Techniques For Collaboration in Virtual Reality
Master Thesis
Mohammad Moeen Valipoor

Madrid - June 2019
This thesis is submitted to the ETSI Informáticos at Universidad Politécnica de Madrid in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering.

Master Thesis
Master Universitario en Ingeniería del Software – European Master on Software Engineering
Thesis Title: Techniques For Collaboration in Virtual Reality
Thesis no: EMSE-2019-11
June 2019

Supervisor:
Angélica de Antonio Jimenez
Department: Lenguajes y Sistemas Informáticos e Ingeniería de Software ETSI Informáticos
Universidad Politécnica de Madrid

External Supervisor:
Raúl Martínez López
Site Lead at Gofore Spain
To whom that are far from each other.
# Table of Contents

1 Introduction ........................................................................................................................................... 6

2 State of the art ........................................................................................................................................ 7

2.1 Current Solutions ............................................................................................................................. 7
  2.1.1 InsiteVR .................................................................................................................................. 8
  2.1.2 Bigscreen ............................................................................................................................... 8
  2.1.3 MeetinVR ............................................................................................................................... 9

2.2 Project Background ......................................................................................................................... 10
  2.2.1 Virtual Conference Room .................................................................................................... 10
  2.2.2 Existing Functionalities ....................................................................................................... 11
  2.2.3 Project Architecture ........................................................................................................... 12

2.3 Design Sprint .................................................................................................................................. 12
  2.3.1 Defining goal and finding solutions ..................................................................................... 12
  2.3.2 Decision Making .................................................................................................................. 13
  2.3.3 Sketching ............................................................................................................................ 14
  2.3.4 Prototyping .......................................................................................................................... 15
  2.3.5 Testing .................................................................................................................................. 16
    2.3.5.1 Testing Results ............................................................................................................... 17

3 Requirements ..................................................................................................................................... 19

3.1 Requirements Specification ............................................................................................................. 19
  3.1.1 High-level Requirements ....................................................................................................... 19
  3.1.2 Non-verbal Behaviour ........................................................................................................... 20
    3.1.2.1 Lip Sync ......................................................................................................................... 20
    3.1.2.2 Expressing emotions ................................................................................................. 20
    3.1.2.3 Expression of Like ...................................................................................................... 21
    3.1.2.4 Expression of Question ............................................................................................. 21
  3.1.3 Trello Integration ..................................................................................................................... 22
    3.1.3.1 Trello board Demonstration .................................................................................... 22
    3.1.3.2 Add a new card ............................................................................................................ 22
    3.1.3.3 Remove a card .............................................................................................................. 22
    3.1.3.4 Modify a card ................................................................................................................ 23

4 Development Strategy ....................................................................................................................... 23

4.1 Agile Approach ............................................................................................................................... 23

4.2 Version Control ............................................................................................................................... 24

4.3 Tools ............................................................................................................................................... 24
  4.3.1 Game Engine & programming language ............................................................................. 24
  4.3.2 SDK & libraries ..................................................................................................................... 25
  4.3.3 3D Modeling ........................................................................................................................ 28
  4.3.4 Constraints ............................................................................................................................ 28

4.4 Project time & effort estimation .................................................................................................... 29
  4.4.1 Estimated Time ..................................................................................................................... 29
5 Software Architecture ................................................................. 31
6 Non-Verbal Behaviour Feature .................................................. 33
   6.1 Difference between Immersion and Presence .................................. 33
   6.1.2 Realism vs. Believability of Avatars ........................................ 34
       6.1.2.1 AvatarSDK .................................................................. 35
       6.1.2.2 Uncanny Valley ......................................................... 36
6.2 Implementation ....................................................................... 38
   6.2.1 Facial Expressions .................................................................. 38
   6.2.2 Different Solutions ............................................................ 39
7 Agile Practices in VR ................................................................. 44
   7.1 Agile Methodology ................................................................. 44
   7.2 Trello Integration .................................................................. 45
   7.3 Showing the Trello Board in VR ............................................... 46
   7.4 Adding, Removing and modifying cards .................................... 47
   7.4.1 Adding cards ..................................................................... 47
   7.4.2 Removing cards .................................................................. 49
   7.4.3 Editing cards ..................................................................... 49
8 Testing ..................................................................................... 50
   8.1 Non-Verbal Behaviour ........................................................... 50
   8.1.1 Testing of expressing emotions using tone analyzer .................. 51
   8.1.2 Testing of expressing emotions using controller ....................... 54
   8.2 Trello Integration Testing ....................................................... 56
   8.2.1 Testing of Trello Board ...................................................... 56
9 Conclusions, Challenges and Future Work .................................. 57
   9.1 Conclusions ................................................................. 57
   9.2 Challenges and Future Work ................................................ 58
10 References ................................................................. 60
1 Introduction

New advancements of technology is dramatically changing the way we communicate. Nowadays, the way we work and collaborate with each other is very different even from a decade ago. A lot of projects, especially in the IT industry, are being done remotely with the collaboration of remote teams. In different organizations, employees are using different services like VOIP, video calls, text chat and email to communicate. These types of communication has gotten more popular during the past years.

According to a study conducted by Zug, Switzerland-based serviced office provider IWG(International Workplace Group) [1], 70% of people tend to work remotely at least once a week and 53% of the people working from a distance at least half of the week. This phenomenon is called telecommuting. Although working from a home office seems very comfortable and it saves time because employees do not have to commute to the office, it has its own disadvantages. For example, if they want to communicate with one of their colleagues, the quality of a Skype call is not as high as a face to face conversation. On a video call, some parts of the non-verbal behaviour that is related to facial expressions or the movements of the body is missed. This issue is even more severe when it comes to text chat, emails or voice calls. Written communication can cause a lot of misunderstandings. There is a famous research by Albert Mehrabian [2][3] in which he claims that 90% of our communication is non-verbal. Based on his communication model and the results of his research, in a normal face to face conversation, 7% of the communication is the literal content of the message, 38% is the volume of the voice, intonation and tone. Moreover, up to 55% of the communication consists of body language.

One of the technologies that can make a revolutionary change in the quality of communication and take it to the next level, is Virtual Reality. With the potential of this technology, it is possible to build environments in which people can interact and communicate with each other in a way that is closer to the reality. The main difference of VR compared with the 2D monitors in which one can have video calls or voice calls is that there is a 6 DOF(degrees of freedom). You are immersed in an environment that is not reality anymore, but in some cases it might feel more believable. For example, the feeling of presence in a virtual environment like a conference room and seeing believable avatars of other people, can make you feel more present in a meeting compared with a
virtual voice call meeting that one doesn’t know if other participants are taking a nap on the other side or paying attention to the meeting.

In this thesis, I worked on an existing Virtual Conference Room that was made during my internship with the collaboration of another EMSE student. During the time I was working on my thesis, I decided to focus on techniques for collaboration in VR and elaborate the existing VR conference room by adding new features to it. Since the main purpose of this project was to create an environment for communication and collaboration, based on my investigations I came to conclusion that non-verbal behaviour is one of the most important aspects of the communication. Therefore, I decided to work on research and implementation of a non-verbal communication in this environment. Additionally, I investigated the potential of VR for managing Agile Practices and having daily meetings in this environment in the second part of the thesis.

