A Project Report on

Using AR/VR For Shopping

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Jaaie Kadam(18104017)
Prachi Manera(17104013)

Under the Guidance of

Prof. Anagha Aher

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2021-2022

Approval Sheet

This Project Report entitled ***"Using AR/VR For Shopping"*** Submitted by ***"Jaaie Kadam"(18104017),"Prachi Manera"(17104013)*** is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Anagha Aher
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Using AR/VR For Shopping"*** submitted by ***"Jaaie Kadam" (17104013),"Prachi Manera" (17104013)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2021-2022.

(Prof. Anagha Aher)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **Using AR/VR For Shopping.** We take this opportunity to express our sincere thanks towards our guide **Prof. Anagha Aher** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name: Jaaie Kadam
Student ID: 18104017

Student Name: Prachi Manera
Student ID: 17104013

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

Jaaie Kadam (18104017)

_____

Prachi Manera (17104013)

Date:

Abstract

The COVID-19 virus outbreak began in December 2019 and rapidly spread to every continent on Earth. The analysts have predicted that COVID-19 and other similar pandemics will continue in the coming decade and badly affect offline businesses. As a result, the offline platform is also shifting to the online platform and online demands are increasing daily. The traditional twodimensional E-Commerce websites are designed to provide simple, browser-based interfaces to allow users to access available products and services. Whilst virtual representations are an essential consideration in establishing trust, most virtual representation sites fall short in mimicking real-life human representation. The primary goal of V-Mart's design is to create a shopping mall that accurately represents a "physical, brick-and-mortar" store. Retail items can be picked up for closer inspection by the econsumer by clicking a button and then returned to their original positions, thereby fulfilling the Selection aspect of a good customer experience. In V-Mart, we provide all product information when the product is hovered over, allowing e-consumers to make a more informed decision. The e-consumer navigates the 3D store in real time by moving a simulated observer's viewpoint through the 3D environment and interacting with 3D animated objects such as retail items. VMart's ability to display images of a 3D environment smoothly and quickly enough creates the illusion of a real-time immersive "walkthrough," providing the e-consumer with an intuitive and natural 3D user interface for information exploration.

Contents

List of Figures

List of Tables

List of Abbreviations

AR:	Augmented Reality
VR:	Virtual Reality
V-Mart:	Virtual Mart
HMD:	Head Mounted Device
DSR:	Smartphone for VR

Chapter 1

Introduction

V-Mart is a virtual reality-based smartphone application that will allow users to experience the immersive-ness and lifelike quality of a typical brick-and-mortar shop. The World Health Organization (WHO) proclaimed the new pathogenic Covid-19 outbreak to be a Public Health Emergency of International Concern following the outbreak. Everything was shut down, and no one was permitted to be out in public for any purpose. When the government imposed social distance rules and a limit on the number of people permitted in shopping malls, consumers flocked to internet shopping since it was more convenient. But, when it comes to buying, the most essential component is "trust," which e-commerce websites do not provide since consumers do not have the opportunity to inspect the goods in person, which is a key factor in allowing the consumer to make an educated decision. As a result, this VMart solution was created, which combines the benefits and convenience of e-commerce websites with the in-store experience. Virtual reality (VR) refers to technologies for substituting the perceived reality. With the recent proliferation of consumer-grade head-mounted VR displays, several industries have started to wake up to the possible potential of virtual reality. One typical area in the early stages of the adoption of these technologies is marketing, and especially its sub-areas of retail and shopping. However, there has been a dearth in our understanding of how VR technology has been investigated in retail-related literature. V-Mart focuses on providing people with all of their monthly necessities. Customers that use VMart will be able to shop at their leisure, from anywhere and whenever they choose. Users may stroll through the mall, selecting and inspecting anything they desire to purchase in 3D. When the user selects the product, the screen will display all of the relevant information about the product. If the user does not want to spend too much time exploring, they can utilize the navigation or search functions. Users will be able to rate and express comments on their shopping experience and particular goods, and they will also receive suggestions based on their input and purchase history.

Ecommerce websites have a 2D display which convenient to browse, view and requires low maintenance costs. For example, Amazon, eBay etc. Such websites are not very complex and are easy to access and understand for the customers. But at the same time, they have unsatisfactory shopping experience which is not personalised as traditional commerce. Online shoppers have a hard time to grasp the product quality and other dimensions. Consumers lose the ability to interact with physical products, other users, and vendors. While, retail websites offer faster and easier transactions, it lacks the realistic shopping experience. Gap between Ecommerce websites and traditional shopping can be bridged through virtuality.

VR means a completely immersive computergenerated simulation in which the user can interact with Artificial 3D environment. An AR/VR enabled virtual shopping mall allows the actor (web customer/ online shopper) to use the website to make purchases online. A 3d mall allows the users to navigate through the 3d environment and get the look and feel of the product using 360-degree view to get a life like experience. Virtual malls can have many shops for clothing, groceries, shoe shops, book shops etc. The 3-D technology and virtualization systems are very attractive to users as [1] it is like video games. Making it appealing and enjoyable. A virtual mall with multiuser system, allows the user to go shopping with their peers, can chat and interact with vendors to negotiate for price etc making it more interactive. The plus point of Online Virtual shopping malls is that it can be used even during national/international lockdowns or calamities such as Covid-19 pandemic. Social distancing can be maintained using this system as the users need not physically move at all to the shops. They can shop or browse products anytime and anywhere. With AR/VR this experience would attract more users, especially the young generation.

Chapter 2

Literature Review

In 1987, G. M. Nielson and D. R. Olsen has published, "Direct manipulation techniques for 3D objects using 2D locator devices". [1] The methodology proposed is of the classic input devices, a Mouse and a Keyboard. The ability to map a 2d mouse interaction to a 3D space and the high degree of technological adoption makes this approach preferred by many beginners in VR. However due to the natural limitations of these devices such as their limited number of stated and necessity to use complex key combinations, navigation using such devices is often non intuitive and complicated.

In 2014, M. Roup, P. Bosch-Sijtsema, and M. Johansson has published, "Interactive navigation interface for virtual reality using the human body". [2] The methodology used is Xbox 360 Kinect Sensor System. The interface of the XBOX Kinect makes it possible to navigate in VR and has the potential to provide a more user-friendly and natural interface with the 3D-model. The Xbox Kinect (Microsoft, 2013). captures the depth information in the scene, and is used as a human body interactive interface for navigation and interaction in gaming environments. The movements, poses and gestures from the viewers' body are captured, encoded and translated into an event that triggers an action that controls something in the application. However it Requires too much space and it is very expensive. By far the most annoying part of the Kinect is the hand-over hovering menu system. While some of the games (such as Dance Central) omit this, the built in XBox menu interface has you hovering your arm over menu blocks for three seconds before anything happens.

In 2015, A. Kitson, B. E. Riecke, A. M. Hashemian and C. Neustaedter has published, "Navichair: Evaluating an embodied interface using a pointing task to navigate virtual reality". [3] The methodology used is of Specialized equipment like gaming input devices(e.g. joysticks and pads) and dedicated VR devices(e.g. tracked controllers and haptic arms). The idea was to integrate multiple input devices into the same simulation to create an experience where the user interaction felt natural and intuitive. The HTC Vive and Leap Motion were used as input devices. The HTC Vive was used to track the orientation and position of the user's head, while the Leap Motion was used to track the user's hands. However The Leap Motion added some limitations. The device driver requires that the Leap Motion is mounted at the front of the Virtual Reality headset, which was achieved by using double sided tape. The issue with the Leap Motion comes from the fact that, while it has an impressive 150◦ field of view, it is far too narrow to give real, functioning hand tracking for VR.

