**SGRAPHICAL CHECKSUM VERIFIER APPLICATION**

**MUNALULA SIKAZWE**

**Reg No: 17221351006**

**Guide**

**Mr.Vernon Sivubwa**

Project Report

*Submitted*

*In partial fulfillment of the requirements for the Degree of*

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

**JULY 2020**



**DMI-ST.EUGENE UNIVERSITY**

**CHIBOMBO, ZAMBIA**

PROFORMA FOR APPROVAL OF PROJECT PROPOSAL

*APPENDIX II*

Proposed Project Team:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Reg. No. | Name of the students | Semester | Branch |
| 1 | 17221351006 | MUNALULA SIKAZWE | VII | SCIENCE COMPUTER SCIENCE |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title of the Project: | | | GRAPHICAL CHECKSUM VERIFIER APPLICATION | | |
|  | | | | | |
| Subject Area: | | DATA INTEGRITY IN CYBERSECURITY | | | |
| Name of the Guide: | | | | MR.VERNON SIVUBWA | |
| Designation : | | | | ASSISTANT LECTURER | |
| Address with Phone No. +91 9994061407 | | | | | |
| Office: | Department of Computer Science Engineering, DMI-St. Eugene University, Chibombo, Zambia. | | | | |
| Residence: | DMI-St. Eugene University, Chibombo, Zambia. | | | | |
| No. of projects & students currently working under the Guide: | | | | | 25 |

**Signature of the Student Signature of the Guide**

**Date......................... with seal**

N.B.: Please do not forget to enclose the synopsis of the project and the Bio-data of the Guide. In case the complete and signed Bio-Data of the Guide is not enclosed, the proposal will not be entertained**.**

**For Office Use only:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SYNOPSIS** | APPROVED |  | NOT APPROVED |  |
| **GUIDE** | APPROVED |  | NOT APPROVED |  |

**Comments / Suggestions for reformulation of the Project.**

**Date...................... SIGNATURE OF THE HOD**

# **BIO-DATA OF THE PROPOSED GUIDE FOR PROJECT WORK**

1. **PERSONAL INFORMATION**

NAME (in block letters) : VERNON SIVUBWA

Date of Birth & Age : 26 February 1966, Age (53)

Sex : MALE

Academic Qualification : MSc.BA.Ed

Official Address : Department of CSE,

DMI-St. Eugene University,

Chibombo, Zambia.

Phone No. and Fax. : +260974636405

Residential Address : Department of CSE,

DMI-St. Eugene University

Chibombo, Zambia

Phone No., and e-mail id : vsivubwa@gmail.com

**2. DETAILS OF EMPLOYMENT**

Designation : Assistant Lecturer

Field of Specialization : Computer Science

Teaching Experience (in years) : 29 YEARS

Industrial Experience (in years) : NONE

Particulars of contribution / experience in the field of specialization: MANAGEMENT

IT PROGRAMMES

No. of Projects guided : 23

I **VERNON SIVUBWA** do hereby accept to guide **MUNALULA SIKAZWE** the student of the **Bachelor of Science in Computer Science** programme of the **DMI-St. Eugene University, Chibombo**.

**Signature of the Student Signature of the Guide with Seal**

CERTIFICATE OF THE GUIDE

*APPENDIX IV*

This is to certify that the project work entitled GRAPHICAL CHECKSUM VERIFIER APPLICATION is a bonafide work of Mr. MUNALULA SIKAZWE, of Registration No 17221351006in partial fulfillment for the award of the Bachelor of Engineering in Computer Science of **DMI-ST.EUGENE UNIVERSITY** under my guidance. This Thesis work is original one and not submitted earlier for the award of any degree / diploma elsewhere.

**Signature of Student:**

**Signature of the Guide:**

DECLARATION BY THE CANDIDATE

*APPENDIX V*

I **MUNALULA SIKAZWE**hereby declare that this project report **GRAPHICAL CHECKSUM VERIFIER APPLICATION** submitted to **DMI-ST.EUGENE UNIVERSITY** in the partial fulfillment of requirements for the award of the Bachelor of Engineering in Computer Science is a record of the original work done by me under the supervision of **MR.VERNON SIVUMBWA**

**Enrolment No. :**

**Register No. : 17221351006**

**Date :**

**Signature :**

|  |
| --- |
| LOGO_New Zambia  *APPENDIX VI*  DMI-ST.EUGENE UNIVERSITY  CHIBOMBO - ZAMBIA  BONAFIDE CERTIFICATE  Register No: **17221351006**  Certified that this is bonafide record of work done in (Project)……………………………………………………  ………………………………. by **Mr. MUNALULA SIKAZWE** of Degree course in **Bachelor of Science In Computer Science** in **DMI-St. Eugene University, Zambia.** During the academic year **2017**  INTERNAL EXAMINER EXTERNAL EXAMINER |

*APPENDIX VII*

# **ACKNOWLEDGEMENT**

Sincere gratitude to the **Almighty God** for the blessings showered on me at every stage of this undertaking of completing my project without any obstacles.

I stand indebted in gratitude to our beloved Founder/ Chancellor **Rev. Fr. J.E. ARUL RAJ.OMI** for all the facilities provided at our institution – DMI St. Eugene University, Chibombo, Zambia. I would like to thank the President of our University Council **Dr. T.X.A. ANANTH,** DMI group of Institutions, Zambia, Director of Education **Dr. IGNATIUS A. HERMAN**, DMI group of Institutions, Zambia; for according me an opportunity to do this project successfully.

