**CHAPTER 1**

**INTRODUCTION**

* 1. **AIM**

The scope of our project is to make shopping experience easier between the seller and the customer. The basic idea is to make the system recognize the user’s pose and in turn respond with the image in which the virtual jewels rendered. The aim is to build an android application based on Augmented Reality through which the customers can virtually try on the jewels.

* 1. **PROBLEM STATEMENT**

A lot of shoppers have encountered a lot of problems while shopping at a high-end place for jewels, especially during peak hours, such as weekends. Numerous restrictions and enormous crowds make it quite an unpleasant experience. Huge number of customers, and minimum numbers of sales representatives results in quite a lot of waiting time for customers, ultimately resulting in dissatisfaction. Due to security reasons, there is also a restriction on the number of jewels that can be taken at one instance of time for trial. It increases the overall shopping time due to multiple trips from the shelves to the trials.

From the seller’s point of view, a large percentage of thefts happen because of sneaking in shops while trailing the jewels. Also they are unable to show the customers the fresh stock that is supposed to be delivered to the shop in the coming few days. Therefore, to overcome these problems, we propose a mobile application which uses Augmented reality concept. This application brings the entire shop within your hands and it makes the shopping easy and effective.

* 1. **DESCRIPTION**

***AUGMENTED REALITY***

Augmented reality (AR) is a live direct or indirect view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or [GPS](http://en.wikipedia.org/wiki/GPS) data. It is related to a more general concept called [mediated reality](http://en.wikipedia.org/wiki/Mediated_reality), in which a view of reality is modified by a computer. As a result, the technology functions by enhancing one’s current perception of reality. By contrast, [virtual reality](http://en.wikipedia.org/wiki/Virtual_reality) replaces the real world with a simulated one. Augmentation is conventionally in [real-time](http://en.wikipedia.org/wiki/Real-time_computing) and in semantic context with environmental elements, such as sports scores on TV during a match. With the help of advanced AR technology, the information about the surrounding real world of the user becomes [interactive](http://en.wikipedia.org/wiki/Interactive) and digitally manipulatable. Artificial information about the environment and its objects can be overlaid on the real world.

Modern mobile augmented reality systems use one or more of the following tracking technologies: [digital cameras](http://en.wikipedia.org/wiki/Digital_cameras) and/or other [optical sensors](http://en.wikipedia.org/wiki/Image_sensor), [accelerometers](http://en.wikipedia.org/wiki/Accelerometer), [GPS](http://en.wikipedia.org/wiki/GPS), [gyroscopes](http://en.wikipedia.org/wiki/Gyroscope), [solid state compasses](http://en.wikipedia.org/wiki/Digital_magnetic_compass), [RFID](http://en.wikipedia.org/wiki/RFID) and wireless sensors. These technologies offer varying levels of accuracy and precision. Most important is the position and orientation of the user's head. Tracking the user's hands or a handheld input device can provide a 6DOF interaction technique. The computer analyses the sensed visual and other data to synthesize and position augmentations.

***IMAGE REGISTRATION***

A key measure of AR systems is how realistically they integrate augmentations with the real world. The software must derive real world coordinates, independent from the camera, from camera images. That process is called [image registration](http://en.wikipedia.org/wiki/Image_registration) which uses different methods of [computer vision](http://en.wikipedia.org/wiki/Computer_vision), mostly related to [video tracking](http://en.wikipedia.org/wiki/Video_tracking). Many computer vision methods of augmented reality are inherited from [visual odometry](http://en.wikipedia.org/wiki/Visual_odometry). Usually those methods consist of two parts.

* First detect [interest points](http://en.wikipedia.org/wiki/Interest_point_detection), or [fiducial markers](http://en.wikipedia.org/wiki/Fiducial_marker), or [optical flow](http://en.wikipedia.org/wiki/Optical_flow) in the camera images. First stage can use [feature detection](http://en.wikipedia.org/wiki/Feature_detection_%28computer_vision%29) methods like [corner detection](http://en.wikipedia.org/wiki/Corner_detection), [blob detection](http://en.wikipedia.org/wiki/Blob_detection), [edge detection](http://en.wikipedia.org/wiki/Edge_detection) or [thresholding](http://en.wikipedia.org/wiki/Thresholding_%28image_processing%29) and/or other [image processing](http://en.wikipedia.org/wiki/Image_processing) methods.
* The second stage restores a real world coordinate system from the data obtained in the first stage.

***AUGMENTED REALITY AND MARKET***

AR can enhance product previews such as allowing a customer to view what's inside a product's packaging without opening it. AR can also be used as an aid in selecting products from a catalogue. Scanned images of products can activate views of additional content such as customization options and additional images of the product in its use. AR is used to integrate print and video marketing. Printed marketing material can be designed with certain "trigger" images that, when scanned by an AR enabled device using image recognition, activate a video version of the promotional material. A major difference between Augmented Reality and straight forward image recognition is that you can overlay multiple media at the same time in the view screen, such as social media share buttons, in-page video even audio and 3D objects. Traditional print only publications are using Augmented Reality to connect many different types of media.

***OPENCV***

OpenCV (Open Source Computer Vision) library. OpenCV is a library of programming functions for real time computer vision applications. This library provides a comprehensive computer vision infrastructure and thereby allows users to work at a higher abstraction layer. Additionally, library functions are optimized for fast and efficient processing. OpenCV uses an xml file that contains all of the characteristics to detect the face. This xml file is read by the function “CV Haar Detect Objects” and it is compared with a region of interest of the input image and the classifier returns 1 if the object is detected, 0 otherwise. If every simple classifier is positive, the cascade classifier is positive, otherwise it is negative.

* 1. **BENEFITS**
* Users can virtually try out the jewels.
* Users do not want to adjust with the jewel position, Jewels are aligned correctly with the user position and pose.
* Jewels can move and fold realistically.
* Jewels can be realistically rendered into the environment with ambient lighting.
  1. **OTHER APPLICATIONS**

The applications of Augmented reality are limitless. Some of them are,

* Textile shops
* Saloons
* Optical stores
* Decorative purposes

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 EXISTING SYSTEM**

The existing systems are Virtual Trail rooms and Virtual trail websites. The virtual trail rooms are considered as a worthless. Because the Virtual Trail room needs expensive accessories such as Microsoft kinetic sensors for tracking the user’s pose accurately, and it also needs an expensive game engine for rendering the virtual clothes or jewels on the user. For the realistic rendering of jewels, it also needs ambient light sensors.

Several commercial products exist for VFR implementation. Styku presents a body scanner that creates a complete 3D model of the user. This 3D model is then used in other web pages to try the clothing items on. The model can be rotated, it can match any size and it even uses a color map to analyze the fit. The body scanning is implemented using Microsoft’s Kinect and Asus’ Xtion devices. A VFR implementation by JCPteen gets an image of the user and using adobe flash player displays the clothing items. At the beginning, it shows a shadow on the screen where users have to fit themselves and after that the cloth is displayed. In this system if the user is moving, the item won’t follow or track him. Zugara offers a VFR that is similar to the JCPteens since the items don’t move once they are displayed. It is based on the augmented reality concept. The VFR doesn’t consider the proportions of the user, only shows how it looks as a fixed template. Similarly, Swivel is labeled as a try-on system that let users to see how clothes and accessories look on them in real-time. On the Ray-Ban web page, there is a Virtual Mirror where a user can see how the glasses fit on him. If the user turns his head, the model fits the glasses. At home or in store, Swivel doesn't just replicate the excitement-inducing, sales-boosting brand interaction of a traditional fitting room, it also adds unique capabilities that make it a home run for any retailer.

