MEDI-AR

A mini-project report submitted to the **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD** in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Vignesh J	**(15071A0576)**
Charan Reddy	**(15071A05A4)**
Prudvi raj B	**(15071A05B8)**
Manoj	**(16075A0524)**

Under the guidance of

M. Manasa Devi

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

VNR Vignana Jyothi Institute of Engineering and Technology (An Autonomous Institute, NAAC Accredited with 'A++' Grade, Approved by AICTE, New Delhi)

Bachupally(v), Hyderabad, Telangana, India. October,2018.

VNR Vignana Jyothi Institute of Engineering and Technology (An Autonomous Institute, NAAC Accredited with 'A++' Grade, Approved by AICTE, New Delhi)
Bachupally(v), Hyderabad, Telangana, India.

CERTIFICATE

This is to certify that **Vignesh J (15071A0576), Charan Reddy (15071A05A4), Prudhviraj B (15071A05B8), Manoj (16075A0524)** that have successfully completed their project work at CSE Department of VNR VJIET, Hyderabad entitled " **MEDIAR** " in partial fulfillment of the requirements for the award of B. Tech degree during the academic year 2018-2019.

This work is carried out under my supervision and has not been submitted to any other University/Institute for award of any degree/diploma.

M. Manasa Devi
Project Guide
Assistant Professor
CSE Department
VNRVJIET

Mrs. B. V. Kiranmayee
Associate Professor and Head
CSE Department
VNRVJIET

DECLARATION

We hereby declare that the project entitled "**MEDIAR**" submitted in partial fulfillment of the requirements for award of the degree of Bachelor of Technology in Computer Science and Engineering at **VNR Vignana Jyothi Institute of Engineering and Technology** in partial fulfillment of requirement for the award of Bachelor of Technology in Computer Science & Engineering is a bonafide report of the work carried out by us under the guidance and supervision of **M. Manasa Devi**, Assistant Professor, Department of Computer Science & Engineering , VNR Vignana Jyothi Institute of Engineering and Technology.

To the best of our knowledge, this has not been submitted to any other University/Institute for award of any degree/diploma.

Vignesh J	**Charan Reddy**	**Prudvi raj B**	**Manoj**
15071A0576	15071A05A4	15071A05B8	16075A0524
CSE	CSE	CSE	CSE
VNRVJIET	VNRVJIET	VNRVJIET	VNRVJIET

ACKNOWLEDGEMENT

Firstly, I would like to express my immense gratitude towards our institution **VNR Vignana Jyothi Institute of Engineering & Technology**, which created a great platform to attain profound technical skills in the field of Computer Science, thereby fulfilling our most cherished goal.

I am very much thankful to our H.O.D- **Mrs. B.V. Kiranmayee** madam for extending her cooperation in doing this project.

I extend my heartfelt thanks to our guide, **M. Manasa Devi** madam, for her enthusiastic guidance throughout the course of our project.

Last but not the least, our appreciable obligation also goes to all staff members of Computer Science & Engineering Department and to our fellow classmates who directly or indirectly helped us.

Vignesh J **(15071A0576)**

Charan Reddy **(15071A05A4)**

Prudhviraj B **(15071A05B8)**

Manoj **(16075A0524)**

ABSTRACT

Day by day the medical facilities are improving along with new diseases. which results in development of new drugs daily into the market which has different uses at different situations. The small change in the dosage misunderstanding leads to lose of life's. So give general idea to every Doctors/patients/pharma companies/illiterate person/Blind people to recognize what medicine to be taken in which quantity by simply scanning the drug using the camera of any smartphone.

Methodology

The most advanced technology AR (augmented reality) is used to resolve this problem. By using drugs labels as the image targets as the input source, A detailed pop of the drug which contains purpose, usage & quantity is augmented as output.

- For this purpose, we are going to use unity 3D editor with Vuforia support
- Basically, we are going to have the database which contains image targets as the label of drug's & the content related to each of the label Which describes the quantity & purpose of drug.
- In the case of blind & illiterate people the voice-over in different regional languages is provided.

Results

- This project help's in giving guidance & description to the people who are in emergency/ need of information for drug usage

CONTENTS

CHAPTER 1 Introduction…….…… 1

CHAPTER 2 Existing system and Proposed system……..... 2

2.1 Purpose of the Project

2.2 Existing System

2.2.1 Drawbacks of existing system

2.3 Proposed System

CHAPTER 3 Feasibility study……..... 3

3.1 Scope

3.2 Users of the System

CHAPTER 4 System Analysis…….....….... 4

4.1 Software and Languages used

4.1.1 JAVA

4.1.2 Unified Modeling Language

4.1.3 Google API for android

4.1.4 Optical Character Recognition

4.2 Tools Used

4.2.1 Android Studio

4.2.2 Star UML

Chapter 5 Software Design…… 8

5.1 Process Flowchart of Medi-AR

5.2 Flowchart of the system

5.3 Use Case Diagram

5.4 Activity Diagram

5.5 Class Diagram

Chapter 6 Implementation…….. 25

6.1 Screenshots

6.1.1 Scanned Object

6.2 Source code for the Image Tracking

6.3 Source code for Event Handler for Trackable Object.

Chapter 7 Testing……. 27

7.1 Test case Reports

Chapter 8 Conclusion……... 29

8.1 Present work

8.2 Future enhancements

References……. 30

1.Introduction

This is an application which scans the table using the mobile camera, analyses the data and creates an augmented text, speech of the analyzed tablet.