In 2015, J. J. LaViola, Jr has published, "Context aware 3D gesture recognition for games and virtual reality". [4] The methodology used is Gesture recognition. This technique focuses on providing an intuitive natural interface which is user friendly even for non-experienced users. However the problem with using these natural interactions to design content is that it requires a significant physical effort from the user. For example, the average time a user can comfortably use Leap Motion which is a device for gesture recognition is about 20 minutes. Thus, this technique is not suitable for designing VR environments, as itis often a process that requires a long time and high accuracy.

In 2015, S. Gebhardt, S. Pick, H. Voet, J. Utsch, T. al Khawli, U. Eppelt, R. Reinhard, C. Bscher, B. Hentschel, and T. W. Kuhlen has published, "flappassist: How the integrationof vr and visualization tools foster the factory planning process". [5] The methodology used is Context based approach. Contextbased approach is an interaction technique popular in computer games, in particular simulations for example The Sims and Simcity series by Maxis and adventure games. This approach is not in itself based on specific input devices but focuses on the use of available devices to navigate through a real-time contextual interface. The content of this interface depends on the current state of the environment and its objects for example time, position, current object state. The Context based approach is often used in modern VR environments. However this approach is uncomfortable due to the mismatch between classic user interface elements that is buttons, menus, charts and the 3D virtual environment. In addition this technique is best suited for a limited number of possible states and is not convenient for entering data such as text or numbers. This is a serious limitations when the interface is used for the content design.

In 2017, G. Cortes, E. Marchand, J. Ardouinz and A. Lcuyer has published, "Increasing optical tracking workspace of vr applications using controlled cameras ". [6] The methodology used is Marker tracking. Marker-based augmented reality experiences require a static image also referred to as a trigger photo that a person can scan using their mobile device via an augmented reality app. The mobile scan will trigger the additional content (video, animation, 3D or other) prepared in advance to appear on top of the marker. If the marker image is prepared correctly, marker-based AR content provides quality experiences and tracking is very stable, the AR content doesn't shake. When the mobile camera is moved away from the marker, AR experience disappears and the trigger photo has to be scanned again. It is possible to use extended tracking, but in most cases, extended tracking makes things worse. Scanning will not work if markers reflect light in certain situations (can be challenging with large format OD banners in ever-changing weather conditions).

In 2017, T. Piumsomboon, G. Lee, R. W. Linderman, and M. Billinghurst has published, "Exploring natural eye-gaze-based interaction for immersive virtual reality". [7] The methodology used is Eye tracking. One advantage of eye-tracking technology is it records actual eye movements. In consumer behavior research, reactivity (when participants change their behavior due to being observed) is a major concern. Eye-tracking technology reduces this concern because real eye movements are documented and consumers often forget their eye movements are being recorded. 6 Eye-tracking recordings accurately depict natural eye movement and fixations. Eye-tracking technology also provides flexibility in terms of research locations. One of the main disadvantages of eye-tracking technology is not all eyes can be tracked. Contact lenses, glasses, and pupil color can all impact the eye-tracking

camera's ability to record eye movements. Consequently, not everyone (typically 10– 20% of the sample) can participate in an eye-tracking study. As a result, the representativeness of the sample will be impacted. Eye-tracking studies also require considerable financial, time and labor resources Eye-tracking equipment (i.e., camera, computer, software) and training can be expensive. Additionally, only one person can be recorded at a time. Individual participation instead of group participation takes considerably more time and labor. The use of multiple eye tracking devices could help reduce the total experiment time, but could be more labor intensive.

In 2017, D. Zielasko, N. Neha, B. Weyers and T. W. Kuhlen has published, "A nonverbal vocal input metaphor for clicking". [8] The methodology used is Verbal/Vocal input. Convenience is undoubtedly the primary driving factor behind the rise of ecommerce. However, the sector has yet to figure out a way to make up for the lack of personal interaction consumers experience when selecting goods, asking questions, or making purchases. People naturally like to engage with a flesh and blood assistant or customer service agent at some point in the buying process. Once again, AI is coming to the rescue – this time, in the form of virtual shopping assistants. The AI-powered shopping assistant learns about your tastes and interests while shopping online, and then uses this data to inform you about products you might like to purchase, creating a virtual experience that's more like shopping in a virtual store. However its reluctance to understand and inability to hear due to various reasons like background noise,etc because it cannot differentiate between a person's voice and other background noises.

In 2018, J. Sokolowski and K. Walczak has published, "Semantic modeling of user interactions in virtual reality environments". [9] The methodology used is CAVE. Fully immersive CAVE virtual reality systems create collaborative environments ideal for design, engineering and simulation. In the automotive engineering and design fields, CAVEs allow teams to view exactly how parts fit with one another and interact directly with data for ergonomic tests. CAVEs can be particularly valuable by eliminating inefficient part placement and streamlining interior design. For all their value, the downsides of CAVEs must be weighed against their potential benefit. Classic CAVEs tend to be small, with a capacity of only a few people.

Chapter 3

Objectives

• To advance the current Shopping system by providing user an immersive experience.

• To understand the concepts of VR and Unity and develop a prototype using the findings.

• To create the database for the retail items, shops and customers, demonstrating their relationships through queries and functions on the user interface.

• To integrate the store user interface under one application, to facilitate accessibility and time saving.

Chapter 4

Project Design

4.1 Existing System

Brick-and-mortar shopping entails customers' physical presence and face-to-face interaction with retail items and employees. Though brick-and-mortar businesses have many advantages such as in-person interaction with the product, no risk of fraud, free delivery, consumers can take home products' the same day they bought it, and the most significant difference is that it is extremely difficult to replicate the experience these brick-and-mortar stores provide. However, there are a few limitations in the current system, which are as follows:

Existing system constraints:

Operating hours and timings: There are no stores that can remain open 24 hours a day, seven days a week. As a result, the customer must adjust their schedule and be available according to the store's operating hours.
Crowd: In times when social distancing is practiced, going to physical stores where people come from different cities and mass gatherings occur is extremely dangerous.
Delivery: No physical stores deliver to customers' homes.
Queues: Customers must wait in line for their turn to scan each item and pay, which takes a long time.
Size constraint: Products are limited due to store size constraints, which is why physical stores have limited products.
Shopper preferences: People of the younger generation are more technologically savvy, so they will always prefer online shopping over traditional shopping. This project aims to address the limitations of the current system by creating a solution that combines the immersive and lively experience of traditional stores with the additional functionalities of e-commerce from the comfort of one's own home.

4.2 Proposed System Architecture

4.2.1 User Block

Any new user will have the option of registering or browsing the mall and adding products to their cart without registering. If the user tries to check out, he or she must first register and log in to the app with their credentials in order to use any applicable coupons and place the

order. To login to the system, the user is required to fill email and password fields. While entering the password, the password will be masked using different characters and this is done using Nodejs package called bcrypt, which is used for hashing and decrypting passwords. During the registration procedure, the user's email address will be confirmed. New users will receive an email after successfully registering, and if they want to access through numerous accounts, they must use separate emails.

