

AN INTERACTIVE MOBILE
MARKERLESS AUGMENTED REALITY
FOR INTERIOR DESIGN

NUR DZAQIRAH BINTI MOHD TARMIDI

Bachelor of Computer Science (Graphics &
Multimedia Technology) With Honors

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : NUR DZAQIRAH BINTI MOHD TARMIDI

Date of Birth : 2 APRIL 1996

Title : AN INTERACTIVE MOBILE MARKERLESS AUGMENTED REALITY FOR INTERIOR DESIGN

Academic Session : 2018/2019

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)

960402-11-5002

New IC/Passport Number
Date: 4th January 2019



(Supervisor's Signature)

Dr Danakorn Nincarean A/L Eh
Phon

Name of Supervisor
Date: 4th January 2019

AN INTERACTIVE MOBILE MARKERLESS AUGMENTED REALITY FOR
INTERIOR DESIGN

NUR DZAQIRAH BINTI MOHD TARMIDI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Graphics & Multimedia Technology) with Honors

Faculty Computer System & Software Engineering


UNIVERSITI MALAYSIA PAHANG

JANUARY 2019



SUPERVISOR'S DECLARATION

I hereby declare that I have read this project and in my opinion this project is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Graphics & Multimedia Technology).

Signature : 

Supervisor's Name : DR. DANAKORN NINCAREAN A/L EH PHON


Position : SENIOR LECTURER

Date : 4th JANUARY 2019



STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged.

Signature : 
Name : NUR DZAQIRAH BINTI MOHD TARMIDI
ID Number : CD15040
Date : 4th JANUARY 2019

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to successful the completion of this project. The special Thank goes to helpful supervisor, Dr Danakorn Nincarean A/L Eh Phon whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the project. The supervision and support that he gave truly help the progression and smoothness of the project. The co-operation is much indeed appreciated.

I am also grateful to my family and friends, for supporting and encouraging me to complete this project successfully. Finally, I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one.

ABSTRAK

“Markerless” ialah satu teknik dalam sistem “Augmented Reality”. Pengguna tidak perlu menggunakan sebarang penanda untuk menghasilkan aplikasi interaktif dengan model 3D. “An Interactive Mobile Markerless Augmented Reality for Interior Design” ialah satu aplikasi yang membenarkan pengguna untuk mendekorasi bilik mereka secara maya tanpa menggunakan sebarang penanda dengan teknologi pengesanan canggih dan model 3D. Model 3D di ambil dari pangkalan data yang sedia ada. Ia memudahkan pengguna untuk mendekorasi bilik kerana model 3D dan persekitaran yang hampir sama dengan kehidupan yang sebenar. Pengguna bebas untuk mendekorasi bilik berdasarkan 72 model perabot dengan pelbagai variasi warna dan reka bentuk. Masalah yang selalu di hadapi oleh pengguna ialah mereka sibuk dengan kerja dan tidak mempunyai masa untuk pergi ke pelbagai kedai bagi membeli perabot untuk kediaman mereka. Bila mereka mempunyai masa untuk membeli belah, ia menyukarkan bagi mereka untuk mengenal pasti bagaimana sesetengah meja atau kerusi akan kelihatan di dalam bilik sebelum ia didekorasi kerana tidak dapat mengukur saiz yang sepatutnya produk itu akan sesuai dengan persekitaran bilik mereka. Ia agak sukar untuk memenuhi keperluan pengguna untuk menghiasi bilik mereka tanpa gambar yang sebenar. Semua masalah yang pengguna hadapi boleh di selesaikan dengan menggunakan aplikasi ini. Objektif projek ini ialah untuk menghasilkan sebuah aplikasi telefon “markerless Augmented Reality” untuk hiasan dalaman. “Rapid Application Development” di pilih sebagai metodologi untuk projek ini bagi memastikan pembinaan projek ini berjaya. Fasa dalam model RAD merangkumi “Requirement Planning, User Design, Construction,” dan “Cutover”. Aplikasi ini telah dinilai oleh sasaran penonton untuk mengenalpasti keberkesanannya dan persepsi terhadap aplikasi tersebut menggunakan “User Acceptance Test”. Penggunaan “Augmented Reality” dalam aplikasi ini membuktikan bahawa ia memudahkan tugas pengguna untuk mendekorasi bilik mereka.

ABSTRACT

Markerless is one of the techniques in the Augmented Reality system. User does not need to use any marker to produce interactive apps with the 3D model. An Interactive Mobile Markerless Augmented Reality for Interior Design is an application that allows user to decorate their room virtually without using any marker with advance tracking technology and 3D models. 3D models are load from the existing database. It can make user's task easier to decorate the room because the 3D models and environment is almost same as real life. User can freely decorate the room based on 72 models of furniture with various colour and design. Problems that user generally faced by the people who are busy with their work and have no time to go to various stores to buy furniture for their home. And when they have time for shopping, it is difficult for them to visualize how a particular table or chair will look in a room before it is decorated because could not check for the proper size of the product that fits their room environment. There is difficult to fulfil the customer's need of decorating their room without imaginary picture to refer. All the problem that user faced can be solve by using this application. The objective of this project is to develop a mobile markerless Augmented Reality application for interior design. Rapid Application Development was chosen as a methodology for this project to ensure the success of the project development. The phases in RAD model consist of Requirements Planning, User Design, Construction, and Cutover. The application was evaluated by the target audience to investigate its effectiveness and user perception towards the application using the User Acceptance Test. The use of the Augmented Reality in this application proves that it can make user's task easier in order to decorate their room.

TABLE OF CONTENTS

CONTENT	Page
SUPERVISOR'S DECLARATION	ii
STUDENT'S DECLARATION	iii
ACKNOWLEDGEMENTS	iv
ABSTRAK	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	xi
LIST OF FIGURES	xiii
LIST OF ABBREVIATIONS	xvii
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scope	4
1.5 Thesis Organization	5
CHAPTER 2 LITERATURE REVIEW	6
2.1 Introduction	6

2.2	Augmented Reality	6
2.3	Types of Augmented Reality	7
2.4	Review of Existing System	8
2.4.1	Application 1 – Room Remix	8
2.4.2	Application 2 – Roomle 3D/AR Furniture Catalog	11
2.4.3	Application 3 – IKEA Catalogue	14
2.5	Comparison of Three Existing Systems	17
2.6	Summary	21
CHAPTER 3 METHODOLOGY		23
3.1	Introduction	23
3.2	Types of Software Development Life Cycle (SDLC)	23
3.2.1	Waterfall Model	24
3.2.2	ADDIE Model	25
3.2.3	Rapid Application Development (RAD)	26
3.2.4	Iterative Model	28
3.2.5	Comparison between SDLC Model	29
3.3	Chosen Methodology	32
3.3.1	Requirements Planning	33
3.3.2	User Design	39
3.3.3	Construction	39
3.3.4	Cutover	40
3.4	Work Breakdown Structure	41
3.5	Flowchart	41

3.6	Context Diagram	45
3.7	Use Case Diagram	45
3.8	Storyboard	47
3.8.1	First Interface	47
3.8.2	Second Interface	47
3.8.3	Third Interface	48
3.8.4	Fourth Interface	49
3.8.5	Fifth Interface	50
3.8.6	Sixth Interface	51
3.9	Hardware and Software Requirement	51
3.9.1	Software Requirement	51
3.9.2	Hardware Requirement	52
3.10	Gantt Chart	54
3.11	Implementation	55

CHAPTER 4 IMPLEMENTATION, TESTING, RESULTS AND DISCUSSION56

4.1	Introduction	56
4.1.1	Development Tools	56
4.2	Implementation	57
4.2.1	The 3D Models	58
4.2.2	Application Development	67
4.2.3	Coding	74
4.3	Testing and Result Discussion	113
4.3.1	Result Discussion	113

4.3.2	Testing Report	118
4.4	User Manual	118
4.5	Usability Test	118
4.5.1	Test Features	118
4.5.2	Result Report	124
CHAPTER 5 CONCLUSION		125
5.1	Introduction	125
5.2	Research Constraint	126
5.3	Future Work	127
REFERENCES		128
APPENDIX A		131
APPENDIX B		137
APPENDIX C		147

LIST OF TABLES

Table 2.1	Summary of comparison the existing system	17
Table 3.1	Comparison between SDLC Model	30
Table 3.2	First Interface	47
Table 3.3	Second Interface	47
Table 3.4	Third Interface	48
Table 3.5	Fourth Interface	49
Table 3.6	Fifth Interface	50
Table 3.7	Sixth Interface	51
Table 3.8	Software Requirements	52
Table 3.9	Hardware Requirements	52
Table 4.1	DinoController.cs	74
Table 4.2	LoadScene.cs	76
Table 4.3	OptionPanel.cs	76
Table 4.4	ScreenshotPreview.cs	78
Table 4.5	TakeScreenShot.cs	81
Table 4.6	InstantTrackingController.cs	82
Table 4.7	MoveController.cs	90
Table 4.8	GridRenderer.cs	93
Table 4.9	LeanRotate.cs	97
Table 4.10	LeanTapSelect.cs	100
Table 4.11	LeanSelectable.cs	102
Table 4.12	LeanTranslate.cs	106

Table 4.13 StreamVideo.cs

109

LIST OF FIGURES

Figure 2.1	Marker-Based	7
Figure 2.2	Markerless-Based	7
Figure 2.3	Space of the room	9
Figure 2.4	Place the furniture	9
Figure 2.5	Purchase the furniture	9
Figure 2.6	Setting up the furniture	9
Figure 2.7	List of furniture	10
Figure 2.8	Saved scenes	10
Figure 2.9	Shopping Cart	10
Figure 2.10	Setting	10
Figure 2.11	Home	12
Figure 2.12	Browse by Topic	12
Figure 2.13	Mockup Furniture	12
Figure 2.14	All Brands	13
Figure 2.15	Branded furniture	13
Figure 2.16	Favourite	13
Figure 2.17	Scaling the furniture	13
Figure 2.18	Furniture	14
Figure 2.19	Inside of the furniture	14
Figure 2.20	Furniture in the room	14
Figure 2.21	Place furniture in the room	16
Figure 2.22	Living room	16
Figure 2.23	Setting in IKEA apps	16
Figure 2.24	Function button	16
Figure 2.25	Favourite button	16
Figure 2.26	Categories	16
Figure 2.27	Exit	17
Figure 3.1	Waterfall Model	25
Figure 3.2	ADDIE Model	26

Figure 3.3	RAD Model	28
Figure 3.4	Iterative Model	29
Figure 3.5	RAD Model	32
Figure 3.6	This question is to identify what function that user need in the application. User need to choose at least 6 basic function exclude for “all above” and “other”.	33
Figure 3.7	A tutorial menu on how to use the application will guide user step by step.	34
Figure 3.8	An about menu that have all the information of the application.	34
Figure 3.9	Function of buttons in the application. User need to choose at least 4 exclude for "all above" and "other".	35
Figure 3.10	Based on this question, the music/sound effect does not suitable in the application.	35
Figure 3.11	Simple information is necessary	36
Figure 3.12	This question shows the users wants the application can be install in both Android and iOS.	36
Figure 3.13	The application can be use for adult.	37
Figure 3.14	Majority users answer 4 - 6 categories.	37
Figure 3.15	User chose at least TWO (2) designs of furniture with 6 different colours for each 3D model (total 12).	38
Figure 3.16	User chose to mix both modern furniture and antique furniture.	38
Figure 3.17	Animation for the certain 3D models of furniture are suitable.	39
Figure 3.18	List of the application that have been tested in the smartphone.	40
Figure 3.19	Work Breakdown Structure	41
Figure 3.20	Flowchart to place the 3D models	42
Figure 3.21	Flowchart to take image and view the image taken	43
Figure 3.22	Flowchart to open setting	44
Figure 3.23	Flowchart to open setting (continue)	45
Figure 3.24	Context Diagram	45
Figure 3.25	Use Case Diagram	46
Figure 3.26	First Interface	47
Figure 3.27	Second Interface	48
Figure 3.28	Third Interface	49
Figure 3.29	Fourth Interface	50

Figure 3.30	Fifth Interface	50
Figure 3.31	Sixth Interface	51
Figure 3.32	Gantt Chart	54
Figure 4.1	Minimalist Bookshelf	59
Figure 4.2	Normal Bookshelf	59
Figure 4.3	Stackable Bookshelf	60
Figure 4.4	Accent Floor Cabinet	60
Figure 4.5	Semi Classic Cabinet	61
Figure 4.6	Unique Drawer Cabinet	61
Figure 4.7	Basic Chair	62
Figure 4.8	Fabric Chair	62
Figure 4.9	Stool	63
Figure 4.10	Table Lamp	63
Figure 4.11	Touch Lamp	64
Figure 4.12	Two Lamp	64
Figure 4.13	Modern Sofa	65
Figure 4.14	Sectional Sofa	65
Figure 4.15	Wooden Arm Sofa	66
Figure 4.16	Coffee Table	66
Figure 4.17	Dining Table	67
Figure 4.18	Round Table	67
Figure 4.19	Tracking interface before run the application	68
Figure 4.20	Tracking interface after running the application	68
Figure 4.21	Home interface	69
Figure 4.22	Gallery interface	69
Figure 4.23	About interface for this application	70
Figure 4.24	About interface for list of furniture	70
Figure 4.25	Tutorial interface	71
Figure 4.26	VideoPad Video Editor	71
Figure 4.27	Button	72
Figure 4.28	Icon	72
Figure 4.29	Image	73

Figure 4.30	Wikitude SDK for student license key	74
Figure 4.31	Tracking interface in application.	113
Figure 4.32	Home interface in application	114
Figure 4.33	Select and Load the 3D models of furniture.	114
Figure 4.34	Gallery interface in application.	115
Figure 4.35	Setting interface in the application.	116
Figure 4.36	Overall description.	117
Figure 4.37	List of the furniture in the application	117
Figure 4.38	Tutorial menu in the application	118
Figure 4.39	Question 1	119
Figure 4.40	Question 2	119
Figure 4.41	Question 3	120
Figure 4.42	Question 4	120
Figure 4.43	Question 5	121
Figure 4.44	Question 6	121
Figure 4.45	Question 7	122
Figure 4.46	Question 8	122
Figure 4.47	Question 9	123
Figure 4.48	Question 10	123
Figure 4.49	Question 11	124
Figure 4.50	Question 12	124

LIST OF ABBREVIATIONS

3D	Three-Dimensional
AR	Augmented Reality
ADDIE	Analysis, Design, Development, Implementation, Evaluation
CASE	Computer-aided software engineering
CPU	Central Processing Unit
CS	C Sharp
GUI	Graphical User Interface
IT	Information Technology
JAD	Joint Application Development
MN	Minneapolis
NFC	Near-Field Communication
RAD	Rapid Application Development
RAM	Random Access Memory
SDLC	Software Development Life Cycle
UAT	User Acceptance Test

CHAPTER 1

INTRODUCTION

1.1 Background

The Augmented Reality (AR) is an area of computer graphics that allows the introduction of the concept of mixed reality in our daily lives. It is obtained by simultaneous visualization of synthetic visual objects (which are generated with the aid of the computer) with the real ones (Carvalho, Maçães, Varajão, Sousa, & Brito, 2009). Augmented reality can be implemented and viewed in 3D or imaginary platform of objects to interact with the user's environment in real time. Hence, it is also provide the user with useful information that user needed immediately and uses the existing environment to overlay new information on top of it. New technology that different from marker-based augmented reality that called markerless augmented reality has been used widely these days. The features are more realistic, and advance compared to marker-based. Therefore, markerless AR system that does not need artificial markers has been developed. The conventional method of markerless AR systems relies on natural features of the image from the live video (Miyake, Fukuda, Yabuki, & Motamedi, 2017). Meanwhile, in the marker-based method, it is possible to position a 3D model near a marker by using a large-scale marker for visualizing an outcome. This method is relatively easy to implement; however, the camera needs to constantly capture the marker, which brings limitations for the users' movability and challenges related to the installation of large markers (Miyake et al., 2017).

Markerless tracking using a method of positional tracking. Within its environment, markerless need the determination of position and orientation of an object. Many developers used markerless augmented reality to create games and entertainment, commercial product visualization, advertising, and so on. For example, in term of grocery part. When user point the smartphone's camera to the grocery item, it will show the details of that product (Shedole, Satpute, Singh, & Pingale, 2015). It is also can be used to create an interactive mobile markerless augmented reality for interior design. For this project is focused on furniture for interior design in the room. Interior design is the art of enhancing the interior of a building to achieve a healthier environment for the people using the space. Combination ideas of interior design, 3D models and the latest technology, so it can produce an interactive mobile markerless augmented reality for interior design.

An interactive mobile markerless augmented reality for interior design allowed user to design their room with multiple furniture that will provide in the mobile application that include different colours and design. It is very convenient because it is only used smartphone as hardware component and software generated from computer that need to install into the smartphone. User also enable to rotate, capture the designed room, view the captured images, reset, drag and drop to rearrange the furniture based on suitability in the room. An interactive mobile markerless augmented reality for interior design is simple application which means easy to understand by recognizing the icons that represent the furniture, so it can be use by people in all ages. Rotate, drag, and drop functions are made to let user see more detail about the furniture.

1.2 Problem Statement

Various of AR application that implemented markerless-based for users can be found in internet. Unfortunately, the application seems are not be used globally and have certain problems. User have problems to design their room due to lack of this augmented reality technologies itself. Firstly, user need to go shopping and long searching the furniture (Uplaonkar et al., 2015) and tends to waste time. Problems that user generally faced by the people who are busy with their work and have no time to go to various stores

to buy furniture for their home. Time can be save by using this application and user also can capture the virtual designed room as reference.

Secondly, visualizing how a particular table or chair will look in a room before it is decorated is a difficult challenge for anyone (Phan & Choo, 2010). In the case of interior design, the designer essentially applies the three basic principles of interior design which are color, scale, and proportion within a predetermined space. Thus, the proposed AR system is focused on giving the user the flexibility to design using these three basic principles (Phan & Choo, 2010). With this technology, it can avoid user from purchasing the furniture before it can be match with the room.

Thirdly, when they have time for shopping, it is difficult for them to visualize how a particular table or chair will look in a room before it is decorated because could not check for the proper size of the product that fits their room environment. There is difficult to fulfil the customer's need of decorating their room without imaginary picture to refer (Uplaonkar et al., 2015). Every room has different size and own designs. When using this application, it will allow the user to view the product according to their needs and room environment (Uplaonkar et al., 2015).

1.3 Objectives

The goal of this project is to make user's task easier to design the room by using latest technology. The objectives of the project are:

- i. To study the available mobile markerless Augmented Reality application for interior design.
- ii. To design and develop a mobile markerless Augmented Reality application for interior design.
- iii. To evaluate the effectiveness of the developed mobile markerless Augmented Reality application for interior design.

1.4 Scope

User scope:

- i. This application can be use by the anyone in order to design their room.

System scope:

- ii. There will be 6 categories of furniture which are bookshelf, cabinet, chair, lamp, sofa, and table. For each category consist of 3 different designs. Overall it will be 72 models of furniture with different colours, materials, and designs. In the application will have take the screenshot, view the screenshot, select the 3D models, delete the selected screenshot, view the overall description about the application and its function, view the list of the furniture, and the tutorial on how to use the application.

Development scope:

- iii. The codes it will use C# as the programming language for the whole application and Unity 3D as the platform to create the application. Autodesk Maya 2016 is to create all the 3D models. User can install in Android smartphone.

1.5 Thesis Organization

This thesis consists of five chapters. Chapter 1 discuss about introduction to the project, which briefly explain the concept of AR, marker-based and markerless AR, and it is current technology, it also states problem statement, objective, and scope.

Chapter 2 is about the literature review, which are contains the comparison of three existing systems that developed by other parties. This chapter also contains general information about the study of this project and comparison of hardware/method/technique used to develop this application which are suitable to be implement into the project.

Chapter 3 will discuss about the overall approach and also the framework of the project. This chapter will explain the methodologies of the system, tools, and techniques that will be use in the implementation phase.

Chapter 4 (Implementation, Testing, And Result Discussion) will describe the procedure of the development of the system and explanation of the tools and techniques that use in this phase. This chapter also defines the sample data and the outcome of the system by testing the system.

Chapter 5 is the conclusion which contains the conclusion of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses about existing systems of augmented reality that use the same concept but different design and functionality of applications by other parties. Not only than that, in this chapter will contains information about the study of the project in general. This chapter also will briefly explain the techniques/method/hardware or technologies that suitable to implement in this project.

2.2 Augmented Reality

Augmented reality is an interactive, reality-based display environment that used computer generated display, sound, text and effects to make user feel the same as real-world experience. It's combines real and computer-based scenes and images to display and produce the complete application, then enhance view of the world. Augmented reality has many different implementations models and applications. The primary focus in the augmented reality is to give user a rich audio-visual experience. Image and speech recognition, animation, head-mounted and hand-held devices are examples in employing computerized simulation and techniques that works on augmented reality.

2.3 Types of Augmented Reality

Augmented reality markers or short AR-markers are visual cues which trigger the display of the virtual information (“What are augmented reality markers ?,” 2013). Image recognition is an imperative component of augmented reality systems (Yariv Levski, 2016). Visual markers are embedded in the system and physical world objects are detected the markers to produce the 3D model. Many applications employ the tracking technique to estimate the orientation and position of a camera same as real world frame. Tracking technique also can be known as markers based. To decode the markers that are implemented in the system by cross-matching, it can be use various of technique. The 3D model will be display from the database.

Markerless augmented reality is a technique that allows the user to use the physical environment as the base or target for the placement of virtual objects (Yariv Levski, 2016). Markerless AR depends on the natural features of a surrounding compared to marker-based. Some markerless systems can extract and store characteristics and information about the environments they are used on for later usage. Markerless are enabling to locate and interact with the AR resources by using the GPS feature in built in the smartphones or others (Arbelález-estrada & Osorio-gómez, 2013). Using the markerless, user limitless to produce the 3D model compared to marker-based that only based on the markers.



Figure 2.1 Marker-Based



Figure 2.2 Markerless-Based

2.4 Review of Existing System

This section explained about three applications that have similar function but different quality. The applications are Room Remix, Roomle 3D/AR Furniture Catalog, and IKEA catalogue.

2.4.1 Application 1 – Room Remix

Room Remix is a free application, augmented reality furniture shopping app created by Praxik, LLC, a start up in Minneapolis, MN. The idea of Room Remix came from Room Remix's team who had a common frustration about online furniture shopping. Room Remix's team found the easier way to visualize furniture after they faced experiencing a few regrettable online furniture purchases. This application allowed user to place 3D models in the room before hitting the purchase button. Room Remix also host hundreds of furniture models from different brands including Amazon and Wayfair.

Room Remix is a simple application that have certain steps that need to follow to get perfect result. Firstly, user must take a picture of the room or space that user want to decorate as shown in Figure 2.3. The pictures can be capture in portrait or landscape mode. Secondly, user can add furniture and decor pieces with the push of a button as shown in Figure 2.4. When the furniture is in the space, user can freely position them according to user satisfaction and swap in new furniture until found the furniture that they want to purchase. Thirdly, it's time to buy the furniture as shown in Figure 2.5. The collaboration with trusted online retailers, Room Remix has hundreds of furniture items and it is provided with free shipping to user. Room Remix used concept of markerless and with combination of 3D modelling to make it looks so real. This version of application for iOS user only.



Figure 2.3 Space of the room

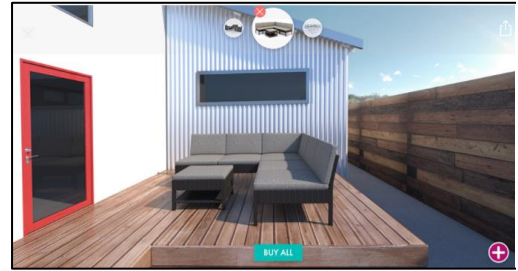


Figure 2.4 Place the furniture

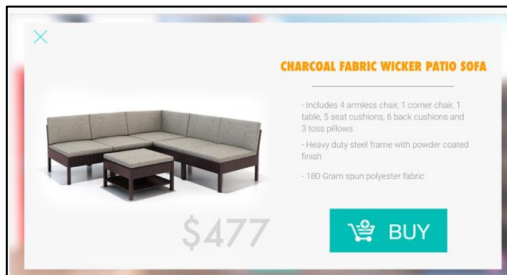


Figure 2.5 Purchase the furniture



Figure 2.6 Setting up the furniture

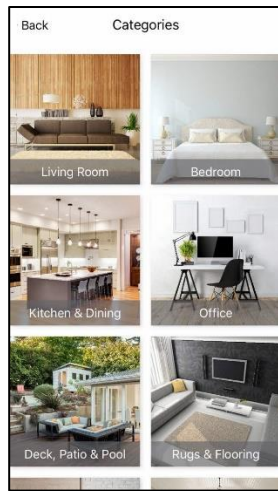


Figure 2.7 List of furniture



Figure 2.8 Saved scenes



Figure 2.9 Shopping Cart

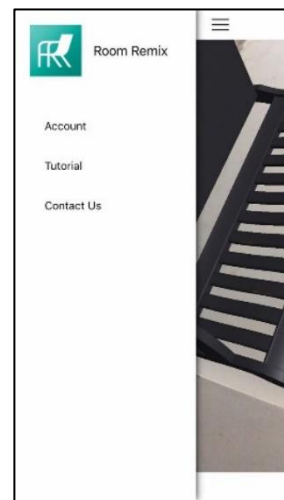


Figure 2.10 Setting

2.4.2 Application 2 – Roomle 3D/AR Furniture Catalog

Roomle 3D/AR Furniture Catalog is a layout editor that supports both 2D and 3D views for easy visualization of user creations. It allowed user to remodeling, room rearranging, and anything else. Roomle for quick mock-up or set the scale and creating a full layout with furnishings as shown in Figure 2.13. In this application provide a lot of items such as doors, windows, tables, cabinets and even small details like plants and rugs as shown in Figure 2.12. No need to sign up, but if user create an account, user have speciality to save the creations.

Roomle provide the cloud-based platform enabling planning-based interactive furniture shopping for furniture retailers, brands, and end consumers, for both mobile and in the web browsers (Jason Fitzpatrick, 2010). In this application, it's contains only 3 button functions because application disabled certain function such as save button of user's creations to those who do not have an account. User can access tutorial button, favourite button, and zoom in button only as shown in Figure 2.20. User can change colour and other things of the furniture in 3D/AR Configurator. But, for the certain furniture only. In 3D/AR Configurator, also stated the width of the furniture in centimetre (cm) and it can be change. User can get the furniture by setting up the size of the furniture and this application will list up the various furniture and the price. User allowed to choose 3 main search function such as rooms, products, brands, or direct search for the furniture. In every search function have sub-search function named as browse by topic as shown in Figure 2.12.

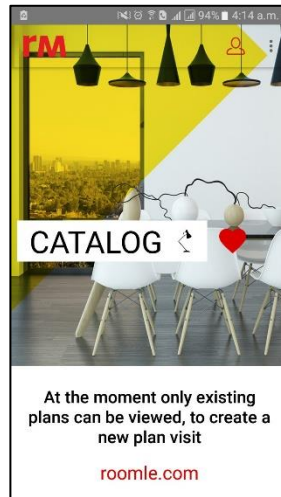


Figure 2.11 Home

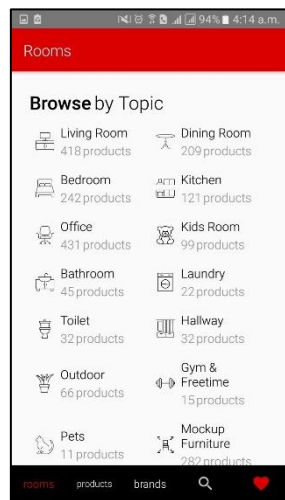


Figure 2.12 Browse by Topic

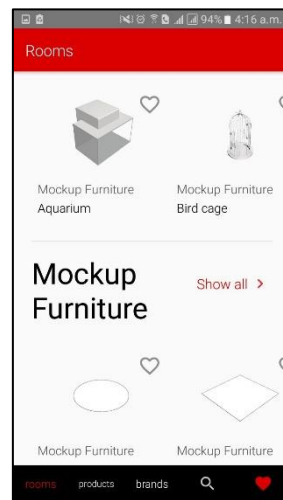


Figure 2.13 Mockup Furniture



Figure 2.14 All Brands

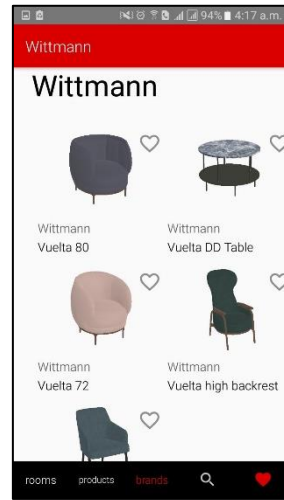


Figure 2.15 Branded furniture



Figure 2.16 Favourite

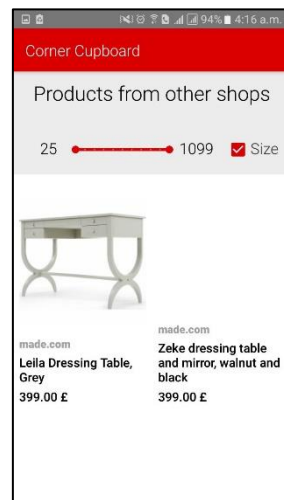


Figure 2.17 Scaling the furniture

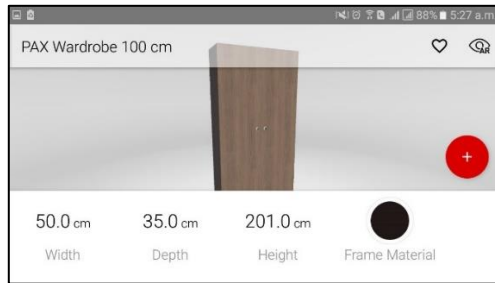


Figure 2.18 Furniture

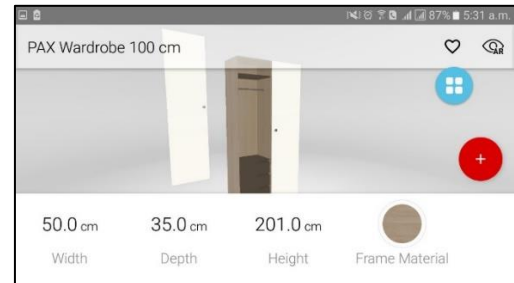


Figure 2.19 Inside of the furniture

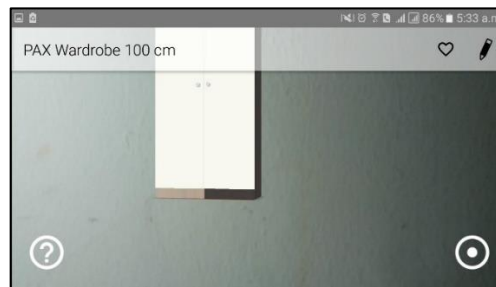


Figure 2.20 Furniture in the room

2.4.3 Application 3 – IKEA Catalogue

IKEA is a multinational group that designs and sells ready-to-assemble furniture, kitchen appliances and home accessories. The company sells item to be known as modernist designs for various types of furniture and appliances and they have provided example of designed rooms based on categories. In 2014, IKEA announces the launch of IKEA Catalogue, an Augmented Reality application that allowed user to design any space especially home, office, school or studio. All product in IKEA Catalogue application are 3D and true scale so that every item is just the right size. Both iOS and android user can install this application through their smartphone or tablet. Unfortunately, not all types of smartphone support the function of the application. Requires android version 5.0 and above to support the application. In this application also provide catalogue of the furniture with the description and price of items. They have kitchen price which means including

appliances such as oven, hob, extractor, fridge/freezer, and dishwasher with the price of €2810.

To be able to use the 3D feature, the device must have back camera, a gyroscope sensor and support NFC. This application include function about intro, living room, dining room, kitchen, bedroom, bathroom, product, feature stories, and information. They provide direct link to the webpage of IKEA. In setting section, user can change units either in cm or inch and user also can change height of the furniture as shown in Figure 2.23. IKEA Catalogue have 12 categories separated for specific furniture which are Bathroom, Bedroom furniture, Children's IKEA, Decoration, Dining, Home organization, Kitchen, Lightning, Livingroom, Secondary storage, Storage and organise, and Workspace as shown in Figure 2.26. The camera can be in portrait or landscape mode.

User are enable to place more than 1 furniture at same time. The furniture also can be zoom in and zoom out, rotate, and drag to move along the smartphone or tablet screen. Each furniture has favourite, information, delete, and replace button. But, when user close to the virtual furniture, the size and position of the furniture remain the same. Furthermore, it also has camera that capture the room with the furniture that has been positioned. User can share, save and cancel the picture taken before. At the top of the application, exit button to main page, favourite button, and settings was provided.



Figure 2.21 Place furniture in the room

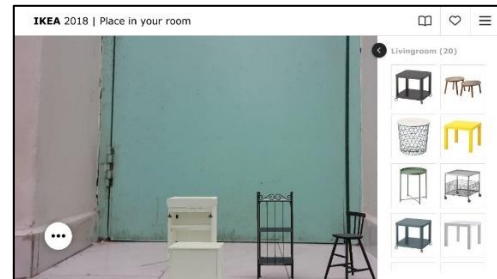


Figure 2.22 Living room



Figure 2.23 Setting in IKEA apps



Figure 2.24 Function button

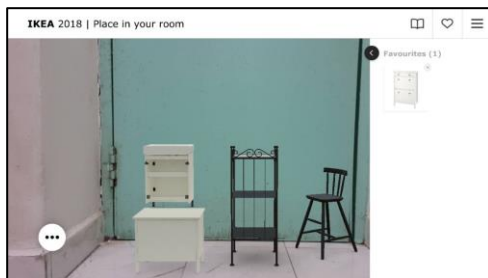


Figure 2.25 Favourite button



Figure 2.26 Categories

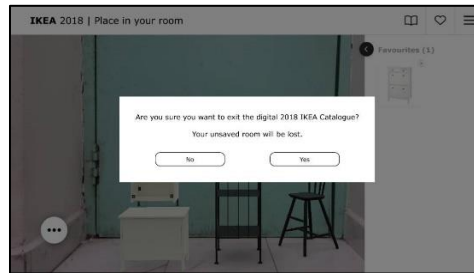


Figure 2.27 Exit

2.5 Comparison of Three Existing Systems

Table 2.1 shows a comparison of three existing systems. There are 10 elements that used for comparison in that three existing systems.

Table 2.1 Summary of comparison the existing system

Application Name	Application 1 – Room Remix	Application 2 - Roomle 3D/AR Furniture Catalog	Application 3 – IKEA Catalogue
GUI	This application is almost complete and easy to handle. But it should support the android user also. It makes user feel the same as in real life.	When user open the application, it makes user lost to enter main function of the application. Quite messy and poor GUI.	This application combines the catalogue and the 3D features to design user's room. The catalogue is a guideline to user to choose furniture and read the details and price before user purchase it.