I further proudly express my esteemed gratitude to **Dr. H. SYED OTHUMAN,** Deputy Vice Chancellor (Admin), **Dr. R. KAVITHA,** Assistant Registrar (Academic), **Dr.A. KALIRAJAN**, Principal(Academic) and **Rev. Fr. M. AROCKIA KUMAR, MMI** Vice Principal (Administration).

Finally, I thank Mr**. PATRICK MWENYA,** HOD, Department of Computer Science and Engineering, internal guide **VERNON SIVUBWA** andstaff members of the Department for their valuable support to finish my project.

**Signature of the Student Signature of the Guide with Seal**

ABSTRACT

*APPENDIX VIII*

This paper seeks to present a data integrity checking system designed based on MD5algorithm to ensure the integrity of files sent and received via transmission lines between departments and the Exams and record unit of the Federal Polytechnic, Bauchi. The data for the system were collected using observation, interviews and review of existing documentation. The Structured Systems Analysis and Design (SSADM) and Knowledge Engineering Methodologies were used in the problem analysis and design of the system. The program for the system was written using Microsoft Visual Basic 6.0 as part of Microsoft Studio. The system implementation shows that documents management greatly improved as document within the units were all encrypted making it impossible for falsification, results manipulations, theft and misuse.

TABLE OF CONTENTS

[PROFORMA FOR APPROVAL OF PROJECT PROPOSAL i](#_Toc469832957)

[BIO-DATA OF THE PROPOSED GUIDE FOR PROJECT WORK ii](#_Toc469832958)

[CERTIFICATE OF THE GUIDE iii](#_Toc469832959)

BONA-FIDE CERTIFICATE…………………………………………………………………………………....v

[DECLARATION BY THE CANDIDATE iv](#_Toc469832960)

[ACKNOWLEDGEMENT vi](#_Toc469832962)

[ABSTRACT vii](#_Toc469832963)

TABLE OF CONTENTS……………………………………………………………………………..….......viii

LIST OF FIGURES………………………………………………………………………………................x

LIST OF TABLES………………………………………………………………………………….………xi

[CHAPTER I 1](#_Toc469832966)

[INTRODUCTION 1](#_Toc469832967)

[1.1 About the Project 1](#_Toc469832968)

[CHAPTER II 3](#_Toc469832970)

[SYSTEM STUDY 3](#_Toc469832971)

2.1 Introduction……………………………………………………………………………………3

[2.2 Problem definition……………………………………………………………....…………….3](#_Toc469832972)

2.3 Existing System……………………………………………………………………………….4

2.4 Proposed System……………………………………………………………..….…………….4

2.5 System Objectives…………………………………………………………………………….4

[2.6 System Specification………………………………………………………………………….5](#_Toc469832973)

2.6.1 Software Specification...........................................................................................................5

[2.6.2 Hardware Specification……………………………………………………………….…….5](#_Toc469832974)

[CHAPTER III 6](#_Toc469832975)

[SYSTEM DESIGN 6](#_Toc469832976)

[3.1 Introduction 6](#_Toc469832977)

[3.2 System Architecture 6](#_Toc469832979)

3.3 Use Case Diagram…………………………………………………………………………….7

3.4 Dataflow Diagram……………………………………………………………………………..7

3.5 Table Design…………………………………………………………………………………..8

[3.6 Input Design 10](#_Toc469832980)

[3.7 Output Design 11](#_Toc469832981)

[CHAPTER IV 15](#_Toc469832982)

[SYSTEM DEVELOPMENT ...15](#_Toc469832983)

[4.1 Introduction 15](#_Toc469832984)

4.2 Module Description……………………………………………………………………….…15

4.3 Project Description………………………………………………………….……………….16

[CHAPTER V 67](#_Toc469832985)

SYSTEM TESTING…………………………………………………………..…………………67

[5.1 Introduction 67](#_Toc469832986)

5.2 System Testing…………………………………………………………….…………………67

5.2.1 Unit Testing…………………………………………………………………..……………67

[5.2.3 Verification Testing……………………………………………………………...………...69](#_Toc469832988)

[5.2.4 User acceptance testing 69](#_Toc469832989)

[CHAPTER VI 70](#_Toc469832990)

[SYSTEM IMPLEMENTATION 70](#_Toc469832991)

6.1 User Training and Documentation……………………………………….…………………..70

CHAPTER VII………………………………………………………………………………...…71

PROBLEMS FACED AND SOLUTIONS……………………………………………...………71 CHAPTER VIII…………………………………………………..…………………...…..……..72

CONCLUSIONS AND SUGGESTIONS……………………………...………………………..72

CHAPTER IX………………..…………………………………………………….………….…73

SUGGESTION FOR PROJECT EXTENSION……………..…………………………………73

[BIBLIOGRAPHY 74](#_Toc469832997)

