Bilkent University

Department of Computer Engineering

Senior Design Project

MoveIt, Indoor Manipulation : 3D Semantic Reconstruction, Display and Manipulation System

Analysis Report

Barış Can

Faruk Oruç

Mert Soydinç

Pınar Ayaz

Ünsal Öztürk

Supervisor: Associate Professor Selim Aksoy, Ph.D.

Jury Members: Assistant Professor Shervin Rahimzadeh Arashloo,Ph.D.

Assistant Professor Hamdi Dibeklioğlu, Ph.D.

Analysis Report

November 11, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

Analysis Report

MoveIt, Indoor Manipulation : 3D Semantic Reconstruction, Display and Manipulation System

1. Introduction

The popularity of smartphones rises daily. While they have many uses, it can be argued that taking and sharing photos are among the most widespread uses of today's incredibly powerful smartphones. Photos capture a moment in nature or an indoor scene, and they all have visual and spatial information of the objects contained within them. It is possible to extract this information, and more information can be extracted as one adds additional angles and positions from which the original scene is viewed. The amount of information one can extract is directly proportional to the extra number of angles and positions, and the quality of these images. This information then may be used to model the environment captured by these images in 3D.

MoveIt: Indoor Manipulation aims to bring our homes into virtual reality by recreating 3D indoor scenes using smartphone camera. With MoveIt, a quick scan of a room will bring the objects contained within this room into VR where they can be freely moved, interacted with, and manipulated. It is also a helpful tool that enables the user's aesthetic ability by allowing them to redecorate their room with ease while also providing functional help such as object boundary and collision detection. This 3D recreation of the room will be fully visualized in a VR environment that enables the user to freely move, scale, and rotate objects in the 3D reconstruction of the room using a VR Headset.

2. Current Systems

Below, similar apps to MoveIt will be discussed.

2.1. Hololens

- It is mixed reality technology that allows users to perform several actions such as playing games, changing the objects in the room, or get some holographic statistics [10].
- Users can interact with objects in the room via a VR headset.
- VR headset scans the room, performs semantic segmentation and users can grab and move objects, rotate them, scale them.
- Users can design hologram objects.

2.2. The Sims

- The Sims is a 3D life simulation game series that includes a mechanic to furnish houses as the user wishes. This furnishing mechanic is the part of this game that MoveIt takes inspiration from. [11]
- Players can build rooms and houses as they wish and furnish these rooms with the objects that the game provides.
- Players select the object from the objects menu which is divided by various categories, and place the object into the room. They can rotate the objects, change their colors and designs and scale them.
- Players can look at the rooms in any angle they prefer in third person perspective.

2.3. Roomplanner

- It is a mobile application that is similar to sims in a way that users can plan a virtually created room[12].
- Users can change the size of the room, add windows or doors and see those changes in 3D.

- Uses IKEA dataset (objects).
- Users can rotate, scale and change the color of the objects.
- Users can see the size of the objects from the air and a map of the room or house. This makes the application more realistic and useful for designers.

2.4. IKEA

- Users can make a scene for their room, select furniture from IKEA database and place them into the AR scene.
- Users can rotate or scale furniture that you placed.
- Users can search for furniture from categories and check their prices from IKEA.
- There are several restrictions such as the room should be well lit, or the floor should be free of clutter.
- You can share your place with social apps like Whatsapp or Instagram [9].

3. Proposed System

The following parts are the details of the proposed system.

3.1. Overview

MoveIt: Indoor Manipulation aims to develop an application which will semantically reconstruct the geometry of indoor areas, such as a living room or a kitchen, in 3D by using videos, live scans/feed obtained from the camera of a mobile device. The users will need to create an account either by our sign up screen or their Google account. These accounts will contain users' email addresses, their objects and their scenes. After they scan, the 3D reconstruction process will produce a 3D scene and will label the reconstructed objects and geometry semantically by attaching a semantic label to each object. It will also allow the users of the application to manipulate the scene by allowing the users to move, rotate, scale, and deform the geometry of the objects. Texture obtained from the live feed will also be mapped to the corresponding meshes. The application will store the 3D reconstruction information along with the texture, label, and mesh information of a given object on a database for later data analytics and machine learning purposes, and will be open to the public as a labelled set of 3D objects, which may be suitable for supervised learning purposes. However, these geometric meshes will be added to the database if and only if the user gives permission to do so, and these meshes will be stored anonymously. If they decide to upload their furniture' meshes, they can also tag them as they wish.

The main purpose of the application is to allow the users to manipulate an indoor scene without actually having to add, move, or alter the objects in the scene. For instance, if the user wishes to furnish a room, the user is able to use this application to reconstruct the geometry of the room and the current objects within the room as a 3D scene in VR. The application will allow the necessary facilities to manipulate the positions, rotations, scales, and the geometry of the scene, while also allowing the user to place other objects in this scene. These new objects can be fetched from the database that the application keeps or may be supplied by the user in the form of a 3D mesh and texture. Objects can be queried and retrieved from the database using semantic labels or uploaders’ specified tags. As an example, the user will be able to request a 3D mesh representation of a chair, and if available, the application will present a set of chairs that the user can add to the scene present in the database. A user will not be able to upload objects directly to the database, and user-provided objects will only be stored locally. Another relevant feature is to save a scene with a given set of objects, geometry, and the texture locally on a device for later access and modification.

Another purpose, though not the main purpose of this application is to create a publicly available dataset of labelled 3D objects and textures corresponding to these objects. This dataset might be used to study various aspects of 3D reconstruction, texture analysis, and the relationship between these and the semantic labels provided alongside the objects. The dataset may also be used for supervised learning purposes, and the more the app is used, the larger the dataset will be.

As for the platforms and the hardware on which the application will be deployed, MoveIt: Indoor Manipulation will rely on many other frameworks and paradigms. Retrieval of live scans/feed will depend on Android infrastructure; the texture analysis, texture to geometry mapping, 3D reconstruction, semantic segmentation of the live scan and semantic labelling of meshes will be carried out on a remote server with sufficient hardware. The models used for the semantic segmentation of images will be based on pre-trained convolutional neural networks, and for the 3D reconstruction and texture mapping step, popular frameworks such as OpenCV will be used [8]. Once a model of the scene is created, the user may choose to have this scene rendered on a VR headset, or on a personal computer. The user will be able to control and manipulate the scene using VR controllers, or keyboard and mouse.

3.2. Functional Requirements

This system has 4 main components, these are mobile phone components, computer vision components, cloud computing and VR Headset together with a personal computer. For smartphone components, we will employ a set of Android APIs for the gathering of the required data and sending this data to be processed on a remote, monolithic server. This server component will utilize Microsoft Azure cloud computing services for the semantic segmentation of the image, the reconstruction of objects contained within these images into 3D objects, and the storage of the reconstruction of these objects in a database. Computer vision components will carry out semantic segmentation by employing either GoogLenet, ResNet or YOLO libraries that are already trained for this purpose. VR headset will be utilized for the 3D display of the 3D renderings of the reconstruction of the indoor scene through Unity VR Game Engine.

Mobile Phone Component

For mobile phones, the system's required functionalities include:

- Scan the contents of the indoor area by the use of its camera.
- Pass this information along to the cloud server.
- Upload scanned objects to the database with their tags.
- Add new objects from the database to user account.

Cloud Computing

The cloud server will provide the following functionalities:

- Provide enough computational power to computer vision algorithms.
- Provide the necessary storage for the already scanned household items.

Computer Vision

The system will use pre-trained neural networks that are specialized on semantic segmentation for labeling the reconstructed 3D environment of the room, which will follow these steps:

- Detect surfaces and objects in the indoor setting.

- Create titles and bounding boxes for the objects.
- Clearly separate the boundaries of the room from the boundaries of the interior objects using semantic labels.
- Use certain assumptions in order to reconstruct the obstructed angles and views of the room and the objects.
- Create a detailed 3D map of the indoor environment with distinct separate objects.

VR Headset with a personal computer

In the final step, this component will provide these functionalities:

- Fetch the required room data from the cloud server.
- Recreate the room in 3D Virtual Reality with fully interactable objects.
- Place the user in the created room.
- Allow the user to create additional copies of the existing furniture or simply copy - paste some parts of them to elsewhere.

3.3. Non-functional Requirements

In the following subsections, usability, reliability, security, smartphone friendliness and availability will be discussed.

Usability

- The MoveIt application should have a user-friendly interface that would enable different users from various age and knowledge groups to use the application with ease, and it should not take more than 30 minutes for a user to get accustomed to the interface.
- The application should include a user manual, an instructions page and YouTube videos explaining the users the key concepts of the application,

such as how to record a room and how to move objects around. These tutorials should not take more than two minutes for a user to complete.

- Users should be able to move around objects with natural controls that are easy to perform and comes as natural, and a user should not spend more than five seconds on average to perform a given operation on an object.

Reliability

- 3D reconstruction of the objects in a room must be as detailed as possible and the application should be able to recreate obscure areas of the objects with high accuracy by making correct assumptions. The gaps in the reconstructed textures should not be more than 10% of the visible surfaces of the scene, and the artifacts due to the reconstruction should not yield an error more than 10% in terms of volume of the individual object.
- Different components of the application should work together harmoniously. To elaborate, MoveIt will have a smartphone application as well as a background side where 3D reconstruction will take place. Since the user interacts with the smartphone application, the background computation times should be optimized as much as possible to guarantee responsiveness. The computation for the reconstruction should not take more than five minutes.
- When the reconstructed objects are manipulated by the user (e.g. moved around), the previous place of the object as well as the new one should be altered and displayed correctly by the application. The only tolerable error should be the floating point representation rounding-off errors.

Security

- The application should be able to protect the data that the user uploads to the system such as recordings or just their email addresses since these data could be classified as personal. A secure hashing scheme must be used, and this hashing scheme should be considered as "having minor weaknesses" in the worst case [18].

Smartphone Friendliness

- The users are expected to use smartphones as an interface to interact with the MoveIt application. Therefore, the application should not be very taxing on smartphones and the power usage as well as the mobile data usage should be optimized to ensure user satisfaction. The user should be able to use the application for at least two hours before the battery of the smartphone runs out. The reconstructed scene should run at, at least, 24 frames per seconds.

Scalability

- Since the computations will be done on a remote server, it might require too much computational power if several users send their videos at the same time. In order to solve this, either computational power of the server should be increased by hiring a server with double the number of cores and double the amount of memory. Once scaling up costs exceed 16 times the initial server rent, distributed versions of the algorithms to be developed should be introduced, and multiple servers should be used to carry out the computations.

Availability

- The system should be available for the users and function at all times. But since server failures are inevitable, we are aiming for an SLA level of 99.9% (Yearly 8h46m of downtime)[7].

3.4. Pseudo requirements

- Smartphones are not powerful enough to do the processing of the scan by themselves so Microsoft Azure's cloud computing service will be used [1].
- The object oriented programming paradigm will be followed during the development of the application.
- Semantic segmentation will be done on Python in order to label the objects correctly.
- The visual components of the application such as the movement of the objects will be done on Unity Game Engine with C#. [2]
- The mobile application will be developed on Android Studio with Java. [3]
- C++ will be utilized for the implementation of the 3D reconstruction as it is the fastest one applicable for this.
- The programming languages must be able to interact with each other.
- We will use powerful pre-trained neural networks, such as GoogLeNet, ResNet or YOLO, for the semantic segmentation of the objects. [4, 5, 6]
- A database for the object meshes and their labels will be hosted on Microsoft Azure. [1]
- MoveIt should run in Android phones (7.0 or above), Windows 8/10 and a VR device.
- The webpage of the project which contains all necessary information is https://barisc22.github.io/MoveIt/.

3.5. System models

The following are the scenarios,use - case model, object - class model, dynamic models and the user interfaces.

3.5.1. Scenarios

Use case Name:	**Sign Up**
Participating actors:	User
Stakeholders/Interests:	User decides to sign up to MoveIt application.
Flow of events:	1. User opens the application. 2. User clicks on the ‘Signup’ button. 3. Registration page opens. 4. User enters their email address and sets a password. 5. User clicks ‘Signup’ button. 6. System creates an account for the user.
Entry conditions:	User opens the application.
Exit conditions:	User is navigated to their homepage.

Use case Name:	**Login**
Participating actors:	User
Stakeholders/Interests:	User decides to sign in to MoveIt application.
Flow of events:	1. User opens the application. 2. User enters their email address and password. - User could sign in with their Google accounts by clicking “Sign in with Google” button. 3. User clicks 'Sign In’ button. 4. System loads the account of the user.
Entry conditions:	User opens the application.
Exit conditions:	User is navigated to their homepage.

Use case Name:	**Capture Video**
Participating actors:	User
Stakeholders/Interests:	User decides to capture a video to be reconstructed.
Flow of events:	1. User clicks on ‘Take a New Video’ button. 2. The application enters into video capturing mode and the camera opens. 3. User clicks on ‘Start Capturing’ button. 4. User starts to move the mobile phone around the room to capture an indoor video. 5. User clicks on ‘Stop Capturing’ button to end capturing a video. - If user accepts the capture, the system uploads videos to the server to reconstruct. - If user cancels the capture, it is deleted from the application.
Entry conditions:	User is logged in to their account and is in their homepage.
Exit conditions:	The captured scene is either deleted or uploaded to the server to be reconstructed.

