**Thread Count Per Square Inch in Woven Fabric using Image Processing**



**SESSION 2018-2022**

**BACHELOR OF SCIENCE**

**IN**

**COMPUTER SCIENCE**

**SUBMITTED BY**

**Aimen Anwaar 18-NTU-1033**

**Hafsa Aslam 18-NTU-1037**

**Muhammad Faizan Ali 18-NTU-1057**

**SUPERVISED BY**

Dr. Rehan Ashraf

**Co-SUPERVISED BY**

Ms. Sana Ikram

**DEPARTMENT OF COMPUTER SCIENCE NATIONAL TEXTILE UNIVERSITY, FAISALABAD-37610, PAKISTAN**

# CERTIFICATION

This thesis, which has been written by Muhammad Faizan Ali, Aimen Anwaar, Hafsa Aslam under the directions of out supervisor and co-supervisor and also approved by all members of thesis committee, has been presented to and accepted Chairman, Department of Computer Science, in fulfillment of the requirement of the degree of Bachelor of Science in Computer Science.

**--------------------------------- --------------------------------**

**Supervisor Date**

Dr. Rehan Ashraf

Assistant Professor

**--------------------------------- --------------------------------**

**Internal Examiner Date**

**--------------------------------- --------------------------------**

**FYP Convener Date**

Prof. Waqar Ahmad

Assistant Professor

**--------------------------------- --------------------------------**

**Head of Department Date**

Dr. Muhammad Asif Habib

Associate Professor / Chairman

# Declaration

We hereby declare that this project report “Thread count per square inch using OpenCV” has been written by us as and all the efforts have been done by us. No part in this report has been copied or taken without mentioning the reference of the article.

**Submit by Signature with Date**

**M. Faizan Ali** ……………………….

18-NTU-CS-1057

**Aimen Anwaar** ……………………….

18-NTU-CS-1033

**Hafsa Aslam** ……………………….

18-NTU-CS-1037

After Completion of the Project development. This project is submitted to the Department of Computer Science at National Textile University, Faisalabad in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science. This project was divided into different weeks and summing up the efforts, This project is equivalent to 32 weeks of full-time studies. We have read the report and confirm that this report meets the minimum requirements for the degree of Bachelor of Science in Computer Science (BSCS).

**Contact Information**

Author(s):

M. Faizan Ali \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

18-NTU-CS-1057 Signature Date

Aimen Anwaar \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

18-NTU-CS-1033 Signature Date

Hafsa Aslam \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

18-NTU-CS-1037 Signature Date

Department of Computer Science Web: [*www.ntu.edu.pk*](http://www.ntu.edu.pk)

National Textile University Phone: +92 41 92 30 081

P.O ± 37610 Faisalabad Fax: +92 41 92 30 098

# Acknowledgement

Thanks to Allah Almighty for this wisdom and perseverance that He has bestowed us during this project and during our whole life: “We can do everything through His Guidance and Who gives us strength”.

Our Special praise to the Holy Prophet (PBUH) and His loved ones, who everlastingly have been a signal of information, shrewdness, and direction for the entire mankind.

We feel a lot of delight in offering my heartiest thanks to our mentors and teachers for their dynamic supervision and invaluable support during the entire period of our bachelor’s and specifically for this Final Project, as without their support and guidance, it might be not possible for us to reach at this stage.

This project really required huge effort and declarations and we would be very thankful to our Group Members and Class fellows who helped us a lot to complete this project. Their Assistance and motivation have a big role in this accomplishment. The great contributions of group members have made it possible. We want to express our gratitude to our teachers as mentors. We are thankful for their aspiring guidance, invaluably constructive criticism, and friendly advice during the project work.

In addition, We would like to pay our huge appreciation to our close companions and especially our group members for the full support and remaining a helping hand in any type of difficulty. We have always been a supportive hand for each other any type of obstacle and pressure.

Sincerely Love to our Parents who generally love us, give us care and do countless prayers for us. They generally guide us and back us when I face any kind of obstacle in our lives.

We would like to special thanks to our Honorable Supervisor **Dr. Rehan Ashraf** and our Co-Supervisor **Ms. Sana Ikram** for this excellent supervision and management. We are immensely grateful to them and much obliged to respect them. It’s his pure dedication to give us time from hectic routine which made this project to complete.

# Abstract

In the current era, Life is getting much faster than before and things are getting automated due to different kinds of machines. Our final year project is totally based on the automation of counting Threads Per Inch in woven fabrics. This is a manual task that takes so much time and effort to count manually using pick glass and other means of counting. We have tried our level best to develop this system. The major objective was to automate this process to count the no. of Threads Per Inch because it was taking so much labor time and cost. This was still a hectic task because all the project was based on R&D (Research and Development). So, the proposed solution was to implement machine learning Image Processing algorithm to process the fabric sample image by making the background black and getting the white pixels prominent. We have developed a mobile app on the abstract level to hide all the background processes and get input from the individual as well as show the actual result in the form of counts of warp and weft and total count in numeric value. This app can take an image and send them for the background process and then returns the result. This struggle has made us to contribute to the problem that was a major issue for textile industries as well as naïve consumers. We have tried our level best to contribute to this problem and also did our best to solve this problem by using image processing.

Table of Contents

[CERTIFICATION 2](#_Toc111645248)

[Declaration 3](#_Toc111645249)

[Acknowledgement 5](#_Toc111645250)

[Abstract 6](#_Toc111645251)

[Table of Figures 12](#_Toc111645252)

[Chapter 1: Introduction 13](#_Toc111645253)

[1.1 Introduction: 14](#_Toc111645254)

[1.2 Problem Statement: 14](#_Toc111645255)

[1.3 Purpose 14](#_Toc111645256)

[1.4 Project Goals 15](#_Toc111645257)

[1.5 Objectives of Project 15](#_Toc111645258)

[1.6 Project Limitation 15](#_Toc111645259)

[1.7 Project Scope 15](#_Toc111645260)

[1.8 Project Scheduling 16](#_Toc111645261)

[1.9 Risk Management 16](#_Toc111645262)

[1.10 Types of Risks 17](#_Toc111645263)

[1.10.1 Cost Risk 17](#_Toc111645264)

[1.10.2 Performance Risk 17](#_Toc111645265)

[1.10.3 Change of Requirements 17](#_Toc111645266)

[1.10.4 Unforeseen Circumstances 18](#_Toc111645267)

[1.10.5 Scheduling and Time Risk 18](#_Toc111645268)

[1.10.6 Requirement Gathering 18](#_Toc111645269)

[1.10.7 Architecture and Design 18](#_Toc111645270)

[1.10.8 Coding and Implementation 19](#_Toc111645271)

[2.1 Existing System: 21](#_Toc111645272)

[2.2 Current Scenario: 23](#_Toc111645273)

[2.3 Edge Detection: 23](#_Toc111645274)

[2.4 High-Speed, real-time vision system for texture tracking and thread counting: 24](#_Toc111645275)

[2.5 Measuring Thread densities using Fourier Transformation: 25](#_Toc111645276)

[Chapter 3: Methodology 27](#_Toc111645277)

[3.1 Choosing the Right Methodology: 28](#_Toc111645278)

[3.2 Agile Development Methodology: 28](#_Toc111645279)

[3.3 Rapid Application Development: 28](#_Toc111645280)

[3.4 Spiral Model: 29](#_Toc111645281)

[3.5 Extreme Programming: 29](#_Toc111645282)

[3.6 Waterfall Model: 29](#_Toc111645283)

[3.7 Dev Ops: 29](#_Toc111645284)

[3.8 Adapted Methodology: 30](#_Toc111645285)

[3.8.1 History of Scrum: 30](#_Toc111645286)

[3.8.2 Pros: 30](#_Toc111645287)

[3.8.3 Cons: 31](#_Toc111645288)

[3.8.4 Scrum Framework: 31](#_Toc111645289)

[Chapter 4: System Requirements 32](#_Toc111645290)

[4.1 Project Scheduling 33](#_Toc111645291)

[4.2 Hardware Requirements. 33](#_Toc111645292)

[4.3 Software Requirements. 33](#_Toc111645293)

[4.4 Functional Requirements. 33](#_Toc111645294)

[4.4.1 User Requirements. 34](#_Toc111645295)

[4.4.2 User Sign in. 34](#_Toc111645296)

[4.4.3 Authentication Of user. 34](#_Toc111645297)

[4.4.4 Take Picture of the Sample. 34](#_Toc111645298)

[4.4.5 Upload Picture of the Sample. 34](#_Toc111645299)

[4.4.6 User’s Profile: 35](#_Toc111645300)

[4.4.7 Database: 35](#_Toc111645301)

[4.5 Non-Functional Requirements: 35](#_Toc111645302)

[4.5.1 Performance: 35](#_Toc111645303)

[4.5.2 Security: 35](#_Toc111645304)

[4.5.3 Availability: 35](#_Toc111645305)

[4.5.4 Maintainability: 36](#_Toc111645306)

[4.5.5 Reusability: 36](#_Toc111645307)

[4.5.6 Extensibility: 36](#_Toc111645308)

[4.5.7 Operability: 36](#_Toc111645309)

[4.6 Overall system Requirements: 36](#_Toc111645310)

[4.6.1 Technical Feasibility Study: 37](#_Toc111645311)

[4.6.2 Economic Feasibility Study: 37](#_Toc111645312)

[4.6.3 Development cost: 37](#_Toc111645313)

[4.6.4 Operational feasibility: 37](#_Toc111645314)

[4.7 Cost Benefit Analysis: 38](#_Toc111645315)

[4.7.1 Formula for Cost Benefit Analysis: 38](#_Toc111645316)

[4.8 Class Diagram: 38](#_Toc111645317)

[4.9 Use Case Diagram: 40](#_Toc111645318)

[4.10 User Authentication: 41](#_Toc111645319)

[4.11 Use Case Tables: 42](#_Toc111645320)

[4.11.1 Sign Up: 42](#_Toc111645321)

[4.11.2 Sign In: 43](#_Toc111645322)

[4.11.3 Detection of Thread count: 44](#_Toc111645323)

[Chapter 5: Architecture Design 45](#_Toc111645324)

[5.1 Architecture: 46](#_Toc111645325)