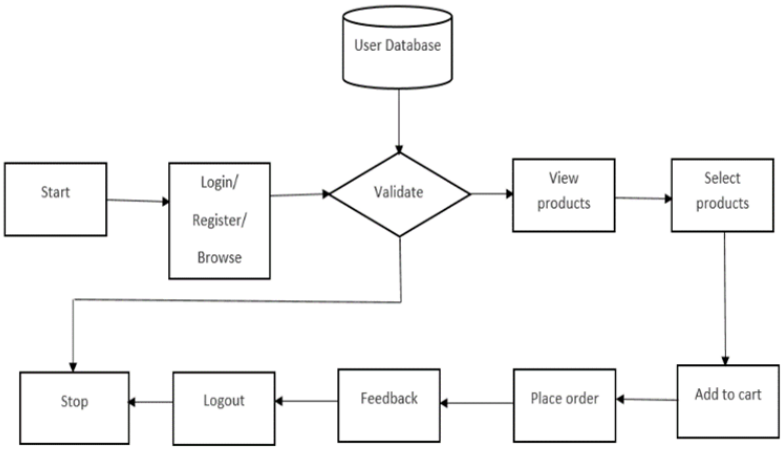

Figure 4.1: User end process

Figure. 4.1. illustrates the process followed by the system from the user's perspective. A profile must be created in advance by the user who wishes to shop. The profile will contain all of your personal information, such as your location, which will be used to determine whether or not the app's features are available to you. During the login procedure, the user's credentials will be checked against those stored in the database to see if they match. Once the person has been validated through this way, they are free to continue purchasing. With the help of a virtual reality headset, the user can browse multiple products and take a walk through a virtual mall. Users can add an item and control the quantity according to their preferences. Users can see the products they have added to their cart in my cart. Following that, the user can place an order. When it comes time to place an order, the user can see the total amount and choose any payment method. Users may view their orders in order history, and they can provide feedback on any product or the entire shopping experience.

4.2.2 Admin Block

Figure. 4.2. illustrates the process followed by the system from the admin's perspective. This system will be used by the administrator to track users and their activity, as well as manage the store's inventory. A database is used to track and update user profiles. The customer profile contains the e-personalized consumer's data as well as that of related individuals, such as people who share similar interest, so that V-Mart can offer the user the selections and information that may be of interest to him, as well as dynamically categorizing the relevant products to match each specific customer profile. The customer profile is immediately updated in the database as a result of user behaviors (e.g., shopping list creation).

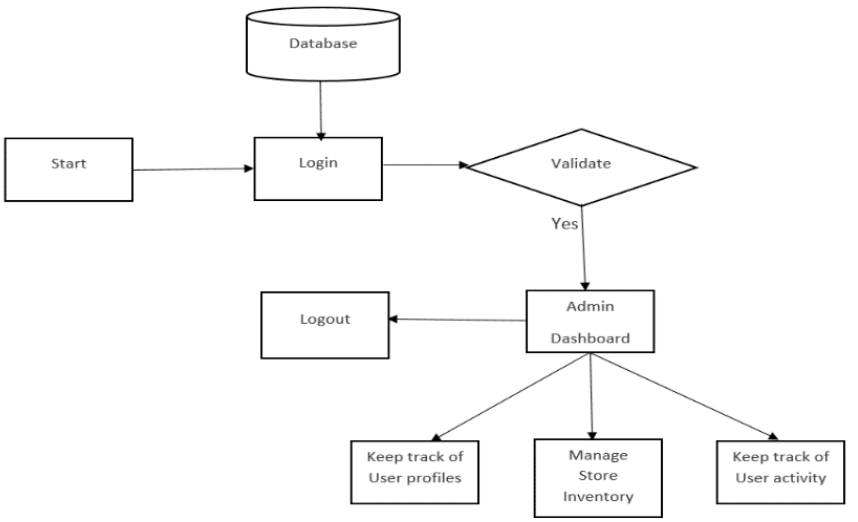

Figure 4.2: Admin end process

4.3 UML Diagrams

4.3.1 Use Case Diagram

In this project there are two modules. One is admin module and another one is user module. Admin can login using the web panel. Admin can keep track of the user and their shopping activities. Admin can keep track of store inventory information. Users need to first register themselves whatever information asked by this application. After that users can login using email id and password. User can see multiple products with the help of virtual reality. If the users want to buy any product, then they can buy the products using this application Users can buy the product as per their choice. Users can see their added products in my cart. After that user can place order. When it's time to place an order user can see the total amount and they have online payment option. User can see my orders history and see profile information. Users can give a feedback for particular product.

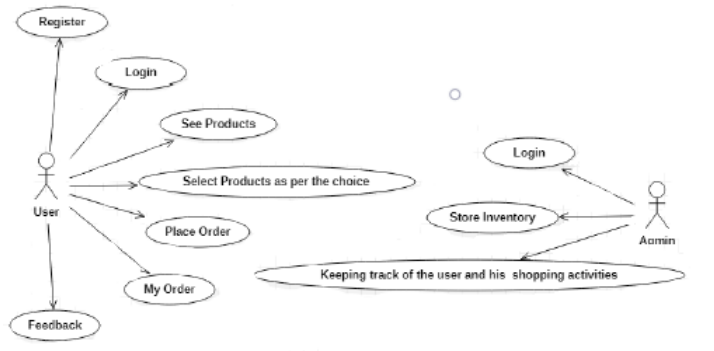

Figure 4.3: Use Case Diagram

4.3.2 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. Activity diagram is eesentially and advanced version of flow chart that modelling the flow from one activity to another activity. This is the more advanced version of the flow diagram which gives you the application flow is a more detailed format. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. In this activity diagram, we will discuss the login flow. The user will first log in to the system; if they are not already registered, they will need to do so. After logging into the system, the user can view the products, add them to the cart, and place an order after successfully paying. After placing the order, the user can review his order history and provide feedback.

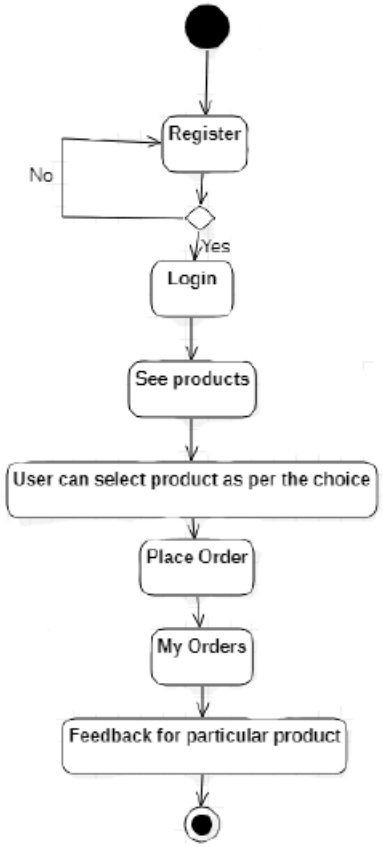

Figure 4.4: Activity Diagram

4.3.3 Class Diagram

Class diagram is a static diagram. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. In this class diagram, the class User is linked to class Login representing that he can login to view all products using the association relationship that encompasses just about any logical connection or relationship between classes, he can add the products to the cart procedding with placing a order and giving feedback.

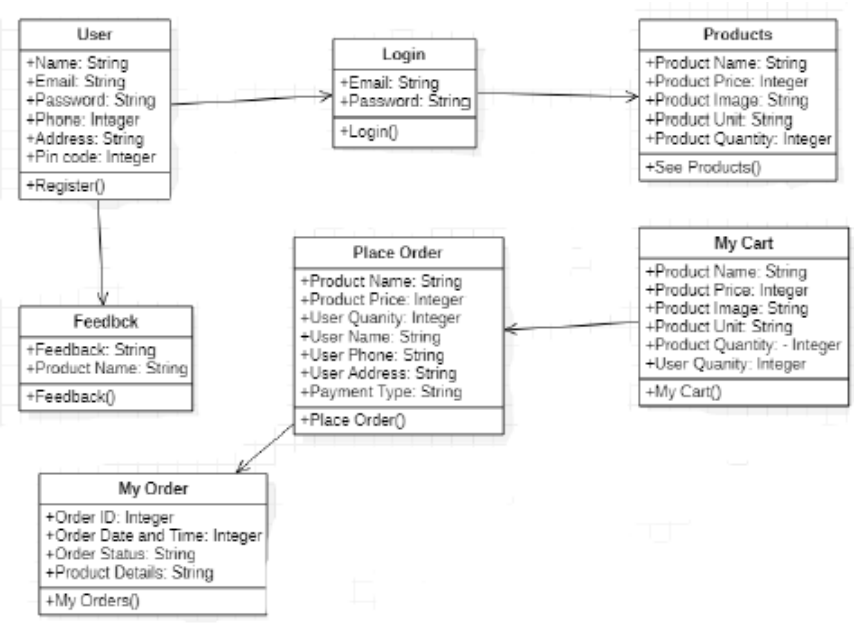

Figure 4.5: Class Diagram

Chapter 5

Project Implementation

5.1 Login

To access the system, the user must enter their email address and password. The password will be masked using different characters as you enter it, and this is done using the Nodejs package bcrypt, which is used for hashing and decrypting passwords. If the user forgets his password, he can reset it using the otp provided in the email. For the email service, we use Sendgrid.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using Lean.Gui;
// using UnityEngine.XR.Management;