Use case Name:	**Capture Furniture**
Participating actors:	User
Stakeholders/Interests:	User decides to capture a furniture to be reconstructed and added into their objects database.
Flow of events:	1. User clicks on ‘New Object’ button from ‘My Objects’ page. 2. The application enters into object capturing mode and the camera opens. 3. User clicks on ‘Start Capturing’ button. 4. User starts to move the mobile phone around the object to capture a furniture.

	5. User clicks on 'Stop Capturing' button to end capturing a scene. - If user accepts the capture, the system uploads furniture to the server to reconstruct. Then, the system saves the reconstructed furniture into the objects database of the current user. - If user gives consent to upload the furniture to the publicly available database, they can add a tag to it, otherwise, it will not appear in the Workshop. - If user cancels the capture, it is deleted from the application.
Entry conditions:	User is logged in to their account and is in their 'My Objects' page.
Exit conditions:	The captured furniture is either deleted or uploaded to the server to be reconstructed and then saved into the object database of the user.

Use case Name:	**View Videos/Scenes**
Participating actors:	User
Stakeholders/Interests:	User decides to view one of their previously captured and reconstructed scenes.
Flow of events:	1. User enters into their homepage. 2. System displays User's previously captured videos and reconstructed scenes. 3. User clicks on one of the videos or scenes. 4. User views the selected video or scene on view mode. 5. User clicks the 'Return' button to go back to the scene selection.
Entry conditions:	User is logged in to their account and is in their homepage.
Exit conditions:	User is shown a previously captured video/scene in view mode.

Use case Name:	View Objects
Participating actors:	User
Stakeholders/Interests:	User decides to view a furniture from their saved furniture.
Flow of events:	1. User enters into 'My Objects' tab. 2. System displays User's previously captured and reconstructed objects. 3. User clicks on one of the objects. 4. User views the selected object on furniture view mode. 5. User clicks the 'Return' button to go back to the furniture selection.
Entry conditions:	User is logged in to their account and is in their homepage. User is using the PC application.
Exit conditions:	User is shown a previously captured object in furniture view mode.

Use case Name:	Download From Database
Participating actors:	User
Stakeholders/Interests:	User decides to add a new furniture to their saved furniture from the application database.
Flow of events:	1. User enters into 'Workshop' tab. 2. System displays various categories of furniture. 3. User clicks on one of the categories or searches for the user specified tag. 4. System displays the furniture on the database with the selected category. 5. User browses through the furniture, selects one and clicks on it. 6. Selected furniture is displayed in detail.

	- If user clicks on 'Confirm' button, the furniture is added into user's objects database and becomes available to place into reconstructed scenes. - If user clicks on 'Back' button on the PC application, the furniture is not added to the user's database and user goes back to the furniture display page.
Entry conditions:	User is logged in to their account and is in their 'My Objects' page.
Exit conditions:	User adds a furniture to their object database.

Use case Name:	**Place New Objects**
Participating actors:	User
Stakeholders/Interests:	User decides to place a new object to a scene.
The flow of events:	1. User opens the application. 2. User opens a scene from the homepage. 3. User clicks on an object from the bottom bar which contains all of their objects and drags and drops the selected object into the scene. 4. User then might perform additional operations on the object. 5. User clicks to 'Confirm Placement' button on the VR controller/keyboard. 6. User confirms the change and saves the scene or the object.
Entry conditions:	User opens the application. User has to have a saved scene.
Exit-conditions:	User is navigated to their personal homepage. User might not confirm the change.

Use case Name: **Scale Object**

Participating actors:	User
Stakeholders/Interests:	User decides to change the size of an object.
The flow of events:	1. User opens a saved scene or captures a new scene. - If the user captures a new scene, he/she waits for the semantic segmentation. 2. User clicks on a detected object. 3. User might change the scale of the object by pulling from its edges. 4. User confirms the change and saves the object or the scene.
Entry conditions:	User opens the application. User has to have a saved scene or capture a new scene. The object has to be detected.
Exit-conditions:	User is navigated to their personal homepage. User might not confirm the change.

Use case Name: **Replicate Part**

Participating actors:	User
Stakeholders/Interests:	User decides to replicate a part of an object.

The flow of events:	1. User opens a saved scene or captures a new scene. - If the user captures a new scene, he/she waits for the semantic segmentation. 2. User clicks on a detected object. 3. User selects a part of the object by changing its size with a bounding box. 4. After selecting the part, he/she might cut the part to paste it somewhere else or copy the part to keep the existing one and replicate it somewhere else. 5. User clicks to 'Confirm' button from the VR controller/keyboard. 6. User confirms the change and saves the scene or the object.
Entry conditions:	User opens the application. User has to have a saved scene or capture a new scene. The object has to be detected.
Exit-conditions:	User is navigated to their personal homepage. User might not confirm the change.

Use case Name:	**Rotate Object**
Participating actors:	User
Stakeholders/Interests:	User decides to change the orientation of an object.
The flow of events:	1. User opens a reconstructed scene. 2. User clicks on a detected object. 3. User changes the orientation of the object by pressing and holding down the left or right rotation keys. 4. User confirms the change by pressing the 'Confirm' button and the new orientation of the object is saved on the scene.

	- If user presses the 'Cancel' button, the change is not saved and the previous orientation of the object is restored.
Entry conditions:	User opens the application. User has to have a saved scene. The object has to be detected.
Exit-conditions:	User is navigated to their personal homepage. User might not confirm the change.

Use case Name:	**Move Object**
Participating actors:	User
Stakeholders/Interests:	User decides to change the location of an object.
The flow of events:	1. User opens a reconstructed scene. 2. User clicks on a detected object. 3. User changes the location of the object by pressing and holding down the move key and simultaneously dragging the object to the desired location. 4. User confirms the move by pressing the 'Confirm' button and the new location of the object is saved on the scene. - If user presses the 'Cancel' button, the change is not saved and the object goes back to its old place in the scene.
Entry conditions:	User opens the application. User has to have a saved scene. The object has to be detected.
Exit-conditions:	User is navigated to their personal homepage. User might not confirm the change.

Use case Name:	**Replicate Object**
Participating actors:	User
Stakeholders/Interests:	User decides to replicate an object in the scene.
The flow of events:	1. User opens a reconstructed scene. 2. User clicks on a detected object. 3. User duplicates the object by pressing the duplicate key. 4. System creates a copy of the object and places it somewhere appropriate in the scene close to the original object. 5. User confirms the duplication by pressing the 'Confirm' button and the new object is saved on the scene. - If user presses the 'Cancel' button, the change is not saved and the copy object is deleted and removed from the scene.
Entry conditions:	User opens the application. User has to have a saved scene. The object has to be detected.
Exit-conditions:	User is navigated to their personal homepage. User might not confirm the change.

Use case Name:	**Change Bounding Box**
Participating actors:	User
Stakeholders/Interests:	User decides to change the bounding box of an object assigned at reconstruction.

The flow of events:	1. User opens a reconstructed scene. 2. User clicks on a detected object. 3. User changes the boundaries of the bounding box of the object by dragging the boundary. 4. User confirms the change by pressing the 'Confirm' button and the new bounding box of the object is saved on the scene. - If user presses the 'Cancel' button, the change is not saved and the previous bounding box of the object is restored.
Entry conditions:	User opens the application. User has to have a saved scene. The object has to be detected.
Exit-conditions:	User is navigated to their personal homepage. User might not confirm the change.

Use case Name:	**Reconstruct Scene**
Participating actors:	User Server
Stakeholders/Interests:	User decides to reconstruct a saved video.
The flow of events:	1. User opens a saved video from 'Videos and Scenes' page. 2. User clicks on the 'Convert to Scene' button. 3. Server starts reconstruction. 4. Video is deleted and the scenes is added to the 'My Scenes' page.
Entry conditions:	User opens the application. User has to have a saved video.
Exit-conditions:	Selected video is reconstructed as a scene.

Use case Name:	**Process and Upload**
Participating actors:	User Server
Stakeholders/Interests:	User decides to scan his/her room and reconstruct it.
The flow of events:	1. User opens 'Take a New Video' page from main menu. 2. User clicks on the 'Scan' button. 3. After the scan complete, server starts semantic segmentation 4. After the semantic segmentation, server starts 3D reconstruction. 5. After the 3D reconstruction, server uploads the created scene and the objects to the database that only the user can see. - After users' consent, objects could be shared in the publicly available dataset. 6. User can see their scenes from 'Videos and Scenes' page.
Entry conditions:	User opens the application.
Exit-conditions:	The room is reconstructed as a scene and uploaded to database.

3.5.2. Use case model

Below you can see the use-case model of our system.

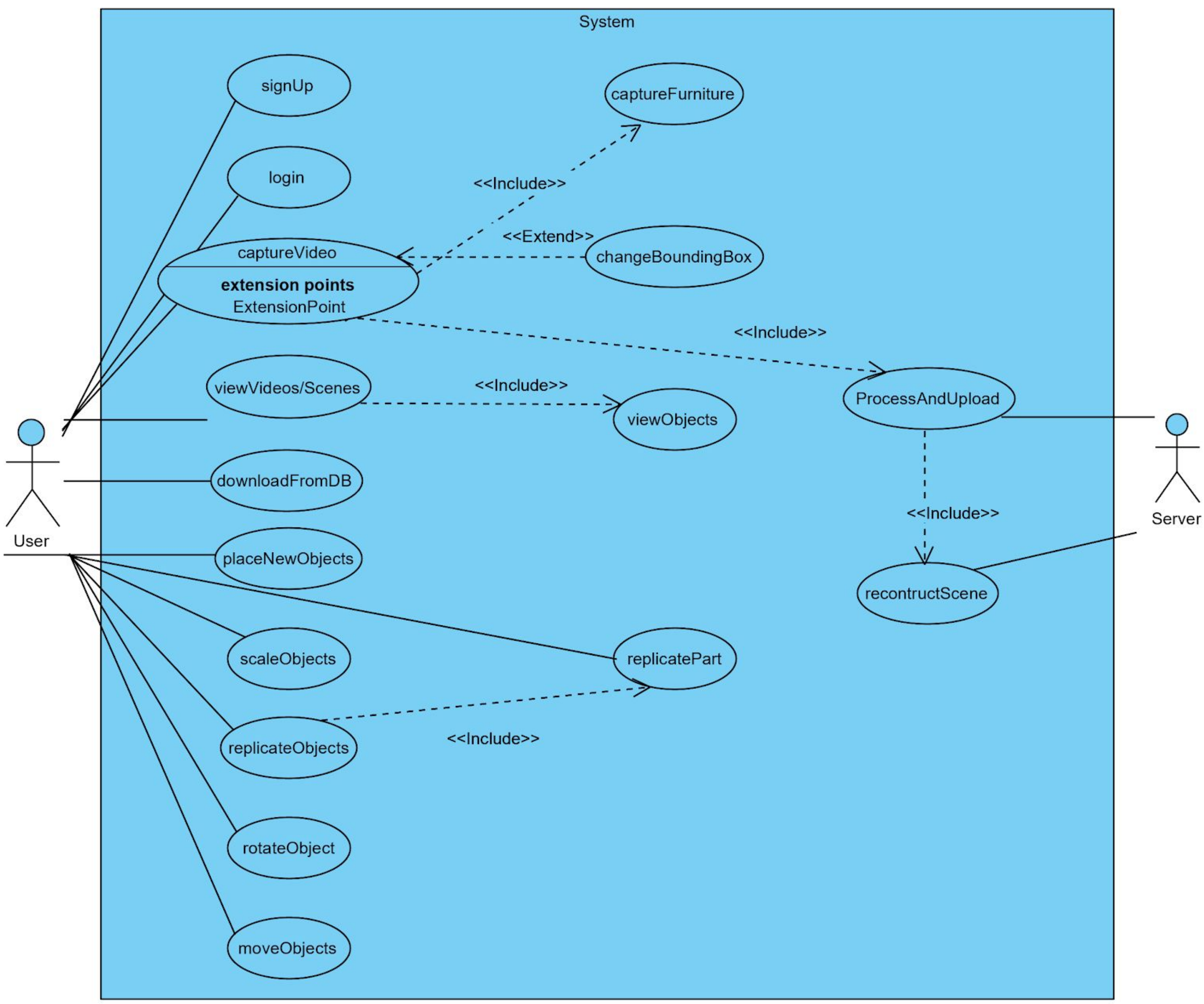

Figure 1: Use-case Diagram

3.5.3. Object and class model

The following diagram is available at a higher resolution at https://imgur.com/a/J4idpsA.