			GUI in this application is very good, convenient and easy to use to all users.
Mobile OS	iOS only.	iOS, Android and Web Browsers.	iOS and Android.
Language	English only.	English only.	English, German – Austria, Dutch, French, Bulgarian, Chinese – simplified, Croatian, Greek, Czech, Danish, Arabic, Finnish, German, Chinese – traditional, Hungarian, Iceland – Iceland, Indonesian, Hebrew, Italian, Japanese, Lithuanian, Morocco – Arabic, Morocco – French, Norwegian, Polish, Portuguese, Romanian, Russian, Serbian, Slovakian, Korean, Basque,

			Catalan, Galician, Spanish, Swedish, French – Switzerland, German – Switzerland, Italian – Switzerland, Thai, English – American, and Spanish – Latin American.
Connection Type	Need the internet connection.	Need the internet connection.	Need the internet connection.
Audience	Adult.	Adult.	Adult.
Topic Covered	Tracking the suitable place in the room and 3D models of furniture provided.	Choose the furniture from the catalogue and it will be come out with 3D model. Multiple brands of furniture are covered.	Scanned catalogue as reference and 3D models of furniture.
Type of AR	Markerless.	Markerless.	Markerless.
Main Function	Rotate, drag, resize, tracking, reset, AR camera, capture pictures, view captured pictures, share the result of creatures, and shopping cart.	Rotate, drag, resize, reset, AR camera, 3D/AR Configurator, AR visualization, HD Rendering, and Floor Planner.	Rotate, drag, resize, reset, AR camera, and capture pictures.

Advantages	<ol style="list-style-type: none"> 1. Tracking the suitable place in the room before the application place the furniture. 2. 3D models in the right position even when user tilt or move the AR camera. 3. Easy to understand the button that represent the functions. 	<ol style="list-style-type: none"> 1. Have a lot of brands and user can set up the scale of the product to get the exact scale of furniture. 2. User can directly go to the website of the brands. 3. Application also inserted the example of image when the furniture place in the room. 	<ol style="list-style-type: none"> 1. All 3D models used looks so real. 2. Fast tracking to place the furniture. 3. Smooth movement of furniture along the room.
Drawbacks	<ol style="list-style-type: none"> 1. Do not have the details of the furniture. 2. Cannot change the colour of the furniture. 	<ol style="list-style-type: none"> 1. Cannot place more than 1 furniture at one time. 2. Cannot save existing designed room. 	<ol style="list-style-type: none"> 1. Do not have search button to make user search the furniture in fastest way. 2. Limited models according to

	3. No description for the first timer user.	3. Rotation can be done on side by side only. Not overall rotation can do such as for top and bottom.	the catalogue. 3. Cannot change colour of the furniture.
--	---	---	---

2.6 Summary

From this three-existing system that can be conclude is their Graphical User Interface (GUI) has level of poor, middle and good GUI. Colours that has been used in all application were satisfied. Moreover, the interaction between the user and the application for all application throughout the functions is great. iOS, android and web browsers supports the application respectively. Room Remix that support iOS only seems almost perfect compared to others based on their performance. Type of Augmented Reality for all application are using markerless. The most basic element in the three applications are rotate, drag, resize, reset, and AR camera. Besides that, all application has their own advantages and drawbacks.

From these three applications, what can be conclude is Room Remix is the most excellent application among others because the function almost complete and suitable for everyone. The function that can be conclude from the comparison and will apply into the an interactive mobile markerless augmented reality for interior design is rotate, capture the designed room, view the captured images, reset, drag and drop to rearrange the furniture based on suitability in the room. The 3D models of furniture will have 10 modules with multiple colour. This application can be installing for android user only and using English as main language. Connection of internet are needed to load the 3D models, and everyone can use the application. This application will be using markerless

technology and it will track the suitable place in the room and 3D models of furniture also will provided in the application.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology is an important phase, which acts as an implementation guideline to carry out the project in an effective and efficient way. It is defined as the theoretical analysis phase in the development of the project, which requires time and effort. In this chapter, the methodologies, tools, and techniques that will be used in the implementation phase will be briefly discussed. The methodology for developing the software is known as software development life cycle (SDLC), which contains various phases to have better planning and management. Many SDLC models are used such as Waterfall Model, Rapid Application Development (RAD), ADDIE Model, and Iterative Model. In this project, Rapid Application Development (RAD) is most suitable as a Methodology.

3.2 Types of Software Development Life Cycle (SDLC)

In this subsection, four SDLC will be briefly explained and compared which are Waterfall, ADDIE, Rapid Application Development, and Iterative models. At the end of this subsection, the best methodology is chosen to explain the overall method of this project.

3.2.1 Waterfall Model

The waterfall model is the flow of the sequence. In this Waterfall model, there are 6 phases of software development activity as shown in Figure 3.1 and each phase is different and comprises a series of tasks and has different goals to achieve. By flowing steadily downwards like a waterfall throughout the software implementation phases, the development progress can be tracked. It is not the main thing to focus on changing the requirements, but the customer has very clearly documented requirements. The phases of the waterfall model fall from higher to lower levels, the same as the actual waterfall.

There are six phases of the waterfall model that must be followed for the development of software. Firstly, all possible requirements of the system that want to develop are recorded in the requirement analysis and must have been documented in a requirement specification document. Brainstorming and walking through the need to make the understanding of the need more efficient. System design is the second phase. Previous phase of the requirement specifications and system design are investigated. System design can help to define requirements for hardware and software. In addition, the designs must be documented.

Phase three is Implementation. The system design input from the previous phase to create the codes and integrate the codes for the next phase. Each code is developed and tested for the functionality of unit code testing. Fourth phase can be known as System Testing. Integrate the unit testing code and test the code to make sure it works as expected. In addition, all testing activities (Functional and Non- Functional) must be carried out to ensure that the system complies with the requirements and records the test progress and reports all testing activities.

Phase fifth is System Deployment. After complete functional and non - functional testing, the product is placed on the market or deployed in the customer setting. Lastly, system maintenance is the sixth phase. There may be some problems in the customer environment. Patches are released to solve these problems and to improve the product,

new and better versions are available. Provide such changes in the customer environment after system maintenance has been completed.

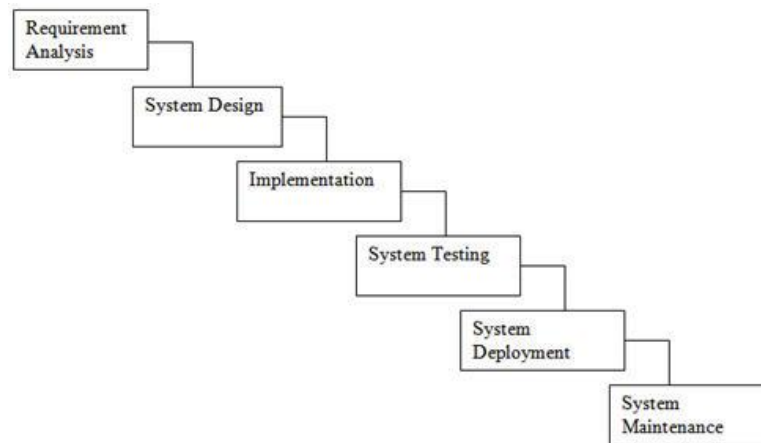


Figure 3.1 Waterfall Model

3.2.2 ADDIE Model

The ADDIE model is a generic process commonly used by teaching designers and developers of training. It consists of five phases, Analysis, Design, Development, Implementation and Evaluation as shown in Figure 3.2 which represent a flexible and dynamic guideline for the development of effective training and support tools for performance. Although this model is the most common design model, some weaknesses of the ADDIE model lead to multiple spin - offs or variations (David L, 2014).

Analysis is the first phase. In this phase, the designer determines the learning problem, the goals and objectives, the existing knowledge, the needs of the audience and all other relevant features. The analysis phase also recognizes the learning environment, the delivery options, all limitations and project timeline. Design is the second phase. An organized process of defining learning goals. Detailed prototypes and storyboards are often produced and in this phase the look and feel, user interface, graphic design and content are determined.

Phase three is development. The actual development of learning materials and content in the design phase. The development phase is in which the developers build and collect the content assets created during the previous design phase. Programmers continue to develop and integrate technology and test procedures. The project is reviewed and examined based on any feedback received. Phase four is implementation. The plan is implemented and a training process for teachers and learners is developed during the implementation. Materials are distributed or supplied to the group of students. After delivery, the effectiveness of training materials is evaluated.

The fifth phase of the last phase is evaluation. This phase involves formative and summative assessment. There is a formal evaluation at each stage of the ADDIE process. Summative evaluation consists of tests designed to provide feedback from users for criterion-related referenced items. Revisions are made as necessary.

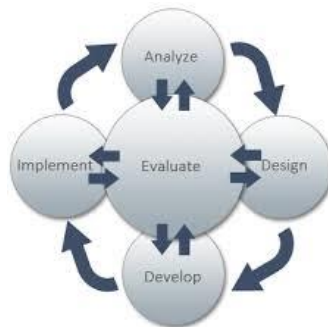


Figure 3.2 ADDIE Model

3.2.3 Rapid Application Development (RAD)

Rapid application development utilizes the minimal planning for rapid prototyping in the methodology of software development as shown in Figure 3.3. A prototype is a functional model which is the same as a product component. It can be easier to incorporate the changes into the development process because no detailed pre-planning is available. Incremental and iterative RAD projects have domain experts, customer representatives,

small teams of developers and IT resources working on their prototype or component successively.

The planning is required for the development of the product for the software itself. The development of rapid applications focuses primarily on meeting customer requirements through focus groups or workshops. The customer tests the prototypes early using the iterative concept, reuse the existing prototypes, continuous integration and fast delivery.

The first phase is the planning phase of requirements. In this phase, the SDLC 's system planning and system analysis phases combined elements. IT personnel, managers and users discuss and agree on the scope of the project, business needs, system requirements and limitations. It will end with the agreement of the team. Phase two is user design phase. Users need to interact with system analysts in this phase and develop prototypes and models that represent all system processes, outputs and inputs. The Rapid Application Development (RAD) groups or subgroups mainly use a combination of CASE tools and JAD techniques to convert user needs into working models.

The construction phase is the third stage in the rapid application development (RAD). The construction phase focuses on application and program development, a task similar to the SDLC. However, users continue to participate in rapid application development (RAD) and can still make improvements or changes as reports or actual screens are developed. The coding, programming and application development, unit integration and system testing are the tasks. Phase four is the Cutover Phase. This is the final phase of the implementation phase of the SDLC, including testing, data conversion, user training and switching to the new system.

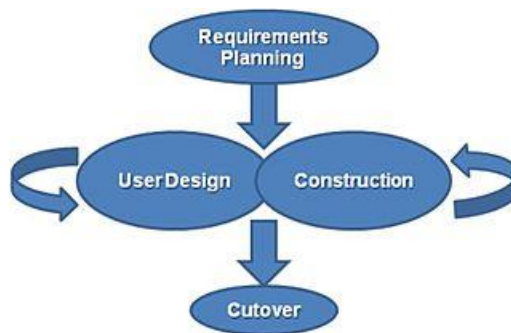


Figure 3.3 RAD Model

3.2.4 Iterative Model

An Iterative Model begins with the development of the part of the software, as shown in Figure 3.4. In this Iterative Model, you also need to begin with a full set of requirements. This means that the evolving versions are iteratively increased until the entire system is reached and ready for deployment. Iteration is every release of the Iterative Model developed over a fixed period of time (Amir Ghahrai, 2017).

The model consists of 5 phases. The Planning Phase is the first phase in the Iterative Phase. This is the first stage of this Iteration model to start the other stage. Proper planning is carried out by the team, which can help them to set software or hardware requirements, to prepare specifications documents for the next phases of the cycle. Phase two is the phase of analysis and design. After the planning has been completed in the previous phase, an analysis will be carried out to identify the appropriate business logic, database models and any other requirements of this phase. In addition, the design stage also takes place in this phase.

The implementation phase is the third phase in the iterative model. All specifications, design and design documents to date are coded and applied in this initial project iteration. Phase four is testing. The current build iteration is coded and applied, tests are proposed in the cycle to analyze and locate any possible problems or bugs in the

software. The fifth stage, finally, is the evaluation phase. This is the final phase in which the entire team and the client examine the status of the project and approve it according to the proposed requirements.

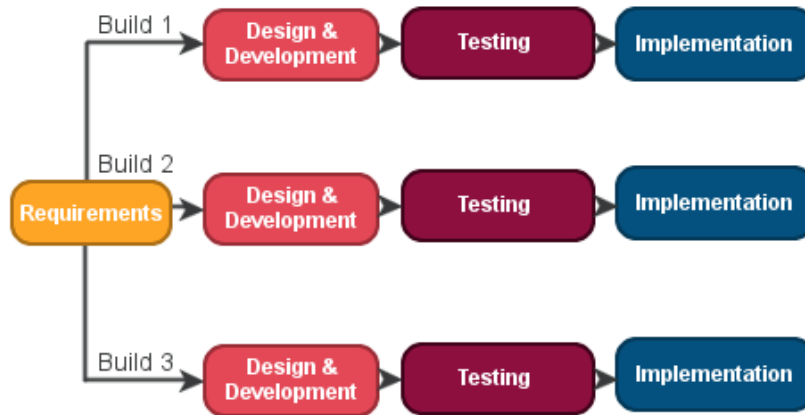


Figure 3.4 Iterative Model

3.2.5 Comparison between SDLC Model

Table 3.1 shows the comparison between SDLC Model. There are 29 elements that used for comparison which are flexibility, understanding requirements/requirement specification, duration, cost, guarantee of success, initial product feel, client satisfaction, risk involvement, customer involvement, simplicity, maintenance, overlapping phases, implementation, required expertise, documentation and training required, process, framework type, testing, basic business knowledge required, suitable project size, cost control, rework cost, changes incorporated, reusability, availability of working software,

customized product, customer control over administrator, primary objective, and release cycle.

Table 3.1 Comparison between SDLC Model

Features	Waterfall Model	ADDIE Model	Rapid Application Development (RAD)	Iterative Model
Flexibility	Rigid	More flexible	Very flexible	More flexible
Understanding requirements / Requirement specification	Beginning	Beginning	Not all in the beginning and frequently changed	Well understood at beginning
Duration	Very long	Long	Short	Long
Cost	Almost as estimated	Almost as estimated	Almost as estimated	Above estimated cost
Guarantee of success	Low	High	Very good	High
Initial product feel	No	No	Yes	No
Client satisfaction	Low	High	High	High
Risk involvement	High	Medium	Very Low	Low
Customer involvement	Low	Low	High	High, after each Iteration
Simplicity	Simple	Intermediate	Simple	Intermediate
Maintenance	Least Maintainable	Maintainable	Easy Maintainable	Maintainable
Overlapping phases	No overlapping	Yes	Yes	Yes, as parallel

				development is there
Implementation	Easy	Easy	Easy	Easy
Required expertise	Moderate	Moderate	Moderate	Moderate
Documentation and training required	Necessary	Necessary	Limited	Yes, but not much
Process	Heavy weight process	Heavy weight process	Light weight process	Light weight process
Framework type	Linear	Linear	Prototype and Iterative	Combination of Linear and Iterative
Testing	After coding phase completed	After coding phase completed	After completion of coding	After every Iteration
Basic business knowledge required	Not much	Not much	Required	Moderate
Suitable project size	Small scale	Medium scale	Low to Medium scale	Low to Medium scale
Cost control	Yes	Yes	No due to Urgent Requirement	No
Rework cost	High	High	Not very high	Almost high
Changes incorporated	Difficult	Difficult	Easily	Easily
Reusability	Least possible	High	Yes	To some extent

Availability of working software	At the end of the life cycle	At the end of the life cycle	At the end of the life cycle	At the end of every Iteration
Customized product	Least possible	Least possible	Possible	Much possible
Customer control over administrator	Very Low	Low	Yes	Yes
Primary objective	High assurance	Rapid development	Rapid development	Rapid development
Release cycle	Big band (All functionality at once)	Big band(All functionality at once)	In phases	In phases

3.3 Chosen Methodology

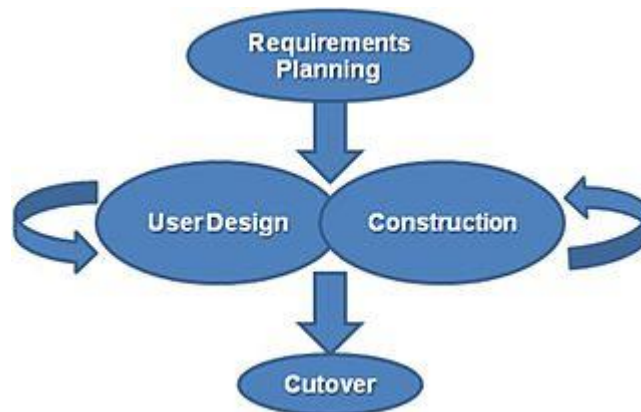


Figure 3.5 RAD Model

Based on the analyzation and comparison above, the Rapid Application Development as shown in Figure 3.5 was selected due to the many reasons, which are this SDLC support is implementation is easy, flexibility and guarantee of success is high, duration is short, risk involvement is low, and simplicity is simple. Each phase of this

model thoroughly explains in this section. As mentioned above, there are four phases in this model. Such as Requirements Planning, User Design, Construction, and Cutover phase.

3.3.1 Requirements Planning

This is the result of the questionnaire that used to collect the user requirement for this project. 69 respondents had participated in this survey.

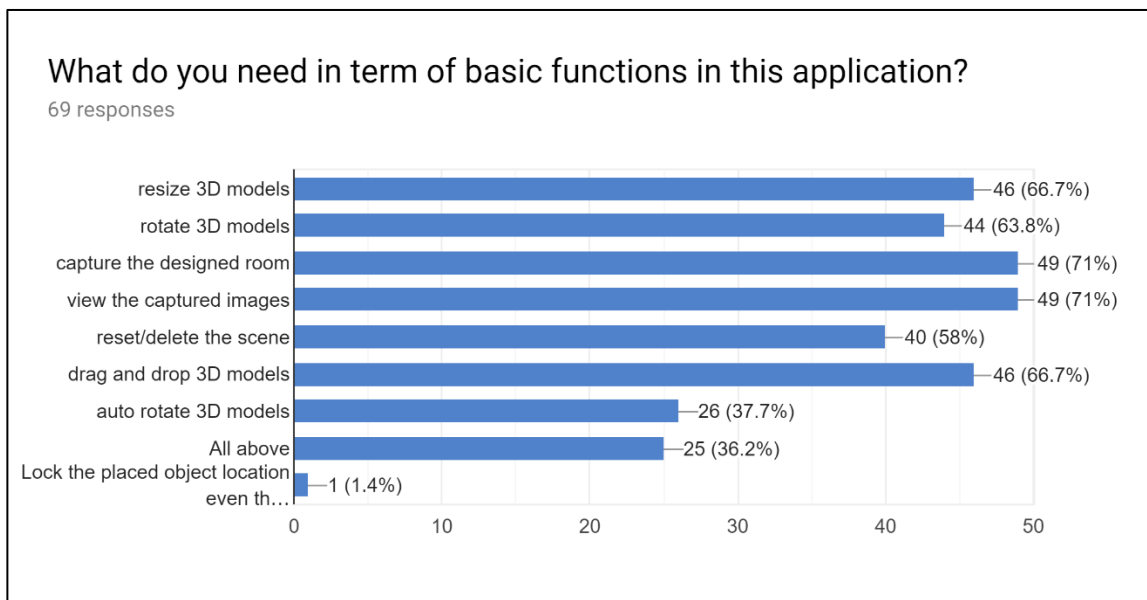


Figure 3.6 This question is to identify what function that user need in the application.

User need to choose at least 6 basic function exclude for “all above” and “other”.

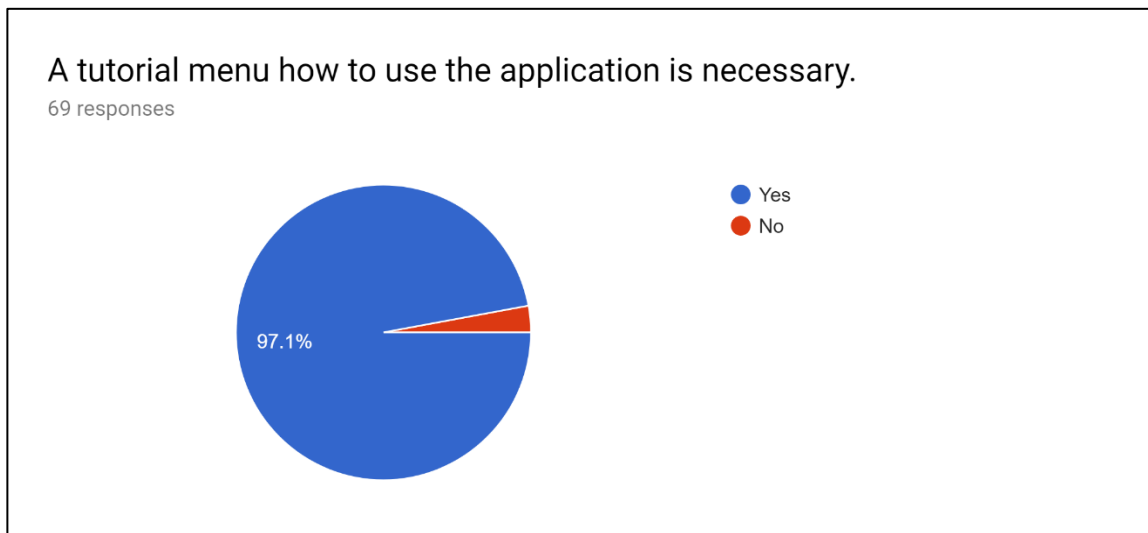


Figure 3.7 A tutorial menu on how to use the application will guide user step by step.

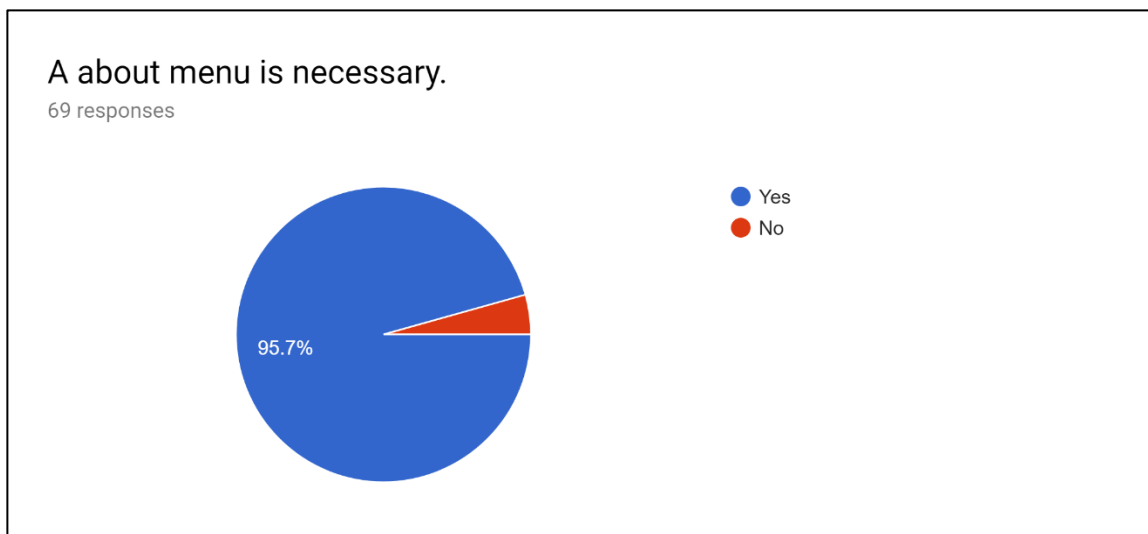


Figure 3.8 An about menu that have all the information of the application.

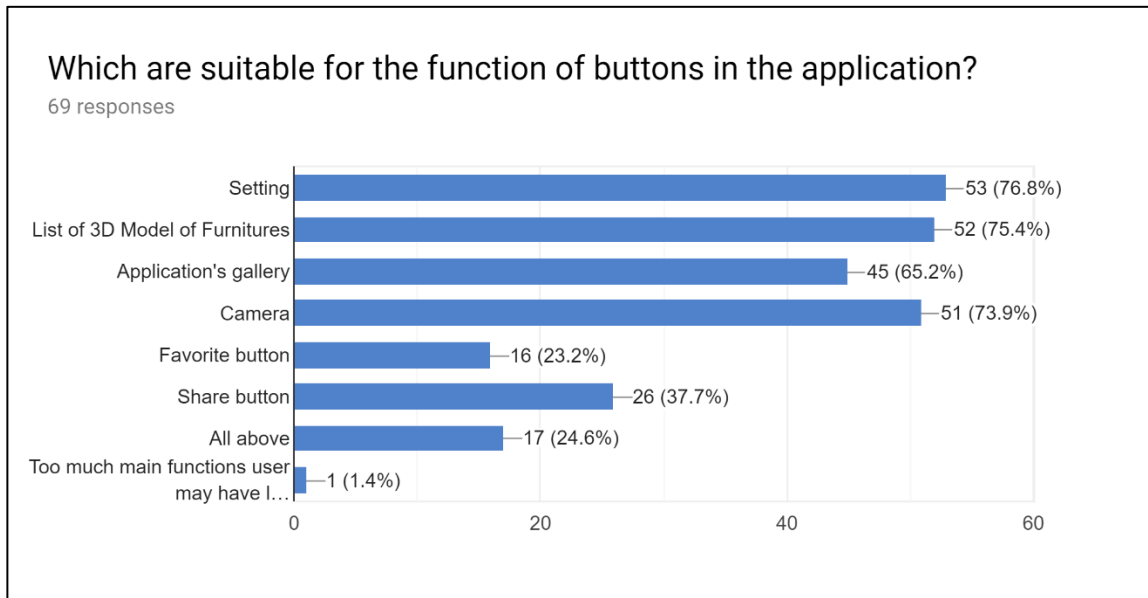


Figure 3.9 Function of buttons in the application. User need to choose at least 4 exclude for "all above" and "other".

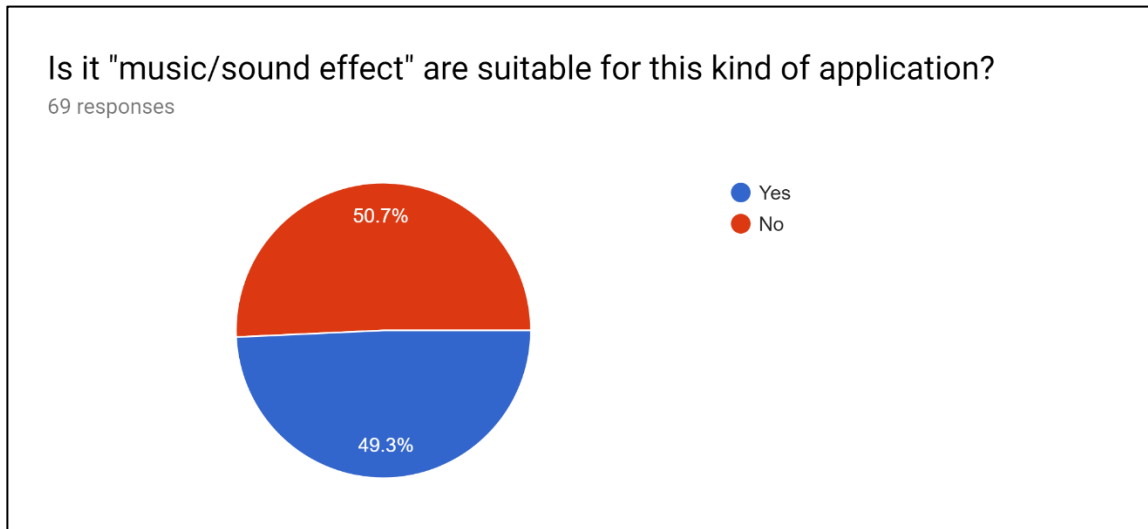


Figure 3.10 Based on this question, the music/sound effect does not suitable in the application.

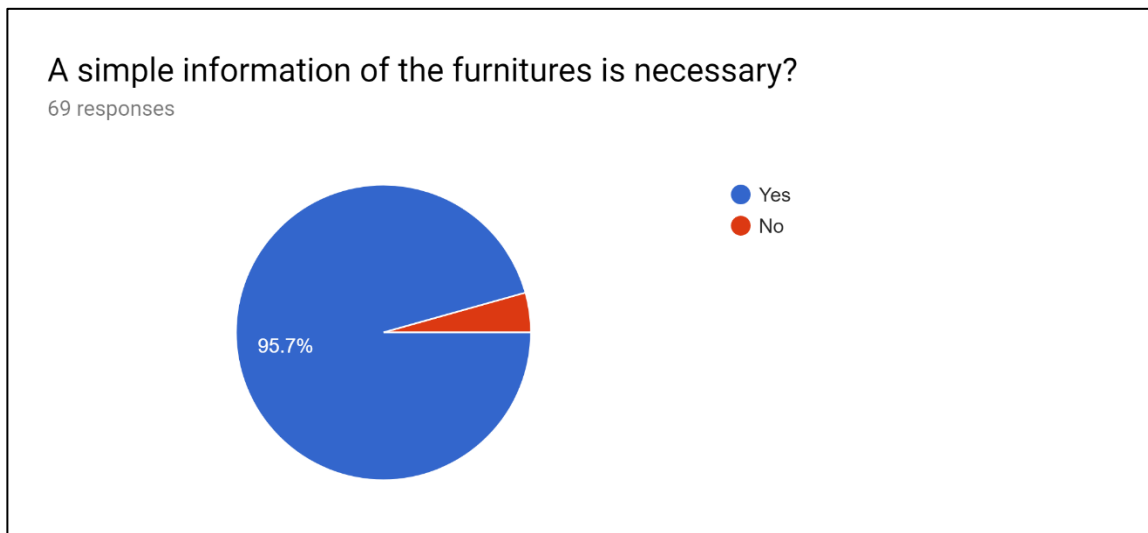


Figure 3.11 Simple information is necessary

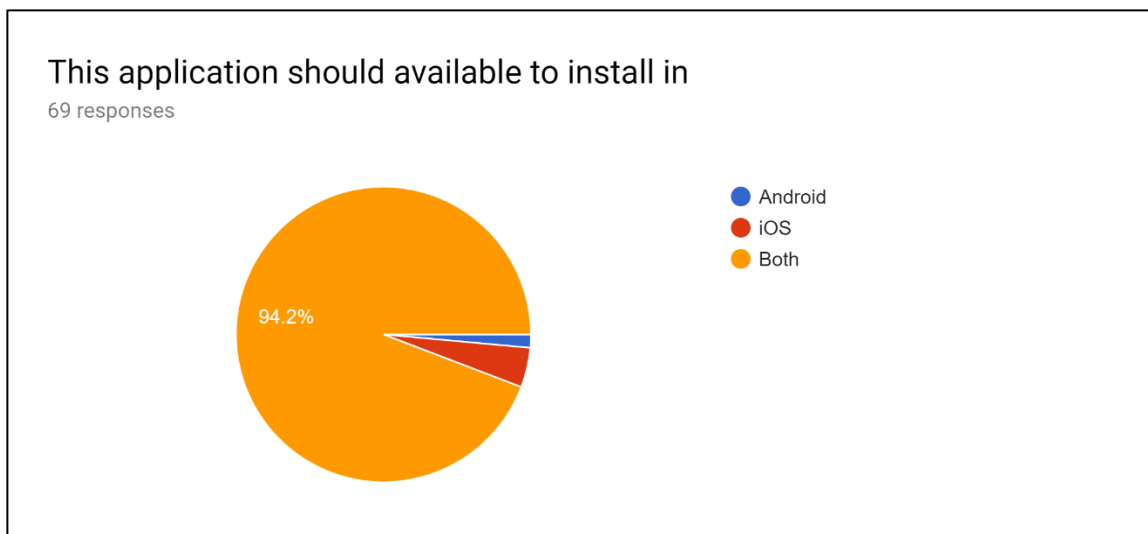


Figure 3.12 This question shows the users wants the application can be install in both Android and iOS.

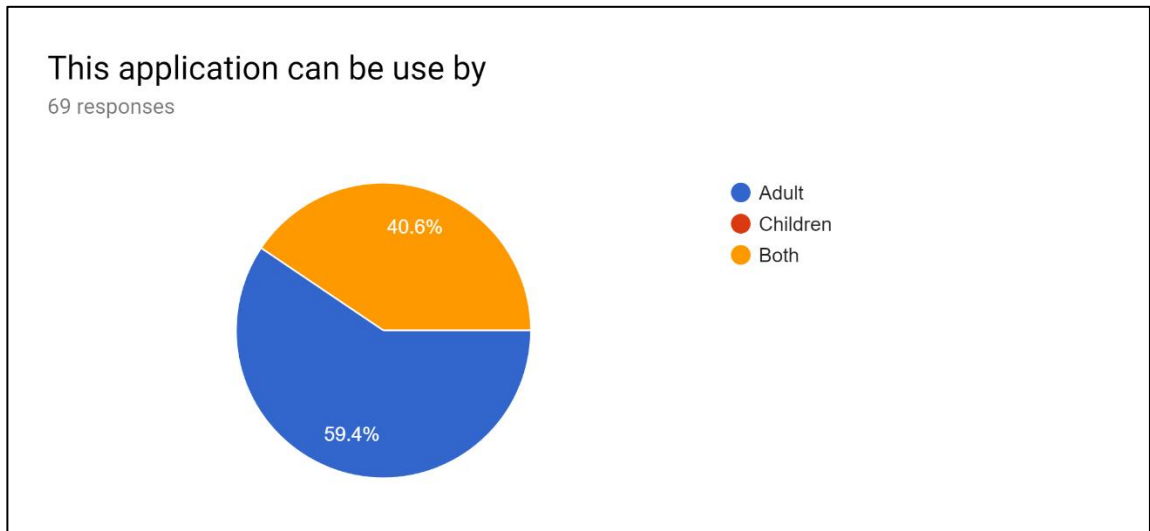


Figure 3.13 The application can be use for adult.