[5.2 System Architecture: 46](#_Toc111645326)

[5.3 Sequence Diagrams: 46](#_Toc111645327)

[5.3.1 User Sign-In Sequence Diagram: 47](#_Toc111645328)

[5.3.2 User Sign-Up Sequence Diagram: 48](#_Toc111645329)

[5.3.3 Image Processing Sequence Diagram: 49](#_Toc111645330)

[5.4 Activity Diagram: 50](#_Toc111645331)

[5.4.1 User Sign up Activity Diagram: 50](#_Toc111645332)

[5.4.2 User Sign in Activity Diagram: 51](#_Toc111645333)

[5.4.3 Image Processing Activity Diagram: 52](#_Toc111645334)

[Chapter 7 User Interface 53](#_Toc111645335)

[7.1 Interfaces: 54](#_Toc111645336)

[7.1.1 Splash Screen: 55](#_Toc111645337)

[7.1.2 Welcome Screen: 56](#_Toc111645338)

[7.1.3 Sign Up Interface: 57](#_Toc111645339)

[7.1.4 Sign In Interface: 58](#_Toc111645340)

[7.1.5 Take Image of Fabric Sample Interface: 59](#_Toc111645341)

[7.1.6 Calling API to Process Image Taken from Gallery or Camera: 60](#_Toc111645342)

[7.1.7 Result Interface: 61](#_Toc111645343)

[Chapter 8: Test Cases 62](#_Toc111645344)

[8.1 Black Box Testing: 63](#_Toc111645345)

[8.2 White Box Testing: 63](#_Toc111645346)

[8.3 Test Cases: 63](#_Toc111645347)

[8.3.1 Test Case 1: Sign Up: 63](#_Toc111645348)

[8.3.2 Test Case 2: Sign In: 63](#_Toc111645349)

[8.3.3 Test Case3: Take Image from Camera: 64](#_Toc111645350)

[8.3.4 Test Case4: Take Image from Gallery: 64](#_Toc111645351)

[8.3.5 Test Case5: API\_Call: 64](#_Toc111645352)

[Chapter 9 References 65](#_Toc111645353)

# Table of Figures

[Figure 2.1: Warp Weft Interloped [1] 21](#_Toc111627170)

[Figure 2.2: Pick Glass [2] 22](#_Toc111627171)

[Figure 2.3: Lunometer [3] 22](#_Toc111627172)

[Figure 2.4 Structure of Fabric Density [8] 26](#_Toc111627173)

[Figure 3.1 Scrum Model [16] 31](#_Toc111627174)

[Figure 4.1 Overall system Class Diagram 39](#_Toc111627175)

[Figure 4.2 Overall system Use case Diagram 40](#_Toc111627176)

[Figure 4.4 Authentication of the user 41](#_Toc111627177)

[Figure 5.1 Sign in Sequence Diagram 47](#_Toc111627178)

[Figure 5.2 Sign Up Sequence Diagram 48](#_Toc111627179)

[Figure 5.3 Image Process Sequence Diagram 49](#_Toc111627180)

[Figure 5.4 Sign Up Activity Diagram 50](#_Toc111627181)

[Figure 5.6 Image Processing Activity Diagram 52](#_Toc111627182)

[Figure 7.1 App Splash Screen 55](#_Toc111627183)

[Figure 7.2 App Welcome Screen 56](#_Toc111627184)

[Figure 7.3 Sign up Screen 57](#_Toc111627185)

[Figure 7.4 Sign in Screen 58](#_Toc111627186)

[Figure 7.5 Upload Image Screen 59](#_Toc111627187)

[Figure 7.6 API Call 60](#_Toc111627188)

[Figure 7.7 Result 61](#_Toc111627189)

# Chapter 1: Introduction

# Introduction:

In this chapter, we will discuss the short introduction about the project, purpose, goals, objective, scope and project schedule with risk management factors. These describe relative domain problem, solution of the problem statement, effect on the user, market scope, and manage risk factors. Project Scheduling describes the project timeline as the period in which you will complete your project.

# Problem Statement:

In this modern world, the main problem of every textile industry or factories to count the number of yarns in fabrics using manual devices These manual devices are mainly small thread counting microscopes and the traditional use of a Lunometer which is really a hectic task for them. It is time-consuming and a waste of resources such as time and cost. There is no automated system to count the number of threads per square inch in a short period of time. If there is a client and he demands a cotton fabric of 105 counts. Once it is developed, it is a very difficult job to check the actual number of threads that are provided by the seller. Also, the sellers take too much time to do this job. There are many mathematical algorithms to calculate these things manually. So, it is a big problem and should be solved.

# Purpose

The main purpose of developing this system is to automate the process of counting the number of threads per square inch in an automated manner so that it can really help the consumer as well as the seller to count the number in an efficient manner. It can really decrease the time period and cost. A real-time camera can be used for counting the number very fast and efficiently using computer vision or any other camera that can take a picture of good resolution of fabric. The automated system can be programmed to output the number of threads according to the type of fabric as input given by the user. It can be very useful in any textile industry and a layman can use it either.

# Project Goals

The main Goals of our team to create a highly responsive product, so we can provide the officials or authorities a highly and fast way to check the number of counts per square inch in the woven fabrics. We will develop an easy user-friendly interface that will be really easy for both consumer and seller and also a common person can use it efficiently. We will provide this system globally as well as in Pakistan. We can also provide this system to both seller and consumers to check the fabric counts in very little time. As Mobile apps are trending nowadays. So, users may easily understand the working and they can also check the fabric anywhere anytime they want because it will be available in your mobile phone. So, we will try our level best to develop this software as a mobile app that will be really helpful for all of the textile-related people.

# Objectives of Project

* Providing a user-friendly interface for the user to use efficiently.
* User can view the details of the fabric.
* The System will auto alert if the number of threads is not up to the mark.
* Consumer/Seller can get a detailed view of the fabric and report to the desired person if the user will not get the expected output.
* Providing user Registration who will use this application.

# Project Limitation

The main project limitation can be the use of a less-resolution digital camera because this can be an issue to count the number of threads per square inch in woven fabric. A bad resolution camera cannot detect the actual warp weft interlopes because pixels can be damaged so the program may not detect the threads more accurately and efficiently. Errors can me minimum if you use a High-Definition Digital Camera For Example DSLR.

# Project Scope

For creating the limits for our project, we first decided the actual domain of the current problem in which we are going to apply our idea and deploy this project in the textile industry only. This project can be made flexible for other domains of textile industries that require some extra functionalities related to fabric or thread specifically. The ultimate feature of this project is universal which can be applicable to any other textile related domain

# Project Scheduling

Project planning is actually a scheduled time period that we follow to accomplish the project such as a Gantt chart to plan. **Gantt chart** helps the team to plan the work around the provided deadlines and properly allocate resources. Tasks are completed in the given time that is mentioned in the Gantt chart. It explains the start and finishes dates of major activities to be performed or already performed to complete the overall project. Project resources are time and cost. Time and cost both are flexible so they can be changed if there are ups and downs during the development process.

|  |  |
| --- | --- |
| **Activity** | **Tentative Date** |
| **Project Proposal** | 15-11-2021 |
| **Project Management Plan** | 31-11-2021 |
| **Project Requirements Specification** | 27-12-2021 |
| **Coding and Implementation** | 31-04-2022 |
| **Project Test Documentation** | 15-06-2022 |
| **Submission of Final Documentation +Viva Voice + Defense** | 28-8-2022 |
|  |  |

Table 1.1 Project Scheduling Chart

# Risk Management

Risk management is a major part of the project. Risk management means the future risks that can happen during the development process. This can be any cost or time issues or any other compatibility issues that happen for the user as well as the consumer. So, this type of management reduces the risks that can be dangerous for the expected output of the project. As we are developing a system for thread counting using image processing, there can be many ups and downs during the development so we should be aware of future problems such as cost, performance, design issues and scheduling problems etc. There are some other risks tha are unpredictable which we will get to know when we will proceed further.

# Types of Risks

There are many types of risks that can happen during the development lifecycle.

# Cost Risk

Cost Risk is the risk that the project costs more than expected before starting the project. Cost is always approximate and not exactly mentioned. Cost analysis the done before starting the development of project. If we want to minimize the cost risk in the future. We have to further divide our project into small chunks and analyze each chunk one by one. Categorize the project into different parts such that management cost, hardware cost so that we can precisely divide the total cost.

# Performance Risk

The performance risk is the latency of the system, program or any other software that shows in future that will not deliver as expected. There may be performance risk in our project that may occur in future. There can be Camera result issue or Software other performance risk can occur. We will try our level best to overcome all the performance risks to make the system optimize and effective. It may reduce the 30 to 50% labor of persons who wants to count the no. of threads per square inch using manual means.

# Change of Requirements

Change of requirements means there may be many changes during the process of development. The Requirements can change according to the client and by the owner who will own this system. We will try our level best to adapt all the requirements of this software and gather each and every requirement that will be needed to develop this software. There may be many requirements that may change during the development of the system.

# Unforeseen Circumstances

Nobody knows what can happen in future. So, we have to ready for the rainy day. There can be many technical difficulties or failures, revolutionary changes in technology or any natural disasters. These risks may or may not effect on our project. Most Unfortunate possibility can be anything sever you can imagine.

# 1.10.5 Scheduling and Time Risk

Carefully dividing the project development into small modules and solving them step by step and adding time to develop modules of the whole project can acquire a proper schedule. Scheduling risks may occur due to many factors such as if your team members are non-serious about the project.

# Requirement Gathering

Most major risk or issue cause during requirement gathering is the continuous change of the requirement as we are also not sure about our final decision and what will happen in the future. Requirements can be changed, and development process can be changed. Improper time estimation may also occur in the development lifecycle. This can cause the project to take more time to complete.

# 1.10.7 Architecture and Design

Every previous risk really effects on the next stages of development. During architecture and designing, the major and foremost risk is the cost risk. It’s the most basic development stage in which it is to be defined how our project will work in future and how much cost will be required in the development of the project. Designing the architecture of project is the most crucial task for any designer because the need of the hour is to select which developmental model you are going to select whether it is a waterfall model or any other model depending upon the state of the art.