2 State of the art

Virtual Environments in which individuals can communicate with each other are getting more and more popular with the appearance of new generations of the VR devices. Comparing these environments with the real world environment, there are different obstacles and limitations that should be tackled. For example, participants are usually represented with an avatar in the virtual environments. The avatar should have a satisfying appearance that gives the feeling of having a close identity to the participant’s characteristics. Besides that, it should be able to express the participant’s verbal and nonverbal behavior.

2.1 Current Solutions

Nowadays, there are some solutions for communication in VR, like VRChat, but they are mostly made for entertainment reasons. There are very few solutions for corporate usage that has a formal atmosphere in which users can hold meetings or interact with each other. For example, having a presentation in a virtual environment or holding sprint review meetings in which users can show the backlog items, is still not very common. In the following section some of the current VR solutions for corporate usage are introduced.
2.1.1 InsiteVR

This VR solution is made for engineers, architects and constructors. Using this application, users can see the 3D interior and exterior plan of a building and modify it. It is possible to walk around the building and make changes by drawing on the different parts of the building. This solution increases the customer engagement and trust because they can experience it from the human perspective (not looking at a raw 2D plan). This also eliminates design ambiguity and helps to identify problems before construction begins and it’s too late.

![Image 1: InsiteVR](image)

However, as you can see the users are represented with only the name and a VR headset 3d model. The controllers also show the movement of the hands. This is one of the most important drawbacks of this solution. Because users will not be able to see any kind of facial expressions or non-verbal cues. This might affect the quality of communication in this environment.

2.1.2 Bigscreen

This is one of the best existing collaborative VR solutions. It has the following features:

- Virtual Worlds
- Custom Avatars
- Private & Public Rooms
- Voice Chat
- Desktop Sharing
- Controllers

However, this application doesn’t aim the corporate usage and it is more appropriate for entertainment purposes, but it can also be used in a corporate environment because it has features like desktop sharing. The avatars in this application, are one of the best available ones in the market.

Image 2: Big Screen

2.1.3 MeetinVR

This solution is not yet released, but apparently, It is going to be one of the game-changing solutions in the near future. The avatars are pretty realistic and at the same time believable. Oculus Avatar SDK is used for the user’s representation. This SDK has the following features:

- LipSync
- Facial expression
• Torso representation
• Hand movements and nonverbal expressions with hands

In addition to virtual representation (avatars) that affects the participant’s feeling of presence, the virtual environment (e.g. objects, room, the scale of objects, lights and so on) is also a key factor in giving the feeling of presence to the participants. However, this thesis is more focused on the virtual representation of the participants in the virtual environment than the appearance and effects of the virtual environment itself on the feeling of immersion and presence.

![Image 3: MeetinVR](image)

### 2.2 Project Background

#### 2.2.1 Virtual Conference Room

Gofore is a large size IT service management company with more than 550 employees distributed in 9 different countries. Currently, the company has 4 offices in Finland, 1 in the UK, 1 in Germany, 1 in Spain and 1 in Estonia. This international atmosphere has led the team to work remotely on different projects for more than 120 customers that are geographically distributed around Europe. In the Spain branch, by considering the widely distribution of teams and customers around the globe, the company decided to start a research project about virtual reality. This project started to evaluate the capabilities and potentials of Virtual Reality in the field of virtual collaboration and communication. The idea elaborated and eventually, they decided to create a virtual reality conference room in which employees can communicate with each other, hold
meetings like daily standups and so on. The project started in October 2018 with 2 interns (including myself) from the EMSE program. The research, development, validation and demonstration of the project continued until the end of February 2019.

2.2.2 Existing Functionalities

By the end of the internship, the Virtual Conference Room had the following functionalities:

- **Creating a meeting room for up to 8 people**
- **Voice transmission of users:**
  Users can talk with each other inside the meeting room.
- **Head and hands movement transmission:**
  Users can see the avatar of the other users and when they move their head and hands in the real world other users can see the movements inside the virtual room.
- **Customizing avatars:**
  Users can choose their avatar type and choose their hairstyle, hair color, name, and t-shirt color.

When the internship finished, there were functionalities that remained in the backlog of the project which could enhance the performance of the program. Some of the new possible functionalities that could be added were:

- **Adding face and facial expressions to the avatars:**
  Changing faceless avatars in the existing version.
- **Enhancing the user’s hand movements detection:**
  The movement of the hands does not have enough accuracy. They are tracked from the wrist. It would look more natural if they are being tracked from the arm.
- **Presentations and/or screen sharing in the VR environment:**
  Users should be able to view their presentation slides or share their desktop screen with other users.
2.2.3 Project Architecture

The architecture of the program was designed and evolved in time. For example, in the beginning there was no Lobby concept and it was added to the project after the 3rd Sprint. This is the final version of the component composition view after the internship and before the start of the thesis (figure 1).

![Diagram](image)

Figure 1: Component composition view before the thesis project

2.3 Design Sprint

As I was supposed to continue working on the project for my thesis and new interns wanted to start working on the project, we decided to have a Design Sprint in order to have a better understanding of the current state of the project. We created a design sprint team which included the new interns, previous interns and a supervisor from the company.

2.3.1 Defining goal and finding solutions

With the collaboration of supervisors at Gofore, during the design sprint, the team for design sprint evaluated the current version of the project to find out how the current existing functionalities of
the program could be improved and which new possible functionalities could be added to it. The design sprint was a one week sprint that had the following phases:

![Design Sprint Phases](image)

### 2.3.2 Decision Making

In the beginning of the sprint, the team had to define a final long term goal for our project. There was a session in which members were developing ideas by writing them down on sticky notes. Afterwards, each person voted for them by putting badges on them. The supervisor was the final decision maker and her vote had more impact compared with the other members. Votes related to supervisor are in blue and yellow. The bigger the badge sticker, the higher the value of the vote(Image 5).

![Voting process](image)
The goal with the most votes was: “In two years time we should have a VR meeting room that people are willing to use.”

Later on in the next phase, each team member had a short presentation called “The Lightning Demo” and presented a current existing technology which is similar to our project. The main goal of these presentations was to widen our horizons and gather ideas for the project. In this phase, after hearing and talking about different ideas, the team discussed the existing problems of the project and possible solutions for solving them.

2.3.3 Sketching

After gathering ideas, each team member came up with a sketch that could be a possible scenario for the new features that could be added to the project. Again we voted for sketches and the following sketch was the winner of the election(Image 6).

Image 6: Selected sketch
2.3.4 Prototyping

Then the team wrote a testing scenario for the sketch and made a prototype to test it and get some feedback from the users. Although the process of making the prototype were supposed to take only one day based on the sprint planning, it took us 3 days to make it. This is because the team decided to make a high fidelity prototype so that it seems more realistic to the user. Besides that, Prototyping in VR is not as easy as other kinds of prototyping like mobile app prototypes.

In the prototype, which was made by one of the new interns from our design sprint team, he tried to simulate functionalities like having a presentation in VR, Slack text chat and video call integration and Trello board integration. However, except for the presentation screen which showed different slides, other functionalities were not actually working. They were a presentation of what could be implemented in the future.

