TED University

Department of Computer Engineering

Senior Design Project

Project short name: Virtuanance

Analysis Report

Cem TIRPANLI, Hakan Ahmet TEKİN, Ali Can KESKİN, Doğukan TERZİ

Supervisor: Tolga K. Çapın
Jury Members: Aslı GENÇTAV, Venera ADANOVA

Due Date of the Report
November 23, 2020

This report is submitted to the Department of Computer Engineering of TED University of course CMPE491-Senior Project

Table of Contents

Analysis Report

Project Short-Name: Virtuanance

1 Introduction

1.1 Purpose of the System

Virtuanance is being designed for industrial maintenance use. The main goal is to connect a professional with a technician that requires assistance over long distances. The project aims to create a platform which will allow the peers to communicate through verbal and visual actions for completing an operation with highest efficiency.

1.2 Scope of the System

Virtuanance focuses on the 'maintenance' field of remote solutions. There are multiple works done for 'training' and 'assembly/disassembly' fields. Maintenance field mostly tries to fix an already built machine with problems. The maintenance field is selected because any remote maintenance application negates the need for training personnel since the technicians do not need any knowledge about the system.

1.3 Objectives and Success Criteria

The success of Virtuanance will be determined by:

- Task Completion Rate
- Error Rate
- Time It Takes to Complete a Task

Task time and completion rates will show whether the application provided necessary information and assistance to both sides for understanding the operation being done.

Error Rates will show if the visual and verbal communication relayed enough information from one side to another. Verbal communication is the main source of information exchange, but it has limits. The visual cues aim to make peers understand what is happening more easily.

The project will be considered successful if the error rates and Completion Rates are like an on-site operation.

1.4 Definitions, Acronyms and Abbreviations

- HMD: Head Mounted Device
- AR: Augmented Reality
- VR: Virtual Reality
- Leap Hands: A digital version of user's hand that are placed in the simulation world.
- Model Target: 2D/3D models created by Vuforia

2 Current System

The current version is a tech demo showing off the capabilities of the technologies and a basic version of the project scope. It supports the most basic functionalities, essential technologies that are going to be used (LeapMotion, HTC Vive, Vuforia etc.) are integrated together. Basic object detection is set for creating digital models on top of the physical machinery. Leap Hands support hand menus and basic interaction with the digital models.

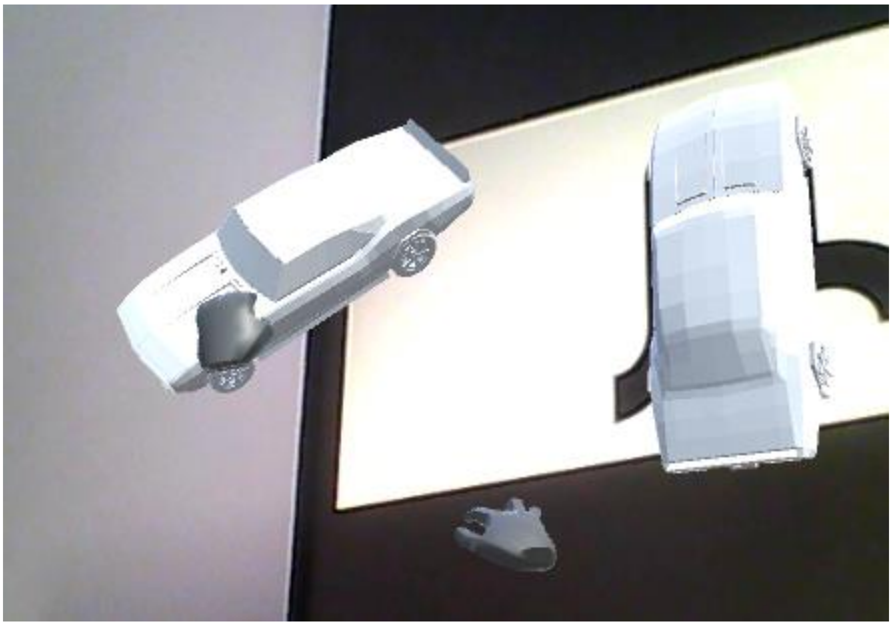

Figure 1 A car model is grabbed with Leap Hands

3 Proposed System

3.1 Overview

The project is divided into two main titles according to its objectives. The first is to show a technician who is inexperienced and does not know how to repair the machine, how to repair the machine through a direct contact with a professional. Therefore, the most important thing to do is to establish a good network. Besides, this project is to make an application that can be done for all products with 3D models, not for a specific machine repair. Therefore, this is a more general project than other remote maintenance projects. If Users use this application and if they have a 3D model, they will be able to repair the machine as they wish. The second main topic is to provide repair training. In today's conditions, it is almost impossible for a professional who specializes in a machine used in any company to go to any country and provide training. For this reason, the program we will make is for training and will save both time and money.

3.2 Functional Requirements

• Both the technician and the expert will connect the system through their own devices.

• Both users will be able to interact with the models, which are created by the application, via using their own devices.

• As controllers, users will use their own hands in order to interact with the application.

• Pre-defined hand gestures will be used by the users for in order to interact with the application.

• Application will recognize and track the real-life objects, which will be specified by the user, when the objects are inside the field of view of the technician’s device.

• The users will add their own model in order the application can recognize, track and allow them to interact with the models.

• Both users will be able to communicate with each other via the application.

• The application will give visual cues and feedbacks to the users about the objects that they can interact with.

3.3 Non-functional Requirements

3.3.1 Security

Virtuanance will make sure that all the data transferred between the expert and the technician will be encrypted.

3.3.2 Reliability

Reliability is an important goal of the Virtuanance. Object tracking and recognition should not be interrupted unless there is a major obstacle like the environment being too dark for the device in order it to be able to have a clear vision or an obstacle between the object and the device which will block the line of sight. Also, one of the main goals of the project is to support a range of 3D model types in order not to limit the users.