At the beginning the user has to match the face within a shape and position the eyes in a line that it is shown so it takes references of the head. After that it displays the model of glasses that have been chosen. On the Google Play there is one app for Android mobile devices, Divalicious, called itself as a virtual dressing room with more than 300 brands. It works by changing the clothes of a default model. Finally, there is AR-Door which has also has a product based on Microsoft Kinect. With this system, the camera tracks the person’s body and a 3D copy of clothing is superimposed on top of the users’ image.

In the virtual trail websites, user needs to enter into the websites in order to access any information. Moreover the user pose tracking in these websites are based on markers (i.e) the user needs to provide the markers to the system. Based on these markers the jewels are rendered over the image which is provided by the user. No one can easily access these system without prior knowledge. But what if the system is available within our hands and can be accessible easily by anyone.

**2.2 PROPOSED SYSTEM**

The proposed system will offer the user to try out jewels virtually with their hand held devices. The key difference in our approach is the lack of any proprietary hardware components or peripherals. Proposed VFR is software based and designed to be universally compatible as long as the device has a camera. For the Android application, the minimum API version supported is 4.4(KitKat). Additionally, proposed algorithm can track and resize the jewels according to user’s spatial position.

In order to create the Android app, we have developed a human-friendly interface which is defined as an interactive computing system providing the user an easier way to communicate with the machines. In particular, this can be achieved through touch-screen operations and gestures similar to what people naturally feel with their five senses. Creating intuitive interfaces with a few buttons that illustrate the basic functionality to the user is paramount for the wider acceptance of the virtual reality applications. This was one of the key objectives of this study.

First step of the proposed VFR method is the acquisition of the shape of the face to get reference points. Reference points are then used to determine where to display the jewels. In order to obtain the face, we applied several techniques:

* HaarCascade Classifier
* Motion detection or skeleton detection wherein multiple frames were analyzed for any movement

In this system, we get the frame continuously from the camera and it undergoes to the process. The face detection is made with a sum of these detected samples in a predefined position. Once the face is detected a rectangle is plotted around the face and the location of a reference point is chosen based on the supposition that the neck is placed at the middle of the head, half of the rectangle’s height. Also it is taken that it measures approximately a third of the head height, hence a third of the rectangle height. Note that only one face will be detected and it will be the one closest to the camera. If the face is detected, then it will show the jewel on the neck of the user. Users are able to select a jewel and they can adjust the position of the jewel. In order to change the jewel, next and previous image of the jewel is selected using the red cap (red pixel) moving to the specified region.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 FEASIBILITY STUDY**

A feasibility study is a study made to decide whether or not the proposed system is worthwhile. It involves forecast production of perishable goods. It involves in testing the application for its implementation.

**3.1.1 TECHNICAL FEASIBILITY**

The technical feasibility is the study of the configuration of the system. While studying the system configuration following things are to be studied,

* The exact configuration of the system
* The required number of workstations
* The different ways of interconnection
* Speed at which the input is given and the speed at which the output has to be generated

The tender document can be prepared using this study. Then the manufacturer or dealer can provide the equipment based on desired logical needs. It is necessary to examine more than one configuration with the same key technical aspects so that the choice of better one is made. The technical feasibility study is supposed to be most important and different study.

**3.1.2 OPERATIONAL FEASIBILITY**

The operational feasibility is based on the human factors and political aspects. It can be performed by answering following questions,

* The changes made with the system
* The factors of disturbing organizational structure.
* The new skills that are required for the improvements in operation.

The operational infeasibility does not lead to rejection of the project. The operational feasibility study is made by small group of people who are involved in the project analysis and design activity.

**3.1.3 ECONOMIC FEASIBILITY**

This kind of feasibility study is done for the cost or benefit analysis. In this study of benefits of the proposed system are identified and the corresponding costs are determined. The purpose of the economic feasibility is to determine the positive economic benefits to the organization that the proposed system will provide.

**3.1.4 MANAGEMENT FEASIBILITY**

The management feasibility means checking whether the management will accept the proposed project or not. If top level management does not agree upon the project idea then it is considered as a non-feasible project. It mainly focus on meeting the requirements specifies and also analysis of the potential of the proposed project which is based on extensive investigation and research to support the process of decision making.

**3.1.5 LEGAL FEASIBILITY**

The legal feasibility means finding out whether the proposed project is legally acceptable or not. A data processing system must comply with the local Data protection Acts.

**3.1.6 TIME FEASIBILITY**

The time feasibility means identifying whether the proposed project will be completed within the stipulated time or not. If the project runs for a long time then it is considered to be feasible project.

**3.1.7 SOCIAL FEASIBILITY**

Whether the project will be accepted by the people or not is called a social feasibility. It aims to rationally uncover the strengths and weaknesses of an existing business or proposed venture.

**3.2 HARDWARE USED**

* Processor : Intel Pentium IV and above
* RAM : 512 MB of RAM
* Hard Disk : 40GB of hard disk space
* Modem : 2-3 Mbps.
* Android Mobile : OS 4.4(Kit Kat)

**3.3 SOFTWARE USED**

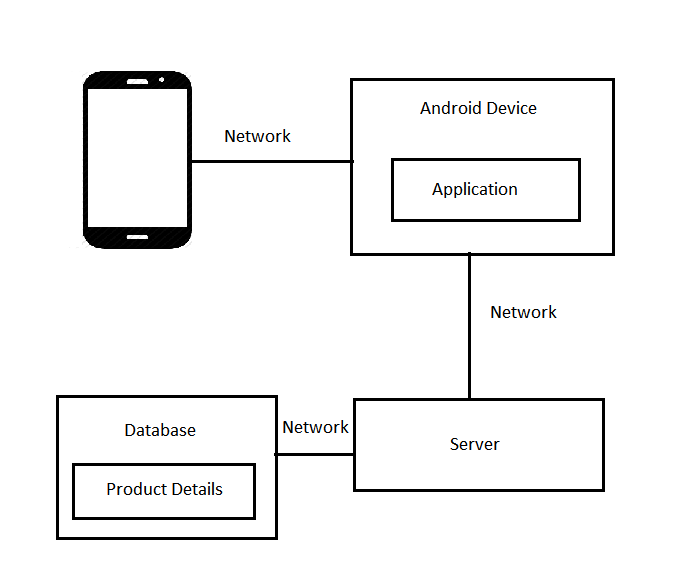
* Operating System : Windows 7/8, Linux
* Coding Language : C++, JAVA
* Software : Eclipse
* Platform : OpenCV (Linux-Library)

**CHAPTER 4**

**DETAILED DESIGN**

**4.1 SYSTEM ARCHITECTURE**

The Architecture explains the process involved in the system. The architecture of the project consists of three core end.

**Fig 4.1 SYSTEM ARCHITECTURE**

The system architecture consists of a mobile with an android application,

a server with Haarcascade classifier and a database in which all the product details are present.

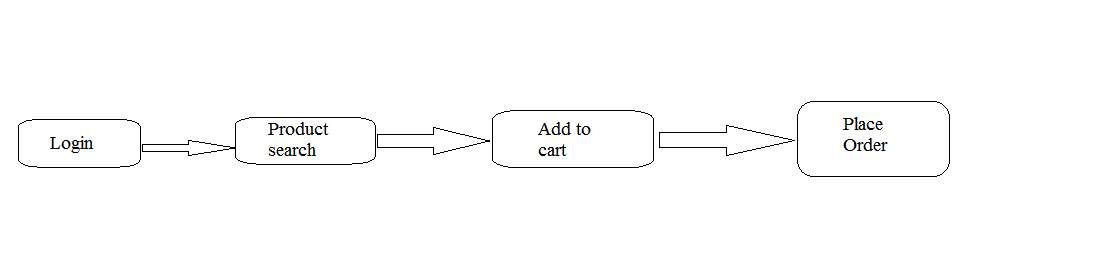
**4.1.1 MODULES**

* Front-End Design Module
* Back-End Design Module
* Connection Module

**FRONT-END DESIGN MODULE**

This module allows the user to view the following

* Login
* Categories
* Search
* Cart
* Wish list
* Try on

****

**Fig 4.2 OVERVIEW OF FRONT END DESIGN MODULE**

**Login –** It is provided for the Authentication process. The user have to provide their Username and Password in order to book or manage their orders.