![Image 7: Screenshot of the prototype](image-url)
2.3.5 Testing

In the testing scenario, the tester were joined to a meeting room to see a user giving a presentation in the virtual environment. Tester could see the avatar of the presenter and also hear her voice and vice versa. The presenter could change the slides for the user as the presentation was going on.

The user who was giving the presentation was actually one of the team members of the design sprint. The tester was located in the VR room accompanied by one of the team members(Image 9, 10) and the presenter was located in a sperate room with the other team members. The rest of the members were watching and hearing the user with a camera installed inside the VR room. Team members wrote the positive and negative points that were observed from the user’s verbal and non-verbal behavior.
After the testing session, one of the team members held an interview with the tester and asked questions about his/her experience. In the next section the results of the testing is gathered.

2.3.5.1 Testing Results

This testing result report is gathered by Christopher Rauser, one of the interns at the company:
The users liked the tool when using it and were really positive about it. However, for future improvements of the product some viewpoints came up which are listed here, their listing does not reflect any valorisation of importance of the different points:

**Space/Environment:** The current environment felt big and cold to the user. A warmer environment would be appreciated. Also was the positioning of the user relative to the screen posing different issues. Through being far away from some screens, things on the screens became unreadable, and currently there was no option of moving closer to the screen to improve that. Also when walking they were not aware of the frontiers of where they walk, in usual VR applications, when moving and getting close to the edge of the environment, a blue line appears in front of the user. Different solutions to the closeness of the screens could exist to let the user walk around, or to introduce the teleport function in the environment.

**Presence:** Testers reported that they felt closer to the participants in the VR tool, compared to a normal video conference. Also they appreciated to see the number of people being in the meeting. What testers expressed as a wish to improve the product was to see actual emotions of the other participants. In an ideal world they said to see the actual face of the person.

**Interaction:** The tester really enjoyed having the pointer on the controller to point out things on the slides and other screens. At the same time the option to sketch things on the slides was missing for them. Something that was proposed was, to point the laser on the screen and draw directly on the screen. The user should be able to choose if the drawing stays on the screen permanently or only 5 seconds, as people are used to from slack screen sharing.
A tester said as well that he was missing something to do in moments when the presentation would not be that interesting, or to maybe pull up the trello/JIRA board to maybe go through the list of to dos etc. while listening. S/he proposed to have individual screens right in front of the user on which s/he could interact with.

The main positive points observed by team members were:

- Having avatars inside the meeting room
- Feeling like they are in the same place as the presenter
The idea of having a presentation in the VR environment was interesting for them.

The main negative points observed by team members were:

- No instructions for the users to know how to use the controller and interact with the environment
- The limited movement inside the environment
- Not having an avatar with a face and not being able to express emotions

3 Requirements

3.1 Requirements Specification

3.1.1 High-level Requirements

After the Design Sprint and prototype testing, the team could approximately understand which features are more important to the user. The team discussed all the possible high-level requirements that could be added to the project. Later on two of them which had a higher priority were chosen to be the main focus of this thesis.

The following high level requirements were obtained in a meeting after the design sprint:

1. The possibility of expressing non-verbal behaviour of the user. (e.g. facial expressions, hand and body movements)
2. Holding sprint review meetings with the ability to have access to kanban board (e.g. Trello, JIRA and so on)
3. Giving presentation in the meeting room on shared screen between all the users in the room.

Once the high level requirements were defined, the team prioritized them based on the goals of the project and with the help of supervisors and company’s priorities. Eventually, 2 of them were chosen for the master’s thesis work:
1. The possibility of expressing non-verbal behaviour of the user. (e.g. facial expressions, hand and body movements)
2. Holding sprint review meetings with the ability to have access to kanban board (e.g. Trello, JIRA and so on)

To begin with the thesis work, working on non-verbal behaviour had the highest priority. In different discussions with the supervisor at the university and the company a list of requirements in detail was provided.

### 3.1.2 Non-verbal Behaviour

#### 3.1.2.1 Lip Sync

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Lip Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>The lips of the user’s avatar should move when he/she speaks.</td>
</tr>
</tbody>
</table>
| Basic Flow:    | 1. User joins a meeting room  
                 2. User starts to talk and the lips start to move  
                 3. Other users can see the movements of the user’s lips |

#### 3.1.2.2 Expressing emotions

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Express emotions</th>
</tr>
</thead>
</table>
| Summary:       | User should be able to express the following emotions:  
                 1. Laugh  
                 2. Tentativeness  
                 3. Surprise  
                 4. Sadness |
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>Smile</td>
</tr>
<tr>
<td>6.</td>
<td>Anger</td>
</tr>
</tbody>
</table>

**Basic Flow:**

1. User joins a meeting room and is able to express any of the mentioned emotions
2. Other users should be able to see her/his emotions

### 3.1.2.3 Expression of Like

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Like Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>User should be able to show his interest to what the other speakers say with a like symbol.</td>
</tr>
</tbody>
</table>

**Basic Flow:**

1. User joins a meeting room
2. Like symbol appears on top of users head if he/she wants
3. Other users should be able to see the Like symbol.

### 3.1.2.4 Expression of Question

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Question Mark Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>User should be able to express that he/she has a question in a meeting.</td>
</tr>
</tbody>
</table>

**Basic Flow:**

1. User joins a meeting room
2. Question mark appears on top of users head if he/she wants.
3. Other users should be able to see the Question Mark symbol.
3.1.3 Trello Integration

The second part of the thesis is related to agile practices in VR. Based on the feedback we received during the design sprint, we came with the conclusion that a simplistic version of a Trello board inside meeting room would be useful for users during the sprint review meetings.

3.1.3.1 Trello board Demonstration

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Trello board Demonstration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>User should be able to see the trello board in the meeting room</td>
</tr>
<tr>
<td>Basic Flow:</td>
<td>1. User joins the meeting room</td>
</tr>
<tr>
<td></td>
<td>2. User should see the trello board with the existing lists and cards</td>
</tr>
</tbody>
</table>

3.1.3.2 Add a new card

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Adding cards to an existing trello board</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>User should be able to add a new card to the trello board</td>
</tr>
<tr>
<td>Basic Flow:</td>
<td>1. User joins the meeting room</td>
</tr>
<tr>
<td></td>
<td>2. User should see the trello board with the existing lists and cards</td>
</tr>
<tr>
<td></td>
<td>3. User should be able to add a new card to the existing lists</td>
</tr>
</tbody>
</table>

3.1.3.3 Remove a card

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Removing cards from an existing trello board</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>User should be able to remove a card from the trello board</td>
</tr>
</tbody>
</table>
Basic Flow:  
1. User joins the meeting room  
2. User should see the trello board with the existing lists and cards  
3. User should be able to remove a card from the existing lists

3.1.3.4 Modify a card

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Modifying cards in an existing trello board</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>User should be able to modify a card in the trello board</td>
</tr>
</tbody>
</table>

| Basic Flow: | 1. User joins the meeting room  
2. User should see the trello board with the existing lists and cards  
3. User should be able to modify a card in the existing lists |

4 Development Strategy

4.1 Agile Approach

The development of this project was done in an agile way since the very beginning. Both supervisors at the company and the university, decided to continue the same development strategy for the master thesis project.