3.3.3 Performance

Virtuanance aims to provide well optimized and smooth tracking of the objects so that the users will not be disturbed and distracted. Animation data should be partially sent between the peers. Possible media transfer should be compressed for minimalizing the required bandwidth. Application should work in a decent framerate in order to ensure that the users will not be disoriented.

3.3.4 Scalability

Even if there will be multiple objects to track and recognize, Virtuanance will not have a decrease in the performance. The connection between users will be peer-to-peer so there will not be any server-related connection issues between the technician and the expert.

3.3.5 Usability

One of the key points for Virtuanance is to be easy to use and understand. Hand motions required to interact with certain aspects of the system (UI, models) will be minimalized to make it both easier to remember and use. There will be visual cues in order to ease the work of the technician. Vocal and written communication will be provided in order to make it easier for peers to understand each other and express their thoughts.

3.4 Pseudo Requirements

• Virtuanance will be developed as a desktop and AR/VR device application and will be implemented using Unity Engine.

• We will use Git as version control system integrated with GitHub and Jira to assure a better tracking and management of the tasks.

• C# will be the main development language, along with Vuforia [1] and Leap Motion [2].

3.5 System Models

3.5.1 Scenarious

3.5.1.1 Import Models

- Use-case Name: Import Models
- Actor: Technician and Professional
- Entry Condition: User interacts with the import model settings in home page of the program.
- Exit Condition: User import 3D CAD model to the system.
- Flow of events:
    - A button will appear in the program that allows to import
    - File chooser will open when the button is pressed.
    - The user will select a 3D format model and upload it to the system.

3.5.1.2 Rendering the environment

- Use-case Name: Rendering the environment
- Actor: Technician
- Entry Condition: Launch the application
- Exit Condition: Successfully rendering environment without any error.
- Flow of events:
    - As soon as user launches the application, AR/VR camera will initialize.
    - System will start to track pre-defined objects.

3.5.1.3 Connecting the Users

- Use-case Name: Connecting the Users
- Actor: Professional and Technician
- Entry Condition: Establishing connection between technician to professional
- Exit Condition: Users terminates the connection
- Flow of events:
    - User and technician enter a shared code to provide connection
    - System will connect users peer to peer
    - System will start transmitting data between technician and professional

3.5.1.4 Interaction with Rendered Objects

- Use-case Name: Interaction with Rendered Objects
- Actor: Technician
- Entry Condition: Starting interaction with object with using real tools
- Exit Condition: User deselects the object
- Flow of events:
    - Technician will choose a real tool to interact with the object
    - System sends the data of selected object to the professional

3.5.1.5 Professional Feedbacks to the Technician

- Use-case Name: Feedback
- Actor: Professional and Technician
- Entry Condition: Professional starts interacting with the model data that sent from the technician’s client.
- Exit Condition: Sending feedback to the technician
- Flow of events:
    - Professional interacts with the model with predefined UI tools
    - Confirms changes and sends the feedback

3.5.1.6 Terminating the Application

- Use-case Name: Terminating the Application
- Actor: Professional and Technician
- Entry Condition: Technician and Professional done repairing part.
- Exit Condition: User exits successfully
- Flow of events:
    - When users done with successfully repairing, clicks the exit button.