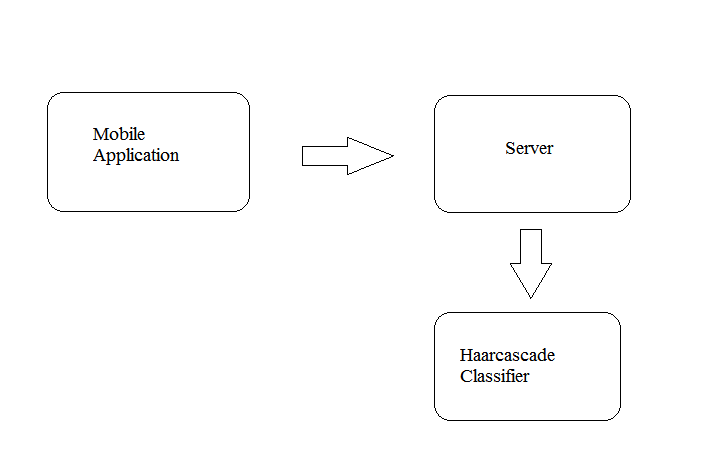
**Categories –** It provides a list in which the products are categorized.

**Cart –** The products must be added to the cart for placing the order.

**Wish list –** If the user wishes to buy some products later, then they can add those products to the wish list. Later they add it to the cart and place the order.

**Try On –** This allows the user to trail the jewels virtually using their mobile phone.

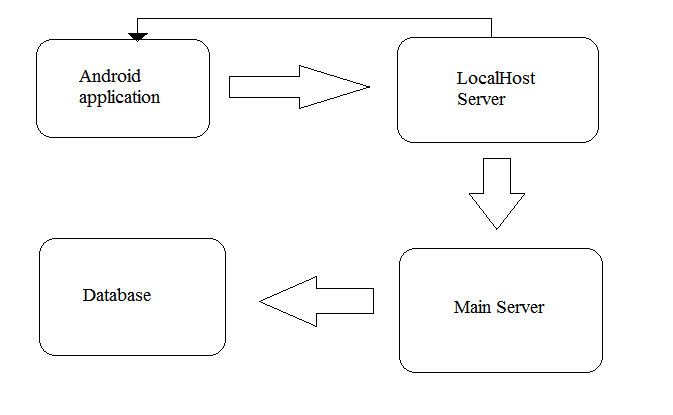
**BACK-END CREATION MODULE**

In this module, the server is created using the Opencv library and the haar-cascade classifier is added to detect the faces. Once the face is detected, then the jewels are rendered over the face and it is streamed to the mobile application.

**Fig 4.3 OVERVIEW OF BACK END DESIGN MODULE**

**CONNECTION MODULE**

This module is the core module of this application, which focus on streaming the user’s pose to the server and stream the rendered image back to the mobile phone from the server. This module plays a vital role in placing the order and managing the orders.

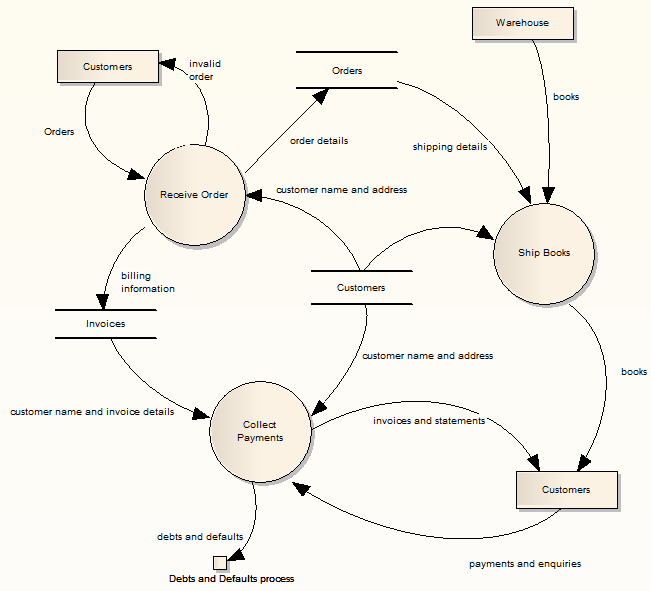


**Fig 4.4 OVERVIEW OF CONNECTION MODULE**

**4.2 FLOW DIAGRAM**

A data flow diagram is graphical tool used to describe and analyse movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams.

The development of DFD’s is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists of a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.



**Fig 4.5 FLOW DIAGRAM**

**4.3 UML DIAGRAMS**

The Unified Modelling Language (UML) is the industry-standard modelling language used for specifying, visualizing, constructing and documenting the artifacts of a software system. The UML is also effective for modelling business and of the Non-software systems.

One of the major goals of UML is to present a common modelling language that all developers can use. It is a language whose vocabulary and rules focus on the conceptual and physical representation of the system.

The UML uses Diagrams to represent different views of the systems being modelled. The purpose of diagram is to present a set of modelled elements, which are rendered as shapes and connectors.

The UML defines nine graphical diagrams:

1. Class Diagram
2. Object Diagram
3. Use- Case Diagram
4. Sequence Diagram
5. Collaboration Diagram
6. State-chart Diagram
7. Activity Diagram
8. Component Diagram
9. Deployment Diagram

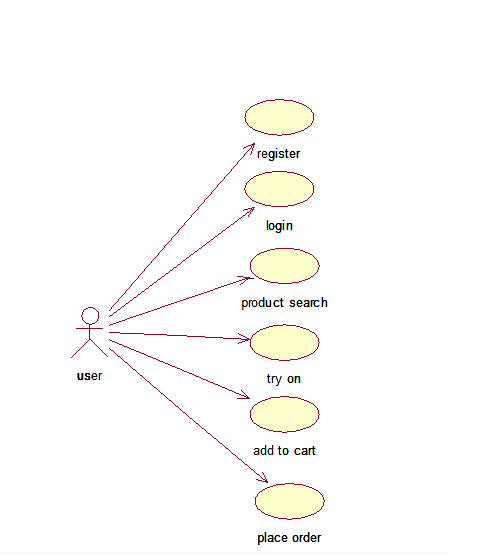
**4.3.1 USE CASE DIAGRAMS**

A use case is a description of a set of sequences of actions, including variants that a system performs to yield an observable result of value to an actor. A use case involves the interaction of actors and the system Actors (admin) can be human are they can be automated systems. It is the simplest representation of users interaction with the system and depicting the specification of a use case. Use case diagram can portray the different uses of the system and the various ways that can interact with the system.

Use case diagram is typically used in conjunction with textual use case and the other UML diagrams as well. It might drill into a lot of detail about every possibility and it can provide higher level view of the system. Simply, it can be the blueprint for the system.

**I. USE CASE DIAGRAMS OF USERS**

It is a description of a set of sequences of actions that are related to user activities.

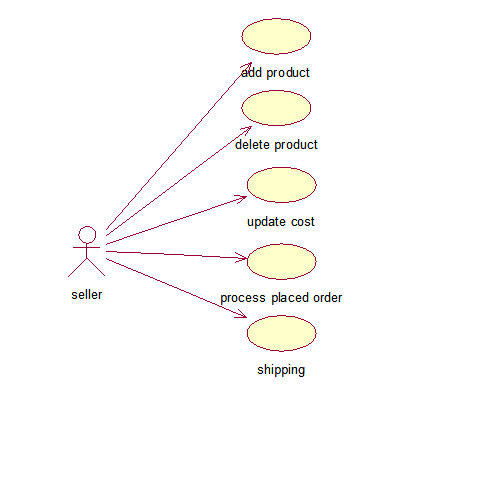


**Fig 4.6 USE CASE DIAGRAM OF USER**

Fig 4.6 represents the use case diagram for the user who wishes to buy the products through online.