In order to start implementation, firstly, the requirements that were already on a Trello kanban board, were prioritized. Every two weeks a meeting was held with the professor to track the progress of the project. There was a short demo presentation each week along with the review of the thesis document. Everything related to challenges in each sprint were discussed and if there was a problem in the process and help was needed, it was shared with the internal and external supervisors. Besides that, in the company, there were daily stand up meetings every day in which every person talked about the tasks they did the day before. If there was a problem or struggle(blockers), other team members helped each other to solve it.
4.2 Version Control

The source code of the project was kept on a Bitbucket repository that was provided by the company for this project. The development process were tracked on the Bitbucket repository. For the version control, Git was used and during the development, there were different branches related to each requirement:

- **AvatarEmotions**: Every commit related to the non-verbal behaviour functionality were pushed to this branch.
- **Trello**: Every commit related to the trello board integration functionality were pushed to this branch.
- **Emotion Test**: This branch was made to test the integration of Avatar Emotions in the project.
- **Master branch**: The fully functional version from the other branches were eventually pushed to this branch.
- **Emotion Experiment (Repository)**: This was a separate experimental sub-project that was made to test different functionalities of IBM Watson’s Speech to text and tone analyzer API.

4.3 Tools

4.3.1 Game Engine & programming language

There are many different game engines currently in the market but few of them support VR development. The two leading game engines that are currently being widely used by VR developers are Unity3D and Unreal Engine. Unreal engine is more focused on advanced applications and is not a suitable game engine for beginners. However, Unity3D is considered as a game engine for “everyone”. Many people working in different fields(e.g. Civil engineers or architects) use this game engine because it is more user-friendly compared with Unreal Engine. Because the project was not a big project and did not have advanced requirements, the preferred game engine was Unity3D(Image 11).
There are two options of programming languages in Unity3D (Javascript and C#). Since javascript for unity is now deprecated and there are less documentation and code examples on the internet about it, the team was more interested in using C#.

4.3.2 SDK & libraries

VR development in unity relies on some external dependencies that are SDKs and libraries that gives the possibility of VR development. Some additional assets and frameworks were also used in the project to facilitate the use of VR accessories. Utilized assets are as follows:

SteamVR SDK: An official library made by Valve that makes the development easier for all of the HTC VR headsets. It is currently free on the Asset Store and supports the HTC VR headsets. This SDK is needed to handle position tracking and anything related to VR, and for now there are no alternatives to it.

VRTK: It is a framework containing a collection of useful scripts, concepts and components that aid building VR solutions rapidly and easily. It covers a number of common solutions for user interactions (e.g. pointing, grabbing or touching objects), body physics, UI elements (e.g. buttons, scroll lists) and so on. A major benefit of VRTK is that it provides abstraction from the headset...
hardware. Therefore, switching between different vendors (e.g. HTC, Oculus or Google) is handled by the VRTK with minimum amount of setting.

**Photon Unity Networking Classic**: Photon framework is used for the network programming of the program. It is a unity asset that facilitates the complications of sending data over the network. It is more efficient compared with the deprecated network programming language of the Unity(UNet) itself.

However, the free version of Photon is being used for this project and it supports up to 20 players. For having more players, the premium version should be purchased.

Photon uses Cloud Servers and they can be accessed using the photon API. There are servers in the US, EU, and Asia. After getting an API-key, it is possible to connect to Photon servers and use its services.

The following concepts are used in Photon for network management:

1. **Lobby**: Users join the lobby when they connect to photon servers
2. **Player**: Every user that joins photon servers is considered as a player. Each player has a specific name.
3. **Room**: After joining the lobby, users can join different rooms depending on the meeting they want to participate in.
4. **RPC(Remote procedure calls)**: These calls are used to send data over the network to other users. These calls could be sent to different RPC Targets. (e.g. Other users or a specific user)

**Photon Voice**: Photon Voice is used in the project for sending the user’s voice over the network. It is a separate unity asset created by the Photon team.

**IBM Watson**: IBM Watson is a question answering computer system capable of answering the questions posed in natural language based on cognitive computing. Watson was named after IBM's first CEO, industrialist Thomas J. Watson.

The challenge of creating IBM Watson started in 2004 and in 2011 Watson competed Jeopardy! Against the best player Brad Rutter and won the clash.
Later on, in recent years the capabilities of Watson has been extended. It got evolved and now there are new services offered by IBM (e.g. Watson on IBM cloud) which offers different services like:

- **Compute**: Offers various computing resources, including bare-metal servers, virtual servers, serverless computing and containers, on which enterprises can host their workloads.
- **Network**: Provides cloud networking services, such as a load balancer, a content delivery network (CDN), virtual private network (VPN) tunnels and firewalls.
- **Storage**: Offers object, block and file storage for cloud data; Provides tools to manage and monitor cloud deployments.
- **Security**: Includes services for activity tracking, identity and access management and authentication; Provides SQL and NoSQL databases, as well as data querying and migration tools.
- **Analytics**: Offers data science tools such as Apache Spark, Apache Hadoop and IBM Watson Machine Learning, as well as analytics services for streaming data.
- **Artificial Intelligence (AI)**: Uses IBM Watson to deliver services such as machine learning, natural language processing and visual recognition.
- **Internet of Things (IoT)**: Includes the IBM IoT Platform, which provides services that connect and manage IoT devices, and analyzes the data they produce.
- **Mobile**: Enables a development team to build and monitor mobile applications and their back-end components.
- **Developer tools**: Includes a command-line interface (CLI), as well as a set of tools for continuous delivery, continuous release and application pipelines.
- **Integration**: Offers services to integrate cloud and on-premise systems, or various applications, such as API Connect, App Connect and IBM Secure Gateway.

For this thesis project, the Artificial Intelligence (AI) services were used to detect the tone of the user’s voice and also the speech to text service for capturing what the user says and converting it to text.
- **Tone Analyzer**: Analyzes text, such as email messages or tweets, or longer documents, such as articles or blog posts. Monitor social media to understand what customers are saying about a brand and to determine whom to target with specific messaging.
- **Speech to text**: Receive the user’s voice input and converts it to text.

**Trello API:**
Trello is a web-based application for organizing tasks. It can be used in many different areas. These days development teams who are using agile methodologies use this tool for organizing their work. In Trello, it is possible to make a kanban board and track the development process of a project. Trello provides a simple RESTful web API where each type of resource (e.g. a card, a board, or a member) has a URL that you can interact with. In the Trello integration part of the thesis, its API is used to show the trello board in the virtual environment to the user.

### 4.3.3 3D Modeling
For this project, Unity3D and Blender were used for modelling almost everything in the environment. Blender was chosen because it is a free open-source 3D tool for creating 3D models, animations, visual effect and etc. Moreover, there are many video tutorials for this tool and it is a good choice for beginners. Since we didn’t have experience with 3D modeling, this was a better option to use.

**Blender** is a free open-source 3D computer graphics software tool used for creating animations, visual effects, art, 3D models and video games. In our case, Blender was used to create custom elements of the app (e.g. menu rectangles with smoothed angles or user avatars).

