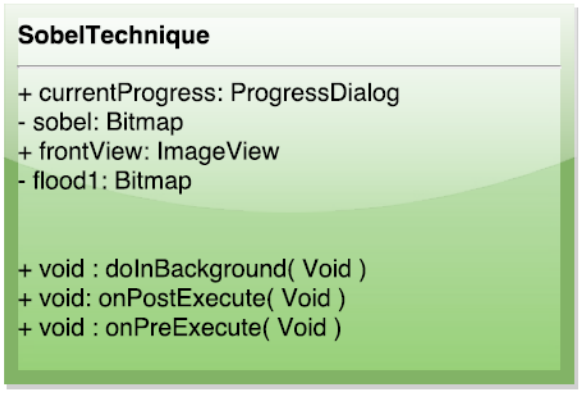

UserMode
- backView : ImageView
- frontView : ImageView
- userView : RelativeLayout
- bimtap: Bitmap
- int xCord
- int yCord;
- selectedImage : Uri
- imageUri : Uri
- point : Point
- umd : UserModeDrawing
- draw : false
- targetColor : int
- replacementColor : int
+ GALLERY_REQUEST : int
+ SELECT_PICTURE: int
+ void: onCreate()
+ boolean : onCreateOptionMenu(Menu)
+ boolean : onOptionsItemSelected(Menu)
+ void : onActivityResult()
+Bitmap : sendToFill(Bitmap, int, int)
+ void : onBackPressed()
- void : initializer()
- void : colorPicker()
- void : callTheGalIntent()
- void : onSavePressed()

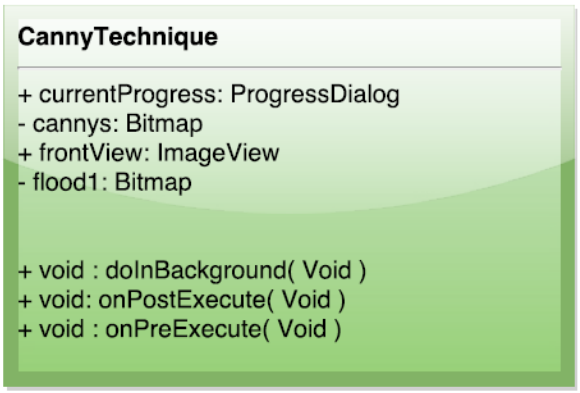

QueueLinearFloodFiller
+ image : Bitmap
+ tolerance : int
+ width : int
+ height : int
+ pixels : int
+ fillColor : int
+ startColor : int
+ pixelsChecked : boolean
+ ranges : Queue
+ void : setTargetColor(int)
+ int : getFillColor()
+ void : setFillColor(int)
+ int : getTolerance()
+ void : setTolerance(int)
+ Bitmap : getImage()
+ void : useImage(Bitmap)
+ void : prepare()
+ void : floodFill(int , int)
+ void : LinearFill(int , int)

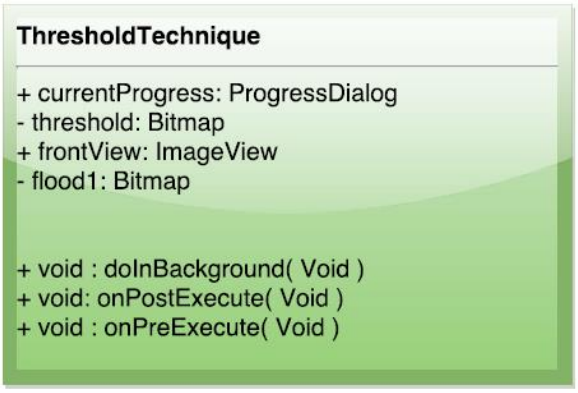

UserModeDrawing
- x : int
- y : int
- drawPaint : Paint
- path : Path
- bmp : Bitmap
- targetColor : int
- replacementColor : int
- canvas : Canvas
- point : Point
+ boolean : onTouchEvent()
+ void : onDraw()