public class LoginViewController : MonoBehaviour
{
    public InputField emailField, registerEmailField, registerNameField;
    public InputField passwordField, registerPasswordField;
    public Text errorText;
    public GameObject loadingContainer;
    public ClientAPI clientAPI;

    public LeanWindow errorModal;
    public GameObject loginContainer, registerContainer;

    void Start()
    {
        // StopXR();
        // Auto Login if token is present
        // PlayerPrefs.DeleteKey("AUTH_TOKEN");
        string userToken = PlayerPrefs.GetString("AUTH_TOKEN", null);
```

```
    Debug.LogError("userToken: " + userToken);
    if (userToken == "")
    {\newline
        // No auth Token found\
        loadingContainer.SetActive(false);
        Debug.LogError("No auth found");
    }
    else if (userToken != "") //
    {
        // Auth token found
        // Check with server
        StartCoroutine(clientAPI.CheckUser(userToken,
        OnCheckUserRequestComplete));
    }
}

public void Login()
{
    // Get email and password
    string email = emailField.text;
    string password = passwordField.text;
    Debug.Log(email);
    Debug.Log(password);

    Login loginData = new Login()
    {
        email = email,
        password = password
    };

    loadingContainer.SetActive(true);

    StartCoroutine(clientAPI.Login(loginData, OnLoginRequestComplete));
}

public void OnLoginRequestComplete(AuthToken authToken)
{
    loadingContainer.SetActive(false);
    if (authToken.token != null)
    {
        Debug.Log(authToken.token);
        PlayerPrefs.SetString("AUTH_TOKEN", authToken.token);
        PlayerPrefs.SetString("AUTH_EMAIL", authToken.email);

        loadingContainer.SetActive(true);
        UnityEngine.SceneManagement.SceneManager.LoadSceneAsync
        ("Main Menu");
    }
```

```
    else
    {
        // Show error modal
        string error = "<size=60>Oops!</size> \n \n" + authToken.error;
        //Show modal here
        errorText.text = error;
        errorModal.Toggle();
        Debug.Log(authToken.error);
    }
}

public void showLoginContainer()
{
    loginContainer.SetActive(true);
    registerContainer.SetActive(false);
}
```

5.2 Register

The user is created in this case with the help of the modal schema. To register, you'll need to provide information such as your full name, email address, and password. To ensure password confidentiality, the hashing algorithm is used here. The user enters a password, then a salt (random string) is added, followed by the hashing algorithm, which hashes the password and salt and stores it in the database. The password and salt are stored in the database so that when the password and salt are retrieved, the process is reversed or the password and salt are rehashed.

```
public void Register()
{
    // Get email and password
    string email = registerEmailField.text;
    string password = registerPasswordField.text;
    string name = registerNameField.text;
    Debug.Log(email);
    Debug.Log(password);

    Register registerData = new Register()
    {
        email = email,
        password = password,
        full_name = name
    };

    loadingContainer.SetActive(true);

    StartCoroutine(clientAPI.Register(registerData,
    OnRegisterRequestComplete));
```

```
}

public void OnRegisterRequestComplete(AuthToken authToken)
{
    loadingContainer.SetActive(false);
    if (string.Equals(authToken.status, "success"))
    {
        loadingContainer.SetActive(false);
    }
    else
    {
        // Show error modal
        string error = "<size=60>Oops!</size> \n \n" + authToken.error;
        //Show modal here
        errorText.text = error;
        errorModal.Toggle();
        Debug.Log(authToken.error);
    }
}
public void showRegisterContainer()
{
    loginContainer.SetActive(false);
    registerContainer.SetActive(true);
}
public void OnCheckUserRequestComplete(AuthToken authToken)
{
    if (authToken.error == null)
    {
        Debug.Log(authToken.token);
        loadingContainer.SetActive(true);
        UnityEngine.SceneManagement.SceneManager.LoadSceneAsync
        ("Main Menu");
    }
    else
    {
        loadingContainer.SetActive(false);
    }
}

public void LoginAsGuest()
{
    string deviceIdentifier = SystemInfo.deviceUniqueIdentifier;
    PlayerPrefs.SetString("AUTH_IS_GUEST", "true");
    PlayerPrefs.SetString("AUTH_GUEST_TOKEN", deviceIdentifier);
    loadingContainer.SetActive(true);
    UnityEngine.SceneManagement.SceneManager.LoadSceneAsync
    ("Main Menu");
}
```

```
    void StopXR()
    {
        // var xrManager = XRGeneralSettings.Instance.Manager;
        // if (!xrManager.isInitializationComplete)
        //     return; // Safety check
        // xrManager.StopSubsystems();
        // xrManager.DeinitializeLoader();
    }
}
```

ClientAPI

```
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;
using System;
using UnityEngine.Networking;
// using UnityEngine.Video;

[Serializable]
public class Product
{
    public string _id;
    public string product_name;
    public string description;
    public string price;
}

[Serializable]
public class Products
{
    public string status;
    public List<Product> products;
}

public class ClientAPI : MonoBehaviour
{
    public string loginURL;
    public string registerURL;
    public string checkUserURL;
    public string submitScenarioURL;
    public string submitOtherInfoURL;
    public string androidAssetsURL;
    public string iosAssetsURL;
    public string allProductsURL;
```

```
public string placeOrderURL;

public uint androidCRC;
public uint iosCRC;

public string token;

// public VideoPlayer vp;

void Start()
{
    token = PlayerPrefs.GetString("AUTH_TOKEN");
    Debug.Log(token);
}

public IEnumerator Login(Login login, System.Action<AuthToken>
loginCallback)
{
    var jsonData = JsonUtility.ToJson(login);
    Debug.Log(jsonData);

    using (UnityWebRequest www = UnityWebRequest.Post(loginURL,
    jsonData))
    {
        www.SetRequestHeader("content-type", "application/json");
        www.uploadHandler.contentType = "application/json";
        www.uploadHandler =
        new UploadHandlerRaw(System.Text.Encoding.UTF8.GetBytes
        (jsonData));

        yield return www.SendWebRequest();
        if (www.isNetworkError)
        {
            Debug.Log(www.error);
        }
        else
        {
            if (www.isDone)
            {
                var result = System.Text.Encoding.UTF8.GetString
                (www.downloadHandler.data);
                // result = "{\"result\":" + result + "}";
                var authToken = JsonUtility.FromJson<AuthToken>(result);

                loginCallback(authToken);
            }
            else
```

```
            {
                Debug.Log("Error!");
            }
        }
    }
}

public IEnumerator Register(Register register, System.Action<AuthToken>
loginCallback)
{
    var jsonData = JsonUtility.ToJson(register);
    Debug.Log(jsonData);

    using (UnityWebRequest www = UnityWebRequest.Post(registerURL,
    jsonData))
    {
        www.SetRequestHeader("content-type", "application/json");
        www.uploadHandler.contentType = "application/json";
        www.uploadHandler =
        new UploadHandlerRaw(System.Text.Encoding.UTF8.GetBytes
        (jsonData));

        yield return www.SendWebRequest();
        if (www.isNetworkError)
        {
            Debug.Log(www.error);
        }
        else
        {
            if (www.isDone)
            {
                var result =
                System.Text.Encoding.UTF8.GetString
                (www.downloadHandler.data);
                // result = "{\"result\":" + result + "}";
                var authToken = JsonUtility.FromJson<AuthToken>(result);

                loginCallback(authToken);
            }
            else
            {
                Debug.Log("Error!");
            }
        }
    }
}
```

5.3 Get all products

This section of code is responsible for retrieving a list of all products in the frontend. The content type of all products will be json, and the authorization and usertoken will be validated. If a network error occurs, the modal will display an error message; otherwise, the result will be uploaded and downloaded in JSON format.

