**Dharmsinh Desai University, Nadiad**



**Faculty of Technology**

**Department of Computer Engineering**

**B. Tech**, Computer Engineering (**CE**)

**Sem** - **VI**

**Subject**: System Design Practice

**Project** **Title**: *Augmented Reality and It’s use cases*.

**By**: Moxank Patel CE-001

Aatish Chaudhari CE-003

Jaimin Chauhan CE-018

**Guided By**: Prof. Jigar M Pandya

**CERTIFICATE**



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

This is to certify that the project work titled

***Augmented Reality and It’s use cases***

is bona fide work done by

***Moxank Patel,***

***Aatish Chaudhari,***

***Jaimin Chauhan***

carried out in the partial fulfillment of term work for the subject Advanced Technologies in

The academic session

2019-2020

**DR C. K. Bhensdadia** **Prof. Jigar M Pandya**

Dept. Of Computer Eng. Dept. Of Computer Eng.

**Table of Contents**

1. Abstract (What is AR?) 4
2. Technologies Used 6
3. SRS 7
4. Diagrams 16
5. Deployment Descriptor 20
6. Test Case 23
7. Versioning 25
8. Screenshots 26
9. Conclusion 39
10. Limitation and Future Extension 42
11. References 43

**What is Augmented Reality (AR)???**

* It’s an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic and olfactory (smell).
* Augmented reality is the result of using technology to superimpose information —sounds, images and text.
* In simple words adding something virtually to real world object. It can be visual, audio or any other similar stuff



You might have come across this popular game ‘Pokémon Go’, it may come as surprise but it is uses Mixed Reality Concept which has partial use of Augmented Reality.

**Technologies (Behind the scene)**

We have majorly used Google AR Core library for implementing use case. We have used Google AR Core because it is open source and currently in trend. Other library such as Vuforia is also used to demonstrate a simple use case of object augmentation. Vuforia wasn’t used in other use cases because of its limited support.

Unity is used to develop application making our application almost platform independent.

C# is language behind development as unity development can be done with help of C#.

Google Cloud Service is being internally used by Google AR Core library to retain some of the information.



*Google AR-Core Vuforia Unity*

**Table of Contents of SRS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Type** | **Why it is/not created** | **Phase in which it is generated** | **Is Implemented** |
| SRS | It is created to represent project idea to developers. So that, they can have clear cut idea about what they are developing and how they proceed towards development. | Initial phase.  During planning, requirement gathering stage it was being prepared. | Yes |
| ER Diagram | It is entity relationship diagram which is prepared to represent dependency and relationship among table.  And in our project, there no major uses of database, so it is omitted. |  | No |
| DFD Diagram | It is created to represent data flow among class.  Here in our project there are no classes which are implemented externally, all classes in out project are derived from unity or google arcore library.  And we just integrated unity classes and google arcore classes with the help of function and unity life cycle.  So, we don’t require DFD. |  | No |
| Class Diagram | As mentioned above that, we didn’t create any external classes.  So, no Class Diagram would there.  There were no classes created because while developing apps in unity it follows certain life cycle.  It is,  Initialization,  Update,  Destroy.  And only that classes were created with no meaning to entity. |  | No |
| USE Case Diagram | It is created to represent interaction of particular functionality with a certain type of user and also to show dependency among them. | It created right after SRS is created. | Yes |
| ACTIVITY Diagram | It is used to represent the flow of system.  As mentioned earlier unity follows certain life cycle.  So, every time same activity/pattern is followed, hence no activity diagram. |  | No |
| Sequence Diagram | It is created to represent flow among objects or in other words how they interact.  And we don’t have any external meaningful classes and also the classes/script which are there aren’t interacting with each other as they loosely coupled. |  | No |
| Component  Diagram | It is used to represent abstract view of classes. |  | No |
| Deployment Diagram | It is used to describe system hardware level |  | Yes |

**SRS**

SRS for **LiveNewsPaper**

#List of Users

1. End User

2. Admin

#SRS

\* Higher Level Requirements

1. Add Video

2. View Video

##Functional Requirements

1. Add Video

1.1 Scan Image

I/p: Input Image

O/p: options of browsing corresponding video is displayed

1.2 Attach Video

I/p: Path to video

O/p: successful added message or error message

2. View Video

I/P: Scan Image

O/P: Image is replaced by video

SRS for **ObjectAugmentation**

#List of Users

1. End User

#SRS

\* Higher Level Requirements

1. View Object

##Functional Requirements

1. View Video

I/P: Scan Image

O/P: Image is replaced by object (Cube)

SRS for **TreasureHunt**

#List of Users

1. Client
2. Host

#SRS

\* Higher Level Requirements

1. Add Room

2. Join Room

3. Add Anchor

4. Add Star

5. Add Text

##Functional Requirements

1. Add Room

I/P: Host clicks on button

O/P: Room is created and displayed

1. Join Room

I/p: Client enter room id

o/p: successful message or retry message is rendered

1. Add Anchor

I/P: Client click on screen

o/p: anchor point object is spawned

1. Add Star

I/P: Client click on screen

o/p: star object is spawned

1. Add Text

I/P: Client click on text area

o/p: text is updated

SRS for **FaceFilt**

#List of Users

1. End User

#SRS

\* Higher Level Requirements

1. View Thought

##Functional Requirements

1. View Thought
   1. Get Face

I/P: Scan Image

O/P: Image is scanned and set coordinates for face are returned.