# 1.10.8 Coding and Implementation

Coding and implementation are the most major task of the project development. This can be most time taking process of the development life cycle. Programmers have the most difficult task to perform this task in an efficient way. After the Coding is complete, implementing the project may or may not a crucial task for all the team. During Coding and implementation time and scheduling is crucial. An Average team management can also be affected in code integration during the development process. Such risks can extend the time load of the development.

**Chapter2: Literature Review**

# Existing System:

The Standard counting techniques propose two methods to find the density of woven fabric per square inch irrespective of their composition (i.e.: whether they are made of cotton, silk wool, jute, man-made fiber, or a combination of two or more such fibers), manufacturing process and finishing treatments. **Warp threads** are the threads which lie along the length of fabric as woven, the vertical threads. On the other hand, **Weft Thread** is the threads that lie across the length of woven fabrics, the horizontal threads.

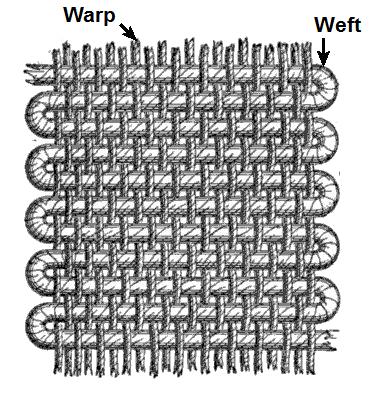


Figure 2.1: Warp Weft Interloped [1]

There are two methods that are using manually for counting the number of threads per square inch:

1. The number of threads in per square inch are detected using a microscopic glass that has a one-inch square attached with it. The square inch is placed on the piece of cloth and one count the number of threads horizontally and vertically manually.



Figure 2.2- Pick Glass [2]

1. An instrument named Lunometer is also used to determine the number of threads. It counts the threads and number of lines with an accuracy of +/- 1%. It is laid across the sample of cloth in proper orientation. It measures in two units’ inches and centimeters.



Figure 2.3: Lunometer [3]

# 2.2 Current Scenario:

Currently there are only fewer (almost near to 1%) of industries using automated thread counting mechanism using image processing. There is not such system installed in any industry as well as in any factory. Many of the industries perform this task manually and that takes so much time in counting the threads per square inch. They are just counting these things through 2 3 apparatuses only by industry workers and officials of industry or factory. It means right now there is need of convergence of the old techniques.

We are going to design a system, which will be very useful for our proposed Industry “Abdul Razzaq Weaving Factory” as well as for the whole areas of industries and other factory areas.

It is going to provide following services to the industries:

* Thread Counting Facility
* Save a lot of time.
* Save a lot of effort.
* Manage a Centralized Database.
* Integrity, accuracy, and consistency in the counting mechanism.
* Accuracy in edge detection.

There may be other thesis available in the libraries and on the internet that will be really helpful for us during the development of the project like this: [4].

# 2.3 Edge Detection:

We will definitely see the basics of edge detection using different edge detection algorithms like: ISEF, Canny, Marr-Hildreth, Sobel, Kirsch, Lapla1 and Lapla2 etc. described by M Sharifi with his team members in their research paper [5].

It is a machine learning based approach where cascade functions are trained from many negative and positive images that will be used to detect counts of threads. In this technique, models are trained and then these models are used to perform the task as expected.

Here we will chip away at edge identification. At first, we should distinguish which sort of the edge it **is,** whether there is a stage, Ramp, Line or Roof. Whether the power of picture is spasmodic, whether the force of picture unexpectedly changes from one worth on one side of the irregularity to the different worth on the on the opposite side or line discontinuities where the picture force suddenly changes esteems however at that point gets back to the beginning qualities with-in some brief distance.

# 2.4 High-Speed, real-time vision system for texture tracking and thread counting:

This paper has been taken through google scholar cited by: [6]

A High-Speed continuous vision framework was created for the surface following and string including. It was a programmed sewing machine that required a vision framework that can perceive the texture examples and texture movements. This framework can identify various examples of textures and surfaces. A constant vision framework can consistently check a little area of texture close to the sewing needle and follow tiny movements. In this framework, the need showed up that the framework ought to be fit for counting the no. of strings per inch square which limits the neighborhood market endeavors and furthermore the texture bends on the following exactness. Counting strings shows keeping a total measure of twist and weft strings that have gone through a middle point.

To plan a super quick and strong vision framework that will be capable of string counting, [7] proposed the model in which the idea of string count was first presented, they utilized the Harris corner identification calculation to distinguish corner elements and texture interpretation, and 2D quick Fourier change (FFT) to follow texture points. Nonetheless, the impediment was that the (FFT-calculation) doesn't function admirably for precisely assessing a little pivot point (for example 0.1 °).

Checking the texture strings is connected with tracking down the surface routineness or nature of the texture. In the examination paper, they identified the granularity (i.e., the size of surface natives) for the overall texture surfaces with basic edge recognition methods. Thus, to track down the examples or design of Near Regular Texture (NRT), Hays et al. proposed his paper who previously figured out a grid tracking down the issue as a higher request correspondence issue. This strategy utilizes interest point indicators that consistently propose and proclaims related surface natives, and afterward trust for the ideal grid task by amplifying the pair-wise visual comparability and the mathematical consistency.

Thus, they proposed a creative surface, global positioning framework that tracks the surface designs and recognizes these texture weave designs at an exceptionally high velocity. They have embraced include extraction procedures that won't just distinguish highlight focuses to have the right example matching between pictures yet, in addition, produce layout proposition and the identification of consistent cross-section. For finding the solid neighborhood cross-section, they proposed a computationally productive calculation that uses both nearby appearance similitudes and worldwide topological connections. They have delivered this framework effectively to recognize denim texture designs and string counting, expounding its high potential for ongoing programmed material assembling procedures.

# 2.5 Measuring Thread densities using Fourier Transformation:

To diminish the tedious and lab-escalated manual undertakings to gauge the string densities or manual review, a Fourier change strategy is proposed in this paper that was utilized to count the quantity of string per square inch of the woven textures. This paper has been taken from Google Scholar referred to by [8]. To begin with, Theories of the Fourier change, yarn picture remaking, and limit strategy were presented. Then the means of picture procurement, the Fourier change of the texture picture, and highlight examination are gotten in the recurrence space. Picture handling of the texture yarns was presented, and limit handling are broken down. Then the thread count was calculated through this method which is called thread density. This method of Fourier Transformation for the thread count can actually detect and count thread densities accurately and properly and also can replace the manual work.

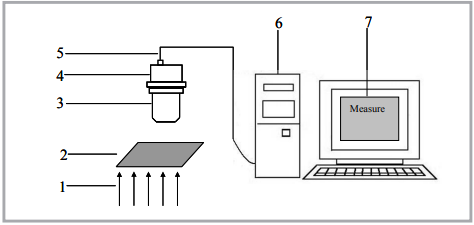


Figure 2.4 Structure of Fabric Density [8]

*Structure of fabric density measurement system 1- transmitted light, 2- fabric sample, 3-microscope, 4- digital camera, 5- USB data cable, 6- computer, 7- software*

Picture investigation has been viewed as the most proficient and compelling technique to count the string densities of woven textures. This entire cycle can be isolated into two significant stages. In the first place, including the number of yarns in the texture picture, and the second is to quantify the actual component of the texture picture. For the most part, the subsequent step is acted in the step of picture procurement. Yet, this momentum research centers around the absolute initial step. The entire course of this strategy is classified into two fundamental classifications

1. Frequency Domain

2 - Time Domain.

A Woven texture picture is a period surface that is shaped by twists and a weft of yarns. It very well may be changed into a recurrence area through Fourier change. The pinnacles or edges of the pictures are addressing recurrence of the occasional components which are situated in the power range that is utilized to compute the string thickness. [9-14]. The occasional data of the twist and weft plan were utilized to recreate the picture of the texture. This reproduction of the picture that contained just twist and weft yarns data is taken on or centered to perceive the thickness of the woven texture utilizing picture handling and Fourier Transformation. [8, 15].

# Chapter 3: Methodology

To deal with an undertaking effectively, the improvement group should pick the product improvement technique that will turn out best for the task. All systems have various advantages and disadvantages and exist for various reasons. Here is an outline of the most utilized programming advancement approaches.

# Choosing the Right Methodology:

There are various methodologies that are available in the field of software engineering and it’s a hectic job to select which methodology fits in the context of the current working project. Developers are spoilt for choices the right methodology. Most of the methodologies fall under the categories of waterfall, iterative or continuous model.

The Iterative model provides an alternative way for software development that focuses on the rigid documentation and design patterns but focus on the constant revisions. Agile and Scrum are two famous iterative methodologies for the software development.

This Continuous development was inspired by Toyota Production system. It is about minimizing the interruptions and ensuring the flow between different phases of development.

# Agile Development Methodology:

Designers utilize the nimble improvement philosophy to limit risk (like bugs, cost, and evolving necessities) while adding new usefulness. In light-footed technique, designers foster the product in emphasis that contain little additions of the new usefulness. There are various types of spry advancement techniques, including scrum, gem, outrageous programming (XP), and highlight-driven improvement (FDD).

# Rapid Application Development:

In the RAD model, the utilitarian modules are created lined up as models and are coordinated to make the total item for quicker item conveyance. Since there is no definite preplanning, it makes it simpler to consolidate the progressions inside the advancement interaction.

RAD projects follow an iterative and gradual model and have little groups including engineers, area specialists, client agents, and other IT assets working logically on their part or model.

# Spiral Model:

The spiral model is a framework improvement lifecycle (SDLC) technique utilized for the risk the board that joins the iterative advancement process model with components of the Waterfall model. The winding model is utilized by programmers and is inclined toward huge, costly, and confounded projects.

The winding model empowers continuous deliveries and refinement of an item through each period of the twisting as well as the capacity to construct models at each stage. The main component of the model is its capacity to oversee obscure dangers after the venture has started; making a model makes this practical.

# Extreme Programming:

Extreme programming (XP) is a variety of the spry programming improvement strategy in light of five qualities: effortlessness, correspondence, criticism, regard, and fortitude. Outrageous programming highlights little groups of engineers, directors, and clients who work intently together during short improvement cycles. It advances the utilization of straightforward plans and practices and incessant little deliveries.