```
public IEnumerator GetAllProducts(System.Action<Products>
getAllProductsCompleted)
{
    using (UnityWebRequest www = UnityWebRequest.Get(allProductsURL))
    {
        www.SetRequestHeader("content-type", "application/json");

        yield return www.SendWebRequest();
        if (www.isNetworkError)
        {
            Debug.Log(www.error);
        }
        else
        {
            if (www.isDone)
            {
                var result = System.Text.Encoding.UTF8.GetString
                (www.downloadHandler.data);
                // result = "{\"result\":" + result + "}";
                var authToken = JsonUtility.FromJson<Products>(result);

                getAllProductsCompleted(authToken);
            }
            else
            {
                Debug.Log("Error!");
            }
        }
    }
}

public IEnumerator CheckUser(string userToken, System.Action<AuthToken>
checkUserCallback)
{
    Debug.Log(userToken);

    using (UnityWebRequest www = UnityWebRequest.Get(checkUserURL))
    {
        www.SetRequestHeader("content-type", "application/json");
        www.SetRequestHeader("Authorization", userToken);
```

```
        yield return www.SendWebRequest();
        if (www.isNetworkError)
        {
            Debug.Log(www.error);
        }
        else
        {
            if (www.isDone)
            {
                var result = System.Text.Encoding.UTF8.GetString
                (www.downloadHandler.data);
                var authToken = JsonUtility.FromJson<AuthToken>(result);

                checkUserCallback(authToken);
            }
            else
            {
                Debug.Log("Error!");
            }
        }
    }
}

public IEnumerator SubmitScenario(RoomData roomData,
System.Action<AuthToken> submitScenarioCallback, string userToken)
{
    string jsonData = JsonUtility.ToJson(roomData);
    Debug.Log(jsonData);

    using (UnityWebRequest www = UnityWebRequest.Post(submitScenarioURL,
    jsonData))
    {
        www.SetRequestHeader("content-type", "application/json");
        if (userToken != "")
        {
            www.SetRequestHeader("Authorization", userToken);
        }

        www.uploadHandler.contentType = "application/json";
        www.uploadHandler =
        new UploadHandlerRaw(System.Text.Encoding.UTF8.GetBytes
        (jsonData));

        yield return www.SendWebRequest();
        if (www.isNetworkError)
        {
            Debug.Log(www.error);
```

```
        }
        else
        {
            if (www.isDone)
            {
                var result = System.Text.Encoding.UTF8.GetString
                (www.downloadHandler.data);
                // result = "{\"result\":" + result + "}";
                // Debug.LogWarning(result);

                var authToken = JsonUtility.FromJson<AuthToken>(result);

                submitScenarioCallback(authToken);
            }
            else
            {
                Debug.Log("Error!");
            }
        }
    }
}

public IEnumerator SubmitOtherInfo(OtherInfoData otherInfoData,
System.Action<AuthToken> submitOtherInfoCallback)
{
    string jsonData = JsonUtility.ToJson(otherInfoData);
    Debug.Log(jsonData);
    using (UnityWebRequest www = UnityWebRequest.Post(submitOtherInfoURL,
    jsonData))
    {
        www.SetRequestHeader("content-type", "application/json");
        www.SetRequestHeader("Authorization", token);
        www.uploadHandler.contentType = "application/json";
        www.uploadHandler =
        new UploadHandlerRaw(System.Text.Encoding.UTF8.GetBytes
        (jsonData));

        yield return www.SendWebRequest();
        if (www.isNetworkError)
        {
            Debug.Log(www.error);
        }
        else
        {
            if (www.isDone)
            {
                var result =
```

```
                System.Text.Encoding.UTF8.GetString
                (www.downloadHandler.data);
                var authToken = JsonUtility.FromJson<AuthToken>(result);

                submitOtherInfoCallback(authToken);
            }
            else
            {
                Debug.Log("Error!");
            }
        }
    }
}

public IEnumerator DownloadAssets(string Platform, System.Action<float>
downloadCallback)
{
    Debug.Log("Downloading for Platform: " + Platform);

    string assetDownloadURL;
    float downloadDataProgress;
    uint crc;

    if (Platform == "android")
    {
        assetDownloadURL = androidAssetsURL;
        crc = androidCRC;
    }
    else
    {
        assetDownloadURL = iosAssetsURL;
        crc = iosCRC;
    }

    using (UnityWebRequest www =
    UnityWebRequestAssetBundle.GetAssetBundle(assetDownloadURL))
    {
        var resultFile = Path.Combine(Application.persistentDataPath,
        "androidvideos");
        var dh = new DownloadHandlerFile(resultFile);
        dh.removeFileOnAbort = true;
        www.downloadHandler = dh;

        UnityWebRequestAsyncOperation operation = www.SendWebRequest();

        if (operation.webRequest.isNetworkError || operation.webRequest.
        isHttpError)
```

```
        {
            Debug.Log(operation.webRequest.error);
        }
        else
        {
            Debug.Log(operation.isDone);
            while (!operation.isDone)
            {
                downloadDataProgress =
                operation.webRequest.downloadProgress * 100.0f;
                Debug.Log(downloadDataProgress);
                downloadCallback(downloadDataProgress);
                yield return null;
            }

            downloadCallback(100f);
            // AssetBundle bundle = DownloadHandlerAssetBundle.
            GetContent(www);
            // var assetLoadRequest = bundle.LoadAssetAsync
            ("Role01Helen", typeof(VideoClip));
            // yield return assetLoadRequest;

            // var getAllAssetNames = bundle.GetAllAssetNames();
            // yield return getAllAssetNames;

            // Debug.Log(getAllAssetNames);
            // GameObject obj = assetLoadRequest.asset as GameObject;

            // VideoClip videoClip =  assetLoadRequest.asset as
            VideoClip;
            // vp.clip = videoClip;
            // vp.Play();
            Debug.Log("Done");
        }
    }
}

public IEnumerator PlaceOrder(Order otherInfoData,
System.Action<AuthToken> submitOtherInfoCallback)
{
    string jsonData = JsonUtility.ToJson(otherInfoData);
    Debug.Log(jsonData);
    using (UnityWebRequest www = UnityWebRequest.Post(placeOrderURL,
    jsonData))
    {
        www.SetRequestHeader("content-type", "application/json");
        www.SetRequestHeader("Authorization", token);
```