Nowadays it is tough for a person to learn details of every medicine and know how to use. Even for the visually impaired people, it is tough to read any text over a paper.

This project is all about scanning a tablet or a paper and converting it into speech & augmented text, which is mainly useful for blind people, visually impaired and illiterate people. This application works using the Augmented Reality concept.

This application is designed in such a way that when we focus the camera over the tablet it scans the image using Vuforia and retrieves the text & speech data.

2.Existing system and Proposed system:

2.1 Purpose of the project

The main aim of this project is to help the people who are trying to learn details of medicine and know how to use them. Helpful to the people who are visually impaired and illiterates.

2.2 Existing System

There are speech generators where we need to type the text that we want to listen. For visually impaired people they need to take help of others.

2.2.1 Drawbacks of existing system

- Difficult to type the complete tablet and generate text & speech.
- It is not always possible for visually impaired people or blind people to find help and access the system.
- Time taking process to get a document speech delivered.

2.3 Proposed System

In the proposed system we can get the outputted text & speech of tablet, which we scan through our MEDIAR application.

- Scanned text is embedded with text blocks.
- Tap on the required buttons to generate the speech.

3.Feasibility study

3.1 Scope

This type of application is available for all the users whoever has access to android mobiles and tablets.

3.2 Users of the System

- Administrator.

Administrator: Admin is responsible for performing technical administrative work like fixing any issues regarding the cam reader application and providing permission, security etc.

- Other Users

Other Users: Other users can use the MEDIAR application whenever they need to scan a document and can get the outputted speech.

4. System Analysis

4.1 Software and languages used

4.1.1 C#

C# is a general-purpose, multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines. Scripts are written in a special language that **Unity** can understand. And, it's through this language that we can talk to the engine and give it our instructions. The language that's **used in Unity** is called **C#** (pronounced C-sharp). All the languages that **Unity** operates with are object-oriented scripting languages.

C# is a programming language from Microsoft that's at the core of the .net framework. Although a cross-platform capable code, **C#** is most often used by programmers within the Microsoft ecosystem.

4.1.2 Unified Modeling language (UML)

The Unified Modeling Language which is also known as UML is used to visualize, specify construct and document the artifacts of a software-intensive system under development. It is similar to the blueprints that are used in other fields. It consists of different types of diagrams to explain design of a system that is going to be built.

UML diagrams also explains about the boundary, structure, and the behavior of the system and the objects within it. UML is best practice for building and documenting different aspects of software and business system modeling.

4.1.3 Vuforia software development kit

Vuforia is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to recognize and track planar images (Image Targets) and simple 3D objects, such as boxes, in real time.

This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real world images when they are viewed through the camera of a mobile device. The virtual object then tracks the position and orientation of the image in real-l time so that the viewer's perspective on the object corresponds with the perspective on the Image Target. It thus appears that the virtual object is a part of the real-world scene.

The Vuforia SDK supports a variety of 2D and 3D target types including 'markerless' Image Targets, 3D Multi-Target configurations, and a form of addressable Fiducial Marker, known as a VuMark. Additional features of the SDK include localized Occlusion Detection using 'Virtual Buttons', runtime image target selection, and the ability to create and reconfigure target sets programmatically at runtime.[2]

Vuforia provides Application Programming Interfaces (API) in C++, Java, Objective-C++ (a language utilizing a combination of C++ and Objective-C syntax), and the .NET languages through an extension to the Unity game engine.[3] In this way, the SDK supports both native development for

iOS and Android while it also enables the development of AR applications in Unity that are easily portable to both platforms. AR applications developed using Vuforia are therefore compatible with a broad range of mobile devices including the iPhone, iPad, and Android phones and tablets running Android OS version 2.2 or greater and an ARMv6 or 7 processor with FPU (Floating Point Unit) processing capabilities.

4.2 Tools Used

4.2.1 Android Studio

Android Studio is the integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for portable on Windows, macOS and Linux based operating systems. It is a replacement for the original Eclipse Android Development Tools as primary IDE for native Android application development.

4.2.2 Star UML

StarUML is an open source project to develop featureful, fast, extensible, flexible, and freely-available UML/MDA platform running on Win32 platform. The goal is to replacement of the commercial UML tools such as Rational Rose, Together and so on. StarUML supports the diagram types specified in UML 2.0. It is currently missing timing and interaction overview diagrams.

