Augmented Reality Mobile Application for Overlaying Images on Textiles

Veronica Brinza

Madrid, July 2016
This thesis is submitted to the ETSI Informáticos at Universidad Politécnica de Madrid in partial fulfilment of the requirements for the degree of Master of Science in Software Engineering

Master Thesis
Master Universitario en Ingeniería del Software – European Master in Software Engineering
Thesis Title: Augmented Reality Mobile Application for Overlaying Images on Textiles
Thesis no: EMSE-2016-06
July, 2016

Author:
Veronica Brinza
Licentiate in Exact Sciences
Universitatea Tehnică a Moldovei

Supervisor:
Angelica De Antonio Jimenez
Ph.D. in Computer Science
Universidad Politécnica de Madrid
Lenguajes, Sistemas Informáticos e Ingeniería de Software
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

ETSI Informáticos
Universidad Politécnica de Madrid
Campus de Montegancedo, s/n
28660 Boadilla del Monte
(Madrid)
Spain
Introduction

Augmented reality is one the relatively new and fast growing branches in modern informatics. It incorporates in simple people lives on a lot of different levels. It can be used everywhere starting from the simple checking of your email and ending up with a complex augmented reality video games or augmented reality learning. From the ancient times actors tried to immerse the spectators in world of the performance using decoration and costumed. In XVII – XIX centuries pictures with different types of blur were used to create three-dimensional image from the point of view of the spectator. So, as you can see the concept of augmented reality is not that new, but it became more and more popular only in last decades. This happened due to the progress in modern technologies. Better resolution of cameras and more powerful processors on the computers and phones made the augmented reality applications be done easier, faster and made them more accessible for the end user. Fast evolution of smart phones and their relatively acceptable price made them one of the most popular targets for augmented reality applications. That’s why exactly development of and android application with augmented reality was chosen for this diploma work.

This diploma work presents the whole process of a software project developing starting from creating software specification and finishing with testing and in production quality management. In the beginning a quick overview of augmented reality is presented to introduce the reader in the world of augmented reality and make him familiar with notions connected to this topic. Chapter two represents the process of software specification for the current (first) and future versions of the application. Chapter three represents development plan, where the are presented hour dedication of each participant of the project and the development time line. The hours dedication is calculated based on the planning poker and based on function point analysis after which the price of the project effort is calculated based on the amount of hours and salary per hour. After separation of the task in sprints following the agile development technique Scrum is presented. Chapter four overviews the measures that were taken in order to assure quality of the project. Chapters five tells about all the stages of the design development and usability testing of low fidelity and high fidelity prototype and analysis of usability testers opinion. Chapter six describes the full development phase starting with initial project creating set-up, adding every module one by one accompanied by UML diagram of classes and description of all algorithms and libraries used and screen-shots of the application after including every module. Last chapter includes the description of all automatic and manual tests to be performed in order to control software quality. This chapter also includes the description of ways to ensure the software is working good in production phase, in other words, when it is published in google play market.

As a result of this master thesis an android application for a company that sells T-shirts was created. This application is created in order to make the process of wearing these T-shirts more fun by applying augmented reality to the T-shirts.
Chapter 1
Augmented Reality

1.1 What Augmented Reality Is?

Augmented reality is not a new, but steal a young concept in scientific world. Nevertheless it gains more and more popularity between the people. So what augmented reality is? Augmented reality - an enhanced image or environment as viewed on a screen or other display, produced by overlaying computer-generated images, sounds, or other data on a real-world environment. [1] First augmented reality application was made around 50 years ago, but now it used in a lot of areas as learning, gaming, cinematography, merchandising etc. Nowadays augmented reality is a way of receiving new contextual information at the moment when you mostly need it.

Talking about the notion of augmented reality we can’t miss mentioning virtual reality. Both are fields in which the lines of distinction are kind of blurred. To put it another way, you can think of virtual reality as the precursor to augmented reality, with some parts overlapping in both. The main difference between the two technologies is that virtual reality does not use a camera feed. All the things displayed in virtual reality are either animations or pre-recorded bits of film. [2] On the other hand augmented reality can be considered a variety of virtual reality. [3] In the flowing table you can see the main points of interconnections and differences between augmented and virtual reality:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Augmented reality</th>
<th>Virtual reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses multimedia resources</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Provides additional info</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Influences all senses of human body</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Works in real-time</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Interferes with real world</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Replaces real world</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Uses fully immersion technologies</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 1.1.1 Difference between augmented and virtual reality systems

1.2 Elements of an Augmented Reality System

Event thought there a lot of different types of the augmented reality systems based on different algorithms all of them have some common key elements:
- **Hardware** – physical (hardware) elements of the device on which the software is run. Here we can place processor, display, camera, systems of interconnection on distance (Blue-tooth, Wi-Fi) and different sensors (GPS, magnetometers, accelerometers, gyroscopes)
- **Software** – programs installed on the device as an example application or a browser. Their aim is to join the layers of information received from the hardware with the virtual elements.
- **Recognition algorithm** – technology for recognition of the real world objects. They are based on location, spatial orientation, recognition of shapes and other attributes and characteristics of the object. Augmented reality services use various device sensors to
identify the users' surroundings. Current implementations generally fall into one of two categories — location-based or computer vision. Location-based offerings use a device's motion sensors to provide information based on a user's location. Computer-vision-based services use facial, object and motion tracking algorithms to identify images and objects. For example, being able to identify a shoe among numerous objects on a table, Google Goggles (imaged-based search), or optical character recognition (OCR). [4]

- **Content** – information which is displayed by software. There are two main types of information: information defined by the object (it supposed exchange of the information between the augmented reality system and the object like QR-code code Bar-code), “open information” (it supposes an external information source to which the augmented reality system makes request to find any information about the real-life object)

1.3 Classification of Augmented Reality Systems

Augmented reality systems can be classified by the **degree of the user participation** in it. In some of the systems user has a passive role, he is just watching the reaction of the system on the changes in real-world objects. On the other hand, there are other systems that require active participation of the user. User can control the system in order to change the final result as in order to change virtual object. So we have next types of augmented reality systems: [5]

- **Autonomous** – this kind of system are used to present additional information about the surrounding object to the user. This kind of system usually use camera to analyse the user surroundings and prove user with reference information about the objects that are placed near him. As an example: user is in a museum, he captures an image of a painting or a sculpture and the augmented reality system gives him information about this artistic object.

- **Interactive** – this kind of systems suppose interaction with the user, which can edit the output of the virtual informational layer. It is obvious that this kind of systems have to have an input device like for example a touch screen. As an example of this system we can mention interactive fitting rooms where user by interaction with the system can select an clothing item from the catalogue and and see his image in this item.

People have a lot of different senses. Since the augmented reality is an intermediary between real-world and person, it has to present information that influences one of the
human senses. So we can divide the augmented reality system according to the way of information representation: [6]

- **Visual** – in the root of such kind of systems lies human vision. The aim of a such system is to create an image that will be used by user to reach his goals. This kind of systems are really expanded because of the features of the human ways of perception information: image is one of the most clear ways to explain something to a person.

- **Audio** – systems of this type are focused on acoustical perception. Mainly this kind of systems are used fro GPS navigation. The system alerts user when he needs to change direction or when he reached the destination point.

- **Audiovisual** – this kind of systems are combination of the two previously described, even though usually audio information is used just like an additional way of representation.

Augmented reality systems are supposed to receive information about the real-life world and according to it build virtual objects. To be able to do that system can posses a big variety of sensors – elements that are used to perceive external information like sound or electro-magnetic changes, acceleration, velocity etc. For classification it makes more sense to consider the purpose of the sensor and not the physical quantity it measures, because similar by it’s nature signals may have different signification. According to the type of the sensors that are used in the augmented reality system we can make next classification: [6]

- **Geo-locational** – this kind of systems first of all take in consideration the signals from such sensors as GPS, compass, gyroscope, accelerometer etc.

- **Optical** - this kind of system use image received from one or several cameras. Cameras can move in synchronization or independently from the system itself.

According to the **mobility** of the augmented reality system we can distinguish two types of systems: [5]

- **Static** – this kind of system are supposed to work in the same place and are not made to be transported easily.

- **Mobile** – the work-flow of this system supposes that they are moved to work dynamically with the object of real-world.

In function of the **functionality** the system proposes to the user we can distinguish next types of systems: [5]

- **Visual search** – this systems provide user with navigation suggestions according to his requests. This systems are not only already known to us navigators but also systems for searching of merchandising, services, people, an other objects with known characteristics.

- **Recognition** – this system provides user with information about the objects that are places in his field of vision. As an example of a such system we can name a system that recognizes the person you are speaking to and provides you with the information about him from the open sources.

- **Human 2.0** – this kind of systems provides you with instruction how to do a concrete task. Fro example searching of a cooking recipe, with control of the amount, sequence, way and time of cooking of the ingredients or for example instruction to repair a car with list of the instruments, details necessary for reparation and the way and sequence of their usage.

- **Screen-lens** – this kind of system are combining virtual objects with real-life image, in order to understand the spatial characteristics of the virtual object. As an example of a such system can be an QR-code-catalogue of the furniture. Only by placing the QR-code in your house and taking a picture of it you can see how a certain furniture element will fit in your house.

- **Visualization of the product in the context** - such type of systems are mostly suitable for the industrial enterprises in order to take constructional and engineering decision. For example the application of augmented reality can show how will an equipment work in specific working conditions as for example how will a car prototype move in real road conditions.
Classification of the augmented reality systems [7] [8] [9] [10] [11]

- Interaction with the user
  - Autonomous
  - Interactive

- Functionality
  - Visual search
  - Recognition
  - Human 2.0
  - Screen-lens
  - Visualization of the product in the context

- Mobility
  - Static
  - Mobile

- Sensor type
  - Geo-locational
  - Optical

- Information representation
  - Visual
  - Audio
  - Audio-visual

Picture. 1.3.1 Classification of the augmented reality systems [7] [8] [9] [10] [11]
Chapter 2
Software Requirements Specification

2.1 Scope of the Product

The software is an additional feature to the line of white coloured T-shirts with a drawing of one of the 3 primitive shapes: circle, square or triangle. The software is basically an application for a device that runs android operation system. It will be used to attract more buyers to the company shops, by making of these T-shirts something more interesting using augmented reality. The application will be used to take photos of the person in the company’s T-shirt, but this picture won’t be just a simple picture, because it would be supplemented with some virtual objects added to the photo by the application.

2.2. Product Features

The main features of the application are:
1. Displaying real-life picture coming from camera
2. Analysing the picture in order to find one of the 3 primitive shapes: circle, square or triangle.
3. Displaying the virtual object corresponding to the shape found in the picture: circle – surfing-board, square – snowboard, triangle - skateboard
4. Taking picture that contains picture coming from camera combined with the virtual object (further augmented reality picture)
5. Saving augmented reality picture to the device internal memory
6. Possibility to share the obtained augmented reality picture

2.3. Hardware Requirements:

As it was mentioned the system is an application for android that leads us to the very first requirement:

R1: The system has be represented as a native application for android system.

Taking in consideration the current version of the android operation system that is android 6 (Marshmallow) we suggest that application support for the previous, current and further version of android operation system:

R2: The system has be able to work on the devices running android 5 (Lollipop) and higher versions of this operating system.

Since the application is working with images coming from the device camera the device mush have at least one camera:

R3: The device which runs the application has to have at least one camera

2.4. Functional Requirements

From the very first system feature we can deduce next requirement

R4: The system should interact with android hardware camera using standard android SDK

From the second system feature we can deduce next requirements:

R5: The system has to analyse image received from the camera hardware
R6: The system has to process the image coming from camera in real-life time
R7: The system has to detect on the picture one the primitive shapes: circle, square, triangle
**R8:** The system has to detect only primitive shapes of black colour on a white background.