```
            www.uploadHandler.contentType = "application/json";
            www.uploadHandler =
            new UploadHandlerRaw(System.Text.Encoding.UTF8.GetBytes
            (jsonData));

            yield return www.SendWebRequest();
            if (www.isNetworkError)
            {
                Debug.Log(www.error);
            }
            else
            {
                if (www.isDone)
                {
                    var result = System.Text.Encoding.UTF8.GetString
                    (www.downloadHandler.data);
                    var authToken = JsonUtility.FromJson<AuthToken>(result);

                    submitOtherInfoCallback(authToken);
                }
                else
                {
                    Debug.Log("Error!");
                }
            }
        }
    }
}
```

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

public class CharacterMovement : MonoBehaviour
{
    Vector3 directionToMove;
    float gravity;
    CharacterController characterController;

    Transform mainCameraTransform;
    public float speed = 2f;

    // Start is called before the first frame update
    void Start()
    {
        gravity = -9.8f;
        directionToMove = Vector3.zero;
        characterController = gameObject.GetComponent<CharacterController>();
```

```
        mainCameraTransform = Camera.main.transform;
    }

    // Update is called once per frame
    void Update()
    {
        if (!characterController.isGrounded)
        {
            directionToMove.y += gravity * Time.deltaTime;
        } else {
            if (mainCameraTransform.eulerAngles.x >= 54f &&
            mainCameraTransform.eulerAngles.x <= 65f)
            {
                directionToMove = mainCameraTransform.TransformDirection
                (Vector3.forward);
                directionToMove += directionToMove * speed;
            } else {
                directionToMove = Vector3.zero;
            }
        }

        characterController.Move(directionToMove * Time.deltaTime);
    }
}
```

Raycasting

```
protected abstract bool PerformRaycast(GvrBasePointer.PointerRay pointerRay,
                                       float radius, PointerEventData
                                       eventData, List<RaycastResult>
                                       resultAppendList);

    private void RaycastHybrid(GvrBasePointer pointer,
                               PointerEventData eventData,
                               List<RaycastResult> resultAppendList)
    {
        CurrentRaycastModeForHybrid = GvrBasePointer.RaycastMode.Direct;
        lastRay = GvrBasePointer.CalculateHybridRay(pointer,
        CurrentRaycastModeForHybrid);
        float radius = pointer.CurrentPointerRadius;
        bool foundHit = PerformRaycast(lastRay, radius, eventData,
        resultAppendList);

        if (!foundHit)
        {
            CurrentRaycastModeForHybrid = GvrBasePointer.RaycastMode.Camera;
            lastRay = GvrBasePointer.CalculateHybridRay(pointer,
```

```
            CurrentRaycastModeForHybrid);
            PerformRaycast(lastRay, radius, eventData, resultAppendList);
        }
    }

    private void RaycastDefault(GvrBasePointer pointer,
                                PointerEventData eventData,
                                List<RaycastResult> resultAppendList)
    {
        lastRay = GvrBasePointer.CalculateRay(pointer, pointer.raycastMode);
        float radius = pointer.CurrentPointerRadius;
        PerformRaycast(lastRay, radius, eventData, resultAppendList);
    }
}
```

5.4 GVR Interaction

Implementation of GvrBasePointer for a laser pointer visual. This script is to be attached to the controller object. The laser visual is important to help users locate their cursor when its not directly in their field of view. This part of code implements Raycasting: The greatest distance that raycast hits will be detected from the pointer. When nothing is hit, the reticle will be drawn at this distance from the pointer. The length of the laser is used as the CameraRayIntersectionDistance by default. The proportion of the reticle mesh that displays the reticle gets the laser beam's visual object.

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

public class GVRButton : MonoBehaviour
{
    public Image fillCircle;
    public GameObject popupTextGO;
    public Text popupText;
    public Text titleText;

    public UnityEvent GVRClick;
    bool gvrStatus;
    public float totalTime = 4;
    public float gvrtimer;

    public string popupString;
    public string titleString;

    // Start is called before the first frame update
```

```
    void Start()
    {
        fillCircle = transform.Find("Fill").GetComponent<Image>();
        // popupTextGO = transform.Find("More Info").gameObject;
        // popupText = transform.Find("More Info/Text").GetComponent
        <Text>();
        // titleText = transform.Find("Text").GetComponent<Text>();
        // titleText.text = titleString;
    }

    // Update is called once per frame
    void Update()
    {
        if (gvrStatus)
        {
            // Show Popup
            // popupTextGO.SetActive(true);
            // popupText.text = popupString;
            gvrtimer += Time.deltaTime;
            fillCircle.fillAmount = gvrtimer / totalTime;
        }

        if (gvrtimer > totalTime)
        {
            UnityEngine.EventSystems.EventSystem.current.
            SetSelectedGameObject(gameObject);
            GVRClick.Invoke();
            OnPointerExit();
        }
    }

    public void OnPointerEnter()
    {
        gvrStatus = true;
        Debug.Log("Pointer Entered GVRButton...");
    }

    public void OnPointerExit()
    {
        Debug.Log("Pointer Exited GVRButton...");