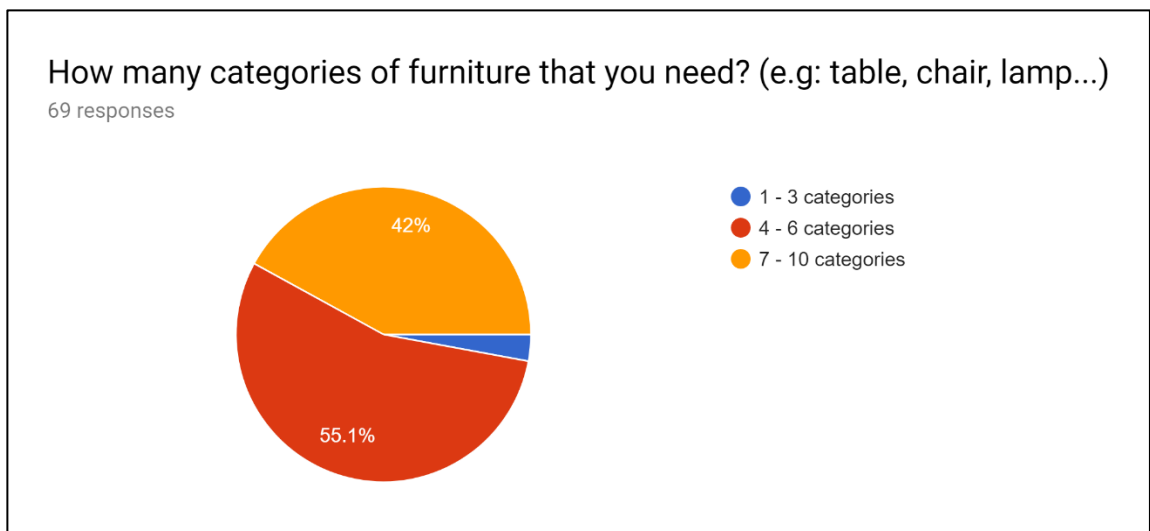


Figure 3.14 Majority users answer 4 - 6 categories.

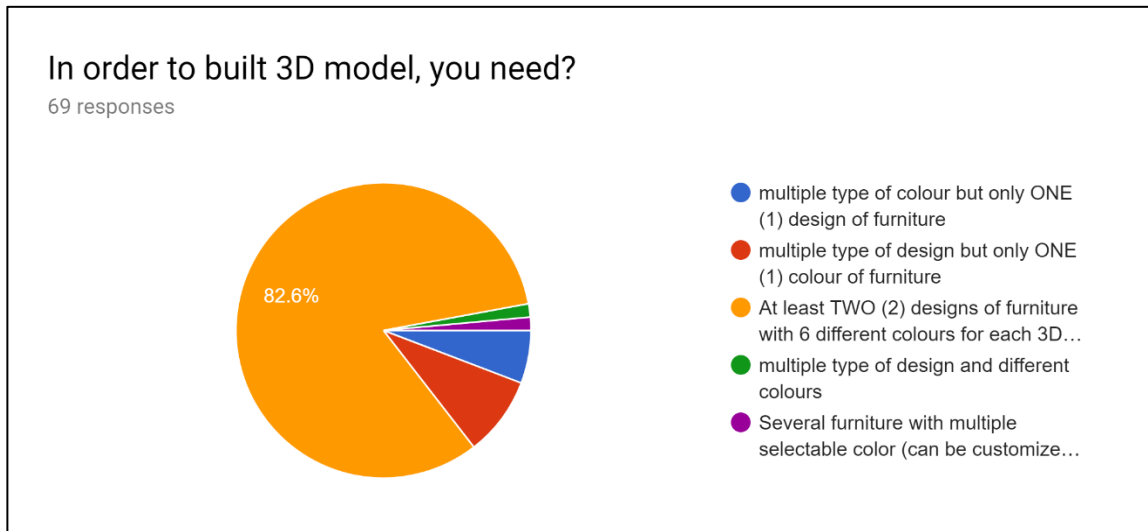


Figure 3.15 User chose at least TWO (2) designs of furniture with 6 different colours for each 3D model (total 12).

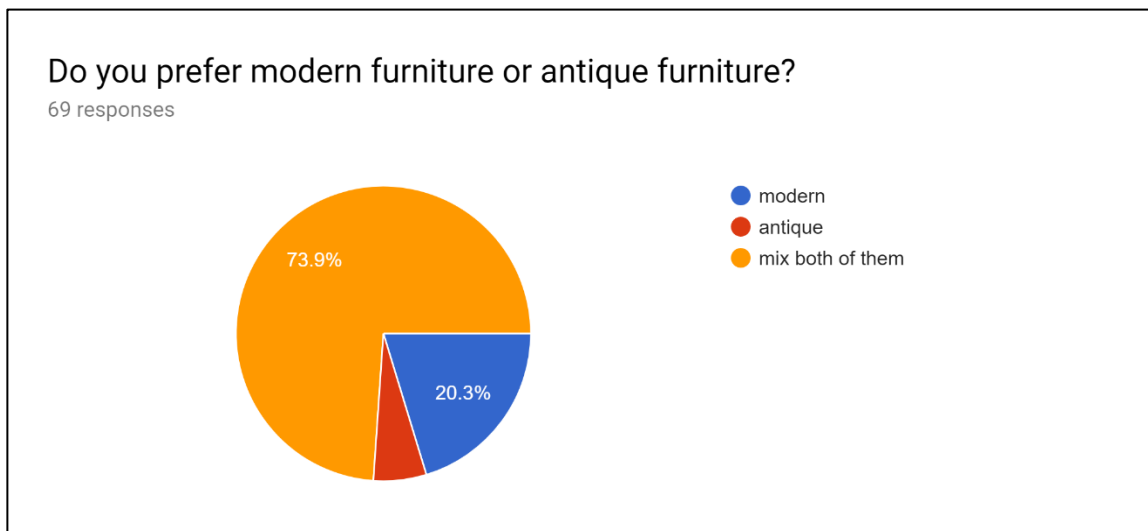


Figure 3.16 User chose to mix both modern furniture and antique furniture.

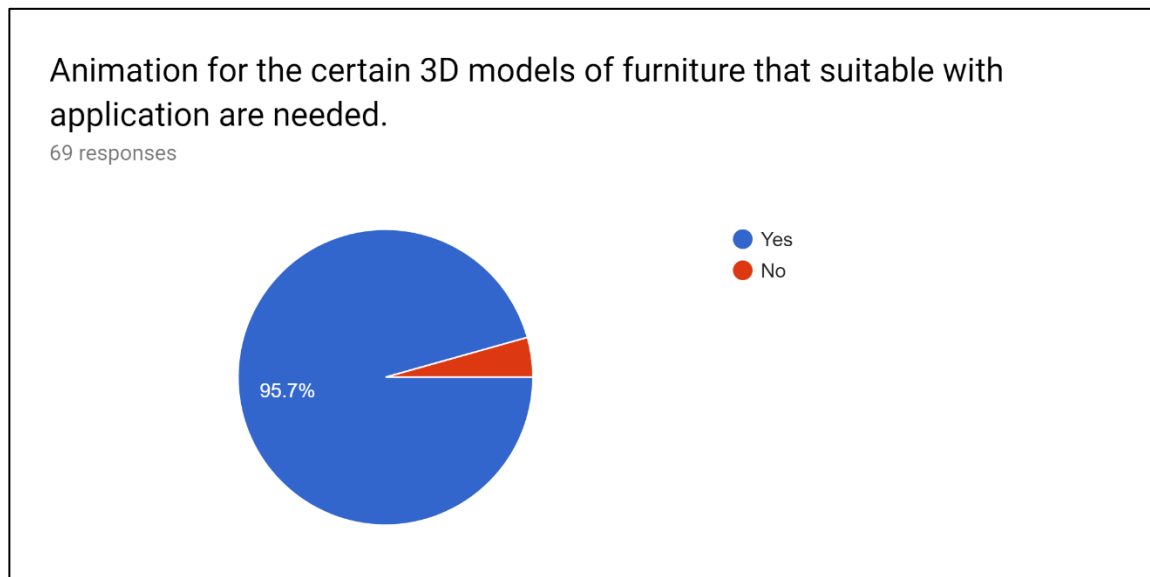


Figure 3.17 Animation for the certain 3D models of furniture are suitable.

3.3.2 User Design

In User Design phase, the storyboard of all the designs and complete description in the application as shown in Table 3.2, Table 3.3, Table 3.4, Table 3.5, Table 3.6, and Table 3.7.

3.3.3 Construction

In construction phase, detailed designs are completed in the markerless AR application and the 3D models of furniture. Markerless AR application was created and tested in order to generate a system that operates at an acceptable level of performance.

Name	Date modified	Type	Size
psmtest52.apk	16/10/2018 1:34 AM	APK File	132,567 KB
psmtest53.apk	17/10/2018 9:16 PM	APK File	132,567 KB
psmtest54.apk	17/10/2018 9:44 PM	APK File	132,567 KB
psmtest55.apk	17/10/2018 10:22 ...	APK File	132,567 KB
psmtest56.apk	18/10/2018 1:00 AM	APK File	132,566 KB
psmtest59.apk	19/10/2018 11:55 ...	APK File	132,510 KB
psmtest60.apk	20/10/2018 12:17 ...	APK File	132,511 KB
psmtest61.apk	20/10/2018 12:52 ...	APK File	132,511 KB
psmtest62.apk	27/10/2018 2:49 PM	APK File	132,495 KB
psmtest63.apk	27/10/2018 4:35 PM	APK File	132,497 KB
psmtest64.apk	27/10/2018 11:29 ...	APK File	132,495 KB
psmtest66.apk	28/10/2018 5:58 PM	APK File	132,827 KB
psmtest67.apk	28/10/2018 6:31 PM	APK File	132,829 KB
psmtest68.apk	28/10/2018 7:34 PM	APK File	132,840 KB
psmtest69.apk	11/11/2018 3:17 PM	APK File	132,842 KB
psmtest70.apk	11/11/2018 3:37 PM	APK File	132,841 KB
psmtest71.apk	11/11/2018 4:39 PM	APK File	132,841 KB
psmtest72.apk	11/11/2018 6:20 PM	APK File	132,572 KB
psmtest73.apk	22/11/2018 12:09 ...	APK File	143,636 KB
psmtest74.apk	22/11/2018 12:43 ...	APK File	143,635 KB
psmtest75.apk	22/11/2018 1:42 AM	APK File	143,635 KB
psmtest76.apk	23/11/2018 1:58 AM	APK File	143,636 KB
psmtest77.apk	23/11/2018 2:52 AM	APK File	143,636 KB
psmtest78.apk	24/11/2018 12:55 ...	APK File	143,941 KB
psmtest79.apk	24/11/2018 1:43 AM	APK File	143,941 KB
psmtest80.apk	24/11/2018 3:12 PM	APK File	143,941 KB
psmtest81.apk	24/11/2018 5:12 PM	APK File	143,636 KB
psmtest82.apk	25/11/2018 3:04 PM	APK File	143,621 KB
psmtest83.apk	26/11/2018 10:14 ...	APK File	145,689 KB

Figure 3.18 List of the application that have been tested in the smartphone.

3.3.4 Cutover

In Cutover stage, the markerless AR application developed in the Construction stage becomes operational in the Cutover stage. It also provides support to resolve any problems that arise immediately after the application becomes operational.

3.4 Work Breakdown Structure

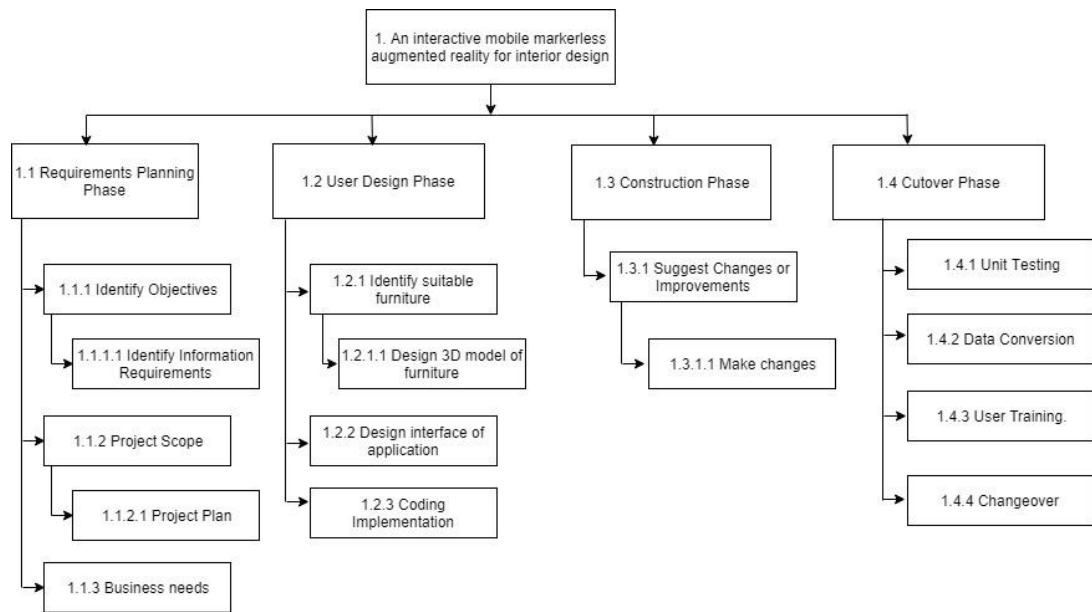


Figure 3.19 Work Breakdown Structure

3.5 Flowchart

Flowchart below show the flow from the beginning of the application until end of the application. It is start from open the application and they will find the decision making until user done with the application.

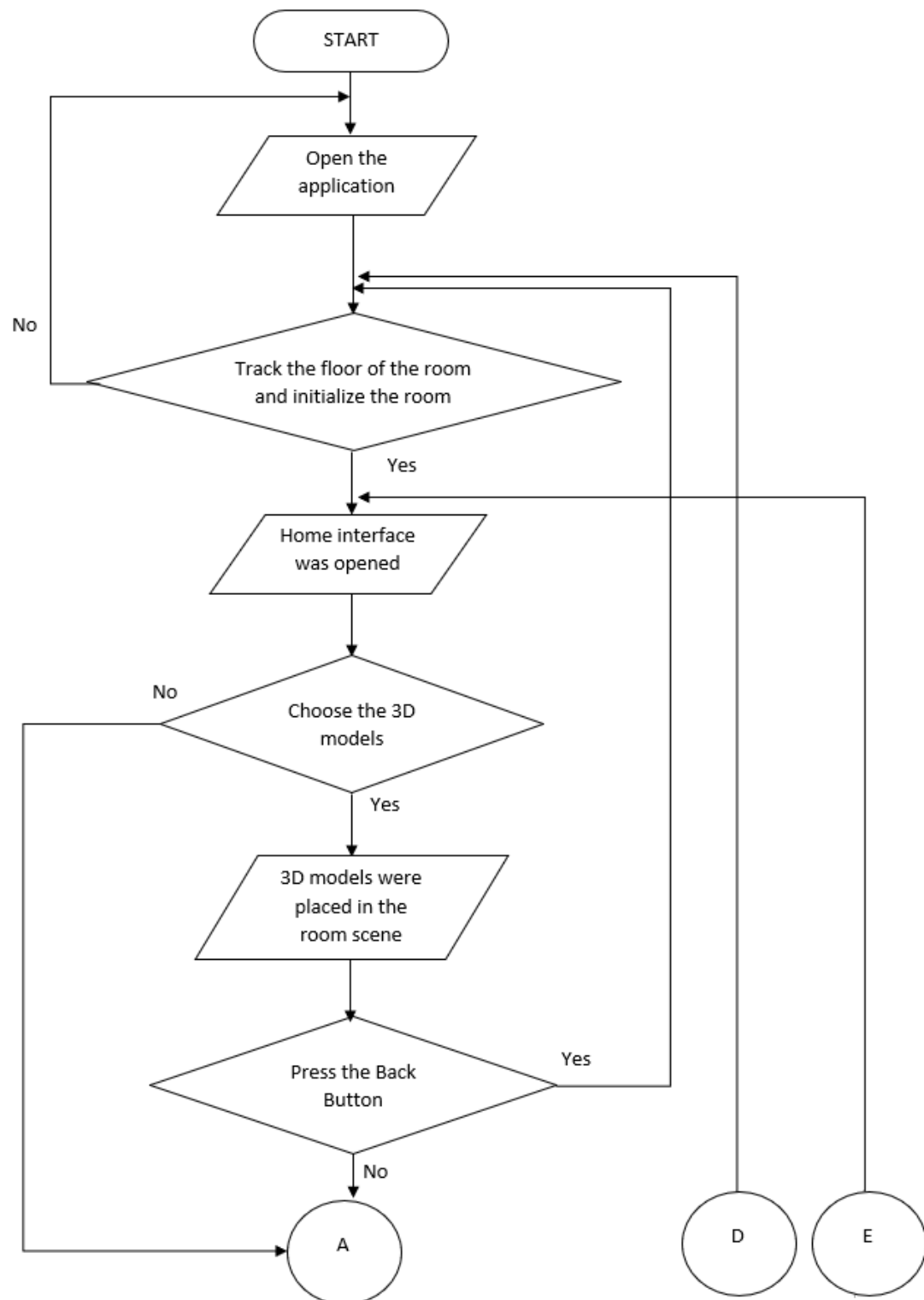


Figure 3.20 Flowchart to place the 3D models

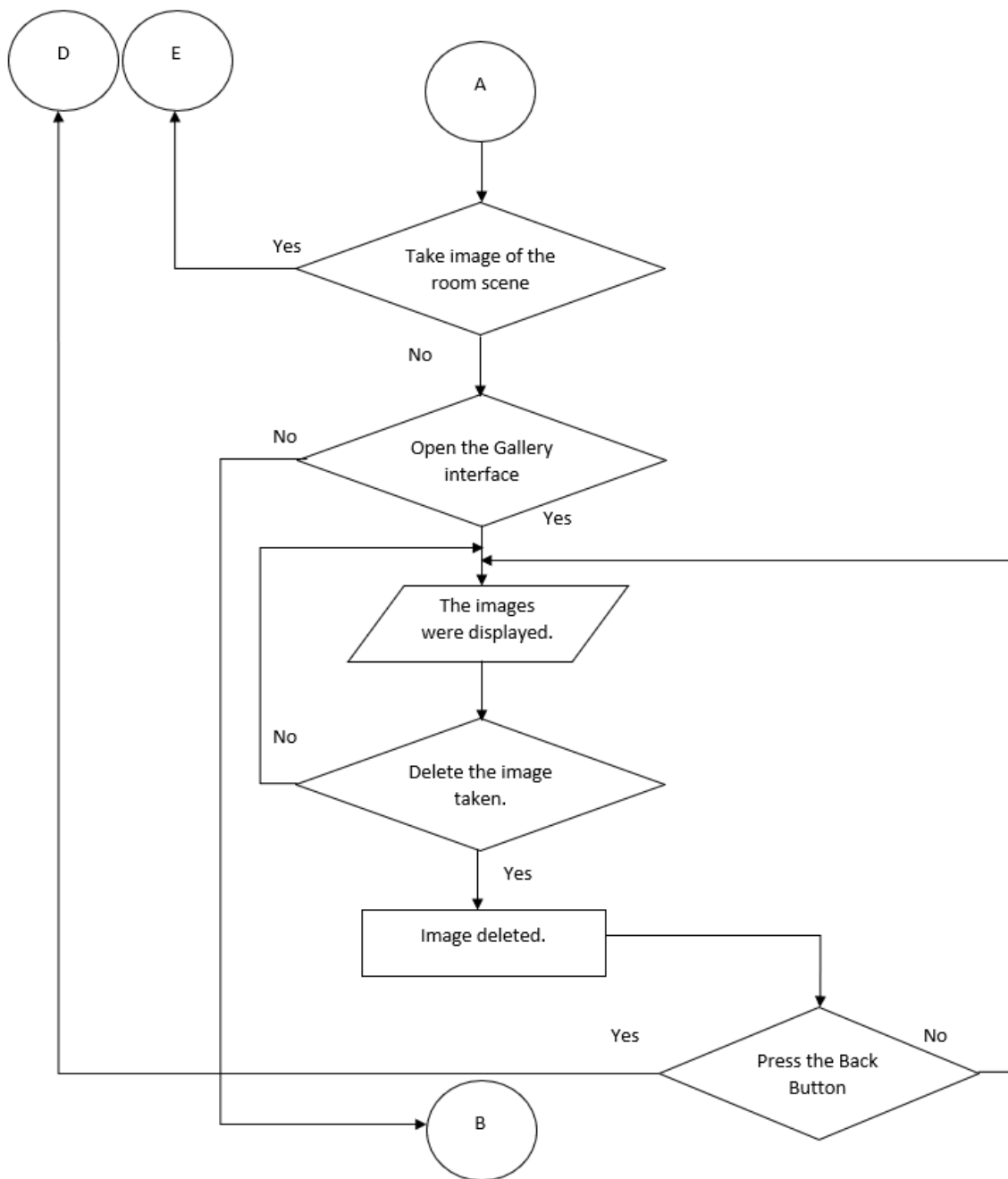


Figure 3.21 Flowchart to take image and view the image taken

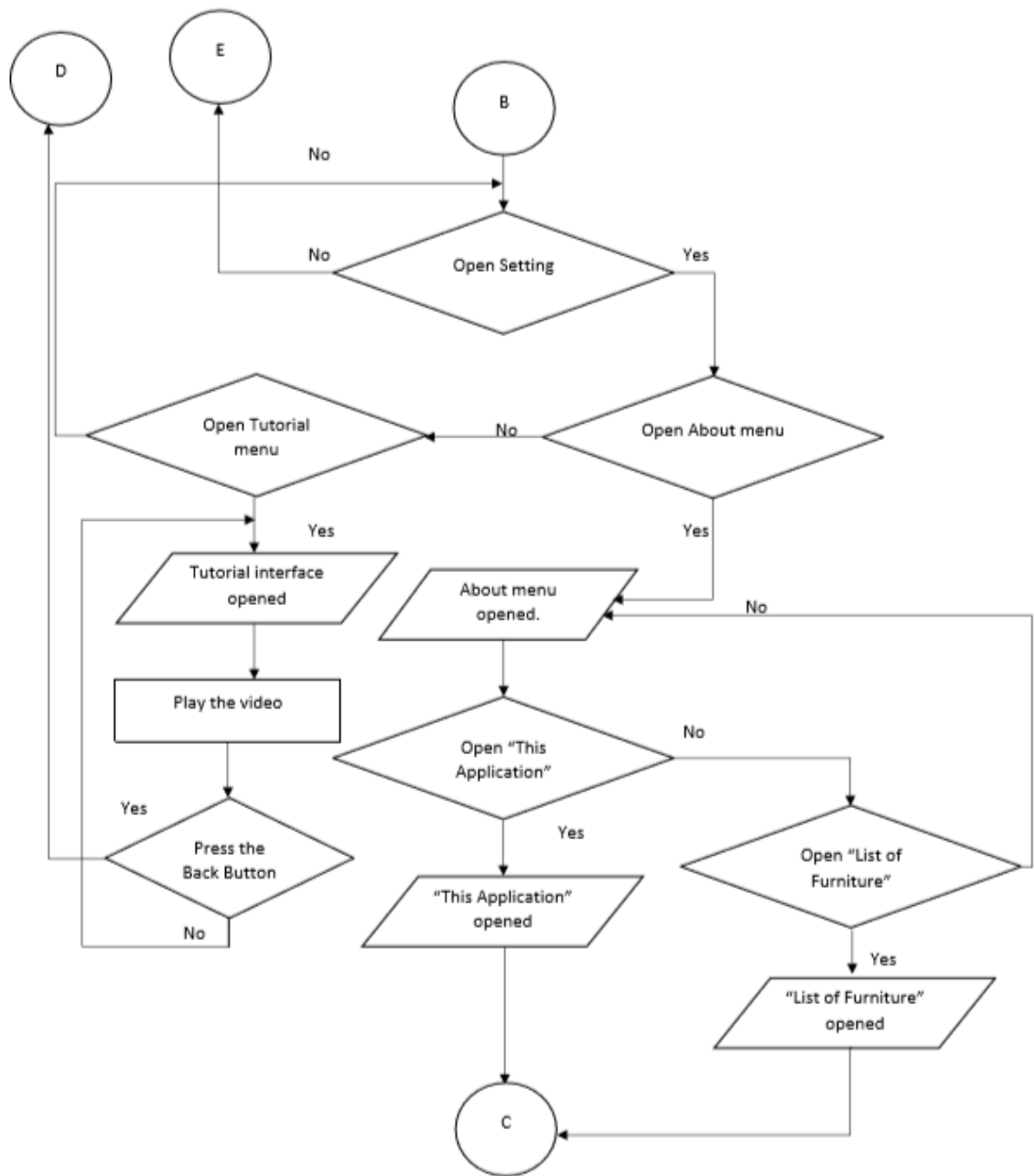


Figure 3.22 Flowchart to open setting

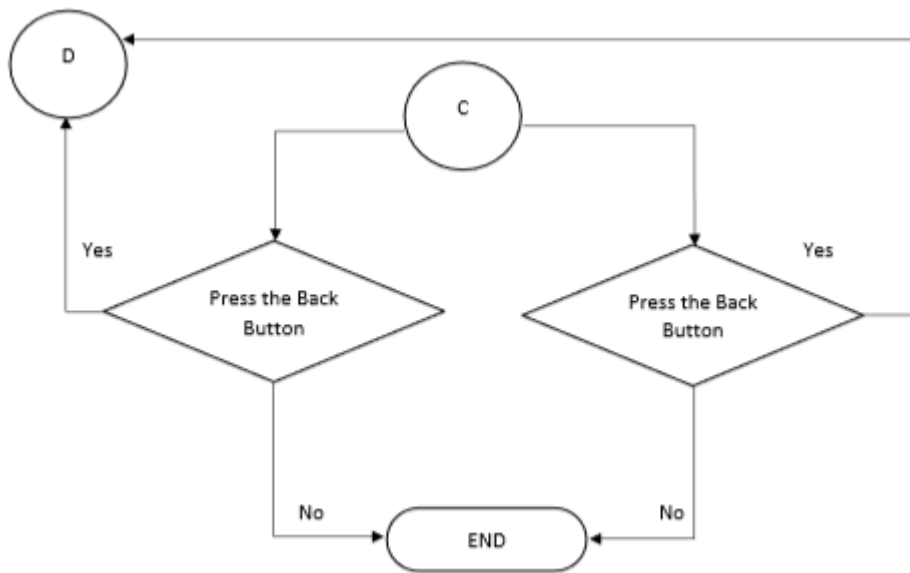


Figure 3.23 Flowchart to open setting (continue)

3.6 Context Diagram

Context Diagram below shows the tasks between developer, system of the application and the user.

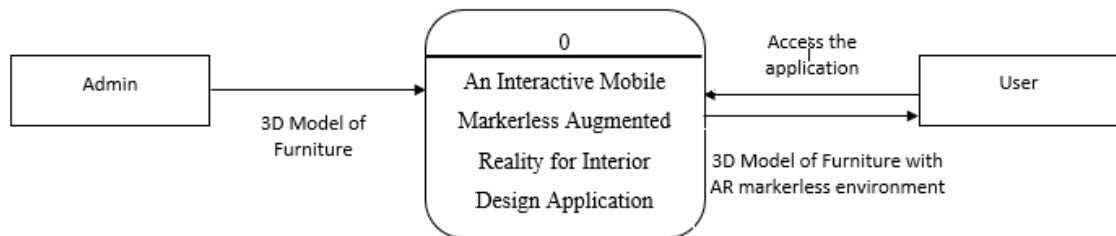


Figure 3.24 Context Diagram

3.7 Use Case Diagram

Use Case Diagram below shows the activities among the developer, system of the application and the user.

An Interactive Mobile Markerless Augmented Reality For Interior Design

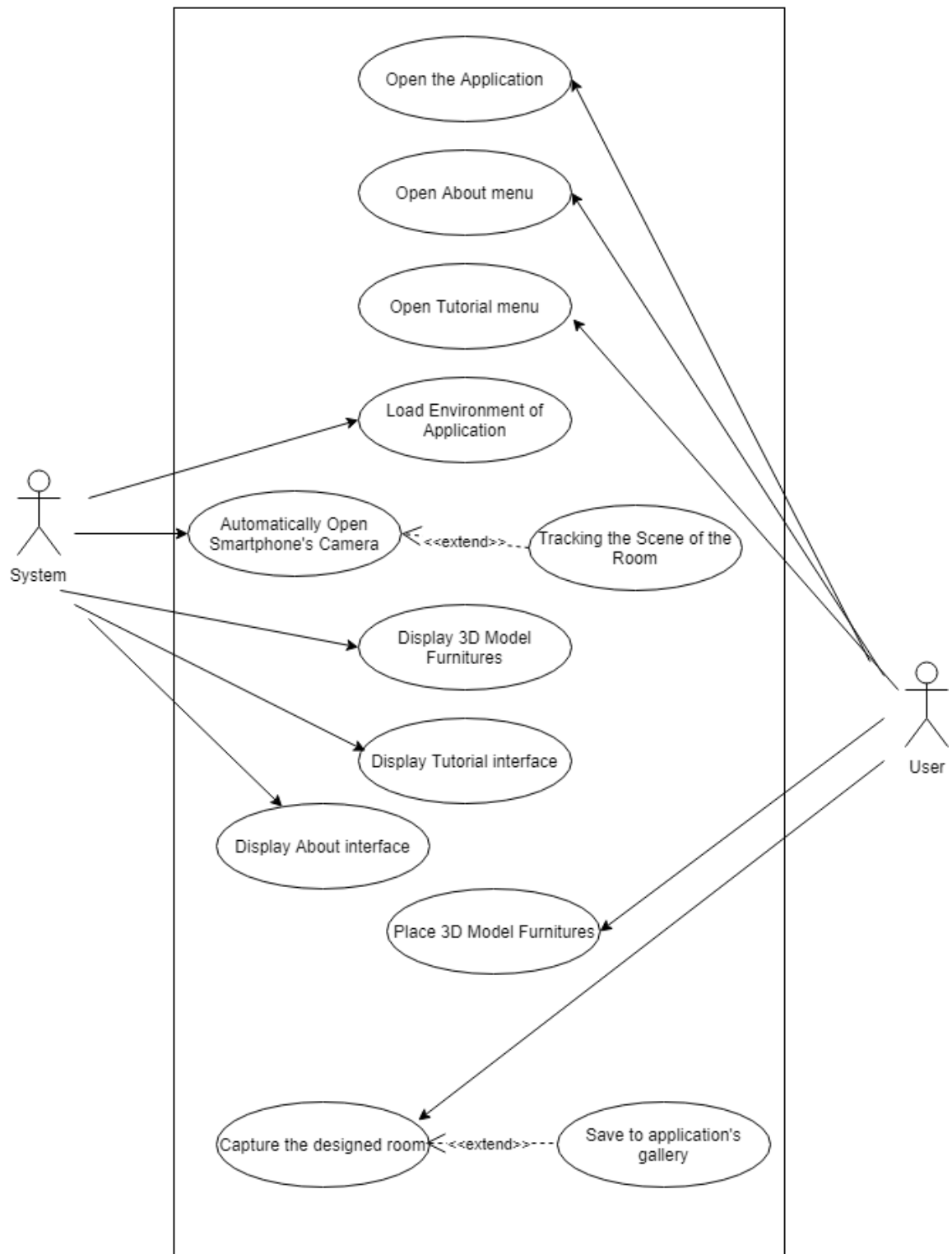



Figure 3.25 Use Case Diagram

3.8 Storyboard

In this section, propose storyboards will briefly describe based on specification and requirements.

3.8.1 First Interface


Table 3.2 First Interface

Screen Title	Main screen
Layout	01
Screen	 <p>Figure 3.26 First Interface</p>
Screen Description	In this screen, it will track or find the suitable place in the room before user are enabling to place the furniture. There are 3 button function which are application's gallery, camera, and setting button.

3.8.2 Second Interface

Table 3.3 Second Interface

Screen Title	Home screen
Layout	02

Screen	 <p data-bbox="542 636 919 667">Figure 3.27 Second Interface</p>	List of 3D Model of Furnitures	Reset
Screen Description	<p>User can freely place the 3D furniture.</p> <p>User also can change to another furniture and change colour. Hence, it will be providing reset button to clear up the screen.</p>		

3.8.3 Third Interface

Table 3.4 Third Interface

Screen Title	Setting
Layout	03
Screen	

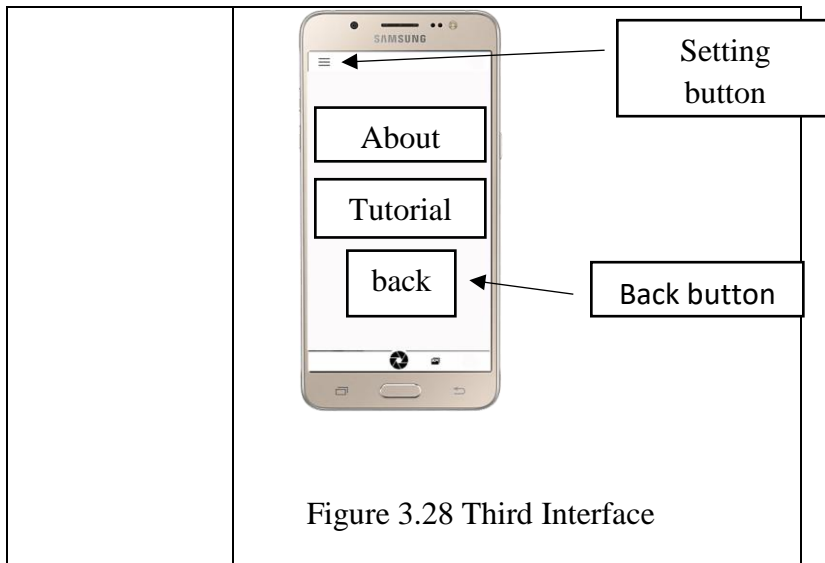


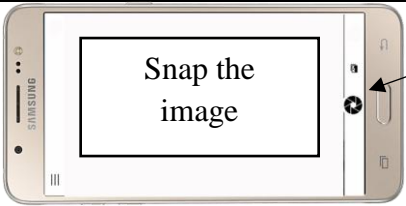
Figure 3.28 Third Interface

<p>Screen Description</p>	<p>In setting function, it will provide Tutorial, About, and Back button. Tutorial button will cover about how to use the application for the user. For About button will cover a simple description of the application to give general ideas to user. Back button is to back to the home screen.</p>
----------------------------------	---

3.8.4 Fourth Interface

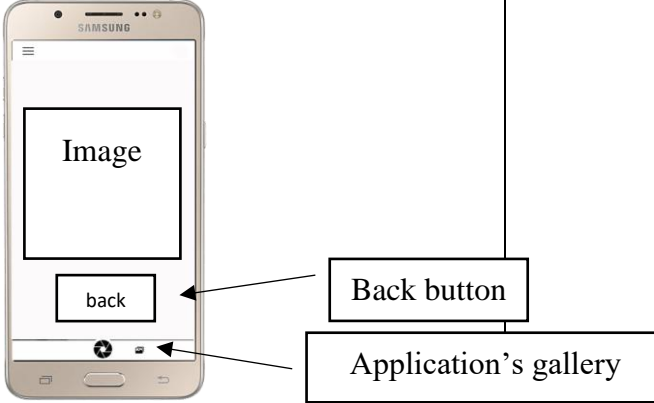
Table 3.5 Fourth Interface

<p>Screen Title</p>	<p>Camera</p>
<p>Layout</p>	<p>04</p>
<p>Screen</p>	

	 <p style="text-align: center;">Figure 3.29 Fourth Interface</p>	<p style="text-align: center;">Camera</p>
<p>Screen Description</p>	<p>User can freely snap the image and it will automatically save to application's gallery.</p>	

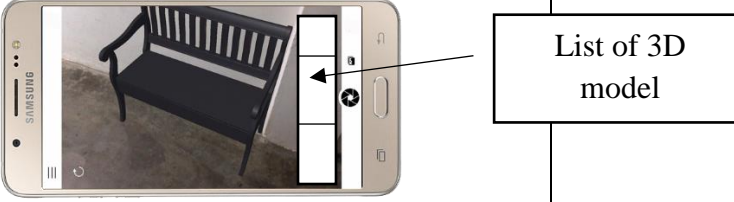
3.8.5 Fifth Interface

Table 3.6 Fifth Interface

<p>Screen Title</p>	<p>Application's gallery</p>	
<p>Layout</p>	<p>05</p>	
<p>Screen</p>	 <p style="text-align: center;">Figure 3.30 Fifth Interface</p>	
<p>Screen Description</p>	<p>User can preview the images that they had taken before.</p>	

3.8.6 Sixth Interface

Table 3.7 Sixth Interface

Screen Title	Furnitures
Layout	06
Screen	 <p>Figure 3.31 Sixth Interface</p>
Screen Description	User can scroll to choose the suitable furniture and various colour of furniture. The furniture will place into the scene of room once user tab on it.