### 4.3.4 Constraints
**HTC Vive Headset**
A VR-Ready laptop with the following minimum functionalities:
1. Intel Core i5-6300HQ processor
2. Nvidia GeForce GTX 980, 1050 Ti, 1060, 1070 or 1080 GPU.
   RTX 2060, 2070 and 2080 are stronger alternatives.
3. 8GB of RAM
4. Two USB 3.0 ports
5. HDMI 1.4
6. 1920 x 1080 display
7. Windows 8 or 10
8. Having Unity3D and SteamVR installed

### 4.4 Project time & effort estimation

#### 4.4.1 Estimated Time

Before starting the project, a time estimation was done based on the prior experience of internship’s research and development effort. For the thesis, I was the only team member doing the tasks. The numbers are estimated assuming that 5 hours was spent each day on doing tasks.

<table>
<thead>
<tr>
<th>Project Summary</th>
<th>Module Name</th>
<th>Feature Name</th>
<th>Sub-Feature Item</th>
<th>Description</th>
<th>Development Estimate(m/d)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VR Conference Room</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td>Non-Verbal Behaviour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Research</td>
<td></td>
<td>Find out how to model 3D animations(facial expressions)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Design facial expressions</td>
<td>Modeling facial expressions in blender and animating them</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Activity</td>
<td>Description</td>
<td>Days</td>
<td>Min:</td>
<td>Max:</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Adding avatars to unity</td>
<td>Importing new avatars to unity and animating them</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research on triggering emotions</td>
<td>Finding different ways of triggering emotions</td>
<td>3</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation of triggering strategy</td>
<td>Implementation of facial expressions during the meetings</td>
<td>7</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trello Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research</td>
<td>Finding out different ways of integrating Trello in the project.</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trello board demonstration</td>
<td>Showing a trello board in the VR room</td>
<td>7</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trello board Interaction</td>
<td>Adding, removing or modifying cards</td>
<td>4</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speech to text</td>
<td>Writing card description using speech to text</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Days</td>
<td></td>
<td></td>
<td>Min:41</td>
<td>Max:67</td>
<td></td>
</tr>
</tbody>
</table>

4.4.2 Real Time
The real time spent on the project was closer to the min time than the max. The whole time spent on the features were as follows:
- Non-verbal behaviour: 25 days
5 Software Architecture

Before starting the implementation, based on the requirements and the results of the research in the previous phases, the design of the component composition view of the project evolved. In this figure, it is shown how different modules of the program communicate with each other and the data flows between them.

**Lobby**: Lobby is the term used for the main menu. It is where the user can customize his/her avatar, see the active meetings or create new meetings. The reason for choosing this name was that in the vocabulary of the Photon Network there is always a place called Lobby where user joins before joining or creating a room.
Once the user is in the lobby and defines the user name, avatar type, the room to join or create, the data will be sent to the Photon framework and after that the user will be redirected to the meeting room. All of the information related to room, user and the kind of request(join/create room) must be sent to Photon before joining/creating a room.

- **Meeting Room:** Once the user sent the needed data to Photon, the data will be processed and if it is the right form of data, user will be joined to the room without any problems. When the user joins the room, every data related to user movements, voice data and user preferences(including t-shirt color, name, hairstyle, hair color) will be sent to Photon. Then it will be broadcasted to the other users in the room.

- **Trello API:** All the information related to the Trello board is processed by the Trello API. Each user sends a request to the Trello API when he/she wants to see the Trello Board in the room. The requests are sent using https protocol. Trello API will reply to the user’s request once it is received. Every request related to adding, editing or removing cards will be sent to the Trello API using the same process. If user wants to add a card description using voice instead of the VR keyboard, a request will be sent to the IBM Watson’s API and once it is processed the text of the speech is sent to the user in the meeting room and it will be written automatically in the description text box.

- **IBM Watson API:** The process of analyzing what user says and detecting emotions are done by IBM Watson. When the user speaks, all the voice data is transmitted to the IBM Watson’s Tone Analyzer. The data will be analyzed by the Tone Analyzer and the results of the emotions detected in the tone are sent back to the user in the meeting room.(Figure 3) Then the corresponding emotion will be triggered based on the IBM Watson’s results.
6 Non-Verbal Behaviour Feature

As it is already mentioned in the section of introduction, non-verbal behaviour is one of the most important factors of communication. Non-verbal behaviour includes facial expressions, body movement and posture, gestures and eye contact. The most important part of the non-verbal communication is the facial expression in a face to face communication. For this reason, the main focus in this thesis is on the facial expression.

In order to learn more about the facial expression and how it can affect the communication, I did research to see how it can be simulated in such environment. There are many different factors that can affect the believability of avatars and also the feelings of immersion and presence in the virtual environment. In the following sections, I explained in detail, the difference between immersion and presence. Additionally, The differences between realism and believability of avatars and how it influences the VR experience are also explained.

6.1.1 Difference between Immersion and Presence

In the vocabulary of Virtual Environments, there is a difference between the terms immersion and presence.

The difference between immersion and presence is a way of measuring the human experience in such environments from an objective and subjective point of view respectively. Immersion,
defined in technical terms, is capable of producing a sensation of Presence, the sensation of being there (part of the virtual environment), as regards the user (Ijsselsteijn & Riva, 2003). The more that a system delivers displays (in all sensory modalities) and tracking that preserves fidelity in relation to their equivalent real-world sensory modalities, the more that it is 'immersive' [4].

Presence, on the other hand, is more of a subjective matter. It is a psychological, perceptual and cognitive consequence of immersion. Presence is thought of as the psychological perception of "being in" or "existing in" the VE(Virtual Environment) in which one is immersed[4-8].

6.1.2 Realism vs. Believability of Avatars

At the beginning of the project(during internship), the possibilities for choosing different kinds of avatars were investigated. During the first few sprints, simple cubes with a smiley face were used(Image 12) to represent the user because the appearance of the avatar was out of the scope of the project at that point.

Image 12: Cubic Avatars

Later when new avatars were chosen for the project we had to find out whether it is better to use cartoonish 3d models or realistic avatars. A collection of possible avatars that could be used were provided by us(interns) during the internship and finally two human-like avatars were chosen. Users could change the t-shirt color and hairstyle of the users. However, these avatars did not have a face and later during the thesis project, faces were added to their heads.
The other options that was also considered for creating avatars was AvatarSDK. It is one of the best 3D face simulators that also simulates the facial expression. But for some reasons which will be discussed in the next section it was not used for generating avatars.

6.1.2.1 AvatarSDK

Avatar SDK seemed like a good option for generating a real avatar based on user photos. In this screen shot(Image 13) from Unity3D, you can see a 3d model generated by Avatar SDK from a photo.

Avatar SDK creates realistic avatars from photos but the most important drawback of this SDK is that it doesn’t create the torso and hands for the avatars. Additionally, a flying head in the meeting room even with believable facial expressions and eye gaze was giving a feeling of alienation to
the user. Based on some research which will be discussed in the next section, we decided to use cartoonish avatars because in some cases simpler characters might even be more adequate and appreciated by the users[9].

6.1.2.2 Uncanny Valley

The issue of alienation of realistic human-like avatars to the users, was first discussed in an article by Masahiro Mori. Mori came up with the term “Uncanny Valley”. He predicted that a computer generated face looks the eeriest when it looks nearly human.