* 1. Augment Object

I/P: Game object [Filter Mesh] from unity

O/P: Augmented Face

* 1. Render Thought

I/P: Coordinates of face

O/P: Randomly selected thought

**Use Case Diagram**

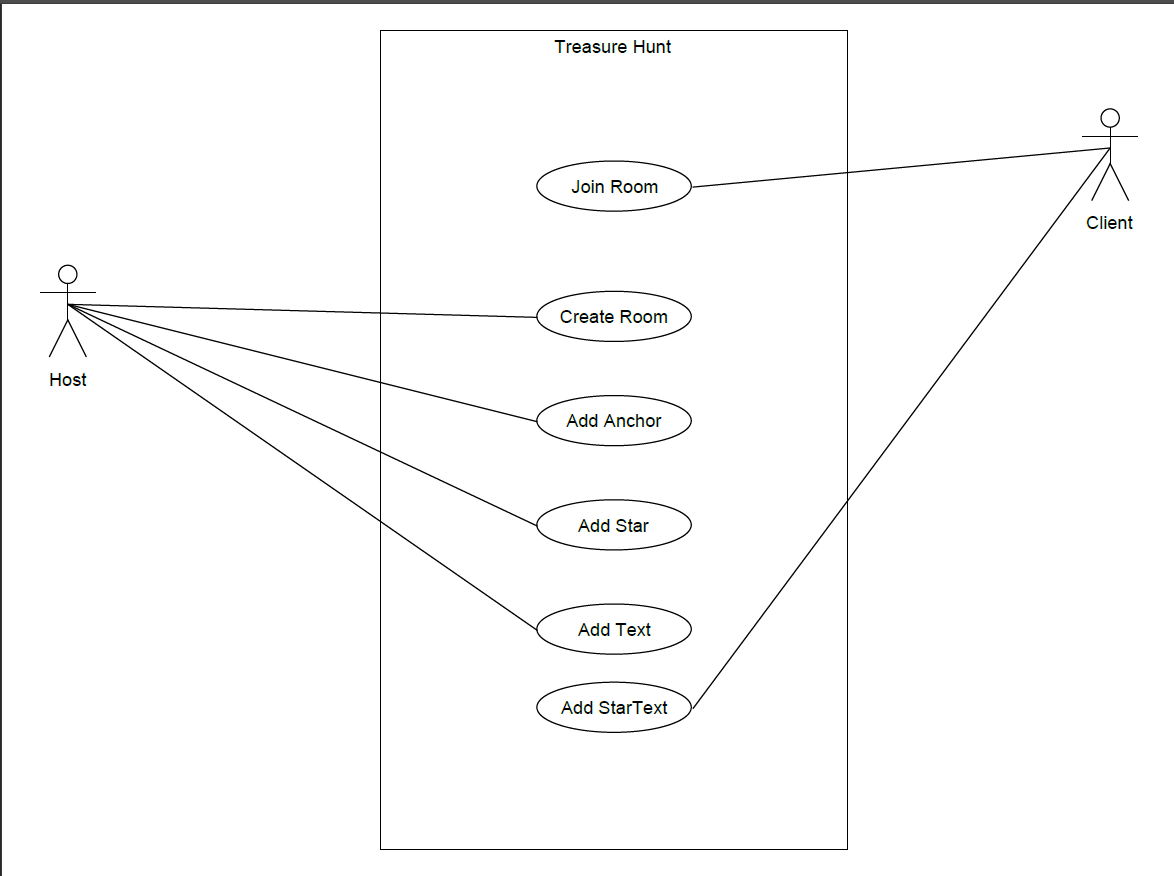


Figure 1. Treasure Hunt Use-Case

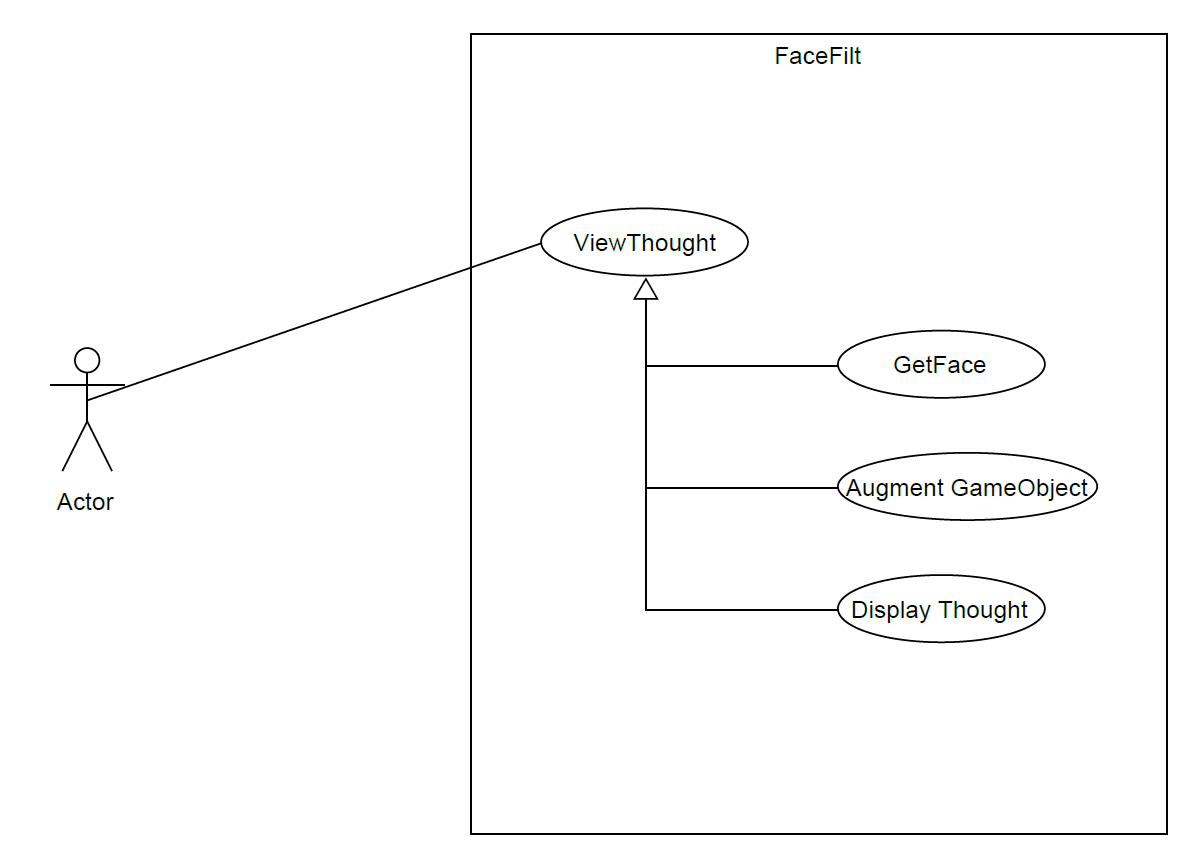


Figure 2. Face Filter Use-Case

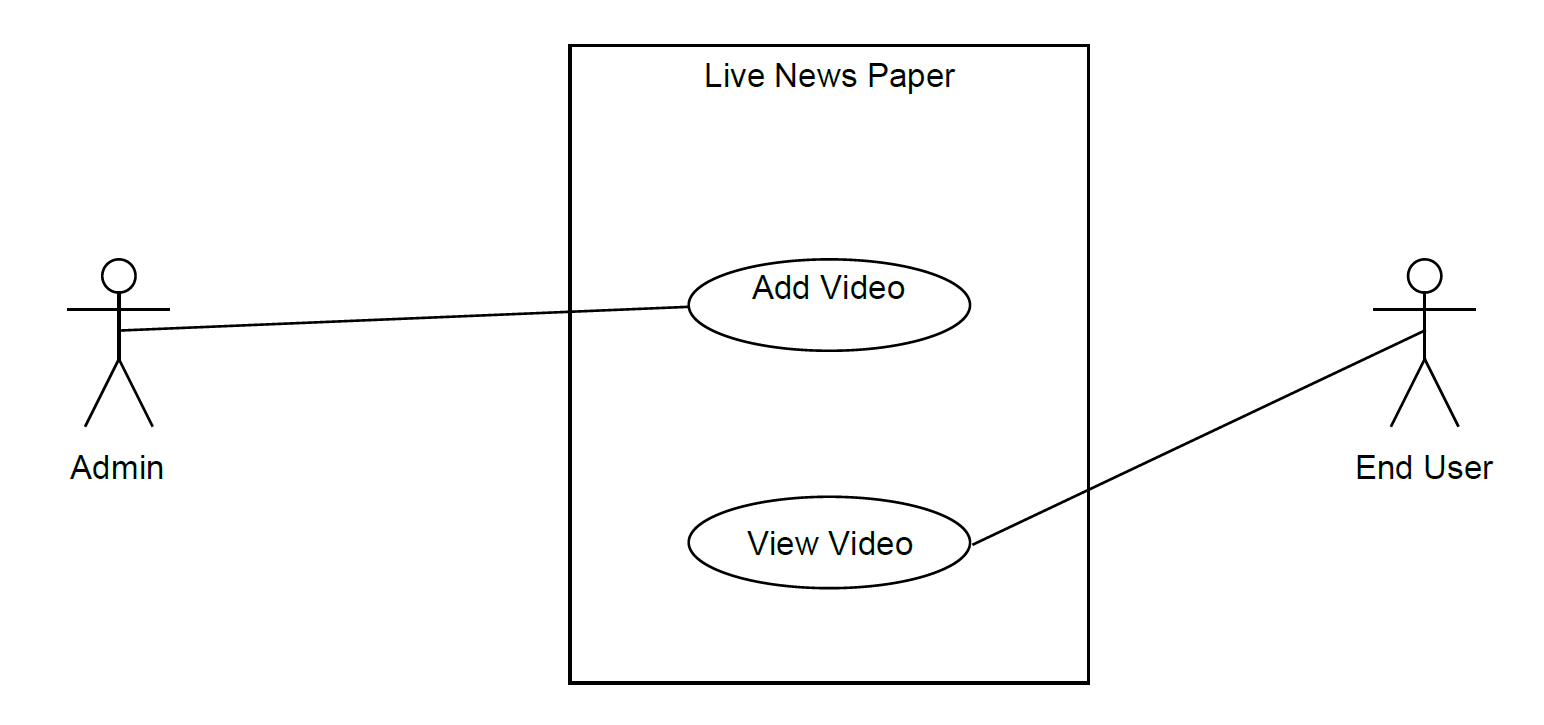


Figure 3. Live News Paper Use-Case

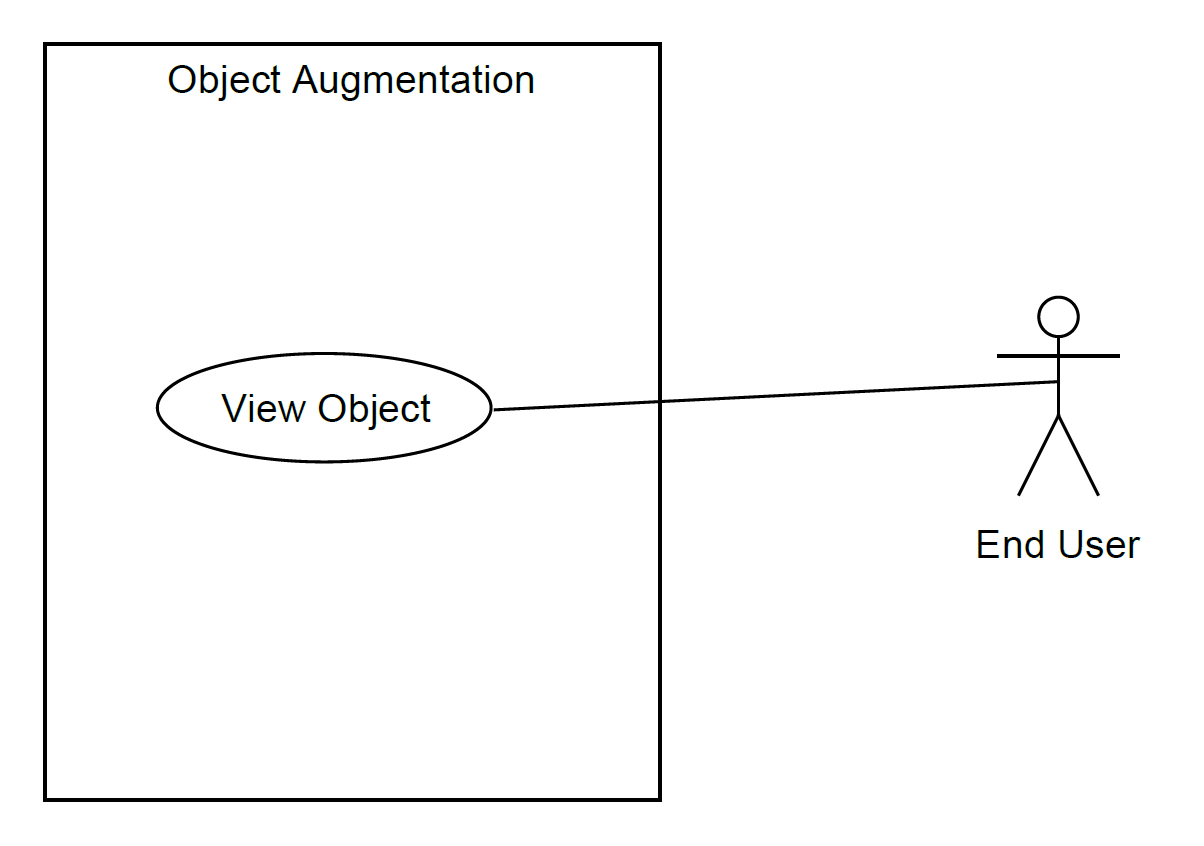


Figure 4. Object Augmentation Use-Case

**Deployments Diagram**

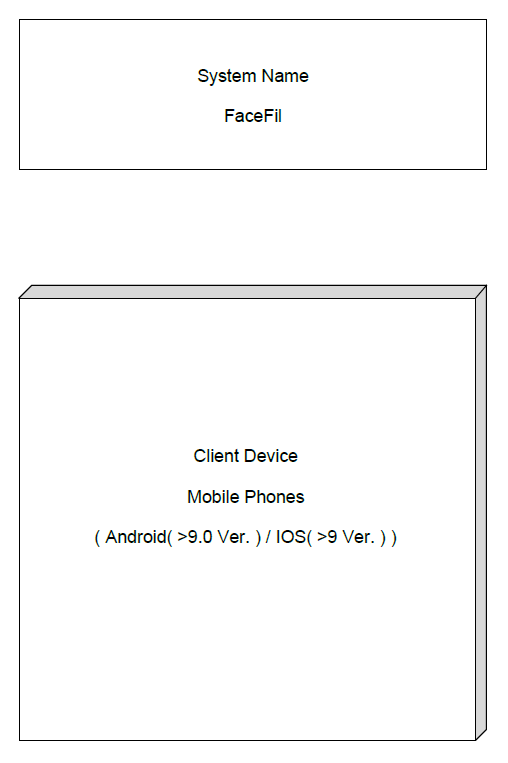


Figure 5. Deployment Diagram for Face Filter

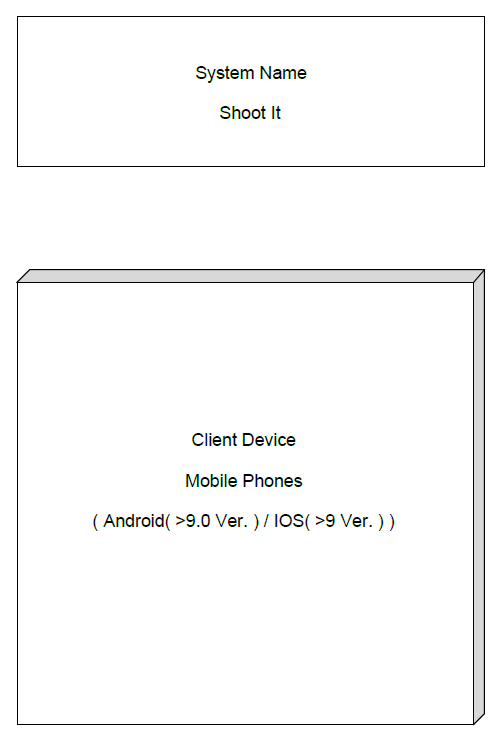


Figure 6. Deployment Diagram for Face Filter

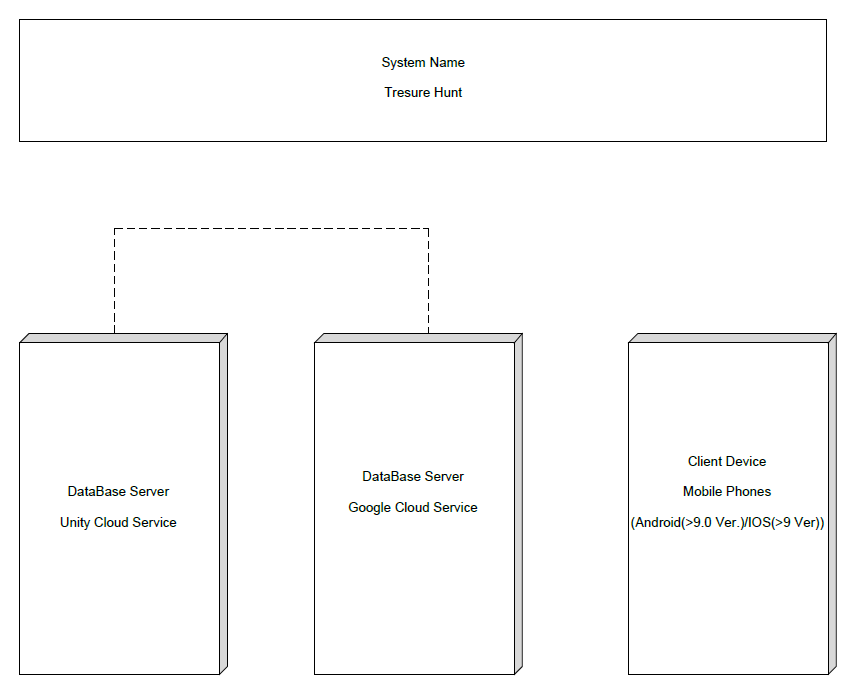


Figure 7. Deployment Diagram for Treasure Hunt