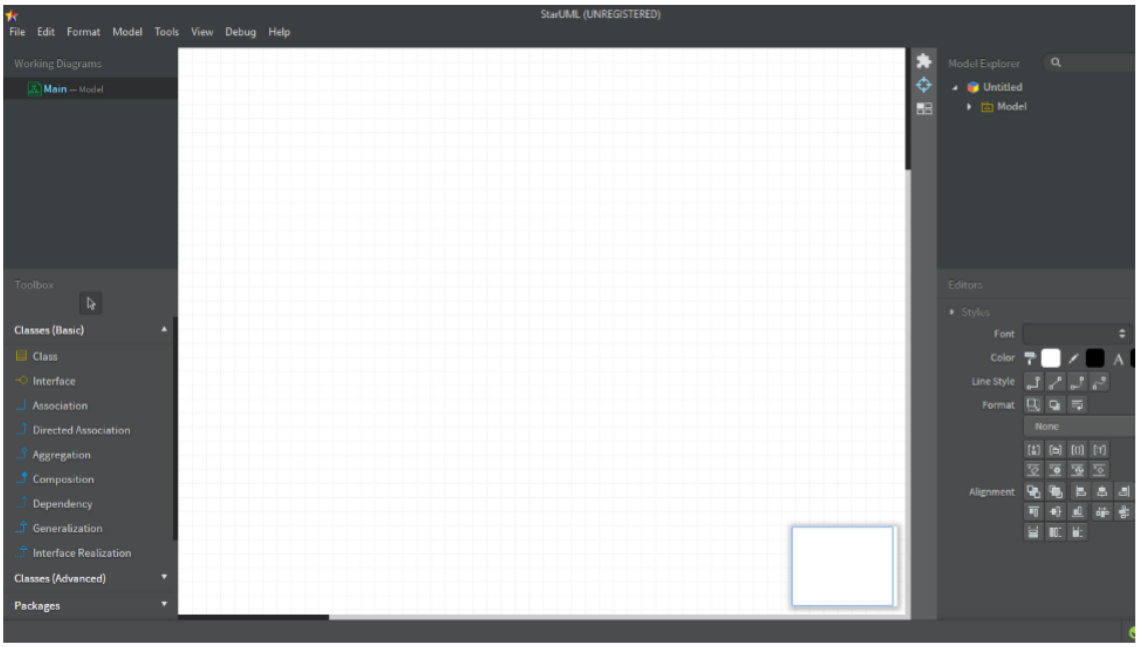


Figure 4.3 Star UML

4.2.3 Unity

Unity is a cross-platform game engine developed by Unity Technologies,[4] first announced and released in June 2005.

Unity gives users the ability to create games in both 2D and 3D, and the engine offers a primary scripting API in C#, for both the Unity editor in the form of plugins, and games themselves, as well as drag and drop functionality. Prior to C# being the primary programming language used for the engine, it previously supported Boo, which was removed in the Unity 5[7] release, and a version

of JavaScript called *UnityScript*, which was deprecated in August 2017 after the release of Unity 2017.1 in favor of C#.

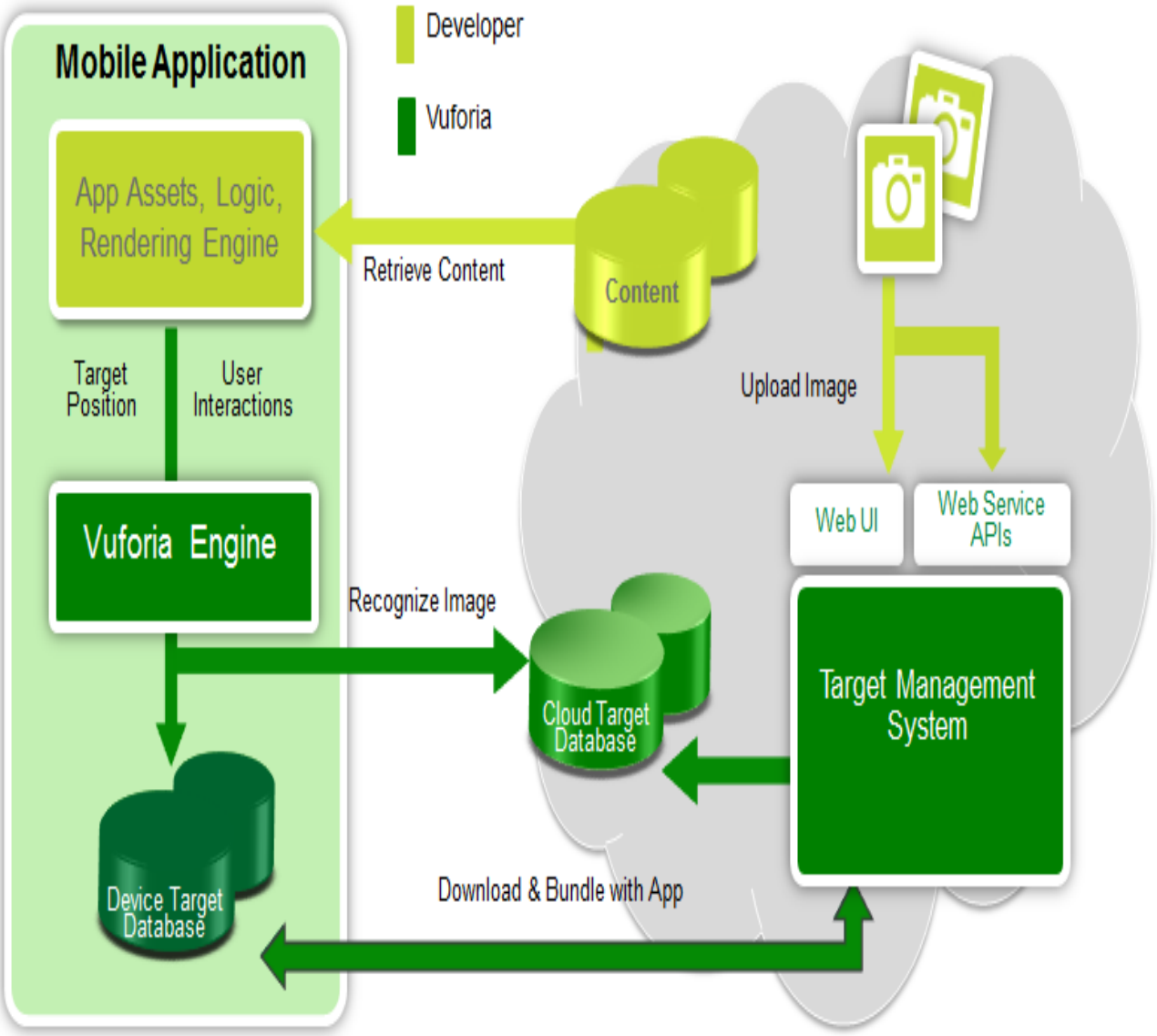
Figure 4.2.3 Unity

5. Software Design

Software design is the process in which an agent generates a specification of a software artifact, intended to accomplish goals, using a set of primitive components which are subjected to constraints. Software design may refer to either all the activities involved in implementing, conceptualizing, commissioning, framing, and ultimately modifying complex systems or the activities that are following requirements specification. Software design involves the problem solving and planning for a software solution.