Masahiro Mori proposed a relation between human likeness and shinwakan, which may be roughly translated as rapport or comfort level: more human-looking robots are perceived as more agreeable until we get to robots that look so nearly human that subtle flaws make them look creepy. This dip
in their evaluation is the uncanny valley. The valley, Mori argued, would be deepened by movement. The term uncanny valley is now commonly applied to animated characters in films and video games.[10]

One reason of this kind of feeling could be that studies show the greater realism of the character the greater the perceived expectation of intelligence.[11] Eventually, we decided to choose a cartoonish avatar because it was more “believable” despite the fact that it was not very “realistic” compared to the avatars generated by Avatar SDK.
6.2 Implementation

6.2.1 Facial Expressions

After doing research and finding the best possible ways of creating the facial expressions. The process of creating expressions started with reimporting the 3d models of avatars in Blender. For creating facial parts(eyes, mouth, eyebrows and so on) and facial expression animations only Blender was used.

After creating different parts of the face, including eyes, eyebrows and mouth, animations were made using the **blend shape** technique of the Blender.

This technique gives the modeler the ability to deform the current shape of the meshes of a model. The location of these positions range from 0 to 100. Meaning that 0 is the starting point of the animation and 100 is the completion point of the animation. Using this technique, the following animations were added to the face parts(Image 16):

- Laugh
- Tentativeness
- Surprise
- Sadness
- Smile
- Anger
- Lipsync

![Facial Expressions](image16.png)
6.2.2 Different Solutions

After creating the facial animations, the main challenge was triggering them. The team thought of the following solutions:

1. Detecting real facial movements with a webcam and then convert them to virtual animations
2. Triggering animations with buttons on the controller
3. Triggering animations using hand gestures (e.g. User raises his/her hand and the laugh animations triggers)
4. Triggering animations using the sentiment analysis of what the user says

Each solution had its own pros and cons. Solution number 1 was not useful because the user’s face is covered with the VR headset and it is not possible to detect facial expressions. Solution number 2 was the most straightforward way of triggering the animations but if the user forgets to trigger it using the emotion buttons on the controllers, the face of avatar will not change and sometimes it might not be a rapid and efficient way of showing facial expressions to go through the emotion menu and choose the emotion by clicking on a button. However, this feature was provided to complement solution number 4 which is explained later in this section. In case the sentiment analysis doesn’t work or if the user wants to express emotions without talking.
In this image(Image 17) you can see the menu provided around the controller in the virtual environment for the user to express emotions.

![Image 17: Controller menu for facial expression](image)

In this solution, users can choose different emotions by moving their fingers on the touchpad of the HTC Vive. The color of the buttons change based on the position of the user’s finger on the touchpad. By pushing the touchpad the chosen button is clicked. The question mark button is used when the user has a question but doesn’t want to interrupt others. By pushing that button, a question mark will appear on top of user head’s avatar. The like button works likewise but it is for showing the approval to what other users are saying. It’s also a kind of non-verbal behaviour that is expressed using symbols.
Solution number 3 was also not a good option because the user could mistakenly move his/her hands in a way that it mistakenly triggers an animation. It is also not efficient to move your hands in different directions for a simple facial expression. Additionally, for an accurate hand gesture detection, Leap Motion(Image) was needed which would add +90 euros to the accessories needed for the setup for each user.

Solution number 4 seemed like another workable solution to go forward with, because it did not need any external gadget like Leap Motion. In this case, IBM Watson’s Tone Analyzer and Speech to text tools were used. When the user speaks, words will be detected by IBM Watson’s Speech to
Text and saved as a string. Then it will be compared locally with a list of words related to emotions to check if the sentence contains any words related to emotions and is worth sending to the Tone Analyzer’s API. This is for reducing the number of requests sent to the API. If local check detects emotions in the sentence, it will be sent to the Tone Analyzer. Then it returns a JSON string containing the strength of different emotions in the sentence. The JSON string is later processed in the program and the emotion with the most intensity(highest number) will be counted as the user’s emotion. Then the related animation will be triggered on Avatar’s face. However, IBM Watson’s text to speech might not work very accurately with people who have a distinct English accent or don’t speak English.

The process of analyzing the emotions of the user’s message by IBM watson is explained in the next figure(Figure 5).

![Figure 5: The flow of analyzing tone](image-url)
This is what Watson’s Tone Analyzer sends to the user based on the input which was the following sentence for this case:

“I hate you.”

```
{
  "document_tone": {
    "tones": [
      {
        "score": 1,
        "tone_id": "anger",
        "tone_name": "Anger"
      },
      {
        "score": 0.931034,
        "tone_id": "fear",
        "tone_name": "Fear"
      },
      {
        "score": 0.916667,
        "tone_id": "sadness",
        "tone_name": "Sadness"
      }
    ]
  }
}
```

As it is shown in the JSON output the score of the feeling of “anger” is the most intense feeling comparing with the other feelings which are “fear” and “sadness”. Consequently, the program will choose the feeling of “anger” as the dominant feeling and triggers the related animation.
7 Agile Practices in VR

7.1 Agile Methodology

Nowadays, Agile Methods are probably the most popular approach to project management among IT companies. This method is very useful for coping with the unpredictable occurrences that can happen during the project. It is also very suitable for changing and pivoting the project construction. This method uses iterative work sequences that are known as sprints.

According to a report by Project Management Institute[13], almost three-quarters (71%) of organizations report using Agile approaches in their projects sometimes, often, or always. Some of the benefits of agile methodology are:

- High product quality
- Higher customer satisfaction
- Increased project control
- Reduced risks
- Faster ROI(Return on investment)

Statistics say that by using agile project management, on average, the time to market is 37% faster and the efficiency of you team is increased with a productivity higher by 16% that the average.

However, in each iteration of the agile development, there are different meetings like sprint review meetings, daily standup meetings and so on in which team members discuss different matters about the project, plan for the next iteration and check the current progress of the work.

There are different lists like the backlog list and kanban boards with which teams organize all the tasks of a project. To manage all of those lists and tasks and bring them to the digital environment, teams use different tools like JIRA or Trello.

Nevertheless, with teams that are working remotely, handling these meetings and tasks can be more challenging and time consuming. In a face to face conversation everything is faster and swifter. But when it comes to the remote communication, each person should pay more attention in the meetings and usually some details are missed. In the second part of the thesis project
development, I evaluated the potential of VR for handling Sprint Review or daily stand-up meetings in a virtual environment. As it is mentioned before, one of the most important aspects of these meetings is the ability to see tasks and being able to organize or reorganize them. For this reason, the integration of a task management tool with the virtual environment was essential. Trello was the tool that has been used for our own project development management from the very beginning of the project.

![Trello board used for the development sprints](image)

### 7.2 Trello Integration

To start with bringing Trello to the VR world, we thought of creating a simple kanban board with three lists of To-do, Doing and Done to simplify the board. This is because showing a big board with a lot of details and functionalities was no feasible to be implemented in the short time of the thesis project. Besides, in the VR world, interacting with everything is different from the 2D web-based applications. The whole process of adding cards, removing them, editing them and specially writing their titles and descriptions is not as easy as the web version for the user. To make the interaction more user-friendly, the functionalities were limited. For this purpose, A separate board was created only for testing in VR (Image 19).
7.3 Showing the Trello Board in VR

In the first phase of the implementation, a very simple 3d model of a board was designed with the color palette that was chosen for the project during the internship. The board automatically sends a request to the Trello API when the game starts. The cards will be generated accordingly after receiving the data from the Trello API.