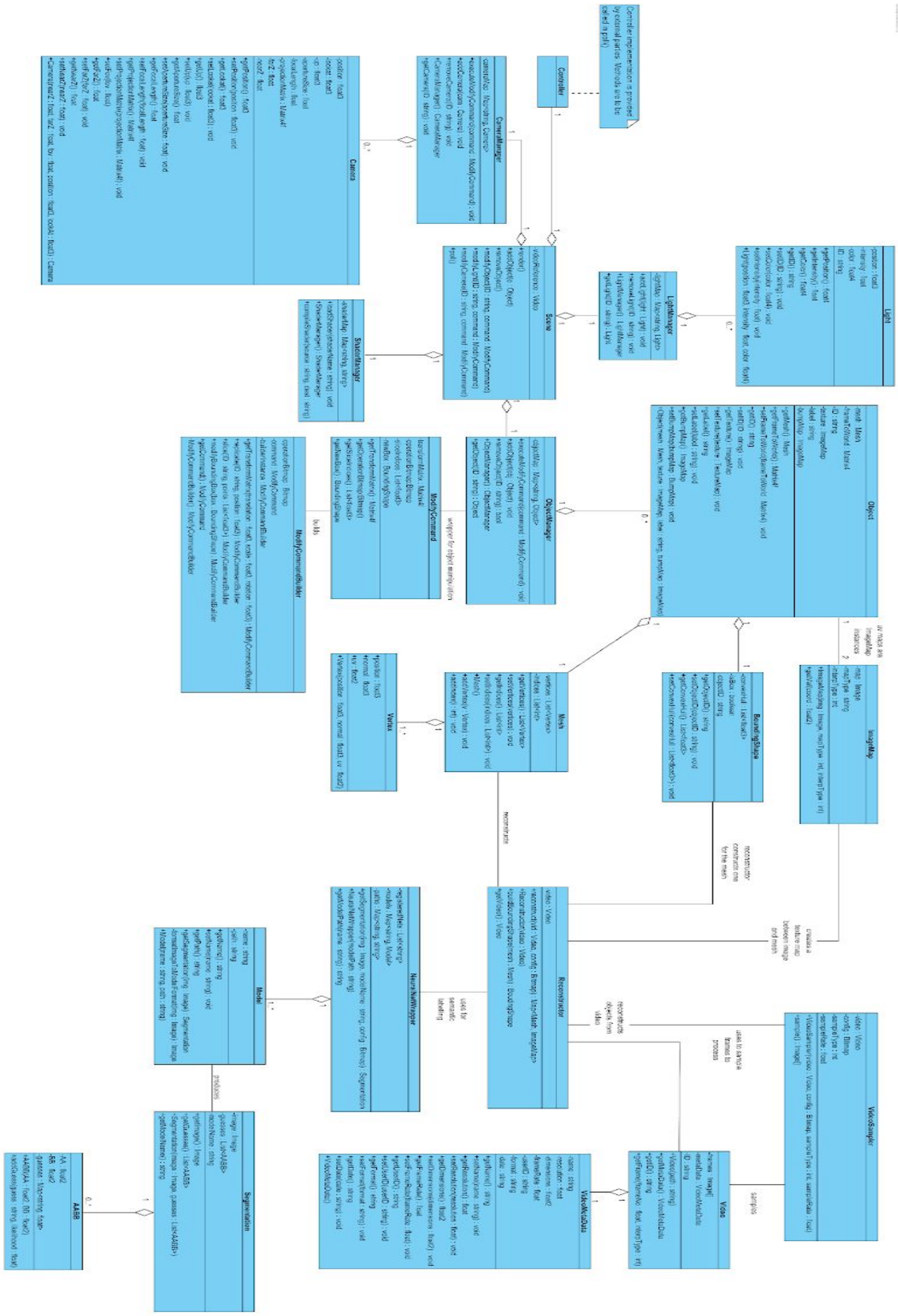

Figure 2: Class Diagram

3.5.4. Dynamic models

3.5.4.1. Sequence Diagrams

The following diagrams are available at a higher resolution at https://imgur.com/a/J4idpsA.