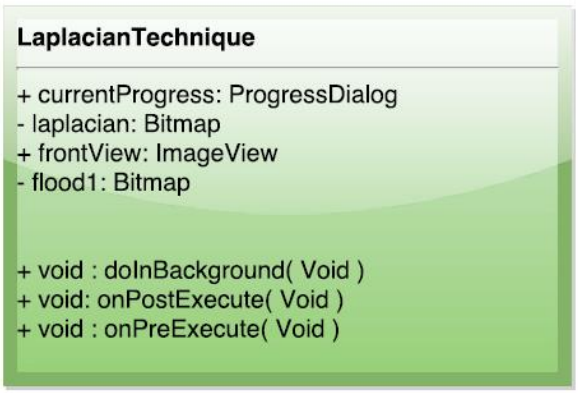

ClearEdges
+ currentProgress: ProgressDialog
+ frontView: ImageView
- white: Bitmap
+ m: Mat
+ void : doInBackground(Void)
+ void: onPostExecute(Void)
+ void : onPreExecute(Void)


SaveCurrentImage
+ currentProgress: ProgressDialog
- resultImage: Bitmap
- root: String
- fname: String
- myDir: File
- file: File
- out: FileOutputStream
- generator: Random
+ view: RelativeLayout
+ frontView: ImageView
+ void : doInBackground(Void)
+ void: onPostExecute(Void)
+ void : onPreExecute(Void)


SobelTechnique
+ currentProgress: ProgressDialog
- sobel: Bitmap
+ frontView: ImageView
- flood1: Bitmap
+ void : doInBackground(Void)
+ void: onPostExecute(Void)
+ void : onPreExecute(Void)

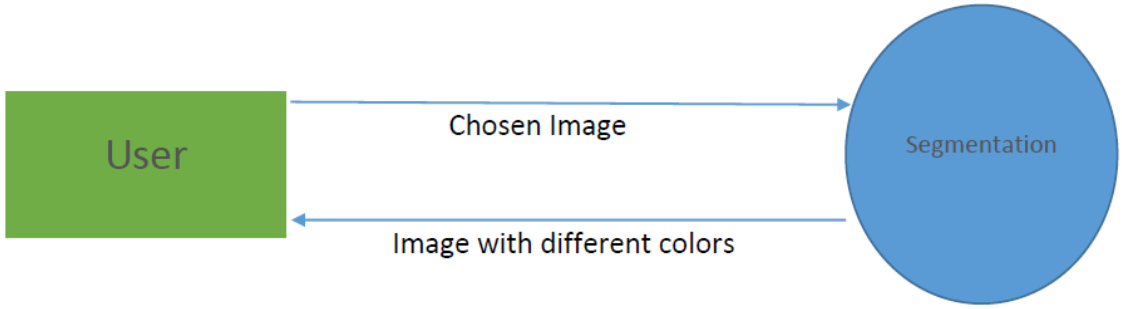

CannyTechnique
+ currentProgress: ProgressDialog
- cannys: Bitmap
+ frontView: ImageView
- flood1: Bitmap
+ void : doInBackground(Void)
+ void: onPostExecute(Void)
+ void : onPreExecute(Void)