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TABLE NAME** | **PAGE NO** |
| 1 | Table for the Input Design for Enter user | 10 |
| 2 | Table for Input Design for Enter recipient name | 11 |
| 3 | Table for input Design for make call | 11 |
| 4 | Table for incoming call | 12 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **FIGURE NAME** | **PAGE NO** |
| 1 | Figure 1.System Architecture | 6 |
| 2 | Figure 2.Use case diagram | 7 |
| 3 | Figure 3.Data flow diagram | 8 |
| 5 | Figure 4:PREVIEW OF OUTPUT DESIGN FOR USER LOGIN | 9 |
| 6 | Figure 5:Preview for table design for enter recipient name | 9 |
| 7 | Figure 6:Preview for table design for make call and cancel call | 10 |
| 8 | Figure 7.Preview of Output Design for incoming call | 12 |

CHAPTER I

## INTRODUCTION

1.1 About the Project

Every institution whether private or public will strive to protect the integrity of the information it generate, safeguard its transmission over a network as well as ensure it proper accessibility when the need arises. Safeguarding information is vital to the success of any business as both origin, use and store of the information is vital to its reliability. To ensure easy of data integrity checking without spending much, a cryptographic has function is therefore developed for use.Cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the "message," and the hash value is sometimes called the message digests or simply digests

This Message Digest 5 (MD5) algorithm is a widely used cryptographic hash function that produces a 128-bit (16-byte) hash value and has been employed in a wide variety of security applications including checking data integrity. MD5 is one in a series of message digest algorithms designed by Professor Ronald Rivest of MIT [8]. When analytic work indicated that MD5's predecessor MD4 was likely to be insecure, MD5 was designed in 1991 to be a secure replacement.The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem. The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 Algorithm does not require any large substitution. It has since been found that MD5 is not collision resistant [3][4][5][6][7][8] and therefore not suitable for applications requiring SSL certificates, digital signatures, encryption of large and extremely sensitive document files. For extremely sensitive document, SHA-2 family of hash function may be recommended. The researchers used MD5 for the implementation of the security of documents of the exams and records units for the following reasons:

[1] The system is meant for internal use by staff within the polytechnic for the encryption of students records and students’ information related. It is not meant for securing financial transactions and accounting functions of the polytechnic.

[2] The unique benefit of information technology is the ability to apply it to providing solutions to internal problem and the ability to use local resources e.g. Manpower, limited funds etc to bring about positive change in a system. To this regard, this system has been able to introduce such changes by securing its own records and using its own personnel to design and implement the security system.

[3] The system so design is upgradable to higher platform as the module that implement the MD5 function can easily be replaced by any other function e.g. SHA-2. This is the advantage of modular programming utilized by the researcher.

Design.

### Functions of this system:

1. This system allows a users to input a hash as a string from a website

2. It provides the function of generating a hash from a file from the system

3. It provides the function of verifying the validity of any file

CHAPTER II

# SYSTEM STUDY

**2.1 Introduction**

1. This system shows the user login name

2. It provides the function of entering the recipient’s login credentials

3. It provides the function of making a call

4. There is database connectivity provided when you login in as someone.

5. The application has a secure authentication process video calling services.

### 2.2 **Problem Definition**

In the modern days of information overload the explosion of information has caused the creation of threats to the integrity of data in a way that created a major stumbling block to information dissemination and utilization . The current solution to this problem has a few faults which will be further clarified in the description of the Existing System.

**2.3 Existing System**

The Existing system is a commandline based tool that does not create a good provisions for usability to the ordinary user .

**2.4 Proposed System**

The Graphical Checksum Verifier provides the user with an intuitive and usable way of verifying the integrity of files that does not require specialized skills and expertise to implement and test data integrity effectively

**2.5 System Objective**

To provide an interface for proper file integrity verification and ensure security against threats to an organisation or individuals personal and private information.

1. To study the functions of Verification algorithms and their effect on file systems.

2. To explore the challenges being faced by the commandline system.

3. To make a software fast in processing, with good user interface.

4. To ensure accurate security and integrity of information .

5. For ensure that security of file redistribution is ensured .

## 

**2.6.1 Software Specification**

This are requirements specification for a [software system,](http://en.wikipedia.org/wiki/Software_system) is a description of the behavior of a system to be developed and may include a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe interactions the users will have with the software.

The software components used for this project are listed below:

1. Dual-core 64-bit processor.
2. 8 GB of memory.
3. Up to 24 GB of internal storage (Kong Visualizer: 4GB, Android SDK: 2GB, Windows SDK: 4GB, BlackBerry NDK: 4GB, plus ample space for multiple complex projects)
4. Network interface card.
5. Windows 10, Windows 8.1 Update, Windows 8, and Windows 7.

### **2.6.2 Hardware Specification**

For a system to be used efficiently and accurately, all computer softwar[e](http://en.wikipedia.org/wiki/Computer_software) needs certain [hardware](http://en.wikipedia.org/wiki/Computer_hardware) components or other software resources to be present on a [computer.](http://en.wikipedia.org/wiki/Computer) These prerequisites are known as (computer hardware specification) and are often used as a guideline as opposed to an absolute rule.

The hardware required includes the following:

1. Processor 2.4 GHZ processor speed
2. Disk space 80 GB (including 20 GB for database Management system)
3. SVGA color monitor or higher quality.
4. RAM 1GB and above.
5. Backup storage hard disk of about 80MB.
6. Computer with either Intel Pentium processor or AMD processor.

### **CHAPTER III**

### **SYSTEM DESIGN**

**3.1 Introduction**.

###### System Design is one of the tasking sections of the Programming. In this section of the project many previews are going to be seen and we are gradually getting close to the new system. System design is a transition from a user-oriented document to a document oriented to programmers or database personnel. The system design is structured into the following parts.

