**Proverifier**

Submitted By:

**Md. Salman Shamil**

**ID: 151382347**

**Md. Rokonozzaman**

**ID: 151382349**

**This report submitted in partial fulfillment of the requirements for the**

**Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by:

**Rashidul Hasan Nabil**

**Lecturer**

**Department of CSE**

**City University, Bangladesh**



**Department of Computer Science and Engineering**

**City University**

**Dhaka, Bangladesh**

**June – 2019**

****

**Certificate**

This is to certify that the work presented in this project entitled “Proverifier” is the outcome of the work done by Md. Salman Shamil and Md.Rokonozzaman under the supervision of Rashidul Hasan Nabil, Lecturer, Department of Computer Science and Engineering, City University during January to June 2019. It is also declared that neither this project/report nor any part it has been submitted or is being currently submitted anywhere else for the award of any degree or diploma.

**Approved By:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor

Rashidul Hasan Nabil

Lecturer

Department of Computer Science and Engineering

City University, City Campus’

Dhaka, Bangladesh.

**I**

**Declaration**

This is to certify that the work presented in this project entitled “Proverifier” is the outcome of the work we have done under the supervision of Rashidul Hasan Nabil, Lecturer, Department of Computer Science and Engineering, City University. It is also declared that this project report or any part of it is not a copy of a document done by any organization or being currently submitted anywhere else for the award of any degree or diploma.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Md. Salman Shah

ID: 151382347, Batch: 38th

Department of Computer Science and Engineering

City University, City Campus’

Dhaka, Bangladesh.

**II**

**Declaration**

This is to certify that the work presented in this project entitled “Proverifier” is the outcome of the work we have done under the supervision of Rashidul Hasan Nabil, Lecturer, Department of Computer Science and Engineering, City University. It is also declared that this project report or any part of it is not a copy of a document done by any organization or being currently submitted anywhere else for the award of any degree or diploma.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Md. Rokonozzaman

ID: 151382349, Batch: 38th

Department of Computer Science and Engineering

City University, City Campus’

Dhaka, Bangladesh.

**III**

Acknowledgement

At first we would like to thank Rashidul Hasan Nabil sir, Lecturer, City University, Bangladesh who continuously helped us throughout the project and without his guidance, this project would have been an uphill task.

We extend our sincere thanks to Mrs. Ayesha Siddika, Md. Ataullah Bhuiyan, Sabbir Muhammad Saleh, Ataus Samad sir as our knowledge obtained from their courses help us a lot to implement this project.

Special thanks should also be given to Md. Safaet Hossain, Head of the Department of CSE, City University, Bangladesh. Thank you for giving us the permission to go with ‘Proverifier’.

I also want to acknowledge my indebtedness and render my warmest thanks to the Honorable Dean of Faculty of Science and Engineering, Co-ordinator and all the faculty members of Computer Science and Engineering Department for their constant support.

Finally I thank my parents and classmates for letting me through all the difficulties.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Md. Salman Shah

ID: 151382347, Batch: 38th

Department of Computer Science and Engineering

City University, City Campus’

Dhaka, Bangladesh.

**IV**

Acknowledgement

At first we would like to thank Rashidul Hasan Nabil sir, Lecturer, City University, Bangladesh who continuously helped us throughout the project and without his guidance, this project would have been an uphill task.

We extend our sincere thanks to Mrs. Ayesha Siddika, Md. Ataullah Bhuiyan, Sabbir Muhammad Saleh, Ataus Samad sir as our knowledge obtained from their courses help us a lot to implement this project.

Special thanks should also be given to Md. Safaet Hossain, Head of the Department of CSE, City University, Bangladesh. Thank you for giving us the permission to go with ‘Proverifier’.

I also want to acknowledge my indebtedness and render my warmest thanks to the Honorable Dean of Faculty of Science and Engineering, Co-ordinator and all the faculty members of Computer Science and Engineering Department for their constant support.

Finally I thank my classmates, my brother and my loving mother for letting me through all the difficulties.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Md. Rokonozzaman

ID: 151382349, Batch: 38th

Department of Computer Science and Engineering

City University, City Campus

Dhaka, Bangladesh.

**V**

Abstract

This paper presents the design and implementation of ‘Proverifier’. Proverifier is a web based document verification system. It has developed to avoid the ancient paper attestation system. Using Proverifier soft copy of any document can be uploaded and sent to verifier for verification. The verifier can verify the document or reject the verification request. If the document is verified, it will be archived to the user profile. Then the user will be able to use the verified document for various purposes where attested or authentic document is needed. So it will reduce the hardship of analog document verification or attestation system. Moreover it is time saving, secure and modern system. Because it will be adaptive for tomorrow’s paperless world.

**Keywords: Verification, Attestation, Paperless, Verifier, Document.**

**VI**

**Contents**

Preface Page no

**1 Introduction**

* 1. Proverivier……………………………………………………………………. 1
  2. Background Study……………………………………………………………. 1

1.2.1 Problems with Current System………………………………………. 1

1.2.2 Advantage of Proverifier…………………………………………….. 1

1.2.3 Related works and Comparisms……………………………………… 2

* 1. Project Aims and Objectives…………………………………………………. 2
  2. Project Scope………………………………………………………………… 2
  3. Project Pitfalls………………………………………………………………. 2