3.9 Hardware and Software Requirement

The list of software and purpose to develop an interactive mobile markerless augmented reality for interior design will be discuss in this section.

3.9.1 Software Requirement

Table 3.8 shows the software that need in developing the markerless AR application.

Table 3.8 Software Requirements

Software	Purpose
Autodesk Maya 2016	Use for developing 3D models.
Unity	Use for creating the application.
Adobe Photoshop CS5	Use for editing the images.
Wikitude SDK	Use for instant tracking.
Microsoft Word 2016	Use for writing the FYP documentation.
Microsoft Project 2016	Use for creating Gantt Chart.
Draw.io	Use for drawing the diagrams.
Adobe Acrobat Reader DC	Use for reading the PDF file.
VideoPad Video Editor	Use for editing video.

3.9.2 Hardware Requirement

Table 3.9 shows the software that need in developing the markerless AR application.

Table 3.9 Hardware Requirements

Hardware	Requirement	Purpose
HP Notebook (laptop)	<ol style="list-style-type: none"> 1. Processor <ul style="list-style-type: none"> • Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz 1.70 GHz 2. Installed RAM <ul style="list-style-type: none"> • 4.00 GB 	Use for developing the application and documentation.

	<p>3. System type</p> <ul style="list-style-type: none"> • 64-bit operating system, x64-based processor 	
Smartphone	<p>1. Name</p> <ul style="list-style-type: none"> • Samsung S8+ <p>2. Internal Storage</p> <ul style="list-style-type: none"> • 64.00 GB <p>3. Android version</p> <ul style="list-style-type: none"> • 8.0.0 (Oreo) <p>4. RAM</p> <ul style="list-style-type: none"> • 4 GB <p>5. Processor</p> <ul style="list-style-type: none"> • Octa-core (2.35GHz Quad + 1.9GHz Quad), 64 bit, 10nm processor 	Use to run the application and test the application's compatibility.
Printer	Canon E560	To print the documentation.

3.10 Gantt Chart

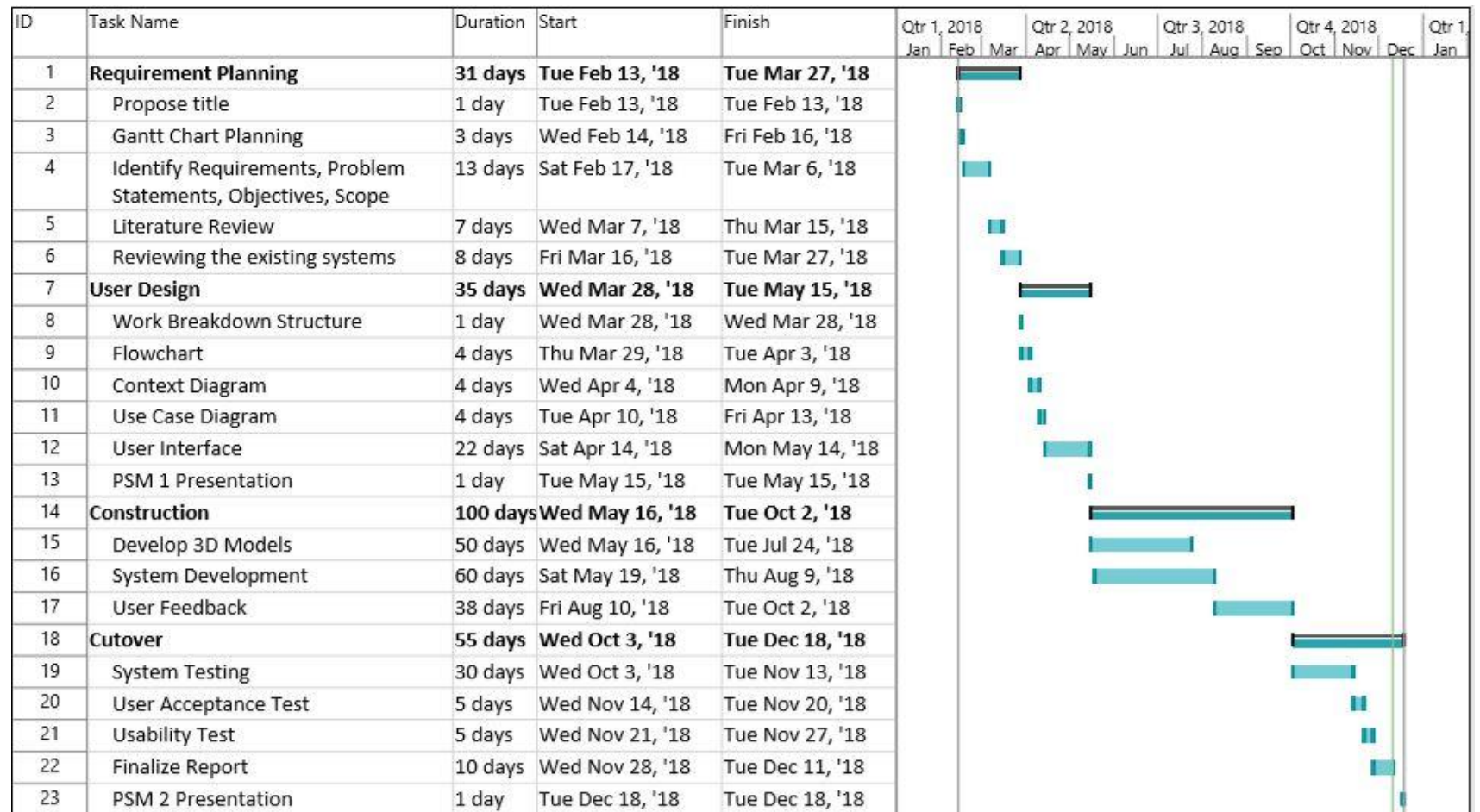


Figure 3.32 Gantt Chart

3.11 Implementation

This application will be implementing to target user of this application that have been specific in earlier phase. This application is not only can be used for interior designer, but it can be used for everyone for daily use. It is more specific for adult user, so it can assist in furniture arrangements. Children also can use the application. User can find the application at the Play Store because it is only support for android user with specific android versions. 26 people will involve in this implementation of application.

CHAPTER 4

IMPLEMENTATION, TESTING, RESULTS AND DISCUSSION

4.1 Introduction

This chapter will discuss the whole development process of An Interactive Mobile Markerless Augmented Reality for Interior Design. It contains how the application is developed from the scratch and what does the processes take part in the application. The software that has been used in the whole development are Autodesk Maya 2016, Unity 3D, Adobe Photoshop CS5, Wikitude SDK, and VideoPad Video Editor. The coding language used in this application is C#.

4.1.1 Development Tools

The development tools are a software or computer program that help software developers create, develop, debug, and maintain the application. The development tools that used in this project are Autodesk Maya 2016, Unity 3D, Adobe Photoshop CS5, Wikitude SDK, and VideoPad Video Editor.

Autodesk Maya 2016 is a 3D computer graphics application that can be runs on Windows, Linux, and macOS. It is used to create interactive 3D models of applications, including animated film, video games, visual effects, or TV series. Using this software, all 3D models of furniture has been created towards the development process. After finish

create the 3D models, the file must be export to .fbx file in order to import it into the Unity 3D for the next phase of development.

Unity 3D includes 2D and 3D scene design tools, cinematics and storytelling, audio system, lighting, particle effects, Sprite management tools and a powerful dopesheet animation system. Unity is a cross-platform game engine developed by Unity Technologies. The application was developed by using Unity 3D. The main function and the system have been developed using Unity 3D combined with the Wikitude SDK. All the 3D models from the Autodesk Maya 2016 was imported into Unity 3D.

Adobe Photoshop is a popular Adobe graphics editor. Adobe Photoshop is known as a raster graphics editor, a computer program that allows users to create and edit images in one of several formats. Logos, texts, and images in the application were created using this software.

Wikitude SDK is a software library and framework that used to create augmented reality world for mobile application. Wikitude SDK supports SLAM (Simultaneous Localization and Mapping) is a technology which understands the physical world through feature points. This allows AR applications to recognize 3D objects and scenes and instantly track the world and overlay digital interactive augmentations. To use the SDK, a valid license is needed to run the application.

VideoPad Video Editor is a free video editing application. This software is used formatting editing the tutorial video in this project.

4.2 Implementation

The implementation used in this project in order to make this project successful such as the step by step to create the 3D models and the application.

4.2.1 The 3D Models

There are the 3D models that have been created in this project using Autodesk Maya 2016. 6 categories of furniture were decided to be implement in the application such as bookshelf, cabinet, chair, lamp, sofa, and table. For each categories of furniture consist of 3 different designs of the furniture. For the bookshelf's category, was named as Minimalist Bookshelf, Normal Bookshelf, and Stackable Bookshelf as shown in Figure 4.1, Figure 4.2, and Figure 4.3 respectively.

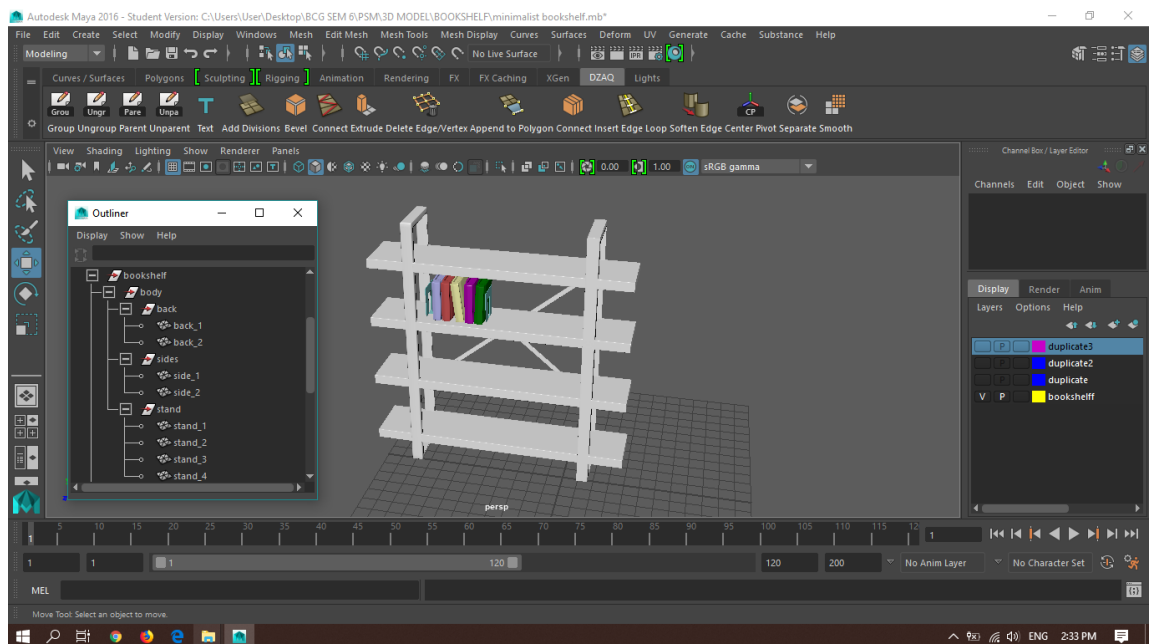


Figure 4.1 Minimalist Bookshelf

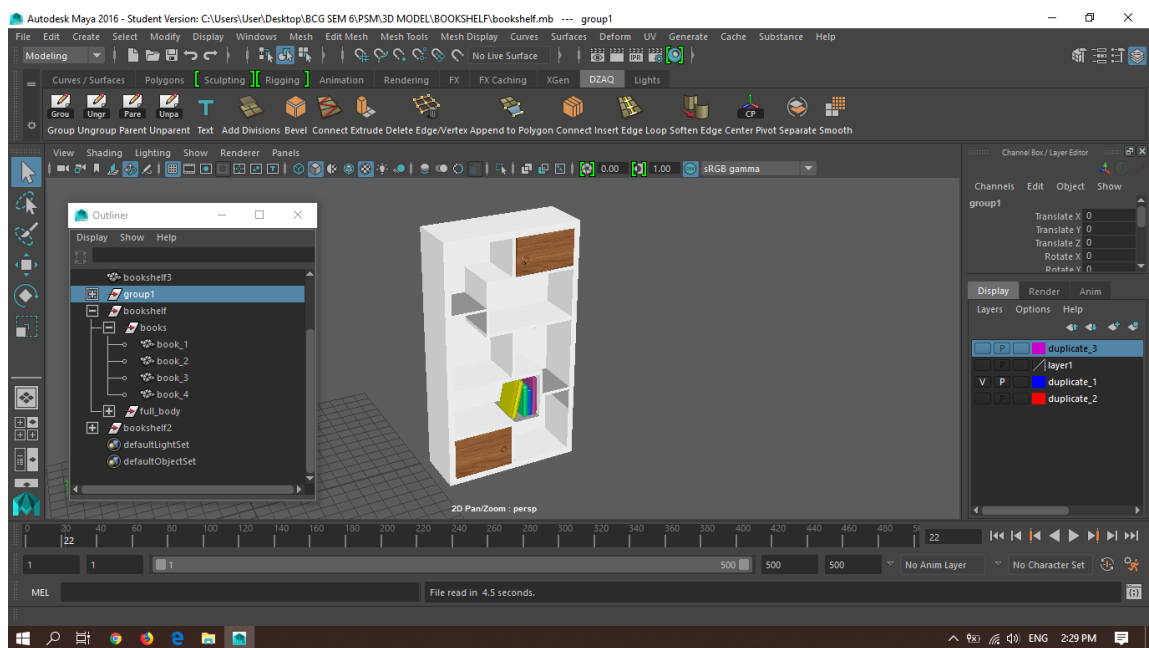


Figure 4.2 Normal Bookshelf

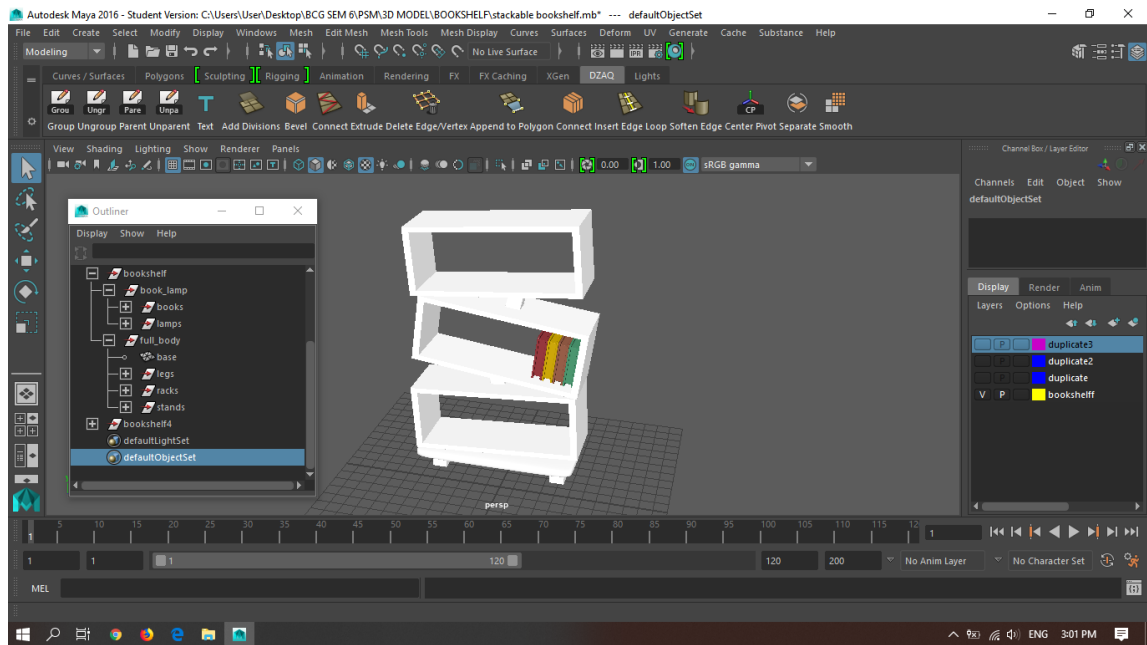


Figure 4.3 Stackable Bookshelf

For the cabinet category, there have Accent Floor Cabinet, Semi Classic Cabinet, and Unique Drawer Cabinet as shown in Figure 4.4, Figure 4.5, and Figure 4.6 respectively.

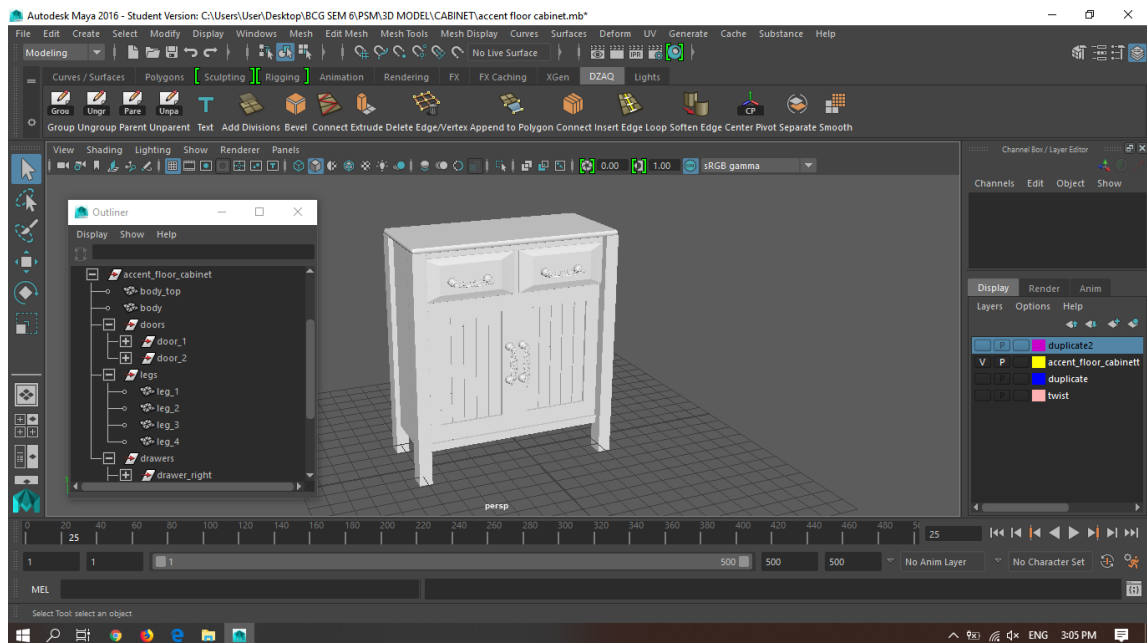


Figure 4.4 Accent Floor Cabinet

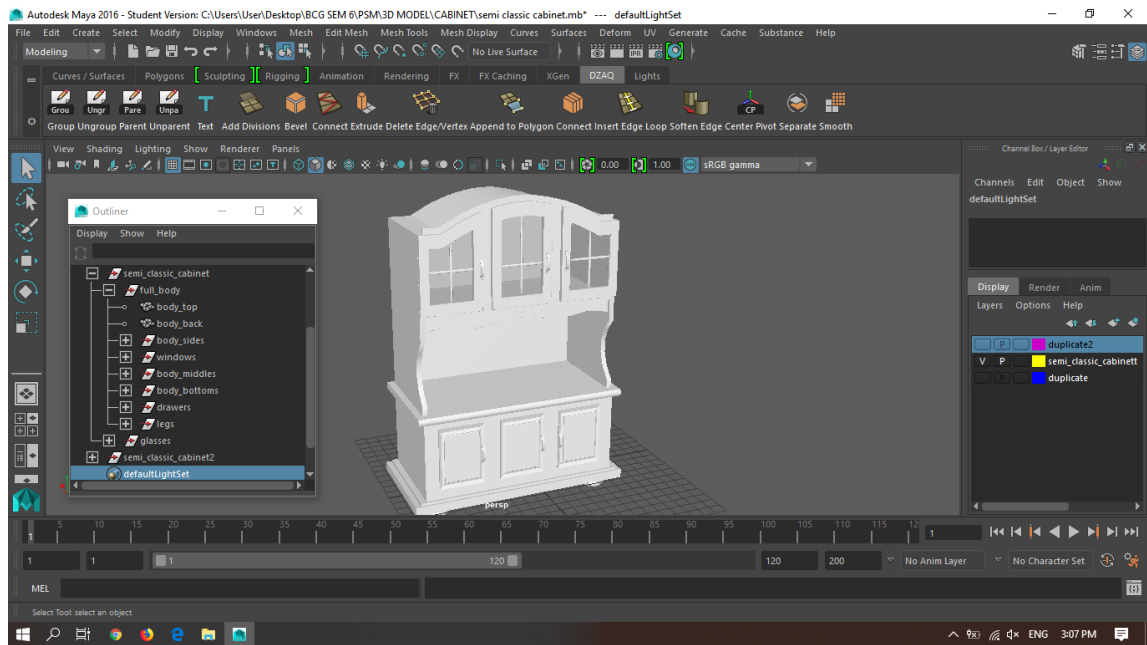


Figure 4.5 Semi Classic Cabinet

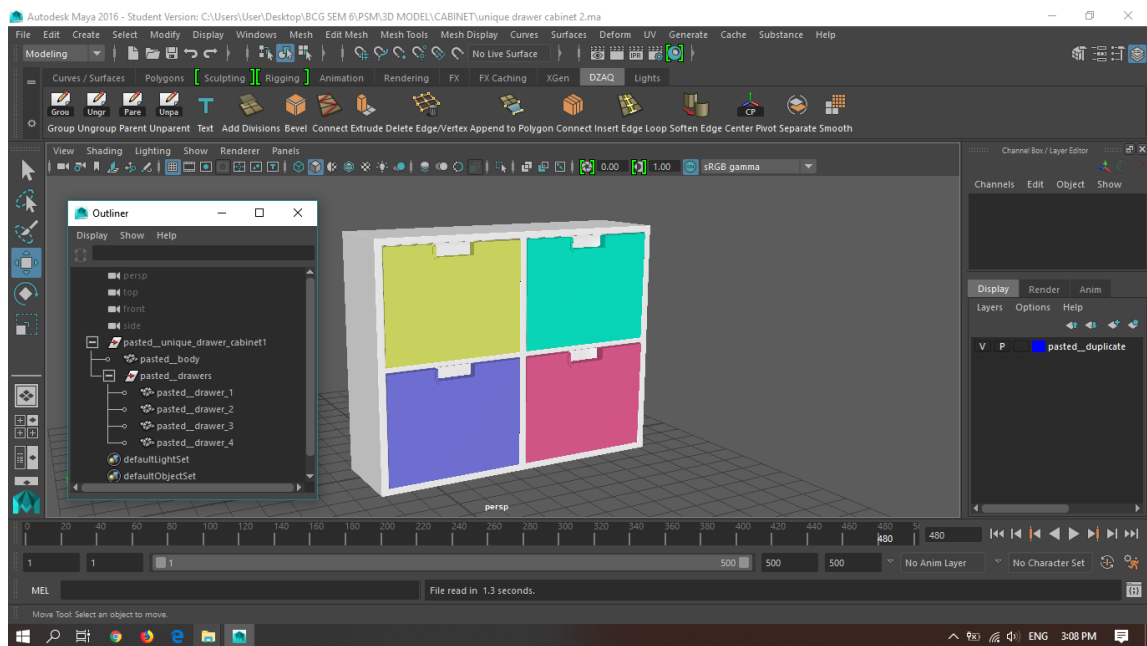


Figure 4.6 Unique Drawer Cabinet

For the chair category, there have Basic Chair, Fabric Chair, and Stool as shown in Figure 4.7, Figure 4.8, and Figure 4.9 respectively.

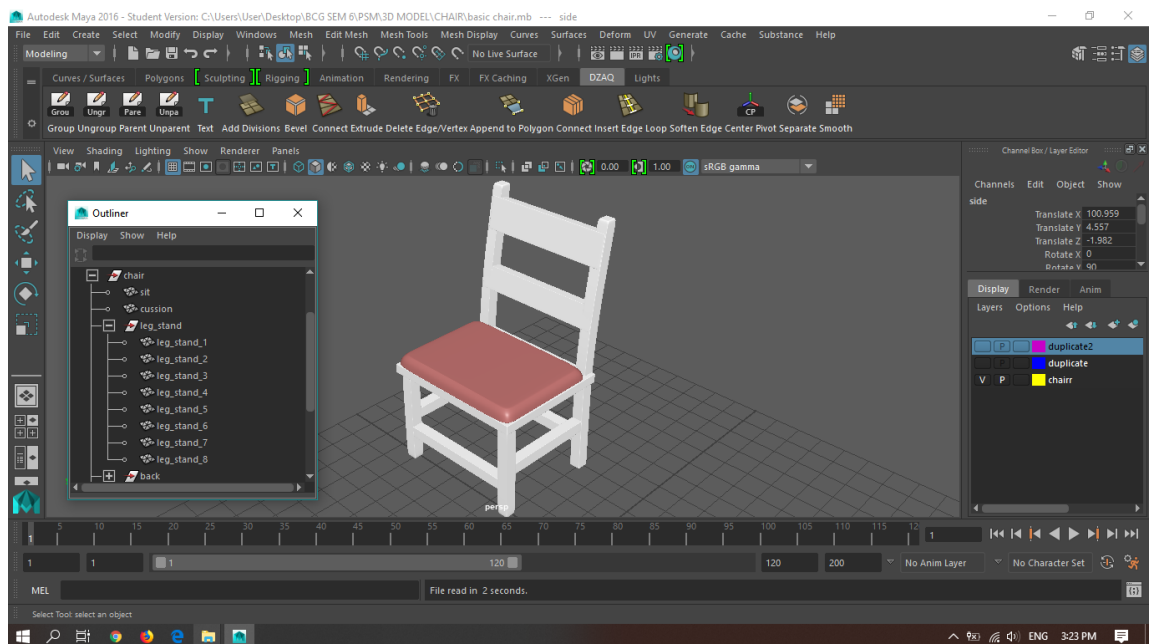


Figure 4.7 Basic Chair

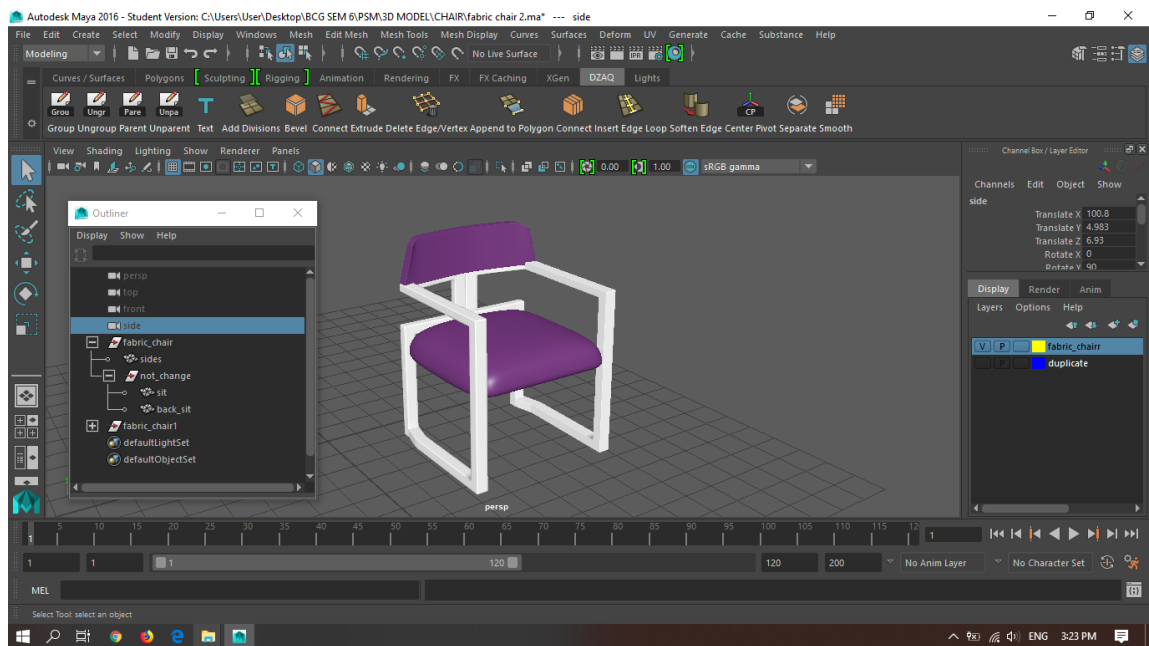


Figure 4.8 Fabric Chair

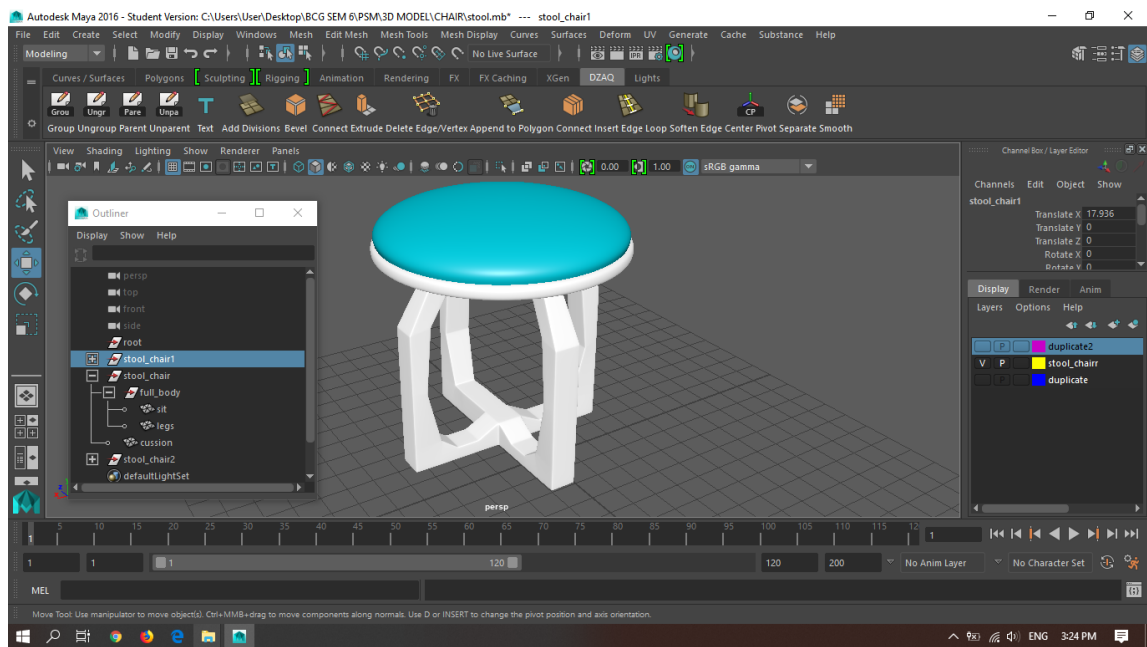


Figure 4.9 Stool

For the lamp category, there have Table Lamp, Touch Lamp, and Two Lamp as shown in Figure 4.10, Figure 4.11, and Figure 4.12 respectively.