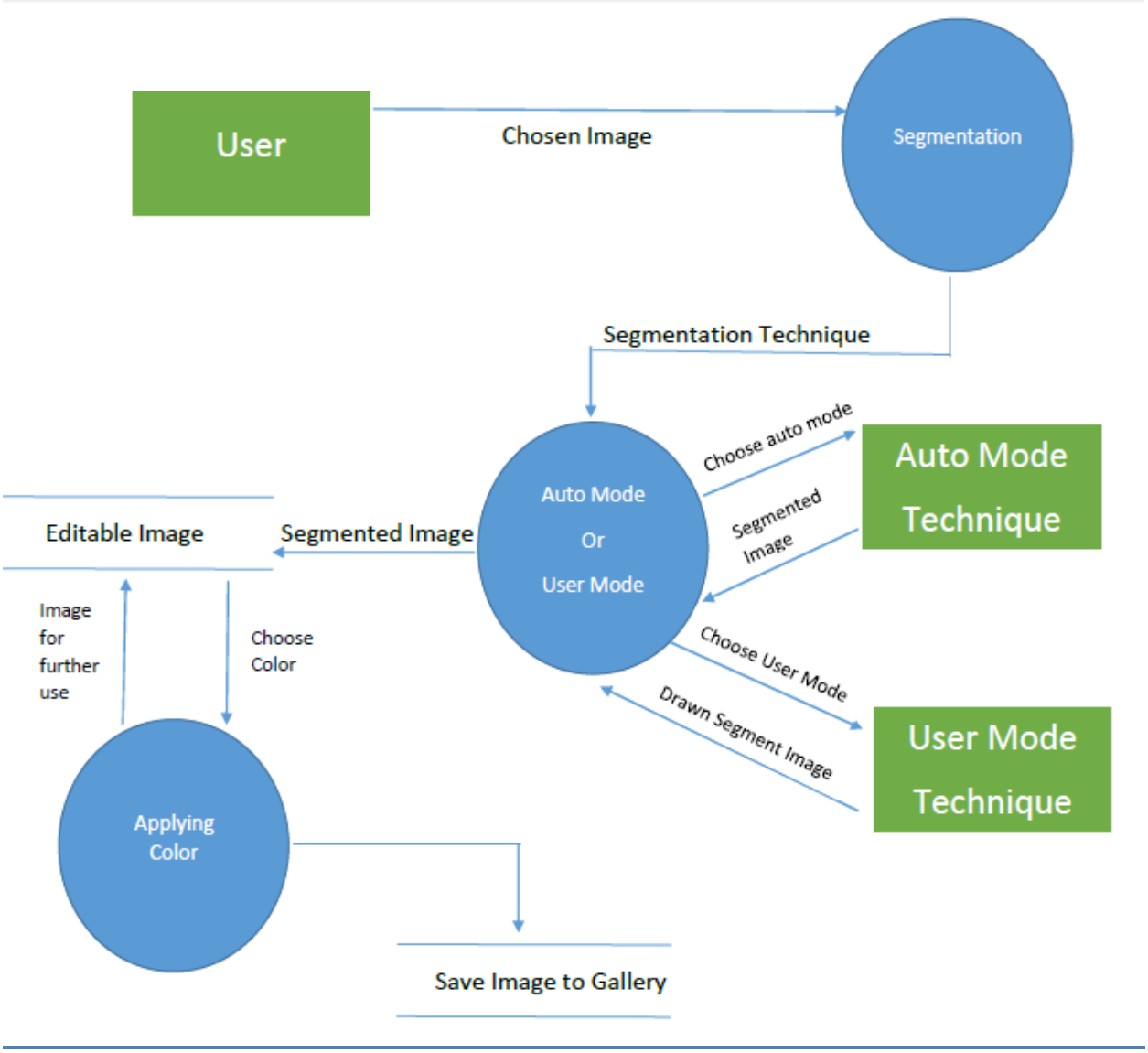

ThresholdTechnique
+ currentProgress: ProgressDialog
- threshold: Bitmap
+ frontView: ImageView
- flood1: Bitmap
+ void : doInBackground(Void)
+ void: onPostExecute(Void)
+ void : onPreExecute(Void)


LaplacianTechnique
+ currentProgress: ProgressDialog
- laplacian: Bitmap
+ frontView: ImageView
- flood1: Bitmap
+ void : doInBackground(Void)
+ void: onPostExecute(Void)
+ void : onPreExecute(Void)