        // popupTextGO.SetActive(false);
        gvrStatus = false;
        fillCircle.fillAmount = 0;
        gvrtimer = 0;
    }
}
```

Chapter 6

Testing

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the system is defect free. It involves the execution of a component to evaluate one or more properties of interest. Testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements.

6.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input. In this application actual functionality is place order. In this application user can see multiple products. Unit testing checks that whether the products appear in the screen or not. User can select their products as per the choice. After that they can place the order. Unit testing checks that whether the order is placed or not.

6.2 Integration Testing

Integration testing is to check whether the application is working or not. In VR shopping application every feature is check. After unit testing each module, all the modules are tested simultaneously. In VR shopping application the first activity is login screen. If the he/she is new user then they need to first register themselves. If the user login is successfully then only user can see products. Integration testing checks that whether the products appear in the screen or not. User can select their products as per the choice. After that they can place the order. Integration testing checks that whether the order is placed or not. Integration testing checks whether the user is able to make payment or not.

Test Cases:

TestID	Test Case Condition	Input	Expected Result
1.	Enter Name	John Mathew	Field should contain only text
2.	Enter Email	abc@gmail.com	Field contain all characters
3.	Enter password	abc12345	Field contain all characters
4.	Enter confirm password	abc12354	Please Confirm the Password
5.	Enter phone number	9874561230	Field contain only numbers
6.	Enter address	Ganesh Nagar	Field contain text and number
7.	Enter city	Mumbai	Field should contain only text
8.	Enter pin code	415263	Field contain only numbers
9.	If details are valid	Click on register	Registration successfully
10.	If details are empty	Details empty	Please fill all the details.
11.	Email without @ sign	johngmail.com	Enter valid email.
12.	Password below 8 digit	123456	Password must be 8 digits.
13.	Passwords do no match	a) 12345678 b) 1234567	Please Confirm the password
14.	Pincode is below 6 digits	41526	Pin code must be 6 digits.
15.	Email and passw is valid	Click on login	Login successfully
16.	User can select product	select product quantity: 5	Product displayed in cart
17.	User see added products	User see added products	User can place the order.
18.	Select payment type	Type:-Cash/Debit	Order placed successfully.
19.	User can see order status	See orders status	See orders status

Table 6.1: Test Cases

Chapter 7

Result

7.1 V-Mart Application

7.1.1 Adding products to cart

We have implemented a UI that if a user firstly interacts with an application, it would be more attractive and user friendly. After navigating the 3D store, user can add the selected products in the cart. The Google cardboard is equipped with the button above the it, which allows the user to add the products to cart and can select quantity as per choice.

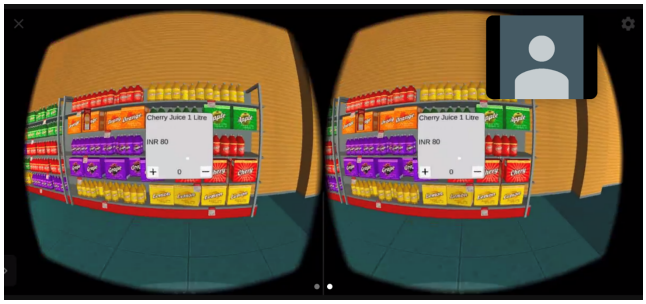

Figure 7.1: Adding products to cart

7.1.2 Checkout

Checkout is where the user finalizes choices about the product, selects any add-ons, confirms shipping options, then provides payment. It plays a very important role in the overall shopping experience on application. The checkout page is the last step before the sale is confirmed, and it requires customers to enter their information and then finalize their purchase.The checkout page is shown at the end of the checkout process, giving the customer a series of payment options and showing them an overview of their shopping cart.

Figure 7.2: Checkout

7.1.3 Payment Activity

A payment method is a way that customers pay for a product or service. As it is a 3D virtual store, accepted payment methods include credit cards, prepaid cards, or net banking. The user will have to enter the card details like card number, CVV number and the expiry date. Once entered, a otp pin will be sent to user's email id to verify the details and providing efficient transaction.

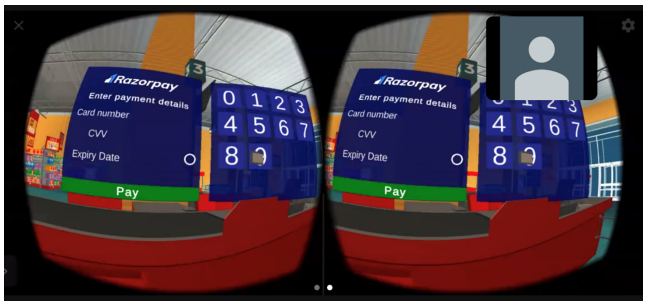

Figure 7.3: Payment Activity

7.1.4 Payment Successful

Once the payment is done by the user, a dialog box box of payment payment successful appears on the screen assuring the user about successful payment. The email about successful transaction is also sent to user's email id providing efficient and secure way of transaction.

Figure 7.4: Payment Successful

7.2 Vmart Web Dashboard

7.2.1 Order Activity

Admin can see order activity. Admin can user's order activity, which users are more active in the online shopping, what products are being more purchased or are more on demand. They can also see the users cart, payment types. It allows store owner to track all the activities of the logged in customers send marketing emails to attract them back to the store. Visiting any product categories by logged in customers will be recorded and admin will be able to view them on backend.

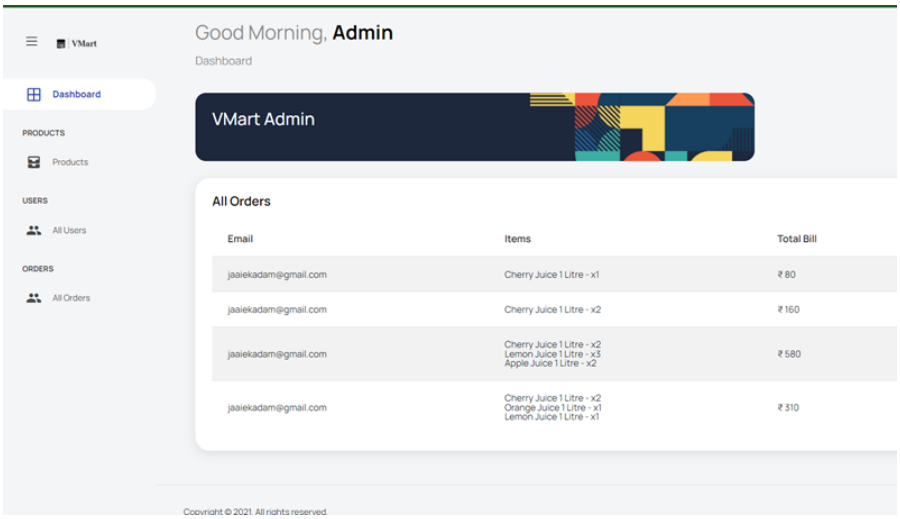

Figure 7.5: Order Activity

7.2.2 Tracking User Profile

VMart web dashboard also gives the admin to track users profile.A session is used to temporarily store the information on the server to be used across multiple pages of the website. It is the total time used for an activity. The user session starts when he logs-in to a particular 3D virtual store application and ends when the user logs out from the application. Tracking user activity is a type of proactive surveillance that helps to prevent misuse of access privileges.

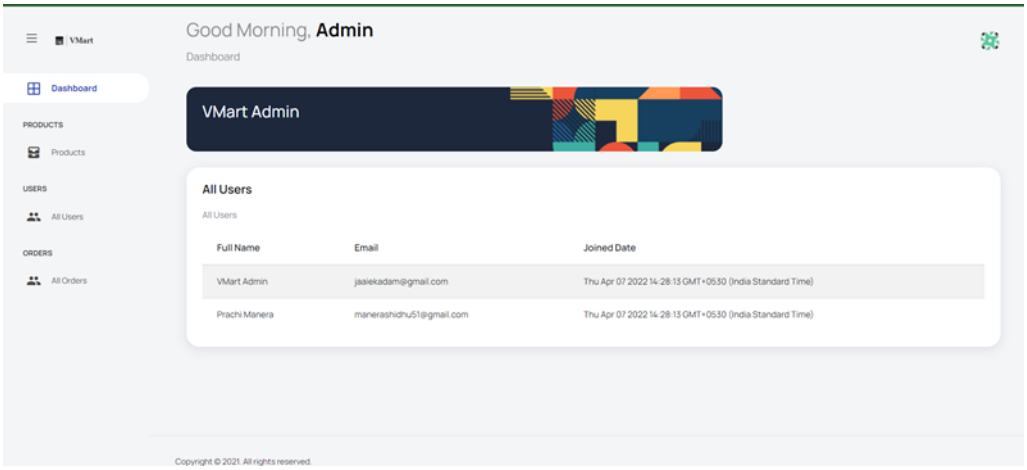

Figure 7.6: Tracking User Profile

7.2.3 Manage product list

It allows admin to manage more products in the store giving the user flexibility to shop. It is the process of ensuring that your product database is organized, structured, and up-to-date. It is the strategic process of managing the 3D virtual store product catalog to ensure the quality of the product data across all sales channels. It includes how admins organize, standardize, and publish the product data to each sales. It is also marketing material for all the products a store has or wants to sell to customers designed to display relevant product details that help buyers make informed purchase decisions.

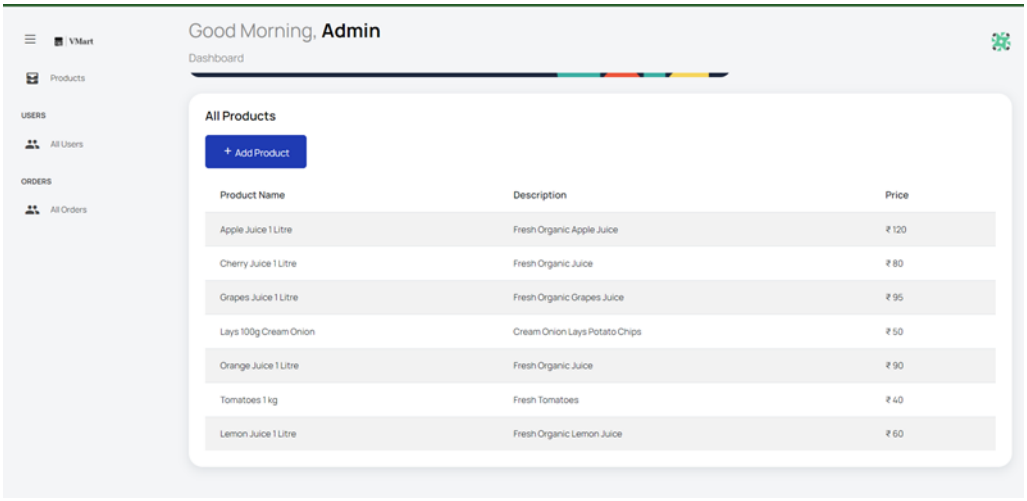

Figure 7.7: Manage product list

7.3 Graph