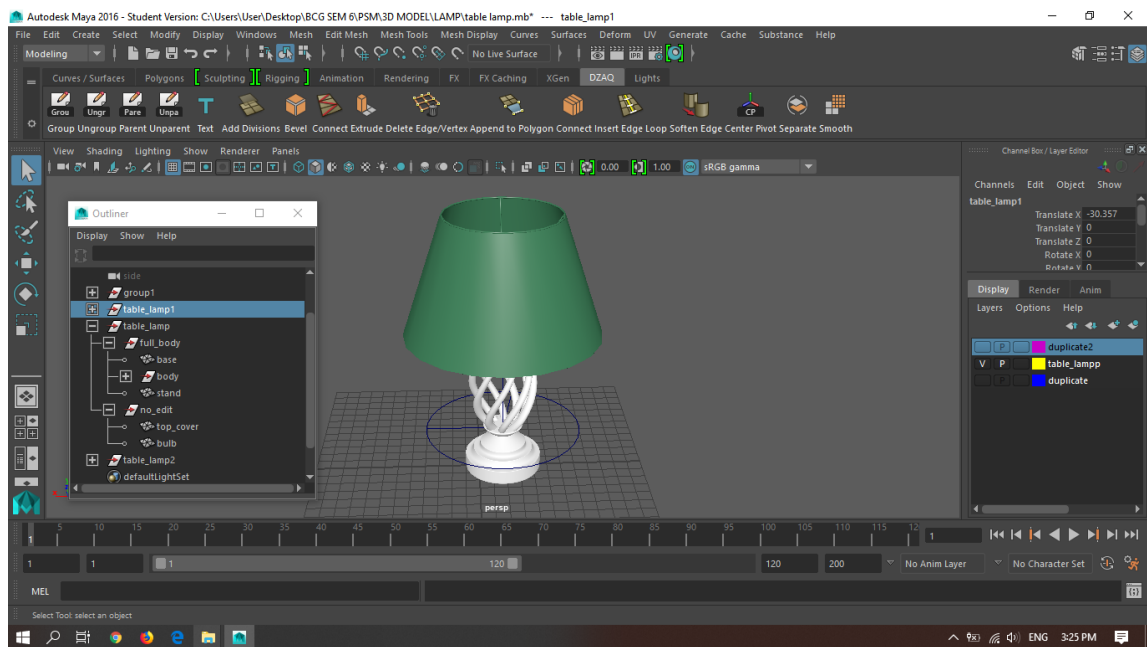


Figure 4.10 Table Lamp

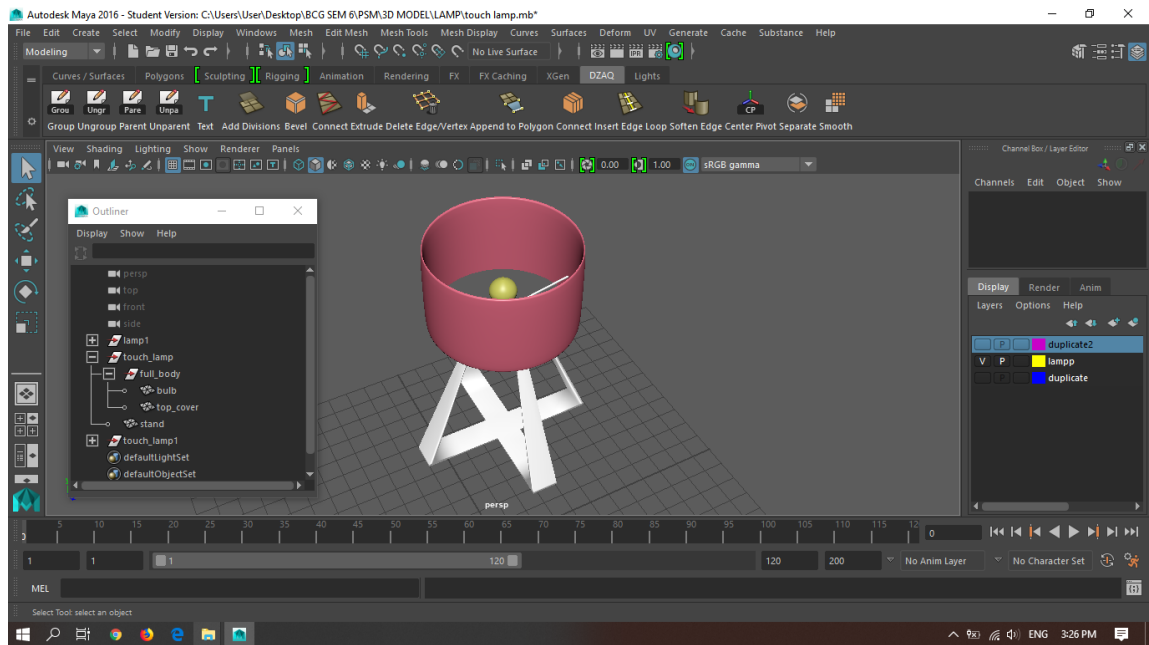


Figure 4.11 Touch Lamp

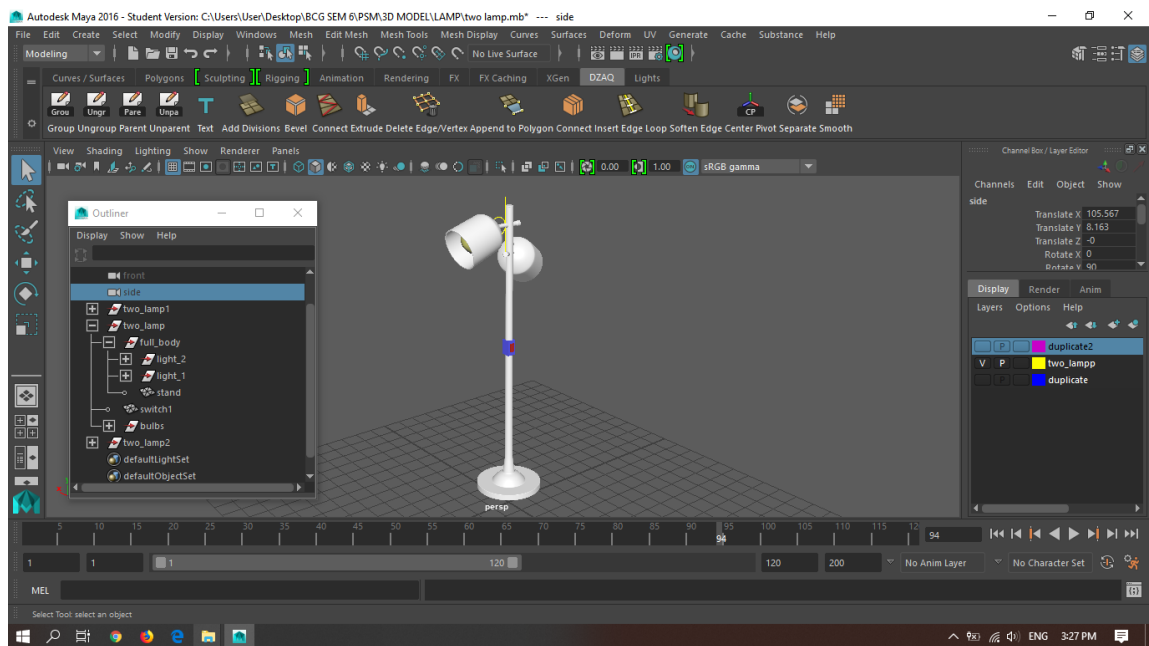


Figure 4.12 Two Lamp

For the sofa category, there have Modern Sofa, Sectional Sofa, and Wooden Arm Sofa as shown in Figure 4.13, Figure 4.14, and Figure 4.15 respectively.

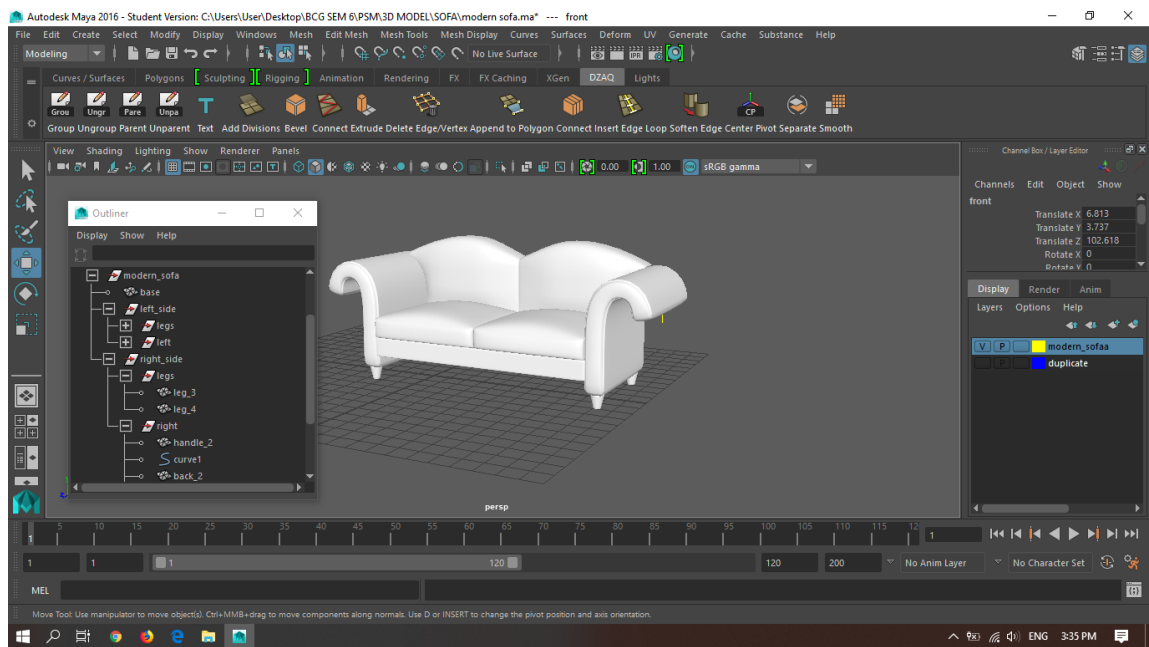


Figure 4.13 Modern Sofa

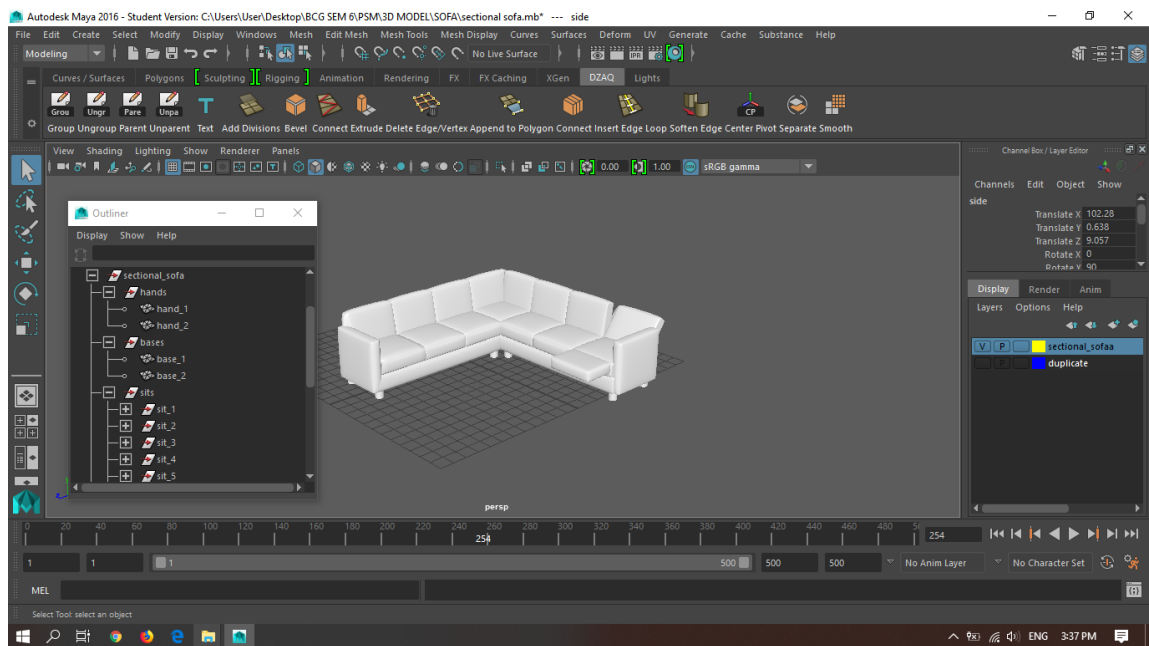


Figure 4.14 Sectional Sofa

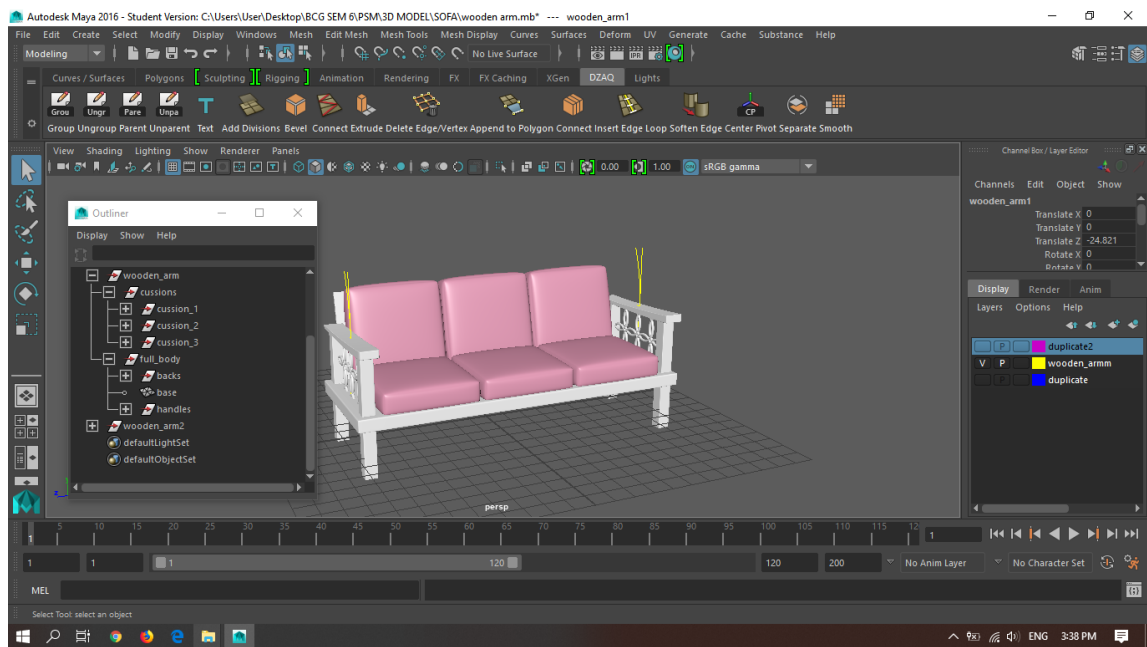


Figure 4.15 Wooden Arm Sofa

For the table category, there have Coffee Table, Dining Table, and Round Table as shown in Figure 4.16, Figure 4.17, and Figure 4.18 respectively.

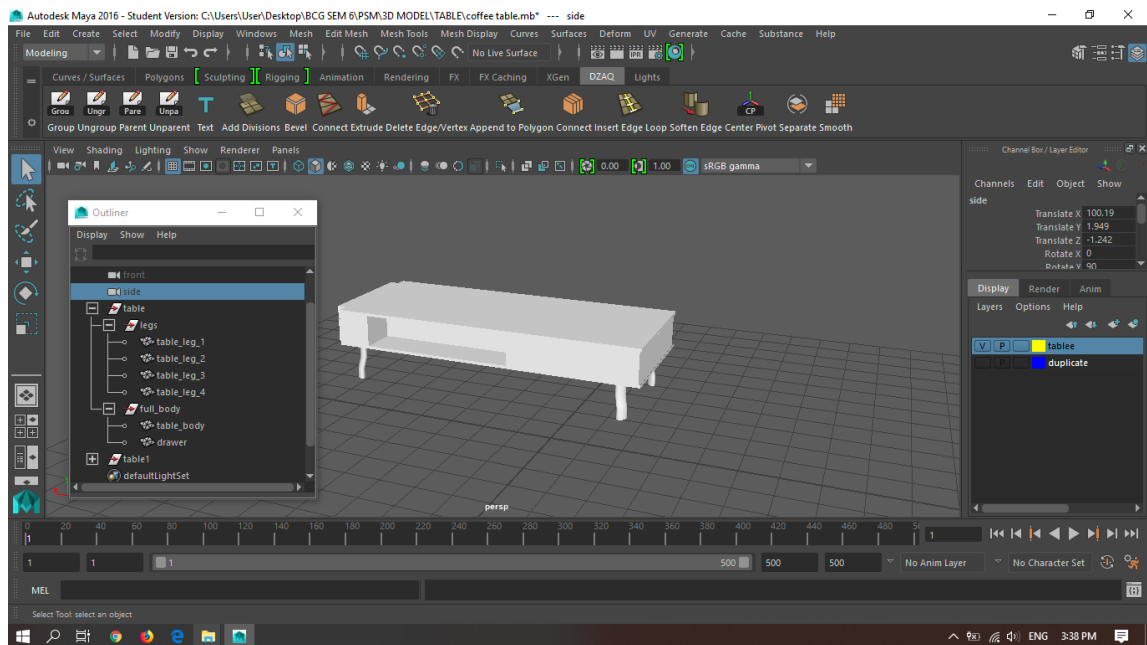


Figure 4.16 Coffee Table

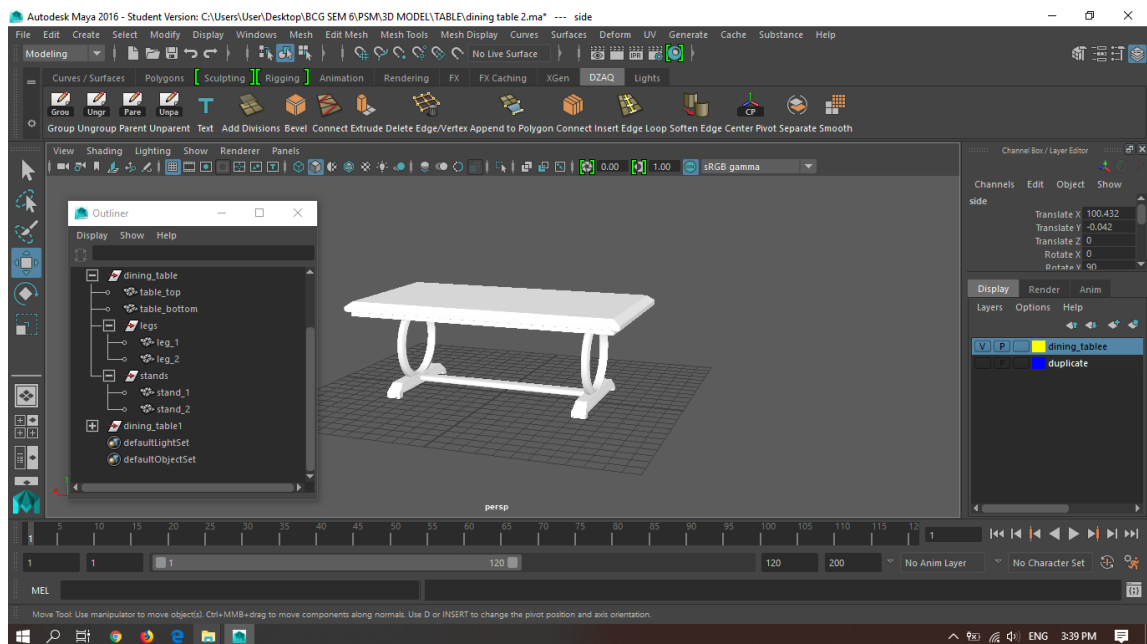


Figure 4.17 Dining Table

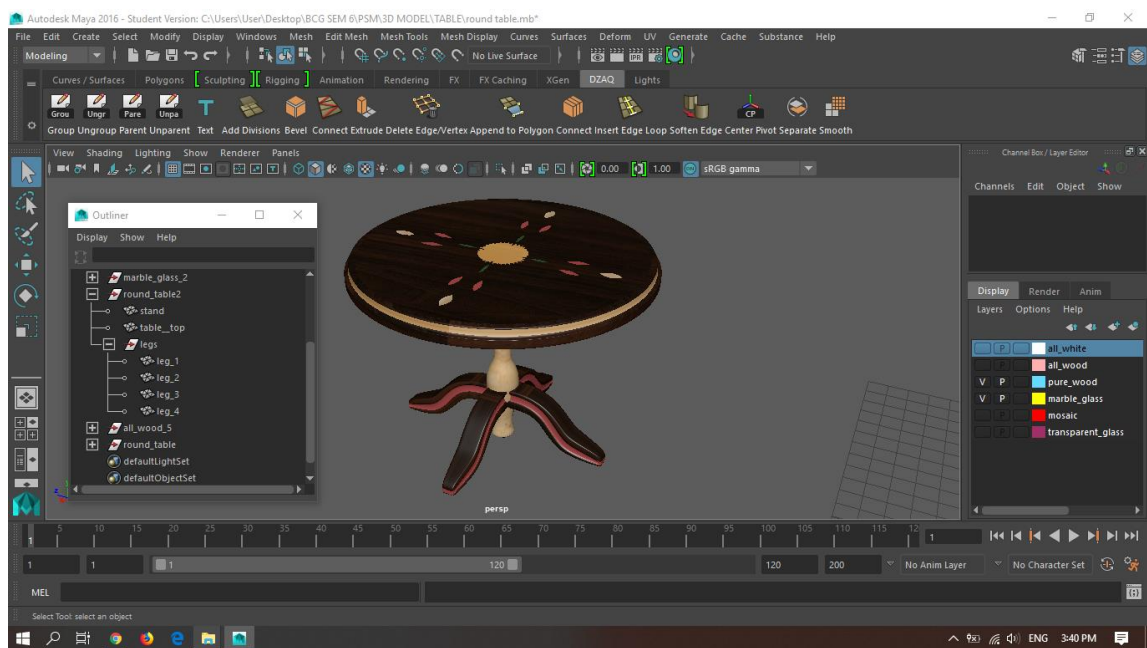


Figure 4.18 Round Table

4.2.2 Application Development

There will be five interfaces that are created in this application using Unity 3D. There was for Tracking interface, Home interface, Gallery interface, About interface, and

Tutorial interface as shown in Figure 4.19 and Figure 4.20 for Tracking interface, Figure 4.21 for Home interface, Figure 4.22 for Gallery interface, Figure 4.23 and Figure 4.24 for About interface and lastly Figure 4.25 for Tutorial interface.

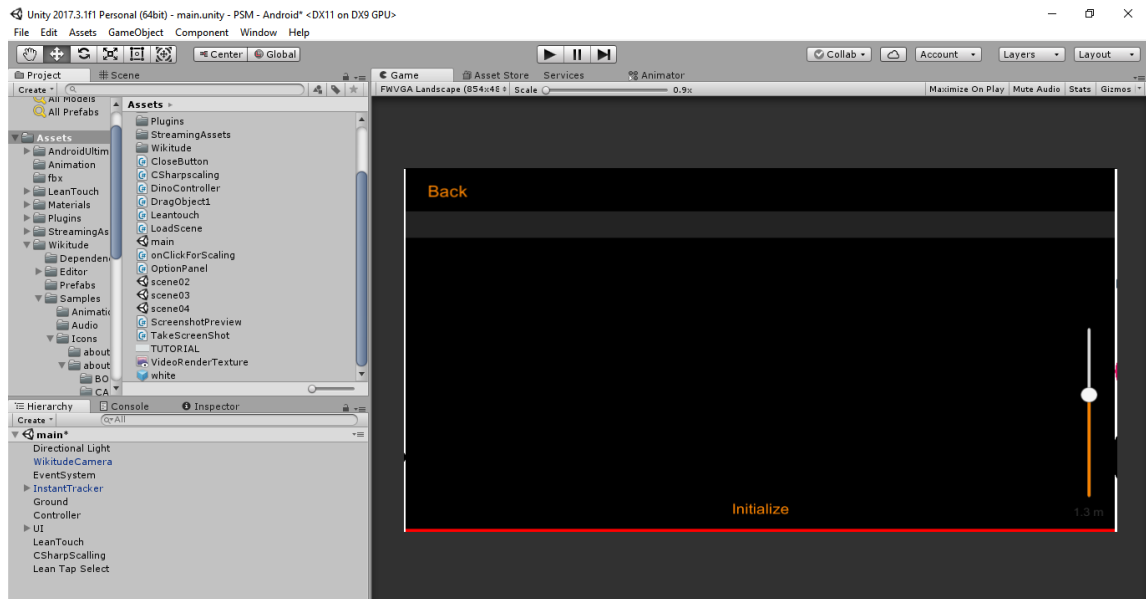


Figure 4.19 Tracking interface before run the application

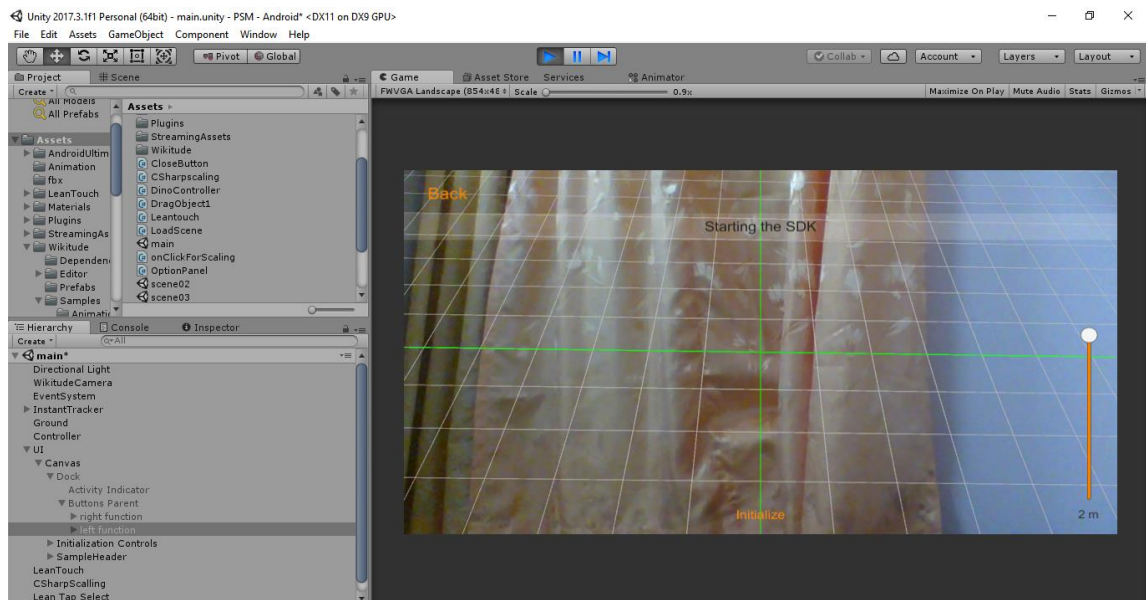


Figure 4.20 Tracking interface after running the application

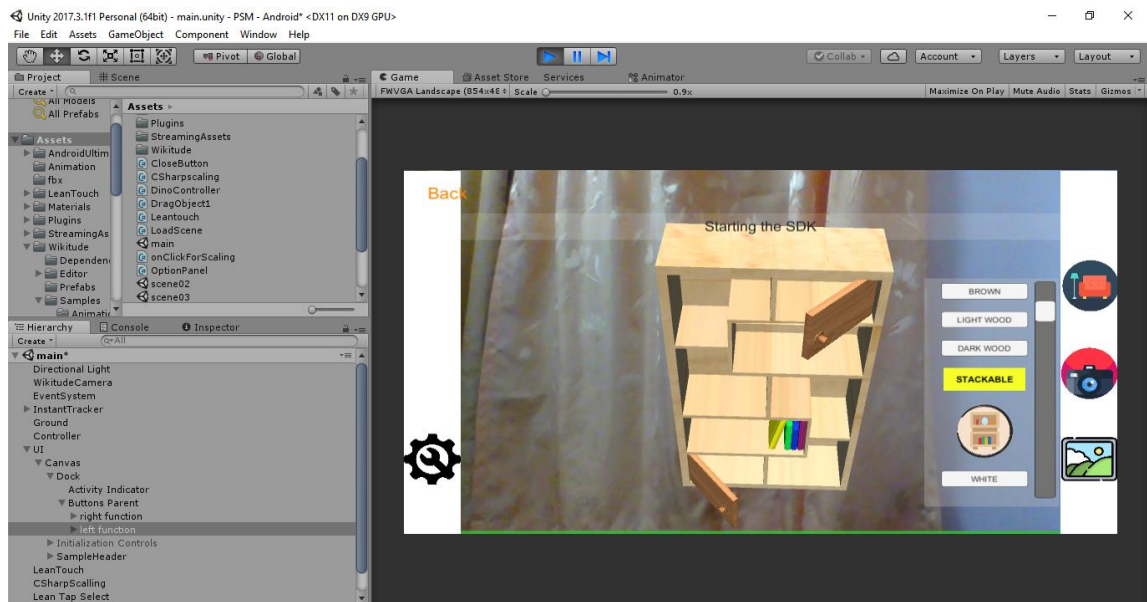


Figure 4.21 Home interface

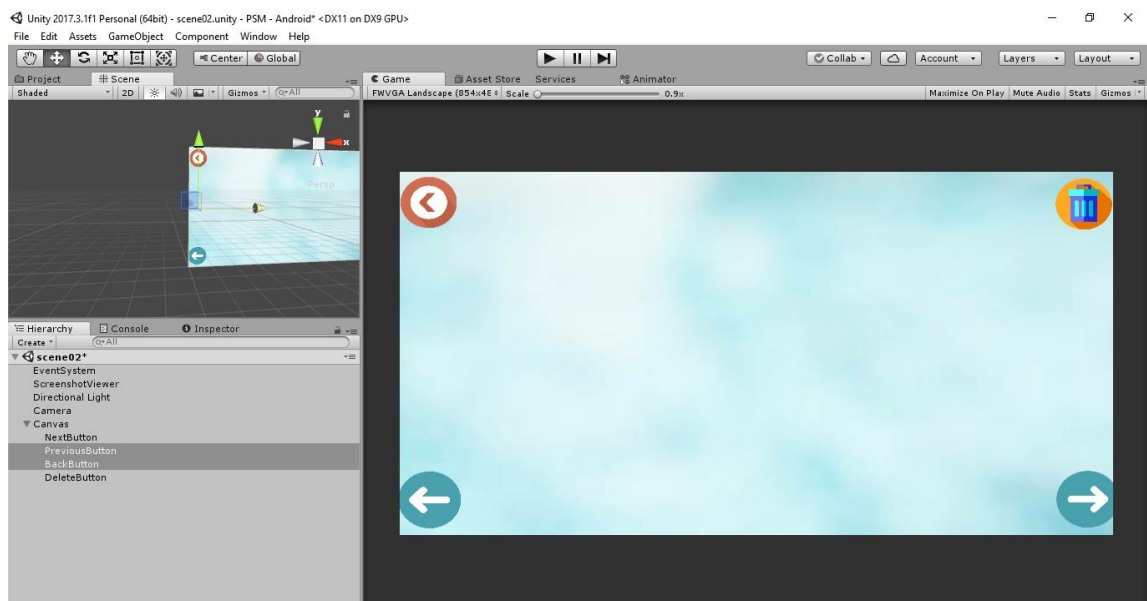


Figure 4.22 Gallery interface

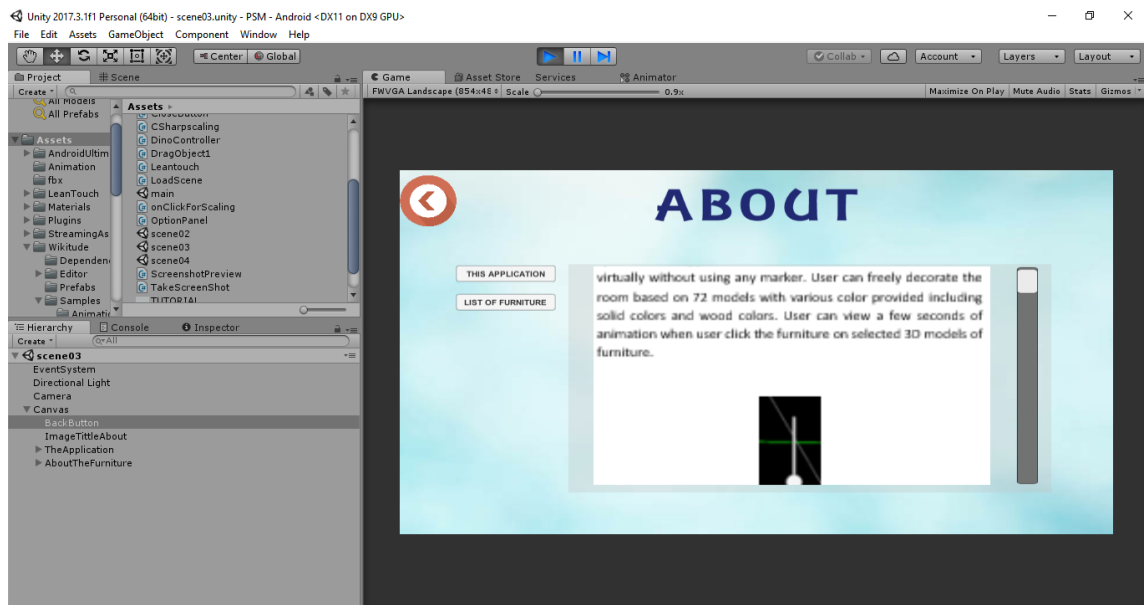


Figure 4.23 About interface for this application

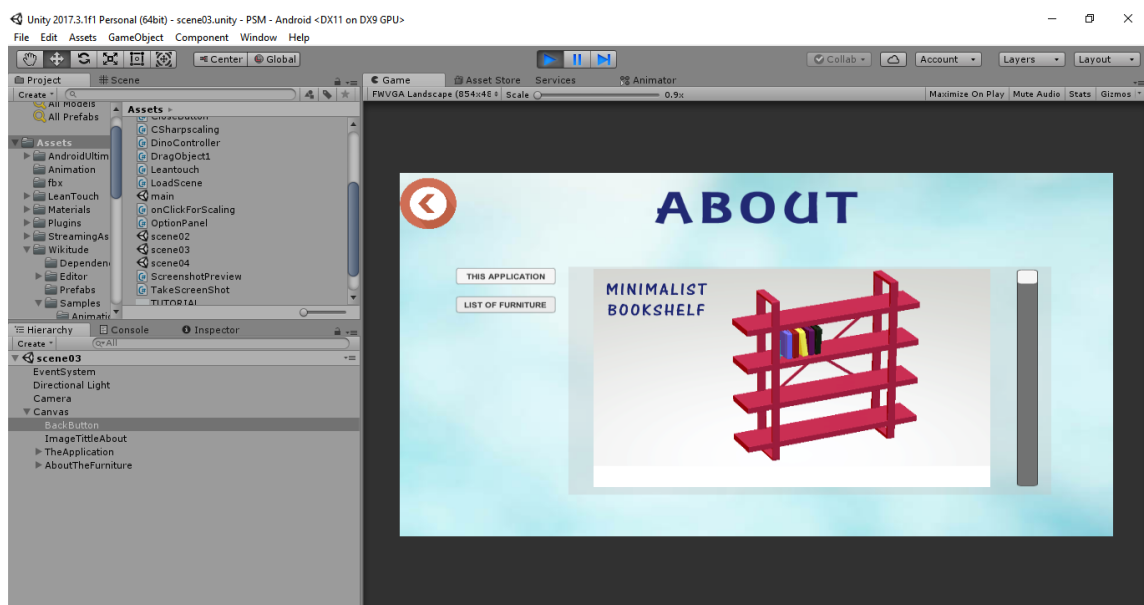


Figure 4.24 About interface for list of furniture

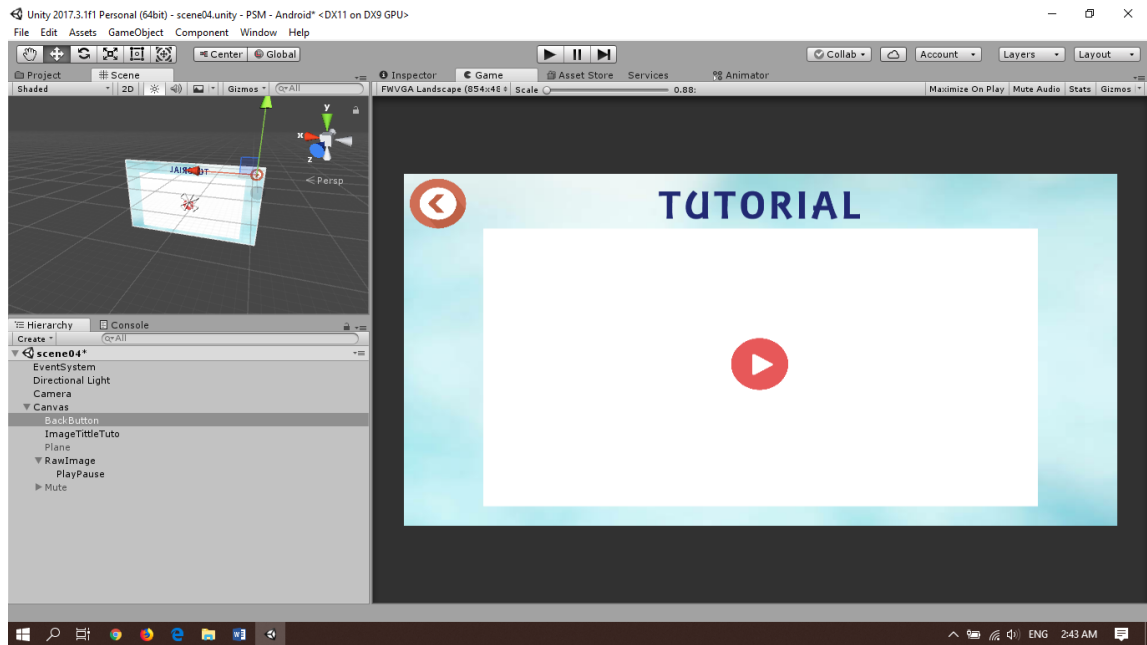


Figure 4.25 Tutorial interface

The video editing for the Tutorial that shows the overall step by step on how to use the application was edited using VideoPad Video Editor as shown in Figure 4.26.