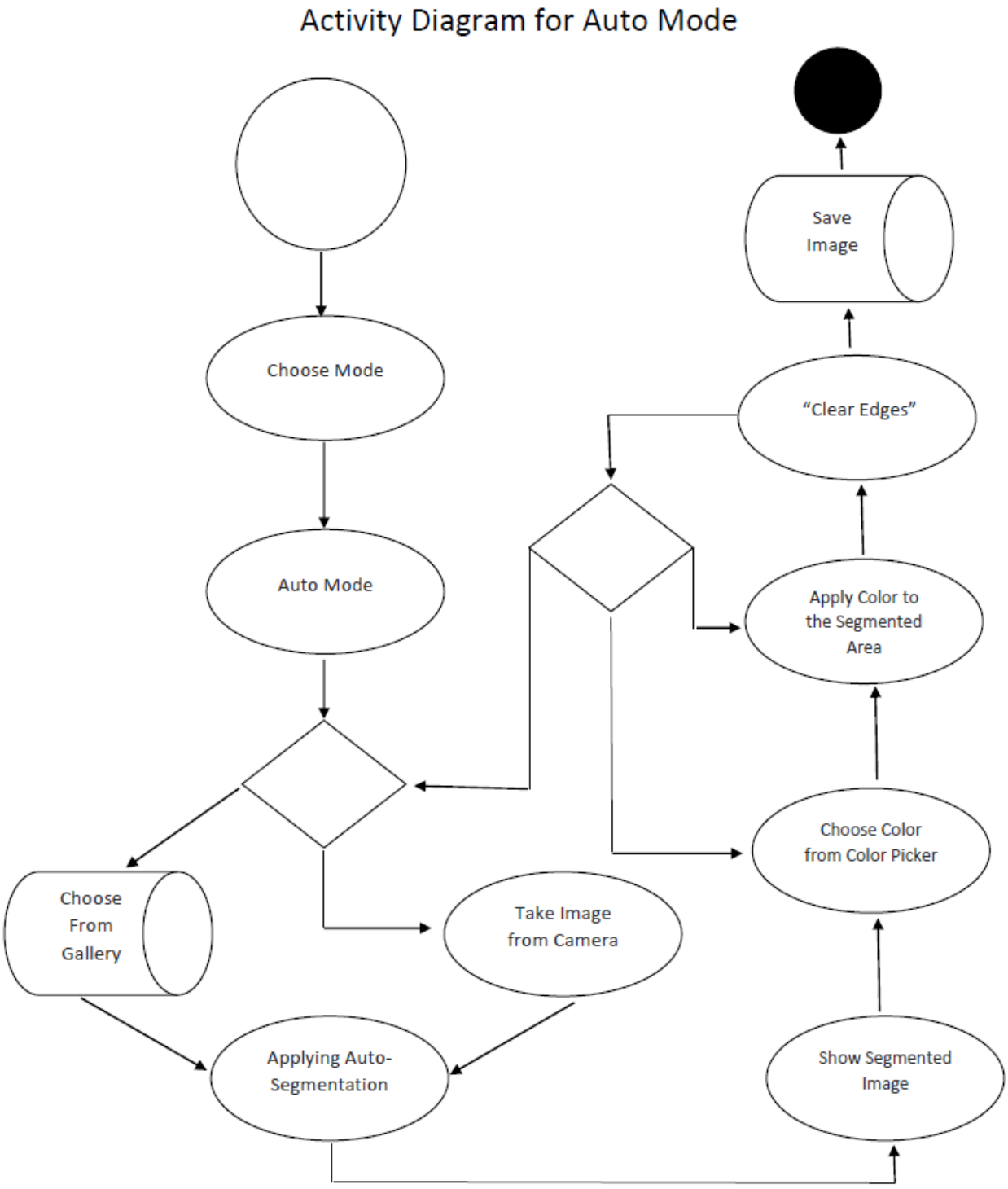
3.2.3. DATA FLOW DIAGRAMS

Level Zero Diagrams

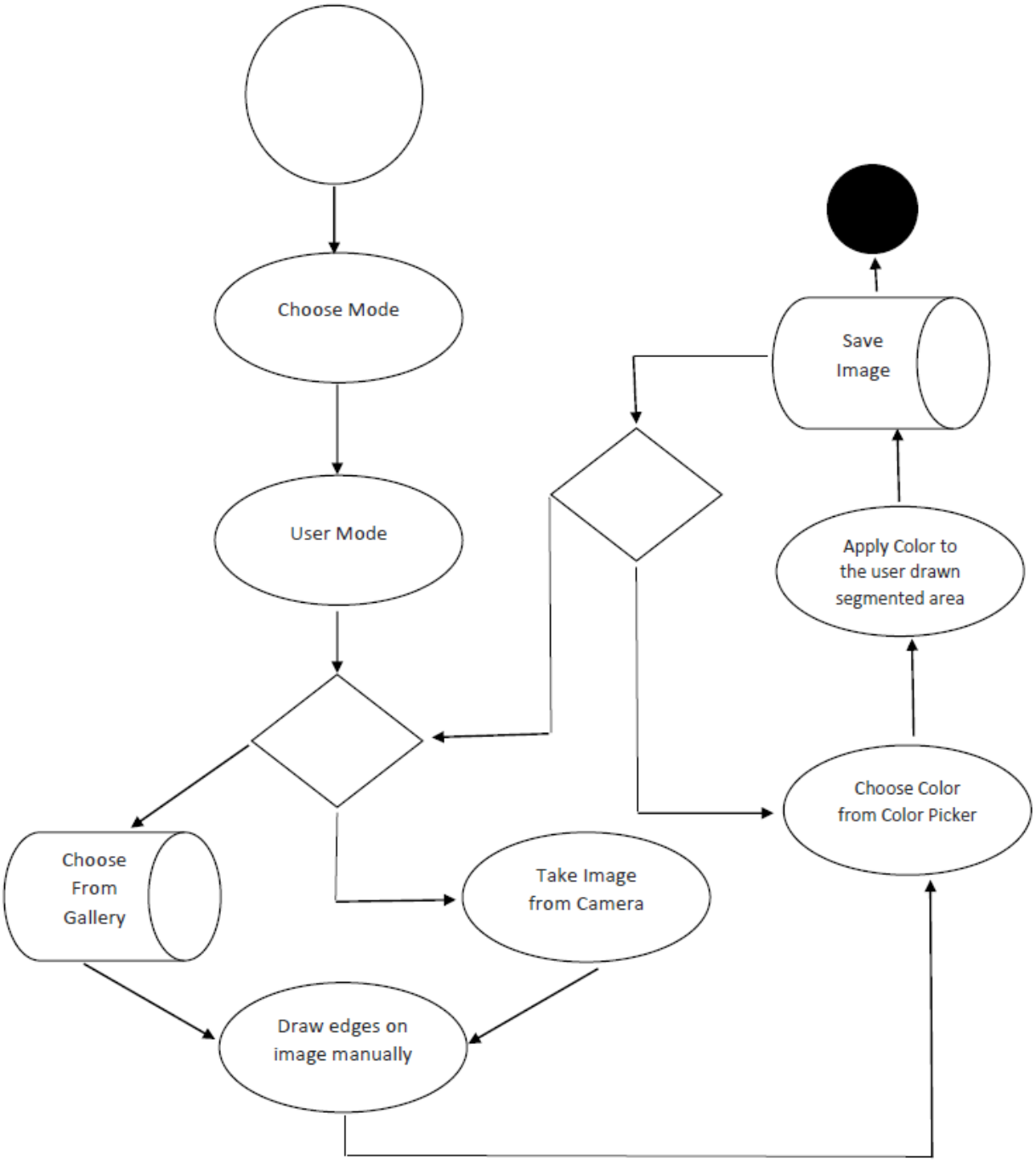



User
Chosen Image
Segmentation
Segmentation Technique
Choose auto mode
Auto Mode Technique
Segmented Image
Auto Mode Or User Mode
Editable Image
Segmented Image
Image for further use
Choose Color
Choose User Mode
Drawn Segment Image
User Mode Technique
Applying Color
Save Image to Gallery