**Deployment Descriptor**

1. Generate signed .apk/ .abb using .jsk file
2. Make Play Store Developers account.
3. Create new project in account.
4. Add Screen Shot of application and privacy policy
5. Create release on the Play Console
6. Upload .apk or .abb file
7. Enter your app details.
8. Apply for approval.
9. After approval select devices on which you want to publish your app
10. After approval app would be published.

**Test Case**:

Manual Testing was done.

Those cases are depicted in below section of screenshots.

**Implementation Details:**

No new modules were created, for any of the use case.

Only scripts were created for object manipulation.

Major of functionality which we have used are inherited for Google AR-Core library and therefore we were not required to create any new major functionality for any of use case.

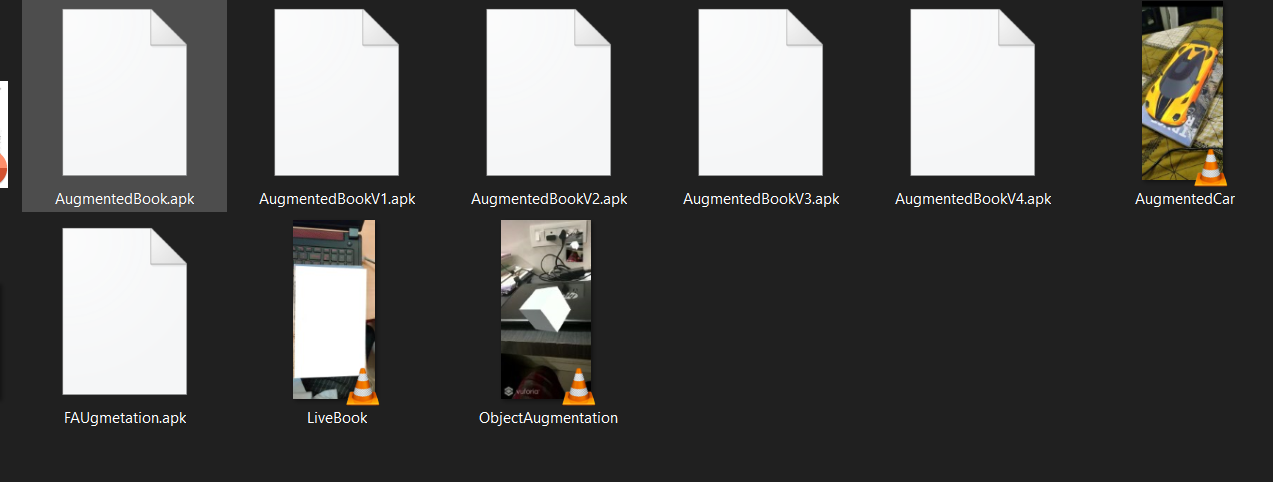
The function which were created were under the script for manipulating the button, resizing the object and etc.

We have just integrated Google AR-Core library service with unity app creation module.

Some of the major functionality which were needed for completion of task were implemented under lifecycle of unity script.

For example, in Treasure Hunt we needed a feature were when user is in certain range of the clue then only, they would be able to view the clue else they clue remain hidden. It was implemented under one the method that is inherited method of MonoBehavior (Unity Script).

**Versioning**



We needed to maintain version in Live Book and for Treasure Hunt.

Their deployed apk were stored according to their version as we can see above.