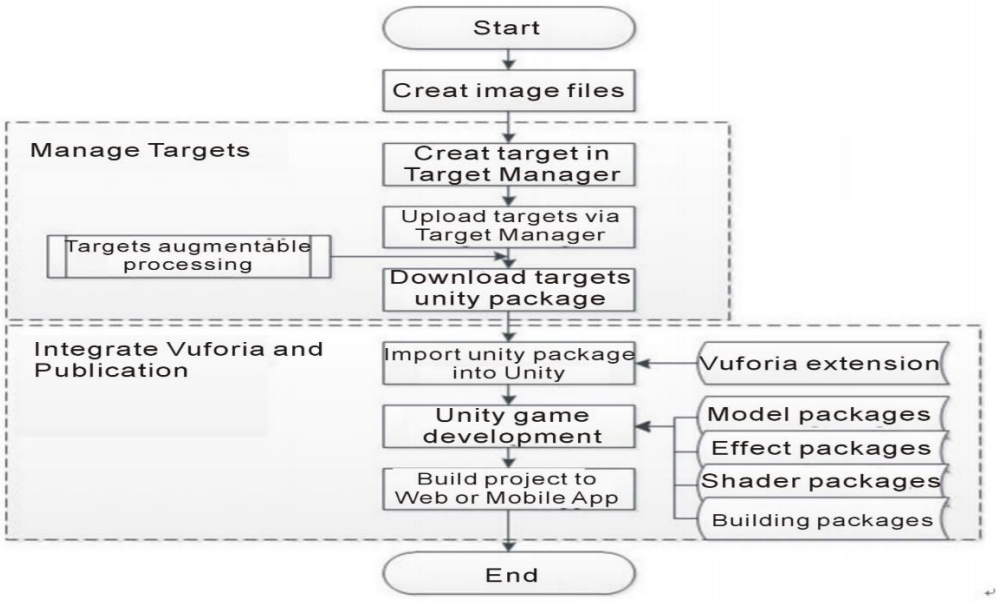
5.1 Process Flowchart of MEDIAR:

Recognition of scanned tablet images using Vuforia is now generally considered to be a solved problem. Components of Vuforia consist of mobile application, Vuforia engine, target management system, device target database and cloud target database.



5.2 Flowchart of the project:

The flowchart of MEDIAR represents the workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. Every symbol represents a part of code written in the program. The start/end symbol, represented by oval shape, can be used to represent either the beginning or ending of a program. The symbol for process, represented by rectangle shape box, allows you to show how the program is functioning. When you decide to enter data, show it on the screen, or print it to paper, you use the input/output symbol, represented by parallelogram. The decision symbol, represented by a rhombus shape, is used for things like 'if statements,' where you must choose an option based on a specified criterion. The flow line denotes the direction of logic flow in the program.



5.2 Use case Diagram:

A use case diagram is a simplest representation of a user's interaction with system which shows the relationship between the user and the different use cases in which the user is involved. This diagram can identify different types of users of a system.

Use Case Description:

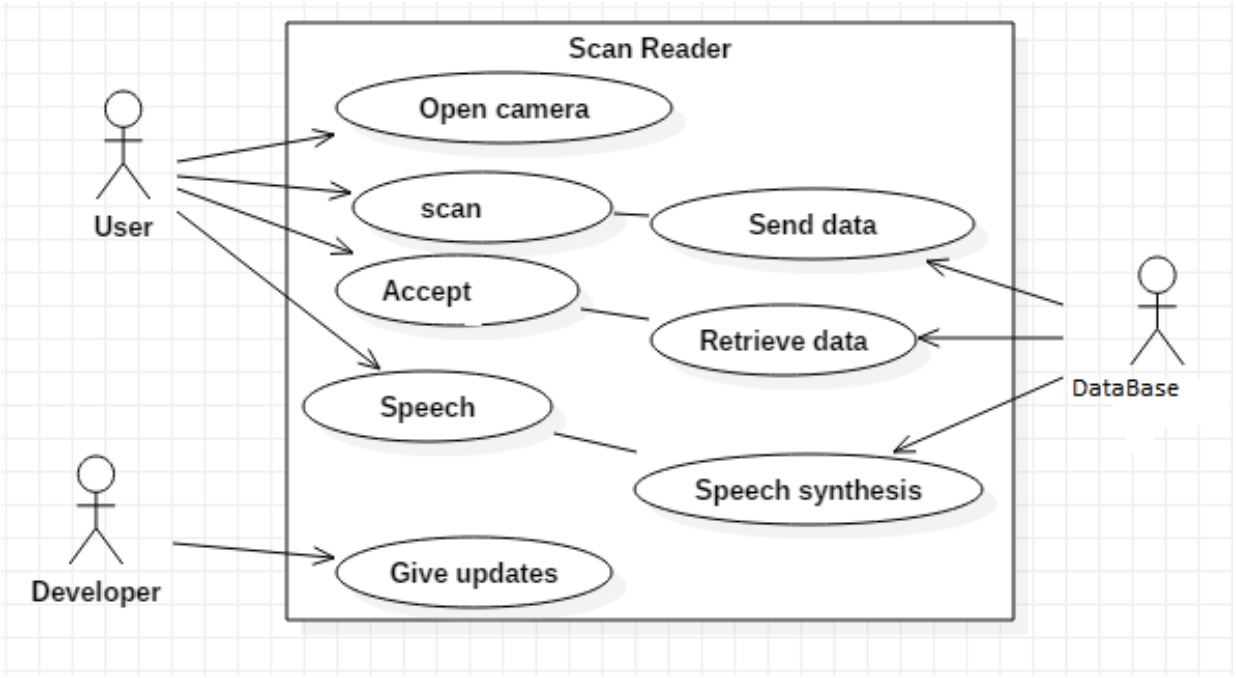


Figure 5-2 Use Case Diagram

Name of the use case: Open Camera

Description: User needs to open camera to scan the document.

Pre-condition: User must have already installed the application.

Post-condition: The user should scan the data.

Name of the use case: Scan Data

Description: It scans the data in the document.

Pre-condition: Camera should be opened.

Post-condition: The data is processed.

Name of the use case: Accept Data

Description: The user accepts the data.

Pre-condition: Data should be retrieved from the application.

Post-condition: Generated speech is the output.

Name of the use case: Speech

Description: The outcome of the text.

Pre-condition: Text is synthesized.

Name of the use case: Send Data

Description: The scanned data is sent to the application.

Pre-condition: Document should be scanned.

Post-condition: Application sends the data to the user.

Name of the use case: Retrieve Data

Description: Retrieve data from the application.

Pre-condition: Data should be sent.

Post-condition: Data should be accepted by the user.

Name of the use case: Speech Synthesis

Description: The data is synthesized from text to speech.

Pre-condition: Data should be accepted by the user.

Post-condition: User gets the data generated as speech.

Name of the use case: User

Description: Users can use the cam reader application whenever they need to scan a document and can get the outputted speech.

Name of the use case: Developer

Description: The developer provides updates for the application.

Name of the use case: Admin

Description: Admin performs technical administrative work like fixing any issues regarding the cam reader application and providing permission, security etc.

5.3 Activity Diagram:

We can make an activity diagram by connecting and joining various activity states. The starting point is usually marked with a dark, filled-in circle with an arrow pointing to the next state usually a rectangle with rounded corners. All the action flows will be represented with arrows indicating the transitions from state to state.

Description:

1. The user opens the application.

2. The camera is displayed.

3. It scans the data in the tablet projected with the camera.

4. Application embeds text and speech

5. Application outputs medicine details.

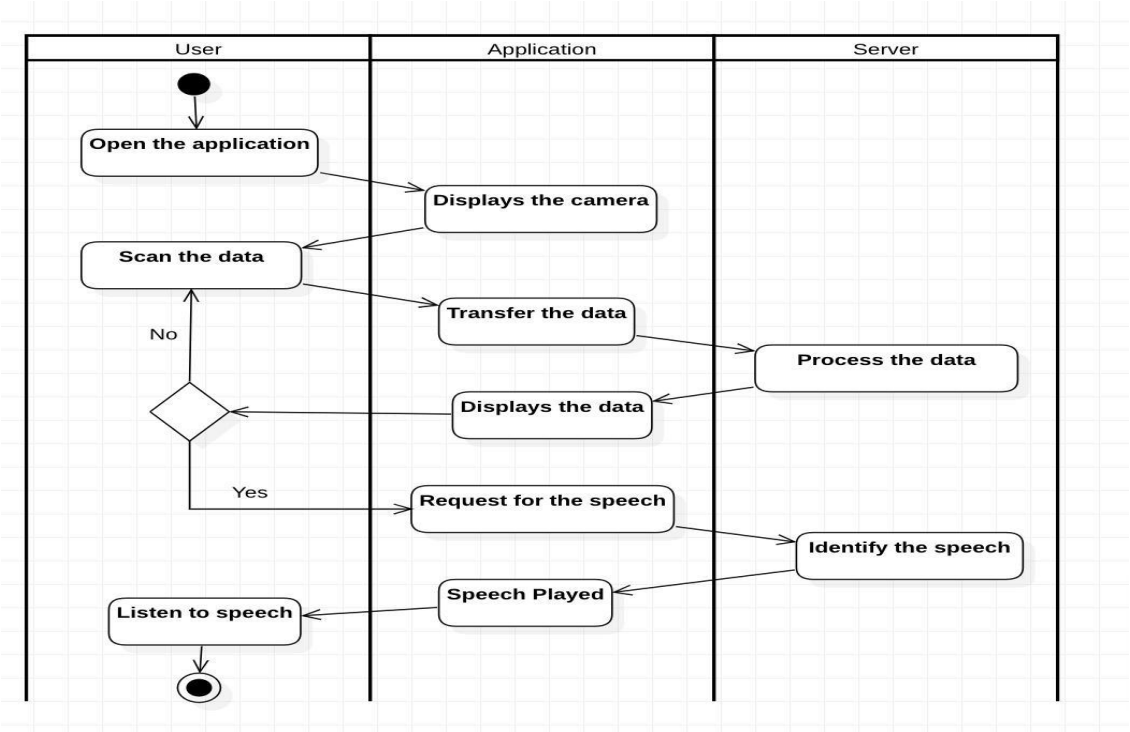


Figure 5-3 Activity Diagram

6. The application displays the scanned data as a highlighted text in blocks of text augemented around subject.

7. If the user does not accept the data then the data should be scanned again.

8. If the user accepts the data scanned, the any block of text can be selected.

9. The application output speech from details of extracted tablet or medicine.

10.The generated speech is delivered as output.