* System Architecture
* User Case Diagram
* Data Flow Diagram
* Table Design Input Design Output Design.

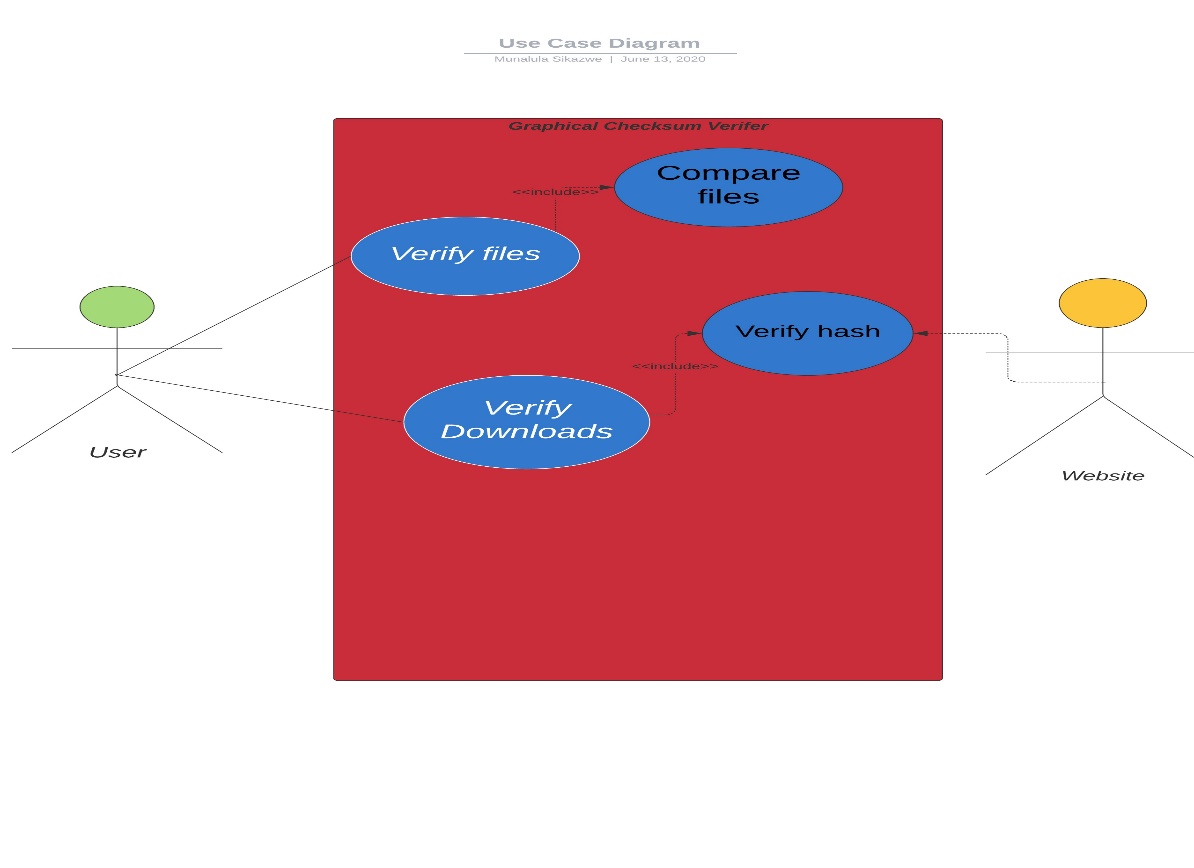