**II.USE CASE DIAGRAMS OF SELLERS**

It is a description of a set of sequences of actions that are associated with the activities of sellers.



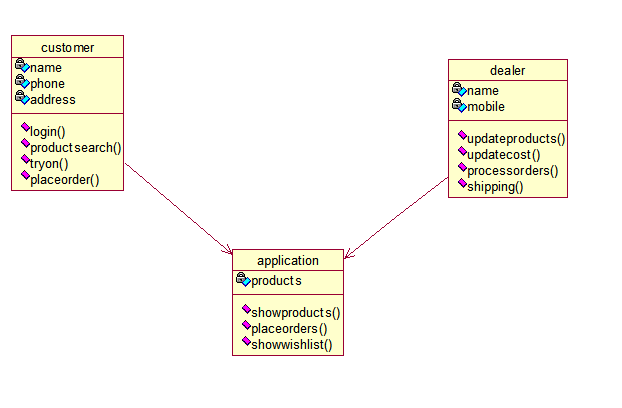
**Fig 4.7 USE CASE DIAGRAM OF SELLER**

Fig 4.7 represent the use case diagram for seller who can manage the list of products details in the server

**4.3.2 CLASS DIAGRAM**

Class diagram are the most common diagram found in modelling object-oriented systems. A class diagram shows a set of classes, interfaces, and collaborations and their relationships.

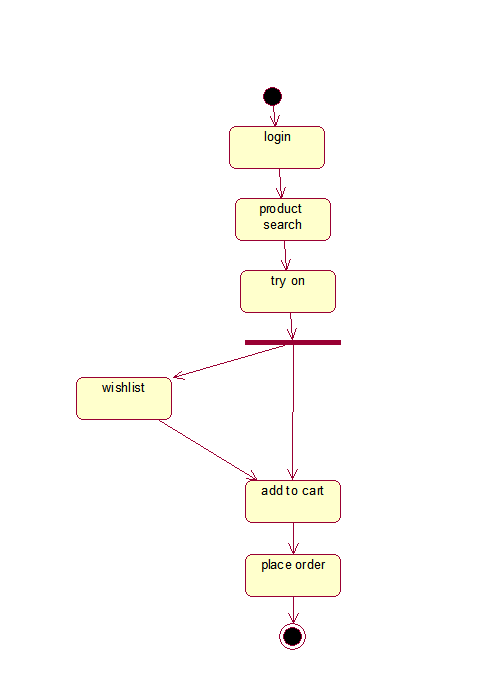
This diagram depicts the individual class and interaction among various classes. It also represents the attribute and methods to be implemented in each class. A class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects.



**Fig 4.8 CLASS DIAGRAM**

**4.3.3 ACITIVITY DIAGRAM**

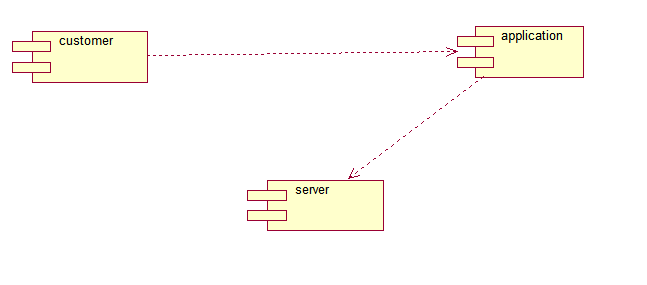
Activity diagram represents the performance of operations and transitions are triggered by completion of operations.



**Fig 4.9 ACTIVITY DIAGRAM**

**4.3.4 COMPONENT DIAGRAM**

Component diagrams model the physical components like source code, executable program user interface in a design. It describes the applications executable form and mainly focus on layouts connectivity with the Database.

****

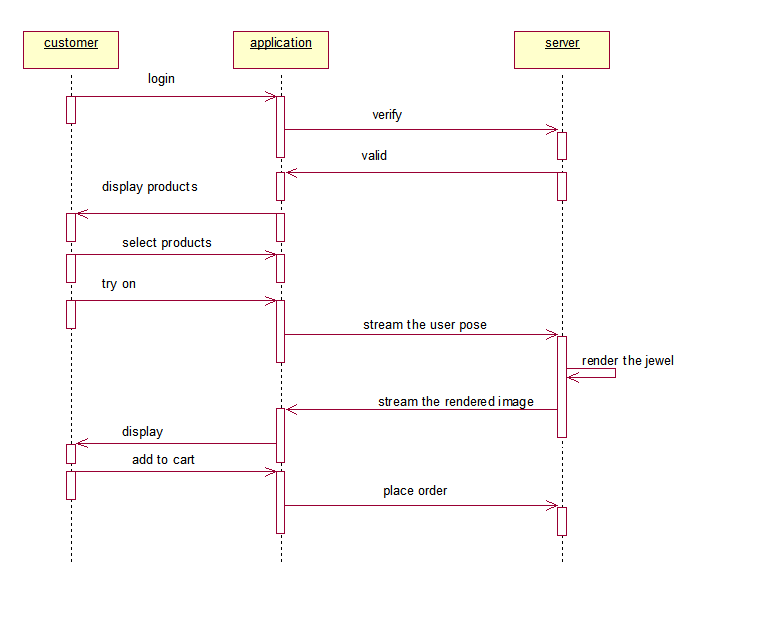
**Fig 4.10 COMPONENT DIAGRAM**

**4.3.5 SEQUENCE DIAGRAM**

An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.

A sequence diagram is an interaction diagram that shows how processes operate with the another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows the object interaction arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of message exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use diagrams are sometimes called event diagrams, event scenarios.

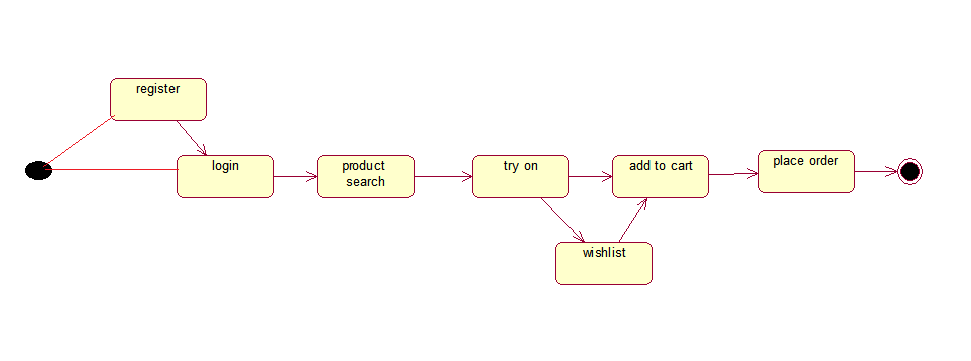
A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



**Fig 4.11 SEQUENCE DIAGRAM**

**4.3.6 STATE-CHART DIAGRAM**

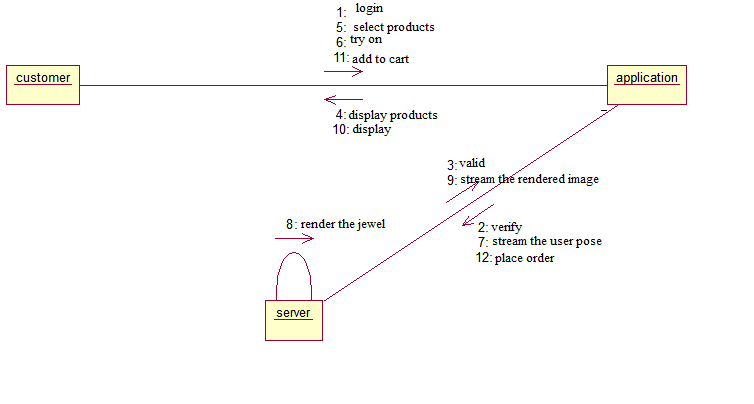
A state chart diagram include a sequence of states that an object goes through its life in response to outside stimuli message. Fig 4.12 represents the state-chart diagram. In this diagram each rectangular box represents the various possible states and flow of data among the states. It also depicts the start and end states during the entire gaming activity.