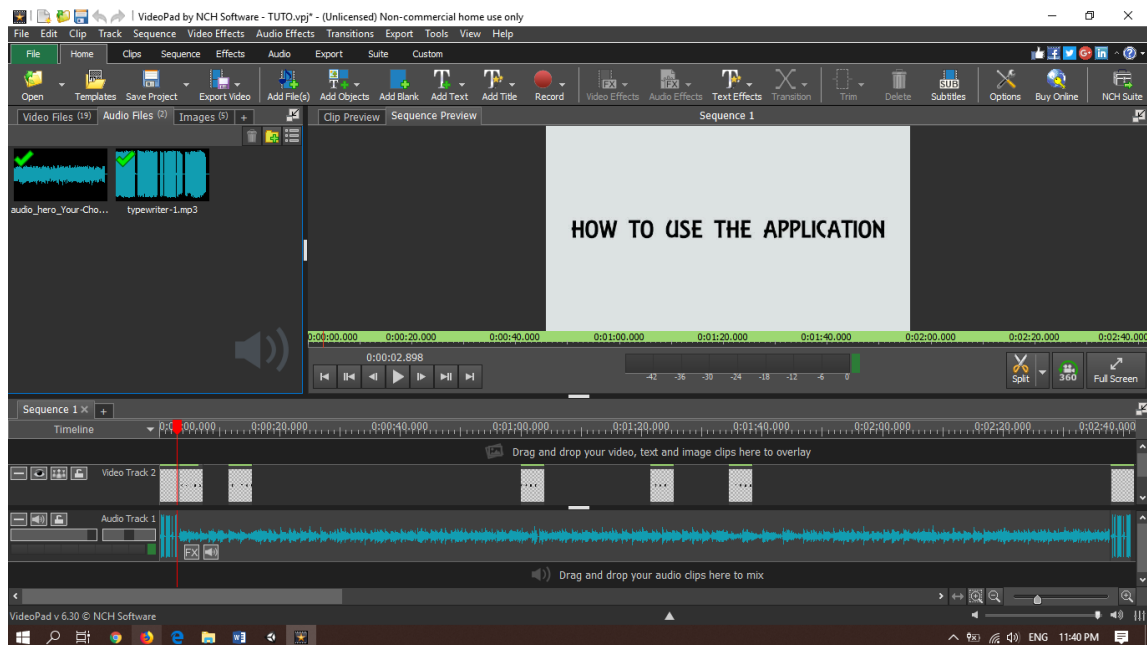


Figure 4.26 VideoPad Video Editor

The editing for all logos, texts, and images for this application was edited using Adobe Photoshop CS5 as shown in Figure 4.27, Figure 4.28, and Figure 4.29.

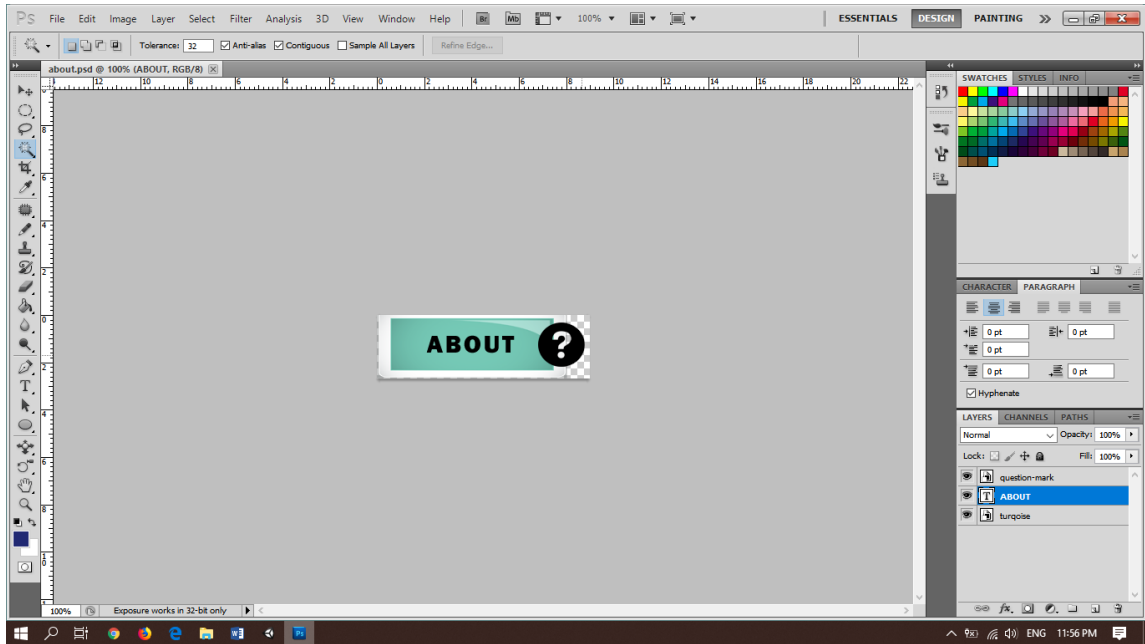


Figure 4.27 Button

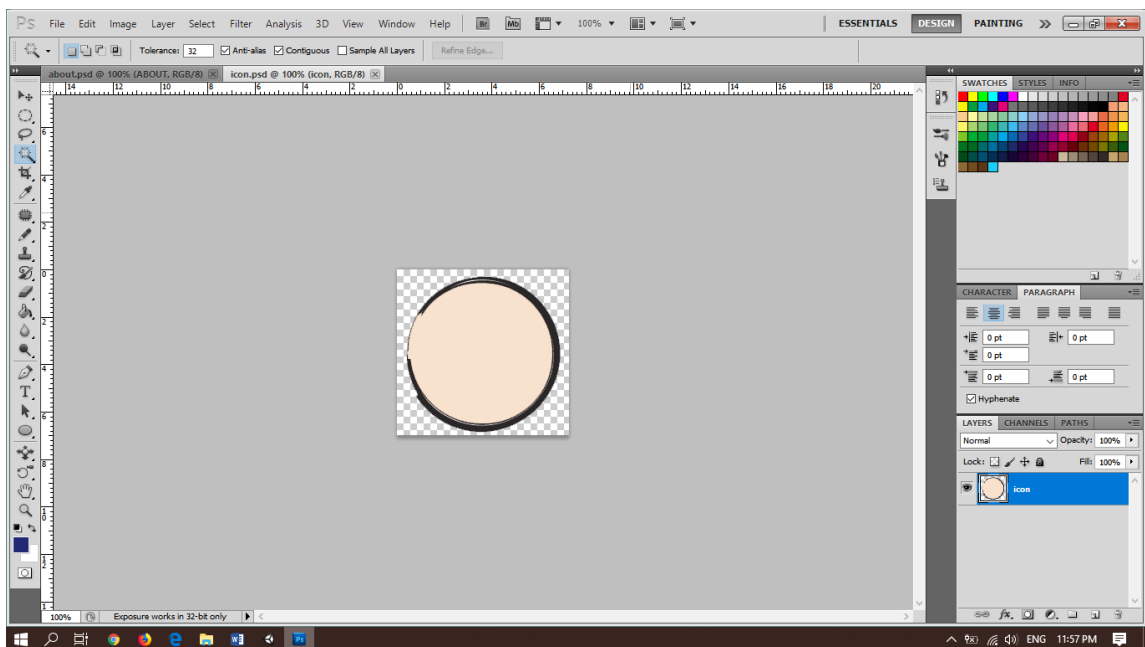


Figure 4.28 Icon

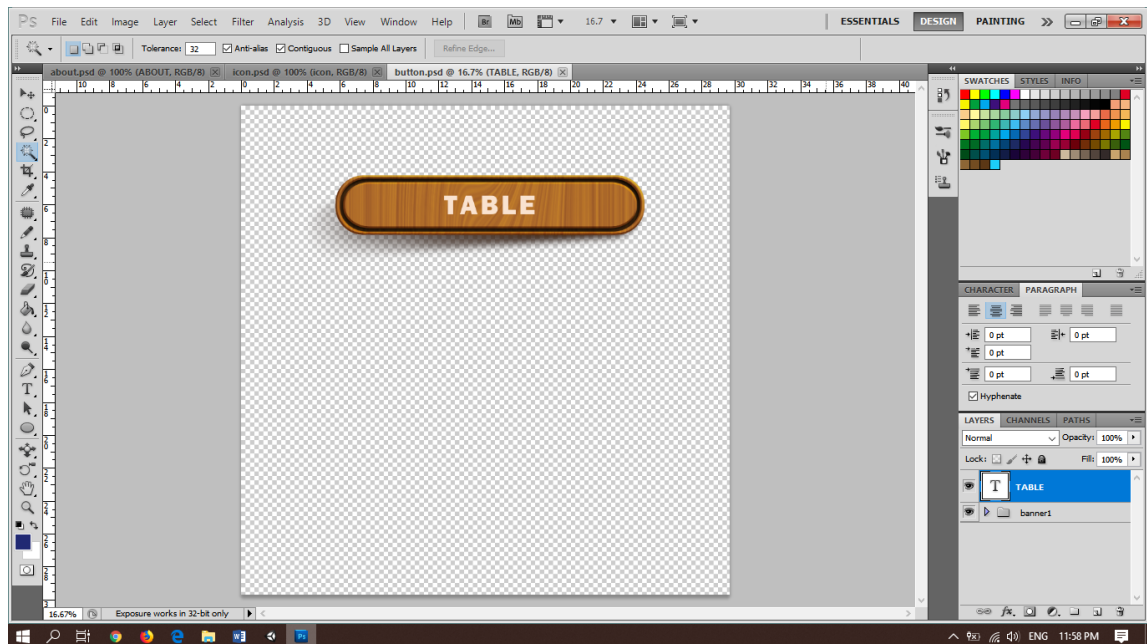


Figure 4.29 Image

The Wikitude SDK is the type of SDK that need to implement in the application in order to run the application inside the android smartphone. The license can be purchase under student license key version or trial key version. For this project was used student license key with the rules and regulations that need to follow.

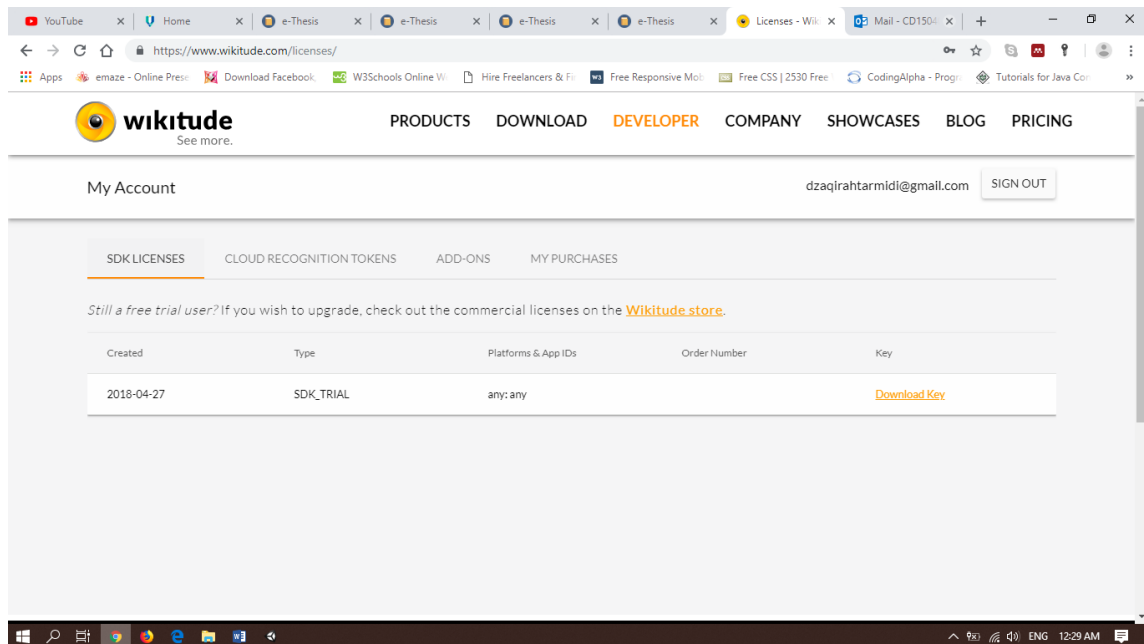


Figure 4.30 Wikitude SDK for student license key

4.2.3 Coding

The codes are used for various function in the application to make the application runs smoothly. For DinoController.cs is to make the mesh grid and the Initialize button disappear when the models active in the scene as shown in Table 4.1.

Table 4.1 DinoController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DinoController : MonoBehaviour {

    private InstantTrackingController trackerScript;
    private GameObject ButtonsParent;
```

```
// Use this for initialization
void Start () {

    trackerScript = GameObject.Find ("Controller").gameObject.GetComponent<InstantTrackingController> ();
    ButtonsParent = GameObject.Find ("Buttons Parent");

    //grid & button disable when 3D model comes out. inactive in start function
    trackerScript._gridRenderer.enabled = false;
    //ButtonsParent.SetActive (false);

}

void OnEnable(){
    //set function inactive if tracking is lost and start again

    trackerScript._gridRenderer.enabled = false;
    ButtonsParent.SetActive (false);

}

void OnDisable(){

    //to set the buttons parent active again when tracking stops

    ButtonsParent.SetActive (true);

}

}
```

For the LoadScene.cs is to load to another scene that means by go to another scene as well as shown in Table 4.2.

Table 4.2 LoadScene.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class LoadScene : MonoBehaviour {

    [SerializeField]
    string sceneName;

    public void Load()
    {
        SceneManager.LoadScene (sceneName);
    }
}
```

For the OptionPanel.cs as shown in Table 4.3 is to make panel for the list of 3D models of the furniture in order to open and close when user click it as shown in Figure 4.21. It also used for Setting function and About interface.

Table 4.3 OptionPanel.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
```

```
public class OptionPanel : MonoBehaviour {  
  
    public GameObject Panel;  
    int counter;  
  
    public void showhidePanel()  
    {  
  
        if (counter % 2 == 1) {  
            Panel.gameObject.SetActive (false); //hide panel  
        } else {  
            Panel.gameObject.SetActive (true); //show panel  
        }  
        counter++; //if odd number, the panel is hiding. if even, it showing  
    }  
}
```

For the ScreenshotPreview.cs it has 5 functions inside the code. Firstly, to get the images after user screenshot it and display back the images in the Gallery interface as shown in Figure 4.22. The screenshotted images can be found in the internal storage of the user's smartphone. The path of the screenshotted images is my files, internal storage, android, data, com.ump.psm, files. The format of the pictures is in .png. Second, when user click the delete icon in the Gallery interface, the current view of the image will be deleted. The function delete images was implemented in the delete icon. Third, the

function of next image also implemented in the application. User can go to the next image. Fourth, the function of the previous image. This function is to go to the previous image.

Table 4.4 ScreenshotPreview.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;

public class ScreenshotPreview : MonoBehaviour {

    [SerializeField]
    GameObject canvas;
    [SerializeField]
    Sprite defaultImage;
    string[] files = null;
    int whichScreenShotIsShown= 0;

    // Use this for initialization
    void Start () {
        canvas.GetComponent<Image> ().sprite = defaultImage;
        files = Directory.GetFiles(Application.persistentDataPath + "/", "*.png");
        if (files.Length > 0) {
            GetPictureAndShowIt ();
        }
    }

    void GetPictureAndShowIt()
    {

```

```

string pathToFile = files [whichScreenShotIsShown];
Texture2D texture = GetScreenshotImage (pathToFile);
Sprite sp = Sprite.Create (texture, new Rect (0, 0, texture.width, texture.height),
    new Vector2 (0.5f, 0.5f));
canvas.GetComponent<Image> ().sprite = sp;
}

Texture2D GetScreenshotImage(string filePath)
{
    Texture2D texture = null;
    byte[] fileBytes;
    if (File.Exists (filePath)) {
        fileBytes = File.ReadAllBytes (filePath);
        texture = new Texture2D (2, 2, TextureFormat.RGB24, false);
        texture.LoadImage (fileBytes);
    }
    return texture;
}

public void DeleteImage()
{
    if (files.Length > 0) {
        string pathToFile = files [whichScreenShotIsShown];
        if (File.Exists (pathToFile))
            File.Delete (pathToFile);
        files = Directory.GetFiles(Application.persistentDataPath + "/", "*.png");
        if (files.Length > 0)
            NextPicture ();
        else
            canvas.GetComponent<Image> ().sprite = defaultImage;
    }
}

```

```
    }  
}  
  
public void NextPicture()  
{  
    if (files.Length > 0) {  
        whichScreenShotIsShown += 1;  
        if (whichScreenShotIsShown > files.Length - 1)  
            whichScreenShotIsShown = 0;  
        GetPictureAndShowIt ();  
    }  
}  
  
public void PreviousPicture()  
{  
    if (files.Length > 0) {  
        whichScreenShotIsShown -= 1;  
        if (whichScreenShotIsShown < 0)  
            whichScreenShotIsShown = files.Length - 1;  
        GetPictureAndShowIt ();  
    }  
}  
}
```

For the TakeScreenShot.cs is to take the screenshot when user click the camera icon in the Home interface as shown in Figure 4.21. Only the current scene will be taken. All buttons in the Home interface would not include in the screenshotted images.

Table 4.5 TakeScreenShot.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TakeScreenShot : MonoBehaviour {

    [SerializeField]
    GameObject blink;

    public void TakeAShot()
    {
        StartCoroutine ("CaptureIt");
    }

    IEnumerator CaptureIt()
    {
        string timeStamp = System.DateTime.Now.ToString("dd-MM-yyyy-HH-mm-ss");
        string fileName = "Screenshot" + timeStamp + ".png";
        string pathToSave = fileName;
        GameObject.Find ("Canvas").GetComponent<Canvas> ().enabled = false;
        yield return new WaitForEndOfFrame();
        ScreenCapture.CaptureScreenshot(pathToSave);
        yield return new WaitForEndOfFrame();
    }
}
```

```

    GameObject.Find ("Canvas").GetComponent<Canvas> ().enabled = true;
    Instantiate (blink, new Vector2(0f, 0f), Quaternion.identity);

}
}

```

For InstantTrackingController.cs is the main component in the application to instant track the scene as shown in Table 4.6. It coordinates the activity between the Instant Tracker, the UI, the augmentations and the touch input.

Table 4.6 InstantTrackingController.cs

```

using UnityEngine;
using UnityEngine.UI;
using System.Collections;
using System.Collections.Generic;
using Wikitude;
using System;

public class InstantTrackingController : SampleController
{
    public GameObject ButtonDock;
    public GameObject InitializationControls;
    public Text HeightLabel;
    public Text ScaleLabel;

    public InstantTracker Tracker;

    public List<Button> Buttons;
    public List<GameObject> Models;

```

```
public Text MessageBox;

public Image ActivityIndicator;

public Color EnabledColor = new Color(0.2f, 0.75f, 0.2f, 0.8f);
public Color DisabledColor = new Color(1.0f, 0.2f, 0.2f, 0.8f);

private float _currentDeviceHeightAboveGround = 1.0f;

private MoveController _moveController;
public GridRenderer _gridRenderer;

private HashSet<GameObject> _activeModels = new HashSet<GameObject>();
private InstantTrackingState _currentState = InstantTrackingState.Initializing;
public InstantTrackingState CurrentState {
    get { return _currentState; }
}
private bool _isTracking = false;

public HashSet<GameObject> ActiveModels {
    get {
        return _activeModels;
    }
}

private void Awake() {
    Application.targetFrameRate = 60;

    _moveController = GetComponent<MoveController>();
}
```

```

    _gridRenderer = GetComponent<GridRenderer>();
}

protected override void Start() {
    base.Start();
    QualitySettings.shadowDistance = 4.0f;

    MessageBox.text = "Starting the SDK";
    // The Wikitude SDK needs to be fully started before we can query for platform as
    // sisted tracking support
    // SDK initialization happens during start, so we wait one frame in a coroutine
    StartCoroutine(CheckPlatformAssistedTrackingSupport());
}

private IEnumerator CheckPlatformAssistedTrackingSupport() {
    yield return null;
    if (Tracker.SMARTEnabled) {
        Tracker.IsPlatformAssistedTrackingSupported((SmartAvailability smartAvaila
bility) => {
            UpdateTrackingMessage(smartAvailability);
        });
    }
}

private void UpdateTrackingMessage(SmartAvailability smartAvailability) {
    if (Tracker.SMARTEnabled) {
        string sdk;
        if (Application.platform == RuntimePlatform.Android) {
            sdk = "ARCore";
        }
    }
}

```

```

} else if (Application.platform == RuntimePlatform.IPhonePlayer) {
    sdk = "ARKit";
} else {
    return;
}

switch (smartAvailability) {
    case SmartAvailability.IndeterminateQueryFailed: {
        MessageBox.text = "Platform support query failed. Running without platform assisted tracking support.";
        break;
    }
    case SmartAvailability.CheckingQueryOngoing: {
        MessageBox.text = "Platform support query ongoing.";
        break;
    }
    case SmartAvailability.Unsupported: {
        MessageBox.text = "Running without platform assisted tracking support.";
;
        break;
    }
    case SmartAvailability.SupportedUpdateRequired:
    case SmartAvailability.Supported: {
        string runningWithMessage = "Running with platform assisted tracking support (" + sdk + ")";

        if (_currentState == InstantTrackingState.Tracking) {
            MessageBox.text = runningWithMessage;
        } else {
            MessageBox.text = runningWithMessage + "\n Move your phone around";
        }
    }
}

```



```

nd until the target turns green, which is when you can start tracking.";
    }
    break;
}
}
} else {
    MessageBox.text = "Running without platform assisted tracking support.";
}
}

protected override void Update() {
    base.Update();
    if (_currentState == InstantTrackingState.Initializing) {
        if (Tracker.CanStartTracking()) {
            _gridRenderer.TargetColor = Color.green;
        } else {
            _gridRenderer.TargetColor = GridRenderer.DefaultTargetColor;
        }
    } else {
        _gridRenderer.TargetColor = GridRenderer.DefaultTargetColor;
    }
}

#region UI Events
public void OnInitializeButtonClicked() {
    Tracker.SetState(InstantTrackingState.Tracking);
}

public void OnHeightValueChanged(float newHeightValue) {
    _currentDeviceHeightAboveGround = newHeightValue;
}

```

```

        HeightLabel.text = string.Format("{0:0.##} m", _currentDeviceHeightAboveGround);
    Tracker.DeviceHeightAboveGround = _currentDeviceHeightAboveGround;
}

public void OnBeginDrag (int modelIndex) {
    if (_isTracking) {
        // Create object
        GameObject modelPrefab = Models[modelIndex];
        Transform model = Instantiate(modelPrefab).transform;
        _activeModels.Add(model.gameObject);
        // Set model position at touch position
        var cameraRay = Camera.main.ScreenPointToRay(Input.mousePosition);
        Plane p = new Plane(Vector3.up, Vector3.zero);
        float enter;
        if (p.Raycast(cameraRay, out enter)) {
            model.position = cameraRay.GetPoint(enter);
        }

        // Set model orientation to face toward the camera
        Quaternion modelRotation = Quaternion.LookRotation(Vector3.ProjectOnPlane(-Camera.main.transform.forward, Vector3.up), Vector3.up);
        model.rotation = modelRotation;

        _moveController.SetMoveObject(model);
    }
}

public override void OnBackButtonClicked() {
    if (_currentState == InstantTrackingState.Initializing) {

```

```
        base.OnBackButtonClicked();
    } else {
        Tracker.SetState(InstantTrackingState.Initializing);
    }
}

#endregion

#region Tracker Events
public void OnSceneRecognized(InstantTarget target) {
    SetSceneActive(true);
}

public void OnSceneLost(InstantTarget target) {
    SetSceneActive(false);
}

private void SetSceneActive(bool active) {

    _gridRenderer.enabled = active;
    foreach (var button in Buttons) {
        button.interactable = active;
    }

    foreach (var model in _activeModels) {
        model.SetActive(active);
    }

    ActivityIndicator.color = active ? EnabledColor : DisabledColor;
}
```

```

    _isTracking = active;
}

public void OnStateChanged(InstantTrackingState newState) {
    _currentState = newState;
    if (newState == InstantTrackingState.Tracking) {
        if (InitializationControls != null) {
            InitializationControls.SetActive(false);
        }
        ButtonDock.SetActive(true);
    } else {
        foreach (var model in _activeModels) {
            Destroy(model);
        }
        _activeModels.Clear();

        if (InitializationControls != null) {
            InitializationControls.SetActive(true);
        }
        ButtonDock.SetActive(false);
    }
}

/// <summary>
/// Used when augmentations are loaded from disk.
/// Please see SaveInstantTarget and LoadInstantTarget for more information.
/// </summary>
internal void LoadAugmentation(AugmentationDescription augmentation) {
    GameObject modelPrefab = Models[augmentation.ID];
    Transform model = Instantiate(modelPrefab).transform;
}

```

```

_activeModels.Add(model.gameObject);

model.localPosition = augmentation.LocalPosition;
model.localRotation = augmentation.LocalRotation;
model.localScale = augmentation.LocalScale;

model.gameObject.SetActive(false);
}

public void OnError(Error error) {
    PrintError("Instant Tracker error!", error);
}
#endregion
}

```

MoveController.cs is for object position and touch position as shown in Table 4.7.

Table 4.7 MoveController.cs

```

using UnityEngine;

public class MoveController : MonoBehaviour
{
    private Transform _activeObject = null;

    private Vector3 _startObjectPosition;
    private Vector2 _startTouchPosition;
    private Vector2 _touchOffset;
}

```

```
private InstantTrackingController _controller;

private void Start() {
    _controller = GetComponent<InstantTrackingController>();
}

public Transform ActiveObject {
    get {
        return _activeObject;
    }
}

public void SetMoveObject(Transform newMoveObject) {
    if (_controller.ActiveModels.Contains(newMoveObject.gameObject)) {
        _activeObject = newMoveObject;
        _startObjectPosition = _activeObject.position;
        _startTouchPosition = Input.GetTouch(0).position;
        _touchOffset = Camera.main.WorldToScreenPoint(_startObjectPosition);
    }
}

void Update () {
    if (Input.touchCount > 0) {
        Touch touch = Input.GetTouch(0);
        RaycastHit hit;

        if (_activeObject == null) {
            if (Physics.Raycast(Camera.main.ScreenPointToRay(touch.position), out hit
        )) {
                SetMoveObject(hit.transform);
            }
        }
    }
}
```

```
    }  
  }  
  
  if (_activeObject != null) {  
    var screenPosForRay = (touch.position - _startTouchPosition) + _touchOffset;  
et;  
  
    Ray cameraRay = Camera.main.ScreenPointToRay(screenPosForRay);  
    Plane p = new Plane(Vector3.up, Vector3.zero);  
  
    float enter;  
    if (p.Raycast(cameraRay, out enter)) {  
      var position = cameraRay.GetPoint(enter);  
  
      position.x = Mathf.Clamp(position.x, -15.0f, 15.0f);  
      position.y = 0.0f;  
      position.z = Mathf.Clamp(position.z, -15.0f, 15.0f);  
  
      _activeObject.position = Vector3.Lerp(_activeObject.position, position, 0  
.25f);  
    }  
  }  
} else {  
  _activeObject = null;  
}  
}  
}
```

Wikitude SDK uses a secondary camera to render the camera frame, need to make sure to render the grid only on the main camera. GridRenderer.cs is to make the grid in order to detect and initialize the scene. It renders a grid with 25 cm spacing that can be helpful during initialization and tracking as shown in Table 4.8.

Table 4.8 GridRenderer.cs

```
using UnityEngine;

public class GridRenderer : MonoBehaviour
{
    public static Color DefaultTargetColor = new Color(1.0f, 0.525f, 0, 1.0f);
    private static Color GridColor = new Color(1.0f, 1.0f, 1.0f, 1.0f);
    private static Color MainLineColor = GridColor * 0.9f;
    private static Color UnitLineColor = GridColor * 0.75f;

    public Color TargetColor = DefaultTargetColor;

    private const int NumberOfLinesOnEitherSide = 50;
    private const float LineSpacing = 0.25f;
    private const int IntervalBetweenMainLines = 10;
    private const float TargetSize = 8.0f;

    private Material _lineMaterial;
    private Plane[] _cameraPlanes;

    private void Start() {
        InitializeLineMaterial();
    }

    private void OnRenderObject() {
```


// Because the Wikitude SDK uses a secondary camera to render the camera frame, we need to make sure to render the grid only on the main camera.

```

if (Camera.current == Camera.main) {
    RenderGrid();
}
}

```

```

private void InitializeLineMaterial() {

```

```

    var shader = Shader.Find("Hidden/Internal-Colored");

```

```

    _lineMaterial = new Material(shader);

```

```

    _lineMaterial.hideFlags = HideFlags.HideAndDontSave;

```

```

    _lineMaterial.SetInt("_SrcBlend", (int)UnityEngine.Rendering.BlendMode.SrcAlpha);

```

```

    _lineMaterial.SetInt("_DstBlend", (int)UnityEngine.Rendering.BlendMode.OneMinusSrcAlpha);

```

```

    _lineMaterial.SetInt("_Cull", (int)UnityEngine.Rendering.CullMode.Off);

```

```

    _lineMaterial.SetInt("_ZWrite", 0);

```

```

}

```

```

private void RenderGrid() {

```

```

    _lineMaterial.SetPass(0);

```

```

    _cameraPlanes = GeometryUtility.CalculateFrustumPlanes(Camera.main);

```

```

    GL.Begin(GL.LINES);

```

```

    RenderGrid(Vector3.forward, Vector3.right);

```

```

    RenderGrid(Vector3.right, Vector3.forward);

```

```

RenderLine(Vector3.forward, Vector3.zero, TargetColor, TargetSize);
RenderLine(Vector3.right, Vector3.zero, TargetColor, TargetSize);

GL.End();
}

private void RenderGrid(Vector3 mainAxis, Vector3 perpendicular) {
    for (int i = -
NumberOfLinesOnEitherSide; i <= NumberOfLinesOnEitherSide; ++i) {
        var color = UnitLineColor;
        if (i == 0) {
            color = GridColor;
        } else if (i % IntervalBetweenMainLines == 0) {
            color = MainLineColor;
        }
        float intensity = 1.0f - Mathf.Abs(i * 2.0f) / NumberOfLinesOnEitherSide;
        color.a *= intensity;
        if (color.a > 0.01f) {
            RenderLine(mainAxis, perpendicular * i * LineSpacing, color, (float)Numb
erOfLinesOnEitherSide / 8.0f);
        }
    }
}

private void RenderLine(Vector3 axis, Vector3 offset, Color color, float length) {
    RenderLine(GetTransparent(color), axis * (-length) + offset, color, offset);
    RenderLine(color, offset, GetTransparent(color), axis * length + offset);
}

private void RenderLine(Color startColor, Vector3 startPosition, Color endColor, V

```

```

ector3 endPosition) {
    Color clippedEndColor = Color.Lerp(startColor, endColor, Clip(startPosition, ref
endPosition));
    Color clippedStartColor = Color.Lerp(startColor, endColor, 1.0f - Clip(endPositi
on, ref startPosition));

    GL.Color(clippedStartColor);
    GL.Vertex(startPosition);
    GL.Color(clippedEndColor);
    GL.Vertex(endPosition);
}

private float Clip(Vector3 start, ref Vector3 end) {
    // We clip the vertices to the camera planes because on some Android devices cli
pping of GL_LINES doesn't work correctly.
    Vector3 v = end - start;
    float magnitude = v.magnitude;
    Vector3 vNorm = v / magnitude;

    var startRay = new Ray(start, vNorm);
    float minDistance = magnitude;

    foreach (var plane in _cameraPlanes) {
        if (Vector3.Dot(vNorm, plane.normal) < 0) {
            float enter;
            if (plane.Raycast(startRay, out enter)) {
                if (enter < minDistance) {
                    minDistance = enter;
                }
            }
        }
    }
}

```

```

    }
}

end = startRay.GetPoint(minDistance);
return minDistance / magnitude;
}

private Color GetTransparent(Color color) {
    color.a = 0.0f;
    return color;
}
}

```

LeanRotate.cs is to rotate all the 3D models that has been assigned in the controller as shown in Table 4.9. This script allows user to transform the current GameObject.

Table 4.9 LeanRotate.cs

```

using UnityEngine;

namespace Lean.Touch
{
    // This script allows you to transform the current GameObject
    public class LeanRotate : MonoBehaviour
    {
        [Tooltip("Ignore fingers with StartedOverGui?")]
        public bool IgnoreGuiFingers;

        [Tooltip("Allows you to force rotation with a specific amount of fingers (0 = any

```

```
)"]  
  
    public int RequiredFingerCount;  
  
    [Tooltip("Does rotation require an object to be selected?")]  
    public LeanSelectable RequiredSelectable;  
  
    [Tooltip("The camera we will be moving")]  
    public Camera Camera;  
  
    [Tooltip("The rotation axis used for non-relative rotations")]  
    public Vector3 RotateAxis = Vector3.forward;  
  
    [Tooltip("Should the rotation be performed relative to the finger center?")]  
    public bool Relative;  
  
#if UNITY_EDITOR  
    protected virtual void Reset()  
    {  
        Start();  
    }  
#endif  
  
    protected virtual void Start()  
    {  
        if (RequiredSelectable == null)  
        {  
            RequiredSelectable = GetComponent<LeanSelectable>();  
        }  
    }  
}
```

```

protected virtual void Update()
{
    // If we require a selectable and it isn't selected, cancel rotation
    if (RequiredSelectable != null && RequiredSelectable.IsSelected == false)
    {
        return;
    }

    // Get the fingers we want to use
    var fingers = LeanTouch.GetFingers(IgnoreGuiFingers, RequiredFingerCount)
;

    // Calculate the rotation values based on these fingers
    var center = LeanGesture.GetScreenCenter(fingers);
    var degrees = LeanGesture.GetTwistDegrees(fingers);

    // Perform the rotation
    Rotate(center, degrees);
}

private void Rotate(Vector3 center, float degrees)
{
    if (Relative == true)
    {
        // If camera is null, try and get the main camera, return true if a camera was
found
        if (LeanTouch.GetCamera(ref Camera) == true)
        {
            // World position of the reference point
            var worldReferencePoint = Camera.ScreenToWorldPoint(center);

```