A graph is a diagram that pictorially represents the entire data collection and its output is also more visually appealing.Head-mounted display or HMD is a device worn over your head. It features a display in front of one or both of your eyes. The display streams data, images and other information in front of the wearer's eyes. Certain HMDs such as Google Cardboard, Oculus Rift or HTC Vive have displays over both of their users' eyes. The majority of Virtual Reality (VR) and Augmented Reality (AR) devices are head-mounted displays. The minimalistic Google Cardboard earned the praise for being the best VR headset on a tight budget. This product is comfortable or immersive of the bunch, but it is also a fantastic value, offering a decent introduction to VR at a fraction of the price of the other models.The HTC Vive Pro is the second most preferable as it has a much faster refresh rate than the Oculus. Though the HTC VIVE Pro is a little more expensive than the Oculus Quest 2, it offers a much higher resolution, and stronger performance as a result. The Oculus Quest 2 is lightweight, comfortable, and powerful enough to run impressively detailed virtual reality experiences.

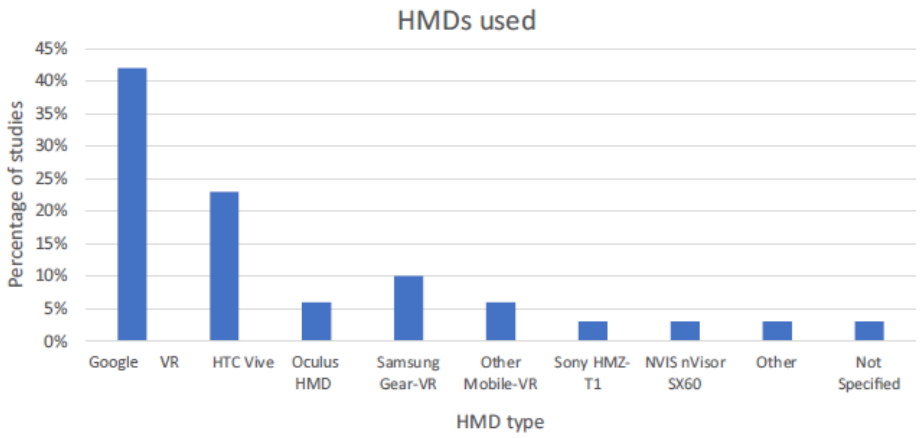

Figure 7.8: Comparison of HMD Devices

Chapter 8

Conclusions and Future Scope

Stronger demand for immersive virtual reality (VR) applications has led to a growing market for head-mounted displays (HMDs). With this popularity, the computation and graphics power of mobile devices also increases, which could catalyze their stagnating tendency. By using smartphone VR, it is possible to use immersive VR applications nearly everywhere, since the availability of mobile web also rises. Besides the gaming sector, use cases can be found in the constantly growing e-commerce sector, which is used by almost every smart-phone user nowadays, whether in online web shops or in-app purchases. Analysts predict that COVID-19 and other pandemics will persist in the coming decade. So, with the present circumstances of the covid-19 epidemic in mind, we planned and constructed a Virtual Reality-based retail mall called "V-Mart". This V-Mart prototype will be able to provide users with the in-person interactivity with products, inspection of retail items, walk and browsethrough quality of a traditional brick-and-mortar store with the benefits of e-commerce websites such as search and navigation functionality, relevant recommendations, availing coupons/discounts, secure payment, and, most importantly, the convenience of shopping from home using only a low-cost VR headset and mobile. Anyone with an internet connection will be able to enjoy this immersive shopping experience through the V-Mart application. With the available technology and our V-Mart prototype implementation, we believe that 3D retail apps are feasible.

Bibliography

[1] S. Altarteer, V. Charissis, D. Harrison, and W. Chan, "Development and heuristic evaluation of semiimmersive hand-gestural virtual reality inter-face for luxury brands online stores," in Augmented Reality, Virtual Reality, and Computer Graphics (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2017, pp. 464–477. doi: 10.1007/978-3-319-60928-7-39.

[2] S. Altarteer, C. Vassilis, D. Harrison, and W. Chan, "Product customi-sation: Virtual reality and new opportunities for luxury brands online trading," in Proc. 21st Int. Conf. Web3D Technol. Web3D, vol. 16, 2016, pp. 173–174.

[3] D. A. Griffith, R. F. Krampf, and J. W. Palmer. The role of interface in electronic commerce: Consumer involvement with print versus on-line catalogs. International Journal of Electronic Commerce, 5(4):135–153, 2001.

[4] K. C. Lee and N. Chung. Empirical analysis of consumer reaction to the virtual reality shopping mall . Computers in Human Behavior, 24(1):88 – 104, 2008. doi: 10.1016/j.chb.2007.01.018

[5] P. Lubos, G. Bruder, and F. Steinicke. Influence of Comfort on 3D Selection Task Performance in Immersive Desktop Setups. Journal of Virtual Reality and Broadcasting (JVRB), 12(2).

[6] M. Speicher, S. Cucerca, and A. Kr¨uger. Vrshop: A mobile interactive virtual reality shopping environment combining the benefits of onand offline shopping. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 1(3):102:1–102:31, Sept. 2017. doi: 10.1145/3130967

[7] M. Speicher, F. Daiber, S. Gehring, and A. Kr¨uger. Exploring 3d manipulation on large stereoscopic displays. In Proceedings of the 5th ACM International Symposium on Pervasive Displays, PerDis '16, pp. 59–66. ACM, New York, NY, USA, 2016. doi: 10.1145/2914920.2915018

[8] M. Speicher, F. Daiber, G.-L. Kiefer, and A. Kr¨uger. Exploring task performance and user's preference of mid-air hand interaction in a 3d docking task experiment. In Proceedings of the 5th Symposium on Spatial User Interaction, SUI '17, pp. 160–160. ACM, New York, NY, USA, 2017. doi: 10.1145/3131277.3134356

[9] M. Speicher, R. Siegel, and A. Kr¨uger. Productfinder: A location aware product information display for retail environments. In Proceedings of the 6th ACM International Symposium on Pervasive Displays, PerDis '17, pp. 23:1–23:2. ACM, New York, NY, USA, 2017. doi: 10.1145/ 3078810.3084351

Appendices

Detailed information, lengthy derivations, raw experimental observations etc. are to be presented in the separate appendices, which shall be numbered in Roman Capitals (e.g. "Appendix I"). Since reference can be drawn to published/unpublished literature in the appendices these should precede the "Literature Cited" section.

Appendix-A: Unity Download and Installation

1. Download Unity hub from https://unity3d.com/get-unity/download

2. To install the unity editor, sign in to Unity Hub Click the install tab. The default install locations are: Windows:C:/Program Files/Unity/Hub/Editor

3. Click Official Releases for released versions of the Editor, or Beta Releases for the latest Beta release of the Editor.

4. Click the Download button of the Editor version you want to install. This opens a dialog box called Add components to your install.

5. In the Add components to your install dialog box, select the components you want to install with the Editor, and click Done.

Appendix-B: MongoDB Download and Installation

1. Download the MongoDB MSI Installer Package from
https://www.mongodb.com/try/download/community

2. Download the MongoDB MSI Installer Package from
https://www.mongodb.com/try/download/community

A. Make sure you are logged in as a user with Admin privileges. Then navigate to your downloads folder and double click on the .msi package you just downloaded. This will launch the installation wizard.

B. Click Next to start installation.

C. Accept the licence agreement then click Next. Select the Complete setup.

D. Select “Run service as Network Service user” and make a note of the data directory, we’ll need this later.

E. Click Install to begin installation.

Publication

Paper entitled **"VR Shoppe: An Application To Shop Using Virtual Reality And Augmentec Reality""** was presented at **""ICCIIT-2022 Elsevier SSRN"** by **"Jaaie Kadam, Prachi Manera" and published in UGC listed Journal.**