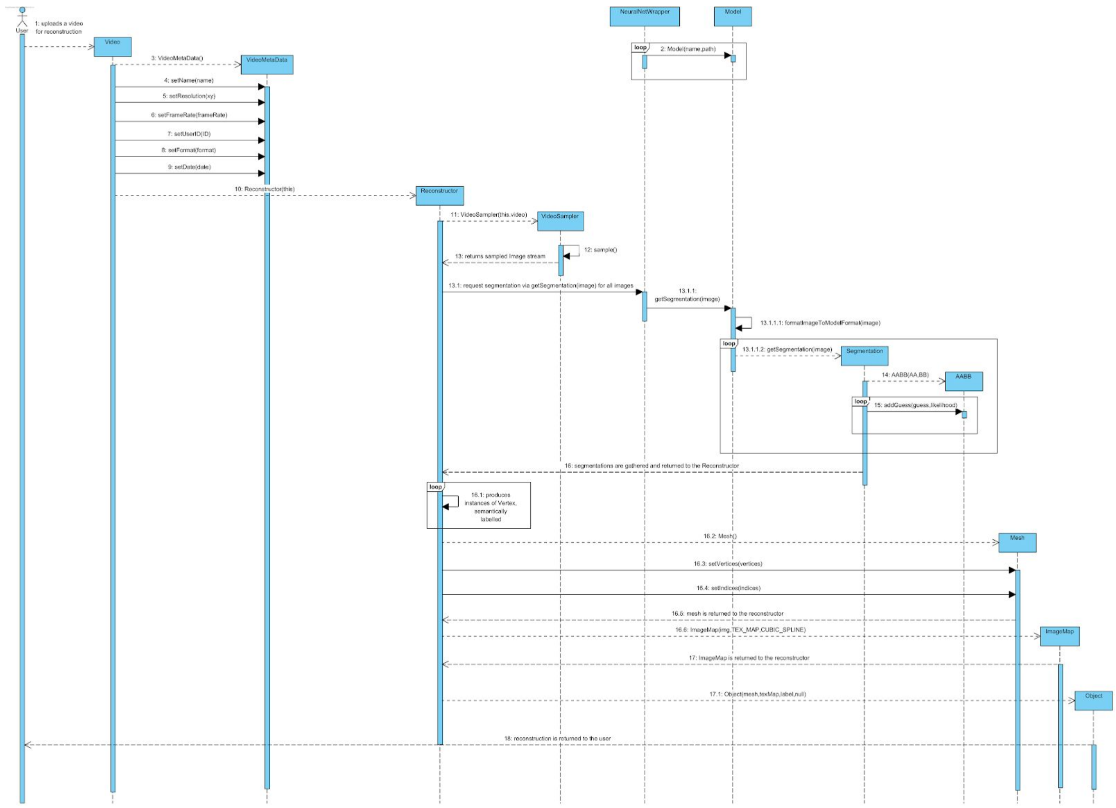

Figure 3: Sequence Diagram for the 3D Reconstruction Process

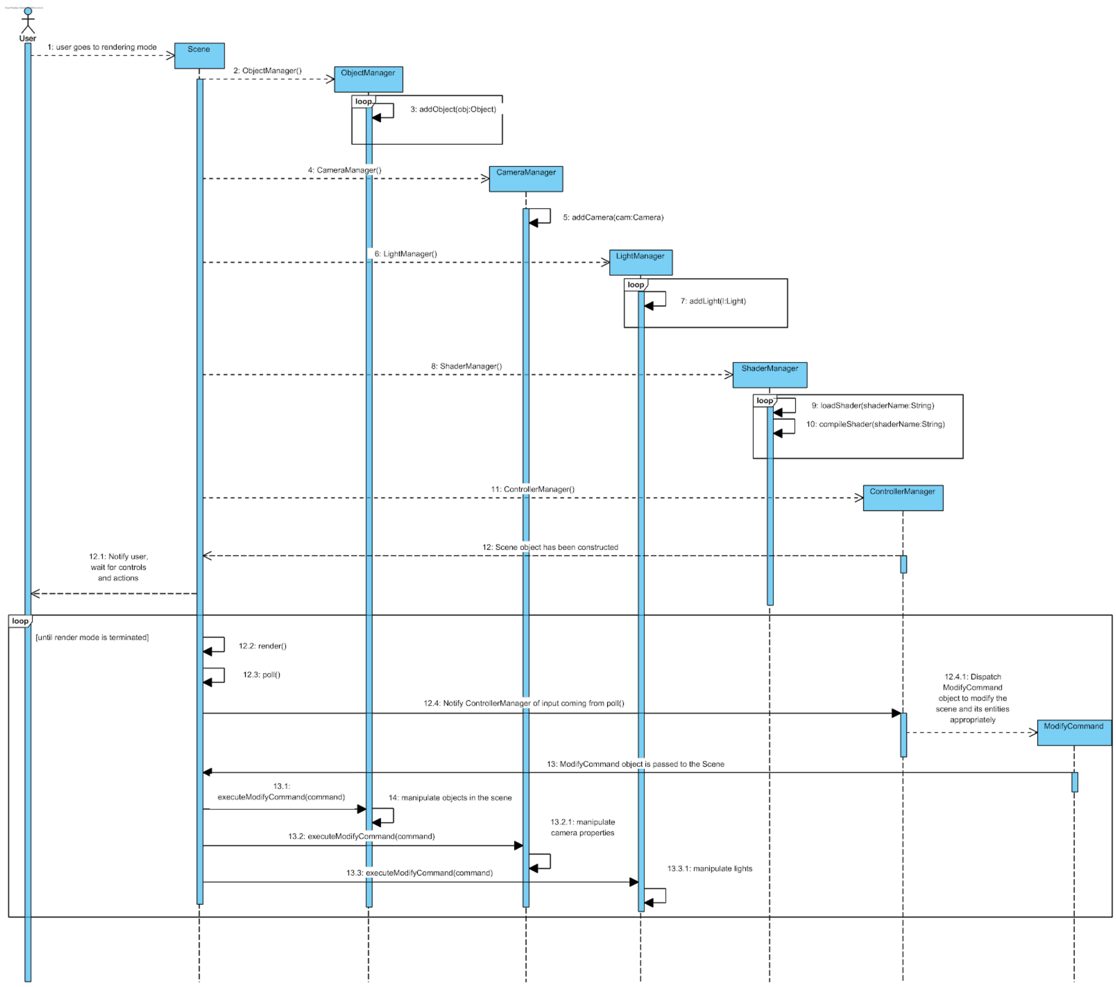

Figure 4: Sequence Diagram for Render Loop

3.5.4.2. Activity Diagrams

3.5.4.2.1. Lifecycle of the PC/VR Application

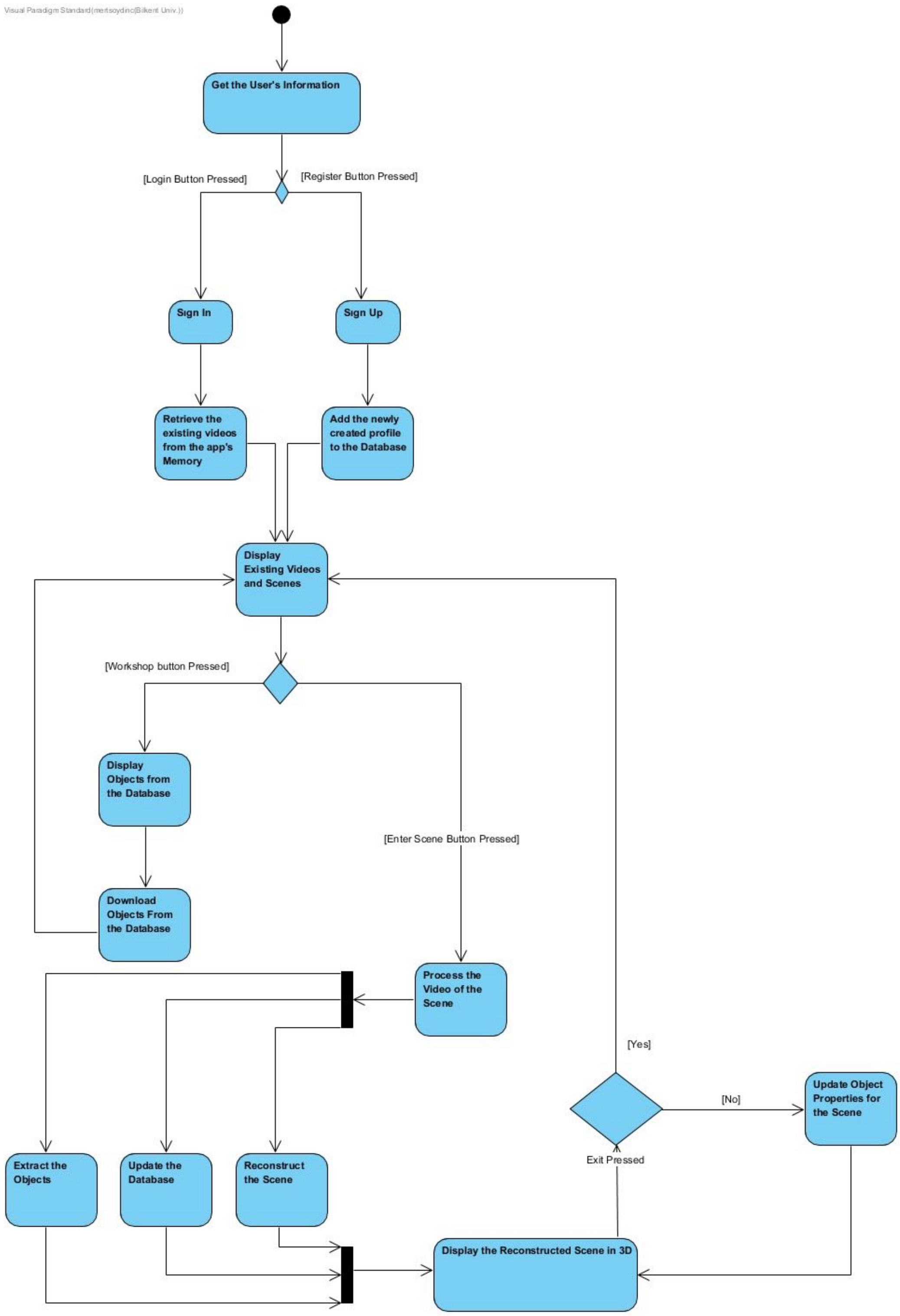

Figure 5: Lifecycle of the PC/VR Application

3.5.4.2.2. Lifecycle of the Mobile Application

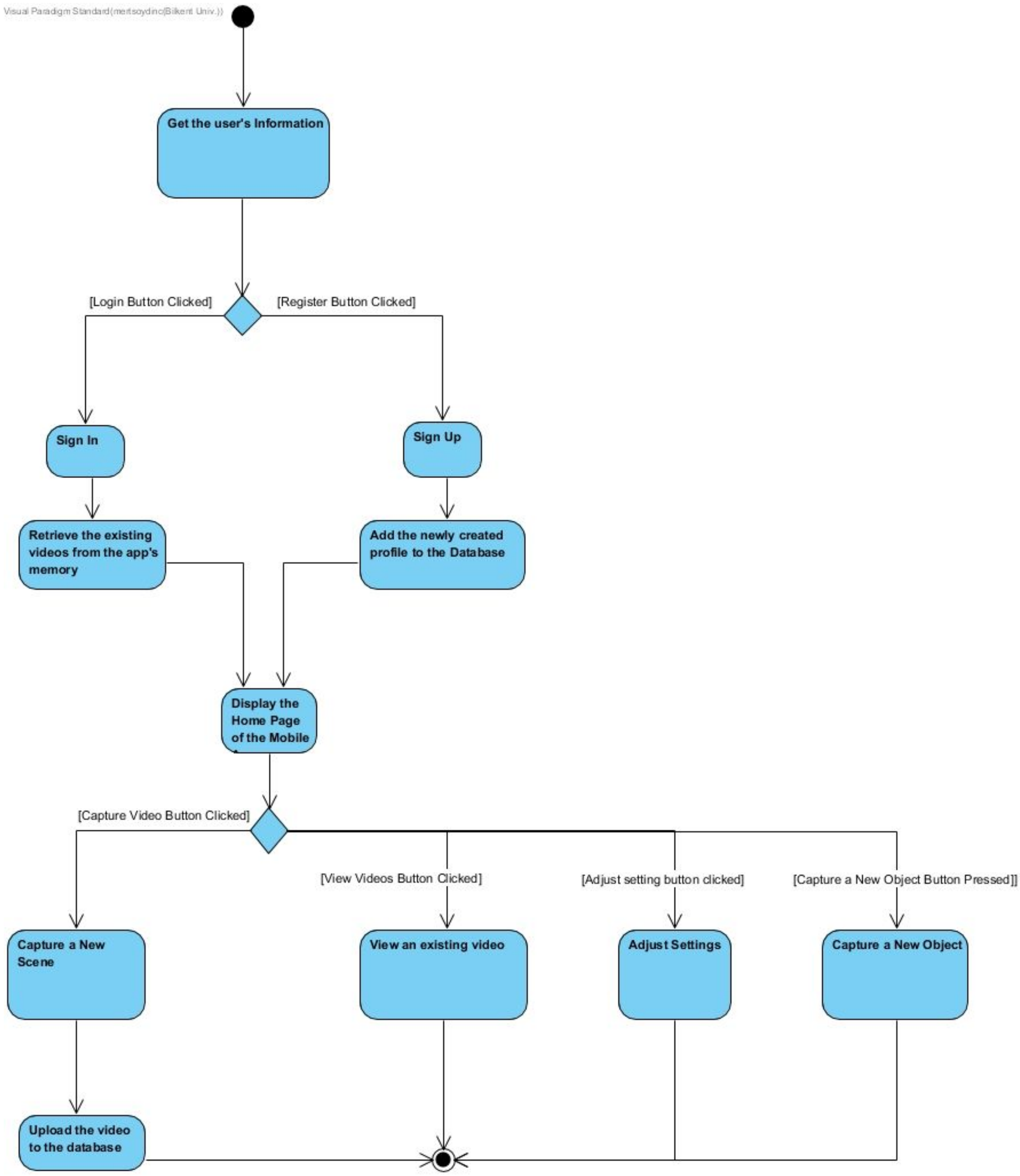

Figure 6: Lifecycle of the Mobile Application

3.5.4.3. State Diagram - Manipulating Objects

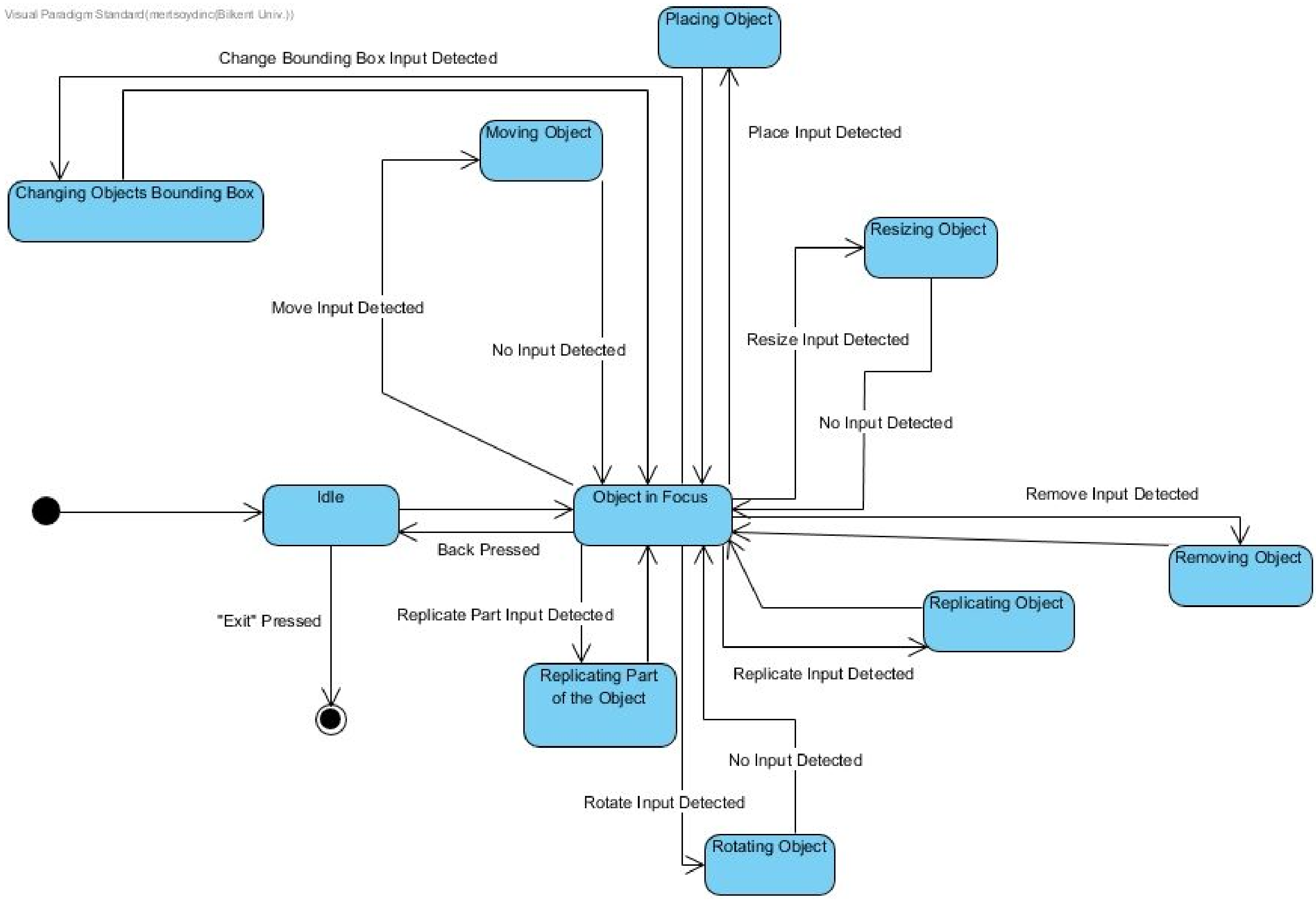

Figure 7: Lifecycle of the Mobile Application

3.5.5. User interface - navigational paths and screen mock-ups

3.5.5.1. Login

Figure 8: Login Screen

First screen encountered by the user when the mobile app is launched. User can proceed to login if they already have an account or signup to create one.

3.5.5.2. Signup

Figure 9: Signup Screen

If the user clicks the signup button on the login screen, this page appears. New user can use this page to create an account using their emails and chosen passwords.

3.5.5.3. Captured Stuff

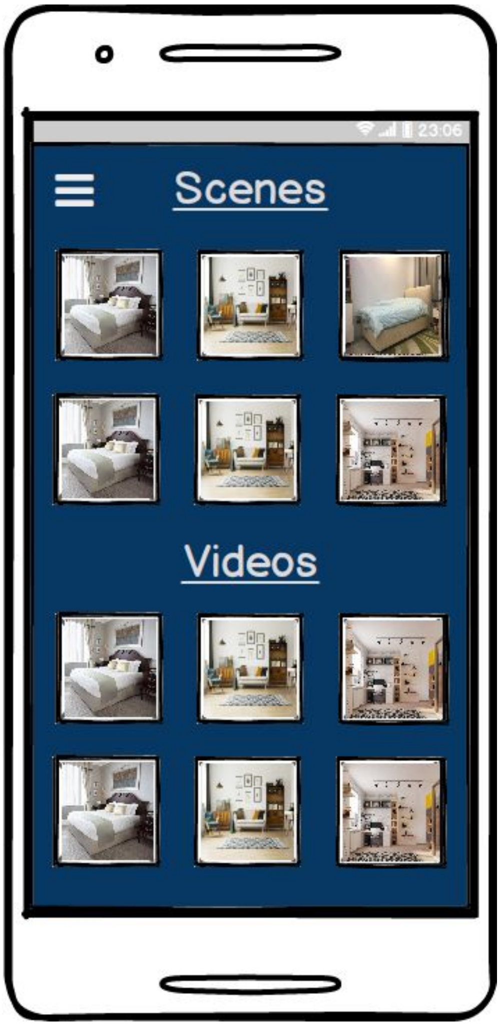

Figure 10: Captured Scenes Screen

Main menu of the application is accessed right after the login/signup screens. Here, previously captured videos and their corresponding scenes can be seen.

3.5.5.4. Side Menu

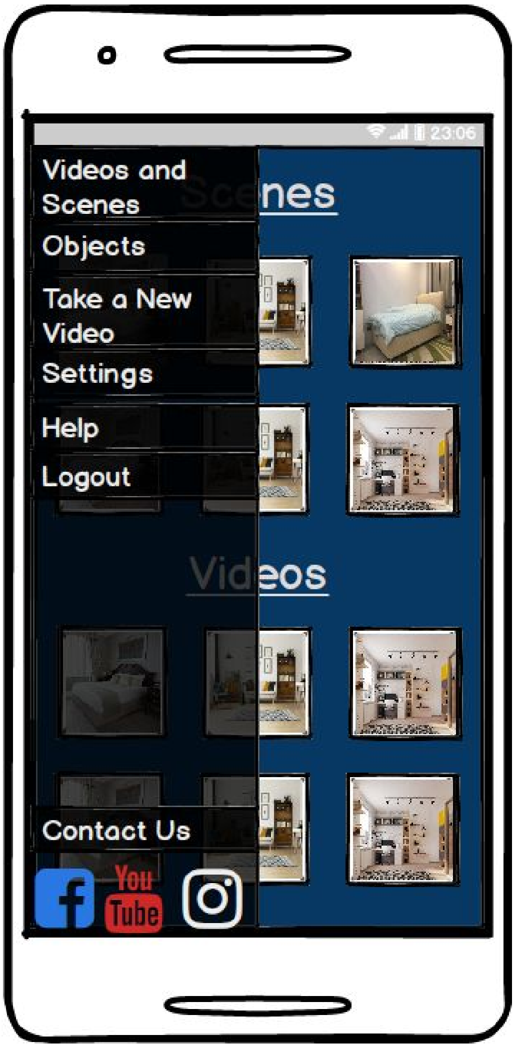

Figure 11: Side Menu Screen

This menu is accessible through the previous screen by clicking the top left icon or by sliding your finger towards the right from the left edge of the phone. It is used to move between different portions of the app.

3.5.5.5. Objects

Figure 12: Objects Screen

Accessible through the side menu, this screen displays the previously captured objects of the user.

3.5.5.6. Scan Object

Figure 13: Scanning Object Screen

Accessible through the side menu, this functionality allows the user to scan a single object instead of the entire room. User rotates nearly 360 degrees around the central object to complete the capturing process.