```

        // Rotate the transform around the world reference point
        transform.RotateAround(worldReferencePoint, Camera.transform.forward, degrees);
    }
}
else
{
    transform.rotation *= Quaternion.AngleAxis(degrees, RotateAxis);
}
}
}
}

```

LeanTapSelect.cs is to detect how many users tap the screen. This script as shown in Table 4.10 allows user to select LeanSelectable.cs as shown in Table 4.11 components using finger taps.

Table 4.10 LeanTapSelect.cs

```

using UnityEngine;

namespace Lean.Touch
{
    // This script allows you to select LeanSelectable components using finger taps
    public class LeanTapSelect : LeanSelect
    {
        [Tooltip("Ignore fingers with StartedOverGui?")]
        public bool IgnoreGuiFingers = true;
    }
}

```

```
[Tooltip("How many times must this finger tap before OnFingerTap gets called?
(0 = every time)")]
```

```
public int RequiredTapCount = 0;
```

```
[Tooltip("How many times repeating must this finger tap before OnFingerTap gets
called? (e.g. 2 = 2, 4, 6, 8, etc) (0 = every time)")]
```

```
public int RequiredTapInterval;
```

```
protected virtual void OnEnable()
```

```
{
```

```
    // Hook events
```

```
    LeanTouch.OnFingerTap += FingerTap;
```

```
}
```

```
protected virtual void OnDisable()
```

```
{
```

```
    // Unhook events
```

```
    LeanTouch.OnFingerTap -= FingerTap;
```

```
}
```

```
private void FingerTap(LeanFinger finger)
```

```
{
```

```
    // Ignore this finger?
```

```
    if (IgnoreGuiFingers == true && finger.StartedOverGui == true)
```

```
    {
```

```
        return;
```

```
    }
```

```
    if (RequiredTapCount > 0 && finger.TapCount != RequiredTapCount)
```



```

    {
        return;
    }

    if (RequiredTapInterval > 0 && (finger.TapCount % RequiredTapInterval) !=
0)
    {
        return;
    }

    // Try and select
    Select(finger);
}
}
}

```

This LeanSelectable.cs component allows user to select 3D models via another component. It stores the finger that began selection of this LeanSelectable. It also will become null as soon as that finger releases, which user can detect via OnSelectUp as shown in Table 4.11.

Table 4.11 LeanSelectable.cs

```

using UnityEngine;
using UnityEngine.Events;

namespace Lean.Touch
{
    // This component allows you to select this GameObject via another component

```

```

public class LeanSelectable : MonoBehaviour
{
    // Event signature
    [System.Serializable] public class LeanFingerEvent : UnityEvent<LeanFinger> {
    }

    [Tooltip("Should IsSelected temporarily return false if the selecting finger is still
being held?")]
    public bool HideWithFinger;

    public bool IsSelected
    {
        get
        {
            // Hide IsSelected?
            if (HideWithFinger == true && isSelected == true && SelectingFinger != null)
            {
                return false;
            }

            return isSelected;
        }
    }

    // This stores the finger that began selection of this LeanSelectable
    // This will become null as soon as that finger releases, which you can detect via
    OnSelectUp
    [System.NonSerialized]
    public LeanFinger SelectingFinger;

```

```
// Called when selection begins (finger = the finger that selected this)
public LeanFingerEvent OnSelect;

// Called when the selecting finger goes up (finger = the finger that selected this)
public LeanFingerEvent OnSelectUp;

// Called when this is deselected, if OnSelectUp hasn't been called yet, it will get
called first
public UnityEvent OnDeselect;

// Is this selectable selected?
[SerializeField]
private bool isSelected;

[ContextMenu("Select")]
public void Select()
{
    Select(null);
}

public void Select(LeanFinger finger)
{
    isSelected = true;
    SelectingFinger = finger;

    OnSelect.Invoke(finger);
}

[ContextMenu("Deselect")]
```

```
public void Deselect()
{
    if (SelectingFinger != null)
    {
        OnSelectUp.Invoke(SelectingFinger);

        SelectingFinger = null;
    }

    isSelected = false;

    OnDeselect.Invoke();
}

protected virtual void OnEnable()
{
    // Hook events
    LeanTouch.OnFingerUp += OnFingerUp;
}

protected virtual void OnDisable()
{
    // Unhook events
    LeanTouch.OnFingerUp -= OnFingerUp;

    if (isSelected == true)
    {
        Deselect();
    }
}
```

```

private void OnFingerUp(LeanFinger finger)
{
    // If the finger went up, it's no longer selecting anything
    if (finger == SelectingFinger)
    {
        OnSelectUp.Invoke(SelectingFinger);

        SelectingFinger = null;
    }
}
}
}

```

LeanTranslate.cs that allows user to transform the current 3D models as shown in Table 4.12.

Table 4.12 LeanTranslate.cs

```

using UnityEngine;

namespace Lean.Touch
{
    // This script allows you to transform the current GameObject
    public class LeanTranslate : MonoBehaviour
    {
        [Tooltip("Ignore fingers with StartedOverGui?")]
        public bool IgnoreGuiFingers = true;
    }
}

```

```
[Tooltip("Ignore fingers if the finger count doesn't match? (0 = any)")]
public int RequiredFingerCount;

[Tooltip("Does translation require an object to be selected?")]
public LeanSelectable RequiredSelectable;

[Tooltip("The camera the translation will be calculated using (default = MainCamera)")]
public Camera Camera;

#if UNITY_EDITOR
    protected virtual void Reset()
    {
        Start();
    }
#endif

protected virtual void Start()
{
    if (RequiredSelectable == null)
    {
        RequiredSelectable = GetComponent<LeanSelectable>();
    }
}

protected virtual void Update()
{
    // If we require a selectable and it isn't selected, cancel translation
    if (RequiredSelectable != null && RequiredSelectable.IsSelected == false)
    {
```

```

        return;
    }

    // Get the fingers we want to use
    var fingers = LeanTouch.GetFingers(IgnoreGuiFingers, RequiredFingerCount,
RequiredSelectable);

    // Calculate the screenDelta value based on these fingers
    var screenDelta = LeanGesture.GetScreenDelta(fingers);

    // Perform the translation
    Translate(screenDelta);
}

private void Translate(Vector2 screenDelta)
{
    // If camera is null, try and get the main camera, return true if a camera was fo
und
    if (LeanTouch.GetCamera(ref Camera) == true)
    {
        // Screen position of the transform
        var screenPosition = Camera.WorldToScreenPoint(transform.position);

        // Add the deltaPosition
        screenPosition += (Vector3)screenDelta;

        // Convert back to world space
        transform.position = Camera.ScreenToWorldPoint(screenPosition);
    }
}

```

```
}  
}
```

StreamVideo.cs is to play and pause the tutorial video in the tutorial menu as shown in Table 4.13.

Table 4.13 StreamVideo.cs

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UI;  
using UnityEngine.Video;  
  
public class StreamVideo : MonoBehaviour {  
  
    public RawImage image;  
    public GameObject playIcon;  
  
    public VideoClip videoToPlay;  
  
    private VideoPlayer videoPlayer;  
    private VideoSource videoSource;  
  
    private AudioSource audioSource;  
  
    private bool isPaused = false;  
    private bool firstRun = true;
```



```
IEnumerator playVideo() {
    playIcon.SetActive(false);
    firstRun = false;
    //Add VideoPlayer to the GameObject
    videoPlayer = gameObject.AddComponent<VideoPlayer>();

    //Add AudioSource
    audioSource = gameObject.AddComponent<AudioSource>();

    //Disable Play on Awake for both Video and Audio
    videoPlayer.playOnAwake = false;
    audioSource.playOnAwake = false;
    audioSource.Pause();

    //We want to play from video clip not from url

    videoPlayer.source = VideoSource.VideoClip;

    //Set Audio Output to AudioSource
    videoPlayer.audioOutputMode = VideoAudioOutputMode.AudioSource;

    //Assign the Audio from Video to AudioSource to be played
    videoPlayer.EnableAudioTrack(0, true);
    videoPlayer.SetTargetAudioSource(0, audioSource);

    //Set video To Play then prepare Audio to prevent Buffering
    videoPlayer.clip = videoToPlay;
    videoPlayer.Prepare();
}
```

```
//Wait until video is prepared
while (!videoPlayer.isPrepared) {
    yield return null;
}

Debug.Log("Done Preparing Video");

//Assign the Texture from Video to RawImage to be displayed
image.texture = videoPlayer.texture;

//Play Video
videoPlayer.Play();

//Play Sound
audioSource.Play();

Debug.Log("Playing Video");
while (videoPlayer.isPlaying) {
    Debug.LogWarning("Video Time: " + Mathf.FloorToInt((float)videoPlayer.time));
    yield return null;
}

Debug.Log("Done Playing Video");
}

public void PlayPause() {
    if(!firstRun && !isPaused) {
        videoPlayer.Pause();
    }
}
```

```
        audioSource.Pause();
        playIcon.SetActive(true);
        isPaused = true;
    } else if (!firstRun && isPaused) {
        videoPlayer.Play();
        audioSource.Play();
        playIcon.SetActive(false);
        isPaused = false;
    } else {
        StartCoroutine(playVideo());
    }
}
}
```

4.3 Testing and Result Discussion

Testing and result discussion will fully be explained in this stage.

4.3.1 Result Discussion

After the project have been successfully developed, the system is then tested to ensure that every item and description is working well. Below are the screenshots of result of the actual project that is being tested.

i. Tracking interface

This is tracking interface that will track the room scene.

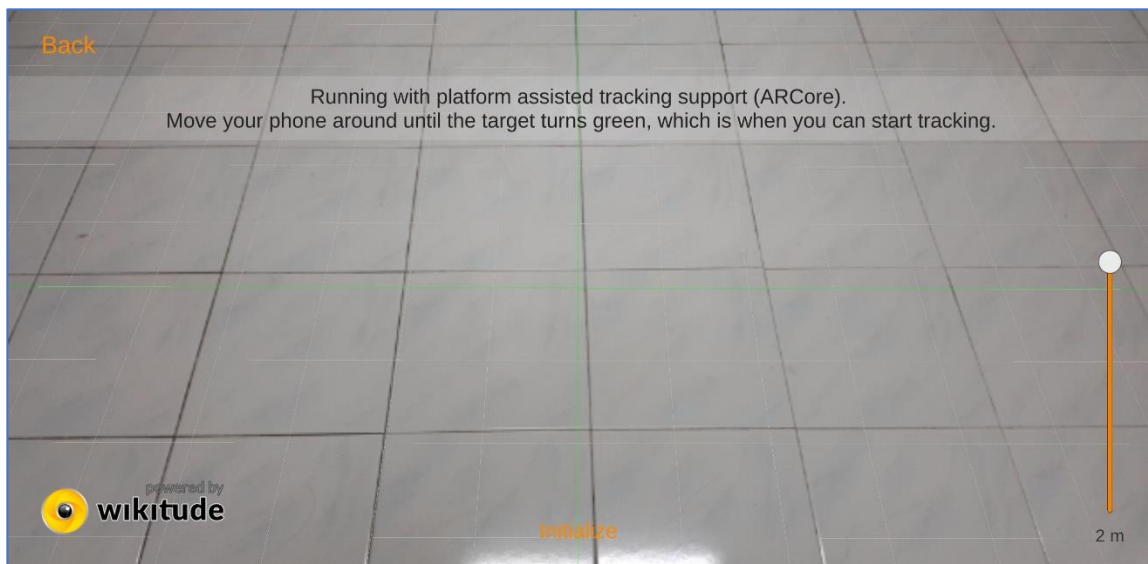


Figure 4.31 Tracking interface in application.

ii. Home interface in application

This is home interface that contains main function. All the functions can be done here.



Figure 4.32 Home interface in application

- iii. Select and Load the 3D models of furniture.

User can choose the 72 of 3D models of furniture as their needs.



Figure 4.33 Select and Load the 3D models of furniture.

iv. Gallery interface in application.

User can preview their images that have been taken before. There was delete button in case user wants to delete current view image.



Figure 4.34 Gallery interface in application.

- v. Setting interface in the application.

User can choose either wants to go to about menu or tutorial menu.



Figure 4.35 Setting interface in the application.

- vi. About menu in the application.

This interface is about the overall discription of the application and its functions for each button.

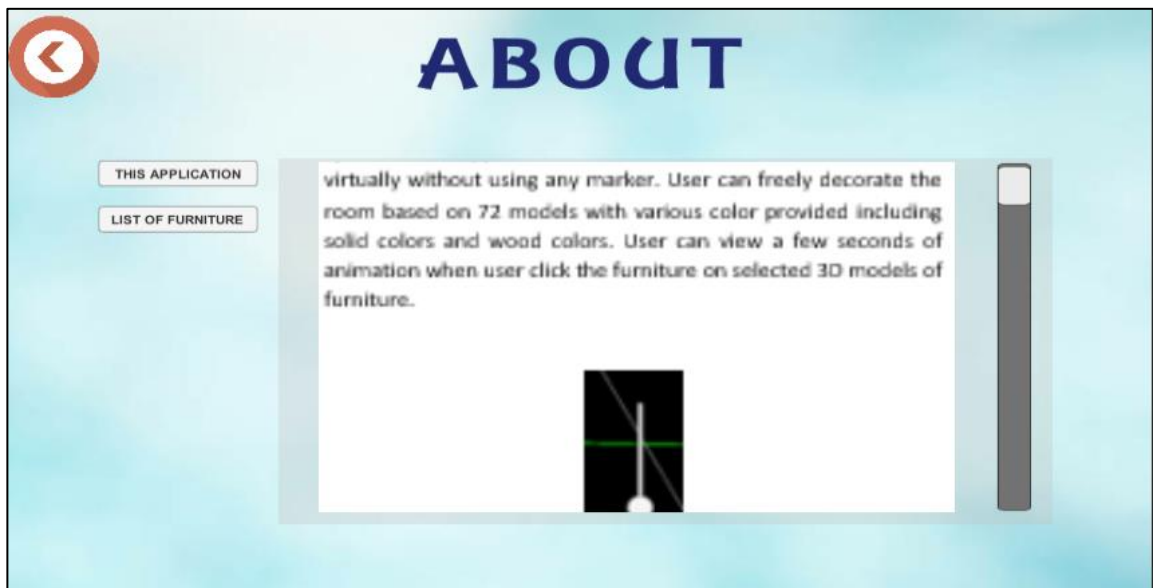


Figure 4.36 Overall description.



Figure 4.37 List of the furniture in the application

vii. Tutorial interface in the application.

This interface will play the video on how to use the application.

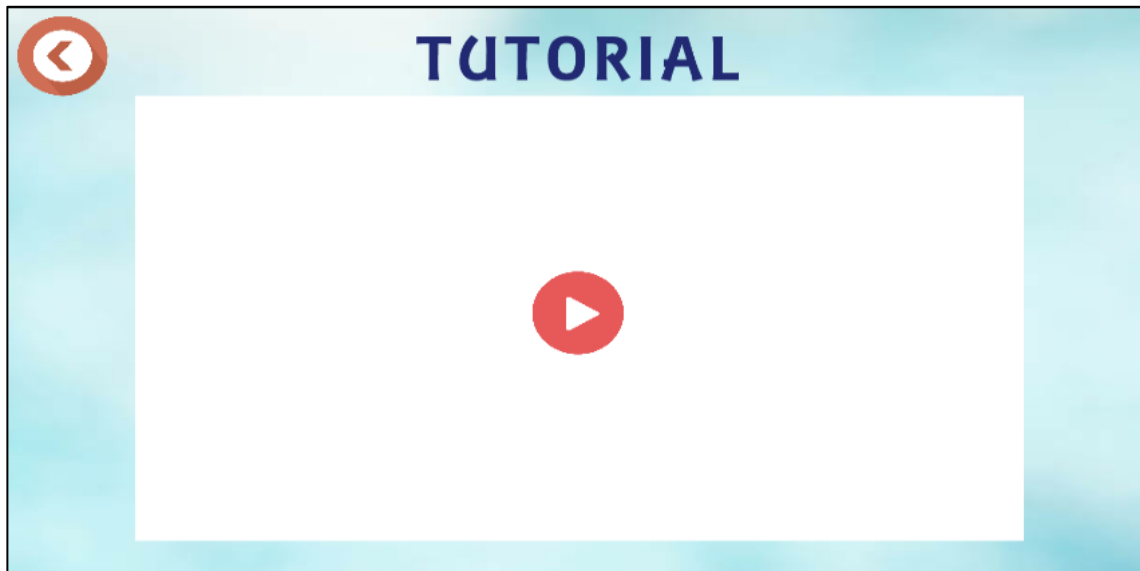


Figure 4.38 Tutorial menu in the application

4.3.2 Testing Report

In this stage, the testing strategy will be conducting using user acceptance test (UAT) in order to determine project is fully tested. The component of UAT and the System Testing Approval will be discussing in the Appendix A.

4.4 User Manual

The application was developed using Unity 3D and Autodesk Maya 2016. Refer Appendix B.

4.5 Usability Test

For the usability test, it used the questionnaire in order to collect data from the tested of this application.

4.5.1 Test Features

There are 26 respondents involved in this usability test. The users were tested to use this application and they were interviewed with the set of questionnaires that can be referred in Appendix C.

Below is the list of result of the questionnaire.

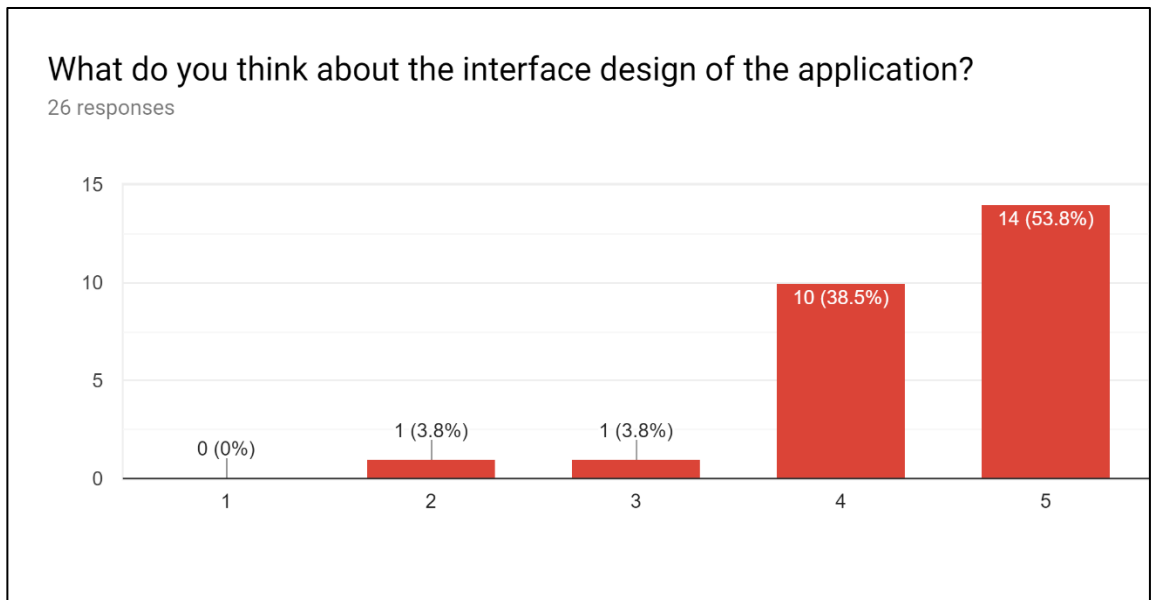


Figure 4.39 Question 1

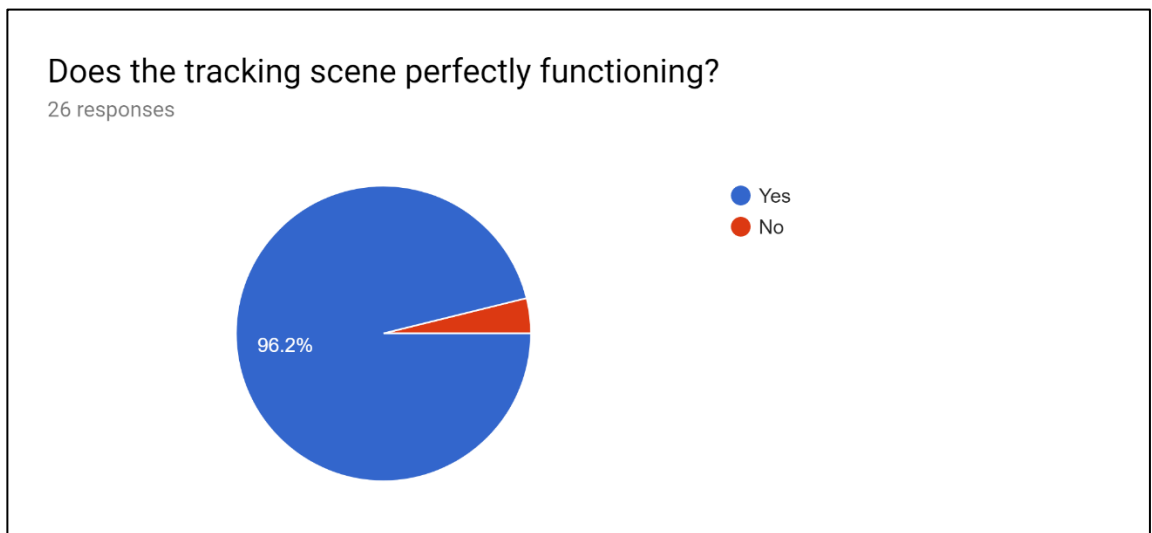


Figure 4.40 Question 2

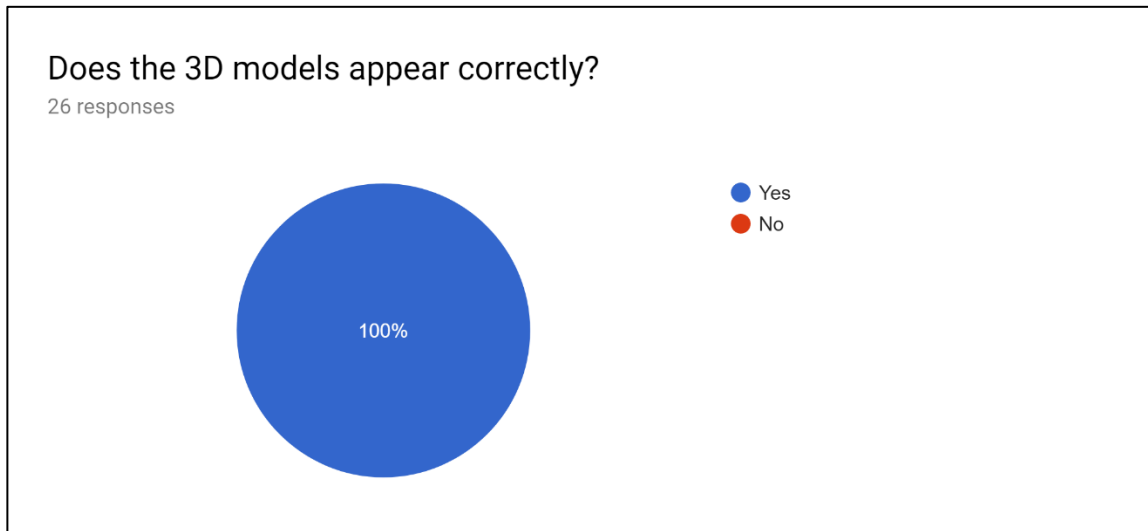


Figure 4.41 Question 3

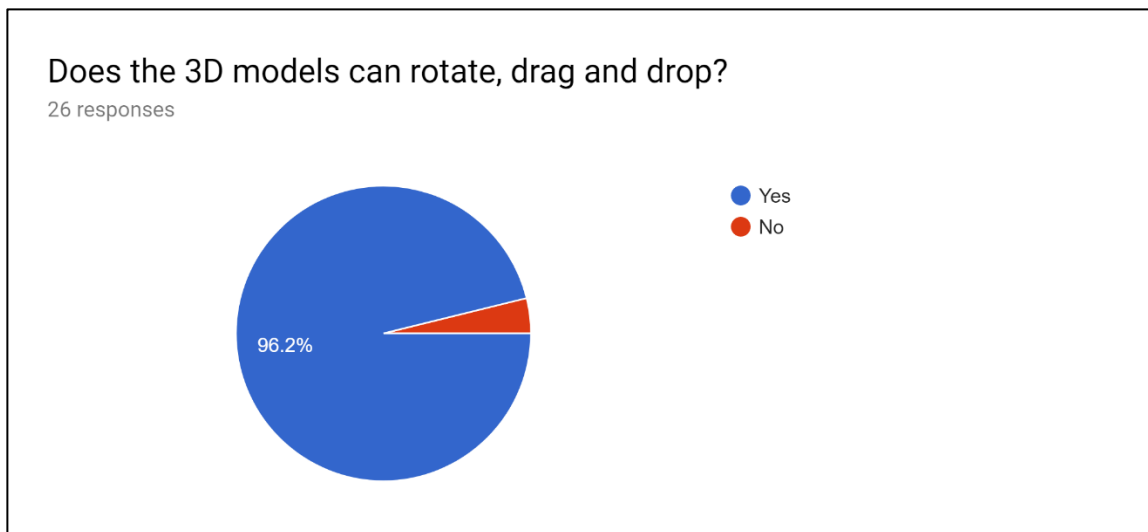


Figure 4.42 Question 4

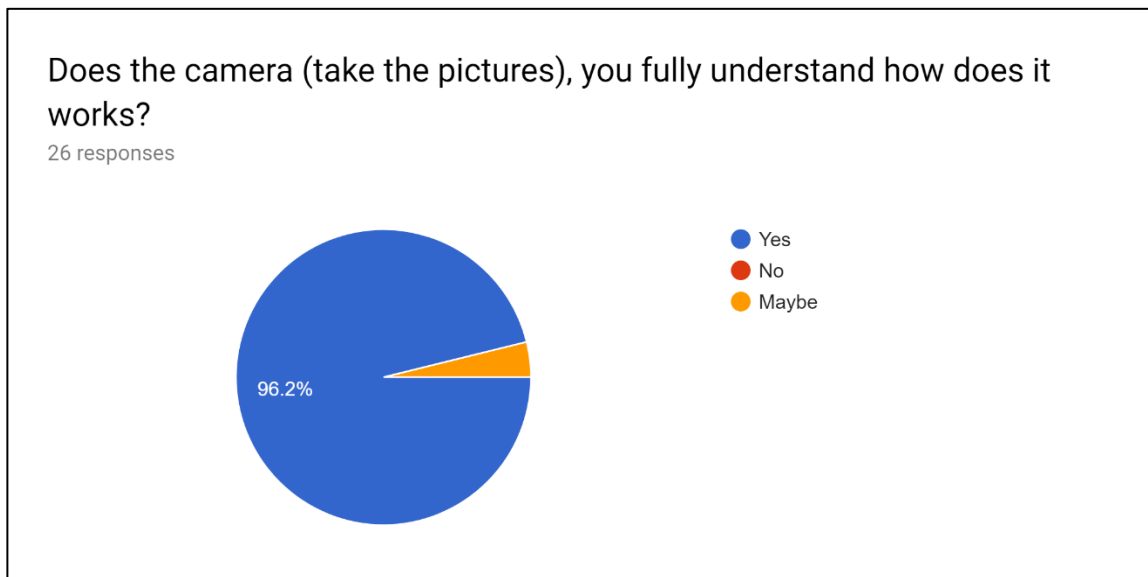


Figure 4.43 Question 5

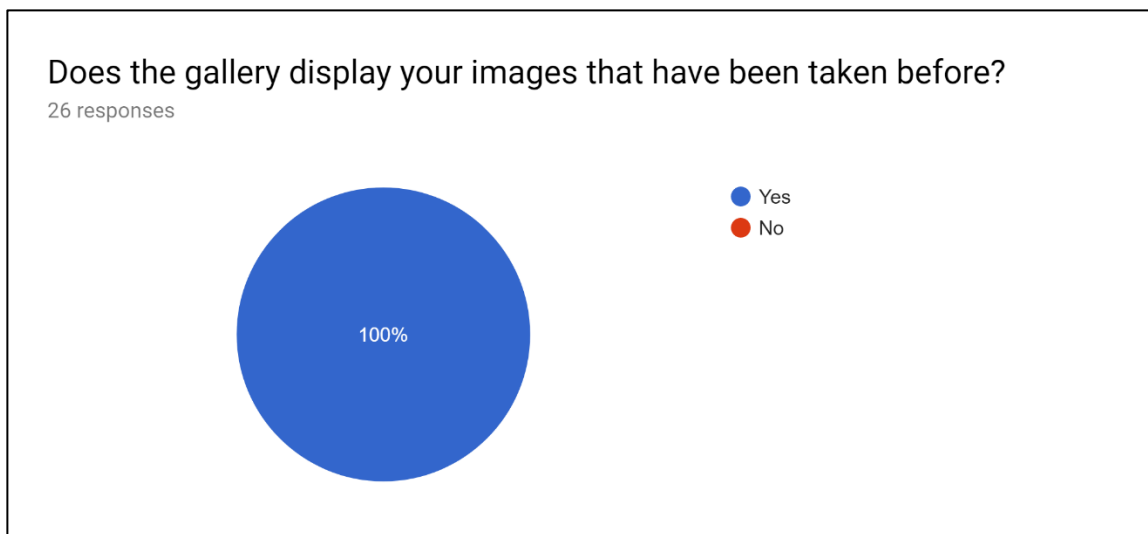


Figure 4.44 Question 6

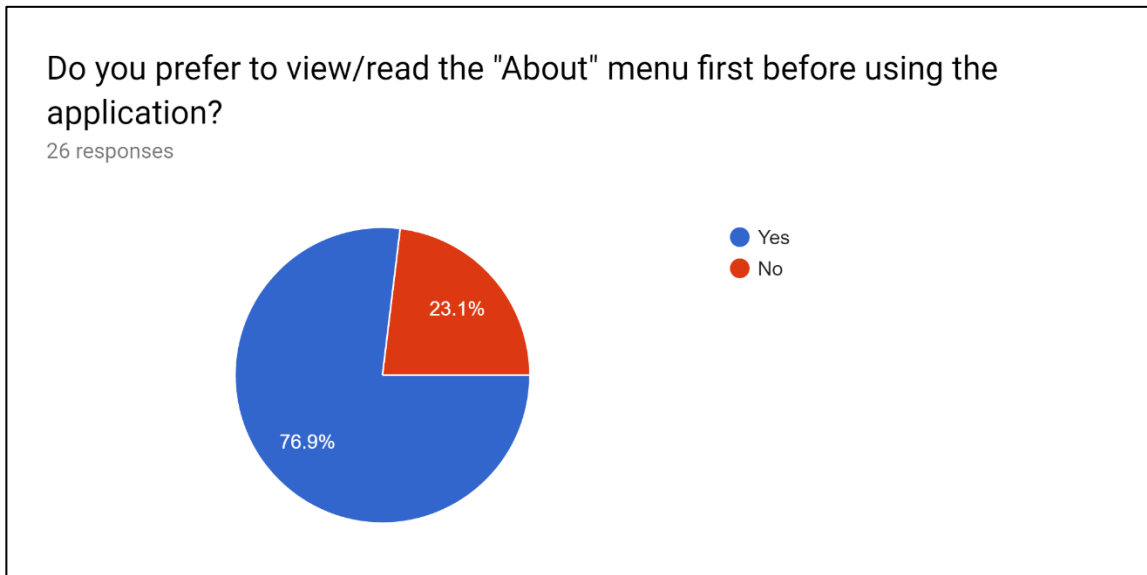


Figure 4.45 Question 7

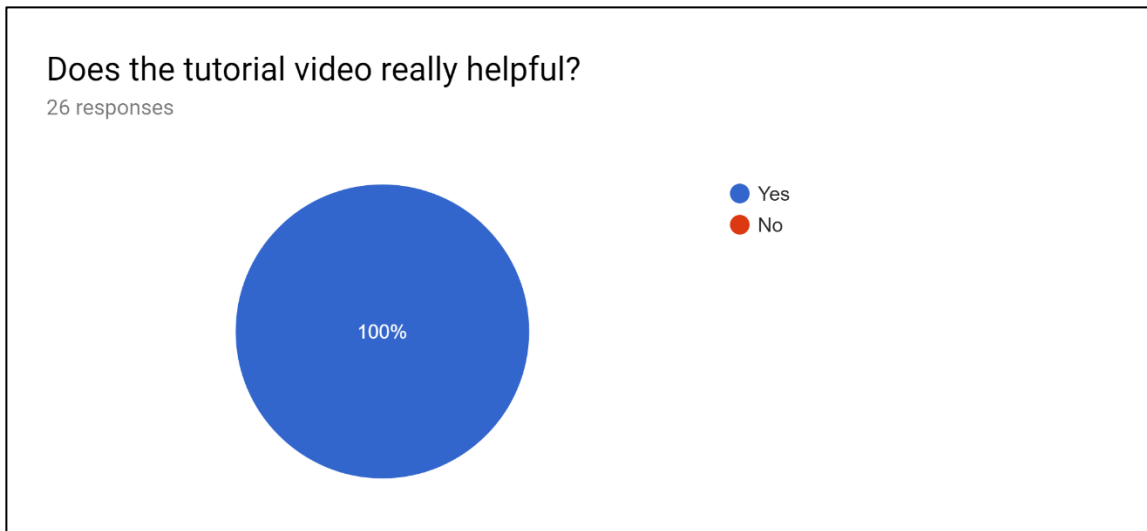


Figure 4.46 Question 8

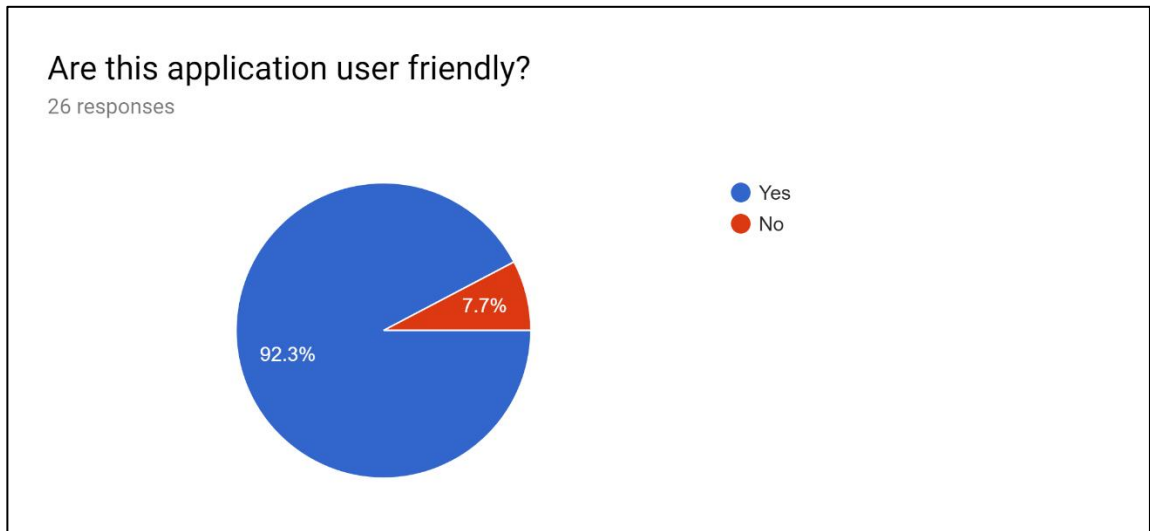


Figure 4.47 Question 9

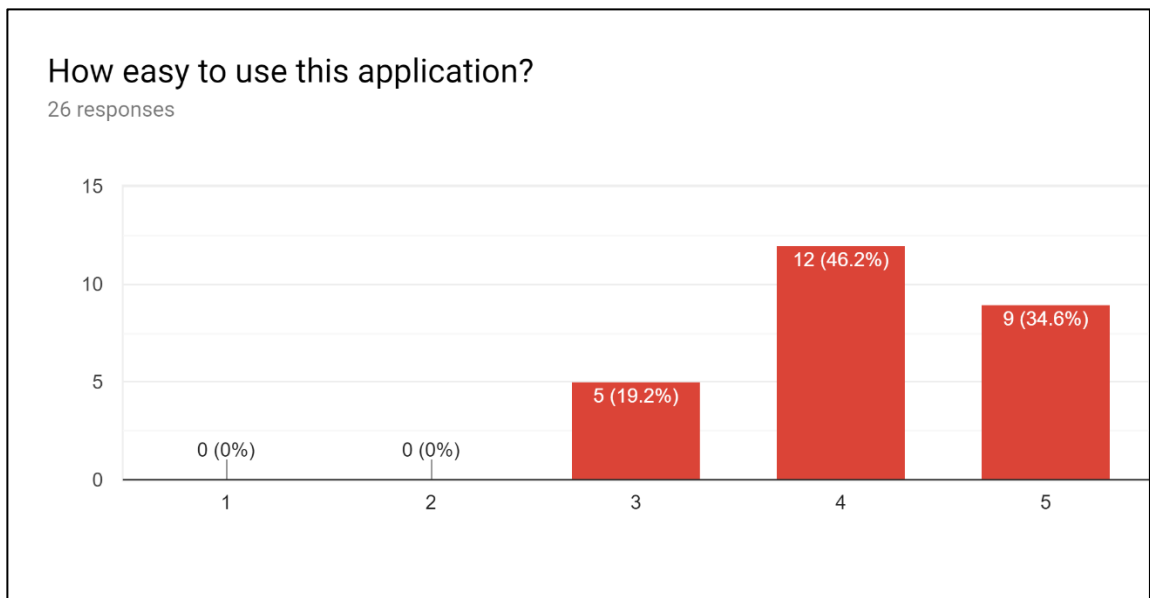


Figure 4.48 Question 10

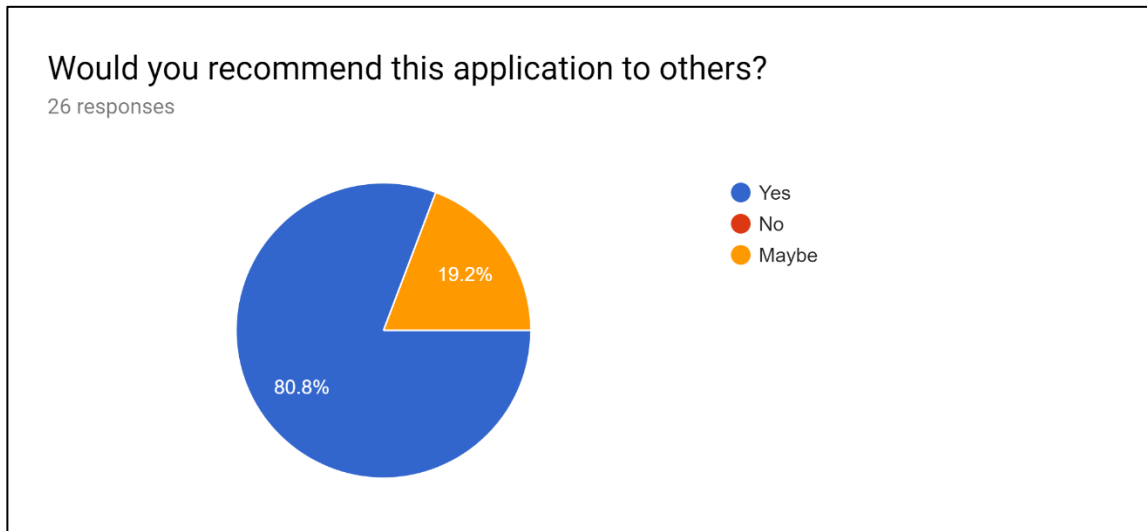


Figure 4.49 Question 11

4.5.2 Result Report

Based on the testing, it can be concluded that most of the users agree that this application is really suitable for interior design. The result is shown in **Error! Reference source not found.** showing 57.7% of users who said that this application suitable for interior design and the other 7.7% and 34.6% for scale of 3 and 4 respectively.