**R9:** To be detected the primitive shape has to be at least 10 cm in height and width (for square), diagonal length (for square), vertical length (for triangle) and be placed on the distance between one and two meters from the camera.

From the third system feature we can deduce next requirements:

**R10:** After detecting the primitive shape the system has to replace it with a corresponding virtual object: circle – surfing-board, square – snowboard, triangle – skateboard.

**R11:** The dimensions of the virtual object have to be calculated taking in consideration dimensions of the found primitive shape.

From the fourth system feature we can deduce next requirement:

**R12:** The system has to take augmented reality picture.

From the fifth system feature we can deduce next requirement:

**R13:** The system has to save the picture taken on the internal memory of the device.

From the sixth and last system feature we can deduce next requirement:

**R14:** The system has to share the obtained picture in any of the applications installed on the device, that permit sharing option.

### 2.5 Non-functional Requirements

**Design requirement:**

**R15:** The system has to have material design according to android developers guidelines.

**Time and click restriction requirements:**

**R16:** The system should manage to process image in less the 3 seconds.

**R17:** The system has to take picture in not more then two clicks.

**R18:** The system should share the picture in not more then two clicks (not taking in consideration the amount of clicks required by the application that is going to share the picture).

### 2.6 Requirements for Future Version

For the future versions the clients asked to add next requirement:

**R19:** Possibly of processing image that is already stored in the device memory and apply searching and replacement algorithm to it.

**R20:** User has to be able to visualize all the pictures taken with the application on list or on a map.
3.1 Development Plan

In order to develop the project mentioned in the previous chapter we will need the next roles: project manager, designer, two android developers (one will be writing the application itself and another one will be writing automatic tests) and a tester (for manual testing). In the next table you can see the general description of the development phase. The values of hours was obtained by playing planning poker:

<table>
<thead>
<tr>
<th>Task</th>
<th>Responsible</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elaborating technical specification</td>
<td>Project manager</td>
<td>93</td>
</tr>
<tr>
<td>Elaborating design</td>
<td>Designer</td>
<td>120</td>
</tr>
<tr>
<td>Elaborating android app</td>
<td>Android developer 1</td>
<td>273</td>
</tr>
<tr>
<td>Elaborating automating tests</td>
<td>Android developer 2</td>
<td>120</td>
</tr>
<tr>
<td>Performing manual testing</td>
<td>Tester</td>
<td>40</td>
</tr>
</tbody>
</table>

**Table 3.1.1** Overall time dedication

In the following table you can see detailed hour repartition for technical specification elaboration:

<table>
<thead>
<tr>
<th>Task</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial meeting with the customer</td>
<td>13</td>
</tr>
<tr>
<td>Elaborating of initial technical specification</td>
<td>20</td>
</tr>
<tr>
<td>Validating of technical specification with the customer</td>
<td>20</td>
</tr>
<tr>
<td>Reworking technical specification according to customer remarks</td>
<td>20</td>
</tr>
<tr>
<td>Software quality management</td>
<td>20</td>
</tr>
</tbody>
</table>

**Table 3.1.2** Project manger time dedication

In the following table you can see detailed hour repartition for design elaboration:

<table>
<thead>
<tr>
<th>Task</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elaborating of the initial design</td>
<td>40</td>
</tr>
<tr>
<td>Validating of the design with usability testers</td>
<td>20</td>
</tr>
<tr>
<td>Analysing the usability testers feedback</td>
<td>20</td>
</tr>
<tr>
<td>Reworking the design according to focus group remarks</td>
<td>40</td>
</tr>
</tbody>
</table>

**Table 3.1.3** Designer time dedication
In the following table you can see detailed hour repartition for android application development phase:

<table>
<thead>
<tr>
<th>Task</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial creation and setup of the project</td>
<td>13</td>
</tr>
<tr>
<td>Connection of the application with camera</td>
<td>40</td>
</tr>
<tr>
<td>Implementation of the recognition algorithm</td>
<td>40</td>
</tr>
<tr>
<td>Implementation of the replacement algorithm</td>
<td>40</td>
</tr>
<tr>
<td>Saving the final result to the internal memory storage</td>
<td>20</td>
</tr>
<tr>
<td>Sharing the picture</td>
<td>20</td>
</tr>
<tr>
<td>Bug-fixes</td>
<td>100</td>
</tr>
</tbody>
</table>

*Table 3.1.4* First android developer time dedication

In the next table you can see detailed development plan for elaborating automatic tests:

<table>
<thead>
<tr>
<th>Task</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit testing of the recognition algorithm</td>
<td>40</td>
</tr>
<tr>
<td>Unit testing of the replacement algorithm</td>
<td>40</td>
</tr>
<tr>
<td>Integration testing of the application in general</td>
<td>40</td>
</tr>
</tbody>
</table>

*Table 3.1.5* Second android developer time dedication

Table 3.1.6 Gantt diagram for time line of the development [26]

### 3.2 Cost of the Project Based on Workload

Based on the numbers workers, amount of their work hours and the level of their salary we can count the cost of the effort for creation of the application:
### Table 3.2.1 Project cost based on workload

<table>
<thead>
<tr>
<th>Person</th>
<th>Hours</th>
<th>Salary per hours*</th>
<th>Total salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project manager</td>
<td>93</td>
<td>17 [12]</td>
<td>1581</td>
</tr>
<tr>
<td>Designer</td>
<td>120</td>
<td>10 [13]</td>
<td>1200</td>
</tr>
<tr>
<td>Android develop 1</td>
<td>273</td>
<td>14 [14]</td>
<td>3822</td>
</tr>
<tr>
<td>Android developer 2</td>
<td>120</td>
<td>13 [14]</td>
<td>2560</td>
</tr>
<tr>
<td>Tester</td>
<td>40</td>
<td>11 [15]</td>
<td>440</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>8603</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.3 Cost of the Project Based on Function Point Analysis

First Step: defining elements of the system:

![Diagram of Android Application modules](image)

*The salary per hour was calculated using the information provided in the sources [12] [13] [14] [15]*
Second of all let's classify system components in 5 major components of function points theory:

<table>
<thead>
<tr>
<th>External Inputs (EI)</th>
<th>User click on device screen (specifically the click user makes in order to navigate through application and that trigger specific methods of the source code to be called)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Images that come from device camera</td>
</tr>
<tr>
<td></td>
<td>Images coming from photo gallery application</td>
</tr>
<tr>
<td>External Outputs (EO)</td>
<td>UI frames shown on device screen</td>
</tr>
<tr>
<td></td>
<td>Images written into device memory</td>
</tr>
<tr>
<td></td>
<td>Images passed to another application to be shared</td>
</tr>
<tr>
<td>External Inquiry (EQ)</td>
<td>none</td>
</tr>
<tr>
<td>Internal Logical Files (ILF)</td>
<td>none</td>
</tr>
<tr>
<td>External Logical Files (ELF)</td>
<td>OpenCv android SDK (specifically wrap for logic for communication with device camera, applying the filters, searching the shapes, etc)</td>
</tr>
<tr>
<td></td>
<td>Android SDK (specifically logic for communication with Android OS using Java)</td>
</tr>
<tr>
<td></td>
<td>Android NDK (specifically logic for execution of code written in C/C++)</td>
</tr>
<tr>
<td></td>
<td>Libraries with Compiled OpenCv code in C/C++ (specifically logic for communication with device camera, applying the filters, searching the shapes, etc)</td>
</tr>
</tbody>
</table>

**Table 3.3.2 Classification of main modules of the system**

Now all the components complexity has to be ranked with label low, average, high:

<table>
<thead>
<tr>
<th>Class</th>
<th>Component</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>User click on device screen</td>
<td>low</td>
</tr>
<tr>
<td>EI</td>
<td>Images that come from device camera</td>
<td>high</td>
</tr>
<tr>
<td>EI</td>
<td>Images coming from photo gallery application</td>
<td>low</td>
</tr>
<tr>
<td>EO</td>
<td>UI frames shown on device screen</td>
<td>high</td>
</tr>
<tr>
<td>EO</td>
<td>Images written into device memory</td>
<td>low</td>
</tr>
<tr>
<td>EO</td>
<td>Images passed to another application to be shared</td>
<td>low</td>
</tr>
<tr>
<td>ELF</td>
<td>OpenCv android SDK</td>
<td>high</td>
</tr>
<tr>
<td>ELF</td>
<td>Android SDK</td>
<td>high</td>
</tr>
<tr>
<td>ELF</td>
<td>Android NDK</td>
<td>high</td>
</tr>
<tr>
<td>ELF</td>
<td>Libraries with Compiled OpenCv code in C/C++</td>
<td>high</td>
</tr>
</tbody>
</table>

**Table 3.3.3 Ranking of components complexity**
Now using values for transaction we convert each low, average or high in a numeric equivalent we can merge everything in the next table. The transactions values were taken from the source [16].

<table>
<thead>
<tr>
<th>Element</th>
<th>Low</th>
<th>Average</th>
<th>High</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>$2 \times 3 = 6$</td>
<td>$0 \times 4 = 0$</td>
<td>$1 \times 6 = 6$</td>
<td>12</td>
</tr>
<tr>
<td>EO</td>
<td>$2 \times 4 = 8$</td>
<td>$0 \times 5 = 0$</td>
<td>$1 \times 7 = 7$</td>
<td>15</td>
</tr>
<tr>
<td>EQ</td>
<td>$0 \times 3 = 0$</td>
<td>$0 \times 4 = 0$</td>
<td>$0 \times 6 = 24$</td>
<td>0</td>
</tr>
<tr>
<td>ILF</td>
<td>$0 \times 7 = 0$</td>
<td>$0 \times 10 = 0$</td>
<td>$0 \times 15 = 0$</td>
<td>0</td>
</tr>
<tr>
<td>ELF</td>
<td>$0 \times 5 = 0$</td>
<td>$0 \times 7 = 0$</td>
<td>$4 \times 10 = 40$</td>
<td>40</td>
</tr>
</tbody>
</table>

**Table 3.3.4 Conversion of modules complexity in numeric value**

Now when we have summed points for every component, we can sum them and get total number of unadjusted Function Points (TUFSP)

$$TUFSP = EI + EO + EQ + ILF + ELF = 12 + 15 + 0 + 0 + 40 = 67$$

Now we need to adjust our function points. In order to do that we need to calculate multiplied values adjustment factor (VAF). In order to do that we need to rank with the value from 0 to 5 next general system characteristic: [16]

<table>
<thead>
<tr>
<th>#</th>
<th>General System Characteristic</th>
<th>Brief Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data communications</td>
<td>How many communication facilities are there to aid in the transfer or exchange of information with the application or system?</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Distributed data processing</td>
<td>How are distributed data and processing functions handled?</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Performance</td>
<td>Was response time or throughput required by the user?</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Heavily used configuration</td>
<td>How heavily used is the current hardware platform where the application will be executed?</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Transaction rate</td>
<td>How frequently are transactions executed daily, weekly, monthly, etc.?</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>On-Line data entry</td>
<td>What percentage of the information is entered On-Line?</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>End-user efficiency</td>
<td>Was the application designed for end-user efficiency?</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>On-Line update</td>
<td>How many ILF’s are updated by On-Line transaction?</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>Complex processing</td>
<td>Does the application have extensive logical or mathematical processing?</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>Reusability</td>
<td>Was the application developed to meet one or many user’s needs?</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Installation ease</td>
<td>How difficult is conversion and installation?</td>
<td>2</td>
</tr>
<tr>
<td>---</td>
<td>------------------</td>
<td>---------------------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>12</td>
<td>Operational ease</td>
<td>How effective and/or automated are start-up, back-up, and recovery procedures?</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>Multiple sites</td>
<td>Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>Facilitate change</td>
<td>Was the application specifically designed, developed, and supported to facilitate change?</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 3.3.5** Ranking of general system characteristics