3.5.5.7. Workshop

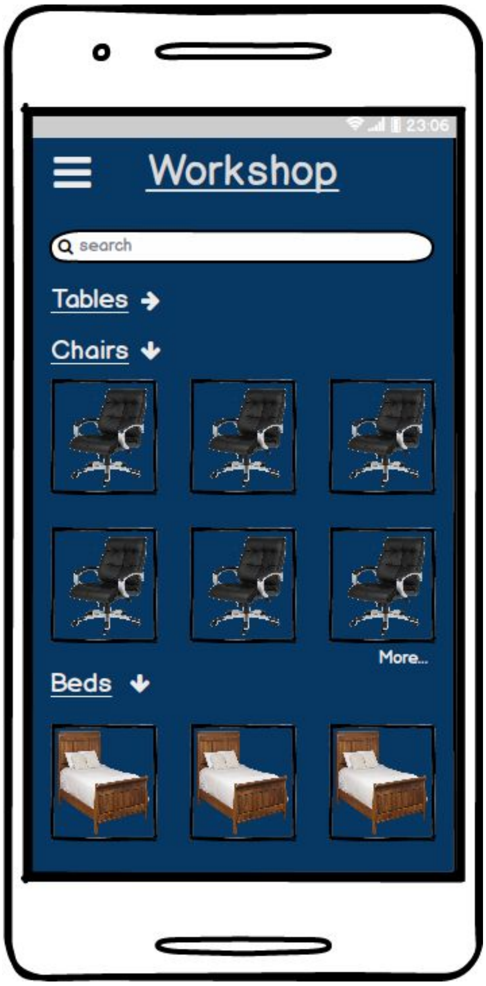

Figure 14: Workshop Screen

Accessible through the side menu, this screen shows the shared library of objects, referred to as the workshop, in the application database. Here, user can select objects to use in their scenes.

3.5.5.8. Scan Room

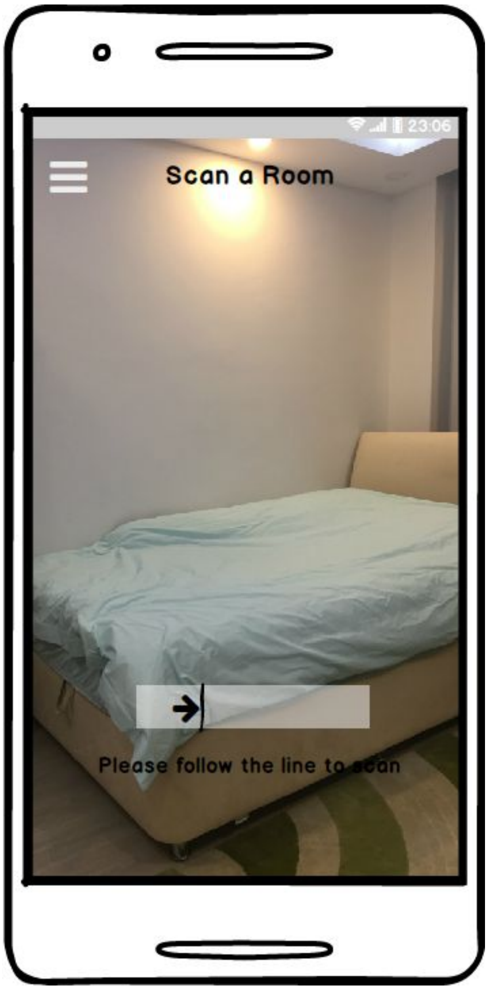

Figure 15: Scanning Room Screen

Accessible through the side menu, this is the video capturing functionality used for scanning rooms. The user is prompted to turn 360 degrees around themselves in order to complete the process.

3.5.5.9. Settings

Figure 16: Settings Screen

Accessible through the side menu, this screen is used to change the settings of the application by the user.

3.5.5.10. PC Login

Figure 17: Pc Login Screen

This is the initial screen encountered by the user when the application is used through a computer. Here, existing users can login while new users can sign up.

3.5.5.11. PC Sign Up

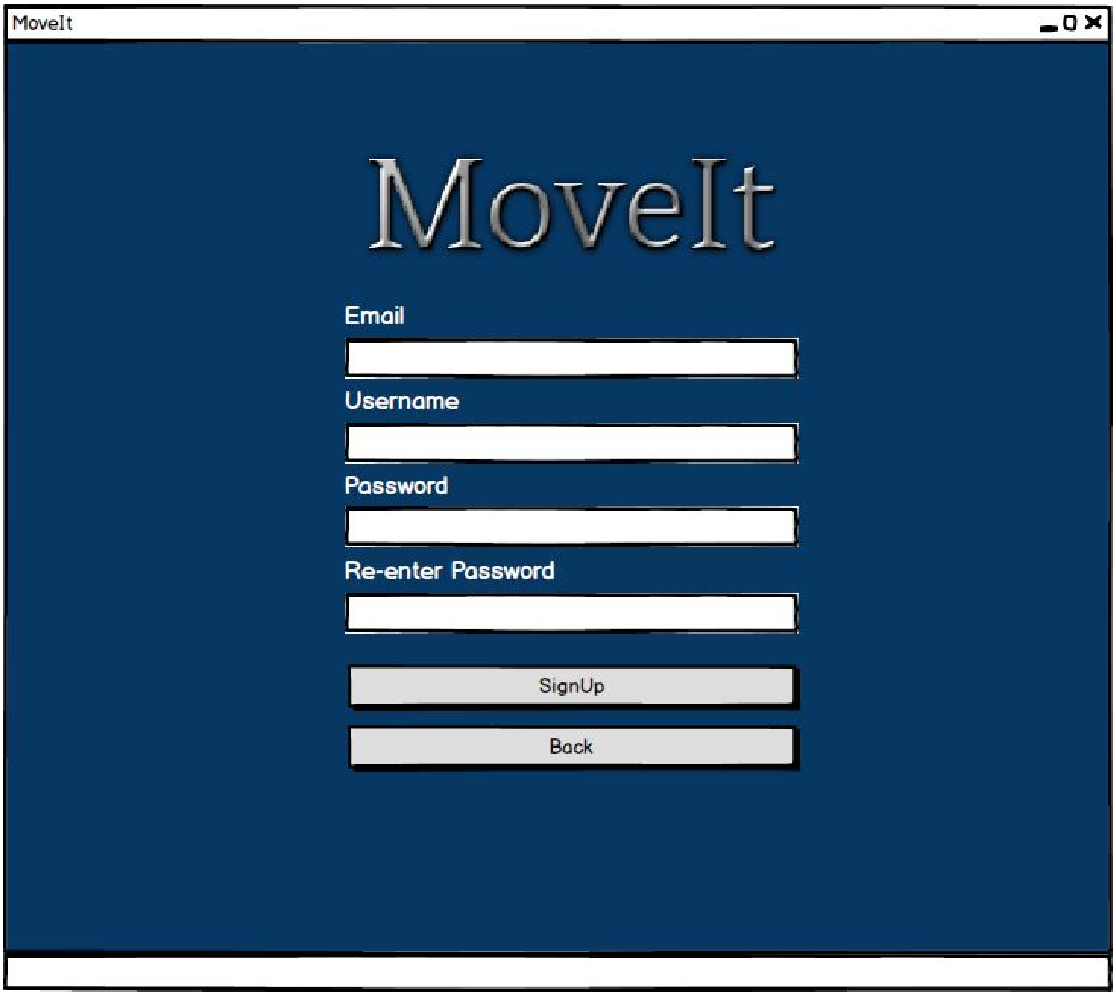

Figure 18: Pc Sign Up Screen

Similar to the mobile version, if the user clicks the signup button on the login screen, this page appears. New user can use this page to create an account using their emails and chosen passwords.

3.5.5.12. PC Main Screen

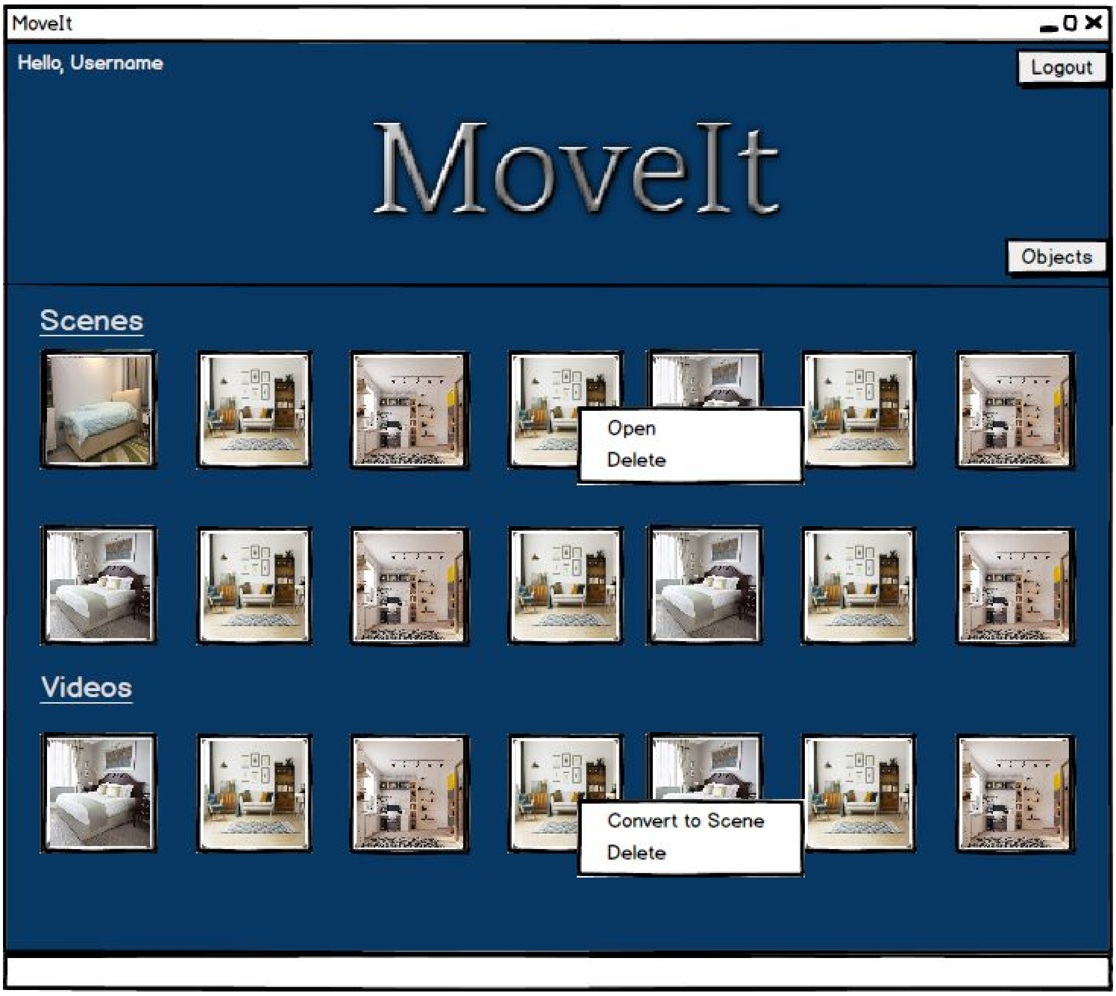

Figure 19: Pc Scenes Screen

Main screen accessed right after the login/signup screen. Here, user can see the existing videos and their respective rendered scenes. If a video does not have an existing screen, Convert to Scene option will be available open clicking on its image. Scenes will have Open and Delete as their two options when clicked. Two buttons to view objects and logging out are present.