****

**Fig 4.12 STATE CHART DIAGRAM**

**4.3.7 COLLABORATION DIAGRAM**

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Graphically, a collaboration diagram is a collection of vertices and arcs. Fig 4.13 represents the collaboration diagram. This diagram depicts the interaction among the class and the information passed between the classes. The arrow and data suggest the information to pass among the classes.



**Fig 4.13 COLLABORATION DIAGRAM**

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

**5.1 IMPLEMENTATION**

This project deals with the interaction between the sellers and the users. It is an android based application. The very beginning step of our project is that user needs to login. If the user do not have the login ID, then the user have to register. Once the user logged in, then the list of products available are displayed to the user. If the user likes to trail some of the jewels, then the user is provided with the “Try On” button, which is used to track the user’s pose using the phone’s camera.

When the Try On button is clicked, then the user’s pose is tracked by the camera, then it is streamed to Augmented server. Next, the server will obtain the reference points from the user using the Haarcaacade classifier. Then the virtual jewels are rendered over the user’s image and it is streamed back for the user’s display.

***MODULES:***

1. Front-End.
2. Middle-ware.
3. Back-End.

***FRONT-END***

The Front-End comprises of the Application where the client request of the content based retrieval from the database.

**Login –** It is provided for the Authentication process. The user have to provide their Username and Password in order to book or manage their orders.

**Categories –** It provides a list in which the products are categorized.

**Cart –** The products must be added to the cart for placing the order.

**Wish list –** If the user wishes to buy some products later, then they can add those products to the wish list. Later they add it to the cart and place the order.

**Try On –** This allows the user to trail the jewels virtually using their mobile phone.

***MIDDLE-WARE***

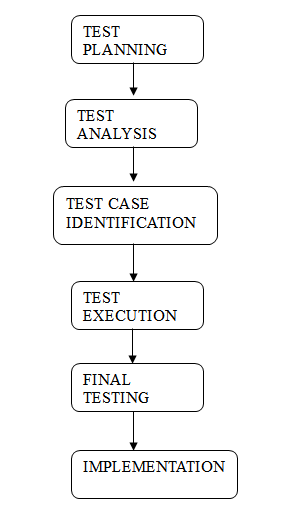
The middle-ware is the core part of the architecture encompasses of API and interaction among the webpage. The webpage request for the data from the back end and it is displayed in the specified webpage.

***BACK-END***

The Back-End is the server part where the face detection process occurs using the haar-cascade classifier and the virtual jewels are rendered over the detected face.

**5.2 TESTING**

Testing is an important phase that focuses on an empirical investigation in which the results describe the quality of the system. It cannot confirm system functions properly under all conditions but can establish that it fails under specific conditions.



**Fig 5.1 PROCESS OF TESTING**

**5.2.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

***FRONT END TEST CASES***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Name** | **Input** | **Actual Value** | **Expected Value** | **Result** |
| Login | Specified Username | Provide Username | Provide Username | No error |
| Password | Database Value | Retrieve the stored value | Retrieve the stored value | No error |

**TABLE NO. 5.1 FRONT END TEST CASES**

***BACK END TEST CASES***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Name** | **Input** | **Actual Value** | **Expected Value** | **Result** |
| Referal points | User’s image | Provide referral points | Provide referral points | No error |

**TABLE NO. 5.2 BACK END TEST CASES**

***TEST STRATEGY AND APPROACH***

Field testing will be performed manually and functional tests will be written in detail.

***TEST OBJECTIVES***

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

***FEATURES TO BE TESTED***

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**5.2.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

***TEST RESULTS***

All the test cases mentioned above passed successfully. No defects encountered.

**5.2.3 FUNCTIONAL TESTING**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**5.2.4 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

***TEST RESULTS***

All the test cases mentioned above passed successfully. No defects encountered.

**5.2.5 SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**5.2.6 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level

**5.2.7 BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**5.3 TEST PLAN**

The project is tested to verify its correctness and identify the bugs. The test plan includes the various test cases that acts as the set of conditions or variables that determine whether the corresponding feature in the system is working as it originally established to do so. When this test plan is executed, the errors spotted are rectified and the final testing yields following result.

**5.4 TEST ANALYSIS**

In this phase of testing, the requirements for software testing are analysed and later its feasibility is determined. In the feasibility study the possibility of project development is found through suitable test cases.

**5.5 RESULT**

The application is tested and found to function as expected with no errors. This application provides an interface for the users to book the ticket in an efficient way. Thus the ticket is booked as per the user’s request using speech recognition.

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

This project presents augmented reality application where in the users are made to try out the jewels that is rendered on a screen over the image of the user. The lightning is adapted to match the intensity of the user’s environment. The jewels are properly aligned according to the user’s positions and movements. The system is an improvement to the existing system where the tracked user is able to try jewel that include jewel simulation and can be viewed from different angles and the jewels acts as real.

**6.2 FUTURE ENHANCEMENT**

As enhancements to this Application, social networking features can be added, such as sharing on social networking platforms, e-mailing a snapshot to a friend or uploading the snapshot somewhere for friends and family to comment. Also, to expand to try out, accessories such as hats, shoes, clothes, bags etcetera can be included. Moreover adapting the illumination of the jewel to the lighting conditions of the captured video stream of the real world could be an useful addition in regard to realism as well. One final enhancement is concerning about the resolution can be meaningful. Because the current resolution of 320 \* 240 is quiet low which results in bad quality of the recorded scene.

**APPENDIX A**

**SAMPLE SOURCE CODING**

**AR.CPP**

#include <opencv2/opencv.hpp>

#include <string>

#include <iostream>

#include <vector>

#include <stdio.h>

#include <time.h>

using namespace cv;

using namespace std;

CvFont font;

CvSeq\* face;

IplImage\* hsv;

Point2f p[4],q[4];

IplImage\* img=NULL;

int found, scale = 1;

cv::Point2f v[4],b[4];

unsigned int j=0, i=0;

int line\_type = CV\_AA; // change it to 8 to see non-antialiased graphics

static CvMemStorage\* storage = 0;

CvPoint pt1, pt2,pt3,pt4,pt5,pt6,pt7;

const int boardWidth=5, boardHeight=4;

Mat overlay, perspMat, pic,frame, templ;

static CvHaarClassifierCascade\* cascade = 0;

const Size boardSize = Size(boardWidth,boardHeight);

const char \*cascade\_name="haarcascade\_frontalface\_alt.xml"; //semi-plam

void merge(cv::Mat &in1, cv::Mat &in2,cv::Mat &out)

{

int nLines = in1.rows;

int nc = in1.cols \* in1.channels();

if(in1.isContinuous())

{

nc = nc\*nLines;

nLines = 1;

}

for(int j=0;j<nLines;j++)

{

uchar\* dataIN1 = in1.ptr<uchar>(j); //fajne :)

uchar\* dataIN2 = in2.ptr<uchar>(j);

uchar\* dataOUT = out.ptr<uchar>(j);

for(int i=0;i<nc; i++)

{

if(dataIN2[i] == 0)

{

dataOUT[i] = dataIN1[i];