3.2 System Architecture

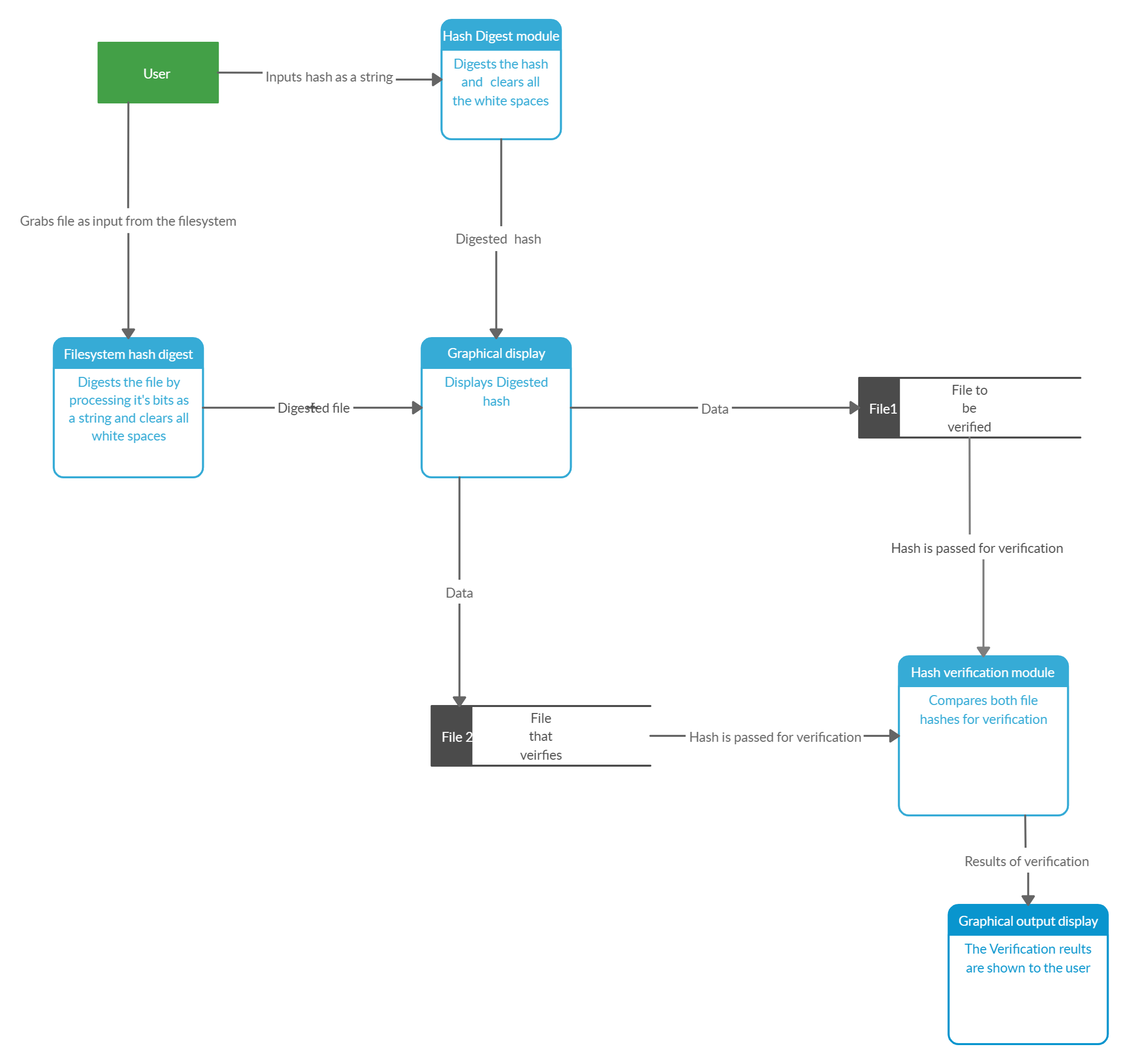
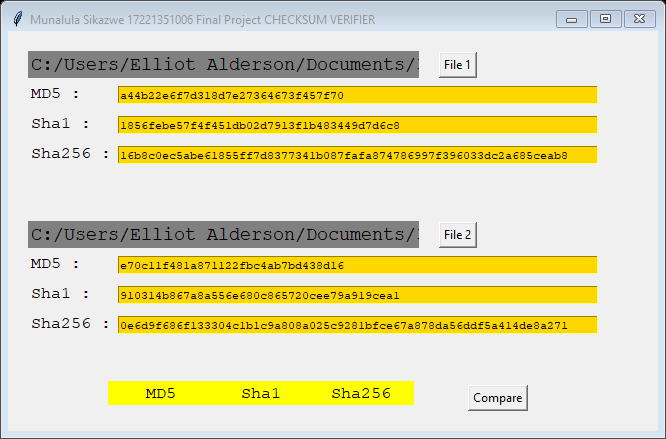
Main Module