3.5.5.13. My Objects

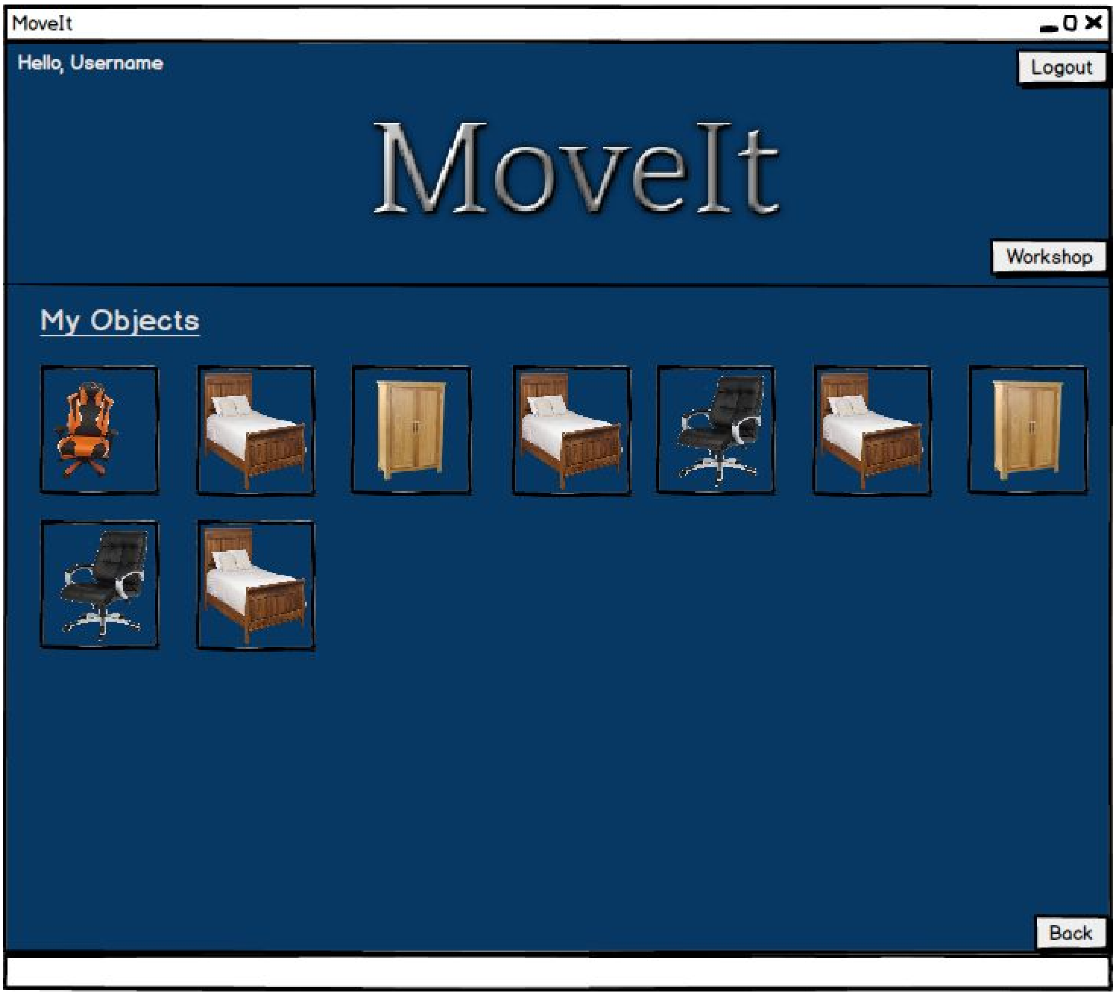

Figure 20: Pc Objects Screen

Accessible through the objects button present in the previous page, this screen displays the previously captured objects of the user similar to the mobile version of the app.

3.5.5.14. PC Workshop

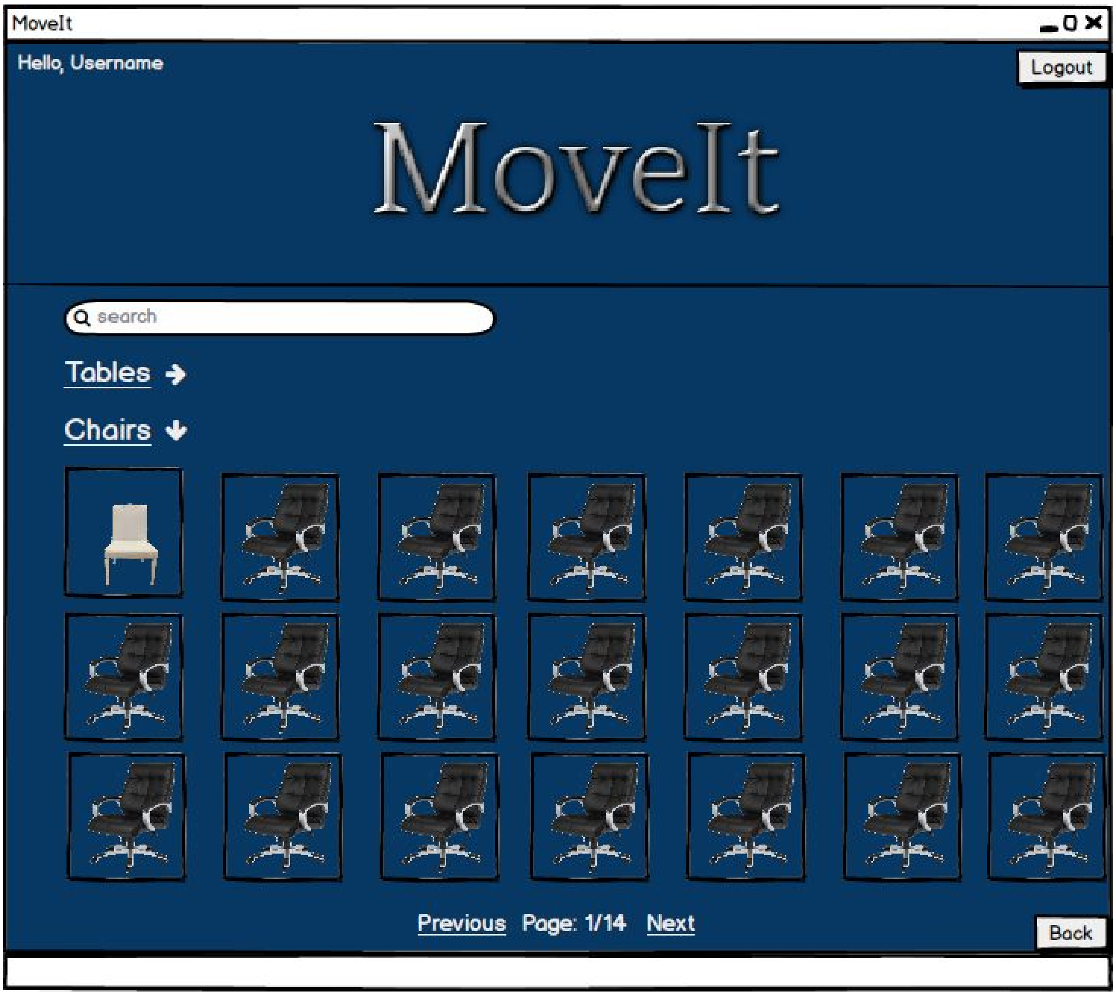

Figure 21: Pc Workshop Screen

Accessible through the Workshop button present in the previous page, this screen shows the shared library of objects, referred to as the workshop, in the application database. Here, the user can select objects to use in their scenes.

Figure 22: Pc Objects Addition Screen

When clicked on a workshop in the previous page this prompt appears. By selecting yes, user add the workshop item to their objects.

3.5.5.15. VR UI

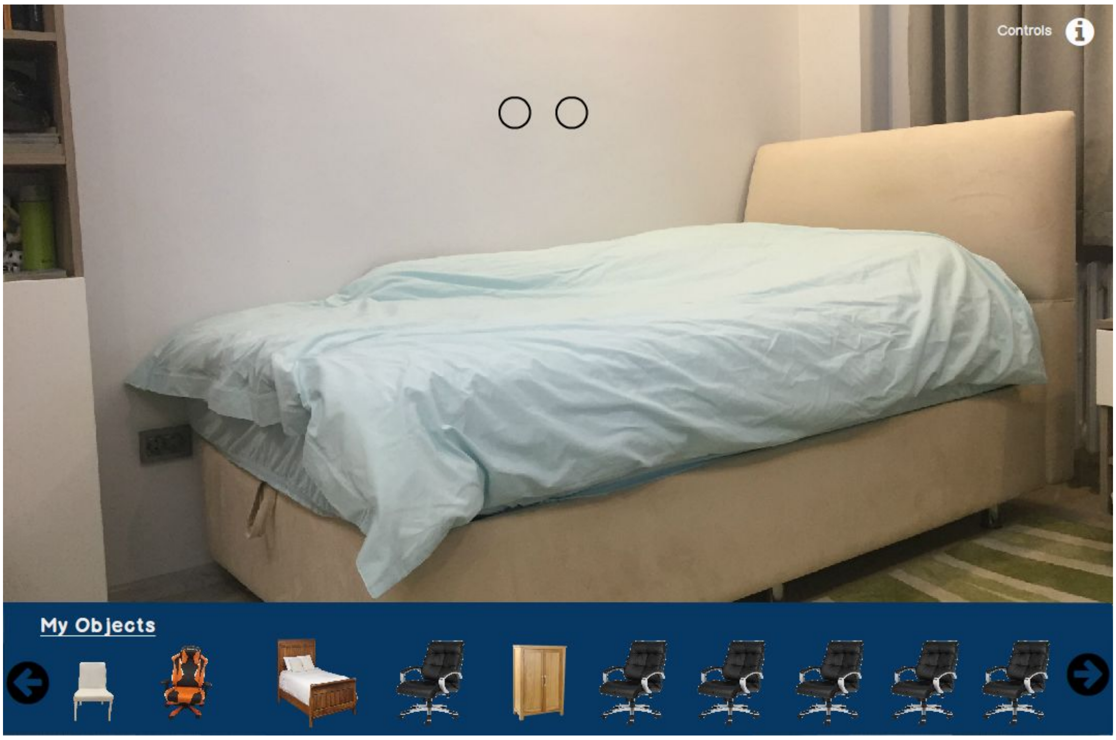

Figure 23: Pc VR Screen

This screen showcases the main functionality of the app. By clicking open for one of the scenes in the Main screen, this screen is created for the 3D VR headset. By wearing the connected headset and holding its control devices, user can fully see and interact with the depicted room. These interactions include such things as removing objects from the scene or placing new objects from my objects bar in the bottom, moving objects around and changing their size.

3.5.5.16. Place Objects

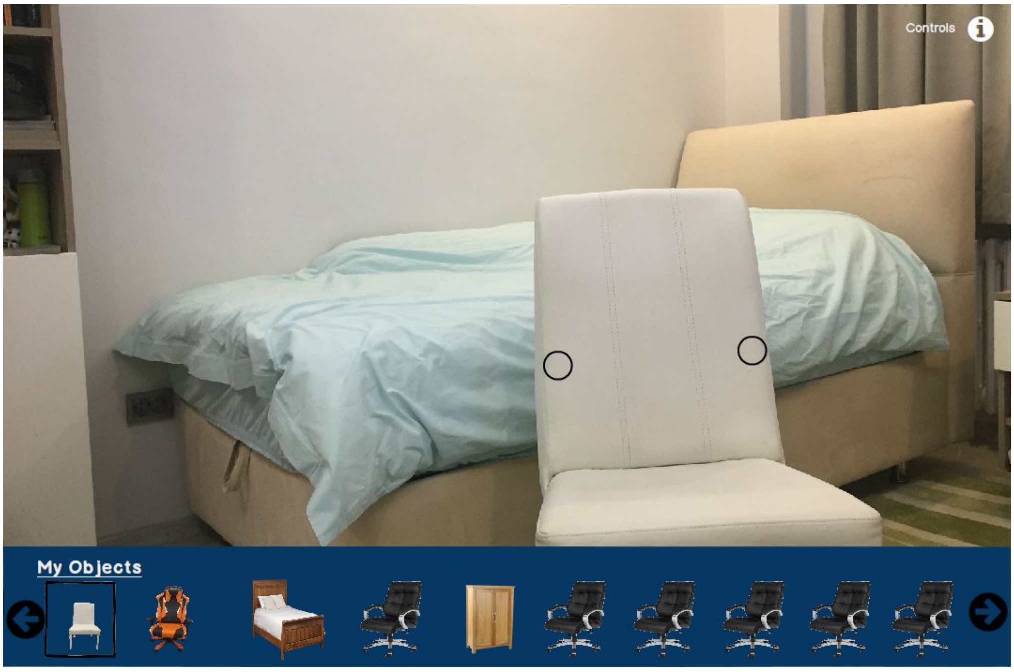

Figure 24: Placement Before Screen

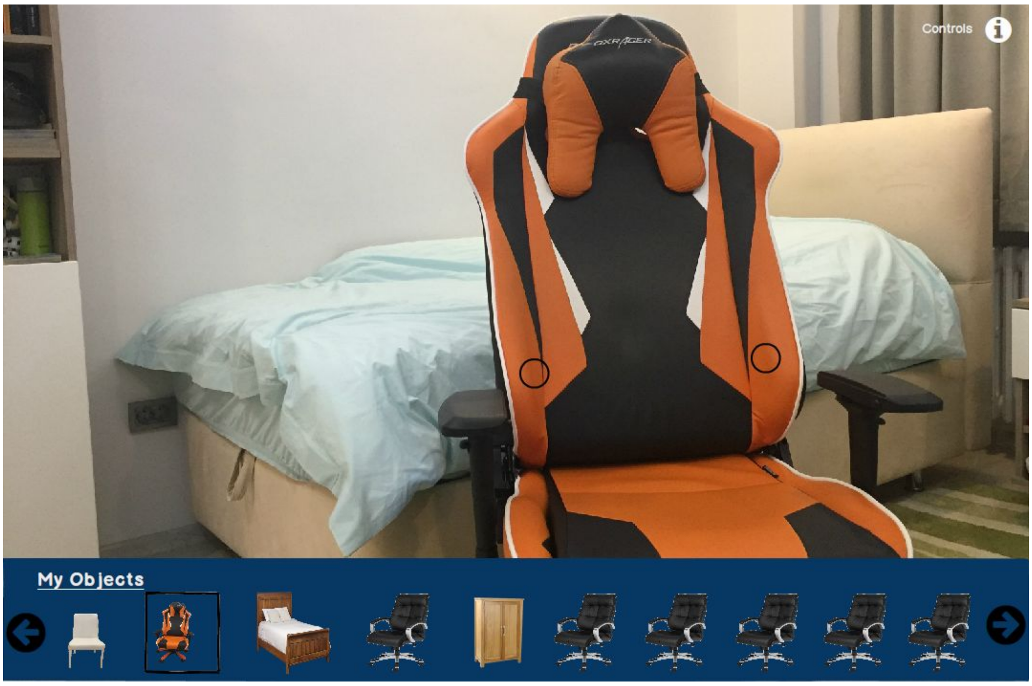

Figure 25: Placement AfterScreen

Two of the previously mentioned functionality can be seen as the white chair is removed from the room and the orange chair is placed in its place

4. Other Analysis Elements

The following is a discussion for various other analysis elements.