3.5.2 Use-Case Model

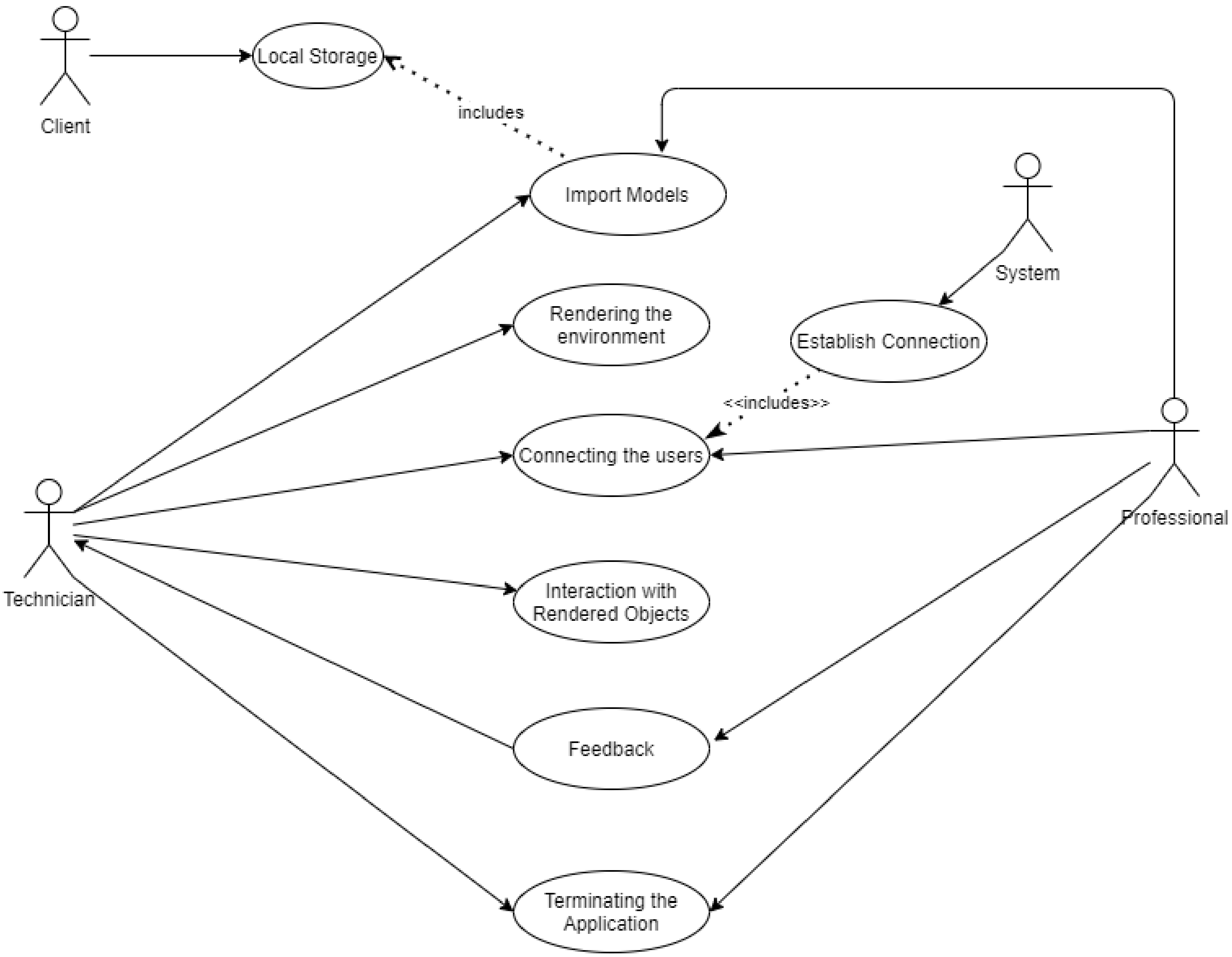

Figure 2 – Use case model

3.5.3 Object and Class Model

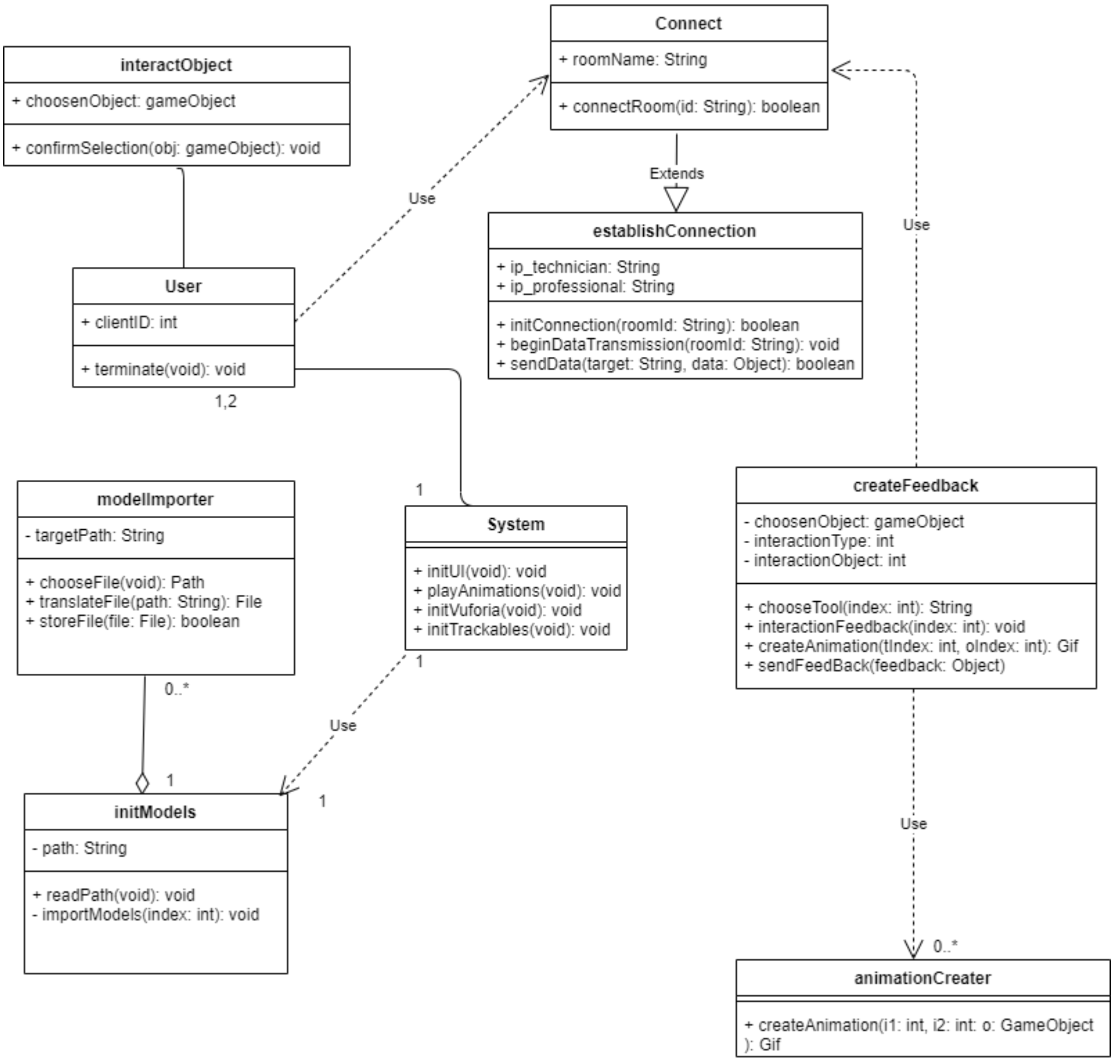

Figure 3 – Class diagram

Classes of Virtuanance

System: Once the user launches the application, system will handle all the initializations. These initializations include initializing models, initializing Vuforia tracker and initializing UI.

initModels: This class is called in the beginning of the launch. This class handles the importing of trackable objects, such as custom imported models.

modelImporter: modelImporter class is called with UI interaction. Basically, whenever the user wants to import a new model, this class will be called, and it will create a file chooser popup menu. After choosing the file, this class will translate the 3d model to .obj format and will store it in the local system.

