|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Requirement Specifications**  The Decorum  Version: [1.0]   |  |  | | --- | --- | | Project Code | CS491 | | Supervisor | Dr. Fahad Samad | | Co Supervisor | --------- | | Project Team | Syed Muhammad Aqeel Abbas(17k3701)  Muhammad Furqan(17k3836)  Muneeb ul Hasan(17k3717) | | Submission Date | 14/12/2020 | |

[Instructions]

* No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.
* All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.
* This’ Instruction’ section should also be removed in final document.
* MS-Word Reviewing feature must be used to get the document reviewed by supervisors or co-supervisors.

Document History

[Revision history will be maintained to keep a track of changes done by anyone in the document.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
|  |  |  | [e.g. Document Created] |
|  |  |  | [Added Non-functional requirements] |
|  |  |  | [Added UseCase x.x.xx] |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
| Dr. Fahad Samad | | Supervisor |
| -------------- | | Co- Supervisor |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[1. Introduction 7](#_Toc178130213)

[1.1. Purpose of Document 7](#_Toc178130214)

[1.2. Intended Audience 7](#_Toc178130215)  
1.3 Abbreviations ……………………………………………………………………………………….. .7

[1.4. Document Convention 7](#_Toc178130216)

[2. Overall System Description 8](#_Toc178130217)

[2.1. Project Background 8](#_Toc178130218)

[2.2. Project Scope 8](#_Toc178130219)

[2.3. Not In Scope 8](#_Toc178130220)

[2.4. Project Objectives 8](#_Toc178130221)

[2.5. Stakeholders 8](#_Toc178130222)

[2.6. Operating Environment 9](#_Toc178130223)

[2.7. System Constraints 9](#_Toc178130224)

[2.8. Assumptions & Dependencies 9](#_Toc178130225)

[3. External Interface Requirements](#_Toc178130226) 9

[3.1. Hardware Interfaces](#_Toc178130227) 9

[3.2. Software Interfaces](#_Toc178130228) 9

[3.3. Communications Interfaces](#_Toc178130229) 9

[4. Functional Requirements 1](#_Toc178130230)0

[4.1. Functional Hierarchy 1](#_Toc178130231)0

[4.2. Use Cases 1](#_Toc178130232)1

[4.2.1. [View Products] 1](#_Toc178130233)1

[4.2.2. [Check products in AR] 1](#_Toc178130233)2

[4.2.3. [Redirect to web for checkout] 1](#_Toc178130233)2

[4.2.4. [Capture Picture] 1](#_Toc178130233)2

[4.2.5. [Share Picture] 1](#_Toc178130233)3

[5. Non-functional Requirements 1](#_Toc178130234)4

[5.1. Performance Requirements 1](#_Toc178130235)4

[5.2. Safety Requirements 1](#_Toc178130236)4

[5.3. Security Requirements 1](#_Toc178130237)4

[5.4. User Documentation 1](#_Toc178130238)4

[6. References 1](#_Toc178130239)4

[7. Appendices 1](#_Toc178130240)4

1. Introduction

* 1. Purpose of Document

This SDS is written in order to give a software development team overall guidance to the architecture of the software project.

* 1. Intended Audience

The intended audience would be the developers and approvers or the technical and academic staff of FAST NUCES karachi.

**1.3.Abbreviations**

AR = Augmented Reality

* 1. Document Convention

For Headings : Font is Calibri and font size is 14.

For Paragraph : Font is calibri and font size is 12.

1. Overall System Description
   1. Project Background

The problem with online shopping is that the customer doesn’t actually knows what the product will look like after it is delivered at their doorstep which makes the customer unsatisfied with the outcome of the product.

Our application will provide an interface to the customers to view the products in real life without actually buying the product. This thing will be done through AR where a customer will click on a product he wishes to buy, our app will show the product in customer mobile phones camera, he will be able to drag and place the product anywhere he wants and can see the outcome of the product without actually buying the product.