4.1. Consideration of Various Factors

MoveIt is an application that touches and enrichens various public issues and global factors. There is no distinct effect of public health issues on design of MoveIt. Concerning the public safety issue, users can modify their room with the instructions given in an earthquake-safe object placement manual. Before they do this modification, they can use MoveIt to find the best possible furniture arrangement and visualize it with ease. Since the aim of the application is providing its users with a convenient way of visualizing their rooms and changing their furniture arrangements in various ways, MoveIt enables the users to plan ahead without doing the actual work and work smarter instead of harder. This makes MoveIt an excellent application for ensuring public welfare and comfort. In terms of global factors, MoveIt is a convenient medium for furniture stores to promote their merchandise and provide their customers with additional features by using the application. This usage contributes to economic factors and makes design of MoveIt affected by global factors. With its publicly available dataset, MoveIt will enable users from all around the world to share their furniture as well as culture with the rest of the users. People will be able to see different objects that they have not seen in their living spaces and cultures which makes cultural factors play an important role in design of MoveIt. The users sharing their objects with the rest of the world is also going to help other people to design their rooms which contributes to social factors and bores the need to take social factors into account when designing MoveIt.

	Effect level	Effect
Public health	0	None
Public safety	3	The need to place furniture in a natural disaster-safe manner
Public welfare	8	People can change their room arrangement without actually changing it in real life
Global factors	5	Furniture stores can market their furniture which contributes to economic factors
Cultural factors	4	With publicly available dataset, people can see furniture from other cultures, and they can share their own furniture with the world
Social factors	5	People could share their objects so that they can help other people design their rooms

Table 1: Factors that can affect analysis and design.

4.2. Risks and Alternatives

	Risk	Likelihood	Effect on the project	B Plan Summary
Risk 1	Not being able to properly find bound of the objects in the scene	4	Since this is the main aim of the MoveIt, it largely reduces the effectiveness of the final product.	Using predetermined meshes (bounds) for the objects.

Risk 2	Lack of computational power during the reconstruction step	3	This causes an increase in the amount of time required to process the users' videos when a lot of videos are being processed at the same time.	Use the users' personal computer to do some of the reconstruction.
Risk 3	Problems in generating VR environments	3	This might cause us to be unable to display our scenes in a VR environment.	Using a conventional 3D environment for the scene.

Table 2: Risks

4.3. Project Plan

The following are the plan for the whole project.

4.3.1. Work Packages

Work package title	Leader	Members involved
WP1: 3D Reconstruction and Semantic Segmentation Research	Pınar	Mert, Ünsal
WP2: Render and Unity Engine Research	Faruk	Barış
WP3: High Level System Design	Barış	Faruk, Mert, Pınar, Ünsal
WP4: Initial Implementation of the Android Application	Mert	Barış, Faruk
WP5: Semantic Segmentation and 3D Reconstruction from Videos	Ünsal	Barış, Faruk, Mert, Pınar
WP6: Low Level System Design	Barış	Faruk, Mert, Pınar, Ünsal

WP7: External Subsystem Design and Implementation	Pınar	Barış, Faruk
WP8: Final Implementation of the Android Application	Mert	Pınar
WP9: PC Application Implementation	Ünsal	Barış, Faruk
WP10: VR Application Implementation	Faruk	Barış, Mert, Pınar, Ünsal
WP11: Final Implementations and Polishing	Ünsal	Barış, Faruk, Mert, Pınar

WP 1: *3D Reconstruction and Semantic Segmentation Research*			
Start date: *04.11.2019* **End date:** *29.11.2019*			
Leader :	*Pınar*	**Members involved:**	*Mert, Ünsal*
Objectives: The main objective of this WP is learning key knowledge about 3D reconstruction. This knowledge will be useful when performing 3D reconstruction of the indoor videos and the objects.			
Tasks: ***Task 1.1 Research 3D Reconstruction*** **:** *This task will consist of researching 3D reconstruction from online sources and peer knowledge. A document consisting of key points of the research will be produced.* ***Task 1.2 Research Semantic Segmentation*** **:** *This task will consist of researching semantic segmentation from online sources and peer knowledge. A document consisting of key points of the research will be produced.* ***Task 1.3 Apply Knowledge on Basic Model*****:** *This task will consist of designing and developing a basic model that does 3D reconstruction and semantic segmentation on a chosen image. The task will be useful in terms of gaining hands on experience in the researched domains.*			

Deliverables ***D1.1:*** *Basic model that does 3D reconstruction and semantic segmentation* ***D1.2:*** *A document consisting of key knowledge acquired in this WP*

WP 2: Render and Unity Engine Research			
Start date: *04.11.2019* **End date:** *29.11.2019*			
Leader :	*Faruk*	**Members involved:**	*Barış*
Objectives: The main objective of this WP is to learn basic rendering for objects and scenes using Unity. This research will be used for rendering scenes and objects before the reconstruction and semantic segmentation. In addition, Unity research will be used for converting the rendered scene and objects into a VR scene.			
Tasks: ***Task 2.1 Unity Research***: *Here basic functions of Unity and what can we do with it will be researched. Moreover, we will learn how to send a rendered scene to VR and how to change the keyboard controls.* ***Task 2.2 Rendering Research***: *Here, we will learn how to render scenes and objects using Unity. Also, we will learn how to convert a video taken from users' phones into renderable scene, and how can we process it.*			
Deliverables ***D2.1:*** *A rendered scene on Unity* ***D2.2:*** *A rendered object on Unity*			

WP 3: High Level System Design			
Start date: *26.11.2019* **End date:** 10.12.2019			
Leader :	*Barış*	**Members involved:**	*Faruk, Mert, Pınar, Ünsal*
Objectives: The main objective of this work package is to create a high level design for the proposed system. It will use the analysis report as basis while designing it. It should also be helpful and reliable in the later stages of the project.			
Tasks: ***Task 3.1 Decompose the System***: *This task will decompose the proposed system into subsystems in order to make it easier to work with.* ***Task 3.2 Hardware/Software mapping***: *This task will determine the I/O of the proposed system and it will also decide the overall view of the Android application. This task will also propose the interface and the controls of the VR/PC usage.*			

***Task 3.3 Persistent Data Management**: This task will determine the architecture of the database system which will reside on a remote server. This will also decide how relevant data will be stored.* ***Task 3.4 Access Control and Security**: This task will determine the required security measures such as the authentication of the user data.* ***Task 3.5 Global Software Control**: This task will determine the required controls for the globalization of the proposed system.* ***Task 3.6 Boundary Conditions**: This task will determine the required conditions for boundary cases such as initialization, termination and the failure of the system.* ***Task 3.7 Subsystem Services**: This task will analyze the properties of the proposed subsystems in the Task 3.1.*
Deliverables ***D3.1:** High Level Design Report*

WP 4: Initial Implementation of the Android Application			
Start date: *12.12.2019* **End date:** *31.12.2019*			
Leader:	*Mert*	**Members involved:**	*Barış, Faruk*
Objectives: *Objective of this WP is implementing an Android application in order to start the semantic segmentation and reconstruction. Basic functions of the application will be implementing UI functions, scanning the room using the camera of the phone and sending the video to reconstruct.*			
Tasks: ***Task 4.1 Learning Android Studio:** The first step will be learning how to use Android Studio and decide which methods and interfaces will be used.* ***Task 4.2 Implementing the GUI**: In order to use and implement the other functions of the application, the GUI part must be done. Even if there is no functionality on the pages, all the GUI part will be done.* ***Task 4.3 Scanning the Room and Sending the Video**: In order to move on and test the semantic segmentation, scanning the room and sending the video to a computer must be done.*			
Deliverables ***D4.1:** An Android application with basic functionalities*			

WP 5: Semantic Segmentation and 3D Reconstruction from Videos			
Start date: *12.12.2019* **End date:** *29.02.2020*			
Leader:	*Ünsal*	**Members involved:**	*Barış, Faruk, Mert, Pınar*
Objectives: The *objective of this WP is implementing semantic segmentation and 3D reconstruction from videos. Implementing this is one of the core tasks of our project.*			

Tasks: ***Task 5.1 Dataset Search for Semantic Segmentation*** **:** *The purpose of this task is finding labeled furniture datasets and training a semantic segmentation model or fine tuning a pre-trained model.* ***Task 5.2 3D Reconstruction from Indoor Video*** **:** *This task will consist of reconstructing a video of an indoor place into a 3D scene.* ***Task 5.3 Semantic Segmentation of 3D Scene*****:** *This task will consist of performing semantic segmentation on the previously reconstructed indoor scene.* ***Task 5.4 Setting Boundary Boxes for Objects*****:** *Purpose of this task is setting accurate bounding boxes for the previously labeled objects.*
Deliverables ***D5.1:*** *3D reconstruction model* ***D5.2:*** *Semantic segmentation model*

WP 6: Low Level System Design			
Start date: *28.01.2019* **End date:** *10.02.2019*			
Leader :	*Barış*	**Members involved:**	*Faruk, Mert,Pınar,Ünsal*
Objectives: The main objective of this work package is to refine the high level design of the proposed system into a more detailed version. It will use the high level design report as basis while designing it. It will specify the details of implementation and the strategies which will be followed. These will be done with respect to the engineering standards.			
Tasks: ***Task 6.1 Determining the Object Design Trade-offs*** **:** *This task will specify the necessary trade-offs caused by some external constraints such as time or economic cost.* ***Task 6.2 Description of Packages*****:** *This task will lay out the details of the packages with class diagrams as well as their descriptions. These packages were specified on the High level design report.* ***Task 6.3 Class Interfaces*****:** *This task will provide interfaces for each class. These classes will be outlined in terms of their high level purpose and low level structure according to their packages.* ***Task 6.4 Setting the Engineering Standards*****:** *During this task, engineering standards which were followed within the project will be finalized.*			
Deliverables ***D6.1:*** *Low Level Design Report*			

WP 7: External Subsystem Design and Implementation
Start date: *02.03.2020* **End date:** *15.03.2020*

Leader :	*Pınar*	Members involved:	*Barış, Faruk*
Objectives: *The main purpose of this work package includes; design of the database system and its proper integration with the rest of the system, setting up the server and using the proposed server to run the 3D reconstruction algorithms. It will also add the results to the integrated database.*			
Tasks: ***Task 7.1 Designing the Database :*** *During this task, the architecture of the database will be finalized and an ER diagram will be produced for further use.* ***Task 7.2 Integration of the Database*** **:** *During this task, the proposed architecture in Task 7.1 will be implemented to the server and will be integrated to the other subsystems of the project.* ***Task 7.3 Integration of the Server*** **:** *During this task, the server will be prepared. After that, the required algorithms for the 3D reconstruction will run on demand.* ***Task 7.4 Management of the Algorithm Results*** **:** *The I/O operations of the server will be inserted to the database within their respective tables.*			
Deliverables ***D7.1:*** *Fully functional database* ***D7.2:*** *Fully functional integrated server*			

WP 8: Final Implementation of the Android Application			
Start date: *16.03.2020* **End date:** *23.03.2020*			
Leader :	*Mert*	**Members involved:**	*Pınar*
Objectives: *The main objective of this WP is finishing the implementation of the Android application. In the previous Android application tasks, some features such as Workshop was not implemented. This WP will take care of the remaining not implemented features and finalize the design of the Android application.*			
Tasks: ***Task 8.1 Implementing the Workshop Feature*** **:** *The purpose of this task is implementing the Workshop feature into the application which allows the user to upload objects to the furniture database or downloading objects from the furniture database.* ***Task 8.2 Finishing the Design of GUI*** **:** *The purpose of this task is finishing the GUI design and implementation of the application.*			

***Task 8.3 Testing the Android Application**: This task will consist of testing and reviewing the application and performing any necessary changes. We will then ensure that the application works correctly and is ready for the demo.*
Deliverables ***D8.1:** The finalized version of the MoveIt Android application*

WP 9: PC Application Implementation			
Start date: *16.03.2020* **End date:** *31.03.2020*			
Leader :	*Ünsal*	**Members involved:**	*Barış, Faruk*
Objectives: *The main objective of this WP is finishing the implementation of the PC application. Here, GUI of the PC application will be completed, and rendering the videos that came from android application will be completed. Also, the functionalities such as Workshop, adding and getting objects from the database will be done.*			
Tasks: ***Task 9.1 Designing and Implementing the GUI of the Application** : This will be needed for the main functionalities of the application.* ***Task 9.2 Connecting Necessary Parts to the Remote Database:** This task will include fetching the required data from the database, such as, the login/sign up information, the implementation of the workshop and the saved objects for an account* ***Task 9.3 Rendering the Scenes with Their Respective Meshes**: This task will be done on Unity Game Engine in order to make an interactive environment from the scene data gathered from the user.* ***Task 9.4 Implementation of the User Controls**: This task will finalize and then implement the user controls in the interactive environment implemented on Task 9.3.*			
Deliverables ***D9.1:** Fully functional PC application*			