User: This class defines the basic user properties. It can terminate the connection established, and it holds a clientID value which is specially given to every user.

interactObject: This class is used by technicians to interact with the augmented objects. Methods of this class is called whenever the user touches any trackable objects.

Connect: This class gets the required values from the peers and passes them to the establishConnection class.

establishConnection: Whenever two peers try to connect to the same room (specified with the roomName in Connect class), this class will try to establish connection between the peers. After establishing connection, it will try to start data transmission between the peers. sendData function will be used to specifically sending data between the peers.

createFeedback: Objects that are being sent from the sendData function of establishConnection class will appear in the professional's screen, and professional will use this class to create feedback animations with the use of predefined tools.

animationCreater: This class will help creating animations that are being specified in createFeedback.

3.5.4 Dynamic Models

3.5.4.1 State Chart

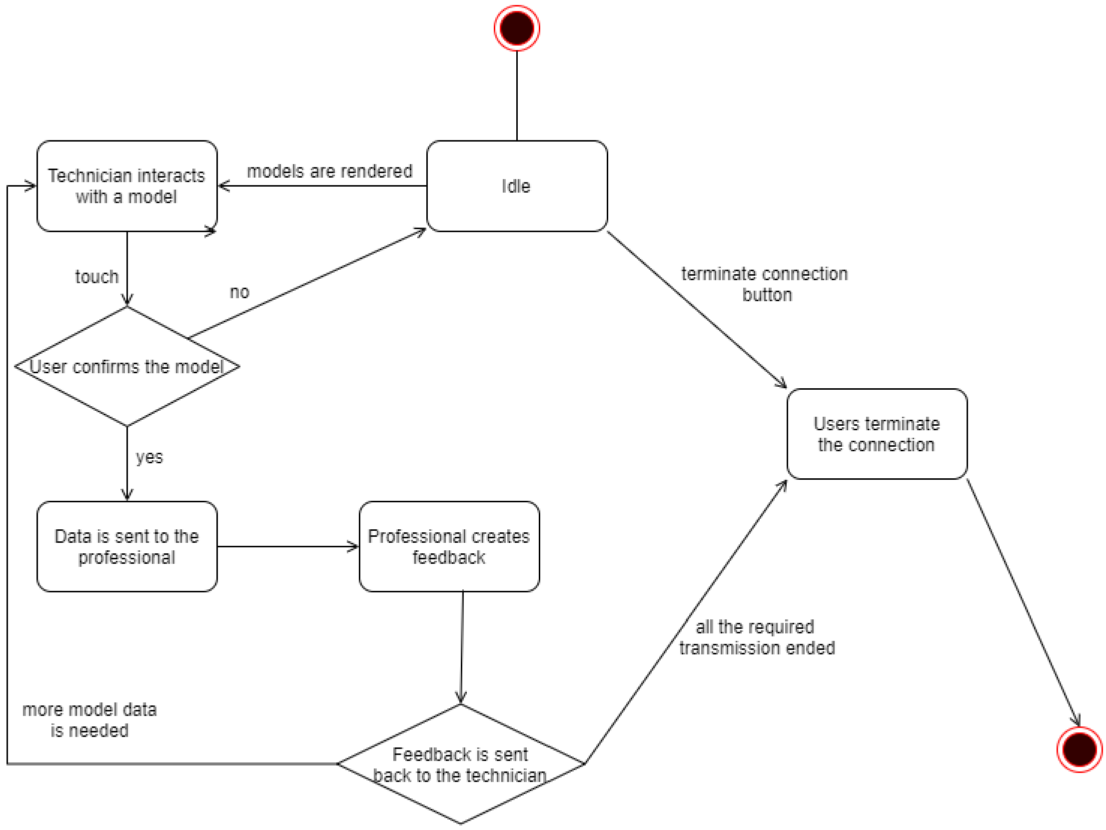

Figure 4 – State chart

This state chart defines the process of technician getting a feedback from the professional. Technician will interact with the rendered model first. Then the model's data will be sent to the professional if user confirms the selection. After data is sent, professional will then begin to process data and create a feedback. After generating the feedback and sending it back to the user, all the required data transmission will come to an end and connection will be optionally terminated. If more model data is needed, technician will keep interacting with them in their client, and professional will be receiving the data.