**2 Background and Literature Review**

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

2.2 System Development Life Cycle(SDLC)……………………………………. 3

2.2.1 SDLC Phases………………………………………………………… 3

2.2.2 Selection of Methodology (Agile Model)…………………………… 4

2.3 Specification of Language Used……………………………………………... 5

2.4 Specification of Frameworks and Plugins…………………………………… 6

2.5 Codeigniter MVC Framework……………………………………………….. 7

2.5.1 MVC Architecture…………………………………………………… 7

2.5.2 Codeigniter Application Architecture……………………………….. 8

**3 Project Planning and Requirement**

3.1 Introduction………………………………………………………………….. 10

3.2 Aspects Considered in Project Planning…………………………………….. 10

3.3 Software Quality Assurance Plan……………………………………………. 10

3.4 Documentation Standards……………………………………………………. 11

3.5 Project Management…………………………………………………………. 11

3.5.1 Project Planning and Modules……………………………………….. 11

3.5.2 Estimation and Scheduling with Gantt Chart……………………....... 12

3.6 Requirements………………………………………………………………… 12

3.6.1 Software Requirements……………………………………………… 12

3.6.2 Functional Requirements……………………………………………. 13

3.6.3 Non Functional Requirements……………………………………….. 13

**4 System Analysis and Design**

4.1 Introduction to System Analysis…………………………………………… 14

4.2 Principles of System Analysis……………………………………………… 14

4.3 Requirements Analysis……………………………………………………... 14

4.3.1 Requirement Elicitation & Analysis………………………………… 14

4.3.2 Requirement Validation……………………………………………... 15

4.4 The System Design………………………………………………………….. 16

4.4.1 Use Case Diagram…………………………………………………… 16

4.4.2 Entity Relationship Diagram………………………………………….

4.4.3 Database Schema……………………………………………………..

4.4.4 Context / Level 0 Data Flow Diagram……………………………….. 17

4.4.5 Level 1 Data Flow Diagram………………………………………….. 17

4.4.6 Signup Activity Diagram……………………………………………. 18

4.4.6 Login Activity Diagram……………………………………………… 18

4.4.8 General User Activity Diagram……………………………………… 18

4.4.9 Verifier Activity Diagram……………………………………………. 20

**5 System Implementation**

5.1 Signup Page………………………………………………………………….. 21

5.2 Login Page…………………………………………………………………… 21

5.3 Dashboard……………………………………………………………………. 22

5.4 Document Upload Form…………………………………………………… 22

5.5 View Uploaded Documents………………………………………………… 23

5.6 Send Document for Verification……………………………………………. 23

**6 Testing**

6.1 Introduction………………………………………………………………… 24

6.2 Web Application Testing…………………………………………………… 24

6.3 Non-functionality Requirement Testing…………………………………….. 24

6.3.1 Load Testing………………………………………………………… 24

6.3.2 Stress Testing………………………………………………………… 24

6.3.3 Compatibility Testing………………………………………………... 25

6.3.4 Usability Testing…………………………………………………….. 25

6.3.5 Security Testing……………………………………………………… 25

6.3.6 Accessibility Testing………………………………………………… 25

6.3.7 Performance Testing…………………………………………………. 25

6.4 Functionality Requirement Testing…………………………………………... 26

6.5 Unit Integration and System Testing………………………………………… 26

6.5.1 Unit Testing………………………………………………………….. 26

6.5.2 Integration Testing…………………………………………………… 26

6.5.3 System Testing……………………………………………………….. 27

**7 Discussion**

7.1 Program Limitations…………………………………………………………. 28

7.2 Future Scope…………………………………………………………………. 28

7.3 Scope of Improvement……………………………………………………… 29

**8 Conclusion**List of Figures

Preface Page no

**Figure- 2.1:** Phases Of SDLC……………………………………………………….

**Figure- 2.2.2:** Phases of agile model………………………………………………..

**Figure-2.5.1:** MVC Architecture……………………………………………………

**Figure-2.5.2:** Codeigniter Application Architecture………………………………..

**Figure-3.5.2:** Gantt chart…………………………………………………………….

**Figure-4.4.1:** Use Case Diagram…………………………………………………….

**Figure-4.4.5:** Context/Level 0 Data Flow Diagram…………………………………

**Figure-4.4.5:** Level 1 Data Flow Diagram…………………………………………..

**Figure- 4.4.6:** Signup Activity Diagram…………………………………………….

**Figure-4.4.7:** Login Activity Diagram………………………………………………

**Figure-4.4.8:** General User Activity Diagram………………………………………

**Figure-4.4.9:** Verifier Activity Diagram……………………………………………

**Figure-5.1:** Signup page……………………………………………………………..

**Figure-5.2:** Login page………………………………………………………………

**Figure-5.3:** General User Dashboard………………………………………………..

**Figure-5.4:** Document Upload Form………………………………………………...

**Figure-5.5:** View Uploaded Documents…………………………………………….

**Figure- 5.6:** Send Document for Verification……………………………………….

Chapter 1

Introduction

**1.1 Proverifier**

Verification plays an important role in our daily personal or official activities. Sometimes without verification, documents might not be accepted by some organizations. Proverifier intension is to make the document verification system easier.

The name ‘Proverifier’ is the combination of two words, property and verification. Assets or property are classified into two categories- tangible and intangible[1]. Tangible properties are the properties that one can touch, hold, or feel. On the other hand intangible properties are quite opposite. In our observation soft documents are intangible property. Using ‘Proverifier’ soft copy (Portable Document File or Image) of any document can be verified. So we have named the system as ‘Proverifier’.

* 1. **Background Study**

Whereas our system is only for verifying the soft copy of a paper document, so we have not considered any other physical verification system like police verification at residence. Sometimes we need to verify paper based documents for various purposes like- submit for job, school/college/university admission, authentic certification etc. what is known as ‘Attestation’. There are so many problems with the conventional system of attestation.

**1.2.1 Problems with Current System**

In this era of digitalization paper verification or attestation system is so conventional. As an example, in Bangladesh for every government job seeker must have to submit their academic, character or nationality certificates attested by first class government officers to the due authority. For this purpose people have to present at the verifier spot with all paper documents. Then verifier verifies the hard copy of the documents. After that he puts his seal and signature on the hard copy, thus attestation completed. No backup copy of the documents he attested is stored to him. There is a huge chance of deception. Often people use fake seal and signature for attestation. Because sometimes it is difficult for the requisition authority to check on it. So it might be embarrassing for the verifier if someone use his identity for fake attestation. Moreover this process can be disturbing for her.