11. The user listens to the speech generated and sees the displayed information about subject i.e tablet or medicine scanned.

12. Thus, user can decide to use tablet or not, based on information given by the application

5.4 Class Diagram:

A class diagram in the Unified Modeling Language is a type of structure diagram(static) that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

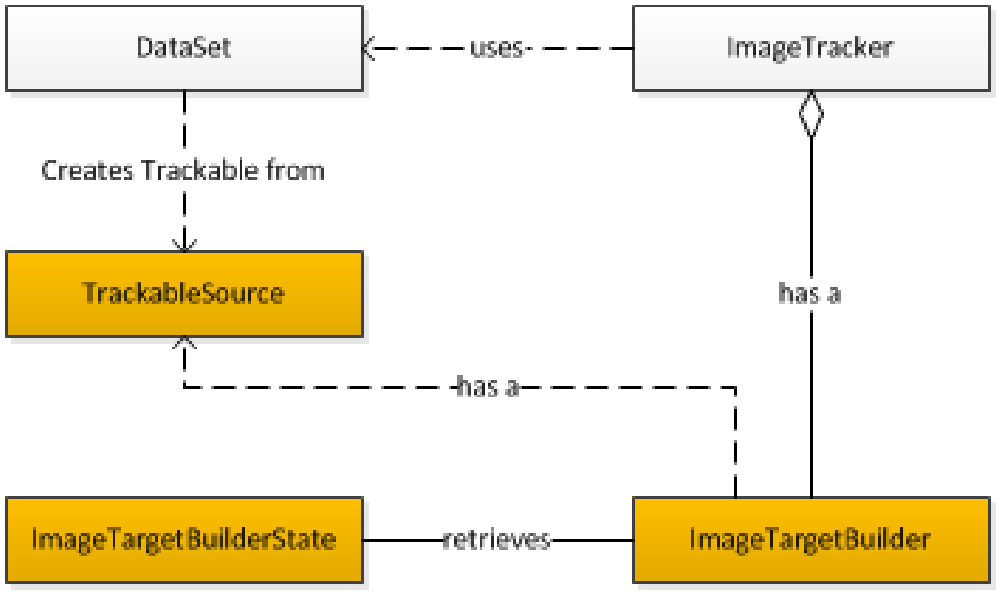


Figure 5-4 Class Diagram

The Class Diagram of Cam Medi-AR describes the static structure of system showing the systems classes, attributes operations and relationships among the objects. There are three main classes in the system namely OnImagedetection, OnTargetDetection and TrackableSource.

6.Implementation

6.1 Screenshots

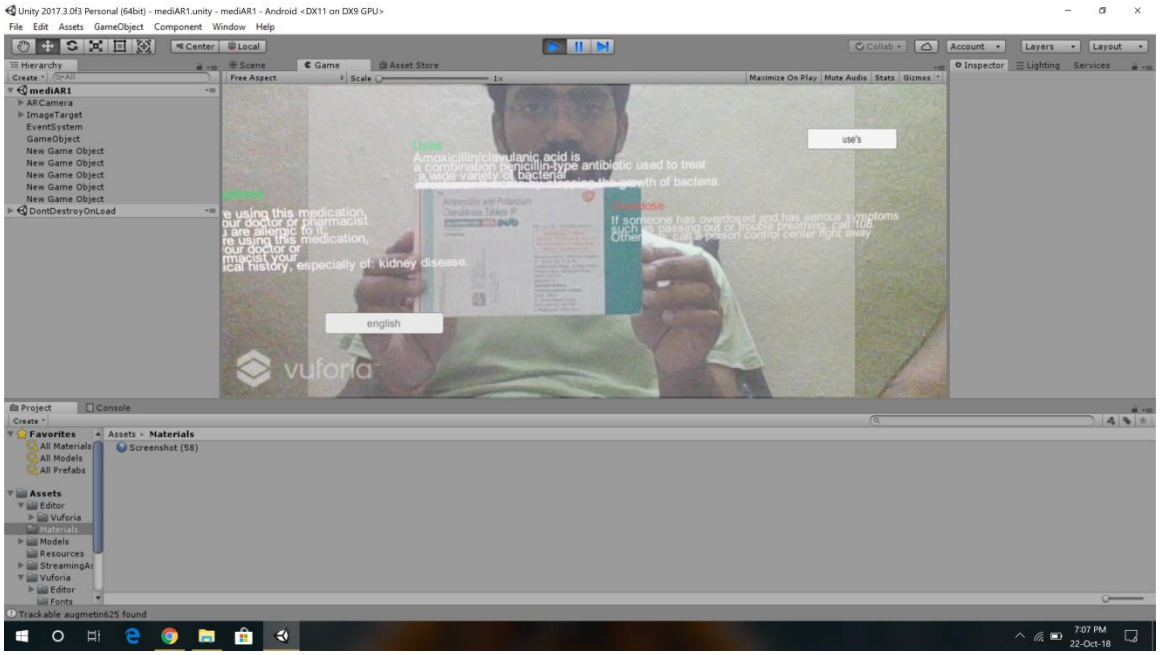


Figure 6.1.1 Document and Scanned Document

From the above Figure we can see that our application has scanned the document and the text is highlighted.