# Waterfall Model:

In a waterfall model, each stage should be finished before the following stage can start and there is no covering in the stages.

The cascade Model represents the product improvement process in a direct successive stream. This implies that any stage in the improvement cycle starts provided that the past stage is finished. In this cascade model, the stages don't cover them. In this Waterfall model, ordinarily, the result of one stage goes about as the contribution for the following stage consecutively.

# Dev Ops:

DevOps is likewise a product quality improvement approach in which you foster unrivaled quality programming proficiently and really and with greater dependability. It contains different transformative phases like the ceaseless turn of events, persistent mix, nonstop testing, consistent organization, and constant checking.

# Adapted Methodology:

Subsequent to Brainstorming, we examined our concerns and the arrangement. After this investigation, we have chosen to adjust the approach of the SCRUM Developmental strategy.

Scrum is an agile way methodology as it is lightweight, iterative, and incremental framework for developing, delivering complex products. This approach is adapted as traditional, sequential approach to product development. In the development where the requirements are not fully known, and we need iterations during the development. We use the methodology of scrum.

Scrum is ostensibly the most adaptable programming improvement strategy. This philosophy is ordinarily utilized in the creation where prerequisites are not really cleared and there need for numerous cycles and changes in necessity during the result of programming. Our venture depends on R&D (Research and development). So, there can be heaps of changes in the thoughts and creation grouping. The entire situation can be changed, and necessities can be reexamined. So, Scrum is the most adaptable strategy that fits this ongoing situation. So, we have chosen to adjust this technique.

Scrum depends on the Product Owner, Scrum Master, and the improvement group.

The item proprietor takes input from the client and guarantees that regardless of whether the group is satisfying the client's prerequisites totally. Scrum ace goes about as facilitator and spotlight on what the functioning group knows about the scrum cycle or not. The group takes the charge of executing the general turn of events.

# History of Scrum:

Hirotaka Takeuchi and Ikujiro Nonaka introduced the terminology of SCRUM in the context of **The New Product development Game** in their 1986 Harverd business Review Article.

# Pros:

* Short Iterations allow quick resolutions to problems.
* Scrum is very responsive as this development ensures regular feedback.
* Scrum is economical and effective.
* Regular meetings ensures that team members on the same pace.
* Contributions of individual members of the Scrum are appreciated through the scrum meetings.

# Cons:

* All the team members should be equally skilled and committed to the scrum process.
* The daily and constant scrum meetings can be frustrating for the development scrum team.
* May time period of production and delivery to the market can be increased.
* Not suitable for large projects.

# Scrum Framework:

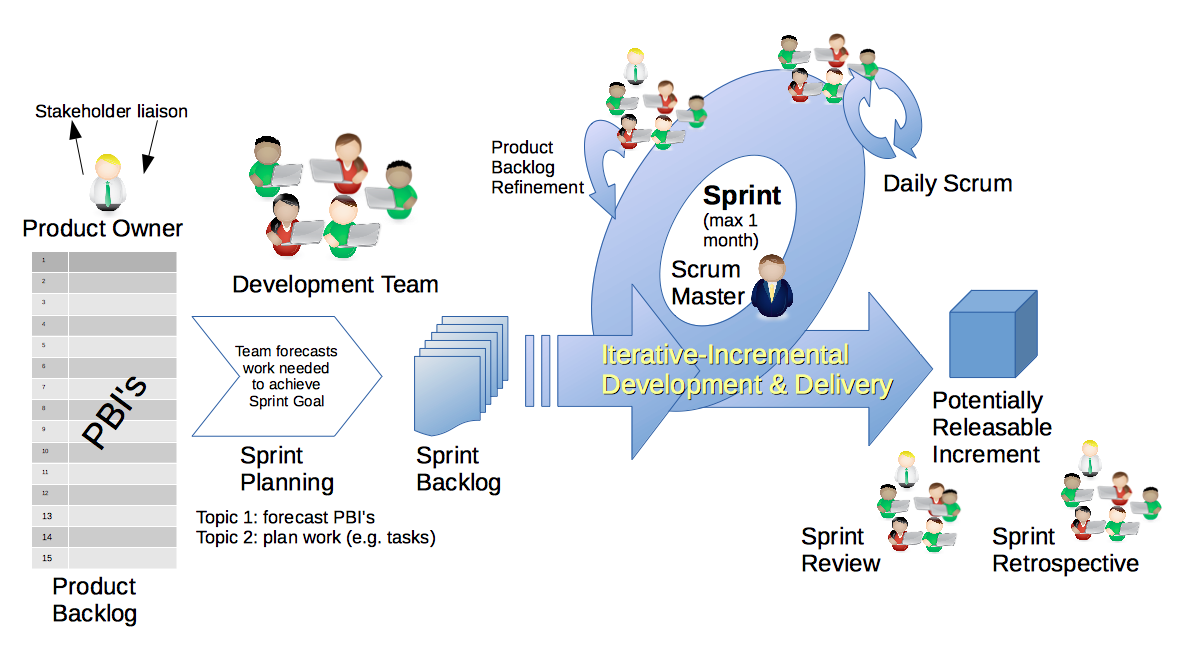


Figure 3.1 Scrum Model [16]

# 

# Chapter 4: System Requirements

# Project Scheduling

The first thing in developing the system is gathering requirements. By understanding the requirements, we develop the system. Understanding the requirements is really a crucial part of the requirement gathering phase because requirements are gathered through the description of the user desired system and should understand the exact meaning of descriptions through which you extract the technical requirements. System requirements are all about the project that will run on a machine by keeping in view the specifications of the project. The requirement gathering task provides information about the performance of the system like the correct flow of the system.

# Hardware Requirements.

Hardware requirements describe the minimum requirements on which the system runs smoothly with having slow or stuck on a particular task without having crash or failure.

The requirements of our system are as follows:

* High resolution and High-speed camera.
* 2.6 GHz processor.
* 2GB of Ram.
* 700MB Graphic Memory.
* ROM desirable.
* I/O ports.
* Good Quality LED.

# Software Requirements.

Software requirements indicate the platform on which the system is developed and on which the system will run. For system development, MATLAB, OpenCV is required for the processing of the image that will be needed to change the DPI and cropping mechanism.

# Functional Requirements.

Functional requirements refer to the working of the system that how the system works and how the outsider interacts with our system, like if a user have to use the system, he/she must go through the following steps:

# User Requirements.

To become a member of our system or our App. He or she must register by providing the correct email addresses and passwords. After giving email passwords, passwords, they will be authenticated by sending a confirmation email to the email address. Other Information like First Name, Last Name, Phone Number will also be needed in the registration process.

# User Sign in.

After registering with the correct email and password, one should log in to the app so that recent information and recent pictures should be saved. Now just open the system by providing the correct credentials.

# Authentication Of user.

Authentication of the user is really an important task as when someone provides email and password the email should be verified that is it a correct email or not. So, all the users must be verified from its national Identity Card and enter its unique username and password.

# Take Picture of the Sample.

Now after completing the process of registration and login process successfully. The very main task is to take the picture of the fabric sample by the eye view. Try to use a good camera in full light so that you can get the actual result or a good quality image. At least 48Mp Camera is required to take a good quality picture of the fabric sample which is woven fabric through eye view. Decreasing the camera quality will be a limitation because low pixels will not give the actual image of threads.

# Upload Picture of the Sample.

The best approach is to take pictures using DSLR camera or you can use your mobile camera but with good results. Also, the image can be taken with the zoom in effect. User also have the option of uploading the pictures of fabric samples through mobile gallery.

# User’s Profile:

There is a feature of the system that user can see his profile and can change or edit their profile like changing the profile picture and changing the email address.

# Database:

Our system can interact with various types of databases like MySQL, Oracle and SQL server and also firebase with is provided by Google. It is like NoSQL database where we don’t need any SQL Queries for the CRUD operations. For our system we have used Firebase with our system because it is easy to use. For the flutter app, Firebase works perfectly otherwise other database use API to manage data from Apps.

# Non-Functional Requirements:

Non-Functional Requirements (NFRs) depicts the system attributes such as system security, reliability, performance of the system. It contains some additional features that are the beauty of the system but not affect the basic behavior of the system. It also known as system qualities.

# Performance:

The best quality of the system is to respond on time in any situation falls in the category of excellent performance. Keeping in view about this factor of the system. It plays a major role in the development of a good system. Performance also refers to the response time of the system.

# Security:

Making a system secure is an essential part of the development of the system. It is an essential factor in case you don’t want to harm your system by any means. The ability of the system to run in saver mode without any outside threat is called security of the System.

# Availability:

The availability is the numeral factor which can be calculated. It requires the ability to be fully committed to its recommended performance which refers to availability of the system. That is the system should be available when required. For Example, a system can, be available for 20 hours a single day. It has the availability factor of 0.80 approx. otherwise its performance may be affected.

# Maintainability:

All systems in the world require maintenance from time to time in order to run smoothly and also not all systems has 100% fault tolerance. They can break down sometimes and maintainability ensures the system that it will be repaired and removing the error in no time.

# Reusability:

Our system is highly reusable because of the single module which can be used anywhere where thread count is needed anywhere in industry as well as for a lay person. The higher the reusability the greater the chances that out project can be demanded in other domains.

# Extensibility:

The secondary aim of the project is to make our project extensible so that it can be handled by giving our project enough flexibility and also to create a best version of it. Extensibility means future modules and or functional requirements can be added to the system without effecting the existing system.

# Operability:

To make systems more operable, automated as well as manual functional handling is added. In case of crash of any module. A user can add details of the subject. This helps in better handling the system and it increases the factor of availability.

# Overall system Requirements:

According to the functional and non-functional requirements, that basically tells us about the features of the system and flow of data, rights of the user and admin.

There are some feasibility studies that are concern with the development of the project and gives you thorough information about how to deal with the system and what are the abilities of the system.

# Technical Feasibility Study:

If you want to check the technical feasibility study of our project, have to study and review some questions, like whether this project may help in counting the number of threads per square inch of woven fabric, or whether this project has the capability of adaptability so that it can be deployed in any textile field and help them for the quality check of the woven fabric.