World of tomorrow will be paperless. Then this analogue process will be useless.

**1.2.2 Advantage of ‘Proverifier’**

Considering the disadvantage of traditional verification and attestation system we feel the need to develop such a system that will be able to make those works easy and authentic. So the online verification system is the demand of time. Using ‘Proverifier’ users will be able to send their documents for verification via internet to the verifier. It will be easier for them than going to a verifier physically. On the other hand verifier can easily access the requested documents what is to be verified from anywhere through internet. It will reduce the gathering around them for paper attestation. Every document file will be converted into a portable document file (PDF). After observing the documents online, if verifier verifies that some sign and information will be added to the pdf file as attestation mark. After verification user can send this file to various company or organization via system email. Verifier also will be able to view the documents he verified if users do not delete that. So it is like a backup copy of documents for the verifier. Moreover the requisition authority to whom the documents have been sent can check the attestation via our website using special id provided with the documents. So it will make the documents more authentic to them. Thus the key advantages of ‘Proverifier’ are:

**1**

* Making the documents as a verified portable soft copy.
* Send verified documents using system email.
* Easy for the requisition authority to check the attestation.
* Verifier access to the documents he verified for all time if that exists.
* Reducing fake attestation or verification.
* Suitable for future paperless world.
* After all it will be time saving for verification and attestation.

**1.2.3 Comparisons with Related Works**

There are not so many project like ‘Proverifier’. Some web based system provides document authentication system. But they do not verify any user document like ‘Proverifier. Of them [trulioo.com](http://www.trulioo.com) is slight similar to this project. Similar system for soft document attestation and verification is rare.

**1.2.4 Project Aims and Objectives**

Although the need of such kind of system envisioned ago, but not yet developed specially for attestation or verification purposes. ‘Proverifier’ can be suitable to meet the necessity of desired system. Here we will discuss about the project hopes to achieve. The objectives of this project enlisted below:

* To create an easy and authentic platform for document verification through internet.
* Making a portable verified documents hub, from where documents can be sent to the due requisition authority.
* Provide a paperless verification system which will be compatible to meet the challenges of future world.
* To provide a solution for reducing fake attestation.

**1.3 Project Scope**

* Proverifier can be used to verify certificates or documents like S.S.C, H.S.C, Honors certificates, Birth certificates or any identity certificates.
* It is designed for the persons or institutions who usually do attestation or verification related work.

**1.4 Project Pitfalls**

* Requires large database storage for archive the user documents.
* Need the soft copy of the documents.

**2**

Chapter 2

Background and Literature Review

**2.1 Introduction**

So far we have discussed about the significance of the problem of present investigation, advantage of our project, aims and objectives of the study and some hypothesis in detail. In this chapter we will try to present a brief review of the related studies. A detailed review of the related studies have done to plan the present study, in selecting methodology, research tool, size of sample, sampling technique and statistical technique used.

A literature review is an evaluative report of studies found in the literature related to one’s selected area[2]. The review should describe, summarize, evaluate and clarify this literature. The form of the literature review may vary with different types of studies.

**2.2 System Development Life Cycle (SDLC)**

Software is a complex product. Developers have to follow some steps from the ideation phase to delivery. Software Development Life Cycle is a conceptual framework or process that considers the structure of the stages involved in the development of an application from its initial feasibility study through to its deployments in the fields and maintenance [3]. Basically SDLC is the collection of some steps involved in a system or software development. It helps the developer firm to deliver a quality product and to maintain the product easily. So before developing any system it need to gather much knowledge about SDLC and select a model.

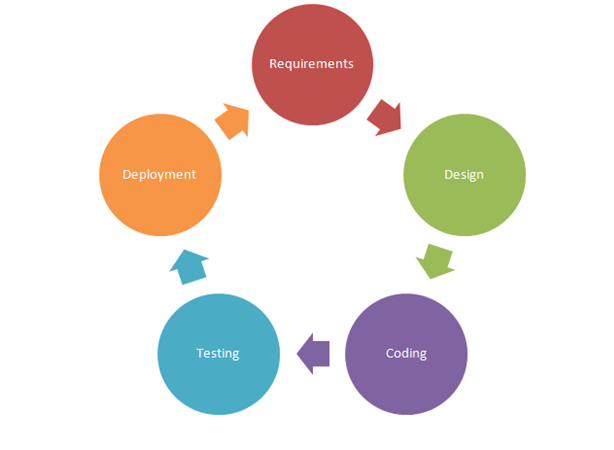
**2.2.1 SDLC Phases**

**Requirements Gathering:** A Software Requirement Specification or SRS is a document which records expected behavior of the system or software which needs to be developed.

**Design:** Software design is the blueprint of the system, which once completed can be provided to developers for code development. Based on the components in design, they are translated into software modules/functions/libraries, etc… and these pieces together form a software system.

**Coding:** During this phase, the blueprint of the software is turned to reality by developing the source code of the entire application. Time taken to complete the development depends on the size of the application and number of programmers involved.

**3**

****

**Figure- 2.1:** Phases Of SDLC

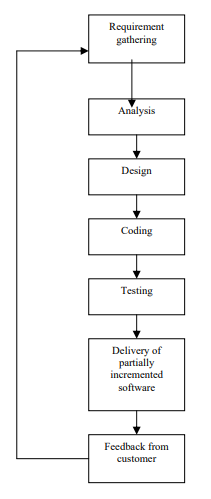
**Testing:** Once the application development is completed, it is tested for various issues like functionality, performance, and so on. This is to ensure that the application is performing as expected. If there are any issues, these issues are fixed before/after going to production depending on the nature of issue and the urgency to go live for the application.

**Deployment:** Once the application is ready to go live, it is deployed on a production server in this phase. If it is developed for a client, the deployment happens in a client premise or datacenter where there client wants to get the application installed.