* 1. Project Scope

The project will be having the following main functionalities:

* 3d Object Tracking.
* Able to select angle of product.
* Take pictures while using AR.
* Share pictures.
* Select available color.
* Select available size.
* Search items.[FYP2]
* Click on product to see details of it on web(other project)[FYP2]
* Recommendation of products[FYP2]
  1. Not In Scope
* Product details.
* Add to cart products.
* Signup/Login
  1. Project Objectives

The objective of this project is to take online shopping to a next level with AR. This project will help the customers to buy the products that will suit with their room by placing it virtually in their room or at anyplace they want and see how the product will look.

* 1. Stakeholders

The stakeholders for this system are as follows:

1. Sellers: These interact with the system to manage their product selling business.
2. Buyers: These interact with the system to buy their desired product.
3. Developers: These are the people who are the technical team for the systems who are responsible for developing and maintaining the system.
   1. Operating Environment

Software Product

1. Database Server Firebase
2. Operating System Android 7.0.0 (Nougat)
3. Development Environment Android Studio
   1. System Constraints

2.7.1 Software/Hardware Constraints

Smart Phone on which our app will be running should have android version(7.0.0) or higher or should be upgradeable to android version 7.0.0.

2.7.2 Legal Constraints

This constraint includes Domain name Acquisition, Privacy policies, etc

2.7.3 Environmental Constraints

Since our project is an ecommerce web app so this constraint is not applicable.

* 1. Assumptions & Dependencies

We are making assumption that the mobile phone that will be using our app will be having Android version(7.0.0) or higher because AR only works on Android version(7.0.0) or higher.

1. External Interface Requirements
   1. Hardware Interfaces

Augmented reality requires Android version 7.0.0 so smart phone that will run this app should have android version 7.0.0 or should be upgradeable to 7.0.0

* 1. Software Interfaces

This app will only work on Android because ARCore api is only for android.

* 1. Communications Interfaces

1. Functional Requirements
   1. Functional Hierarchy

THE DECORUM

Redirect to Web App to Register/Login

Checkout Products

Redirect to App

Open Camera

Checkout on Augmented Reality

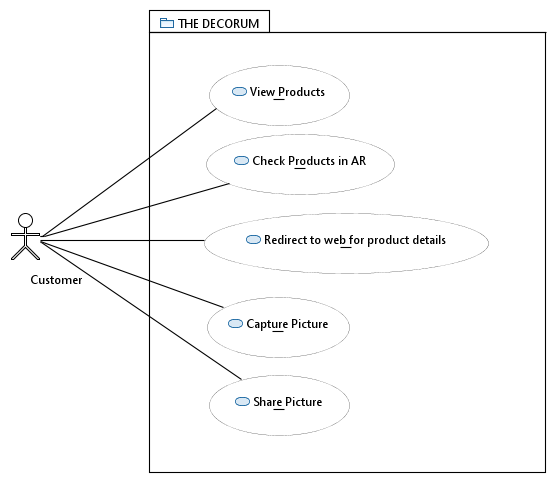
Save as a Picture / Share Picture

Order

Redirect to Web to Confirm Order.