So yes, our project is able to adapt in any textile industry or field as the system we are making can be install in any type of computer or mobile phone but need a good camera for this purpose and can be easily converted to web application or desktop application.

# Economic Feasibility Study:

This feasibility study concerns with the cost of the system that whether this system is affordable for a lay person or not. Well, this project may need some investment in buying a good high-resolution camera or a high-resolution camera mobile phone to get a clear or high-definition picture of woven fabric sample.

# Development cost:

Cost in development of this project may get high because a high-resolution digital Camera is required to take High-definition Pictures of Woven Fabric samples or a Mobile phone with High Megapixel camera. Major devices and technologies used in development of the project

* Hp Laptop with processor i7 7th generation.
* OpenCV for the image processing.
* Python.

# Operational feasibility:

This project has solved many real-life problems related to quality check of woven fabrics in a fast and efficient way. As its responses very fast in recognizing the thread count per square inch when a picture of woven fabric sample is upload or taken. It performs many operations which include:

* Recognizing the weave pattern.
* Cropping the image in a way that square inch can be determined of the fabric.
* Counting the light and dark pixels.
* Count the number of threads per square inch of woven fabric.

# Cost Benefit Analysis:

The cost is the of the project in project management is the cost which is compared to the development of the project and the benefits that can we get after development of the project. This starts with a list of expenses of the project during the development process and the benefits that can we get after development. This is the most important step of the project requirements and project management that helps the professionals to determine the pros and cons of the project after development.

For a successful CBA, the professionals should understand and identify the correct pros and cons of the project and costs and benefits of the proposed actions or event.

According to our project, till now there is ZERO investment and ZERO cost financially but there is investment of time and effort that will lead to the great benefits of the project in future. There are many benefits that will be after development of the project. Suppose we have ZERO Investment of money but have investment of time and effort. So, there will be 100% profit of this project as a final year project.

Inflammation of the cost can be ignored on assumptions because all items rise at the same rate. Also, relative prices also increase from time to time such as oil prices and taxes.

# Formula for Cost Benefit Analysis:

The formula to calculate the cost benefit analysis is given below:

**Benefit-Cost Ratio = ∑ Present Value of Future Benefits / ∑ Present Value of Future Costs.**

# Class Diagram:

Class diagram is the groundwork of item situated displaying. It is utilized for both general reasonable displaying of the application's design and definite demonstrating, which includes making an interpretation of the models into programming code. Class charts can likewise be utilized to show information.

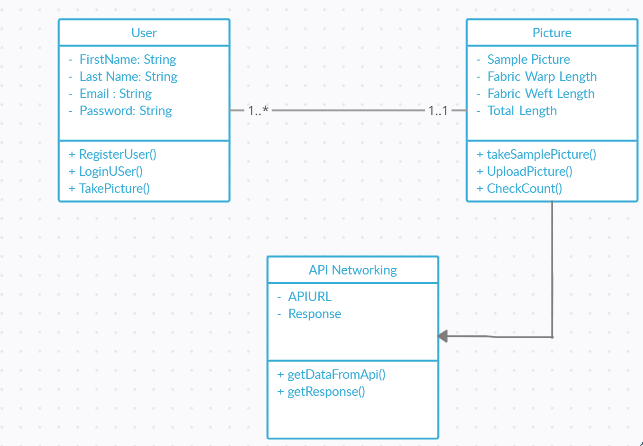


Figure 4.1 Overall system Class Diagram

# Use Case Diagram:

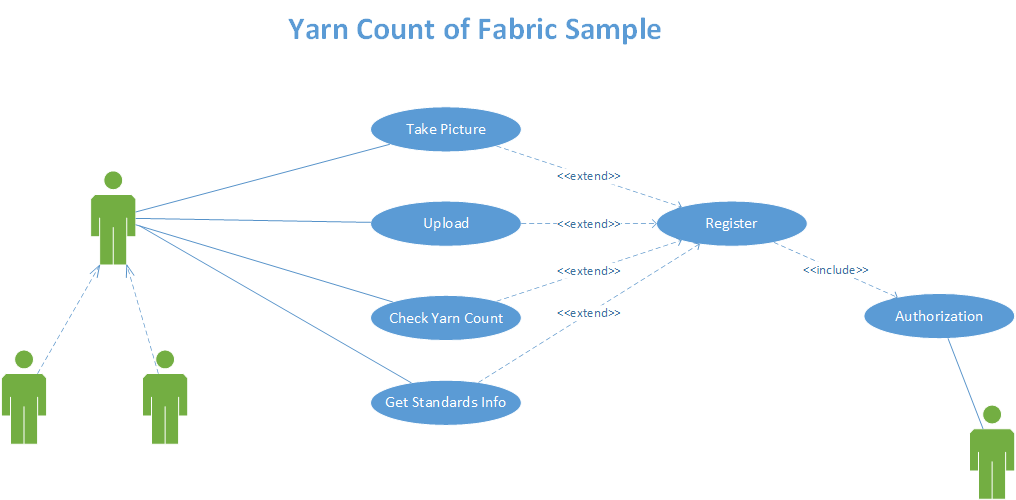


Figure 4.2 Overall system Use case Diagram

# User Authentication:

We also need to describe our system with other use cases as authentication:

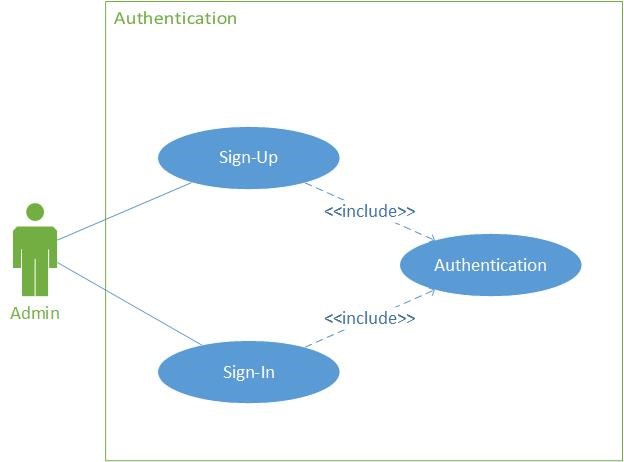


Figure 4.4 Authentication of the user

# Use Case Tables:

These tables address how a solitary action of the framework works and their connected ways and further where movement will bring you.

# Sign Up:

Table 4.1 Sign-up Use case

|  |  |
| --- | --- |
| Use Case Name | Sign Up |
| Priority | High |
| Use Case Id | 1 |
| Actor | Admin |
| Precondition | None |
| Basic Path | * User opens the app. * Click Sign up button. * Sign up page appears. * User fill information. * Click sign up button. |
| Description | User become the Member of our Application |
| Post Condition | User is asked to sign in |

# Sign In:

This scenario represents the authentication of the system

Table 4.2 Sign in Use Case

|  |  |
| --- | --- |
| Use Case Name | Sign In |
| Priority | Low |
| Use Case Id | 2 |
| Actor | Admin/User |
| Precondition | There is no precondition specifically, but user can register if he demands. |
| Basic Path | * User opens the app * Click sign in button * Sign in page appears * User fill information * Click sign in button to proceed |
| Description | User will then login to the app and can check the recent picture he has taken like the history of some information that will be taken from the database. |
| Post Condition | Users access the basic functionality of the system |

# Detection of Thread count:

Following is the use case table to show the scenario of detecting the thread per inch in woven fabrics:

Table 4.3 Thread Count Use-Case

|  |  |
| --- | --- |
| Use Case Name | Check yarn count |
| Priority | High |
| Use Case Id | 6 |
| Actor | User |
| Precondition | User must take picture of fabric sample or upload picture |
| Basic Path | * User will press the button to take the picture or to upload the picture. * Click on recognize the thread count Button. * Browse image or capture image * Start recognizing |
| Description | Fabric Sample will be extracted and will be checked for the thread count. |
| Post Condition | Checks status for registration and stolen car. |

# 

# Chapter 5: Architecture Design

# Architecture:

Basically, The Architecture of the software or the system in all the projects defines the workflow and frameworks of the project model. The architecture of the system is very useful for the programmer or the end-user to understand the model easily and efficiently and can use them easily.

# System Architecture:

System architectures allows the user to understand the workflow of our system that how user and the system will interact with each other or with the system. The end-user can easily use the system by just entering the login credentials as well as the admin can also use this system.

# Sequence Diagrams:

Sequence diagrams are one of the most important diagrams in the project as the name depicts that this diagram shows the overall sequence of the project or of some events. Diagram includes the messages that is communicated among the objects and the responds, every object has some lifetime and messages. Object activation lifetime reactivates when object is used.