3.2.4. ACTIVITY DIAGRAMS



Activity Diagram for User Mode



3.2.4. SEQUENCE DIAGRAMS

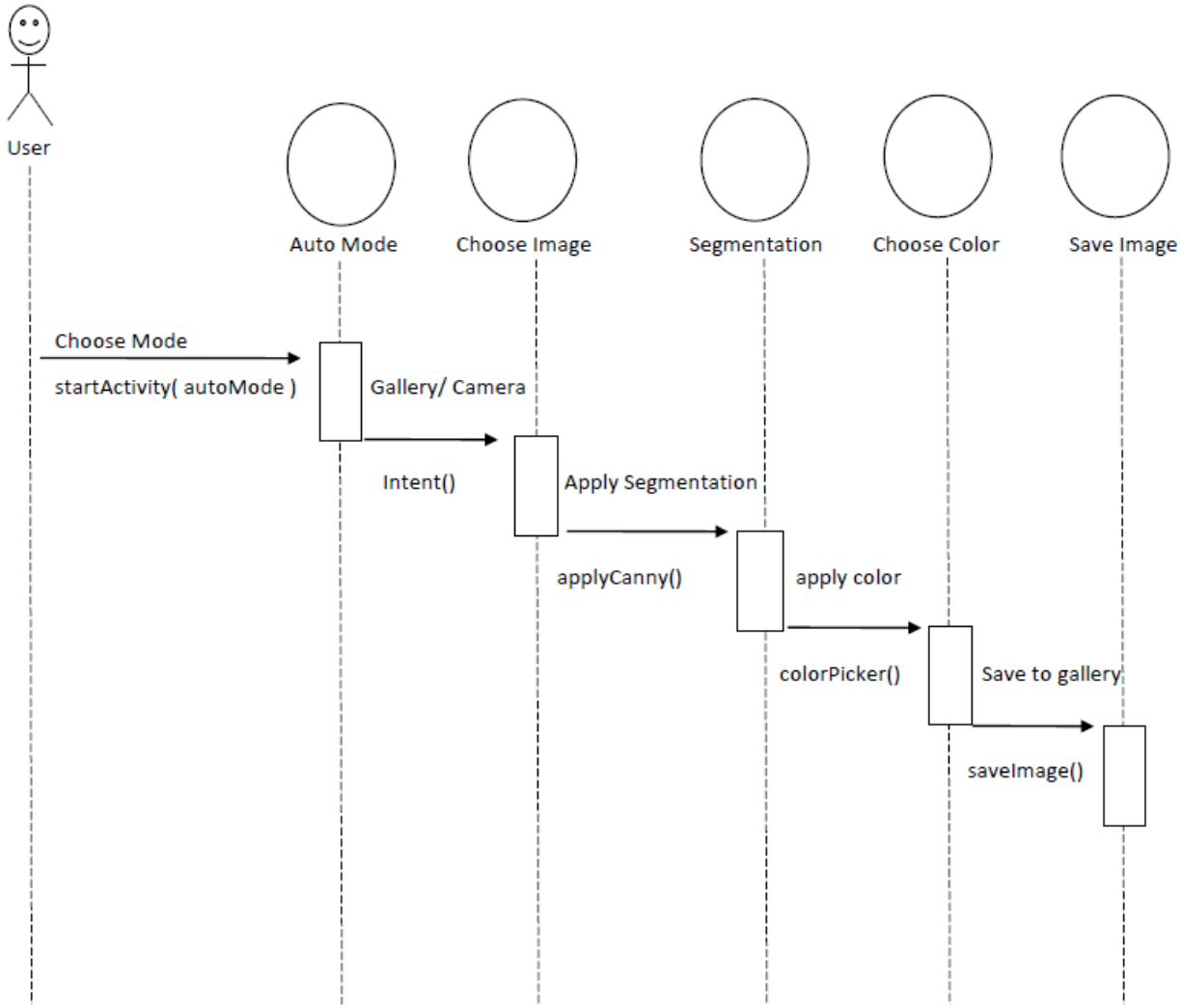


Figure 17: Auto Mode

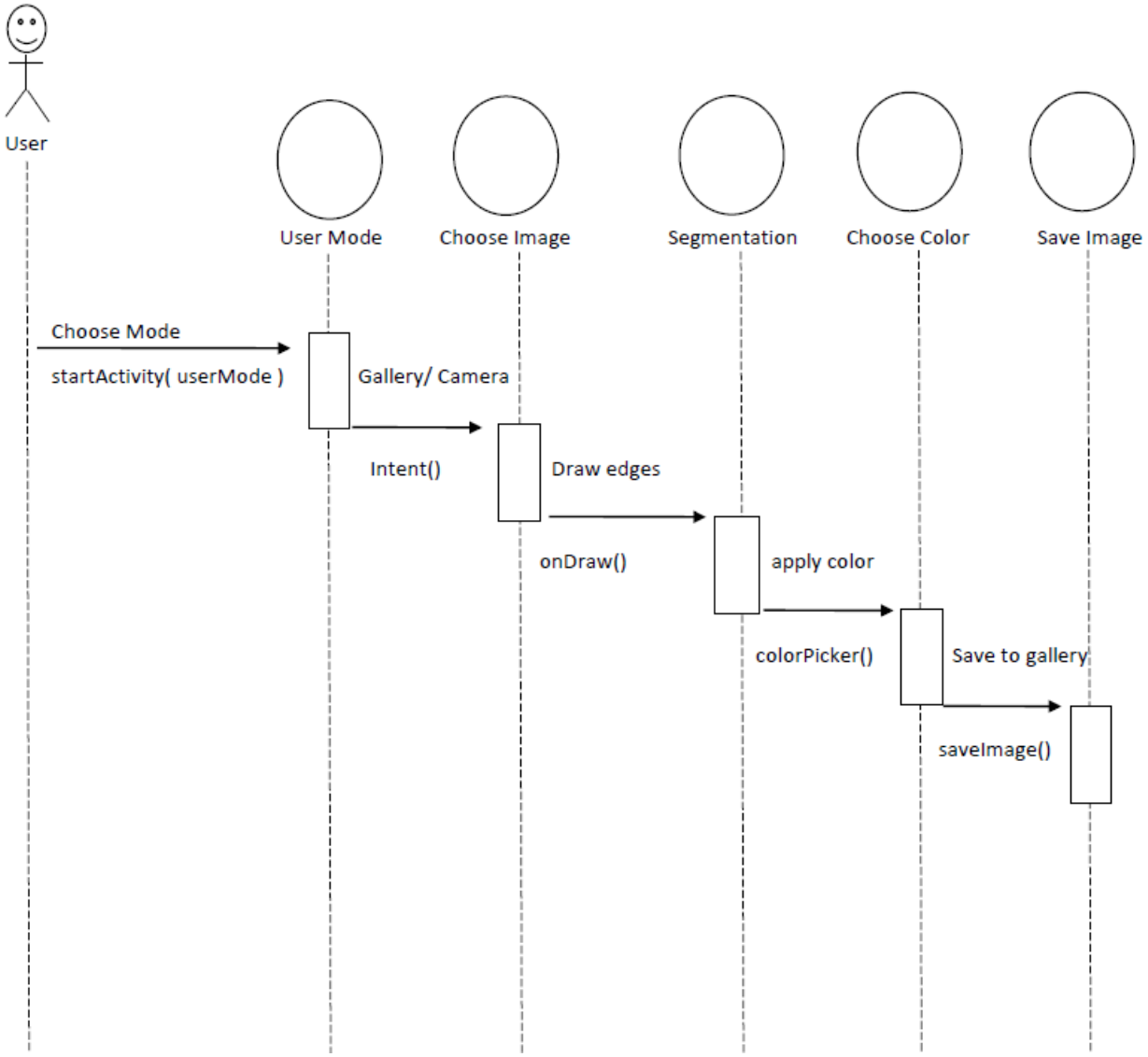


Figure 18: User Mode

3.3. MILESTONES

S. No.	Elapsed time since start of the project	Milestone
1.	Week 01 – 04	Proposal
2.	Week 05 – 08	Topic Research
3.	Week 09 – 12	Model
4.	Week 13 – 16	Model + Prototype
5.	Week 17 – 20	Prototype + Use Cases
6.	Week 21– 24	Implementation
7.	Week 25 – 28	Implementation
8.	Week 29 – 32	Testing
9.	Week 37 – 36	Deployment
10.	Week 37 – 40	

4. EQUIPMENT VITAL FOR PROJECT:

Some of these equipment's are vital for the project such as

- Android Mobile
- 1 Desktop Computer
- Core i5
- RAM 8GB
- 250 GB Hard Drive

5. CONCLUSION:

It is very effective for Paint industry for showing the customers the effects of their paint on the houses in real-time. It is also very useful to the any person confronted with the problem of deciding which colors to choose in the direction to paint their walls, doors and ceilings. .Through our application the user can click photos of their rooms/walls/ceilings, etc., and change their colors virtually and preview them. This will contribute them a good estimate about the final look of their house.

Some other benefits of the project:

- Pre-visualization of your painted walls, ceils etc.
- This tool will be helpful for pre planning of a room.
- An indispensable tool for construction workers, engineers, architects, carpenters, real estate agents, auction sellers, ...

6. REFERENCES:

1. https://github.com/yukuku/ambilwarna
2. http://opencv.org/platforms/android.html
3. https://developer.android.com/sdk/index.html