**2.2.2 Selection of Methodology (Agile Model)**

In software development life cycle, there are two main considerations, one is to emphasize on process and the other is the quality of the software and process itself. Agile software processes is an iterative and incremental based development, where requirements are changeable according to customer needs. It helps in adaptive planning, iterative development and time boxing. It is a theoretical framework that promotes foreseen interactions throughout the development cycle. There are several SDLC models like spiral, waterfall, RAD which has their own advantages. The agile methods are focused on different aspects of the software development life cycle. Some focus on the practices (extreme programming, pair programming), while others focus on managing the software projects (the scrum approach).

**4**



**Figure- 2.2.2:** Phases of agile model

**2.3 Specification of Language Used**

**HTML:** Hypertext Markup Language is simply known as HTML. HTML is used to create hypertext documents that are portable from one platform to another. When a user sends request to a web server it sends some HTML code back to browser. So without HTML we cannot think about web. In our project we have used HTML as Markup language.

**CSS:** Cascading Style Sheet (CSS) is used to control the design of web pages. CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. CSS can be used with HTML following 3 ways- inline, internal and external. CSS frameworks are external. For designing ‘Proverifier’ interface we will mainly use CSS framework, but sometimes inline or custom external CSS will be used.

**5**

**JavaScript:** JavaScript was introduced in 1995 as a way to add programs to web pages in the Netscape Navigator browser[4]. The language has since been adopted by all other major graphical web browsers. It has made modern web applications possible with which can be interact directly without doing a page reload for every action. JavaScript is also used in more traditional websites to provide various forms of interactivity and cleverness. JavaScript is a lightweight, interpreted programming language. JavaScript implementation is easy with HTML. Moreover it is open and supports almost all types of platform. For this reason it’s used widely in web technology.

**PHP:** The elaborated form of PHP is Hypertext Preprocessor. It is a general purpose server side scripting language. The standard PHP interpreter, powered by the ‘[Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine" \o "Zend Engine)’, is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](https://en.wikipedia.org/wiki/Operating_system) and [platform](https://en.wikipedia.org/wiki/Computing_platform), free of charge. There are also many PHP based frameworks for developing web apps.

**2.4 Specification of Frameworks and Plugins**

**Bootstrap:** Bootstrap is a framework for designing website interface. It is an open source toolkit for developing with HTML, CSS, and JS. Bootstrap is the world’s most popular front-end component library for developing responsive and mobile-first projects on the web. It contains design templates for typography, forms, buttons, navigations and other interface components. We have used bootstrap for designing the front end of ‘Proverifier’. Some convenience of bootstrap described below:

* **Fast and Easy Development:** The speed of development is one of the major advantages of bootstrap. It helps to save your coding effort by offering less CSS functionality and pre-built blocks of code rather than structuring code from the scratch.
* **Mobile-First Approach:** The latest version of bootstrap is developed with mobile first approach prospect. With the mobile-first approach it is easy to decide which is the most important part of the content that needs users to see when browsing website on small devices such as phones.
* **Responsive Design:** Using bootstrap responsive design can be done, which make webpage more compatible for smartphones, tablets or desktops.
* **Consistency**: Consistency was the fundamental principle behind the introduction of Bootstrap. It ensures the ultimate consistency regardless of designer/developer, who is working on it. Moreover, the results work uniformly across various browsers and the output remains same.

**jQuery:** jQuery is a JavaScript library that also refered as a mobile framework takes the "write less, do more” principle. Instead of writing unique applications for each mobile device or OS, the jQuery mobile framework allows you to design a single highly-branded responsive web site or application that will work on all popular smartphone, tablet, and desktop platforms.

* Some features of jQuery library is:
* Effects and animations.
* Ajax.
* Extensibility.

**6**

* DOM element selections functions.
* Events.
* CSS manipulation.
* Utilities - such as browser version and the each function.
* JavaScript Plugins.
* DOM traversal and modification.

**DataTables:** DataTables is a powerful jQuery plugin for creating table listings and adding interactions to them. It provides searching, sorting and pagination without any configuration.

* Pagination
* Instant Search
* Multi Column Ordering
* Wide variety of extension
* Use almost any data Source
* Easily theme-able
* Mobile friendly

**Codeigniter:** CodeIgniter is a powerful PHP framework based on the [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) development pattern. It is an open source project which was first released on February 28, 2016. CodeIgniter contains libraries, simple interface and logical structure to access these libraries, plug-ins, helpers and some other resources which solve the complex functions of PHP more easily maintaining a high performance. We have discussed more about codeigniter on on 2.5.2.

**FPDF:** FPDF is a PHP third party library for creating PDF file with pure PHP without using the PDFlib library . It is open source and have almost all the facilities for creating a full featured PDF file. We have used FPDF as a third party library in Codeigniter framework.

**FPDI:** FPDI is a collection of PHP classes facilitating developers to read pages from existing PDF documents and use them as templates in FPDF. So this third party library is used with FPDF. We have also integrate it with Codeigniter for parsing the elements of a PDF file.