**Screenshots**

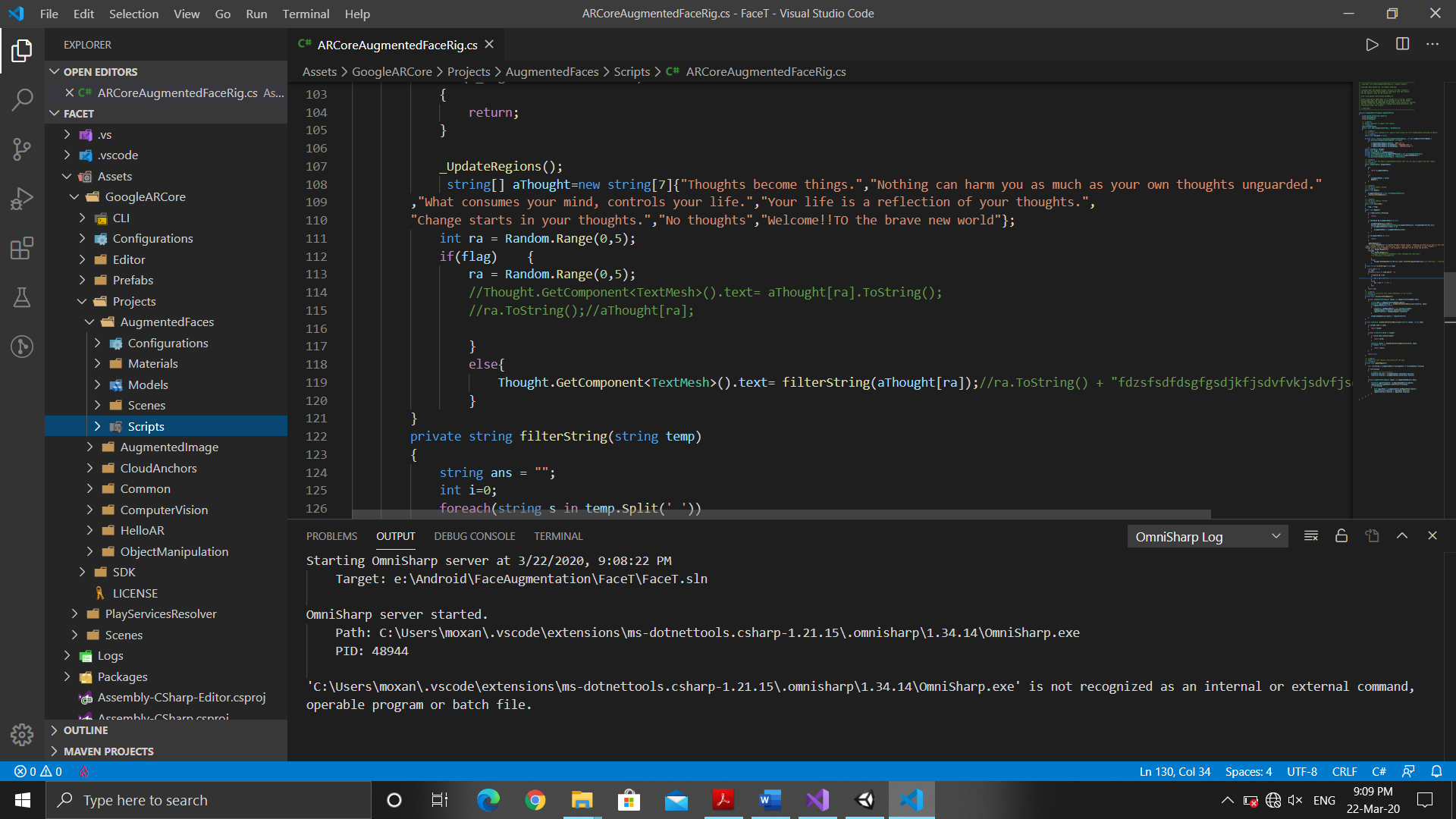


Figure 8. Hierarchy of Project

Google AR-Core: This folder provides modules implementation for some of functionality of AR like augmenting images

and likewise. Moreover, it detects images and gives us coordinate. Some of the other functionality provided by Google-AR Core are:

1. Provide Database
2. Provide Cloud DB Service
3. Detects Image
4. Provides Support for Real Time Sharing, ultimately making it easier to manage multiple players in a game if any.

**Scenes**:

It contains the environments and menus of the game and we can think of it as unique level represents a unique scene.

**Scripts:**

It is folder containing script for manipulating objects of unity. Mainly contains C# code files.

**Models:**

Model are the files that are placed in the Assets folder in project and are automatically imported and stored as Unity Assets. A model file can contain a 3D model, such as a character, a building, or a piece of furniture. The model is imported as multiple Assets.

**Configuration**:

As name itself suggest. Contains configuration for project and is created while unity project is created and developer can edit it according their needs.

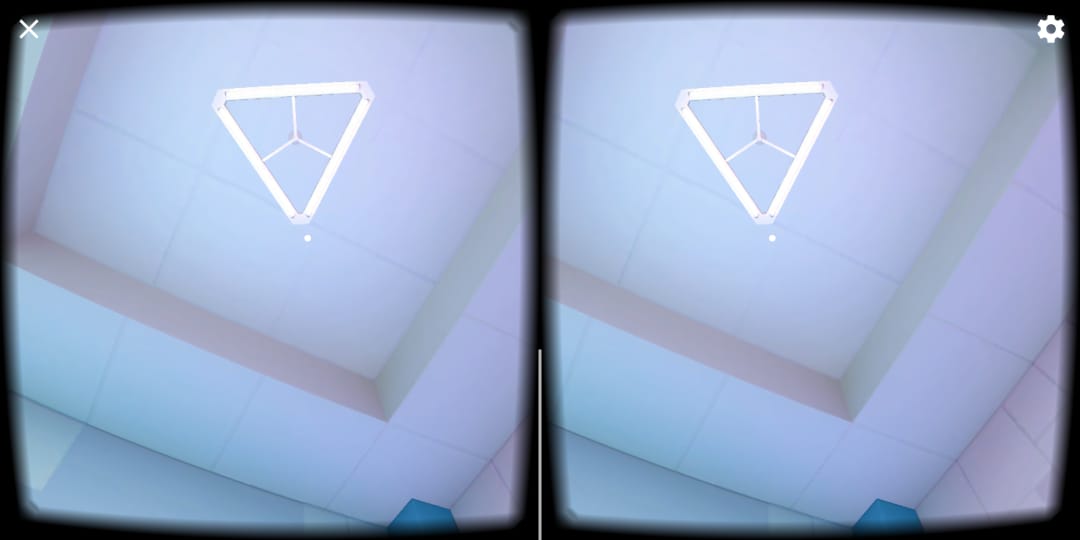
**Materials**:

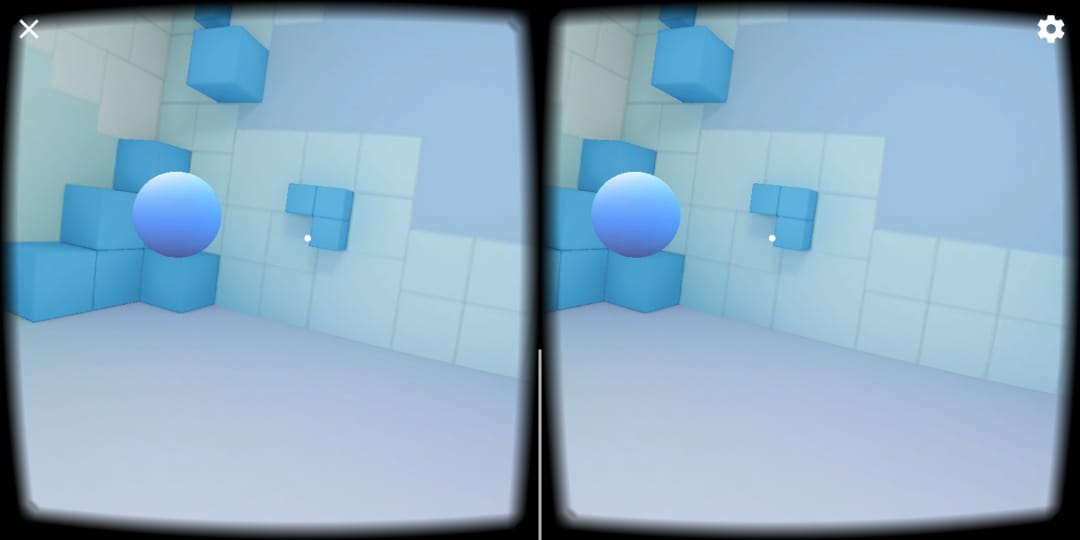
This module is created whenever unity project is created and it contains texture, objects, etc.…

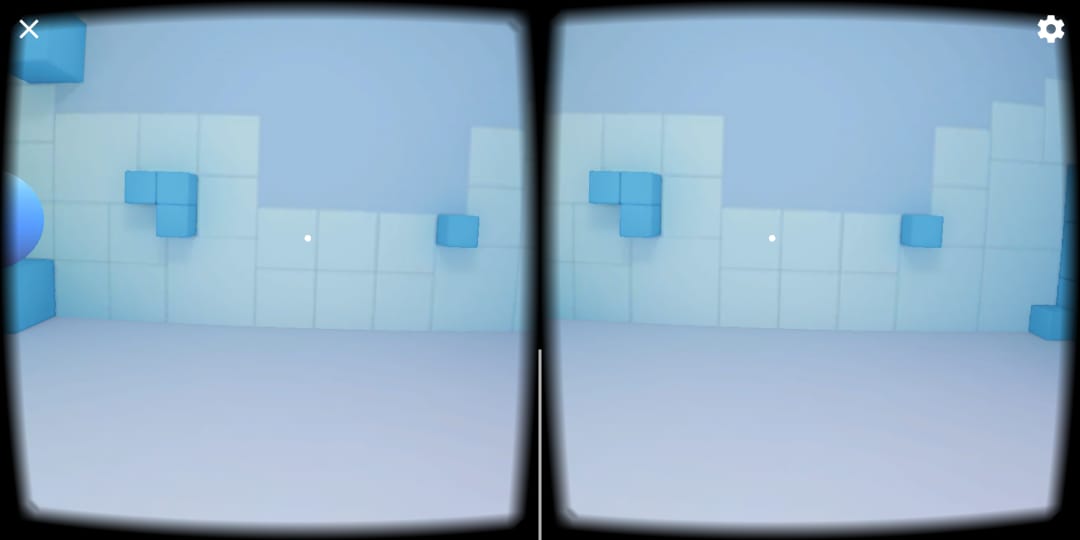
* Every other use case developed contains same hierarchy and the things that changes are functional implementation, material, prefabs, models.
* And the flow remains same for unity application lifecycle.

Proof of Working

***Shoot-It***

 9.A

 9.B

 9.C

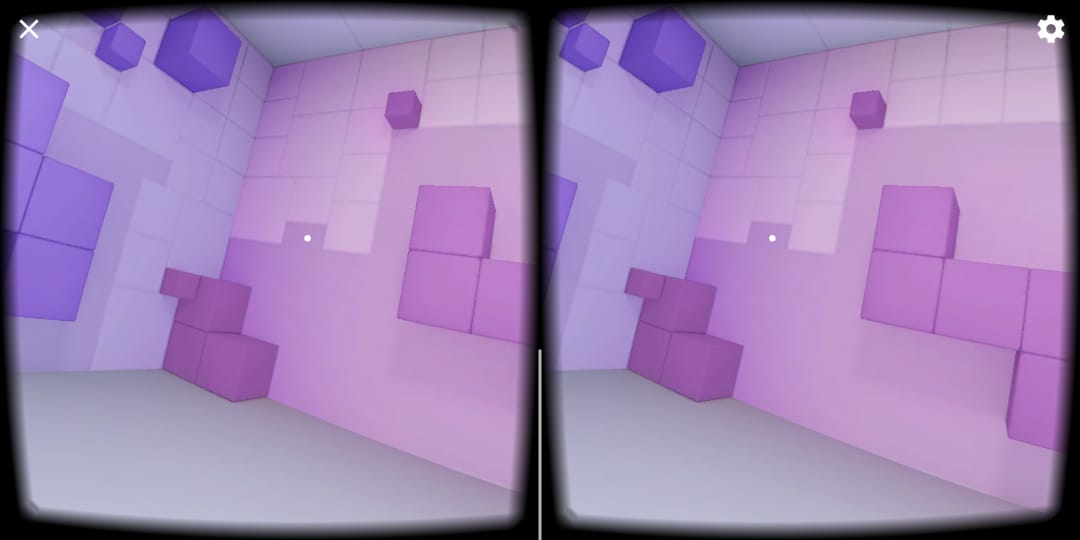
 9.D

Figure 9. Model of Shoot-It

Here in Figure-9 we can see that there is room and white ball is present. The white ball pop up randomly and user needs to shoot it and next ball appears and it continues till end.