6.1.2 Image Target

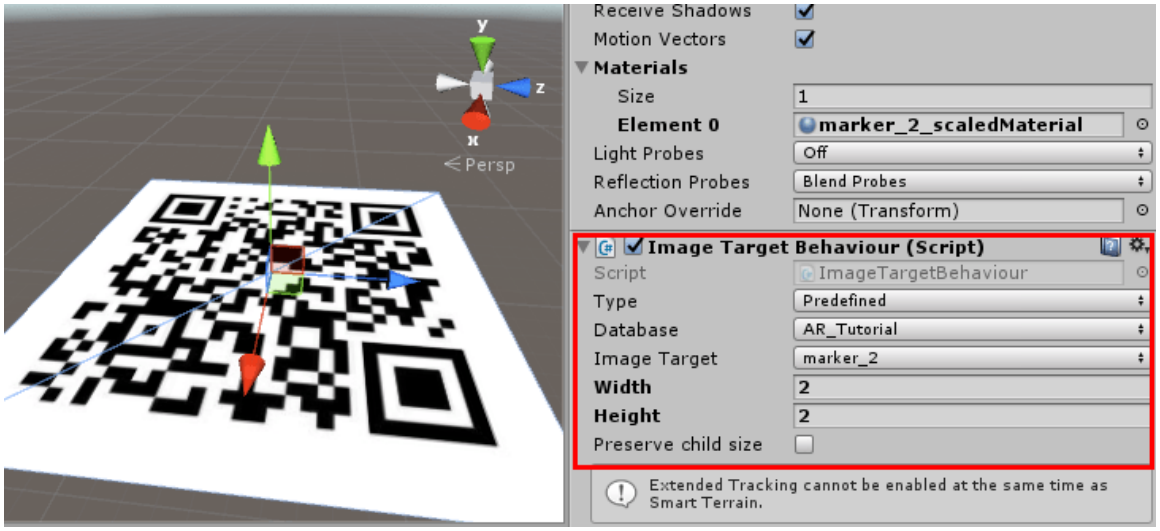


Figure 6.1.2 Target Image

We can tap on the required block of highlighted text and the extracted text is delivered as speech.

The copy of text is displayed on to the screen in the augmented space around subject i.e scanned object of medicine or tablet.

6.2 Source code for the Image Target:

```
/*==============================================================
==================
Copyright (c) 2017 PTC Inc. All Rights Reserved.

Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
==============================================================
================*/

using UnityEngine;
using Vuforia;

/// <summary>
///     A custom handler that implements the ITrackableEventHandler interface.
/// </summary>
public class DefaultTrackableEventHandler : MonoBehaviour,
ITrackableEventHandler
{
    #region PRIVATE_MEMBER_VARIABLES

    protected TrackableBehaviour mTrackableBehaviour;

    #endregion // PRIVATE_MEMBER_VARIABLES

    #region UNTIY_MONOBEHAVIOUR_METHODS

    protected virtual void Start()
    {
        mTrackableBehaviour = GetComponent<TrackableBehaviour>();
        if (mTrackableBehaviour)
            mTrackableBehaviour.RegisterTrackableEventHandler(this);
```

```
    }

    #endregion // UNTIY_MONOBEHAVIOUR_METHODS

    #region PUBLIC_METHODS

    /// <summary>
    ///     Implementation of the ITrackableEventHandler function called when the
    ///     tracking state changes.
    /// </summary>
    public void OnTrackableStateChanged(
        TrackableBehaviour.Status previousStatus,
        TrackableBehaviour.Status newStatus)
    {
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
        {
            Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
found");
            OnTrackingFound();
        }
        else if (previousStatus == TrackableBehaviour.Status.TRACKED &&
                 newStatus == TrackableBehaviour.Status.NOT_FOUND)
        {
            Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
lost");
            OnTrackingLost();
        }
        else
        {
            // For combo of previousStatus=UNKNOWN +
newStatus=UNKNOWN|NOT_FOUND
            // Vuforia is starting, but tracking has not been lost or found yet
            // Call OnTrackingLost() to hide the augmentations
            OnTrackingLost();
        }
    }

    #endregion // PUBLIC_METHODS

    #region PRIVATE_METHODS

    protected virtual void OnTrackingFound()
    {
        var rendererComponents = GetComponentsInChildren<Renderer>(true);
```

```
        var colliderComponents = GetComponentsInChildren<Collider>(true);
        var canvasComponents = GetComponentsInChildren<Canvas>(true);

        // Enable rendering:
        foreach (var component in rendererComponents)
            component.enabled = true;

        // Enable colliders:
        foreach (var component in colliderComponents)
            component.enabled = true;

        // Enable canvas':
        foreach (var component in canvasComponents)
            component.enabled = true;
    }


    protected virtual void OnTrackingLost()
    {
        var rendererComponents = GetComponentsInChildren<Renderer>(true);
        var colliderComponents = GetComponentsInChildren<Collider>(true);
        var canvasComponents = GetComponentsInChildren<Canvas>(true);

        // Disable rendering:
        foreach (var component in rendererComponents)
            component.enabled = false;

        // Disable colliders:
        foreach (var component in colliderComponents)
            component.enabled = false;

        // Disable canvas':
        foreach (var component in canvasComponents)
            component.enabled = false;
    }

    #endregion // PRIVATE_METHODS
}
```