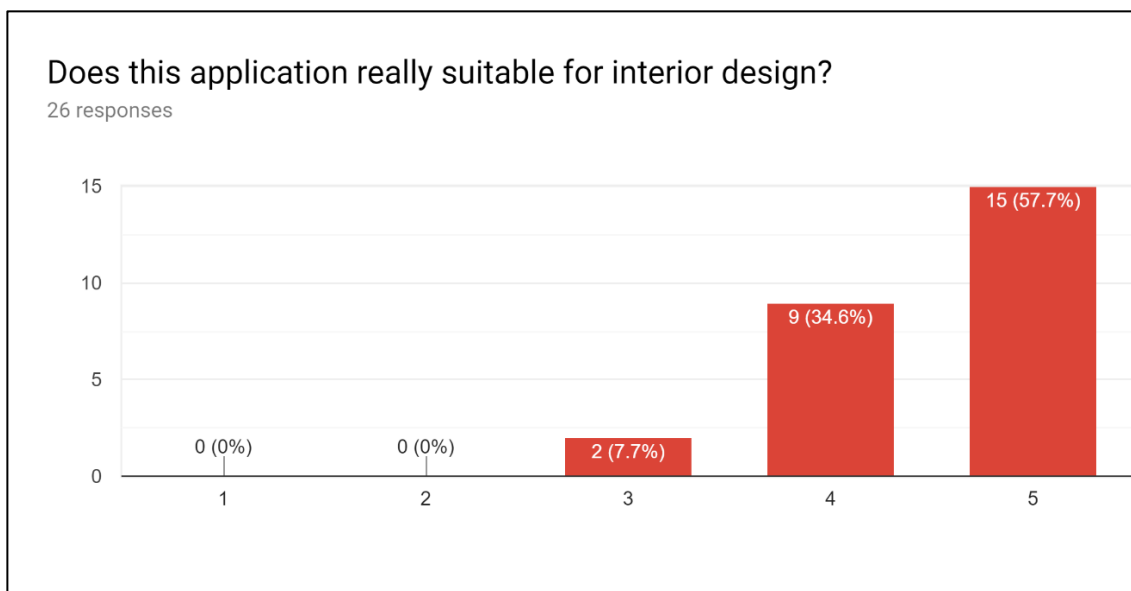


Figure 4.50 Question 12

CHAPTER 5

CONCLUSIONS

5.1 Introduction

This chapter will conclude all the aspects of An Interactive Mobile Markerless Augmented Reality for Interior Design. User does not need to use any marker to produce interactive apps with the 3D model. An Interactive Mobile Markerless Augmented Reality for Interior Design is an application that allows user to decorate their room virtually without using any marker with advance tracking technology and 3D models. User can freely decorate the room based on 72 models of furniture with various colour and design.

This application also provides function such as to take pictures of designed room. So that, user can preview the images many times and it can be as a guideline for them. However, the most important component is user are freely to drag, drop, and rotate the 3D models. Hence, it can prevent user from wasting time, energy, and money. User does not need to move around the furniture and check for suitability of the furniture to fits into their room. At the end of this project, all the objectives already achieved.

Rapid Application Development was chosen as a methodology for this project to ensure the success of the project development. The phases in RAD model consist of Requirements Planning, User Design, Construction, and Cutover. The flow of the RAD really makes this project runs smoothly. It fits well towards any problems and situation. The time and task management go along together as planned before. I hope there is a way

to make user customize the colour of furniture by themselves using colour picker. It just only a way it can be customize by using marker-based augmented reality only. I also hope this application can help others too.

5.2 Research Constraint

There are several constraints when conducting this project:

i. Track the scene.

User must have enough light and the floor must be have some pattern and not only plain.

ii. Smartphone.

User's smartphone must support the ARCore by Google that can be found in Google Play Store. User must meet this requirement specification in order to install the application in the smartphone:

- Android 4.4+ (API Level 19+), High resolution devices (hdpi), Camera, OpenGL 2.0 (or newer), Devices with a quad-core CPU (armv7a with NEON support or armv8a), Compass, and Accelerometer

iii. License key of Wikitude SDK

Students and educators are entitled to one license per platform (iOS or Android). The Wikitude EDU license has a time restriction of up to one year and publishing on the App stores is not permitted. The license may not be resold or shared. Subscriptions must be used exclusively by the educator or student.

iv. Colour Picker

The colour for each type of furniture consist only 4 colour. For markerless AR it seems like it cannot support the function of colour picker. It only can

support for marker-based AR. It is no possible way to implement the colour picker for markerless AR project.

5.3 Future Work

In advance, with the constraint that faced, several future works to improving the application were identified in order to develop a better application in the future. The application may have better tracking function and easier to track the scene. Next, the license key is for educational purpose only. When user use the trial version, there are some features were blocked from using it. The whole screen will have trial watermark and it will have logo of Wikitude. I hope faculty have a permission of using the Wikitude license without strict terms and conditions from Wikitude itself. Furthermore, for better performance and efficiency, I hope there is a way to make the colour pickers for the furniture. So, user can freely pick the colour as they want. The colour picker can be done with the AR marker-based only.

REFERENCES

- Amir Ghahrai. (2017). Iterative Model in Software Development and Testing. Retrieved May 1, 2018, from <https://www.testingexcellence.com/iterative-model/>
- Arbeláez-estrada, J. C., & Osorio-gómez, G. (2013). Augmented Reality Application for Product Concepts Evaluation. *Procedia - Procedia Computer Science*, 25, 389–398. <https://doi.org/10.1016/j.procs.2013.11.048>
- Carvalho, E., Mações, G., Varajão, I., Sousa, N., & Brito, P. (2009). Use of Augmented Reality in the furniture industry, (June).
- David L. (2014). ADDIE Model - Learning Theories. Retrieved May 1, 2018, from <https://www.learning-theories.com/addie-model.html>
- Jason Fitzpatrick. (2010). Roomle Brings Your Room Designs to Life; Furnishings and All. Retrieved March 11, 2018, from <https://lifelifehacker.com/5554266/roomle-brings-your-room-designs-to-life-furnishings-and-all>
- Miyake, M., Fukuda, T., Yabuki, N., & Motamedi, A. (2017). Outdoor markerless augmented reality. *CAADRIA 2017 - 22nd International Conference on Computer-Aided Architectural Design Research in Asia: Protocols, Flows and Glitches*, 95–104.
- Phan, V. T., & Choo, S. Y. (2010). Interior Design in Augmented Reality Environment. *International Journal of Computer Applications*, 5(5), 16–21. <https://doi.org/10.5120/912-1290>
- Shedole, A., Satpute, S., Singh, A., & Pingale, G. (2015). ISSN : 2347-7385 Markerless Augmented Reality For Interior Designing Using Android, 03(06), 10–14.
- Uplaonkar, D., Saoji, S., Paranjape, S., Andhalkar, N., Chorge, R., & Jainapur, R. (2015). Virtual Furniture Application Using Augmented Reality, 3(1), 1–4.

What are augmented reality markers ? (2013). Retrieved March 8, 2018, from <https://anymotion.com/en/wissensgrundlagen/augmented-reality-marker>

Yariv Levski. (2016). Markerless vs. Marker Based Augmented Reality - AppReal. Retrieved March 8, 2018, from <https://appreal-vr.com/blog/markerless-vs-marker-based-augmented-reality/>

APPENDICES

APPENDIX A

USER ACCEPTANCE TEST

**USER
ACCEPTANCE
TEST (UAT)**

TABLE OF CONTENT

1.0	User Acceptance Test	133
2.0	System Testing Approval	136

1.0 User Acceptance Test



The chosen user has been going through each of the instruction in the user manual. Any errors or problems found by the user was noted on this form.

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Main interface	Main interface of Unity shows up.	Unity engine start the interface and showed up the interface after clicked on the Unity icon.	Unity engine start the interface and showed up the interface after clicked on the Unity icon.	Pass	
Tracking	The application start tracks the room	Tracking is successful.	Tracking is successful.	Pass	
Home Interface	Home interface of Unity shows up	Home interface showed up on user's device	Home interface showed up on user's device	Pass	

Load 3D models	Various of 3D models	The application displays the 3D models.	The application displays the 3D models.	Pass	
Take images	Test to take images	User can take images	User can take images	Pass	
Gallery interface	Test to preview the images taken before in gallery interface.	The application display images that have been taken.	The application display images that have been taken.	Pass	
Setting interface	Test the functionality of the setting interface.	The about menu and tutorial menu can be opened.	The about menu and tutorial menu can be opened.	Pass	
About interface	Test all the functionality in the about menu interface.	The list of the furniture and the application can be opened.	The list of the furniture and the application can be opened.	Pass	

Tutorial interface	Test the video playback	The video can be played.	The video can be played.	Pass	
Buttons function	Test all button function	All buttons are function as well	All buttons are function as well	Pass	
3D model's movement	Rotate, drag and drop	All movement are available	All movement are available	Pass	

2.0 System Testing Approval

	Name	Date
<p>Verified by:</p>  <hr/> <p>Developer</p>	<p>NUR DZAQIRAH BINTI MOHD TARMIDI</p>	<p>3/10/2018</p>
<p>Approved by:</p>  <hr/> <p>Client</p>	<p>NUR AMIRA BAIDURI BINTI AZLAN</p>	<p>3/10/2018</p>

APPENDIX B

USER MANUAL

USER

MANUAL

TABLE OF CONTENTS

	PAGE
01 GENERAL INFORMATION	139
1.1 System Overview	139
02 SYSTEM SUMMARY	139
2.1 System Configuration	139
2.2 Contingencies and alternate modes of operation	140
03 GETTING STARTED	140
3.1 Installation	140
3.2 System Menu	142
3.2.1 Tracking Interface	142
3.2.2 Home Interface	142
3.2.3 Place the 3D models	143
3.2.4 Gallery Interface	143
3.2.5 About and Tutorial Menu	144
3.2.6 About Interface	144
3.2.7 Tutorial Interface	145
3.3 Starting the application	146

01 GENERAL INFORMATION

In this section, it will explain the system overview of the application.

1.1 System Overview

Markerless is one of the techniques in the Augmented Reality system. User does not need to use any marker to produce interactive apps with the 3D model. An Interactive Mobile Markerless Augmented Reality for Interior Design is an application that allows user to decorate their room virtually without using any marker with advance tracking technology and 3D models. 3D models are load from the existing database. It can make user's task easier to decorate the room because the 3D models and environment is almost same as real life. User can freely decorate the room based on 72 models of furniture with various colour and design. It is very convenient because it is only used smartphone as hardware component and software generated from computer that need to install into the smartphone. User also enable to rotate, capture the designed room, view the captured images, reset, drag and drop to rearrange the furniture based on suitability in the room. An interactive mobile markerless augmented reality for interior design is simple application which means easy to understand by recognizing the icons that represent the furniture, so it can be use by people in all ages. Rotate, drag, and drop functions are made to let user see more detail about the furniture.

02 SYSTEM SUMMARY

In this section, it will discuss the system configuration and also the contingencies and alternate modes of operation.

2.1 System Configuration

An Interactive Mobile Markerless Augmented Reality for Interior Design application operates on mobile devices with Android operating system. It is compatible with Android 4.4+ (API Level 19+), High resolution devices (hdpi), Camera, OpenGL 2.0 (or newer), Devices with a quad-core CPU (armv7a with NEON support or armv8a), Compass, and Accelerometer. User's smartphone must support the ARCore by Google

that can be found in Google Play Store. The application does not need the internet connection. After installation on the devices, An Interactive Mobile Markerless Augmented Reality for Interior Design application can be used immediately without any further configuration.

2.2 Contingencies and alternate modes of operation

In case the surrounding does not enough lights, the tracking will be failed. Furthermore, in case the floor surfaces is too plain, the tracking will be a little bit difficult.

03 GETTING STARTED

3.1 Installation

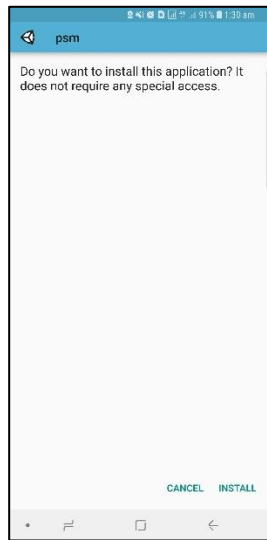


Figure 1 Installation



Figure 2 Installing

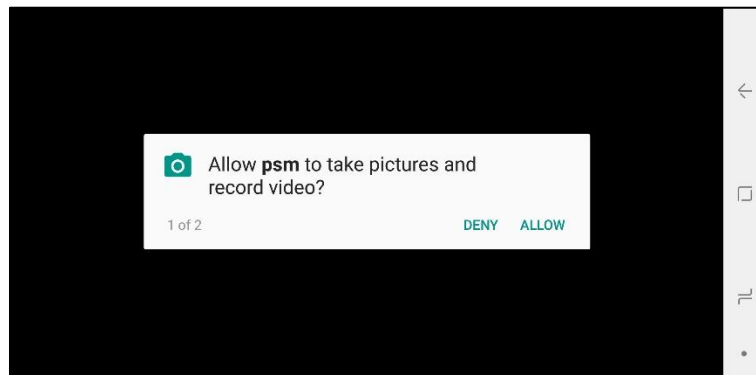


Figure 3 Permission to take pictures and record video

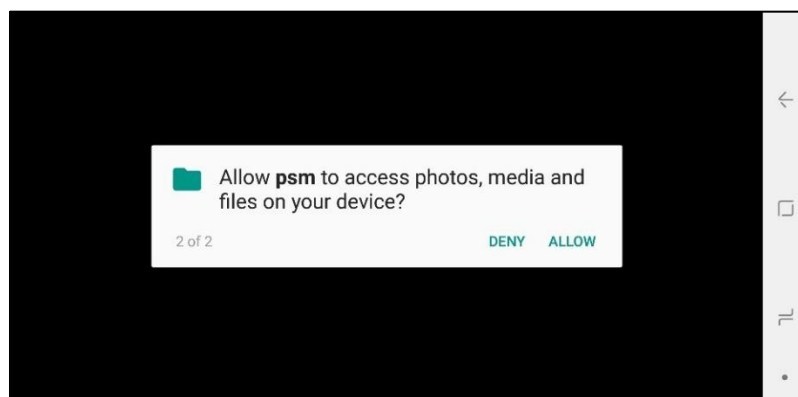
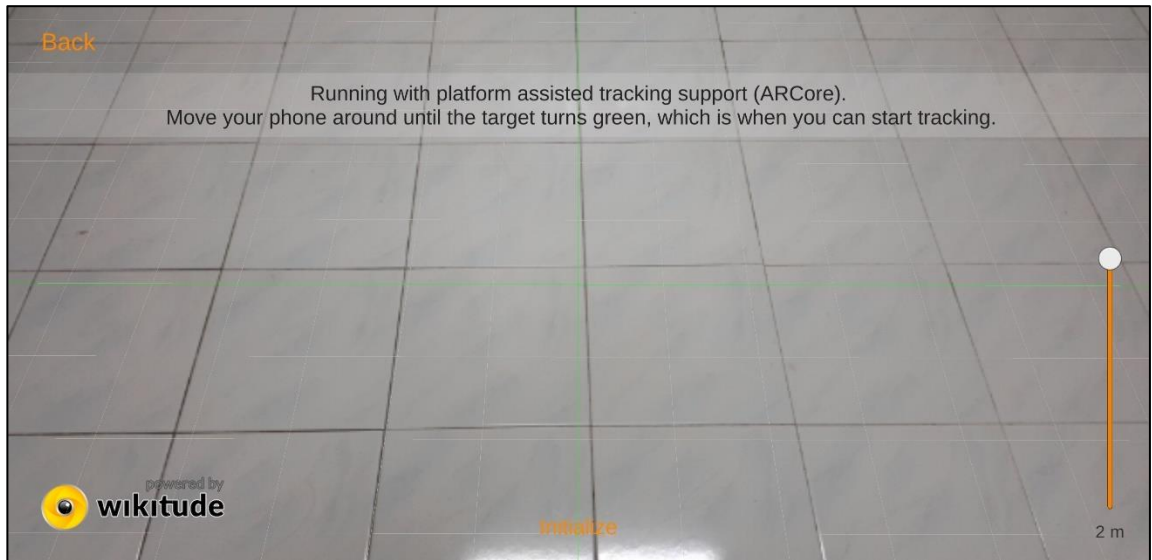


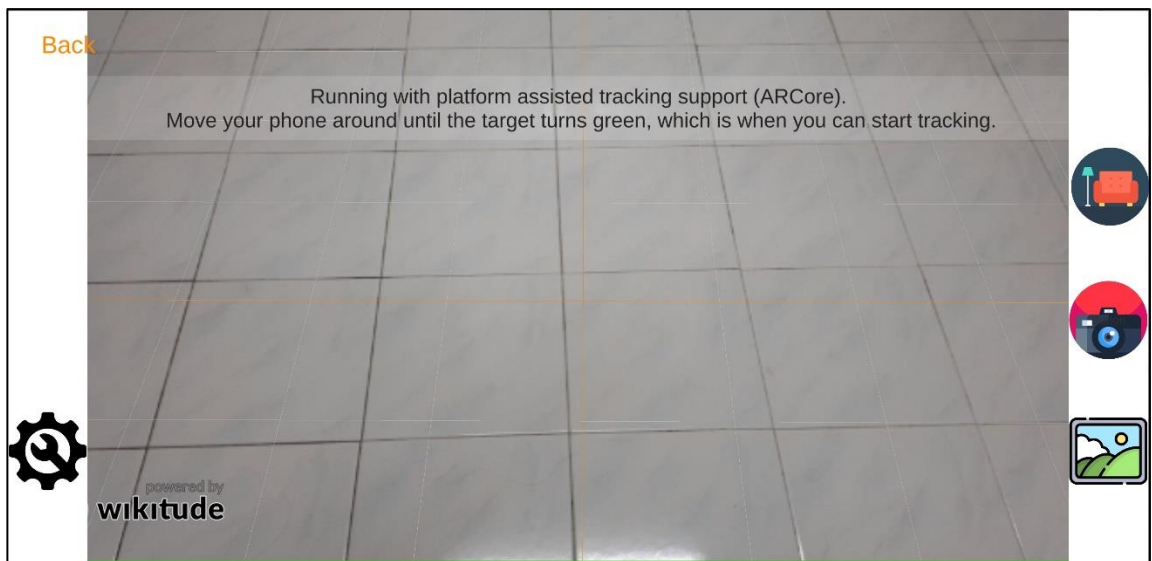
Figure 4 Permission to access photos, media, and files on the device

3.2 System Menu

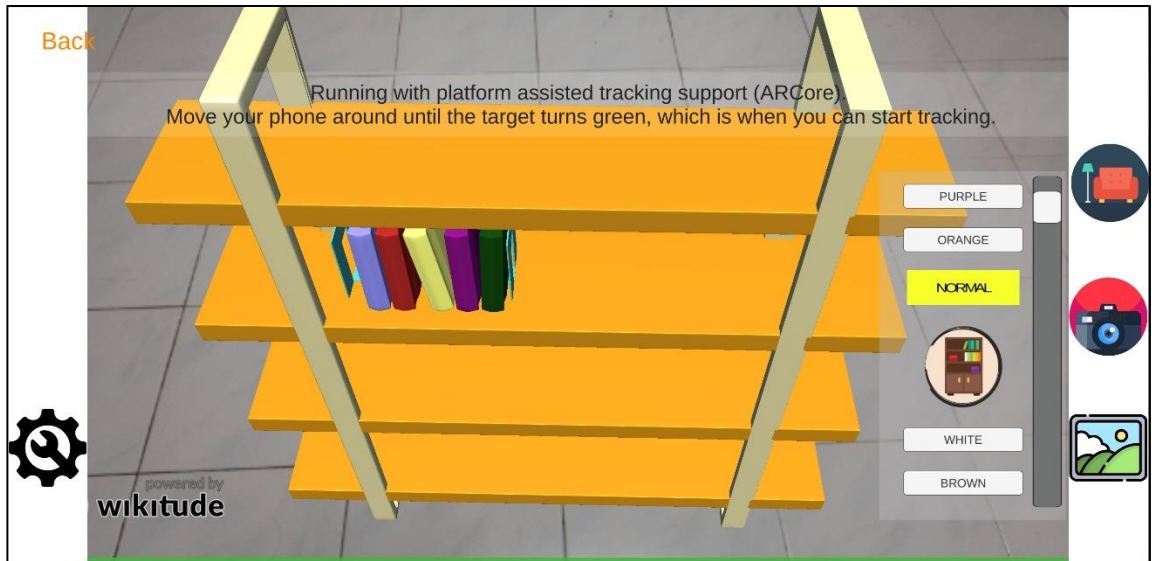
3.2.1 Tracking Interface



3.2.2 Home Interface



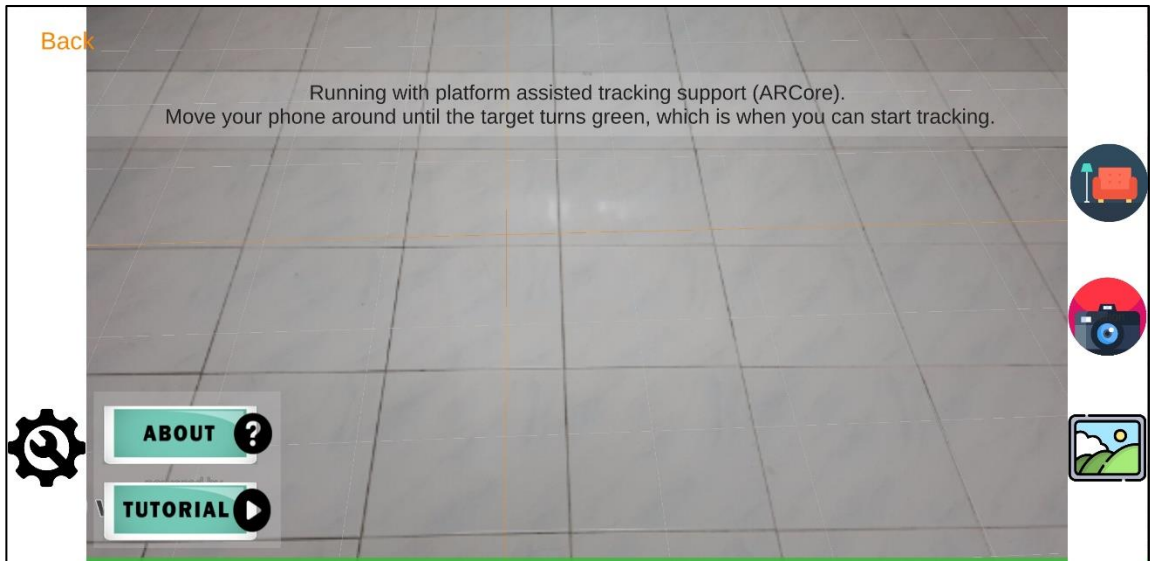
3.2.3 Place the 3D models



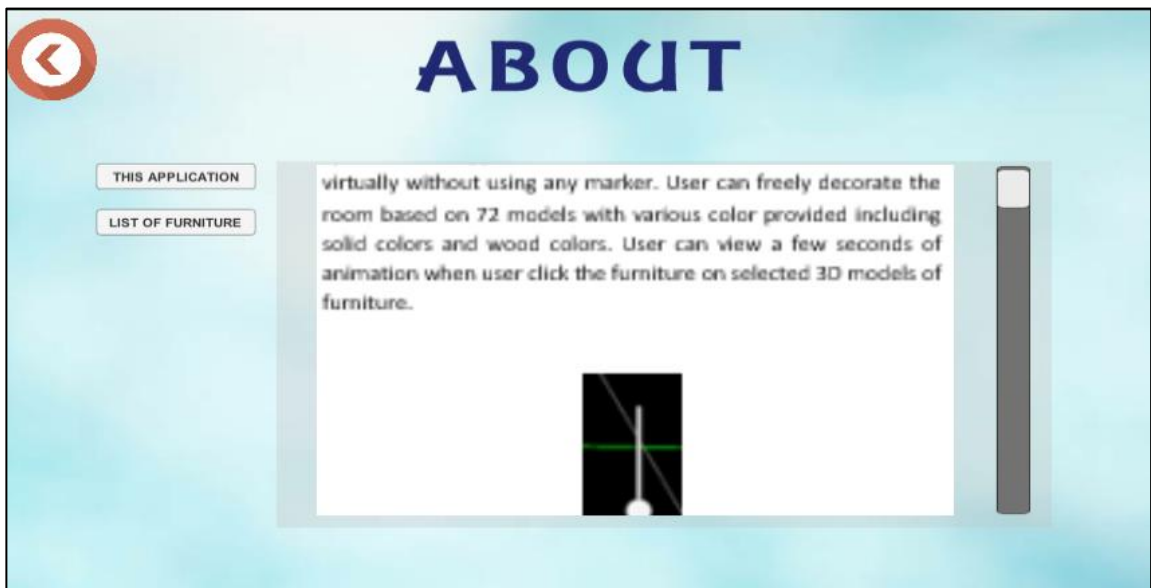
3.2.4 Gallery Interface



3.2.5 About and Tutorial Menu

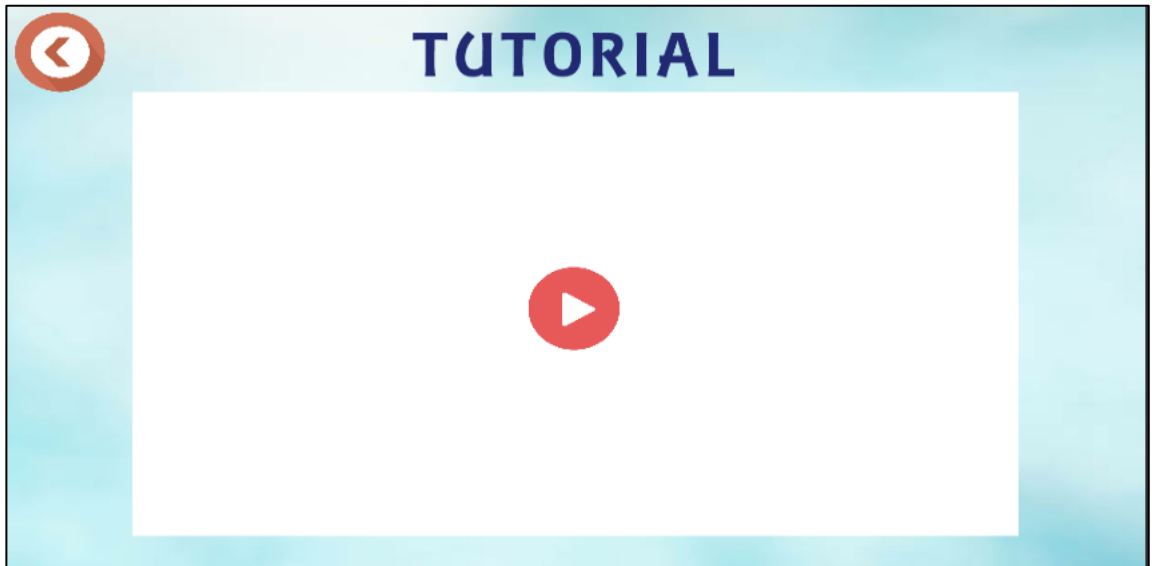


3.2.6 About Interface

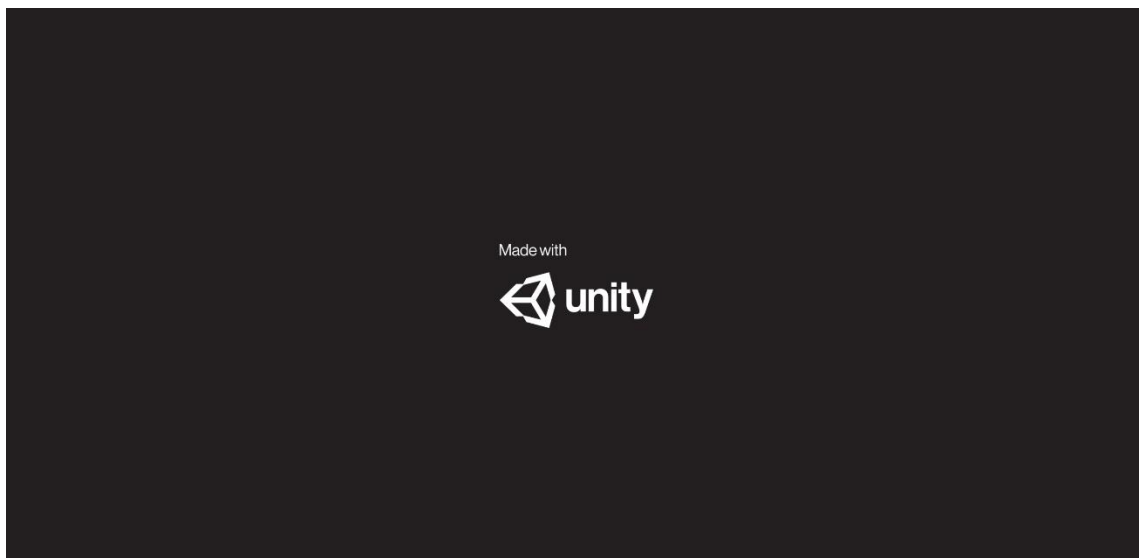




3.2.7 Tutorial Interface



3.3 Starting the application



APPENDIX C

Usability Test

**USABILITY
TEST**

This is the list of usability test (questionnaire) that have been given to the users.

AN INTERACTIVE MOBILE MARKERLESS AUGMENTED REALITY FOR INTERIOR DESIGN

This is a usability test that collect responses from any user.

⋮

What do you think about the interface design of the application? *

	1	2	3	4	5	
Very Dissatisfied	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Satisfied

Does the tracking scene perfectly functioning? *

- Yes
- No

Does the 3D models appear correctly? *

Yes

No

Does the 3D models can rotate, drag and drop? *

Yes

No

Does the camera (take the pictures), you fully understand how does it works? *

Yes

No

Maybe

Does the gallery display your images that have been taken before? *

Yes

No

Do you prefer to view/read the "About" menu first before using the application? *

Yes

No

Does the tutorial video really helpful? *

Yes

No