# User Sign-In Sequence Diagram:

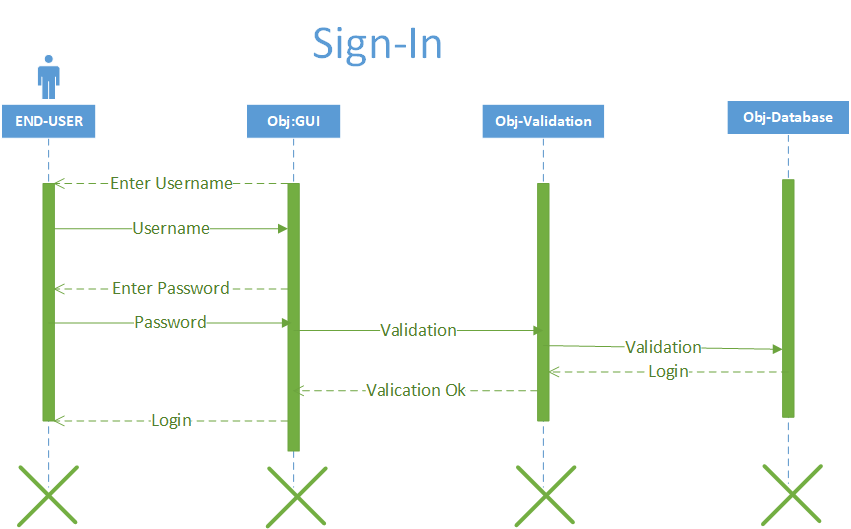


Figure 5.1 Sign in Sequence Diagram

# User Sign-Up Sequence Diagram:

The Sequence diagram of sign-up process is given below:

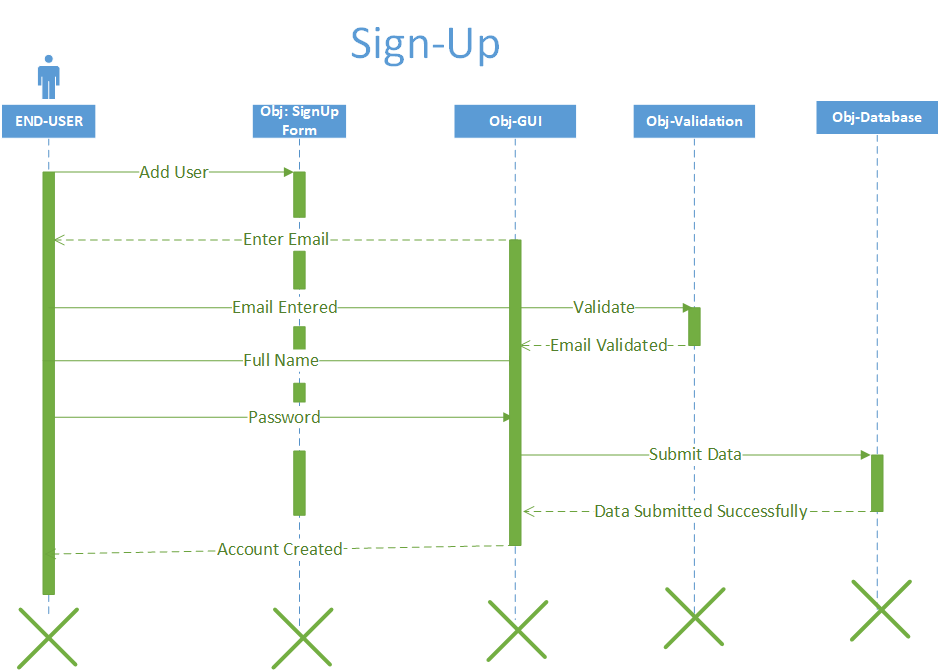


Figure 5.2 Sign Up Sequence Diagram

# Image Processing Sequence Diagram:

This Sequence diagram shows the process how image is being taken and processed.

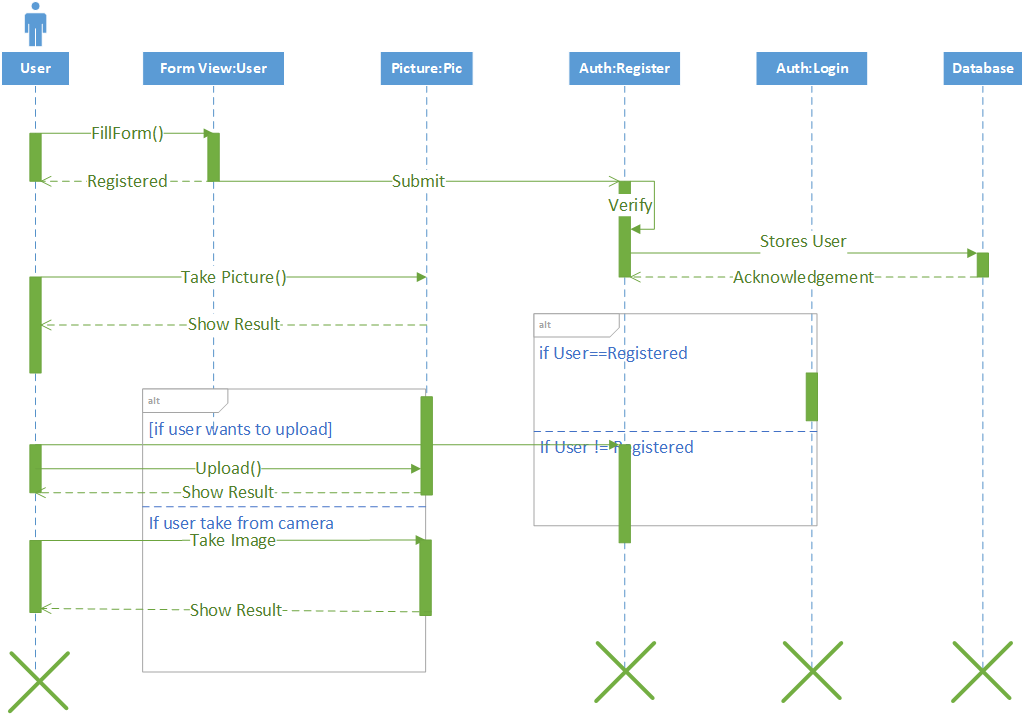


Figure 5.3 Image Process Sequence Diagram

# Activity Diagram:

Activity diagram is also the main part of architecture diagrams. They basically tell the activity of the events and how they take place and flow charts of the event occurs. In our system, we have main 2 actors that is the end user which is going to take the image sample from the camera or the gallery. And the camera itself is an actor through which the image is taken.

Below sections show the activity diagram of our system of both actors:

# User Sign up Activity Diagram:

Below diagram shows the user sign up diagram:

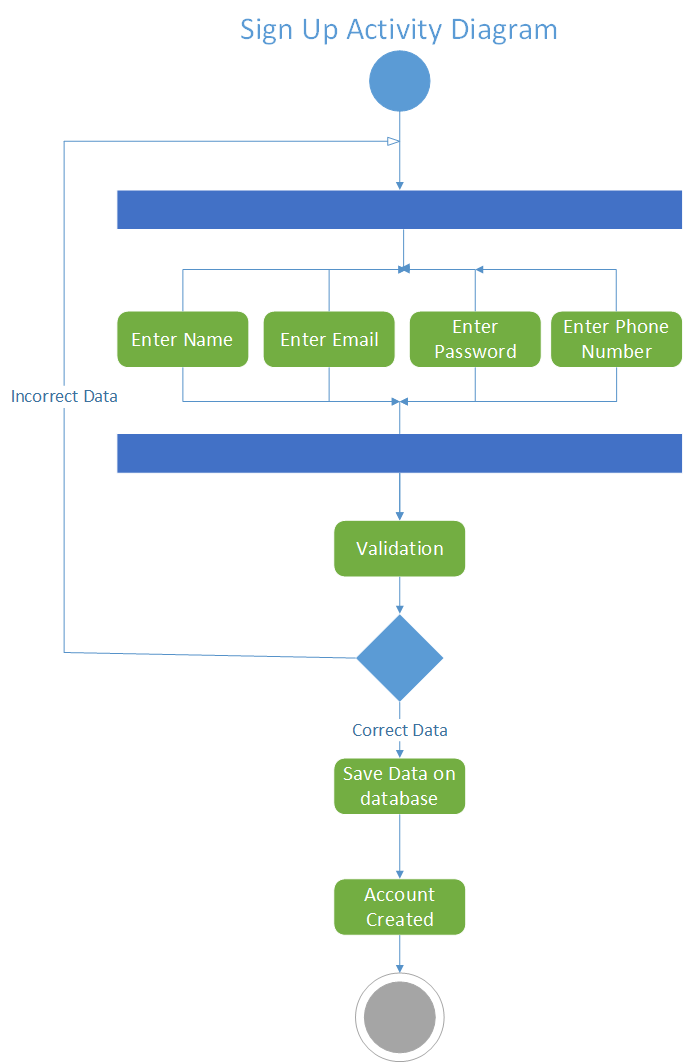


Figure 5.4 Sign Up Activity Diagram

# User Sign in Activity Diagram:

Here is the Activity diagram to show the activity flow of sign in process:

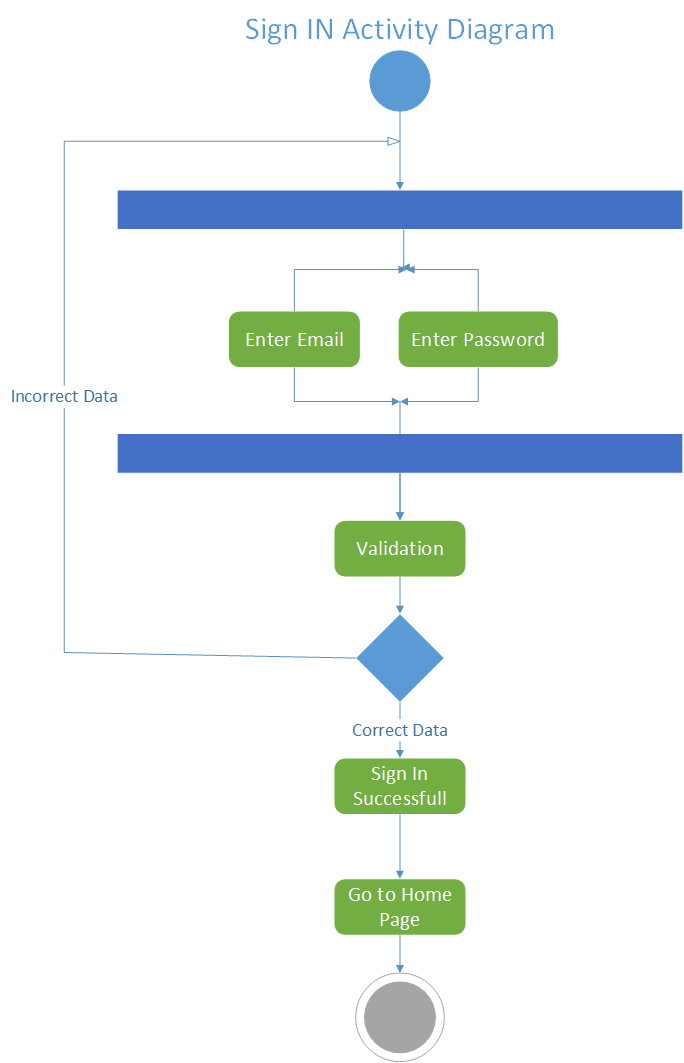


Figure 5.5 Sign in Activity diagram

# Image Processing Activity Diagram:

The diagram below shows the process how the image is processing and showing the result on abstract level:

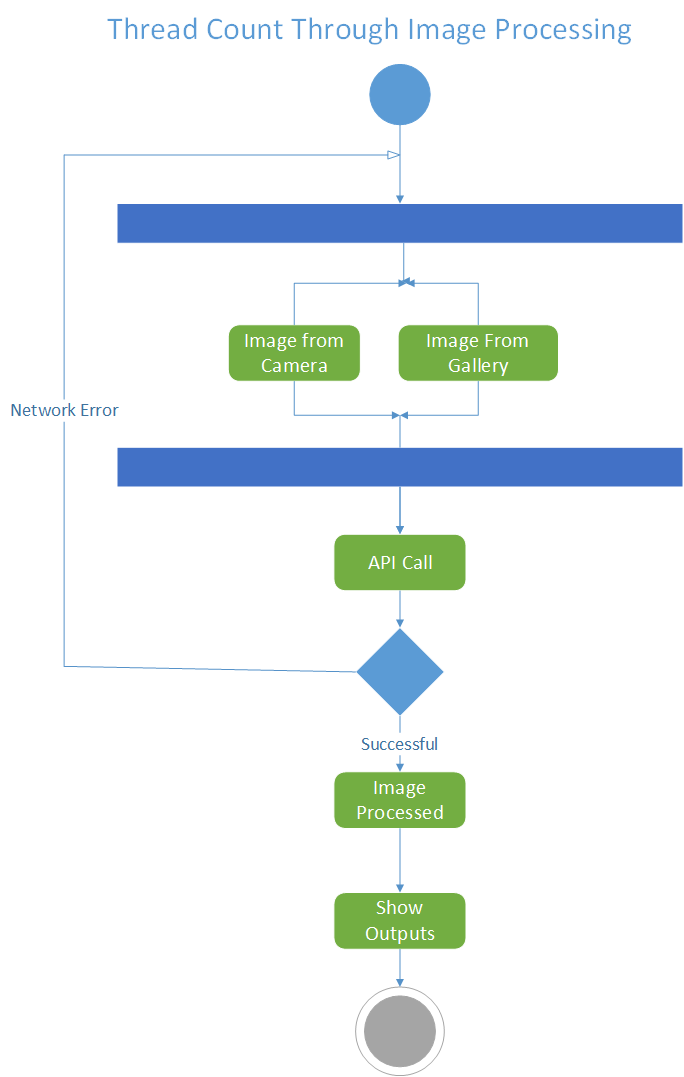


Figure 5.6 Image Processing Activity Diagram

# Chapter 7: User Interface

# 7.1 Interfaces:

Interfaces are the best visual representation that a user can easily understand how to use the system. The following section elaborates all the interfaces TPI System.

# 7.1.1 Splash Screen:

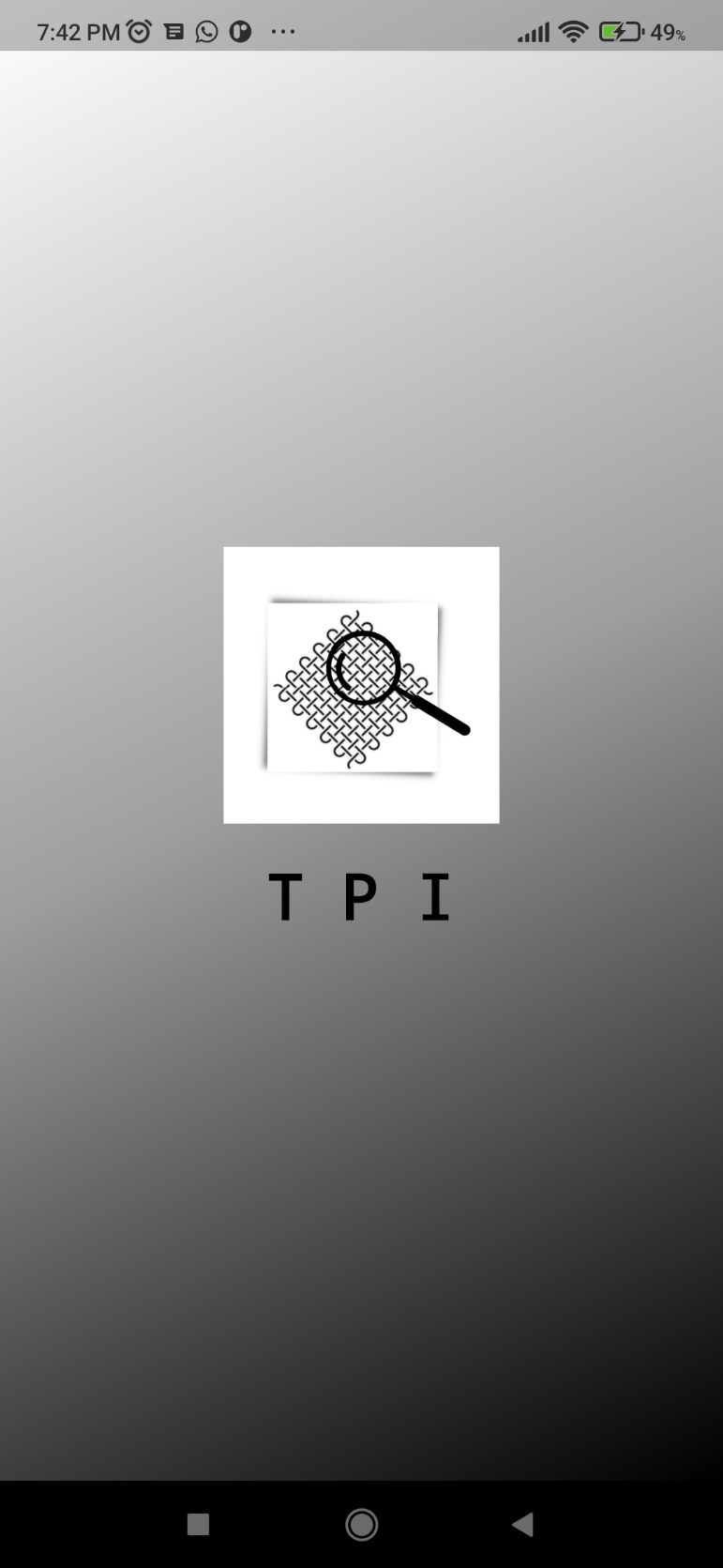


Figure 7.1 App Splash Screen

# 7.1.2 Welcome Screen:

This interface shows the welcome screen of the app.

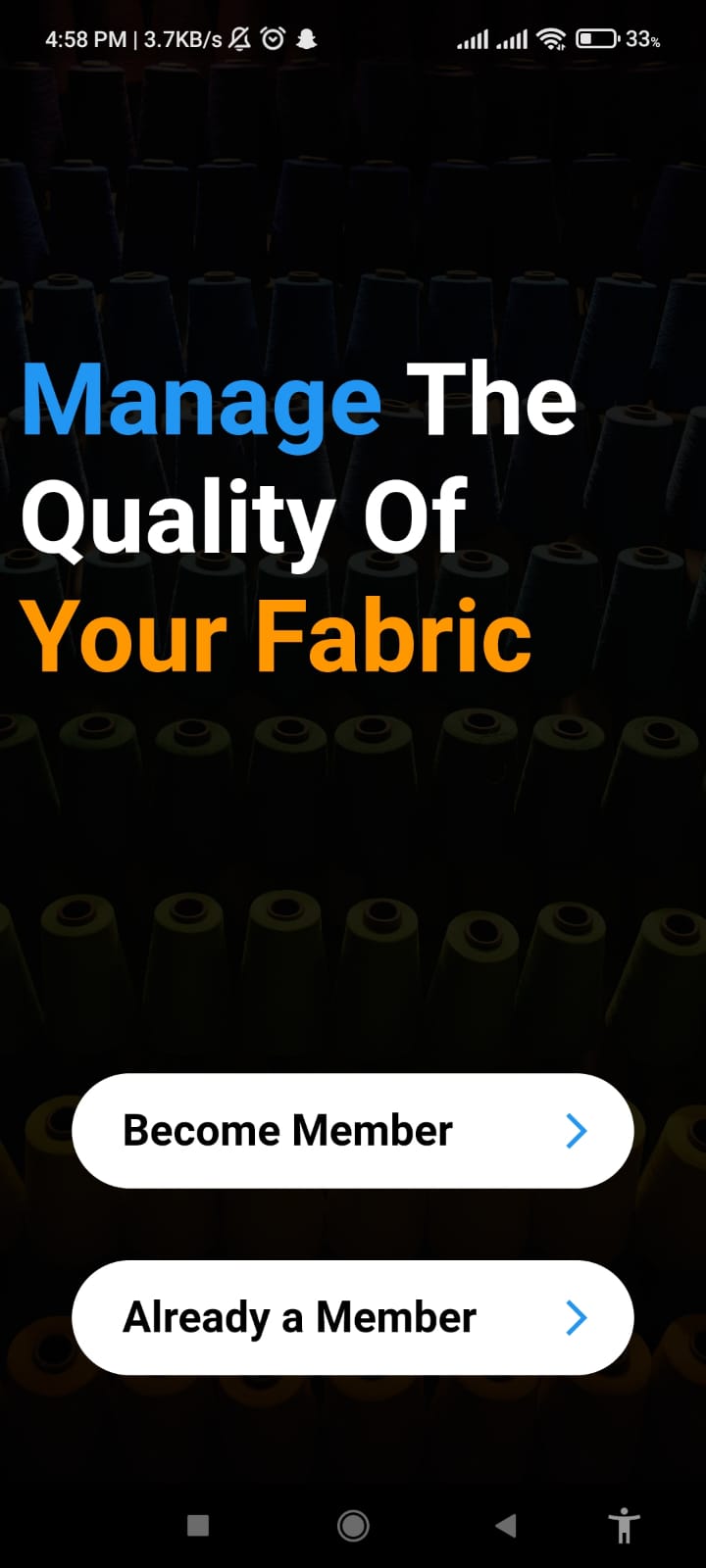


Figure 7.2 App Welcome Screen

# 7.1.3 Sign Up Interface:

Sign up interface is for the user. User can only create a single Account. Interface shown in figure 7.1

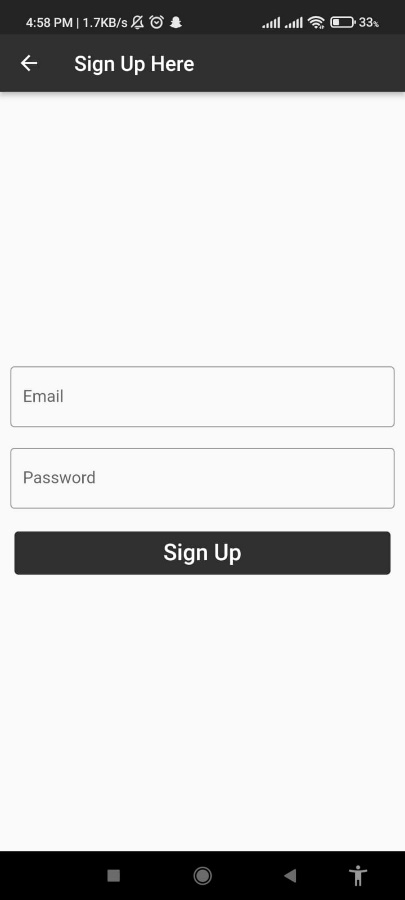


Figure 7.3 Sign up Screen

# 7.1.4 Sign In Interface:

Same Type of Interface for the user. User can Sign into the App and Take the functionality. This is shown in figure 7.2.