**2.5 Codeigniter MVC Framework**

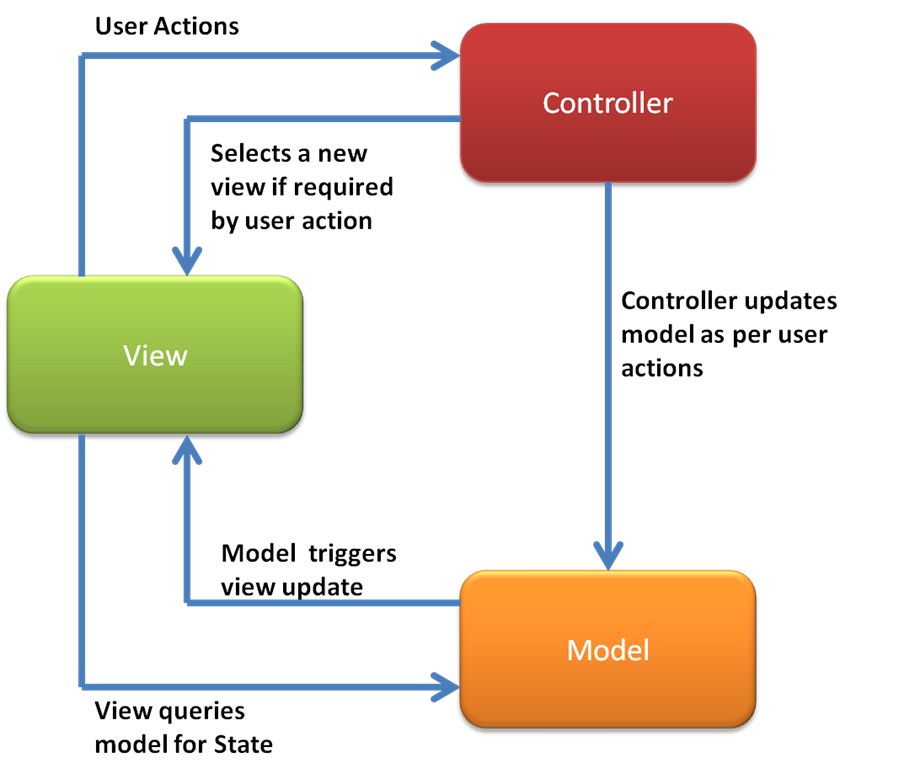
Codeigniter is a MVC pattern based framework. It is lightweight, but very powerfull. It helps to save time, make your site more robust, achieve more sophisticated coding. It makes coding fun again, rather than a chore. There are quite a few frameworks, and not just for the PHP language. Before coding with Codeigniter it need to know about MVC pattern.

* + 1. **MVC Architecture**

**7**

MVC architecture is most used architecture for developing web based software. MVC stands for Model, View and Controller. They done the following job:

* Models for handling data and business logic
* Controllers for handling the user interface and application
* Views for handling graphical user interface objects and presentation

****

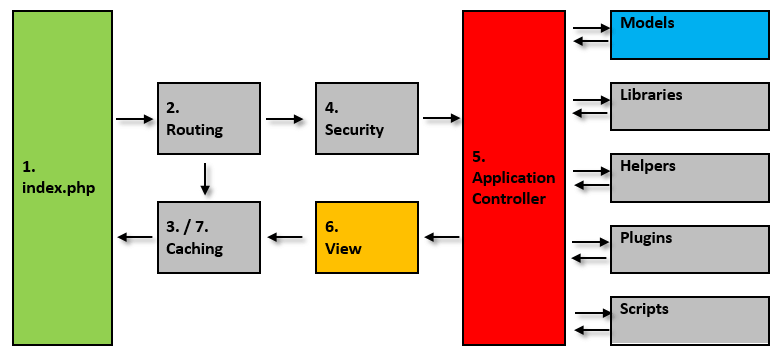
**Figure-2.5.1:** MVC Architecture

The Model-View-Controller architectural pattern helps to tier code for easier maintenance. By keeping the data-manipulating logic separate from the bits that handle the display, you make it much easier to change either the template or the underlying code without touching the other.

**2.5.2 Codeigniter Application Architecture**

The working of Codeigniter Application is mentioned in a simple flowchart given below, which will help you understand the entire process effortlessly in easy steps. Each and every step in the flow chart is explained in elaboration and point wise for your easy grasping.

**8**



**Figure-2.5.2:** Codeigniter Application Architecture.

* As shown in the Flow chart, whenever a request comes to CodeIgniter, it will first go to index.php page.
* In the second step, Routing decides whether to pass the request to step 3 for Caching or to pass the request to step 4 for Security check.
* If the requested page is already in Caching, then Routing will sanction the request to step 3 and the response will go back to the user.
* In case the request page does not exist in the Caching, then Routing, will sanction the requested page to step 4 for Security checks.
* Before passing the request to Application Controller, the Security of the submitted data is checked. After the Security check is done, the Application Controller loads all the necessary Models, Libraries, Helpers, Plugins and Scripts and pass it onto View.
* The View will provide the page with available data and pass that on for Caching, to process this page quickly for future requests.

**9**

Chapter 3

Project Planning and Requirement

**3.1 Introduction**

The project plan defines the work that will be done on the project and who will do it. It is a part of project management. Before starting a software project, it is essential to determine the tasks to be performed and properly manage allocation of tasks among individuals involved in the software development. Hence, planning is important as it results in effective software development.

**3.2 Aspects Considered in Project Planning**

For project planning some important aspects need to be considered. Project planning involves a series of steps that determine how to achieve a particular goal. Some important aspects of project planning are:

* **Requirement Analysis:** Requirement analysis is a must need aspect before starting project development. The requirement analysis we have to identify different methods, techniques and resources. We have identify some tool to develop the project.
* **Resource Plan:** Resource plan indicates available resources for developing a project. This includes an assessment of human resources, capital resources, and financial resources. If the budgeting process does not assess the three areas of resources available it lessens its ability to achieve the goals and objectives.
* **Risk Management:** There might be a risk management plan for developed system. Different types of risk can remain in a system like hardware, information technology, person etc.
* **User Involvement:** User involvement is must for any type of system. Proverifier users are administrator, verifier and general user.
* **Quality Plan:** Quality plan is needed to deliver better product for the users. Different type of testing can ensure software quality. We have discussed about project quality plan in testing chapter.

**3.3 Software Quality Assurance Plan**

Software quality is defined as conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software. We can imagine it as a framework for achieving software quality. Software quality management has three main activities. They are:

**Plan Quality Management:** Identifying the quality requirements and standards for the project and product.

**10**

**Perform Quality Assurance:** Auditing the quality requirements and quality control results to ensure that appropriate quality standards are used.

**Control Quality:** Monitoring and recording the results of quality activities to assess performance and recommend necessary changes.