3.5.4.2 Activity Diagram

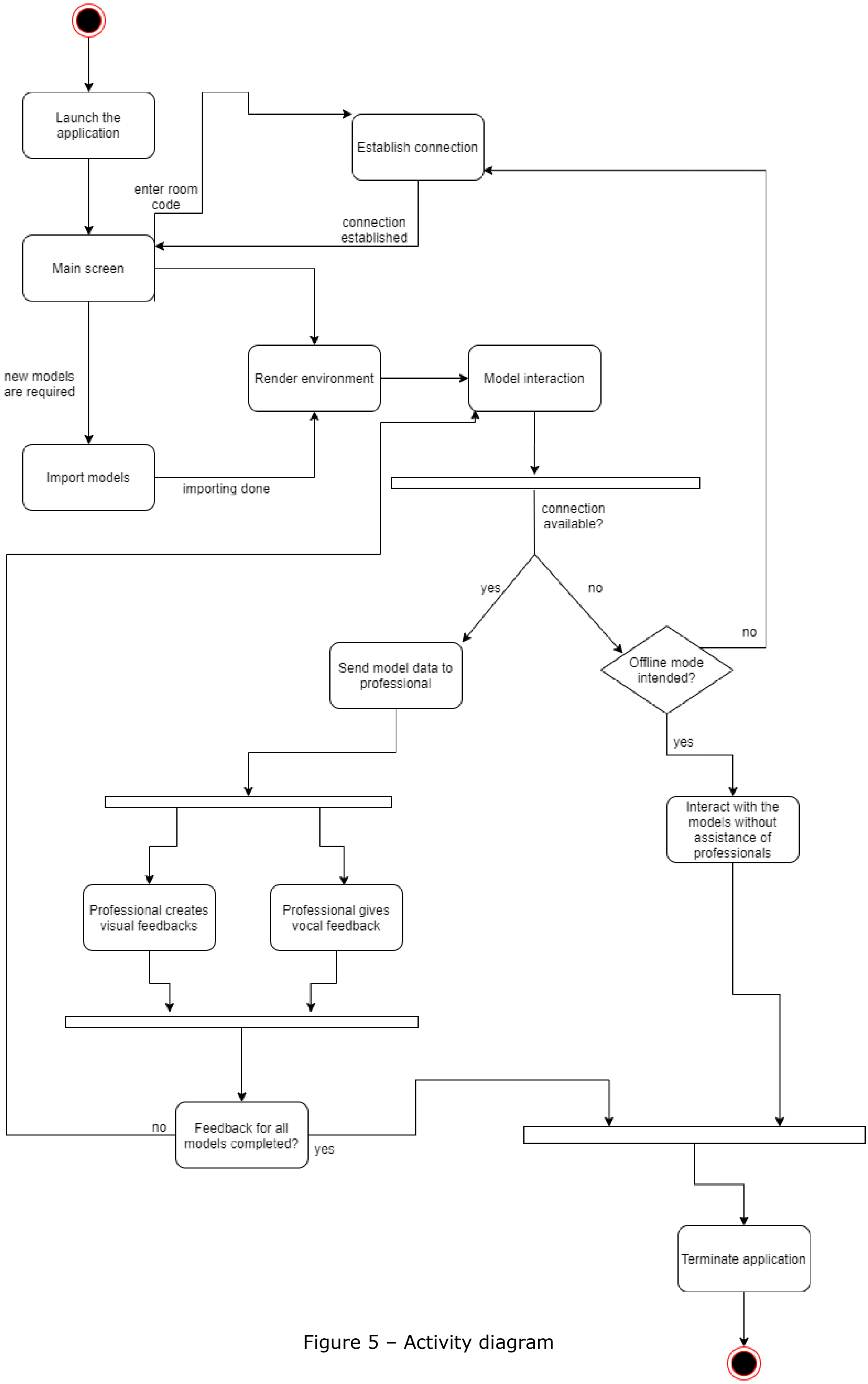

Figure 5 – Activity diagram

In figure 4, there are the possible life cycles of the application. Once the application is launched, users will be redirected to main screen, where they can choose to import new models, establish connection with each other using the room code, or start rendering the environment. After technician's machine starts rendering the environment and starts interacting with the models, if the connection is available, the data will start to be sent to the professional. If not, application will ask the technician if they want to use the offline mode or not. If they don't want to use offline mode, user will be redirected to the connection menu. If they want to use it, then they will interact the models in offline mode, in where they can easily rotate and scale objects but can't get professional assistance. After data is being sent to professional, professional will be able to give feedbacks real-time. They can create animations with their client application to give a feedback, or they can give verbal feedbacks. After the feedback is done, connection persists. If more help is needed on other models, technician can interact with other models and this feedback loop may continue. If done, the procedure is completed, and users can terminate their clients.

3.5.4.3 Sequence Diagram

Scenario #1: Technician and professional launch the application. They connect to the same room. System establish their connection. Technician interacts with the environment. Interacted data is sent to the professional. Professional analyze the data, and give the required feedback by creating animations, or by guiding the technician verbally. After the feedback loop is completed, both users terminate their client applications.

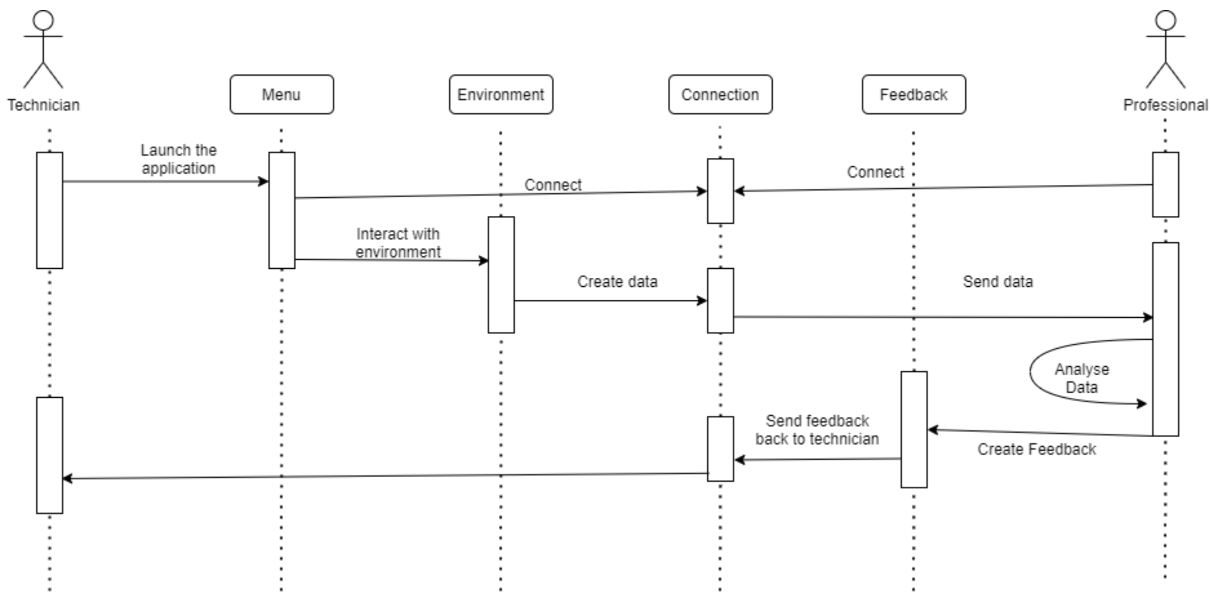

Figure 6 – Sequence diagram with connection available

Scenario #2: Technician launch the application. Professional is not available to connect right away. Technician starts interacting with the environment. When they select a model, they rotate it, move it or rescale it in their visual screen, but not in real environment. They use the program in offline mode to observe the object without interacting with them in real life.