Now summing all the values of general system characteristics we can get total degree of influence (TDI):

\[
TDI = 5 + 0 + 4 + 2 + 3 + 0 + 2 + 0 + 5 + 3 + 2 + 0 + 5 + 0 = 31
\]

Now knowing TDI we can calculate VAF:

\[
VAF = (TDI \times 0.01) + 0.65 = (31 \times 0.01) + 0.65 = 0.96
\]

Now when we have VAF, we can calculate Total Adjusted Function Points (TAFP):

\[
TAFP = TUFP \times VAF = 67 \times 0.96 = 64.32
\]

A mentioned in the table on the page 14 of the source [20] a function point of coding corresponds with 4 hours of work and 303$ of cost. Taking in consideration it’s values we can calculate next values:

Total effort for developing android application (TE) = 64.32 points * 4 h/pint = 257.28h

Total cost for developing android application (TC) = 64.32 points * 303$/point = 19488.96$

We can see that TE in hours obtained using planning poker and using function points method defers only in 16 hours. On the other hand the price defers significantly, but this happens due to the fact that the price used in function point analysis is price for United States and the price used to get the project cost from project effort is calculated taking in consideration Spanish salaries.
3.4 Possible Risks

Every project involves risks that can influence amount of hours needed for development and though increase the cost of the project. This risks have to be analysed and found solutions to reduce the possibility of this risks to happen. Developing this project involves next possible risks:

<table>
<thead>
<tr>
<th>Description</th>
<th>Likelihood</th>
<th>Impact</th>
<th>Mitigating Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of motivation in the teamwork</td>
<td>Low</td>
<td>Very High</td>
<td>Making team workers involve in the project.</td>
</tr>
<tr>
<td>Unrealistic time and cost estimates</td>
<td>Low</td>
<td>Very High</td>
<td>Exhaustive business analysis.</td>
</tr>
<tr>
<td>Requirements misinterpretation or modification</td>
<td>Very Low</td>
<td>Very High</td>
<td>Timely meetings with stakeholders in order to have a good requirement specification.</td>
</tr>
<tr>
<td>Lack of cooperation from stakeholders</td>
<td>Low</td>
<td>Very High</td>
<td>Signed agreement with them, obeying them to be involved in the project.</td>
</tr>
<tr>
<td>Inadequate project management</td>
<td>Low</td>
<td>Very High</td>
<td>Choosing an experimented PM, who had worked in similar projects</td>
</tr>
<tr>
<td>Bad advertisement</td>
<td>Medium</td>
<td>Very High</td>
<td>Exhaustive market analysis.</td>
</tr>
</tbody>
</table>

Table 3.4.1 Possible risks

3.5 Dividing Development of Android Application in Sprints

For the development of the android application agile strategy was chosen and specifically Scrum which is an iterative and incremental agile software development framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" [25]. After each sprint we will get a working product that will satisfy part of the software specification. In the next table you can see how the task of the android developer 1 were separates into sprints and what where the requirements the product satisfied after the sprint:

Note: Requirement marked with a sign P are satisfied partially at the end of the sprint.

<table>
<thead>
<tr>
<th>Sprint Name</th>
<th>Sprint Tasks</th>
<th>Requirement satisfied by the product after this sprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint 1</td>
<td>Initial creation and setup of the project</td>
<td>R1, R2, R3, R15</td>
</tr>
<tr>
<td>Sprint 2</td>
<td>Connection of the application with camera</td>
<td>R1, R2, R3, R4, R15</td>
</tr>
<tr>
<td>Sprint 3</td>
<td>Implementation of the recognition algorithm (for squares)</td>
<td>R1, R2, R3, R4, R5, R6, R7 (P), R8 (P), R9 (P), R15, R16</td>
</tr>
<tr>
<td>Sprint 4</td>
<td>Implementation of the recognition algorithm (for triangles)</td>
<td>R1, R2, R3, R4, R5, R6, R7 (P), R8 (P), R9 (P), R15, R16</td>
</tr>
<tr>
<td>Sprint 6</td>
<td>Implementation of the recognition algorithm (for circles)</td>
<td>R1, R2, R3, R4, R5, R6, R7, R8, R9, R15, R16</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Sprint 7</td>
<td>Implementation of the replacement algorithm (for squares)</td>
<td>R1, R2, R3, R4, R5, R6, R7, R8, R9, R10 (P), R11 (P), R15, R16</td>
</tr>
<tr>
<td>Sprint 8</td>
<td>Implementation of the replacement algorithm (for triangles)</td>
<td>R1, R2, R3, R4, R5, R6, R7, R8, R9, R10 (P), R11 (P), R15, R16</td>
</tr>
<tr>
<td>Sprint 9</td>
<td>Implementation of the replacement algorithm (for circles)</td>
<td>R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R15, R16</td>
</tr>
<tr>
<td>Sprint 10</td>
<td>Saving the final result to the internal memory storage</td>
<td>R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R15, R16, R17</td>
</tr>
<tr>
<td>Sprint 11</td>
<td>Sharing the picture</td>
<td>R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18</td>
</tr>
</tbody>
</table>

Table 3.5.1 Sprints
Chapter 4
Software Quality Management

To establish possible problems that can appear during the development process and find possible solutions one of the 7 software quality management tools is going to be used. Due to the type of data we have I have chosen fish bone diagram. This software quality tool will give us a possibility to establish problems using a brainstorming and find the possible solutions. After brainstorming next ideas why the application may not work correctly were established:

1. Not enough light in the environment.
2. Angle of inclination of the phone when the image is made
3. Too much time taken by the application analyse image.
4. Not enough space to store the image
5. Recognition of too small/too big shapes
6. Recognition of shapes of another colour (not black on white)
7. Application doesn’t has all necessary permission (Due to specifics of android security measures, every application should always ask user for the permissions to access camera and device storage and user can accept or deny them)
8. Deformation of the shape due to the way the t-shirt is being wore.

After classification of the idea we get next bone-fish diagram:

![Fish-bone diagram](image)

**Picture 4.1** Fish-bone diagram

As another measure of quality management we can use check-list base on the requirements specification. This check-list will be used by testers in order establish if application works appropriately or not. The check list can be found in Attachment 1. As it can bee seen first 13 points of check list are based on software specification and the last two points are based on the possible problems established by fish-bone diagram.
Chapter 5
User Interface Design

5.1 Overall Description

In order to create a user friendly design of the application the design development phase will come through next stages:
- Low-fidelity prototype (Paper design)
- First usability testing
- First collection of user feedback
- High-fidelity prototype
- Second usability testing
- Second collection of user feedback
- Final design

By usability it is understood that a group of 5 people are trying to use “the application” (that at this moment is just a bunch of screens that follow one another) according to the given to them scenario of usage. More detailed the testing process will be described in next paragraphs.

By collecting user feedback it is understood that the group of people that tested the prototype are giving their feedback by answering questionnaires prepared by the designer. After which the designer will process the feedback and make changes while creating new prototype according to the people opinion.

5.2 Low-fidelity Prototype

Picture 5.2.1                                                                 Picture 5.2.2
Picture 5.2.3

Picture 5.2.4

Picture 5.2.5

Picture 5.2.6
5.3 First Usability Testing

For this testing phase a group of 5 volunteered people was chosen. All of them come from different backgrounds, have different age* and gender. The more detailed data about the test users you can find in the table below. All of this users were given a scenario for the usage (which you can find in attachment 2) of the application in paper prototype. The users were video taped during the testing process, so after the designer could evaluate if any of them have trouble with using the prototype. The usability testing of the first prototype was performed in the Dunkin’ coffee in the center of Madrid. The usability testing was performed by one person at a time on different days and different time of the day. The participant’s haven’t met each other during the usability testing process.

<table>
<thead>
<tr>
<th>Person</th>
<th>Age*</th>
<th>Gender</th>
<th>Occupation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>18</td>
<td>Male</td>
<td>Pupil</td>
</tr>
<tr>
<td>B</td>
<td>22</td>
<td>Male</td>
<td>Student of tourism</td>
</tr>
<tr>
<td>C</td>
<td>30</td>
<td>Female</td>
<td>Waitress</td>
</tr>
<tr>
<td>D</td>
<td>35</td>
<td>Male</td>
<td>Driver</td>
</tr>
<tr>
<td>E</td>
<td>40</td>
<td>Female</td>
<td>Scientist</td>
</tr>
</tbody>
</table>

Table 5.3.1 First usability testing participants

5.4 First Collection of User Feedback

After the 5 people mentioned in the previous paragraph performed the usability testing they were asked to answer small questionnaire, which you can find in the attachment 3. The questionnaire was completed right after the usability testing was finished, and given to the designer right the way. The answers of the testers you can find in the attachments from 4.1 to 4.5. After all the questionnaires were completed next ranking was made:

<table>
<thead>
<tr>
<th>Statement from the questionnaire</th>
<th>Medium evaluation from users</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The application is easy to use.</td>
<td>9.6</td>
</tr>
<tr>
<td>2. The design of the application self-explainable.</td>
<td>10</td>
</tr>
<tr>
<td>3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to to find them at first glance</td>
<td>10</td>
</tr>
<tr>
<td>4. After using this application once, I will use it more times.</td>
<td>8.4</td>
</tr>
<tr>
<td>5. I will suggest this application to my friends/family.</td>
<td>9.2</td>
</tr>
<tr>
<td>6. I will use this application at least once per month.</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5.4.1 Ranking of statements from questionnaire

From this results we can say that users found the application to be easy to use because the statements 1, 2 and 3 were rated with a high score (above 9.5). On the other hand people were not that sure about where they will use the application many times or no because
statements 4 and 5 were rated with a smaller grade between 8 and 8.5, event though it is quite possible that people who used this application once will suggest it to other people, because the statement number 6 got rated with a mark above 9. Only two users made personal notes about the application:
A: The main screen is too empty. It could use more images.
B: I would prefer buttons to be circles.

5.5 Hight Fidelity Prototype

* the ages of the participants are between 18 and 45 because this are the most probable future users of the application. Person of any age can by the company's T-shirt, but for example the probability of people older then 40-45, having actual android phone, and navigating through android market in search of different funny application is much lower.
5.6 Second Usability Testing

For this testing phase a group of 5 different volunteered people was chosen. As people from the first testing all of them come from different backgrounds, have different age and gender. The more detailed data about the test users you can find in the table below. All of this users were given a scenario for the usage (the same scenario as the first-testing group, which you can find in **attachment 1**) of the application of high fidelity prototype. The users were video taped during the testing process, so after the designer could evaluate if any of them have trouble with using the prototype. The usability testing of the high-fidelity prototype was performed as well in the Dunkin’ coffee in the centre of Madrid. As in the first usability test, this usability testing was performed by one person at a time on different days and different time of the day. The participant’s haven’t met each other during the usability testing process.