}

else

{

dataOUT[i] = dataIN2[i];

}

}

}

}

int main()

{

int b=1,x=0,y=0,c=0,c1=0,a=0,w=0,N=0,L=0;

// CvCapture \*capture = cvCaptureFromCAM(0);

CvCapture \*capture = cvCreateFileCapture("rtsp://10.0.0.90:8086/");

cvSetCaptureProperty(capture, CV\_CAP\_PROP\_FRAME\_WIDTH, 320);

cvSetCaptureProperty(capture, CV\_CAP\_PROP\_FRAME\_HEIGHT, 240);

cvSetCaptureProperty(capture, CV\_CAP\_PROP\_FPS, 10);

/\* CvSize size = cvSize(

(int)cvGetCaptureProperty( capture,

CV\_CAP\_PROP\_FRAME\_WIDTH),

(int)cvGetCaptureProperty( capture,

CV\_CAP\_PROP\_FRAME\_HEIGHT)

);\*/

// CvVideoWriter \*writer = cvCreateVideoWriter("out.jpg",CV\_FOURCC('M','J','P','G'),10,size);

vector<Point> corners;

Point2f p[4],q[4];

overlay = Mat::zeros(319,239,CV\_8UC3);

storage = cvCreateMemStorage(0);

cvClearMemStorage( storage );

cascade = (CvHaarClassifierCascade\*)cvLoad( cascade\_name, 0, 0, 0 );

pic = cv::imread("N1.png");

flip(pic,pic,0);

flip(pic,pic,1);

while(1)

{

c=c1=0;

img = cvQueryFrame( capture );

if ( !img )

{

printf("Connect Camera.....\n");

break;

}

if ( L == 3 )

{

for( y=0; y<10; y++ )

{

L=0;

img = cvQueryFrame( capture );

cvShowImage( "Frame", img );

cvWaitKey(33);

}

}

Mat src(img);

if(!img)break;

face = cvHaarDetectObjects( img, cascade, storage,1.1, 2, CV\_HAAR\_DO\_CANNY\_PRUNING,cvSize(40, 40));

for( i = 0; i < (face ? face->total : 0); i++ )

{

CvRect\* r = (CvRect\*)cvGetSeqElem( face, i );

found =10;

pt1.x = r->x\*scale+50 ;

pt2.x = (r->x+r->width)\*scale+50;

pt1.y = r->y\*scale ;

pt2.y = (r->y+r->height)\*scale ;

//cvRectangle( img, pt1, pt2, CV\_RGB(0,0,0), 3, 8, 0 );

}

pt7.x=20;pt7.y=10;

cvInitFont( &font, CV\_FONT\_HERSHEY\_COMPLEX, 0.3,0.3 );

cvPutText( img, "Next", pt7, &font, CV\_RGB(255,255,0));

cvRectangle( img, cvPoint(0,0),cvPoint(65,20), CV\_RGB(0,255,0), 2, 8, 0 );

pt7.x=280;pt7.y=14;

cvInitFont( &font, CV\_FONT\_HERSHEY\_COMPLEX, 0.4,0.4 );

cvPutText( img, "Prev", pt7, &font, CV\_RGB(255,255,0));

cvRectangle( img, cvPoint(270,0),cvPoint(320,20), CV\_RGB(0,255,0), 2, 8, 0 );

pt7.x=96;pt7.y=14;

cvInitFont( &font, CV\_FONT\_HERSHEY\_COMPLEX, 0.4,0.4 );

cvPutText( img, "Augment Reality", pt7, &font, CV\_RGB(255,255,0));

pt3.x=65;pt3.y=20;pt4.x=270;pt4.y=20;//hori

cvLine( img, pt3, pt4, CV\_RGB(0,255,255), 2, 8, 0 );//bottom-hor

Mat hsv;

cvtColor(src, hsv, CV\_BGR2HSV);

Mat bw;

inRange(hsv, Scalar(0, 50, 170, 0), Scalar(10, 180, 256, 0), bw);//red

vector<vector<Point> > contours;

findContours(bw.clone(), contours, CV\_RETR\_EXTERNAL, CV\_CHAIN\_APPROX\_SIMPLE);

Mat dst = Mat::zeros(src.size(), src.type());

Mat dst1 = Mat::zeros(src.size(), src.type());

drawContours(dst, contours, -1, Scalar::all(255), CV\_FILLED);

unsigned int v1=0,v2=0,v3=0,v4=0,v5=0,v6=0,v7=0,v8=0;

if ( found > 5 )

{

for( y = 8; y < 13; y++ )

{

for( x = 8; x < 55; x++ )

{

a=saturate\_cast<uchar>(( dst.at<Vec3b>(y,x)[0] ));

if ( (a > 230 ) )

{

c =c+1;

c1 =0;

}

}

}

for( y = 8; y < 13; y++ )

{

for( x = 275; x < 317; x++ )

{

w=saturate\_cast<uchar>(( dst.at<Vec3b>(y,x)[0] ));

if ( (w > 230 ) )

{

c1 = c1 + 1;

c = 0;

}

}

}

prev:

if( ( c > 7 ) && ( b == 0 ) )

{

pic = cv::imread("N2.png");

b=b+1;c=0;L=3;

goto Next;

}

if( ( c > 7 ) && ( b == 1 ) )

{

pic = cv::imread("N3.png");

flip(pic,pic,0);

flip(pic,pic,1);

if (N==9){N=0;L=3;c=0;goto Next;}

b=b+1;c=0;L=3;

goto Next;

}

if( ( c > 7 ) && ( b == 2 ) )

{

pic = cv::imread("N4.png");

flip(pic,pic,0);

flip(pic,pic,1);

if (N==9){N=0;L=3;b=b-1;goto Next;}

c=0;L=3;

goto Next;

}

if( ( c1 > 7 ) )

{

b=b-1;

c1=0;

//printf("Prev.............\n");

c=10;

N=9;

goto prev;

}

Next:

found=0;

frame=img;

///printf("argument...........\n");

q[0].x= 0;

q[0].y= 0;

q[1].x= 150;

q[1].y= 0;

q[2].x= 150;

q[2].y= 180;

q[3].x= 0; //center

q[3].y= 180;

v1=pt1.y,v2=pt1.x,v3=pt2.y,v4=pt1.x,v5=pt2.y,v6=pt2.x,v7=pt1.y,v8=pt2.x;

for(i=0;i<corners.size();i++)//find the size of contour

{

cout << "# of corners points: " << corners.size() << endl ;

}

p[0].x= v1;// pt1.x;

p[0].y= v2;//pt1.y;

p[1].x= v3;//pt2.x;

p[1].y= v4;//pt1.y;

p[2].x= v5;//pt2.x;

p[2].y= v6;//pt2.y;

p[3].x= v7;//pt1.x;

p[3].y= v8;//pt2.y;

perspMat = getPerspectiveTransform(q,p);

warpPerspective(pic,overlay,perspMat,Size(frame.cols,frame.rows));

merge(frame,overlay,frame);

}

//cvWriteFrame(writer, img);

cvSaveImage("out.jpg", img);

cvWaitKey( 33);