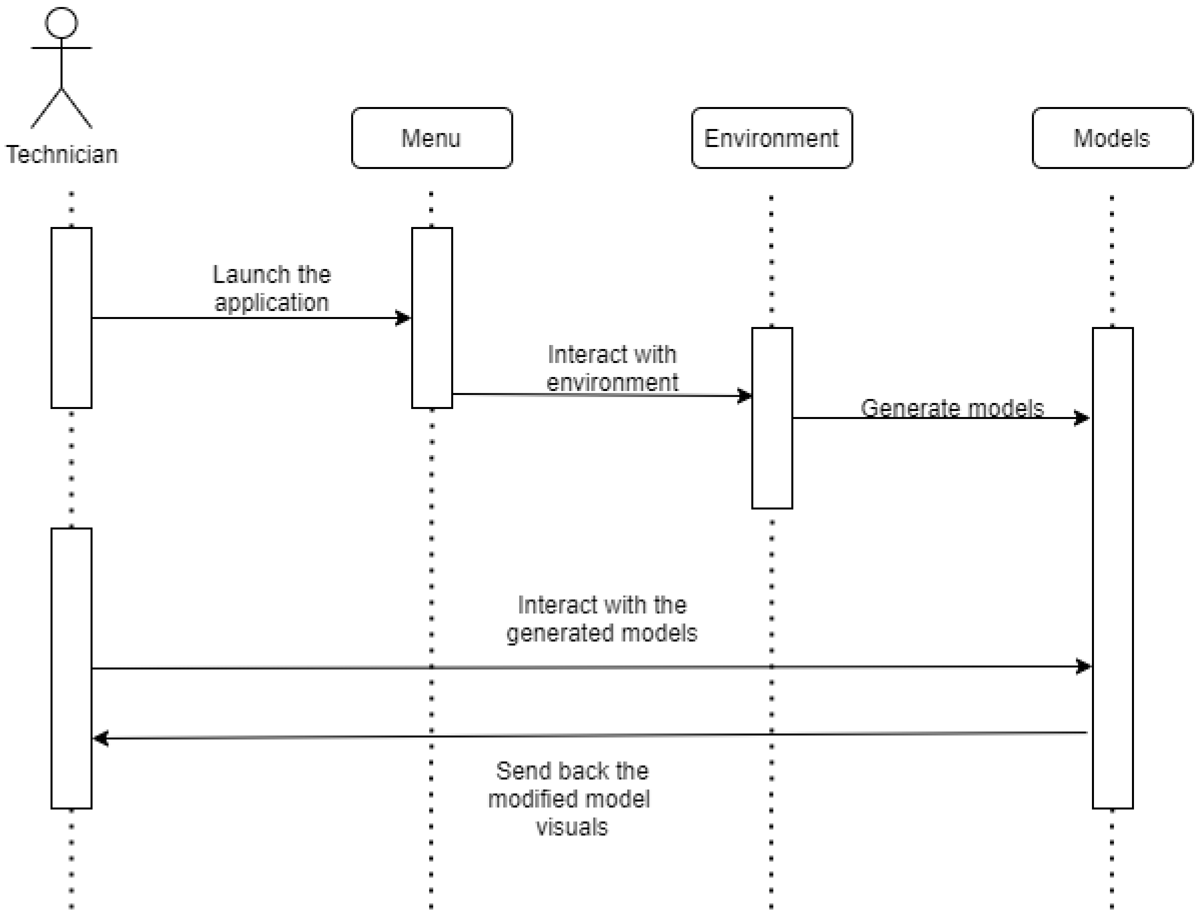

Figure 7 – Sequence diagram with no connection

3.5.5 User Interface

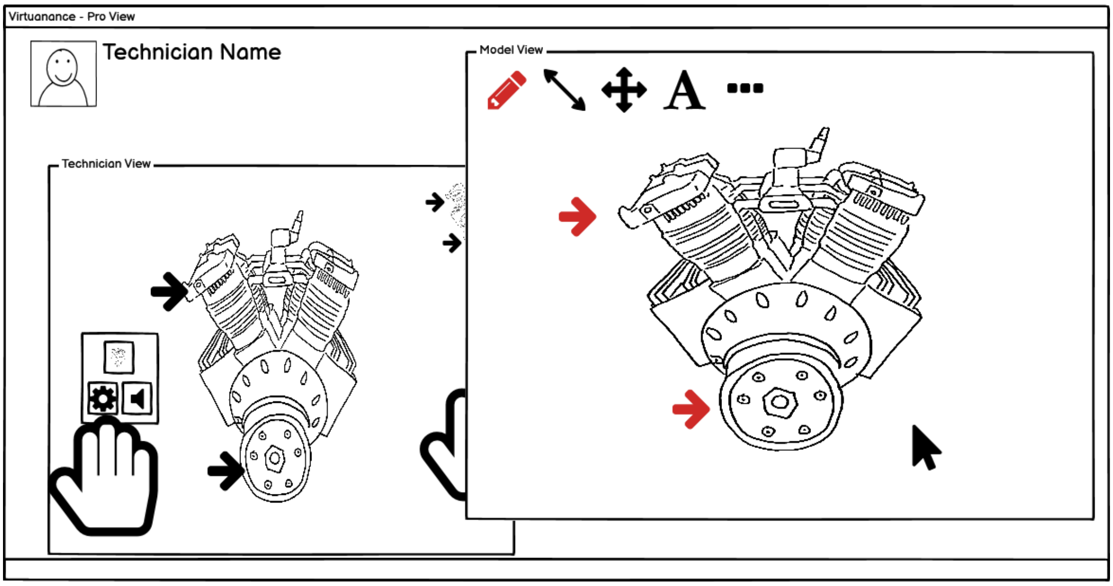

Figure 8 Professional View

- **Figure 8** shows a professional's point of view when an operation is underway. The professional sees what the technician is doing and a model which they can overlay visual cues into.