***Treasure-Hunt***



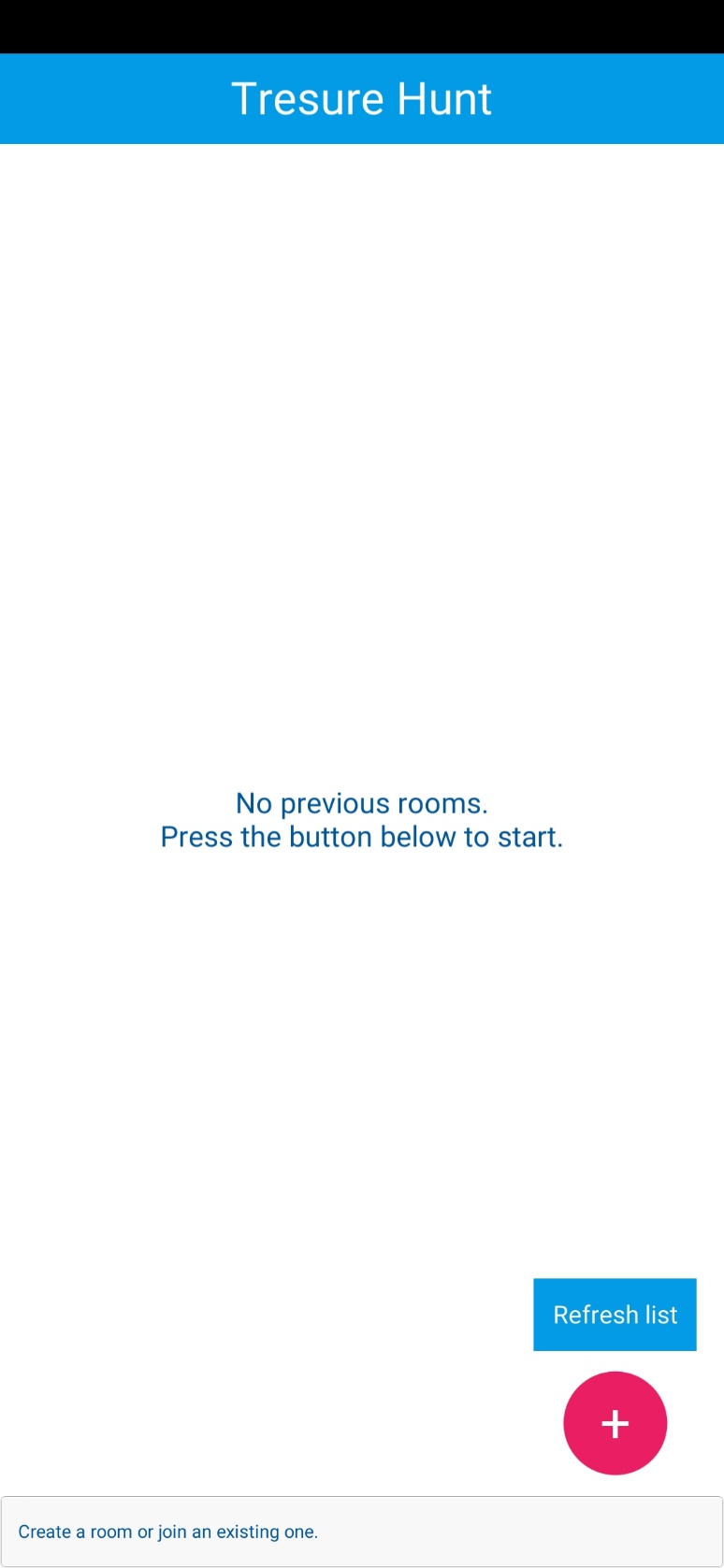
Figure 10. Creation of Anchor Point and Adding Treasure/Clue



***Figure 11. Adding Clue to Treasure***



***Figure 12. Updating Clue of Treasure***



***Figure 13. Creating Room/ Joining Room***



***Figure 14. Hunting Treasure (Client Side)***

**Working of Treasure Hunt!!**

Here in this game there are mainly two parties involved: Organizer/Host and other teams.

Here host sets up a treasure hunt trail i.e. they create clue which leads to other clue and it to other and so on. These continue until client reaches end.

**Setting Up Game**

1. Create Room
2. Create Anchor Point
3. Tap on screen to add treasure
4. Add clue in Treasure

**Playing Game**

1. Join Appropriate Room
2. Scan Anchor Point
3. Find Clues

* Clue only appears when user in certain distance of the clue (1 meter)

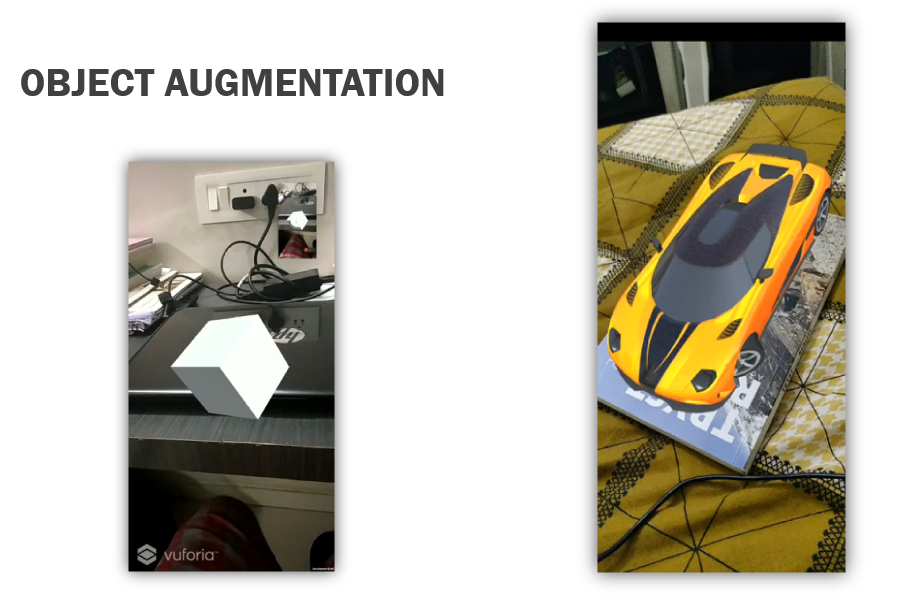


Figure 15. Object Augmentation

Here simply when screen starts the object (3D Model) pop up and user can see it.