WP 10: VR Application Implementation			
Start date: *01.04.2020* **End date:** *20.04.2020*			
Leader :	Faruk	**Members involved:**	Barış, Mert, Pınar, Ünsal
Objectives: The main objective of this WP is finishing the VR implementation using Unity and sending the scene from a PC to VR after the rendering. Also, converting the controllers of keyboard to VR controllers will be done.			
Tasks:			

Task 10.1 Finishing the VR Implementation: *Here, VR implementation will be completed using Unity to reflect the scene from the PC to the VR.* ***Task 10.2 Sending the scene to VR*****:** *After completing the PC application, the scene rendered in the PC will be sent to VR.* ***Task 10.3 Converting the Controls to VR*****:** *Here, switching from keyboard to VR controller will be done.*
Deliverables ***D10.1:*** *A working VR application that shows the scene rendered in the PC.*

WP 11: Final Implementations and Polishing			
Start date: *21.04.2020* **End date:** *08.05.2020*			
Leader :	*Ünsal*	**Members involved:**	*Barış, Faruk, Mert, Pınar*
Objectives: *The main objective of this WP is doing the final checks on the project, performing various tests and polishing. In this WP, the final report of the project will be written, the user manual will be produced and the application website will be finalized.*			
Tasks: ***Task 11.1 Perform Tests*** **:** *In this task, we will test the Android application, the PC application and the VR implementation of the project. We will perform the necessary bug-fixes and changes.* ***Task 11.2 Write the Final Report*****:** *The purpose of this task is writing the final report for the project.* ***Task 11.3 Write the User Manual*****:** *The purpose of this task is writing a user manual for the project. The manual will specify the user interactions and instructions.* ***Task 11.4 Finalize the Webpage Design*****:** *This task will consist of finalizing the webpage of our project. The webpage design will be done and necessary additions will be added to the website.*			
Deliverables ***D11.1:*** *Final Report* ***D11.2:*** *The Fully Functional MoveIt Application* ***D11.3:*** *The finalized MoveIt Application Webpage*			

Table 3: List of work packages

4.3.2. Gantt Chart

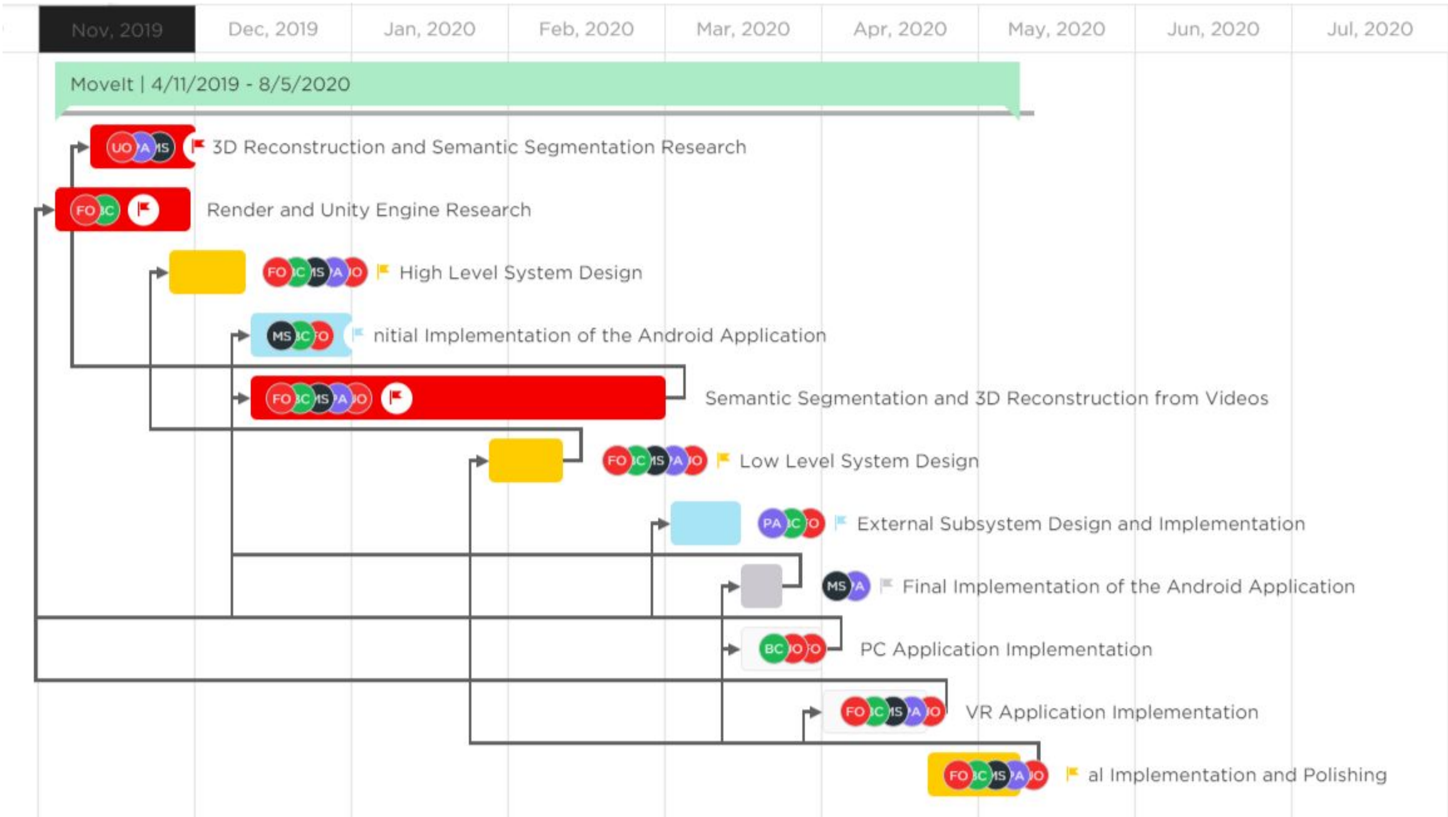

Figure 26: Gantt Chart

4.4. Ensuring Proper Teamwork

In order to ensure a proper teamwork within the team members during every stage of the project, we decided to use ClickUp™ as a tool to manage the distribution of work between the team members [14]. This app allows us to set deadlines for specific tasks while also specifying the ones that are responsible for that specific task. It also allows us to create dependencies between tasks which also provides further flexibility for the task management. It can also allow us to divide the project into different parts such as the documentation, the 3D reconstruction and rendering. This way, it lets us see the deadlines for each of the tasks related to their topic more clearly.

As for the version control of the application while developing it, we will use GitHub since it is the most proficient tool that would allow us to see the work done by each individual more clearly and lets us track the previous commits to the project. As of now, since we are still at the learning stage for the topic, we are currently not committing any code to the project's page. However, we will start to do so after the second week of December.

For the communication between the team members, we are using Discord, which is a similar application to Slack but slightly faster, as we are all familiar with it and it is one of the most stable communication tools available [15]. We are also using WhatsApp for solving urgent problems/answering urgent questions. Considering that we are using the tools mentioned above with the proposed project plan, we will assure that the teamwork between the team members will be appropriate.

4.5. Ethics and Professional Responsibilities

Since MoveIt requires scanned furniture of users' room that they want to change the appearance, we are planning to store every furniture scanned by every user in the database with their labels. However, these objects will be stored anonymously without any of their user information. This way, we will allow the use of non-owned furniture in the application while securing the users' privacy. The models and labels will not be shared by other users or third parties without getting permission in the case that companies want to use them for ads, and will just be kept in the database.

Required permissions such as accessing phone's storage and camera will be taken from users before using the application for the first time and the models will not be saved, or the camera will not open if they do not give the required permissions. No unnecessary permission will be asked. The addition of the scanned models to the dataset will be asked after each scan. This would allow the user not to upload their specified models if they wish to do so. However, they will not be able to use the application if they do not give some permissions such as accessing the camera.

Since all the videos and scenes will be kept in a database, users will be able to access their data via account from the MoveIt. No personal information will be asked during the signup phase, and only an email and a password will be required (or a Google account). All the scenes/videos and objects will be uploaded to the database as anonymous and private but this can be changed with users' permission. User information will not be shared by third companies.

While constructing the datasets for meshes, copyright and privacy issues will be considered and for all purposes, the National Society of Professional Engineers' Code of Ethics will be followed [13].

4.6. New Knowledge and Learning Strategies

The core of our project lies in its ability to extract the scene and object information from the video data provided. We expect this part to be the hardest part of our project to implement and as so most of our initial research went into this aspect of our project. All of our current research so far is gathered through Online Research. We particularly used papers from the Technical University of Munich. We are currently focusing on two different papers in order to find ways to recreate the 3D structure of the room and find boundaries for objects given multiple frames.

In order to recreate the 3D structure of the rooms, we are using "Efficient Online Surface Correction for Real-time Large-Scale 3D Reconstruction"[16]. We are hoping to learn the logic behind the 3D recreation and then implement a variation of it in our project. We are particularly interested in the knowledge presented in the chapter 3 as its content closely resembles our own.

For finding boundaries for household objects given a series of frames, we are looking into the paper "Real-Time Dense Geometry from a Handheld Camera"[17]. Contents of this paper greatly matches our own goals as they also aim to create a 3D depth map for images and reconstruct them using this data from a set of images. Currently we focus on the chapter 2.1 "Stereo Estimation Using Two Images" as it forms the basis for our upcoming work.

5. References

[1] “Microsoft Azure Bulut Bilgi İşlem Platformu ve Hizmetleri,” *Microsoft Azure Bulut Bilgi İşlem Platformu ve Hizmetleri*. [Online]. Available: https://azure.microsoft.com/tr-tr/. [Accessed: 02-Nov-2019].

[2] U. Technologies, “Unity,” *Unity*. [Online]. Available: https://unity.com/. [Accessed: 02-Nov-2019].

[3] “Download Android Studio and SDK tools : Android Developers,” *Android Developers*. [Online]. Available: https://developer.android.com/studio. [Accessed: 02-Nov-2019].

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” *Google AI*, 01-Jan-1970. [Online]. Available: https://ai.google/research/pubs/pub43022. [Accessed: 02-Nov-2019].

[5] Zhang, Ren, Sun, and Jian, “Deep Residual Learning for Image Recognition,” *arXiv.org*, 02-Dec-2015. [Online]. Available: https://arxiv.org/abs/1512.03385. [Accessed: 02-Nov-2019].

[6] J. Redmon, *YOLO: Real-Time Object Detection*. [Online]. Available: https://pjreddie.com/darknet/yolo/. [Accessed: 02-Nov-2019].

[7] R. Eisele, “SLA Uptime Calculator: How much downtime is 99.9%,” *SLA Uptime Calculator: How much downtime is 99.9% • Open Source is Everything*. [Online]. Available: https://www.xarg.org/tools/sla-uptime-calculator/. [Accessed: 02-Nov-2019].

[8] “OpenCV,” *OpenCV*, 14-Oct-2019. [Online]. Available: https://opencv.org/. [Accessed: 02-Nov-2019].

[9] “IKEA Place,” *Google Play'de Uygulamalar*. [Online]. Available: https://play.google.com/store/apps/details?id=com.inter_ikea.place&hl=tr. [Accessed: 02-Nov-2019].

[10] “Microsoft HoloLens: Mixed Reality Technology for Business,” *Microsoft HoloLens | Mixed Reality Technology for Business*. [Online]. Available: https://www.microsoft.com/en-us/hololens. [Accessed: 02-Nov-2019].

[11] “Sims,” *EA Games*, 14-Oct-2019. [Online]. Available: https://www.ea.com/games/the-sims. [Accessed: 02-Nov-2019].

[12]“Room Planner: Home Interior & Floorplan Design 3D - Apps on Google Play,” *Google*. [Online]. Available: https://play.google.com/store/apps/details?id=com.icandesignapp.all&hl=en_US. [Accessed: 02-Nov-2019].

[13] “Code of Ethics,” *Code of Ethics | National Society of Professional Engineers*. [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics. [Accessed: 10-Oct-2019].

[14] “ClickUp,” *clickup.com*. [Online]. Available: https://app.clickup.com/. [Accessed: 02-Nov-2019].

[15] “Discord - Free Voice and Text Chat,” *Discord*. [Online]. Available: https://discordapp.com/. [Accessed: 02-Nov-2019].

[16] R. Maier, R. Schaller, and D. Cremers, “Efficient Online Surface Correction for Real-time Large-Scale 3D Reconstruction,” *Procedings of the British Machine Vision Conference 2017*, 2017.

[17] J. Stühmer, S. Gumhold, and D. Cremers, “Real-Time Dense Geometry from a Handheld Camera,” *Lecture Notes in Computer Science Pattern Recognition*, pp. 11–20, 2010.

[18] *Lifetimes of cryptographic hash functions*. [Online]. Available: https://valerieaurora.org/hash.html. [Accessed: 09-Nov-2019].

6. Website

https://barisc22.github.io/MoveIt/