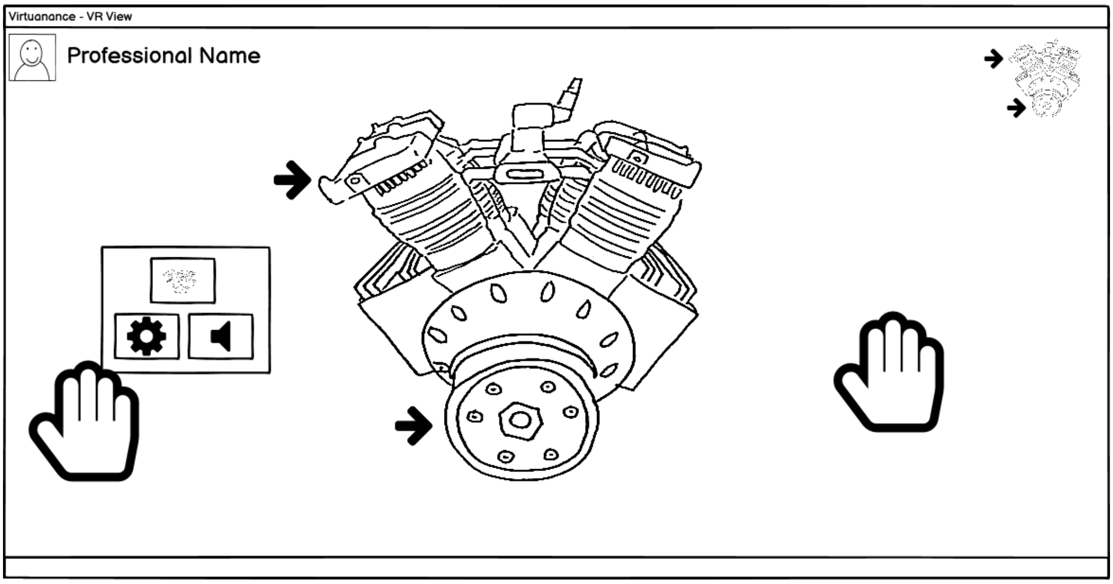

Figure 9 Leap Hands with Menu Opened

In Figure 9, a hand menu is displayed when the left palm is facing towards the user. Also, on the top right corner there is a small copy of the engine which shows the visual cues that are also placed on the actual engine.