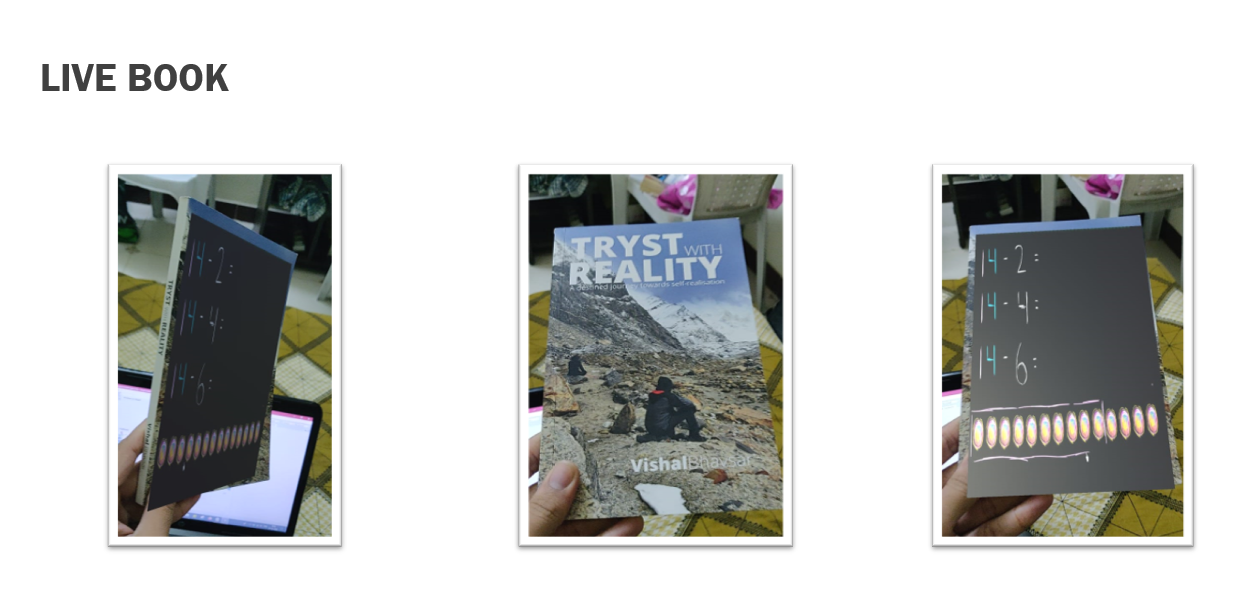


Figure 16. Live Book

Here whenever an appropriate image which is present in DB appears in front of camera after starting application then video pop up.

After tapping on video, it starts playing.

Video can be paused/resumed just tapping on video.

**FaceT**

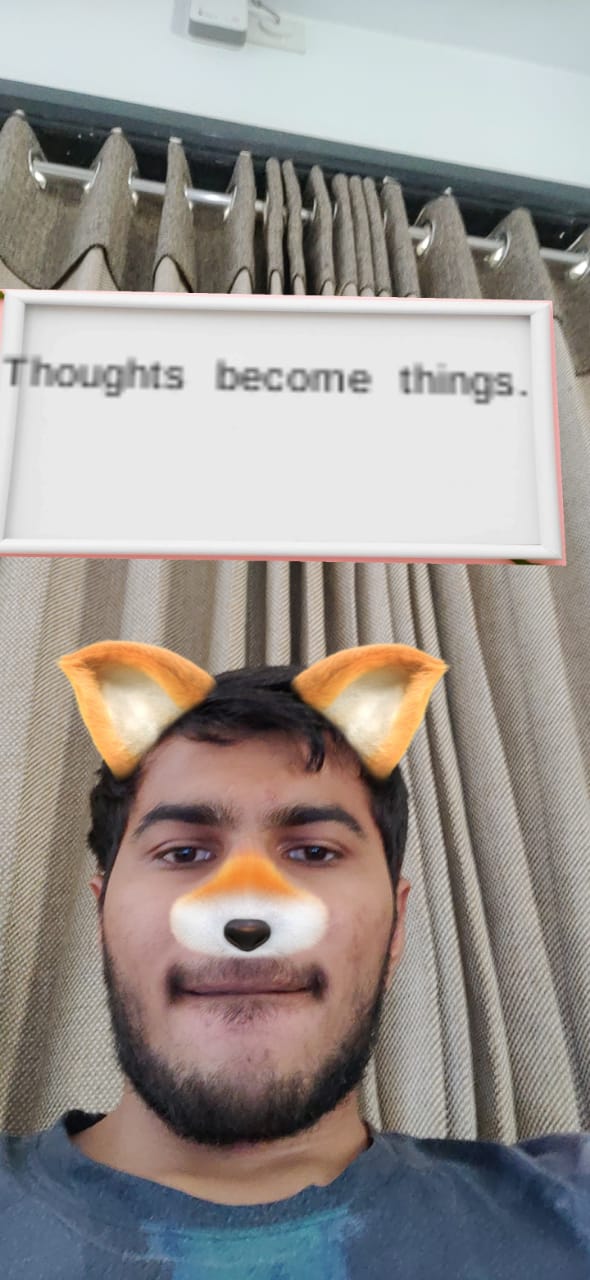


Figure 17. FaceT (Augmenting Thoughts)

Here after starting application whenever face is detected by front camera then their face is augmented with fox ears and random thoughts starts appearing and after a tap on it, it pauses.

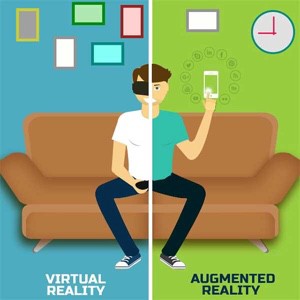
It’s mostly fun app to display random thoughts.

**Conclusion**

By developing the uses case of Augmented Reality, we came across various difficulties, technologies and many various solutions to develop a thing using more efficient ways. On the way of developing AR apps, we learned two new concepts VR and MR.

VR – Virtual Reality

MR – Mixed Reality



In VR there is virtual space which may corresponds to real world and on the basis of map/theme/scene user can virtualize and simulate real environment.

For example, there are some application which lets doctors simulate the behaviour of human body and they could perform a critical operation on simulation and it could life saviour and helps doctors prepare for original operation and ultimately minimize the risk.

Mixed Reality is combination of VR and AR.

These technologies will revolutionize the modern world in every aspect. From children to industries every single thing/being may become dependent on it. Nowadays children use AR to learn new things and also it makes easier for them to perceive things.



Figure 18. Children using AR to learn globe.

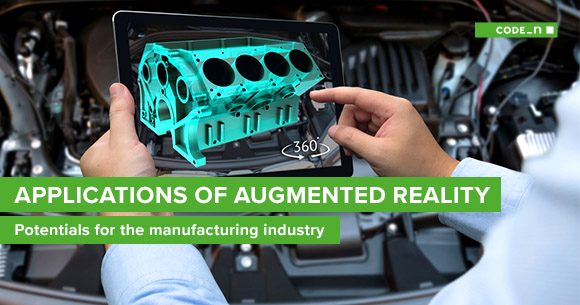


Figure 19. Use of AR in industry to visualize engine

Augmented Reality with artificial intelligence could be the next big step of the century because of its variety of application.

**Limitation and Future Extension**

We have developed use case and they are limitation itself because they individual but along with some real application it has great potential.

For example, if FaceT is developed as service than it could be used to target user on social media so that they can have fun

Apart from it in treasure hunt we have limited range of room and after a certain range the objects starts displacing.

And in Live Book we don’t feature where we could add new video for image dynamically without recompiling or re-analysing all the images, detection time for first time is unusually higher because of initialize of heavy database into main memory and also the video doesn’t remain steady.

All the above limitation could be Future Extension.

**References**

<https://developers.google.com/ar>

<https://library.vuforia.com/content/vuforia-library/en/reference/unity/index.html>

<https://docs.unity3d.com/Manual/index.html>

<https://www.coursera.org/search?query=ar&>