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Gender</th>
<th>Occupation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>20</td>
<td>Female</td>
<td>Student of marketing</td>
</tr>
<tr>
<td>G</td>
<td>26</td>
<td>Female</td>
<td>Nurse</td>
</tr>
<tr>
<td>H</td>
<td>32</td>
<td>Male</td>
<td>Sailor</td>
</tr>
<tr>
<td>I</td>
<td>38</td>
<td>Male</td>
<td>Mechanist</td>
</tr>
<tr>
<td>J</td>
<td>45</td>
<td>Female</td>
<td>Hotel Administrator</td>
</tr>
</tbody>
</table>

**Table 5.6.1 Second usability testing participants**

5.7 Second Collection of User Feedback

After the 5 people mentioned in the previous paragraph performed the usability testing they were asked to answer small questionnaire, which you can find in the **attachment 5**. The questionnaire was completed right after the usability testing was finished, and given to the designer right the way. The answers of the testers you can find in the **attachments from 6.1 to 6.5**. After all the questionnaires were completed next ranking was made:

<table>
<thead>
<tr>
<th>Statement from the questionnaire</th>
<th>Medium evaluation from users</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The application is easy to use.</td>
<td>9.8</td>
</tr>
<tr>
<td>2. The design of the application self-explainable.</td>
<td>9.6</td>
</tr>
<tr>
<td>3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately.</td>
<td>9.4</td>
</tr>
<tr>
<td>4. After using this application once, I will use it more times.</td>
<td>7.4</td>
</tr>
<tr>
<td>5. I will suggest this application to my friends/family.</td>
<td>8</td>
</tr>
<tr>
<td>6. I will use this application at least once per month.</td>
<td>7.4</td>
</tr>
<tr>
<td>7. I like the colour scheme of the application.</td>
<td>9</td>
</tr>
<tr>
<td>8. I find the elements of the application (buttons, texts, images) to have appropriate sizes and shapes.</td>
<td>10</td>
</tr>
</tbody>
</table>

**Table 5.7.1 Ranking of statements from questionnaire**
From this results we can say that users found the application to be easy to use because the statements 1, 2 and 3 were rated with a high score (above 9). Also we can say that people like aesthetic view of the application because statements 7 and 8 got evaluated between 9 and 10. On the other hand people were not that sure about where they will use the application many times or no because statements 4 and 6 were rated with a smaller grade between 7 and 8, even though it is possible that people who used this application. Only one user made personal notes about the application - G: The first screen is unnecessary

Looking at the general ranking of the high-fidelity prototype, we can say the we have chosen a good path in design and we are going to implement it in the application.
Chapter 6
Android Application Implementation

6.1 Overall Description

Once the design was established it’s time to start create the application. Any android application needs include android SDK (Software Development Kit), which is actually a Java based interface that permits communication with android operating system. As it was already mentioned before in order to implement shapes recognition the application will be using external library OpenCv and it’s latest stable version of SDK for android which is 3.1.0 at the moment. (More detailed scheme of the module interaction is described in pic 3.3.1) For the development I will be using Android Studio IDE version 2.1.2, provided by Google for Android developers.

6.2 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. [17]

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 7 million. The library is used extensively in companies, research groups and by governmental bodies. [17]

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan. [17]

It has C++, C, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers. [17]
The source code of the openCV library is written in C/C++ but since Android is working in Java the openCV android SDK is also written in Java, but it refers precompiled libraries in C/C++. Android is able to work with precompiled libraries in C/C++ but in order to do that it needs Android NDK (Native Development Kit), which is a library that permits to connect you java source code with your C/C++ source code.

6.3 Initial Creation and Set-up of the Project

Android studio studio provides quite good and intuitive graphical guide for project creation. If you launch Android Studio and go to File → New → New Project you will see the next window:

![Picture 6.3.1](image)

In this window Android Studio asks you for the name of the folder of your project, that will be stored in your working directory and for the package name. Package name should be a unique identifier of your application. Then if you press next you will get the screen where you can set the android SDK:

![Picture 6.3.2](image)
Since Android Studio has some code templates prepared, in the next step it will ask you if you want to check out any templates. I have chosen a Blank Activity without any template code:

![Android Studio template selection screen](Image)

**Picture 6.3.3**

In the next and last screen Android studio asks for name for the first to create Activity:

![Android Studio activity customization screen](Image)

**Picture 6.3.4**
Now when we have the project created we need to include openCv android SDK library and Android NDK library. Android studio is using Gradle for building packages. So in order to add this two libraries we will have to edit gradle configuration files that were generated by default.

Gradle is an open source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used by Apache Maven of declaring the project configuration. Gradle uses a directed acyclic graph ("DAG") to determine the order in which tasks can be run. Gradle was designed for multi-project builds which can grow to be quite large, and supports incremental builds by intelligently determining which parts of the build tree are up-to-date, so that any task dependent upon those parts will not need to be re-executed. The initial plug-ins are primarily focused around Java, Groovy and Scala development and deployment, but more languages and project work-flows are on the roadmap. [18]

So android SDK is included in the project by default. We we open local.proprieties file we can see next code:

```bash
## This file is automatically generated by Android Studio.
# Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#
# This file must *NOT* be checked into Version Control Systems,
# as it contains information specific to your local configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please read the
# header note.
#Tue May 31 09:59:16 CEST 2016
sdk.dir=/home/nika/Android/Sdk
```

We can see that the path to the android SDK is already included in this file, so we only need to and Android NDK path to the same file. To do that we have to go to File → Project Settings → SDK Location → Android NDK Location and specify the path to Android NDK. After what we will see that in the file local.proprieties appeared new line.

```bash
ndk.dir=/home/nika/Android/android-ndk-r11c/android-ndk-r11c
```

So now we have our project connected bot with Android SDK and Android NDK. It is time to connect openCv SDK, to do that we move the folder java from the files downloaded from official web page of OpenCV, at the same level as the folder app of your project structure and rename it to openCVLibrary310, so you get next structure:

```bash

ndk.dir=/home/nika/Android/android-ndk-r11c/android-ndk-r11c
```

So now we have our project connected bot with Android SDK and Android NDK. It is time to connect openCv SDK, to do that we move the folder java from the files downloaded from official web page of OpenCV, at the same level as the folder app of your project structure and rename it to openCVLibrary310, so you get next structure:
After that we have to copy the folder native from the files downloaded from official web page of OpenCV, inside the folder openCV of your project structure and rename it to jniLibs, so you get next structure:

![Structure of project](image)

**Picture 6.3.6**

Now once we included the file inside our project we have to go to the file settings.gradle. We see next code:

```gradle
include ':app'
```

in order to add openCv to compilation process we need to add next string to this file:

```gradle
include ':openCVLibrary310'
```

After we have edit build.gradle file. If we open it we can see next code in the part of dependencies:

```gradle
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.4.0'
}
```

This is code auto-generated while creation of the project. We need to add next line inside the dependencies module:

```gradle
compile project(path: ':openCVLibrary310')
```

Now we have the projected created and setted up. It’s time to begin implementing the functionality.
6.4 Connection of the Application With Camera

Once the project is set up, the necessary libraries are connected it is time to start writing code, first thing to do is to connect the application with the camera. First we need to create an XML layout to describe how will our screen look like:

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
  <org.opencv.android.JavaCameraView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/image_manipulations_activity_surface_view" />
</LinearLayout>
```

From this code snippet it is seen the in order to display the images coming from camera we will use openCv class JavaCameraView. Now once the XML is ready, logic is to be implemented in the Java class. Since we have to create a class that has a graphical representation it has to extend Activity class from Android SDK and since it is supposed to work with camera it has to implement CvCameraViewListener2 interface from OpenCv SDK, set we get next UML diagram:

![UML diagram of the application so far](image)

As it is seen from the UML diagram class Activity contains methods like onCreate, onResume, onPause, onDestroy, that are methods connected to android application recycle. It is up to any developer to implement them all, to implement some of them or not to implement any, but usually they are used for set up variables, UI or cleaning up the garbage. On the next diagram we can see the explanation of the typical life-cycle of an activity of an android application.
Though the most interesting part is implementation of CvCameraViewListener2 interface since thanks to this interface callbacks application is going to be able to get frames from camera. First off all we implement the method on CameraFrame on order to be simply able to show the image that comes from camera just as it is. We get next code:

```java
public class CameraActivity extends Activity implements CvCameraViewListener2 {
    ... // Methods extended from Activity

    public void onCameraViewStarted(int width, int height) {}

    public void onCameraViewStopped() {}

    public Mat onCameraFrame(CvCameraViewFrame inputFrame) {
        Mat mRgba = inputFrame.rgba();
        return mRgba;
    }
}
```

And if we compile project at this point we get a screen with the frames of images that come from camera as seen on the screen-shot.
Picture 6.4.3 Result of the code execution
6.5 Implementation of the Recognition Algorithm

Following the ideology of object oriented programming all the login for processing the image will be separated in the new class ImageProcessor and we get the next UML diagram of the classes do far:

```
private void applyFilters() {
    imageWithFilters = new Mat();

    Imgproc.cvtColor(originalImage, imageWithFilters, Imgproc.COLOR_BGR2GRAY, 0);
    Imgproc.GaussianBlur(imageWithFilters, imageWithFilters, new Size(9, 9), 5);
    Imgproc.threshold(imageWithFilters, imageWithFilters, 50, 225, Imgproc.THRESH_BINARY);
}
```

As we can see the method applies 3 filters in a sequence:
Now let’s see more detailed info about each type of filter and the functions that apply his filters.

In order to able to recognize primitive shapes as squares, rectangles or triangles first of all the image has to be processed. In the current work next algorithm was chosen:
- Initial image (Image 6.5.2)
- Apply black and white filter: since we are interested in black shapes on white background we can get rid of colours in order to get better performance (Image 6.5.3)
- Apply Gaussian blur filter: in order to get rid of small and unimportant details (like for example small text on the demo image 6.5.3, you can see the result on the image 6.5.4)
- Apply threshold filter: in order to let on the image only borders of the objects (image 6.5.5)

At the beginning I was doubting whether to use Canny, Sobel or Threshold filter, but after making some tests I was able to determine that threshold is the best one for the task.

For applying this filter next openCv methods will be used:
- For converting the image into scale:
  
  
  \[
  \text{public static void cvtColor(Mat src, Mat dst, int code, int dstCn)}
  \]

  Converts an image from one colour space to another.[19]

- For blurring the image:

  \[
  \text{public static void GaussianBlur(Mat src, Mat dst, Size ksize, double sigmaX)}
  \]
Blurs an image using a Gaussian filter. The function convolves the source image with the specified Gaussian kernel. In-place filtering is supported. [19] As a parameter this function receives a size of the Gaussian blur Matrix (ksize), this is a very important parameter because the bigger is the matrix more blur we get. As the both height and width of the Gaussian Matrix have to be bigger then 0 and be divided by 2 with the rest 1, I have chosen 3 initial values: 5, 11, 17 to see which matrix size will be better in our case:

![Picture 6.5.6: ksize = 5](image1.png)  ![Picture 6.5.7: ksize = 11](image2.png)  ![Picture 6.5.8: ksize = 17](image3.png)

From the very first glance at this 3 pictures we can see the image 6.5.8 is blurred too much because the angles of the square don’t look like angle, this can cause problems while detecting the amount of angles in the contour in the next steps. Both picture 6.5.6 and 6.5.7 are not blurred too much, but on the picture 4.5.5 we can still see some small details like text below the rectangle, and this details are not seen on the image 6.5.7, which means that ksize = 11 is the best option in our case..

For applying threshold filter:

```java
public static double threshold(Mat src, Mat dst, double thresh, double maxval, int type)
```
The function applies fixed-level thresholding to a single-channel array. The function is typically used to get a bi-level (binary) image out of a grey-scale image or for removing a noise, that is, filtering out pixels with too small or too large values. There are several types of threshold supported by the function. They are determined by type parameter. [19] As I mentioned before I tries different filters for this step. They were Threshold, Canny and Sobel:

As seen from the pictures all three algorithm were able to successfully separate square from the surroundings. Though Canny and Sobel have determined contours that are useless for us, like for example the hands of the person. On the other hand Threshold was able to separate square and person’s hands. So I decided that it would be the best to use Threshold for our situation. As it is seen from the Threshold function description it contains two parameters threshold and maxValue. This two parameters are very important because when we change them we can obtain a more detailed or less detailed pictures. This values represent the different between the colours which are forming borders, in our case this are white and black. So as maxValue we use 255 to represent the whole pallet of colour. Then I started to experiment with the threshold value:
From the first glance at the pictures we can see that threshold 200 doesn’t work in our situation because the square on the shit isn’t separated from the the rest of the surroundings. Threshold 50/150/200 were able to separate the square from the surroundings, but the 100 and 150 gave too many other unnecessary details. The square on the picture with the threshold 50 is a little smaller that it is on the original picture but we don’t have any unnecessary details on this picture. So I decided to use threshold 50, because it is the best value for detecting black circles on the white background.

Once all the filters are successfully applied the openCV algorithms to find circles, triangles and rectangles are to be applied. First of all let’s see the method that is used to find all the circled on the image:

```java
public Mat findCircle() {
    Mat circles = new Mat();

    Imgproc.HoughCircles(imageWithFilters, circles,
                         Imgproc.CV_HOUGH_GRADIENT,
                         2.0, 100, 100, 300, 30, 150);
```

From the first glance at the pictures we can see that threshold 200 doesn’t work in our situation because the square on the shit isn’t separated from the the rest of the surroundings. Threshold 50/150/200 were able to separate the square from the surroundings, but the 100 and 150 gave too many other unnecessary details. The square on the picture with the threshold 50 is a little smaller that it is on the original picture but we don’t have any unnecessary details on this picture. So I decided to use threshold 50, because it is the best value for detecting black circles on the white background.

Once all the filters are successfully applied the openCV algorithms to find circles, triangles and rectangles are to be applied. First of all let’s see the method that is used to find all the circled on the image:

```java
public Mat findCircle() {
    Mat circles = new Mat();

    Imgproc.HoughCircles(imageWithFilters, circles,
                         Imgproc.CV_HOUGH_GRADIENT,
                         2.0, 100, 100, 300, 30, 150);
```
for (int x = 0; x < circles.cols(); x++) {
    double vCircle[] = circles.get(0,x);

    if (vCircle == null)
        break;

    Point pt = new Point(Math.round(vCircle[0]),
                         Math.round(vCircle[1]));
    int radius = (int)Math.round(vCircle[2]);
    Imgproc.circle(originalImage, pt, radius,
                    new Scalar(0,0,255), 5);
}

return circles;
}

In order to find circles next openCv method is used:

    HoughCircles(Mat  image,  Mat  circles,  int  method,  double  dp,  double
                minDist, double param1, double param2, int minRadius, int maxRadius)

The function finds circles in a grey-scale image using a modification of the Hough transform. [19] After calling this method the the Matrix passed as second parameter to the method will contain the coordinates of all found circles. This method also provides us such parameters as the minimum and maximum radius of the circle and distance from the circle, which permits to discard unnecessary circles that were found on the image but don’t respond to the specifications. The “for” cycle after is used to draw the contours of all circles found using next method:

    public static void circle(Mat img, Point center, int radius, Scalar color, int
                              thickness)

The thing gets more complicated for searching squares and triangles. Since openCv doesn’t have any specific methods for searching exactly for this shapes, it will be necessary to use the methods for finding contours after what it will be necessary to sort the contours to select only those that represent squares and rectangles. As far as finding contours is a common part both for rectangles and triangles I separated it in a separate method called preSearch:

    public void preSearch() {
        applyFilters();

        candidates = new ArrayList<>();
        polygons = new ArrayList<>();
        Imgproc.findContours(imageWithFilters,
                              candidates, new Mat(),
                              Imgproc.RETR_TREE, Imgproc.CHAIN_APPROX_NONE);

        for (MatOfPoint candidate: candidates) {
            MatOfPoint2f candidateConverted =
new MatOfPoint2f(candidate.toArray());
MatOfPoint2f polygon = new MatOfPoint2f();
int angles = (int) candidate.total();

Imgproc.approxPolyDP(candidateConverted,
    polygon, angles * 0.05, true);
polygons.add(polygon);
}

In order to find the contours on the image next openCv method will be called:

    findContours(Mat  image,  List<MatOfPoint>  contours,  Mat  hierarchy,  int
mode, int    method)

The function retrieves contours from the binary image using the border following
algorithm. [19] After calling this method we get all the determined contours on the given
image. After what we have to analyse each contour and get it approximate polygon in order
to be able to separate squares from triangles and from and other possible shapes. To do that I
am going to use next openCV method:

    public  static  void  approxPolyDP(MatOfPoint2f  curve,  MatOfPoint2f
approxCurve, double epsilon, boolean closed)

The functions approxPolyDP approximate a curve or a polygon with another
curve/polygon with less vertices so that the distance between them is less or equal to the
specified precision. It uses the Douglas-Peucker algorithm. [19] Once this preparation step
was made the separate methods for searching triangles and rectangles can be applied. Both
of them will be analysing contours and their approximations to searched for the appropriate
shapes. First let’s see the method for finding triangles:

public List<MatOfPoint> findTriangles() {  
    List<MatOfPoint> triangles = new ArrayList<>();

    for (int i = 0; i < candidates.size(); i++) {
        MatOfPoint candidate = candidates.get(i);
        MatOfPoint2f approxCurve = polygons.get(i);
        double contourArea = Imgproc.contourArea(candidate);

        if (approxCurve.total() == 3 && contourArea > 800 && contourArea < 9000) {

            Point[] points = approxCurve.toArray();
            double x1 = points[1].x > points[2].x ? points[1].x : points[2].x;
            double x2 = points[1].x < points[2].x ? points[1].x : points[2].x;
            double y1 = points[0].y > points[2].y ? points[0].y : points[2].y;
            double y2 = points[0].y < points[2].y ? points[0].y : points[2].y;

            double partOne = x1 - x2 > y1 - y2 ? x1 - x2 : y1 - y2;
            double partTwo = x1 - x2 < y1 - y2 ? x1 - x2 : y1 - y2;

            if (partOne - partTwo < 20) {
                triangles.add(candidate);
            }
        }
    }

    return triangles;
}
Now let’s see code line by line. First we make a cycle from each shape found on the picture:

```java
for (int i = 0; i < candidates.size(); i++)
```

Count perimeter of the shape:

```java
double contourArea = Imgproc.contourArea(candidate);
```

Then we have to analyse if the shape is a triangle and if it has appropriate size:

```java
if (approxCurve.total() == 3 && contourArea > 800 && contourArea < 9000)
```

The shape is triangle if it has three sides. According to our specification the size of the shapes to be recognizes are 10 centimetres hight and have to be situated between one and two meters from the camera. As our specification mentions the size of the shapes in centimetres, but the method that counts the area of the shape return it’s size in pixels I had to find the correspondence between the centimetres. To do that I had to experiment by getting closer or farer to the shape and following the value returned by counturArea. So after several tries I was able to identify that area of a triangle which size is 10 centimetres and is placed between one and two meters has size between 800 and 9000 pixels. Now when we made sure that the shape is a triangle of an appropriate size we have to make sure that all triangle’s sides have the same dimensions.

```java
Point[] points = approxCurve.toArray();
double x1 = points[1].x > points[2].x ? points[1].x : points[2].x;
double x2 = points[1].x < points[2].x ? points[1].x : points[2].x;
double y1 = points[0].y > points[2].y ? points[0].y : points[2].y;
double y2 = points[0].y < points[2].y ? points[0].y : points[2].y;

double partOne = x1 - x2 > y1 - y2 ? x1 - x2 : y1 - y2;
double partTwo = x1 - x2 < y1 - y2 ? x1 - x2 : y1 - y2;
```

As seen from the next picture, where android screen coordinate system is presented (Picture 6.5.16), to make sure that the sides of the triangle are all equal, we have to make sure that the difference between bigger and smaller x coordinate and the difference between bigger and smaller y coordinate:

\[ x_1 - x_2 = y_2 - y_1 \]
If the equation mentioned before is true with the accuracy of 20 pixels (to take into consideration the deformation), that means we detected a triangle that corresponds to our specification. That’s why we add it to the array which is returned from the method:

```java
if (partOne - partTwo < 20) {
    triangles.add(candidate);
    Imgproc.drawContours(originallImage, squares, -1, new Scalar(255, 0, 0),
                         5);
}
```

Now let’s see the method `findSquares`

```java
public List<MatOfPoint> findSquares() {
    List<MatOfPoint> squares = new ArrayList<>();

    for (int i = 0; i < candidates.size(); i++) {
        MatOfPoint candidate = candidates.get(i);
        MatOfPoint2f approxCurve = polygons.get(i);
        double contourArea = Imgproc.contourArea(candidate);
        if (approxCurve.total() == 4 && contourArea > 1200 && contourArea < 15000) {
            Point[] points = approxCurve.toArray();
            double x1 = points[0].x > points[2].x ? points[0].x : points[2].x;
            double x2 = points[0].x < points[2].x ? points[0].x : points[2].x;
            double y1 = points[0].y > points[2].y ? points[0].y : points[2].y;
            double y2 = points[0].y < points[2].y ? points[0].y : points[2].y;

            double partOne = x1 - x2 > y1 - y2 ? x1 - x2 : y1 - y2;
            double partTwo = x1 - x2 < y1 - y2 ? x1 - x2 : y1 - y2;
```

**Picture 6.5.16**

If the equation mentioned before is true with the accuracy of 20 pixels (to take into consideration the deformation), that means we detected a triangle that corresponds to our specification. That’s why we add it to the array which is returned from the method:
if (partOne - partTwo < 20) {
    squares.add(candidate);
    Imgproc.drawContours(originalImage, squares, -1, new Scalar(0, 255, 0), 5);
}
If the equation mentioned before is true with the accuracy of 20 pixels (to take into consideration the deformation), that means we detected a square that corresponds to our specification. That’s why we add it to the array which is returned from the method:

```java
if (partOne - partTwo < 20) {
    squares.add(candidate);
    Imgproc.drawContours(originalImage, squares, -1, new Scalar(0, 255, 0), 5);
}
```

Now it’s time to make second test. For the initial tests I used just paper with 3 shapes that are supposed to be recognized. So after executing the code we have so far, we get the result that you can see on the picture 6.5.18. All three shapes were successfully recognized and highlighted by the appropriative colour. Next step would be change the contour of the shape that is drawn right now with an appropriate image.
6.6 Implementation of the Replacement Algorithm

Following the ideology of object oriented programming and information hiding the logic of drawing will be hidden in the class that represents the picture itself, which leads us to creation of two different interfaces (one for picture that are drawn on a polygon and another for the pictures drawn on the oval form shapes). From now on any picture that is supposed to be drawn has to be represented as a class and extend one this two interface depending on if it’s drawn on a polygon or on an oval shape image. And we get the next UML diagram of the classes do far:

**Picture 6.6.1 UML diagram of the application so far**
Let’s take a look at the method draw of class Snowboard.

```java
public Mat draw(Mat originalImage, MatOfPoint2f where) {
    final Bitmap arImage = BitmapFactory.decodeResource(activity.getResources(), arImageResource, null);
    final Bitmap originalBitmap = Bitmap.createBitmap(originalImage.cols(), originalImage.rows(), Bitmap.Config.RGB_565);

    Utils.matToBitmap(originalImage, originalBitmap);

    double contourArea = Imgproc.contourArea(where);
    double size = Math.sqrt(contourArea);
    Point[] points = where.toArray();

    Canvas canvas = new Canvas(originalBitmap);
    RectF rect = new RectF();

    Point first = points[0];
    rect.set((int) first.x, (int) first.y, (int) first.x + (int) size,
             (int) first.y + (int) size);
    canvas.drawBitmap(arImage, null, rect, null);

    final Mat finalImage = new Mat(originalImage.cols(), originalImage.rows(),
                                    CvType.CV_8UC4);
    Utils.bitmapToMat(originalBitmap, finalImage);

    return finalImage;
}
```

Let’s see the code line by line.