Source code for the Event handler for Object Tracking:

```
/*==============================================================================
Copyright (c) 2017 PTC Inc. All Rights Reserved.

Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
==============================================================================*/

using UnityEngine;
using Vuforia;

/// <summary>
///     A custom handler that implements the ITrackableEventHandler interface.
/// </summary>
public class DefaultTrackableEventHandler : MonoBehaviour,
ITrackableEventHandler
{
    #region PRIVATE_MEMBER_VARIABLES

    protected TrackableBehaviour mTrackableBehaviour;

    #endregion // PRIVATE_MEMBER_VARIABLES

    #region UNTIY_MONOBEHAVIOUR_METHODS

    protected virtual void Start()
    {
        mTrackableBehaviour = GetComponent<TrackableBehaviour>();
        if (mTrackableBehaviour)
            mTrackableBehaviour.RegisterTrackableEventHandler(this);
```

```
    }

    #endregion // UNTIY_MONOBEHAVIOUR_METHODS

    #region PUBLIC_METHODS

    /// <summary>
    ///     Implementation of the ITrackableEventHandler function called when the
    ///     tracking state changes.
    /// </summary>
    public void OnTrackableStateChanged(
        TrackableBehaviour.Status previousStatus,
        TrackableBehaviour.Status newStatus)
    {
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
        {
            Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
found");
            OnTrackingFound();
        }
        else if (previousStatus == TrackableBehaviour.Status.TRACKED &&
                 newStatus == TrackableBehaviour.Status.NOT_FOUND)
        {
            Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
lost");
            OnTrackingLost();
        }
        else
        {
            // For combo of previousStatus=UNKNOWN +
newStatus=UNKNOWN|NOT_FOUND
            // Vuforia is starting, but tracking has not been lost or found yet
            // Call OnTrackingLost() to hide the augmentations
            OnTrackingLost();
        }
    }

    #endregion // PUBLIC_METHODS

    #region PRIVATE_METHODS

    protected virtual void OnTrackingFound()
    {
        var rendererComponents = GetComponentsInChildren<Renderer>(true);
```

```
        var colliderComponents = GetComponentsInChildren<Collider>(true);
        var canvasComponents = GetComponentsInChildren<Canvas>(true);

        // Enable rendering:
        foreach (var component in rendererComponents)
            component.enabled = true;

        // Enable colliders:
        foreach (var component in colliderComponents)
            component.enabled = true;

        // Enable canvas':
        foreach (var component in canvasComponents)
            component.enabled = true;
    }


    protected virtual void OnTrackingLost()
    {
        var rendererComponents = GetComponentsInChildren<Renderer>(true);
        var colliderComponents = GetComponentsInChildren<Collider>(true);
        var canvasComponents = GetComponentsInChildren<Canvas>(true);

        // Disable rendering:
        foreach (var component in rendererComponents)
            component.enabled = false;

        // Disable colliders:
        foreach (var component in colliderComponents)
            component.enabled = false;

        // Disable canvas':
        foreach (var component in canvasComponents)
            component.enabled = false;
    }

    #endregion // PRIVATE_METHODS
}
```

7.Testing

Software testing is one of the critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is the one of the step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy integrates software test case designing methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities which can be planned in advance and conducted systematically. The underlying motivation for the program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

• The process of the executing system with the intent of finding an error.

• Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free to produce the quality product and hence customer satisfaction.

• Quality is defined as justification of the requirements

• Defect is not different but deviation from the requirements

• Defect is nothing but bug.

• Testing --- The presence of bugs

• Testing can be demonstrated by the presence of bugs, but not their absence

7.1 Test case reports

Test case	Description	Test steps	Expected result	Actual result	Remarks
Image-1 	Verify that the application scans and generates correct speech	1.Open the Medi - AR application in mobile.	Application is active and responsive.	Application is active and responsive.	Passed.
		2. Scan the desired subject for details	Exact speech should be generated.	Speech form of details about subject is generated exactly.	Passed.

Test case ID	Description	Test steps	Expected result	Actual result	Remarks
Image-2 	Verify that the application scans and generates correct speech	1.Open the Medi - AR application in mobile	Application is active and responsive.	Application is active and responsive.	Passed.
		2. Scan the desired object for speech synthesis.	Exact speech should be generated.	Speech form of scanned text is not generated.	Failed.

8. Conclusion

8.1 Present work

We aim to create an application which recognizes the scanned object and outputs the details of it in forms of text, into speech, so that it helps the people who are trying to learn any language and the people who are unable to read. With this, any person can understand the basic information like dosage,composition and sideeffects or suitability of particular medicine for given disease based on symptoms or conditions observed.

8.2 Future enhancements

This application can further be extended by integrating it with the Cloud database so that we can generate the meaning of the scaneed object which helps us in understanding the medicine as well as user requirements more.

References

Website:

- https://developers.google.com/vision/
- https://developers.google.com/android/
- https://developer.android.com/reference/android/speech/tts/package-summary
- https://cloud.google.com/vision/docs
- Vuforia Product Page https://www.qualcomm.com/products/vuforia
- Vuforia Developer Page https://developer.vuforia.com
- SDK Download Page https://developer.vuforia.com/downloads/sdk
- Installing Vuforia for Unity extension
- http://developer.vuforia.com/library/articles/Solution/ Installing-the-Unity-Extension
- Tutorials https://developer.vuforia.com/resources/tutorials