Hash Verifier

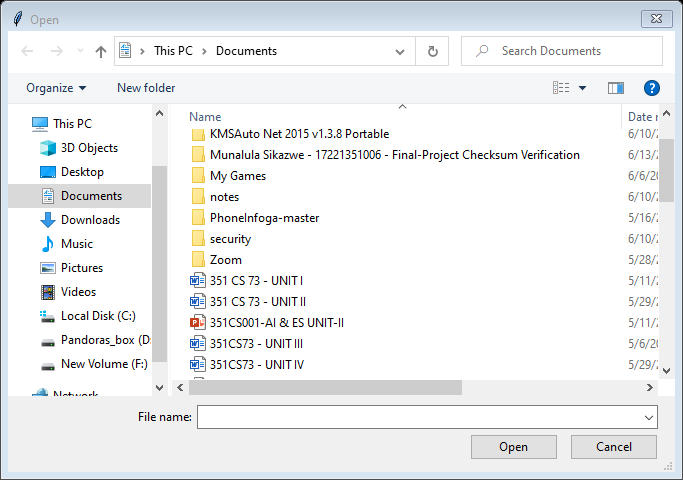
Hash Generator

Figure 1. 3.3 Use Case Diagram



3.4 Data Flow Diagram Input Design

3.5 Input Design



Output design

Figure 5: PREVIEW OF VERIFIED FILE

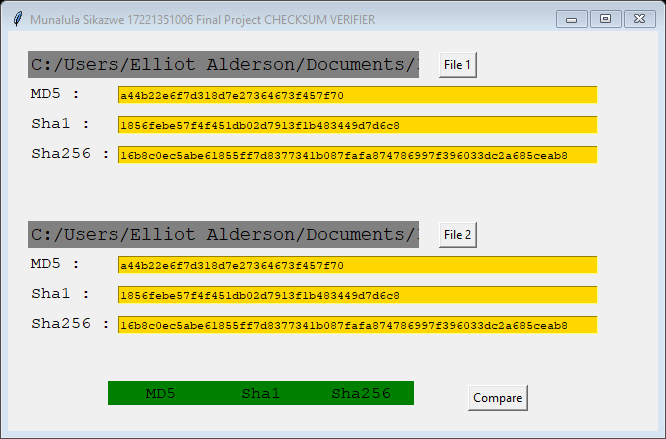


Figure 6: PREVIEW OF FAILED VERIFICATION

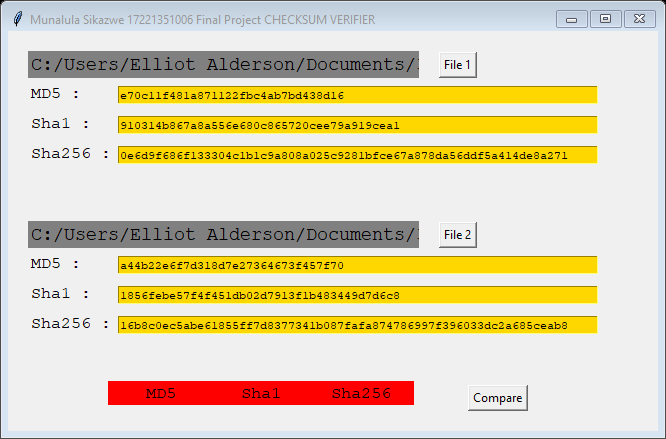


Figure 7:

**CHAPTER IV**

### **SYSTEM DEVELOPMENT**

4.1 Introduction

The system is an Android Mobile Application. The system will provide the following

Main features

1. It provides the accurate Connection between the caller and the receiver
2. Give the user a minimum of one caller to connect with per time
3. Provides efficient quality video for the user
4. Displays the user’s information in case they want to change from just being a user and also apply to be recceiver

**4.2 Module Description**

**File Hashing Module:**

The entire system is intended at checking/verifying the integrity of files that are transferred over the network from any of the polytechnic departments to the examinations and record unit. This module consists of set of codes that will accomplish the above stated functionality.

**Hash Verification Module**:

Verification means to check and confirm that the files that were sent are the actual files received without any change or modification within or along the network [2]. This is accomplished by hashing the copy of the received files and a comparison is made between the newly generated hash and the original hash which was generated by the sender before the files was sent over the network. If the two hashes are found to be the same, conclusion is reached that the integrity of the files has not been tempered with during transmission. The file hash verification module is the one that accomplish the above stated functionality

**4.3 Project Description**

The Graphical Checksum verifier will aim to allow give firm file integrity by ensuring the uniqueness of every file by ensuring a thorough computation of files of many formats and ensuring that these files can be verified any point at which a user chooses with accuracy , formal and secure mathematical algorithims such as the sha hashing algorithm

SOURCE CODE

*from* tkinter *import* \*  
*from* tkinter *import* filedialog  
*import* hashlib  
*import* sys  
  
  
root = Tk()  
  
f1 = StringVar()  
f2 = StringVar()  
Sha256f1 = StringVar()  
Sha256f2 = StringVar()  
MD5f1 = StringVar()  
MD5f2 = StringVar()  
Sha1f1 = StringVar()  
Sha1f2 = StringVar()  
  
  
*def* hash\_bytestr\_iter(*bytesiter*, *hasher*, *ashexstr*=*True*):  
 *for* block *in bytesiter*:  
 *hasher*.update(block)  
 *return* (*hasher*.hexdigest() *if ashexstr else hasher*.digest())  
  
*def* file\_as\_blockiter(*afile*, *blocksize*=65536):  
 *with afile*:  
 block = *afile*.read(*blocksize*)  
 *while* len(block) > 0:  
 *yield* block  
 block = *afile*.read(*blocksize*)  
  
*def* CmpHashes():  
  
 #Check MD5  
 *if* MD5f1.get() *and* MD5f2.get():  
 *if* MD5f1.get() == MD5f2.get():  
 print("MD5 match!")  
 MD5Lbl.config(bg='green')  
 root.update()  
 *else*:  
 print("MD5 does not match!")  
 MD5Lbl.config(bg='red')  
 root.update()  
 *else*:  
 MD5Lbl.config(bg='yellow')  
 print("MD5 data unavalible!")  
 root.update()  
  
 #Check Sha1  
 *if* Sha1f1.get() *and* Sha1f2.get():  
 *if* Sha1f1.get() == Sha1f2.get():  
 print("Sha1 match!")  
 Sha1Lbl.config(bg='green')  
 root.update()  
 *else*:  
 print("Sha1 does not match!")  
 Sha1Lbl.config(bg='red')  
 root.update()  
 *else*:  
 Sha1Lbl.config(bg='yellow')  
 print("Sha1 data unavalible!")  
 root.update()  
  
 #Check Sha256  
 *if* Sha256f1.get() *and* Sha256f2.get():  
 *if* Sha256f1.get() == Sha256f2.get():  
 print("Sha256 match!")  
 Sha256Lbl.config(bg='green')  
 root.update()  
 *else*:  
 print("Sha256 does not match!")  
 Sha256Lbl.config(bg='red')  
 root.update()  
 *else*:  
 Sha256Lbl.config(bg='yellow')  
 print("Sha256 data unavalible!")  
 root.update()   
  