Fig 1: Work breakdown structure of project

* 1. Use Cases



|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: View Products>** | | | |
| **Use case Id:** | | 1 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can view the products | | | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer opens the app. | | App will show the available products. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | If customer wants to check product details he will be redirected to web. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: Check product in AR>** | | | |
| **Use case Id:** | | 2 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can check the product in AR. | | | |
| **Pre-condition:** | | Customer should have clicked on a product he wishes to check in AR. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer clicks on the product. | | App will open camera and user will be able to drag and place the product anywhere. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | If customer wants to checkout he will be redirected to web. | | |
| **2.** | If customer wants to check another product he can go to back to products list. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: Redirect to web for product Details>** | | | |
| **Use case Id:** | | 3 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can view the products details | | | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer clicks on the product details button | | App will redirect customer to web. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | Customer can checkout from web. | | |
| **2.** | Comeback to web and view other products in AR. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: Capture Pictures>** | | | |
| **Use case Id:** | | 4 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can capture picture. | | | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer will click the capture button. | | App will click a picture. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | Customer can share the picture. | | |
| **2.** | Save the picture and check it afterwards. | | |
| **3.** | Cancel and click another picture. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<Use case Id: Share Picture>** | | | | |
| **Use case Id:** | | 5 | | |
| **Actors:**  Customer initiates the use case | | | | |
| **Feature:** Customer can share the picture | | | | |
| **Pre-condition:** | | Customer should have captured a picture in order to share it. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer will click the share button. | | | App will show share options. |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Click another picture. | | | |
| **2.** | Cancel this activity and go to products list. | | | |
| **Use Case Cross referenced** | | | Use Case: Capture Picture should be done before doing this use case. | |

1. Non-functional Requirements
   1. Performance Requirements

Performance will depend on the models that are loaded in the app. If the model is heavy it will take a little time to load.

* 1. Safety Requirements
  2. Security Requirements
  3. User Documentation

FYP-I does not include this requirement however in case of time then we might implement this feature as well.

1. References
   * + 1. Maxst.com. 2020. [online] Available at: <http://maxst.com/#/en/arguide> [Accessed 16 October 2020].
       2. Google Developers. 2020. *Quickstart For Android  |  Arcore  |  Google Developers*. [online] Available at: <https://developers.google.com/ar/develop/java/quickstart> [Accessed 16 October 2020].
       3. "Augmented Reality App Development - A Guide To Getting Started". 2020. *Waracle*. <https://waracle.com/blog/augmented-reality/introduction-augmented-reality-app-development/>.
       4. "Best AR SDK For Development For Ios And Android". 2020. *Thinkmobiles*. <https://thinkmobiles.com/blog/best-ar-sdk-review/>.
       5. "The Rise Of Augmented Reality In Interior Design And Property Development". 2020. *Iflexion*. https://www.iflexion.com/blog/augmented-reality-interior-design.
2. Appendices

**MAIN ACTIVITY:**

**package** com.example.secondar;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
  
**import** com.google.ar.core.Anchor;  
**import** com.google.ar.sceneform.AnchorNode;  
**import** com.google.ar.sceneform.rendering.ModelRenderable;  
**import** com.google.ar.sceneform.ux.ArFragment;  
**import** com.google.ar.sceneform.ux.TransformableNode;  
  
**import** java.util.ArrayList;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
  
 **private** ArFragment arFragment;  
 **private** ArrayList<Integer> imagesPath = **new** ArrayList<Integer>();  
 **private** ArrayList<String> namesPath = **new** ArrayList<>();  
 **private** ArrayList<String> modelNames = **new** ArrayList<>();  
 AnchorNode anchorNode;  
 **private** Button btnRemove;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 arFragment = (ArFragment)getSupportFragmentManager().findFragmentById(R.id.fragment);  
 btnRemove = (Button)findViewById(R.id.remove);  
 getImages();  
  
 arFragment.setOnTapArPlaneListener((hitResult, plane, motionEvent) -> {  
  
 Anchor anchor = hitResult.createAnchor();  
  
 ModelRenderable.builder()  
 .setSource(**this**,Uri.parse(Common.model))  
 .build()  
 .thenAccept(modelRenderable -> addModelToScene(anchor,modelRenderable));  
  