}

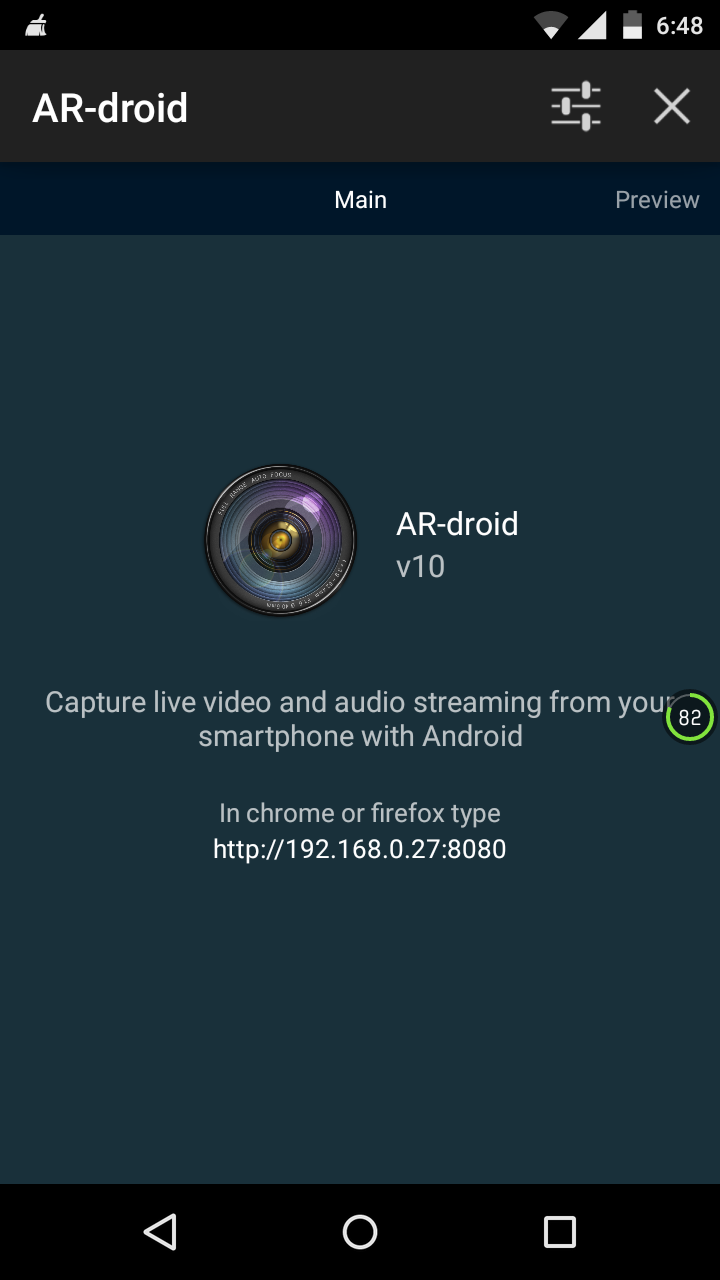
return 0;

}

**APPENDIX B**

**SAMPLE SCREEN SHOTS**

**HOMEPAGE**

****

**Fig B.1 OUTPUT SCREEN OF HOMEPAGE**

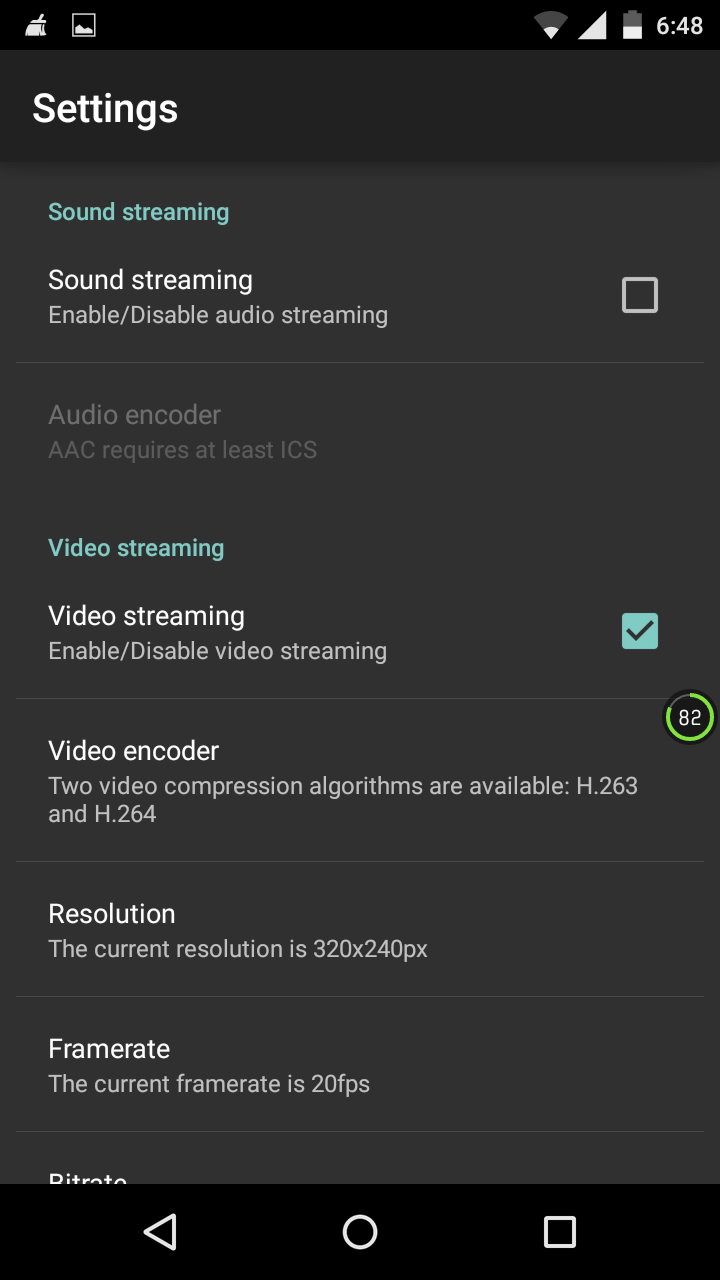
Fig B.1 represents the homepage of the android application which provides the IP address for the connectivity between the server and the mobile phone.