*def* getFile(*fName*, *MD5f*, *SHA1*, *Sha256f*):  
   
 root.filename = filedialog.askopenfilename( )  
 *if* root.filename == "":  
 *return* 0  
 *fName*.set(root.filename)  
 *MD5f*.set(str(hash\_bytestr\_iter(file\_as\_blockiter(open(*fName*.get(), 'rb')), hashlib.md5())))  
 *SHA1*.set(str(hash\_bytestr\_iter(file\_as\_blockiter(open(*fName*.get(), 'rb')), hashlib.sha1())))  
 *Sha256f*.set(str(hash\_bytestr\_iter(file\_as\_blockiter(open(*fName*.get(), 'rb')), hashlib.sha256())))  
  
  
root.geometry('%dx%d+%d+%d' % (650, 400, 125, 125))  
root.title("Munalula Sikazwe 17221351006 Final Project CHECKSUM VERIFIER")  
  
#File one stuff  
  
f1Lbl = Label(root, textvariable=f1, bg="grey", width=35, anchor=W, font=("Courier New", 14)).place(x=20, y=20)  
f1.set("First file or enter a HASH below.")  
  
f1HashLbl1 = Label(root, text="MD5 : ", width=10, anchor=W, font=("Courier New", 12)).place(x=20, y=50)  
f1MD5 = Entry(root, textvariable=MD5f1, bg="gold", width=68, font=("Courier New", 8)).place(x=110, y=55)  
  
f1HashLbl2 = Label(root, text="Sha1 : ", width=10, anchor=W, font=("Courier New", 12)).place(x=20, y=80)  
f1Sha1 = Entry(root, textvariable=Sha1f1, bg="gold", width=68, font=("Courier New", 8)).place(x=110, y=85)  
  
f1HashLbl3 = Label(root, text="Sha256 : ", width=10, anchor=W, font=("Courier New", 12)).place(x=20, y=110)  
f1Sha256 = Entry(root, textvariable=Sha256f1, bg="gold", width=68, font=("Courier New", 8)).place(x=110, y=115)  
  
f1Btn = Button(root, text="File 1", command=*lambda*: getFile(f1, MD5f1, Sha1f1, Sha256f1))  
f1Btn.place(x=450, y=34, anchor="c")  
  
#File two stuff  
  
f2Lbl = Label(root, textvariable=f2, bg="grey", width=35, anchor=W, font=("Courier New", 14)).place(x=20, y=190)  
f2.set("Second file or enter a HASH below.")  
  
f2HashLbl1 = Label(root, text="MD5 : ", width=10, anchor=W, font=("Courier New", 12)).place(x=20, y=220)  
f2MD5 = Entry(root, textvariable=MD5f2, bg="gold", width=68, font=("Courier New", 8)).place(x=110, y=225)  
  
f2HashLbl2 = Label(root, text="Sha1 : ", width=10, anchor=W, font=("Courier New", 12)).place(x=20, y=250)  
f2Sha1 = Entry(root, textvariable=Sha1f2, bg="gold", width=68, font=("Courier New", 8)).place(x=110, y=255)  
  
f2HashLbl3 = Label(root, text="Sha256 : ", width=10, anchor=W, font=("Courier New", 12)).place(x=20, y=280)  
f2Sha256 = Entry(root, textvariable=Sha256f2, bg="gold", width=68, font=("Courier New", 8)).place(x=110, y=285)  
  
f2Btn = Button(root, text="File 2", command=*lambda*: getFile(f2, MD5f2, Sha1f2, Sha256f2))  
f2Btn.place(x=450, y=204, anchor="c")  
  
  
MD5Lbl = Label(root, text="MD5", width=10, bg="yellow", font=("Courier New", 12))  
MD5Lbl.place(x=100, y=350)  
  
Sha1Lbl = Label(root, text="Sha1", width=10, bg="yellow", font=("Courier New", 12))  
Sha1Lbl.place(x=200, y=350)  
Sha256Lbl = Label(root, text="Sha256", width=10, bg="yellow", font=("Courier New", 12))  
Sha256Lbl.place(x=300, y=350)  
CmpBtn = Button(root, text="Compare", anchor="c", command=*lambda*: CmpHashes())#MD5Lbl, Sha1Lbl, Sha256Lbl))  
CmpBtn.place(x=460, y=354)  
  
root.mainloop()

**CHAPTER V**

### **SYSTEM TESTING**

5.1Introduction

This is a segment in the report that looks at the way the system was tested.

5.2 System Testing

1. The entire system
2. Carried out by myself
3. Goal: Determine if the system meets the requirements
4. Functional Testing: Test of functional requirements
5. Performance Testing: Test of non-functional requirements

5.2.1 Unit Testing

What can be tested in units?

1. A functional requirement
2. Given input that satisfies the precondition, whether the output satisfies the postcondition
3. A unit can be a member function, a class, a package or component or a subsystem … Automation is the key! Replace user interaction with the scripts, if possible; replace some units with stubs
4. A unit tested can still have bugs, but most trivial bugs should have been found

What cannot?

1. Generally, test cannot replace the verification or code review
2. Specifically for unit test, interactions between this unit and other units after integration, system and user acceptance are not possible when the system is not ready yet

Table .1 UNIT TESTING TABLE

|  |  |
| --- | --- |
| **JUNIT TEST** | **COMPONENTS** |
| FIXTURE | * Input constraints * Output results * Data integrity * Storage and security |
| TEST CASE | Security   1. Verification 2. Hash check |
| SETUP | System tester. |
| TEARDOWN | * Random inputs * Process overloads |

5.2.2 Integration Testing

The entire system is viewed as a collection of subsystems (sets of classes) determined during the system and object design.