Unfortunately, the JSON parser framework for C# in unity could not parse the JSON sent from the Trello api. For the reason that the API parser of C# in unity is made for very basic JSON files and http requests, it couldn’t convert the JSON format to a list. To solve this issue, I had to manually add some classes to the current JSON parsing framework of the unity to be able to parse it. After solving this issue I could manage to show the cards of the trello board in the Virtual Environment.

Here is a sample of the format that the lists including cards are sent from the Trello API:

```json
[
    {
        "id": "5cda88b7e39827f73c94f01",
        "name": "Todo",
        "closed": false,
        "idBoard": "5cda88adb2cbde684f4931b9"
    }
]
```
"pos": 65535,
"subscribed": false,
"softLimit": null
}

![Image 20: VR Trello Board](image)

### 7.4 Adding, Removing and modifying cards

In the next step, the functionalities related to adding, removing and modifying cards were added.

#### 7.4.1 Adding cards

Users can add a card to the board by clicking on the plus icon on the bottom of each list. After clicking on the “+” icon, a new window will appear in front of the board in which the user can type the card’s Title and description.
Writing in the VR environment is not easy and it takes a lot of time and effort for the user to type a very short sentence. This is because the user has to click on each letter one by one using the laser that comes out of the controller to write something. We decided to add the possibility of writing the description of the cards with IBM Watson's Speech to text.

Eventually, there are two ways for writing the cards description. User can use the VR keyboard(Image 22) or the speech to text capability of the program. Once the user clicks on the microphone icon on the right bottom of the text box, a red circle will appear next to it which indicates that the program is hearing what the user says (Image 23).

![Image 21: Create card Window](image-url)
7.4.2 Removing cards

Users can remove a card by clicking on the bin icon on the right bottom corner of the editing card window. After removing a card the editing window will be closed and the card lists will be refreshed and the card will get removed.

7.4.3 Editing cards

When the user clicks on an existing card on the board, the description of that card will be shown to the user. Same as the creating card process, there are two ways to write the description of a card, with the virtual keyboard or with the speech to text service.
8 Testing

8.1 Non-Verbal Behaviour

The testing of the emotions expression was divided in two parts. In the first phase, the functionality of expressing emotions using tone analyzer was tested. Afterwards, expressing emotions using the menu on the controller was tested. Users were asked to follow the instructions as explained in the tests steps and test data.
### 8.1.1 Testing of expressing emotions using tone analyzer

<table>
<thead>
<tr>
<th>ID</th>
<th>Test Scenario</th>
<th>Test Steps</th>
<th>Test Data</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Pass/Fail</th>
</tr>
</thead>
</table>
| 1  | Check voice to speech                | 1. User starts to speak  
2. Check logs to see the text of what user said | User’s voice                                   | Written text of what user said in the microphone in the logs                  | Written text of what user said in the microphone in the logs                  | Pass ✔️   |
| 2  | Check Lipsync                        | 1. User starts to speak  
2. See if the user’s lips start to move | User’s voice                                   | User’s lips moving when user is speaking and stops moving when the user doesn’t talk | User’s lips move when user is speaking                                       | Pass ✔️   |
| 3  | Check emotion expression (laugh)     | 1. User starts to speak  
2. See if the laugh emotion triggers. | User says: I am very happy about it.          | Happy emotion triggers on the user’s avatar                                    | Happy emotion triggers on the user’s avatar                                   | Pass ✔️   |
| 4  | Check emotion expression (Tentativeness) | 1. User starts to speak  
2. See if the Tentativeness emotion triggers. | User says: I doubt it.                        | Tentativeness emotion triggers on the user’s avatar                           | Tentativeness emotion triggers on the user’s avatar                           | Pass ✔️   |
| 5  | Check emotion expression (Sadness)   | 1. User starts to speak  
2. See if the Sadness emotion triggers. | User says: This is very sad.                  | Sadness emotion triggers on the user’s                                       | Sadness emotion triggers on the user’s                                       | Pass ✔️   |
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 6 | Check emotion expression (Smile) | 1. User starts to speak  
2. See if the Smile emotion triggers. | User says: This is nice. | Smile emotion triggers on the user’s avatar | Smile emotion triggers on the user’s avatar |
|   |   |   |   | Pass | ✓ |
| 7 | Check emotion expression (Anger) | 1. User starts to speak  
2. See if the Anger emotion triggers. | User says: I hate this. | Anger emotion triggers on the user’s avatar | Anger emotion triggers on the user’s avatar |
|   |   |   |   | Pass | ✓ |
| 8 | Check laugh emotion with a long sentence. | 1. User starts to speak  
2. See if the laugh emotion triggers. | User says: I think talking about this topic could be nice. | Laugh emotion triggers on the user’s avatar | Laugh emotion did not trigger on the user’s avatar |
|   |   |   |   | Fail | ✗ |
| 9 | Check tentativeness emotion with a long sentence. | 1. User starts to speak  
2. See if the tentativeness emotion triggers. | User says: I think we should talk about this topic but I am not really sure what to discuss exactly. | Tentativeness emotion triggers on the user’s avatar | Tentativeness emotion did not get triggered on the user’s avatar |
|   |   |   |   | Fail | ✗ |
| 10 | Check surprise emotion with | 1. User starts to speak  
2. See if the surprise | User says: When you told me about | Surprise emotion must not get triggered on the | Surprise emotion triggers on the |
<p>|   |   |   |   |   |   | Fail | ✗ |</p>
<table>
<thead>
<tr>
<th></th>
<th>Check smile emotion with a long sentence.</th>
<th>Check anger emotion with a long sentence.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a long sentence.</td>
<td>a long sentence.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>emotion does not trigger.</td>
<td>emotion does not trigger.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>that matter I was shocked.</td>
<td>user’s avatar just because the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sentence contains the word “shocked”.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>And it refers to past.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. User starts to speak</td>
<td>1. User starts to speak</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. See if the smile emotion triggers.</td>
<td>2. See if the anger emotion triggers.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User says: This is not perfect but I</td>
<td>User says: I have talked about this to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>like it.</td>
<td>you and it really made me feel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>terrible.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Smile emotion triggers on the user’s</td>
<td>Anger emotion triggers on the user’s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>avatar</td>
<td>avatar</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Smile emotion did not get triggered on</td>
<td>Anger emotion did gets triggered on the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the user’s avatar.</td>
<td>user’s avatar.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>❌</td>
<td>✅</td>
<td></td>
</tr>
</tbody>
</table>

The tests for expressing emotions were passing when simple and short sentences were given to the analyzer. But when longer and more complicated sentences were given to the program by the user, the results were not as expected. Therefore, I decided not to integrate this functionality with the master branch because it was not working with enough accuracy. This functionality was developed on a separate branch called “EmotionTest” integrated with the main program. Besides this branch, a separate repository(Emotion Experiment) was made specifically for testing emotions and the integration of the speech to text with the tone analyzer API. Besides these test cases, more testing was done with random sentences to check the performance of the tone analyzer. For example, speaking for 2 or 3 minutes continuously and check how does
the api react to sending requests to the IBM servers. In conclusion, the more data given to the tone analyzer, the less accuracy it had.