**PREVIEW OF CAMERA**

****

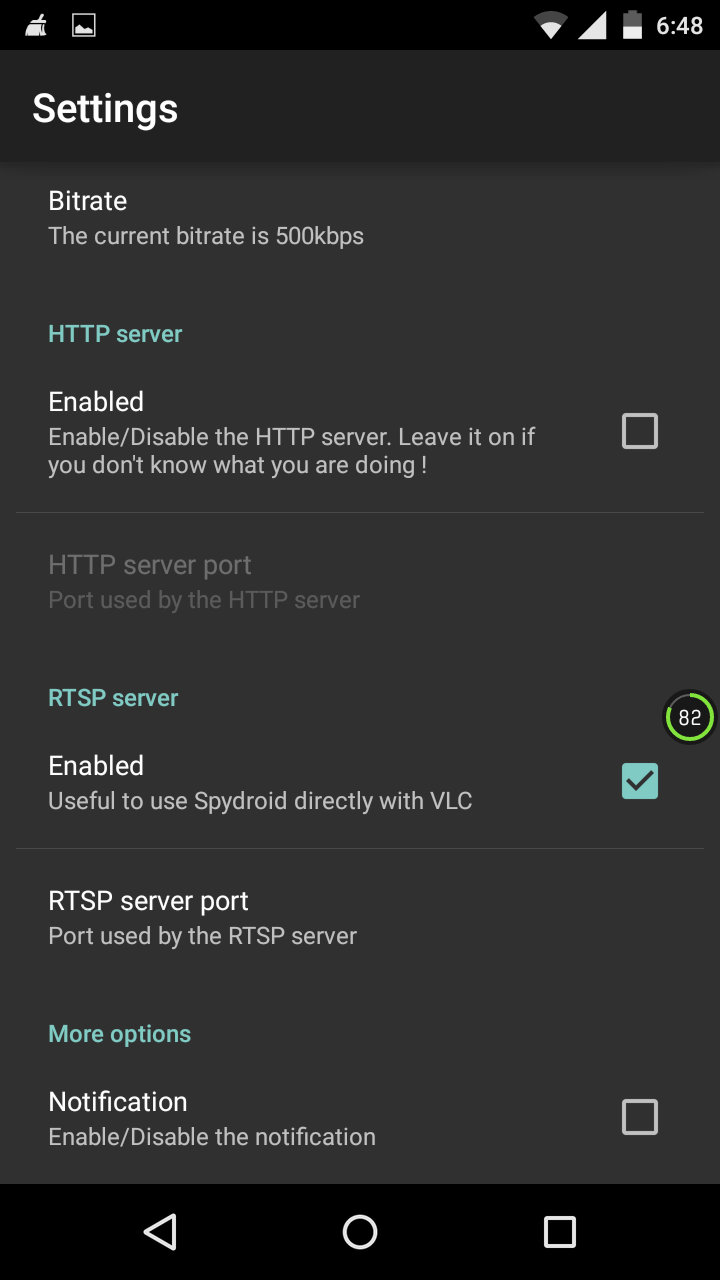
**Fig B.2 PREVIEW OF CAMERA**

**SETTINGS SCREEN**

****

**Fig B.3 (a) OUTPUT SCREEN OF SETTINGS**

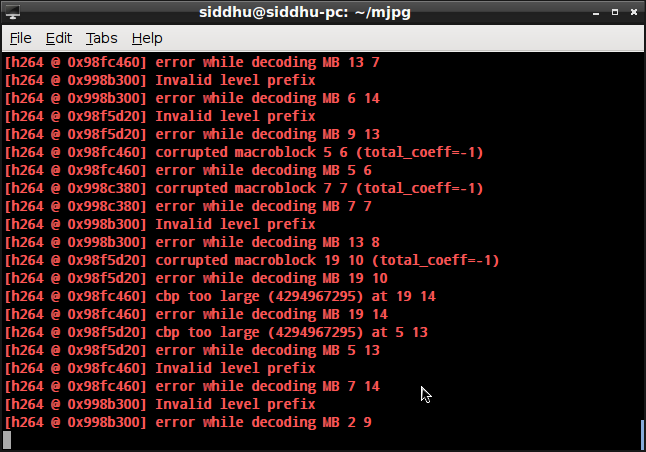
Fig B.3(a) represents the settings page of the android application which list of options such as video encoders, resolution, frame rate, etc.

****

**Fig B.3 (b) OUTPUT SCREEN OF SEETINGS**

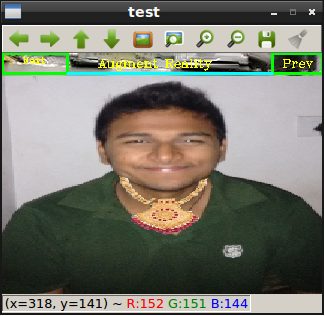
Fig B.3(b) represents the settings page of the android application which provides the options such as bitrate, server options and notifications.

**SERVER OUTPUT**

****

**Fig B.4 (a) OUTPUT IN THE SERVER**

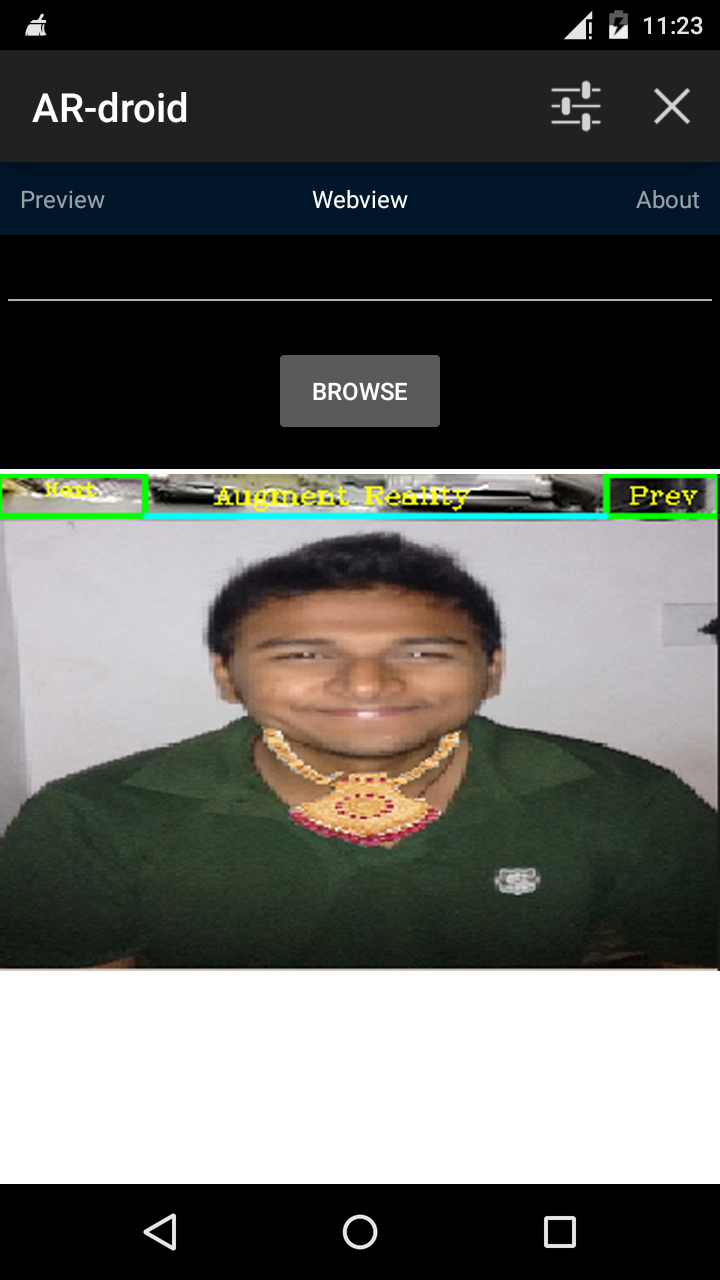
Fig B.4 (a) represents the execution of code from the server side which is responsible for rendering the jewel.

****

**Fig B.4 (b) OUTPUT IN THE SERVER**

Fig B.4 (b) represents the output from the server side where the jewel is rendered over the user image.

**WEBVIEW**

****

**Fig B.5 OUTPUT FROM THE SERVER**

Fig B.5 represents the output from the android application where the jewel is rendered over the user image.

**REFERENCES**

1. *N. Kubota, D. Koudu* and *S. Kamijima*, “Human-Friendly Interface Based on Visual Attention for Multiple Mobile Robots,” Automation Congress World, Budapest, 24-26 July 2006, pp. 1-6.
2. *C. Chai* and *Y. Wang*, “Face Detection Based on Extend- ed Haar-Like Features,” International Conference on Mechanical and Electronics Engineering (ICMEE), Kyoto, 1-3 August 2010, pp. 442-445.
3. *M. Zuo, G. Zeng* and *X. Tu*, “Research and Improvement of Face Detection Algorithm Based on the OpenCV,” International Conference on Information Science and En- gineering (ICISE), Hangzhou, 4-6 December 2010, pp. 1413-1416.
4. *D. Lee*, “A Face Detection and Recognition System Based on Rectangular Feature Orientation,” International Conference on System Science and Engineering (ICSSE), Taipei, 1-3 July 2010, pp. 495-499.
5. Intel Inc., “Open Computer Vision Library,” 2011. http://opencv.org
6. “Haar-Like Feature Face Detection,” 2011. <http://opencv.willowgarage.com/wiki/FaceDetection> – 14-Feb-2014.
7. Skytu, 2011. [http://www.styku.com/business -](http://www.styku.com/business%20-) 23-Feb-2014.
8. JCPteen, 2011. <http://seventeen.ar-live.de/fall/motion> - 08-Mar-2014.
9. Zugara, 2011. <http://zugara.com> – 12-Mar-2014.
10. Swivel, 2011. <http://www.facecake.com/swivel/index2.html> - 13-Mar-2014.