1. Assumption: System Decomposition is hierarchical
2. The order in which the subsystems are selected for testing and integration determines the testing strategy
3. Big bang integration (Non-incremental)
4. Bottom up integration
5. Top down integration
6. Sandwich testing

5.2.3 Verification Testing

In this survey, we look at verification, validation, and testing techniques as they are applied throughout the software development life cycle. All too often, testing is the only verification technique used to determine the adequacy of the software. When verification is constrained to a single technique and confined to the latter stages of development, severe consequences can result, since the later in the life cycle that an error is found, the higher is the cost of its correction.

5.2.4 User acceptance testing

1. UAT requires ‘formal testing’, which means that tests should be designed and executed in a structured way that provides objective evidence of the acceptability or otherwise of the system.
2. The definition speaks of testing with respect to ‘user needs, requirements, and business processes’. It does not mention any particular specification document but it does draw attention to what users need and it goes beyond testing software to include business processes.
3. The definition speaks of satisfying ‘acceptance criteria’, which define what is acceptable to the users.

**CHAPTER VI**

**SYSTEM IMPLEMENTATION**

6.1 User training and documentation

Two main areas of documentation were used:

###### **Process documents**

To guide the development testing, maintenance and improvement of the system, for the system administrator and IT engineers of the organization to help in improving the system.

###### **User documents**

These are to give the users the information they need to use the product, these are in form of teaching materials which include some technical explanation.

###### **System evaluation**

This System is a high standard program that can weather the storm of technology advancement, it is most needed in all pharmacies on rural areas and it is an antidote for poor business speed and transaction with record keeping and maintenance, it will be very helpful to clients and customers in the medical business. All it needs is a computer literate operative to make it work.

**CHAPTERVII**

### PROBLEMS FACED AND SOLUTIONS

###### **Problems Faced**

1. Doing video chatting on other application such as face book, WhatsApp is very expensive in terms of the bundle rate.
2. The user has to create and account such as face book accounts in order to do a video chat.
3. These other application require a lot of information about the person who is making the call and the person who is receiving the particular call.

###### **Solutions**

1. This application displays the details the user and the recipient
2. It provide the accurate Network service to do video chat
3. Give the user a minimum of one user per call
4. Provides good quality for video calling screen display
5. Displays the user name and the recipients details in the application

**CHAPTER VIII**

### **CONCLUSIONS AND SUGGESTIONS**

###### **Conclusion**

Effective implementation of this software will take care of the basic requirements of the mobile application because it is capable of providing easy and effective storage of information related to activities happening in the stipulated area. With these, the objectives of the system design will be achieved.

In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the pharmacy management, whenever the idea arises.

Initially mobile phones were developed only for voice communication but now days the scenario has changed, voice communication is just one aspect of a mobile phone. . There are other aspects which are major focus of interest. Two such major factors are web browser and GPS services. Both of these functionalities are already implemented but are only in the hands of manufacturers not in the hands of users because of proprietary issues, the system does not allow the user to access the mobile hardware directly.

###### **Suggestions**

.Designing this application is not an easy task. It all started from the requirement gathering and passes through so many other stages before completion. Based on the benefits of this system and tremendous value it will add to customer-user satisfaction, the below recommendation will be considered; It is recommended that the new system should be used with the necessary specifications of the system requirements and provision for an uninterrupted power supply should be made available throughout the hours of operation of the pharmacy to avoid power outage. There should also be basic computer knowledge for the users of the software. It is recommended that the software be improved especially in areas of accounting as it will be of great impact to the development of retail pharmacy.

### **CHAPTER IX**

### **SUGGESTIONS FOR PROJECT EXTENSION**

I will recommend that if there is going to be any modification the new writer should endeavor to improve on the limitations such as inclusion of the valid user authentication to further increase the system architecture and to satisfy users need more for writing of the source code. There are some limitations during the development of this mobile application that will require improvement as stated in previous chapter writer should put them in mind and face it as a challenge and not a problem.

In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the mobile application, whenever the idea arises. Also to include GPRS for accurate location.

**BIBLIOGRAPHY**

1. The Android Video calling System by Technopedia P. Priya1, V. Saranya2, S. Shabana3, Kavitha Subramani4 Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India1, 2, 3, 4
2. MBB: SMS Android Application Narendra Gupta1, Ramakant Gawande2 and Nikhil thengadi3 1, 2, 3 Final Year, CSE Dept., JDIET, Yavatmal, India.
3. AN ANDROID APPLICATION by Sultan Turhan.
4. Arif. M. Sreevas. S. Nafseer. K. and Rahul. R. (2012), 'Automated online video Chating database', India Conference (INDICON), Annual IEEE, Print ISBN: 978-1-4673-2270-6, pp. 012 - 017. [4] Spyropoulos. B., Botsivaly. M., Tzavaras. A., and Spyropoulou, P (2009), 'Towards digital bloodbanking', ITU-T Kaleidoscope: Innovations for Digital Inclusions, .K-IDI.E-ISBN: 978-92-6112891-3, Print ISBN: 978-92-61-12891-3, pp.I- 8.
5. Anandroid Paper on Video Chatting and an Idea to use on Smartphone Tushar Pandit, Satish Niloor and A.S. Shinde, Dept. of I.T Sinhgad Academy of Engineering, Pune, India