**3.4 Documentation Standards**

Documentation standards are important in a software project because documents are the only tangible way of representing the software and software process. They are used to produce the documents and also involved in document development. They are used to ensure that a high quality document must be produced. Standardized document have a consistent appearance, structure as well as quality and should be therefore easier to read and understand.

Documentation standards are of three types which are as follows.

Documentation process standards: This defines the process which should be following for the document production. This standard is flexible and able to cope with all types of document.

Document standards: This process governs the structure and presentation of documents. Document should have a consistent style and appearance. The documentation provided along with the “Provifier” carries a consistent appearance throughout.

Document interchange standards: this standard ensures that all electronic copies of documents are very user friendly. This allows the documents of “Provifier”to be easily transferred electronically and re-create in their original form.

**3.5 Project Management**

**3.5.1 Project Planning and Modules**

Our system contains several modules. We planned to develop our system with following modules:

* User Registration Module
* Authentication and Security Module
* Document Verification Module
* Document Parsing Module
* Profile Manage Module

**User Registration Module:** There are two types of user in our system- general user and verifier. Both types of user need registration for using our system. The user registration module will identify what type of user is registering to the system. Then it sends user information to database.

**11**

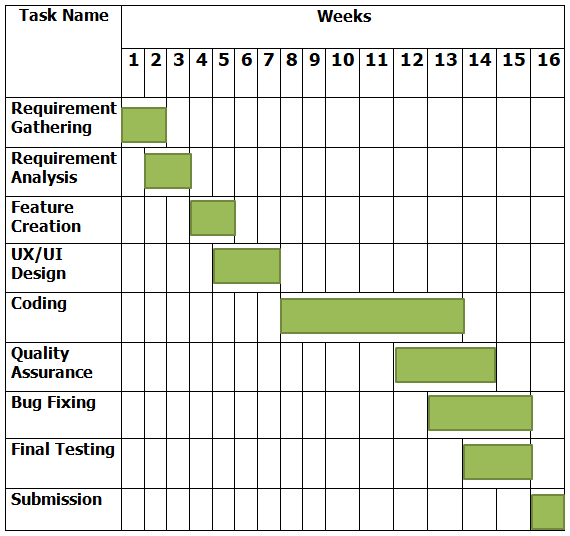
**Authentication and Security Module:** It is the most important module in whole system. This module checks for authentication and give access a valid user to the system. It creates user session. Some other modules get data from this module. Login, dashboard and other internal link access depends on this module. So system security is mostly depends on this module.

**Document Verification Module:** Document verification module task to put some verification mark to the document when a verifier verifies that. Then the document is sent to another directory.

**Document Parsing Module:**  User might upload different types of document of various file formats like-jpg, png, pdf etc for verification. This module parses those documents and converts them info a portable document file.

**Profile Manage Module:** Retrieve profile information from log in session and view them on user profile has been done by this module. Profile info editing, profile pic or logo change, password change are also the task of it.

**3.5.2 Estimation and Scheduling with Gantt Chart**

****

**Figure-3.5.2:** Gantt chart

**3.6 Requirements**

**3.6.1 Software Requirements**

* **Operating System:** Any type server or pc operating system that have web access.
* **Database:** MySQL server.
* **Platform:** PHP based MVC framework (Codeigniter), HTML5, CSS, JavaScript, Bootstrap, jQuery.
* **Library Plugin:** FPDI & FPDF library plugin for Codeigniter.
* **Web Server:** Any PHP compatible web server like Apache. XAMPP/LAMPP/WAMPP for local development.

**12**

**3.6.2 Functional Requirements**

We can define functional requirement as some operations and activities that a system must be able to perform. So the functional requirement specifies something the system should do. Functional requirements of our system are: Accept document file, Document Management, Document verification, File conversion, Document parsing.

**3.6.3 Non Functional Requirements**

Non-functional requirements are not directly related with the system. It deals with customer satisfaction more. Non-functional requirements can include things like response time and reliability. It also called ‘Quality Attributes’. Some non-functional requirement of our project given below:

**Usability:** Usability defines how difficult it will be for a user to learn and operate the system. It mostly depends on user interface and functionality. Our system developed with a very easy user interface and simple functionality that most of people who has minimum knowledge about web and internet will be able to use it.

**Security:** Security refers the protection of user data on the system against unauthorized access. It considers different levels of authorization and authentication across different users roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information. Security also includes protection against viruses and malware attacks.

**Reliability:** Users need to feel secure with the system.  Reliability decreases because of bugs in the code, hardware failures, or problems with other system components.

**Maintainability:** System should be easily maintainable at the admin level. Installation and operation will not be so critical. Our system has those exorcisms.

**Portability:** All the system files and database are portable from on hosting to another. So our application is portable.

**13**

Chapter 4

System Analysis and Design

**4.1 Introduction to System Analysis**

The Merriam-Webster dictionary define system analysis as the “the process of studying a procedure or business that will achieve them in an efficient way”. Another view sees system analysis as a problem –solving technique that breaks down a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose.

**4.2 Principles of System Analysis**

* Understand the problem before you begin to create the analysis model.
* Develop prototypes that enable a user to understand how human machine interaction will occur.
* Record the origin of and the reason for every requirement.
* Use multiple views of requirements like building data, function and behavioral models.
* Works to eliminate ambiguity

**4.3 Requirements Analysis**

Requirement analysis also called requirement engineering is the process of determining user expectation for a new or modifier product. It involves frequent communication with system users. It is a process that involves all the activities required to create and maintain system requirements documentation. The essential principles of requirement engineering activities are:

**Understand the problem:** The problem that user “Proverifier” may face are to be found out and should be solved. This can be done by performing the data gathering techniques like questionnaire, interviews, observation, etc.

**Modeling and analyzing the problem:** Structure analysis the is user as the modeling technique which helps in analyzing the problem found in “Provrifier”.

**Requirement validation:** Requirement conflict negotiation etc. can be solved, because user of the “proverifier”may be different background.