```java
final Bitmap arImage = BitmapFactory.decodeResource(activity.getResources(),
                                                        R.drawable.snow, null);
```

First of all it is needed to retrieve the image of the snowboard from the res folder of our project. To do that native android SDK method:

```java
public static Bitmap decodeResource(Resources res, int id,
                                      BitmapFactory.Options opts)
```

where the id is represented in the next from R.drawable.name_of_the_image, which indicates the number of our image indicated in the binary file called R generated after compilation. This method returns the image in the format of Bitmap. Next step is to convert the image coming from camera that is presented with a Mat to a Bitmap, so we would be able to make a composition of two bitmaps. Next code is doing that:

```java
final Bitmap originalBitmap = Bitmap.createBitmap(originalImage.cols(),
                                                   originalImage.rows(), Bitmap.Config.RGB_565);
```

```java
Utils.matToBitmap(originalImage, originalBitmap);
```
In the first line we create new empty bitmap using native android SDK method:

```java
public static Bitmap createBitmap(int width, int height, Bitmap.Config config)
```

This method returns a mutable bitmap with the specified width and height. After an empty image was created we have to copy the pixel data from Mat to the newly created bitmap with the next method from OpenCV android SDK:

```java
public static void matToBitmap(Mat mat, Bitmap bmp)
```

Now when we have both images in the same format we can merge the. To do that we have find out what is the corresponding dimension and location of the image we have to insert. First let’s count the size of the image we have to insert. As we already know the coordinates of the shape that was found we can calculate it’s area using the openCV function:

```java
double contourArea = Imgproc.contourArea(where);
```

Now knowing the area of image and knowing that the formula of the area of the square is:

\[ S = a \times a, \text{ where } S \text{ is area, } a \text{ is side of the square} \]

we can calculate the size of the square, which should be equal to the size of the image we are going to insert with the next code:

```java
double size = Math.sqrt(contourArea);
```

Next we have to prepare a Canvas from the image coming from the camera in order to be able to draw on it. For this we will use a standard constructor of the class Canvas which is part of Android SDK:

```java
Canvas canvas = new Canvas(originalBitmap);
```

Once the canvas is ready we will be using next method to draw the desired image on the base image:

```java
public void drawBitmap(Bitmap bitmap, Rect src, RectF dst, Paint paint)
```

As second parameter this method receives an instance of class Rect that establishes the bounds where should the second image be drawing on the first image. As we already know the coordinates of the rectangle that was found, we just take the first point out for that represent the square and add to both x and y coordinates the size of the image we calculated in the previous step.

```java
Point[] points = where.toArray();
RectF rect = new RectF();
Point first = points[0];
rect.set((int) first.x, (int) first.y, (int) first.x + (int) size, (int) first.y + (int) size);
```
Now once we have the coordinates where to draw the new image we can draw:

```java
canvas.drawBitmap(arImage, null, rect, null);
```

And the last step is to convert the bitmap back to the Mat in order to be shown on the screen.

```java
final Mat finalImage = new Mat(originalImage.cols(), originalImage.rows(), CvType.CV_8UC4);
Utils.bitmapToMat(originalBitmap, finalImage);
```

The algorithm for drawing skateboard and surfboard are the same as for the square in exception for the formula to calculate the area of the shape. And after compiling the code we have so far, we get the result presented on the screen-shot from the left. All the shapes were successfully recognized and and the corresponding augmented reality pictures were added.
6.7 Saving the Final Result to Internal Memory Storage

After the camera image was complimented with appropriate augmented reality image it is time to prepare algorithm for saving the image. All the logic for image storage is separated in the ImageSaver class. For now the UML diagram of classes looks in the next way:

Picture 6.7.1 UML diagram of the application so far

To handle click user makes on the “take photo” button we add new parameter onClick on the XML description of this button:

```xml
<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_alignParentBottom="true"
    android:background="@drawable/circle_button"
>
Then the method `onTakePhotoClick` should be added to `CameraActivity`. This method contains logic of calling the methods of `ImageSaver` class

```java
public void onTakePhotoClick(View view) {
    ImageSaver saver = new ImageSaver(this);
    saver.saveMat(resultImage);
}
```

Now let’s review the `saveMat` method:

```java
public void saveMat(Mat resultImage) {
    builder = new DialogBuilder(context);
    dialog = builder.buildProgressDialog(R.string.saving_image);
    dialog.show();

    final Bitmap originalBitmap = covertMatToBitmap(resultImage);
    int success = saveToPublicPicturesFolder(originalBitmap);

    dialog.dismiss();
    showSuccessOrErrorDialog(success);
}
```

It creates a loading dialogue, because saving the image takes time, then it converts `Mat` to a `Bitmap` and saves the image, remove the progress dialogue and show success or fail dialogue to announce the user if the picture was taken with success or not. Let’s take a closer look at the method for converting `Mat` to `Bitmap`

```java
private Bitmap covertMatToBitmap(Mat resultImage) {
    final Bitmap originalBitmap = Bitmap.createBitmap(
            resultImage.cols(), resultImage.rows(), Bitmap.Config.RGB_565);

    Utils.matToBitmap(resultImage, originalBitmap);
    return originalBitmap;
}
```

To convert `Mat` to `Bitmap` OpenCV Android SDK method `fromMatToBitmap` is used. Now let’s take a look at the method `saveToPublicPicturesFolder`:

```java
public int saveToPublicPicturesFolder(Bitmap bitmapImage) {
    int result;
    FileOutputStream fileOutputStream = null;
    try {
        fileOutputStream = new FileOutputStream(createFile());
        bitmapImage.compress(Bitmap.CompressFormat.PNG, 100, fileOutputStream);
        result = createFile();
    }
```
result = -1;
} catch (IOException e) {
    e.printStackTrace();
    result = 0;
} catch (Exception e1) {
    e1.printStackTrace();
    result = 1;
} finally {
    closeFile(fileOutputStream);
}

return result;
}

This method creates an OutputStream for a newly created file by the method createFile:

fileOutputStream = new FileOutputStream(createFile());

after which it compresses the image into this output stream

bitmapImage.compress(Bitmap.CompressFormat.PNG, 100, fileOutputStream);

If no exceptions were rose the method returns -1 which means that the file was saved successfully, if and IOException was rose means that there is not enough space in the device memory to save the image and the method returns 0, if any other exception was rose method returns 1 to identify that an unknown error happened while saving the image. The value returned by this message is passed to the showSuccessOrErrorDialog method:

private void showSuccessOrErrorDialog(int success) {
    if (success == -1) {
        AlertDialog dialog = builder.buildSuccessfullyTakePhotoAlertDialog();
        dialog.show();
    } else if (success == 0) {
        AlertDialog dialog = builder.buildNoFreeSpaceAlertDialog();
        dialog.show();
    } else {
        AlertDialog dialog = builder.buildErrorTakingPhotoAlertDialog();
        dialog.show();
    }
}

As it can be seen this method depending on the value of success variables shows different dialog using the DialogBuilder class which has all the dialogue logic implementation. After executing code we will get next output:
Picture 6.7.2

Picture 6.7.3

Hooray!
The picture was successfully stored in your device gallery

TAKE ANOTHER PHOTO
6.8 Sharing the Image

Now when the image is stored in the internal storage, we can share it using all the application that are available on the device and that support sharing images. To do this standard sharing intent will be used. But first of all let’s see the class UML diagram till the point. All the logic for sharing will be stored in in new class ShareActivity and since this class is supposed to have an graphical representation it has to extend Activity.

Let’s see the code of the method to share the image:

```java
public void onSharePhotoClick(View view) {
    Intent shareIntent = new Intent();
    shareIntent.setAction(Intent.ACTION_SEND);
    // Rest of the code
}
```

**Picture 6.8.1** UML diagram of the application so far
shareIntent.putExtra(Intent.EXTRA_STREAM, resultImagePath);
shareIntent.setType("image/*");
startActivity(Intent.createChooser(shareIntent,
    getResources().getText(R.string.send_to)));
}

As it is seen from the code snippet the application creates intent that is necessary to launch another application that will take care of sharing and it fills the intent with appropriate data for application that will share like type of the data to be shared and the path to the image to be shared. After the executing new code we get next output:

![Picture 6.8.2](image1)
![Picture 6.8.3](image2)
Picture 6.8.4
6.9 Creating Initial Screen

Since the software specifications mentions the possibility of processing a photo that is stored in the memory for the future versions, a new initial screen is to be added. On this screen user will be able to select whether he wants to take a new picture or select one that was already taken. Though the second actions won’t be available for now. On the other hand this activity will be in charge of checking if application has all necessary permissions. Now let’s take a look at the final UML diagram of the classes:

```
public void onCameraClick(View view) {
    if (helper.doesTheDeviceHasCamera()) {
        if (helper.doesTheAppHasPermissionToUseCamera() &&
            helper.doesTheAppHasPermissionToWriteFiles()) {
            startCameraIntent();
        } else {
            // handle error
        }
    }
```

Picture 6.9.1 UML diagram of the application so far
This method checks if the device has at least one camera if no it alerts user about this problem. Is the camera is available this method checks for permissions for using camera and permission for storing and reading data from the device. If the permissions are granted CameraActivity is started, otherwise dialogue asking for permission is called. This method is invoked when the user clicks the button to go to take a photo. The invocation of this method is described in XML file of the graphical design of this activity, using the propriety onClick

<ImageView
    android:id="@+id/ma_camera_image_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_camera_selector"
    android:layout_margin="10dp"
    android:layout_alignParentBottom="true"
    android:onClick="onCameraClick" />

After the execution of the code we get next result:

![Image of camera and info icon with "Amazing T" text]

Picture 6.9.2
Chapter 7
Testing

7.1 Automatic Testing

For automatic testing unit testing was chosen and black boxing technique of testing. Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle. [22] This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well [23]. As it was mentioned before automatic testing contains 4 parts:
  1. Recognition algorithm module testing
  2. Replacement algorithm testing
  3. Testing of saving algorithm
  4. Testing of sharing algorithm

Let’s see the results of the test cases designed for each module more detailed. Recognition method algorithm has three parts to be tested. In next tables you can see detailed description of testing of all of them

NB: When in the description of the test the size of the shape is not specifies, it means that size corresponds to software specifications. When the color of the shape or background isn’t mentioned it means that shape is black and background is white.