Another problem which was detected during the testing, was that microphone on the HTC Vive’s headset is not sensitive and powerful enough to detect the user’s voice. To handle that problem the microphone on the laptop were used, which is more sensitive than the one on the HTC Vive.

8.1.2 Testing of expressing emotions using controller

This part of the testing was focused on testing the menu which was designed for the controller(Image 17).

<table>
<thead>
<tr>
<th>ID</th>
<th>Test Scenario</th>
<th>Test Steps</th>
<th>Test Data</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check emotion expression(laugh)</td>
<td>1. User joins the room 2. Clicks on the laugh icon from the emotions menu</td>
<td>Emotions menu</td>
<td>The laugh emotion triggers on the avatar</td>
<td>The laugh emotion triggers on the avatar</td>
<td>Pass ✓</td>
</tr>
<tr>
<td>2</td>
<td>Check emotion expression(Sad)</td>
<td>1. User joins the room 2. Clicks on the sad icon from the emotions menu</td>
<td>Emotions menu</td>
<td>The sad emotion triggers on the avatar</td>
<td>The sad emotion triggers on the avatar</td>
<td>Pass ✓</td>
</tr>
<tr>
<td>3</td>
<td>Check emotion expression(Smile)</td>
<td>1. User joins the room 2. Clicks on the smile icon from the emotions menu</td>
<td>Emotions menu</td>
<td>The smile emotion triggers on the avatar</td>
<td>The smile emotion triggers on the avatar</td>
<td>Pass ✓</td>
</tr>
<tr>
<td>4</td>
<td>Check emotion</td>
<td>1. User joins the room</td>
<td>Emotions</td>
<td>The anger</td>
<td>The anger</td>
<td>Pass ✓</td>
</tr>
<tr>
<td></td>
<td>expression (Emotion)</td>
<td>action 1</td>
<td>action 2</td>
<td>menu</td>
<td>emotion trigger 1</td>
<td>emotion trigger 2</td>
</tr>
<tr>
<td>---</td>
<td>---------------------</td>
<td>----------</td>
<td>----------</td>
<td>------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>5</td>
<td>Check emotion expression (Surprise)</td>
<td>User joins the room</td>
<td>Clicks on the surprise icon from the emotions menu</td>
<td>Emotions menu</td>
<td>The surprise emotion triggers</td>
<td>The surprise emotion triggers</td>
</tr>
<tr>
<td>7</td>
<td>Check emotion expression (Like symbol)</td>
<td>User joins the room</td>
<td>Clicks on the like icon from the emotions menu</td>
<td>Emotions menu</td>
<td>The like symbol appears on top of the user’s head</td>
<td>The like symbol appears on top of the user’s head</td>
</tr>
<tr>
<td>8</td>
<td>Check emotion expression (Question symbol)</td>
<td>User joins the room</td>
<td>Clicks on the question icon from the emotions menu</td>
<td>Emotions menu</td>
<td>The question symbol appears on top of the user’s head</td>
<td>The question symbol appears on top of the user’s head</td>
</tr>
</tbody>
</table>
# 8.2 Trello Integration Testing

## 8.2.1 Testing of Trello Board

The second part of the testing is related to the Trello Board. The test cases are as follows:

<table>
<thead>
<tr>
<th>ID</th>
<th>Test Scenario</th>
<th>Test Steps</th>
<th>Test Data</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Pass/ Fail</th>
</tr>
</thead>
</table>
| 1  | Check Trello Board demonstration            | 1. User joins the room  
2. Clicks on trello board button                                                | Trello board | Trello board must appear in the room                                              | Trello board appears in the room                                               | Pass ✓     |
| 2  | Check Trello Board cards demonstration      | 1. User joins the room  
2. Clicks on trello board button                                                | Trello board | Trello board must appear in the room and load the cards | Trello board appears in the room and loads the cards | Pass ✓     |
| 3  | Check adding cards to trello board          | 1. User joins the room  
2. Clicks on trello board button  
3. Clicks on the plus button                                                      | Trello board | Menu for adding cards must appear and user should be able to add the card title and description and save it. | Menu for adding cards appears and user adds the card title and description and saves it. | Pass ✓     |
| 4  | Check removing cards from trello board      | 1. User joins the room  
2. Clicks on trello board button  
3. Clicks on one of the cards  
4. Clicks on the bin icon                                                          | Trello board | The card must be deleted.                                                          | The card gets deleted.                                                          | Pass ✓     |
<p>| 5  | Check editing                               | 1. User joins the room                                                      | Trello      | Card description                                                                  | The card                                                                    | Pass       |</p>
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Check speech to text for writing card description</td>
</tr>
</tbody>
</table>
| cards in trello board | 2. Clicks on trello board button  
3. Clicks on one of the cards  
4. Clicks on card description and types new description using the virtual keyboard | board | must be changed | description changes. | ✓ |

### 9 Conclusions, Challenges and Future Work

#### 9.1 Conclusions

This master project gave me the opportunity to get first-hand experience with the trendy topic of the virtual reality and its distinctive kind of development and interaction. During these 4 months I worked with different technologies like IBM Watson services, VRTK, Trello API and so on that I was not acquainted before. Working on this domain gave me a different perspective about the future of virtual reality and human and computer interaction.
Besides, the working atmosphere and the culture of the company (Gofore) was very encouraging to me. My supervisors were very kind and helpful to me all the time. They gave me the always chance to make my own decisions and guide me when it was necessary.

9.2 Challenges and Future Work

Although the VR development is fun and it is a new technology that has a lot of interesting sides to be discovered, but there were some challenges during the time I was working on the project. They are listed as follows:

- The whole 3D design of the environment could be improved. This is because there were no experts in 3D modeling at the company and I had to learn everything related to modeling little by little during the time I was working on my master thesis and doing my internship.
- The whole development of VR is still novice and the game engine and the frameworks have bugs that might not work accordingly all the time. There are new updates every day and developers have to adapt themselves with these new technologies all the time. Once updates related to unity and Steam VR caused some problems in the whole project that we were spending two days to fix them.
- There are not enough documentation and learning materials related to VR and sometimes you have to open new discussions on forums and slack channels because other people didn’t have those problems yet. Although in a web development the case is different and because it has been popular for years, It is easier to find solutions for the problems.

After finishing with the project there are still different features that could be added or improved. Here you can find a list of all the possible functionalities that could be added to the project:

- Improving the quality of 3D models (e.g. avatars, tables and the meeting room)
- Adding Screen Sharing feature with which users can share the screen of their computers in the virtual environment for giving presentations or showing a part of their code for code reviews.
- Improving the Google Daydream version
- Adding more user instructions so that users understand how to use the program because it is a new kind of environment that everyone feel alienated at first.
• A wider range of avatars and higher customization (e.g. changing eye color)
• Improving the features supported by Trello API like adding due date of a task, showing bigger boards, creating checklists and adding attachments to cards.
10 References


[9] Monteiro, Diego et. al., Evaluating the Effects of a Cartoon-Like Character with Emotions on Users' Behaviour within Virtual Reality Environments


[12] Leap Motion, https://www.leapmotion.com/