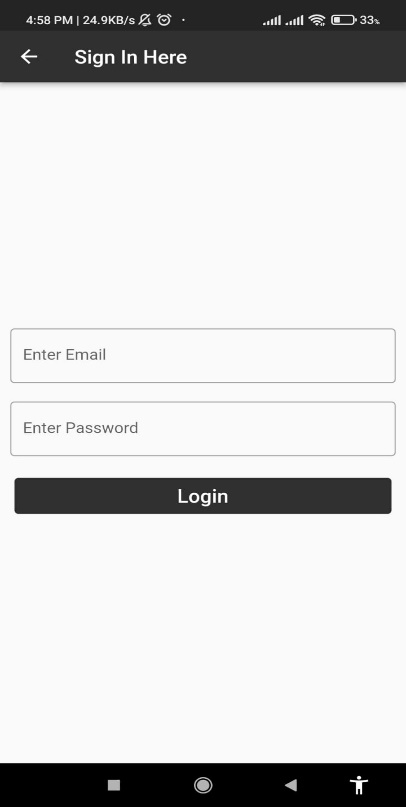


Figure 7.4 Sign in Screen

# 7.1.5 Take Image of Fabric Sample Interface:

A User can tap onto the button or take a picture or upload a picture from gallery then after taking the picture, it can be selected to send to server for further processing and this is shown in figure 7.3



Figure 7.5 Upload Image Screen

# 7.1.6 Calling API to Process Image Taken from Gallery or Camera:

When Image will be uploaded, we will tap on the “Count” Button that will start Calling the API to the backend server and will send the image to the backend server. Python and MATLAB will be working behind to process the image and will return the result onto the App.

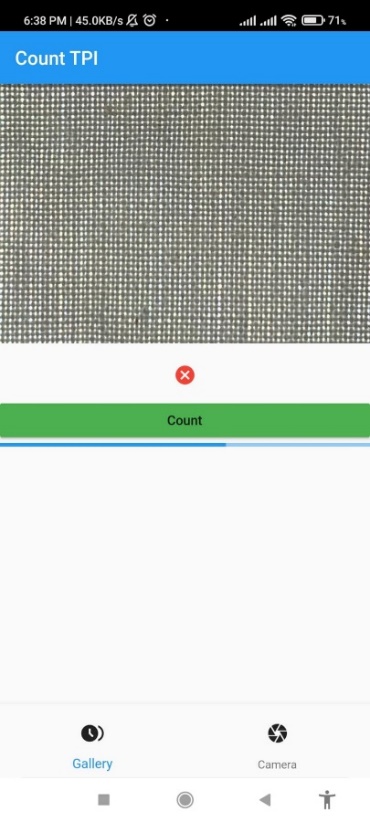


Figure 7.6 API Call

# 7.1.7 Result Interface:

The User can get the results on Result Interface and check the thread count of the processed image after getting response from the server. This Interface is showing in figure 7.5.

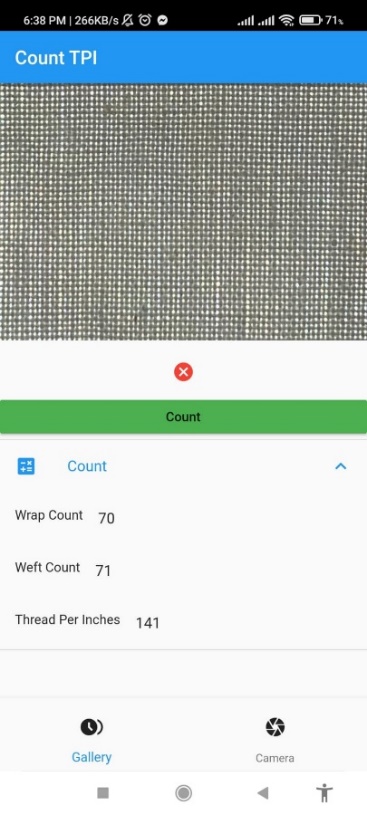


Figure 7.7 Result

# Chapter 8: Test Cases

# Black Box Testing:

Black box testing is a kind of testing procedure of the framework with no earlier information on the framework as a layman. An analyzer gives the information and notices the normal result created by the framework. This makes it conceivable to recognize how the framework answers expected and surprising client activities.

# White Box Testing:

Interestingly, with black box testing, white box testing alludes to the situation where the analyzer profoundly comprehends the entire usefulness of the framework, and framework parts are tried. Acquiring profound comprehension of the framework needs the analyzer to know about the program or code level.

# Test Cases:

Test cases are created to test the overall functionality of the system. These test cases are created for almost every feature of the system and the expected outputs of the system whether or not the system is giving proper expected output or not.

# Test Case 1: Sign Up:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Case Condition | Expected Result | Actual Result | Test Case Result |
| Sign\_Up\_1.1 | Fill out the Form | Registration Successful | Registration Successful | **PASS** |

# Test Case 2: Sign In:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Case Condition | Expected Result | Actual Result | Test Case Result |
| Sign\_In\_2.1 | Fill out the Form | Go to Home Page | Go to Home Page | **PASS** |

# Test Case3: Take Image from Camera:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Case Condition | Expected Result | Actual Result | Test Case Result |
| Image\_From\_Camera\_3.1 | Take Image from Camera | Image Uploaded | Image Uploaded | **PASS** |

# Test Case4: Take Image from Gallery:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Case Condition | Expected Result | Actual Result | Test Case Result |
| Image\_From\_Gallery\_4.1 | Take Image from Gallery | Image Uploaded | Image Uploaded | **PASS** |

# Test Case5: API\_Call:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Case Condition | Expected Result | Actual Result | Test Case Result |
| API \_Call \_5.1 | Calling API | Returns Thread count | Returned Thread Count | **PASS** |

# Chapter 9 References

**References:**

[1] Wikipedia. (August 2021). Available: <https://en.wikipedia.org/wiki/Warp_and_weft#/media/File:Warp_and_weft_2.jpg>

[2] MICROSCOPE-ANTIQUES.COM. (2016). *THREAD-COUNTING MICROSCOPE*. Available: <https://microscope-antiques.com/linen.html>

[3] Lunometer.

[4] P. C. Crews, S. Kachman, A. G. J. T. C. Beyer, and Colorist, "Influences on UVR transmission of undyed woven fabrics," vol. 31, no. 6, 1999.

[5] M. Sharifi, M. Fathy, and M. T. Mahmoudi, "A classified and comparative study of edge detection algorithms," in *Proceedings. International conference on information technology: Coding and computing*, 2002, pp. 117-120: IEEE.

[6] Y. Hu, Z. Long, and G. J. I. S. p. l. AlRegib, "A high-speed, real-time vision system for texture tracking and thread counting," vol. 25, no. 6, pp. 758-762, 2018.

[7] W. J. Book *et al.*, "Automated garment manufacturing system using novel sensing and actuation," 2010: Georgia Institute of Technology.

[8] R. Pan *et al.*, "Measuring thread densities of woven fabric using the Fourier transform," 2015.

[9] J. Escofet, M. S. Millán, and M. J. A. O. Ralló, "Modeling of woven fabric structures based on Fourier image analysis," vol. 40, no. 34, pp. 6170-6176, 2001.

[10] A. Moussa, D. Dupont, D. Steen, and X. J. T. J. o. T. T. I. Zeng, "Structure analysis and surface simulation of woven fabrics using fast Fourier transform techniques," vol. 101, no. 6, pp. 556-570, 2010.

[11] P. Potiyaraj, C. Subhakalin, B. Sawangharsub, W. J. I. J. o. C. S. Udomkichdecha, and Technology, "Recognition and re‐visualization of woven fabric structures," 2010.

[12] M. Ralló, J. Escofet, and M. S. J. A. o. Millán, "Weave-repeat identification by structural analysis of fabric images," vol. 42, no. 17, pp. 3361-3372, 2003.

[13] L. Rubine, C. Renie, S. Sowmiya, and R. H. Prasath, "A CASE STUDY ON FOURIER TRANSFORMATION AND ITS APPLICATIONS ON FABRIC PRODUCTION," 2019.

[14] Y. Xiaobo, X. Binjie, G. Baciu, H. J. R. J. o. T. Jinlian, and Apparel, "Fourier-analysis based satin fabric density and weaving pattern extraction," vol. 11, no. 1, p. 71, 2007.

[15] Y. Han, L. Luo, B. Xie, and C. J. T. R. J. Xu, "Nonparametric histogram segmentation-based automatic detection of yarns," vol. 90, no. 11-12, pp. 1326-1341, 2020.[[1]](#endnote-1)

[16] [Scrum Framework - Scrum (software development) - Wikipedia](https://en.wikipedia.org/wiki/Scrum_(software_development)#/media/File:Scrum_Framework.png)

1. [↑](#endnote-ref-1)