1.1. Recognition of squares (Table 7.1.1)

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1. Call constructor of ImageProcessor passing a Mat of an image that has one square 2. Call preSearch 3. Call findSquares</td>
<td>findSquares returns a List&lt;MatOfPoints2f&gt; with size one</td>
</tr>
<tr>
<td>2</td>
<td>1. Call constructor of ImageProcessor passing a Mat of an image that has three squares 2. Call preSearch 3. Call findSquares</td>
<td>findSquares returns a List&lt;MatOfPoints2f&gt; with size three</td>
</tr>
<tr>
<td>3</td>
<td>1. Call constructor of ImageProcessor passing a Mat of an image that has one square and one triangle 2. Call preSearch 3. Call findSquares</td>
<td>findSquares returns a List&lt;MatOfPoints2f&gt; with size one</td>
</tr>
<tr>
<td>4</td>
<td>1. Call constructor of ImageProcessor passing a Mat of an image that has one square and one circle 2. Call preSearch 3. Call findSquares</td>
<td>findSquares returns a List&lt;MatOfPoints2f&gt; with size one</td>
</tr>
<tr>
<td>5</td>
<td>1. Call constructor of ImageProcessor passing a Mat of an image that has two square and one circle and a triangle 2. Call preSearch</td>
<td>findSquares returns a List&lt;MatOfPoints2f&gt; with size two</td>
</tr>
<tr>
<td></td>
<td>Call of findSquares</td>
<td>Result of findSquares</td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td>------------------------</td>
</tr>
</tbody>
</table>
| 6 | 1. Call constructor of ImageProcessor passing a Mat of an image that has no primitive shapes  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
| 7 | 1. Call constructor of ImageProcessor passing a Mat of an image that has no primitive squares, but has a circle and a triangle  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
| 8 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a square bigger than specified in software specifications  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
| 9 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a square smaller than specified in software specifications  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
| 10 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a green square  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
| 11 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a square on a blue background  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
| 12 | 1. Call constructor of ImageProcessor passing an empty Mat  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
| 13 | 1. Call constructor of ImageProcessor passing an null  
2. Call preSearch  
3. Call findSquares | Application doesn’t rise Exception (NullPointer) and findSquares returns a List<MatOfPoints2f> with size zero |
| 14 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a square with coordinates of top left corner 20px, 20px and bottom right 140px, 140px  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size one and the MatOfPoints2f has top left coordinate 20 px, 20px and bottom right coordinate 140px, 140px with an 10px approximation |
| 15 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a rectangle  
2. Call preSearch  
3. Call findSquares | findSquares returns a List<MatOfPoints2f> with size zero |
# 1.2. Recognition of triangles (Table 7.1.2)

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
</table>
| 16 | 1. Call constructor of ImageProcessor passing a Mat of an image that has one triangle  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size one |
| 17 | 1. Call constructor of ImageProcessor passing a Mat of an image that has three triangles  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size three |
| 18 | 1. Call constructor of ImageProcessor passing a Mat of an image that has one square and one triangle  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size one |
| 19 | 1. Call constructor of ImageProcessor passing a Mat of an image that has one triangle and one circle  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size one |
| 20 | 1. Call constructor of ImageProcessor passing a Mat of an image that has two triangles and one circle and a square  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size two |
| 21 | 1. Call constructor of ImageProcessor passing a Mat of an image that has no primitive shapes  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size zero |
| 22 | 1. Call constructor of ImageProcessor passing a Mat of an image that has no triangles, but has a circle and a rectangle  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size zero |
| 23 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a triangle bigger then specified in software specifications  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size zero |
| 24 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a triangle smaller then specified in software specifications  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size zero |
| 25 | 1. Call constructor of ImageProcessor passing a Mat of an image that has a green triangle  
   2. Call preSearch  
   3. Call findTriangles | findTriangles returns a List<MatOfPoints2f> with size zero |
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has a triangle on a blue background 1. Call preSearch 2. Call findTriangles</td>
<td>findTriangles returns a List&lt;MatOfPoints2f&gt; with size zero</td>
</tr>
<tr>
<td>27</td>
<td>Call constructor of ImageProcessor passing an empty Mat 1. Call preSearch 2. Call findTriangles</td>
<td>findTriangles returns a List&lt;MatOfPoints2f&gt; with size zero</td>
</tr>
<tr>
<td>28</td>
<td>Call constructor of ImageProcessor passing an null 1. Call preSearch 2. Call findTriangles</td>
<td>Application doesn't rise Exception (NullPointerException) and findTriangles returns a List&lt;MatOfPoints2f&gt; with size zero</td>
</tr>
<tr>
<td>29</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has a triangle with coordinates of top corner 90px, 20px, bottom right corner 140px, 140px and bottom left corner 20px, 140px 1. Call preSearch 2. Call findSquares</td>
<td>findTriangles returns a List&lt;MatOfPoints2f&gt; with size one and the MatOfPoints2f has top corner 90px, 20px, bottom right corner 140px, 140px and bottom left corner 20px, 140px with an 10px approximation</td>
</tr>
<tr>
<td>30</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has a triangle with sides that aren't equal 1. Call preSearch 2. Call findTriangles</td>
<td>findTriangles returns a List&lt;MatOfPoints2f&gt; with size zero</td>
</tr>
</tbody>
</table>

1.3. Recognition of circles (Table 7.1.3)

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has one circle 1. Call findCircles</td>
<td>findCircles returns a Mat with number of columns equals one</td>
</tr>
<tr>
<td>32</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has three circles 1. Call findCircles</td>
<td>findCircles returns a Mat with number of columns equals three</td>
</tr>
<tr>
<td>33</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has one circle and one triangle 1. Call findCircles</td>
<td>findCircles returns a Mat with number of columns equals one</td>
</tr>
<tr>
<td>34</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has one circle and one square 1. Call findCircles</td>
<td>findCircles returns a Mat with number of columns equals one</td>
</tr>
<tr>
<td>35</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has two circles and one triangle and a square 1. Call findCircles</td>
<td>findCircles returns a Mat with number of columns equals two</td>
</tr>
<tr>
<td>36</td>
<td>Call constructor of ImageProcessor passing a Mat of an image that has no primitive shapes 1. Call findCircles</td>
<td>findCircles returns a Mat with number of columns equals zero</td>
</tr>
</tbody>
</table>
1. Call constructor of ImageProcessor passing a Mat of an image that has no circles, but has a triangle and a rectangle
2. Call findCircles

findCircles returns a Mat with number of columns equals zero

1. Call constructor of ImageProcessor passing a Mat of an image that has a circle bigger then specified in software specifications
2. Call findCircles

findCircles returns a Mat with number of columns equals zero

1. Call constructor of ImageProcessor passing a Mat of an image that has a circle smaller then specified in software specifications
2. Call findCircles

findCircles returns a Mat with number of columns equals zero

1. Call constructor of ImageProcessor passing a Mat of an image that has a green circle
2. Call findCircles

findCircles returns a Mat with number of columns equals zero

1. Call constructor of ImageProcessor passing a Mat of an image that has a circle on a blue background
2. Call findCircles

findCircles returns a Mat with number of columns equals zero

1. Call constructor of ImageProcessor passing an empty Mat
2. Call findCircles

findCircles returns a Mat with number of columns equals zero

1. Call constructor of ImageProcessor passing an null
2. Call findCircles

Application doesn’t rise Exception (NullPointerException) and findCircles returns a Mat with number of columns equals zero

1. Call constructor of ImageProcessor passing a Mat of an image that has a circle with the center coordinate 90px, 90px and a radius of 70px
2. Call preSearch
3. Call findSquares

findCircles returns a Mat with number of columns equals one and mat[0][0] = 90, mat[0][1] = 90 and mat[0][2] = 70 with a margin of 10dp

Next module is replacement algorithm module and it also contains 3 parts:

2.1 Replacement algorithm for rectangles (Table 7.1.4)

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>Call drawSnowboard() passing it a Mat of an image that has one square and the coordinates of this square as second parameter</td>
<td>The picture of snowboard is drawn on the initial image</td>
</tr>
<tr>
<td>46</td>
<td>Call drawSnowboard() passing it a Mat of an image that has one square and the coordinates of this square as second parameter</td>
<td>The size of the picture of snowboard should have the same area size and the area between coordinates of the second parameter with a margin of 100 px</td>
</tr>
<tr>
<td>47</td>
<td>Call drawSnowboard() passing it a Mat of an image that has one square and the coordinates of this square as second parameter</td>
<td>Top right and left points’ as bottom right and left points’ coordinates of the second parameter should coincide with the corresponding points of show-board picture</td>
</tr>
</tbody>
</table>
### 2.2 Replacement algorithm for triangles (Table 7.1.5)

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>Call <code>drawSkateboard()</code> passing it a Mat of an image that has one triangle and the coordinates of this triangle as second parameter</td>
<td>The picture of skateboard is drawn on the initial image</td>
</tr>
<tr>
<td>49</td>
<td>Call <code>drawSkateboard()</code> passing it a Mat of an image that has one triangle and the coordinates of this triangle as second parameter</td>
<td>The size of the picture of snowboard should have the doubled area size as the area between coordinates of the second parameter with a margin of 100 px</td>
</tr>
<tr>
<td>50</td>
<td>Call <code>drawSkateboard()</code> passing it a Mat of an image that has one triangle and the coordinates of this triangle as second parameter</td>
<td>Top point’s as bottom right and left points’ coordinates of the second parameter should coincide with the corresponding points of skateboard picture</td>
</tr>
</tbody>
</table>

### 2.3 Replacement algorithm for circles (Table 7.1.6)

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>Call <code>drawSurfboard()</code> passing it a Mat of an image that has one circle and the coordinates of this circle as second parameter</td>
<td>The picture of surfboard is drawn on the initial image</td>
</tr>
<tr>
<td>52</td>
<td>Call <code>drawSurfboard()</code> passing it a Mat of an image that has one circle and the coordinates of this circle as second parameter</td>
<td>The size of the picture of surfboard should have the same area size as the area between coordinates of the second parameter with a margin of 100 px</td>
</tr>
<tr>
<td>53</td>
<td>Call <code>drawSurfboard()</code> passing it a Mat of an image that has one circle and the coordinates of this circle as second parameter</td>
<td>The center of circle should coincide with the center of the surfboard and and diameter of the circle should coincide with the height and width of the surfboard image</td>
</tr>
</tbody>
</table>

Next module to test is module for saving the image. Next test cases were created: (Table 7.1.7)

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>Call <code>saveMat()</code> passing it a Mat of an image</td>
<td>A dialog saying that the image was saved successfully is shown</td>
</tr>
<tr>
<td>55</td>
<td>Call <code>saveMat()</code> passing it an empty Mat</td>
<td>A dialog saying that the image was saved successfully is shown</td>
</tr>
<tr>
<td>56</td>
<td>Call <code>saveMat()</code> passing it a null</td>
<td>A dialog saying that there has been an error saving the image occurs is shown</td>
</tr>
</tbody>
</table>
Next module to test is module for sharing the image. Next test cases were created: (Table 7.1.8)

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>Start ShareActivity passing it a path to an existing image in the memory, call onSharePhotoClick</td>
<td>A dialog saying share image is shown</td>
</tr>
</tbody>
</table>

7.2 Manual Testing

The manual testing was performed with three persons that were wearing each one a T-shirt with one of the free shapes to be recognized by the application. The photos were taken in different places and at different time of day during several days. The testing showed that there are still few problems with detecting the shapes in a bad lightened environment and with a certain angles of inclination of the phone. All the test were performed with a Nexus 6 phone running Android 6.0 (Marshmallow). On the next pictures you can see some of the screen-shots that were taken during the manual testing.

![Picture 7.2.1](image1.png)  ![Picture 7.2.2](image2.png)
7.3 Quality Assurance in after Release Time

As it is impossible to cover all possible cases with automatic and manual testing there is always a possibility of bug appearing in the production phase. To be able to be informed about any crashes of the application Fabric will be used. Fabric is a mobile platform with modular kits you can mix and match to build the best applications. Fabric is tightly integrated into your developer environment, making adding new services a breeze. Start with what you need from Fabric today, and quickly add more kits as your needs grow. [24] So now whenever the application crashes on an any user phone, developers get the report with all the detailed information about the crashes and about the phone and Android operation system on this phone. This permits to identify and fix the bug easy and fast.
Conclusions

As the result of preforming of the master thesis an augmented reality android application was created. The application was created from a zero point and passed through all phases of software development:
- software requirement elaboration
- development plan elaboration
- project cost computation
- software quality assurance
- usability testing
- design elaboration
- project code elaboration
- testing

At the beginning a list of software requirements based on the features of the application to be done were elaborated. Requirements for the current and for the future version of the application were created. This will permit the developers team to create a more suitable system architecture.