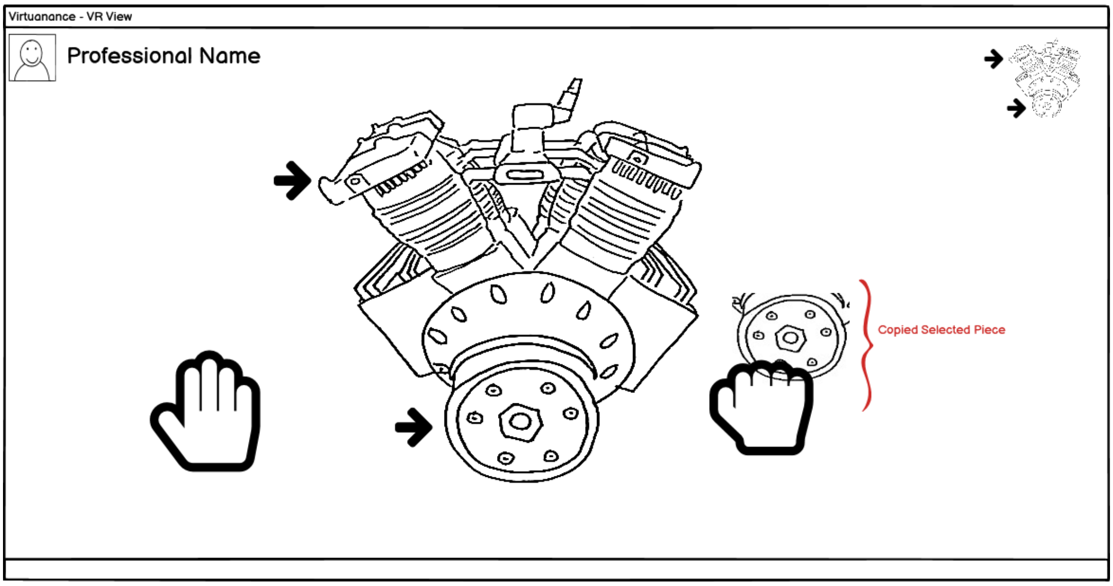

Figure 10 Selecting a piece of the model

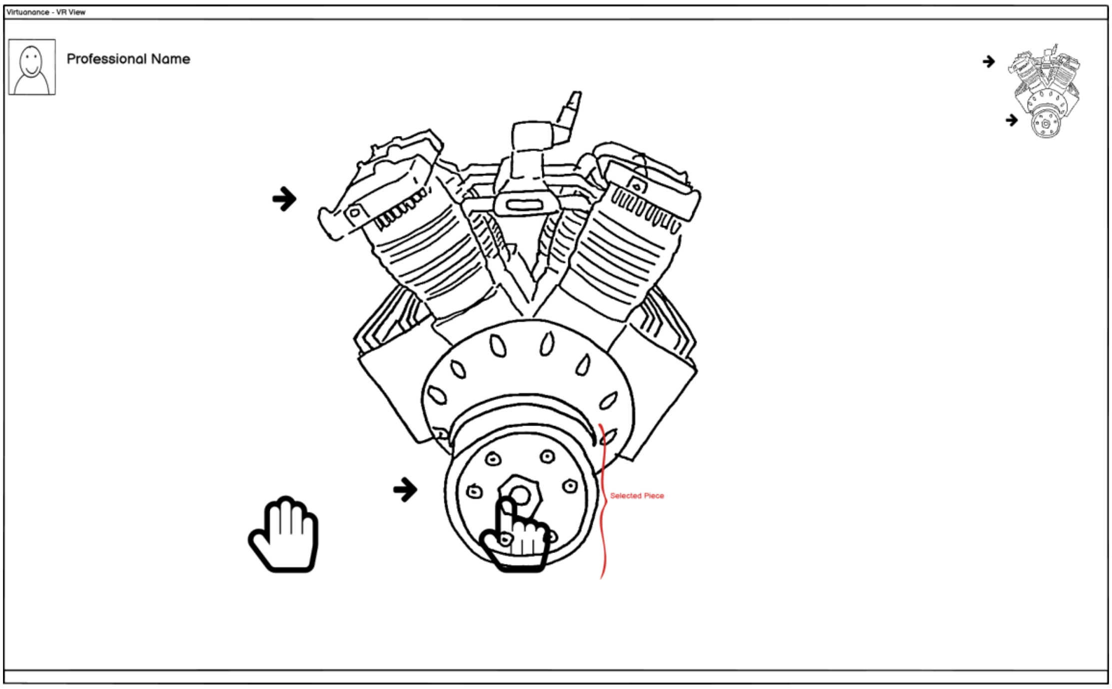

Figure 11 Interacting with a copy of the selected piece

- **In Figures 10-11**, the technician selects a piece of the engine and creates a duplicate of it. Then, this duplicate is moved to a different spot in the world.

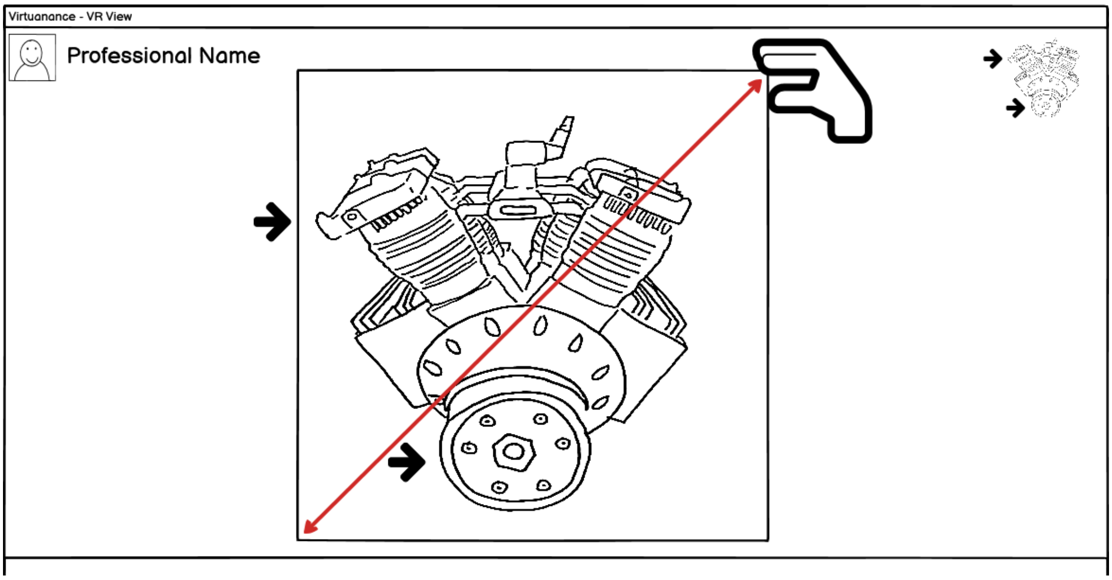

Figure 12 Rescaling the Entire Model

Figure 12 shows an outline denoting the edges of the whole engine. The user can use this outline to interact with the model in various ways (e.g. scaling, repositioning).

Figures 9-10-11-12 show a typical scenario of a maintenance operation being done on an engine from the technicians POV. The technician can use their hands to interact with the simulation objects. The engine image represents a digital model of the engine being overplayed on top of the actual one.

4 Glossary

AR: (Augmented Reality) It is 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, somatosensory and olfactory.

Git: It is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

HMD: It is a display device, worn on the head or as part of a helmet (See Helmet-mounted display for aviation applications), that has a small display optic in front of one (monocular HMD) or each eye (binocular HMD).

HTC Vive: The brand of VR glasses that we use for this project

Leap Motion Controller: it is an optical hand tracking module that captures the movements of your hands with unparalleled accuracy.

Vuforia: It is an augmented reality software development kit for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to recognize and track planar images and 3D objects in real time.

VR: (Virtual Reality) It is a simulated experience that can be similar to or completely different from the real world. Applications of virtual reality can include entertainment (i.e. video games) and educational purposes (i.e. medical or military training).

5 References

[1] “Vuforia Developer Portal”. [Online].

Available: https://developer.vuforia.com/. [Accessed: October 23, 2020]

[2] “Unity Asset for Ultraleap Hand Tracking”. [Online].

Available: https://developer.leapmotion.com/unity/. [Accessed: October 23, 2020]