 });  
  
  
 btnRemove.setOnClickListener(view -> removeAnchorNode(anchorNode));  
 }  
  
 **private void** getImages() {  
  
 imagesPath.add(R.drawable.table);  
 imagesPath.add(R.drawable.bookshelf);  
 imagesPath.add(R.drawable.lamp);  
 imagesPath.add(R.drawable.odltv);  
 imagesPath.add(R.drawable.clothdryer);  
 imagesPath.add(R.drawable.chair);  
 namesPath.add(**"Table"**);  
 namesPath.add(**"BookShelf"**);  
 namesPath.add(**"Lamp"**);  
 namesPath.add(**"Old Tv"**);  
 namesPath.add(**"Cloth Dryer"**);  
 namesPath.add(**"Chair"**);  
 modelNames.add(**"table.sfb"**);  
 modelNames.add(**"model.sfb"**);  
 modelNames.add(**"lamp.sfb"**);  
 modelNames.add(**"tv.sfb"**);  
 modelNames.add(**"cloth.sfb"**);  
 modelNames.add(**"chair.sfb"**);  
  
 initaiteRecyclerview();  
 }  
  
 **private void** initaiteRecyclerview() {  
  
 LinearLayoutManager layoutManager = **new** LinearLayoutManager(**this**,LinearLayoutManager.HORIZONTAL,**false**);  
 RecyclerView recyclerView = (RecyclerView)findViewById(R.id.recyclerview);  
 recyclerView.setLayoutManager(layoutManager);  
 RecyclerviewAdapter adapter = **new** RecyclerviewAdapter(**this**,namesPath,imagesPath,modelNames);  
 recyclerView.setAdapter(adapter);  
  
 }  
  
 **private void** addModelToScene(Anchor anchor, ModelRenderable modelRenderable) {  
  
 anchorNode = **new** AnchorNode(anchor);  
 TransformableNode node = **new** TransformableNode(arFragment.getTransformationSystem());  
 node.setParent(anchorNode);  
 node.setRenderable(modelRenderable);  
 arFragment.getArSceneView().getScene().addChild(anchorNode);  
 node.select();  
 }  
  
 **public void** removeAnchorNode(AnchorNode nodeToremove) {  
 **if** (nodeToremove != **null**) {  
 arFragment.getArSceneView().getScene().removeChild(nodeToremove);  
 nodeToremove.getAnchor().detach();  
 nodeToremove.setParent(**null**);  
 nodeToremove = **null**;  
 }  
 }  
}

**RECYCLERVIEW ADAPTER:**

**package** com.example.secondar;  
  
**import** android.content.Context;  
**import** android.graphics.drawable.Drawable;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** java.util.ArrayList;  
  
**public class** RecyclerviewAdapter **extends** RecyclerView.Adapter<RecyclerviewAdapter.ViewHolder>{  
  
 **private** ArrayList<String> **textNames** = **new** ArrayList<>();  
 **private** ArrayList<Integer> **imagesPath** = **new** ArrayList<>();  
 **private** Context **context**;  
 **private** ArrayList<String> **modelNames** = **new** ArrayList<>();  
  
  
 **public** RecyclerviewAdapter(Context context,ArrayList<String> textNames, ArrayList<Integer> imagesPath,ArrayList<String> modelNames) {  
 **this**.**textNames** = textNames;  
 **this**.**imagesPath** = imagesPath;  
 **this**.**modelNames** = modelNames;  
 **this**.**context** = context;  
 }  
  
 @NonNull  
 @Override  
 **public** ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***layout\_list\_item***,parent,**false**);  
 **return new** ViewHolder(view);  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull ViewHolder holder, **int** position) {  
 holder.**imageView**.setImageResource(**imagesPath**.get(position));  
 holder.**textView**.setText(**textNames**.get(position));  
  
 holder.**imageView**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Common.*model* = **modelNames**.get(position);  
 }  
 });  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return imagesPath**.size();  
 }  
  
 **public class** ViewHolder **extends** RecyclerView.ViewHolder{  
  
 ImageView **imageView**;  
 TextView **textView**;  
  
 **public** ViewHolder(@NonNull View itemView) {  
 **super**(itemView);  
  
 **imageView** = itemView.findViewById(R.id.***imageview***);  
 **textView** = itemView.findViewById(R.id.***text***);  
 }  
 }  
}