Based on the software specification and amount of work a project plan was developed. It included 5 different roles: project manager, designer, two android developers and a tester. All the work that was supposed to be done was estimated in hours effort using planning poker. To confirm that the effort in hours was calculated correct function point analysis was used. This analysis gave the same hour results as planning poker (with a difference about 10 hours). Then knowing the effort in hours and average salary for every of the roles in the team the cost of effort for project development was calculated.

To ensure that the quality of the software products brainstorming to find possible problems was made. Then all the problems were classified and a fish-bone diagram was constructed. Now knowing the possible problems developers will be able to ensure that they don’t appear in the final product. On the other hand a check-list of possible bug was created. This list will be used by the testers to ensure that the product corresponds to software specification and is qualitative.

Next step was the creation of the possible designed based on software specification and usability testing to ensure that the design is understandable and is going to attract users to use the application. Two groups of 5 volunteers were chose to be usability testers. One of them were testing low-fidelity prototype and other once the high fidelity prototype. Before performing the testing they were given a user case scenario with the specifications of what actions they had to perform. After the testing was performed usability testers were given a questionnaire that contained statements which they had to evaluate from 1 to 10 according to their agreement with the statement. After wall questionnaires were fulfilled they were analysed to see if there were need changes to be performed.

After the design was finished the stage of code was began. It was separated into modules and sub modules. After implementation of each module developers had an working prototype that fulfilled part of specifications. Next modules and sub-modules were implemented:
- Connection with camera module
- Recognition algorithm module:
  - Recognition of squares
  - Recognition of triangles
  - Recognition of circles
- Replacement algorithm module:
  - Replacement of squares
  - Replacement of triangles
- Replacement of circles
- Saving module
- Sharing module

Following the object oriented programming strategy every module was encapsulated in a separate class. For implementation of the connection with camera module and recognition module an computer vision library was OpenCv was used. Since this library is written in C/C++ openCv Android SDK and Android NDK were included in the project as well. For the rest of the modules standard Android SDK was used. A special combination of filters and search algorithm from the openCv library were created for reaching the final result of recognition of primitive shapes specified by software requirements.

Final stage of the development was testing. There were performed two type of testing: automatic and manual. For automatic testing unit testing using black-box technique was used. Each of the modules has it;s own set of test cases (57 in general). Not all the tests passes from the first try, but after minor changes in the source code all the tested passed. Manual testing was performed in the circumstances in which the application is supposed into be used. Manual testing has shown that there are still some issues with recognition algorithm depending on the environment: as an example the amount of light or the angle of the inclination of the phone. Also a system for bug tracking in production environment was mounted using Fabric, which will collect all report of application crashes on the users devices, so the developers could easy track bugs and eliminate them.
Bibliography


(Springer, New-York, USA, 2011)

[3] «Multilayer information representation (augmented reality)» (College of Business and Marketing)


http://cyberleninka.ru/article/n/klassifikatsiya-i-perspektivnye-napravleniya-ispolzovaniya- 
technologii-dopolnennoy-realnosti


[7] Source of the image for the scheme 1.3.1  
http://www.mir3d.ru/upload/iblock/bd2/audi_layar.PNG

[8] Source of the image for the scheme 1.3.1  
http://games.mail.ru/pic/pc/gallery/9e/eb/d27f7d16.jpeg

[9] Source of the image for the scheme 1.3.1  

[10] Source of the image for the scheme 1.3.1  
http://cs412324.vk.me/t77071684/video/y_a017a590.jpg

[11] Source of the image for the scheme 1.3.1  


[19] http://docs.opencv.org/3.1.0/d7/d1b/group__imgproc__misc.html

[21] Source of the image for the scheme 4.4.2  


[26] https://app.ganttpro.com/#!/app/home
## Attachment 1
### Quality management check-list

<table>
<thead>
<tr>
<th>#</th>
<th>Case</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application works on Android 5.0 or later</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Application displays images from camera in real-time</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Application detects primitive shapes: circles, squares and triangles</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Applications detects inly black shapes on white background</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Application detects shapes of size 10 centimeters on the distance from 1 till 2 meters</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Application compliments camera image with appropriate augmented reality image: surfing-board for circle, skateboard for triangle and snowboard for rectangle</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Application draws the augmented reality image of appropriate size</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Application draws the augmented reality image in appropriate place</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Application permits to save image</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Application permits to share image</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Application has tutorial</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Application processes camera image in less then 3 seconds</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Application permits to take picture in no more then two clicks</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Application permits to share image in no more then two clicks</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Application informs user if there is no free space left on the device</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Applications ask for necessary permissions and if it don’t get permission it doesn’t crash</td>
<td></td>
</tr>
</tbody>
</table>
Dear participant,

First of all we want to thank you for participation in our usability testing. In this document you can find explanation of the situation in which you are supposed to use our application prototype. Imagine that you meet with your friend in the Dunkin’Coffee in the centre of Madrid. He shows you a new amazing T-shirt that has just bought in our store. After he tells you that he was told in the store about an application that works with this T-shirt. He downloads the application and asks you to make a photo of him using this application in order to see how this application works. You make the photo and after your friends asks you to share this photo in his Instagram with the description “Man, this T-shirt is really amazing”.

Attachment 3
First Usability Testing Questionnaire.

Dear participant:

Since you have already performed usability testing, we would like to answer some short question about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

2. The design of the application self-explainable.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to to find them at first glance.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

4. After using this application once, I will use it more times.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

5. I will suggest this application to my friends/family.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

6. I will use this application at least once per month.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

7. Would you like to make any special remarks about the design of the application?
   __________________________________________________________________________
   __________________________________________________________________________
   __________________________________________________________________________
   __________________________________________________________________________
Dear participant:

Since you have already performed usability testing, we would like to answer some short questions about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

2. The design of the application is self-explainable.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to find them at first glance.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

4. After using this application once, I will use it more times.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

5. I will suggest this application to my friends/family.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

6. I will use this application at least once per month.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

7. Would you like to make any special remarks about the design of the application?
   The main screen is too empty. It could use more images!

[Signature]

Person A.
Dear participant:

Since you have already performed usability testing, we would like to answer some short question about the prototype you used. In the question from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of you agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree 1 2 3 4 5 6 7 8 9 (10) Agree

2. The design of the application self-explainable.
   Don’t agree 1 2 3 4 5 6 7 8 9 (10) Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to to find them at first glance.
   Don’t agree 1 2 3 4 5 6 7 8 9 (10) Agree

4. After using this application once, I will use it more times.
   Don’t agree 1 2 3 4 5 6 7 (8) 10 Agree

5. I will suggest this application to my friends/family.
   Don’t agree 1 2 3 4 5 6 7 (8) 10 Agree

6. I will use this application at least once per month.
   Don’t agree 1 2 3 4 5 6 7 (8) 10 Agree

7. Would you like to make any special remarks about the design of the application?

   [Handwritten text: I would prefer the buttons to be circled □]

[Signature or Initial: B]
Dear participant:

Since you have already performed usability testing, we would like to answer some short questions about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

2. The design of the application self-explainable.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to find them at first glance.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

4. After using this application once, I will use it more times.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

5. I will suggest this application to my friends/family.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

6. I will use this application at least once per month.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

7. Would you like to make any special remarks about the design of the application?

______________________________
______________________________
______________________________

C
Dear participant:

Since you have already performed usability testing, we would like to answer some short question about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree: 1 2 3 4 5 6 7 8 9 10 Agree

2. The design of the application self-explainable.
   Don’t agree: 1 2 3 4 5 6 7 8 9 10 Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to find them at first glance.
   Don’t agree: 1 2 3 4 5 6 7 8 9 10 Agree

4. After using this application once, I will use it more times.
   Don’t agree: 1 2 3 4 5 6 7 8 9 10 Agree

5. I will suggest this application to my friends/family.
   Don’t agree: 1 2 3 4 5 6 7 8 9 10 Agree

6. I will use this application at least once per month.
   Don’t agree: 1 2 3 4 5 6 7 8 9 10 Agree

7. Would you like to make any special remarks about the design of the application?

   ____________________________________________
   ____________________________________________
   ____________________________________________
   ____________________________________________
Dear participant:

Since you have already performed usability testing, we would like to answer some short questions about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

2. The design of the application is explainable.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words, I was able to find them at first glance.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

4. After using this application once, I will use it more times.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

5. I will suggest this application to my friends/family.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

6. I will use this application at least once per month.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

7. Would you like to make any special remarks about the design of the application?

__________________________________________________________________________
__________________________________________________________________________

80
Dear participant:

Since you have already performed usability testing, we would like to answer some short question about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of you agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

2. The design of the application self-explainable.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to to find them at first glance.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

4. After using this application once, I will use it more times.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

5. I will suggest this application to my friends/family.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

6. I will use this application at least once per month.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

7. I like the color scheme of the application.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

8. I find the elements of the application (buttons, texts, images) to have appropriate sizes and shapes.
Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

9. Would you like to make any special remarks about the design of the application?
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
Attachment 6.1
Usability Testing Questionnaires Answers (Person F).

Dear participant:

Since you have already performed usability testing, we would like to answer some short question about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

2. The design of the application self-explainable.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to find them at first glance.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

4. After using this application once, I will use it more times.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

5. I will suggest this application to my friends/family.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

6. I will use this application at least once per month.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

7. I like the color scheme of the application.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

8. I find the elements of the application (buttons, texts, images) to have appropriate sizes and shapes.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

9. Would you like to make any special remarks about the design of the application?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

F
Dear participant:

Since you have already performed usability testing, we would like to answer some short question about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

2. The design of the application self-explainable.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to find them at first glance.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

4. After using this application once, I will use it more times.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

5. I will suggest this application to my friends/family.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

6. I will use this application at least once per month.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

7. I like the color scheme of the application.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

8. I find the elements of the application (buttons, texts, images) to have appropriate sizes and shapes.
   Don’t agree 1 2 3 4 5 6 7 8 9 10 Agree

9. Would you like to make any special remarks about the design of the application?
   The first screen is unnecessary.
Dear participant:

Since you have already performed usability testing, we would like to answer some short questions about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of you agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

2. The design of the application self-explainable.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to find them at first glance.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

4. After using this application once, I will use it more times.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

5. I will suggest this application to my friends/family.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

6. I will use this application at least once per month.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

7. I like the color scheme of the application.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

8. I find the elements of the application (buttons, texts, images) to have appropriate sizes and shapes.
   Don’t agree  1 2 3 4 5 6 7 8 9 10 Agree

9. Would you like to make any special remarks about the design of the application?


Dear participant:

Since you have already performed usability testing, we would like to answer some short questions about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of your agreement with the phrase. Thank you for your answers.

1. The application is easy to use.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

2. The design of the application self-explanable.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to to find them at first glance.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

4. After using this application once, I will use it more times.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

5. I will suggest this application to my friends/family.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

6. I will use this application at least once per month.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

7. I like the color scheme of the application.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

8. I find the elements of the application (buttons, texts, images) to have appropriate sizes and shapes.
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

9. Would you like to make any special remarks about the design of the application?

________________________________________________________________________
________________________________________________________________________

I
Dear participant:

Since you have already performed usability testing, we would like to answer some short question about the prototype you used. In the questions from 1 to 6 please circle one of the numbers from 1 to 10 which corresponds to the level of you agreement with the phrase. Thank you for your answers.

1. The application is easy to use.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

2. The design of the application self-explainable.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

3. I found graphical design elements of the application (buttons, texts, images) to be placed appropriately. In other words I was able to to find them at first glance.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

4. After using this application once, I will use it more times.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

5. I will suggest this application to my friends/family.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

6. I will use this application at least once per month.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

7. I like the color scheme of the application.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

8. I find the elements of the application (buttons, texts, images) to have appropriate sizes and shapes.  
   Don’t agree  1 2 3 4 5 6 7 8 9 10  Agree

9. Would you like to make any special remarks about the design of the application?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________