**Requirement management:** In the early phase of continue to evolve Software development phase. So there is a need to manage the changes as the problem evolves.

**4.3.1 Requirement Elicitation & Analysis**

The techniques used for requirement elicitation and requirement analysis:

**14**

**Domain understanding:** This state that the developers of the “Proverifier” should be understand the domain and find out exactly how such system are working in other organizations so that this system can be designed accordingly.

**Requirement collection:** The users of the “Proverifier” are the users , instructor, administrator and their requirements are to be known prior to the system design process.

**Classification:** The activity take the understand collection of requirements and organizes them into logical group. Requirement Elicitation is the practice of collecting the requirements of a system from users, Customer and other stakeholders. Requirement Elicitation and analysis is an iterative process with continual feedback from each activity to other activities. The process cycle starts with domain understanding and ends with requirements checking.

**Prioritization:** The technique enables the “proverifier” designers to arrange the requirement according to their priorities as per the users.

**Requirement checking:** The requirements are checked in accordance with what users really want from the system to discover if they are complete and reliable.

**4.3.2 Requirement Validation**

Validation checks that the product design satisfies or fits the intended usage (High level checking), i. e. the software meets the user requirement. The process of evaluating software during or at the end of the development process to determine whether satisfied requirement of the user.

**Requirement reviews:** It is a normal process which involves multiple readers for checking the requirements of document. The reviewers may also cheek for verification, comprehensibility as well as traceability.

**Prototyping:** In this, the system’s executable model is to be demonstrated by the end users of the system. The prototyping techniques include:

* Use of high level languages.
* Application assembly.
* Database programming.

**Testing:** Requirements should ideally be testable. It allows the developer to appreciate and understand the risk of software implementation. It also improves the quality of the software development.

**Consistency analysis:** To check the consistency, the CASE tool must be built to check all the requirements in this database for the record of all the information.

**15**

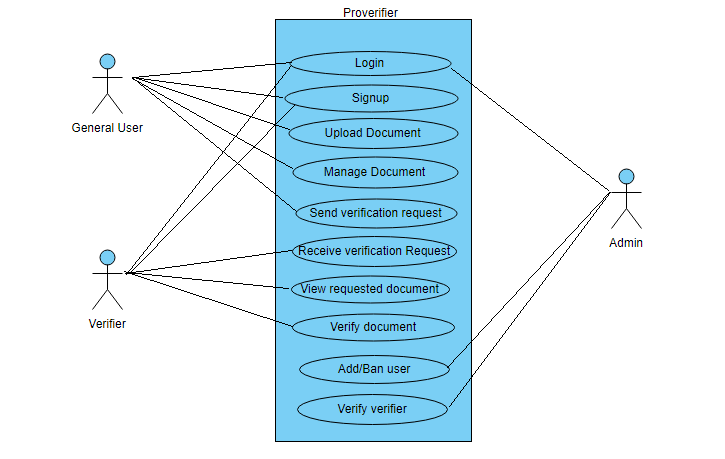
**4.4 The System Design**

System design is the process of defining system methods, functions, objects, and the overall structure and interaction of your code so that the resulting functionality will satisfy your user’s requirements. By system modeling we do not mean expressing a scientific theory or algorithm in system. This is what scientists traditionally call a system model. What we mean here by system modeling is larger than an algorithm or a single method.

System modeling should address the entire system design including interfaces, interactions with other system, and all the system methods. System models are ways of expressing a system design.

The system design phase describes the functional capabilities of the proposed system. This is divided into the following design phases:

**4.4.1 Use Case Diagram**

****

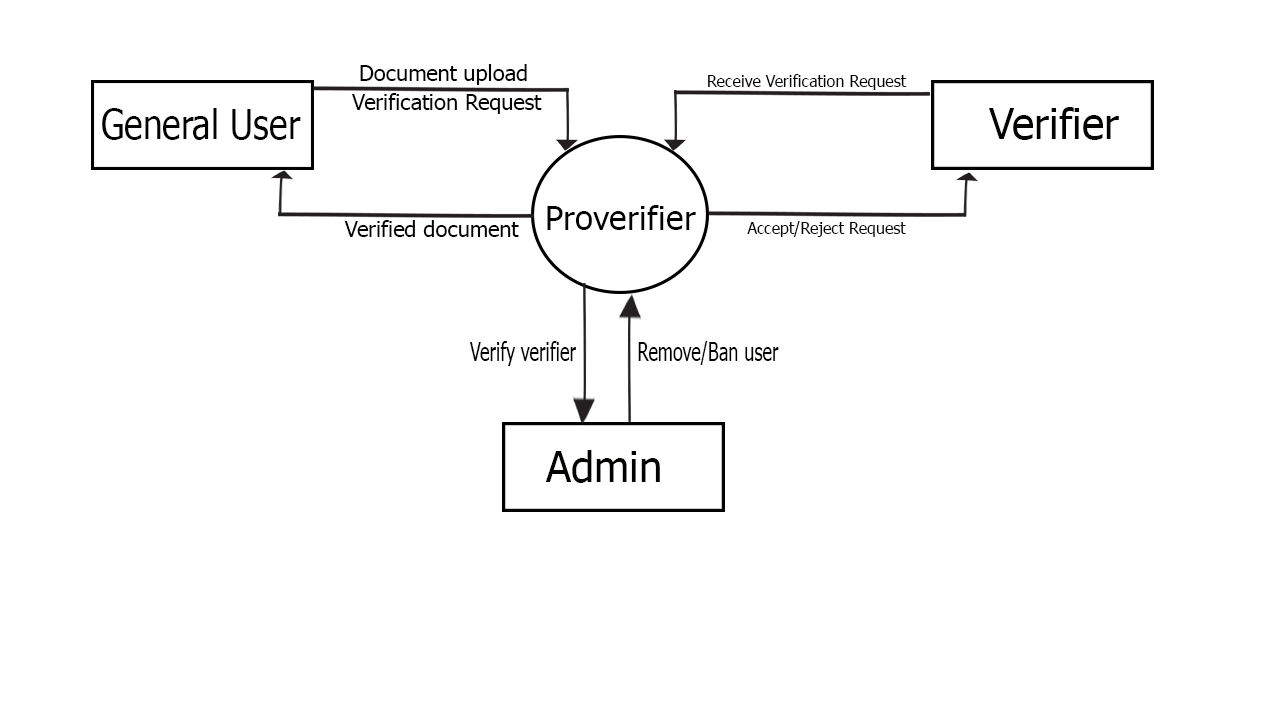
**Figure-**4.4.1: Use Case Diagram

**4.4.2 Entity Relationship Diagram**

**4.4.3 Database Schema**

**16**

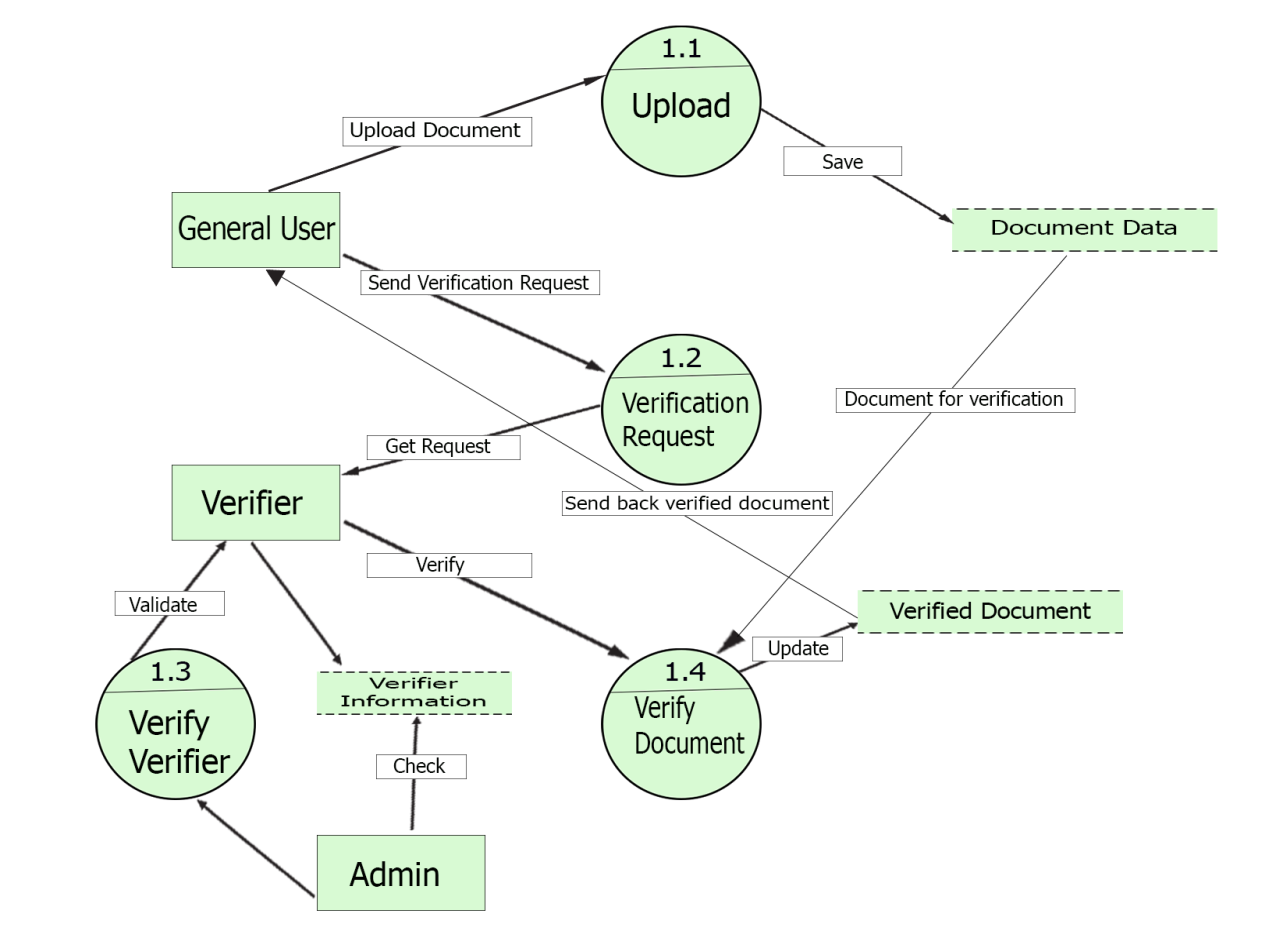
**4.4.4 Context / Level 0 Data Flow Diagram**

****

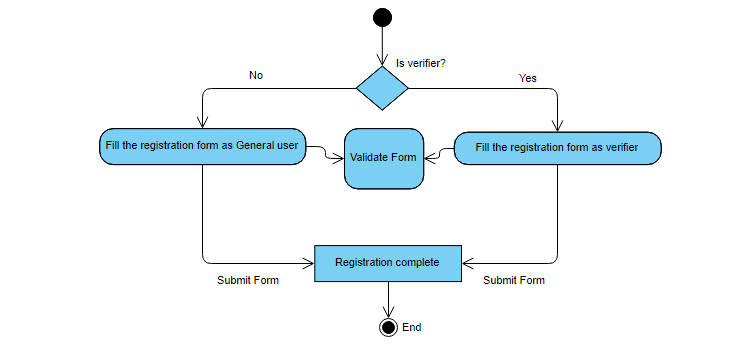
**Figure-**4.4.5: Context/Level 0 Data Flow Diagram

**4.4.5 Level 1 Data Flow Diagram**

**17**

**Figure-**4.4.5: Level 1 Data Flow Diagram

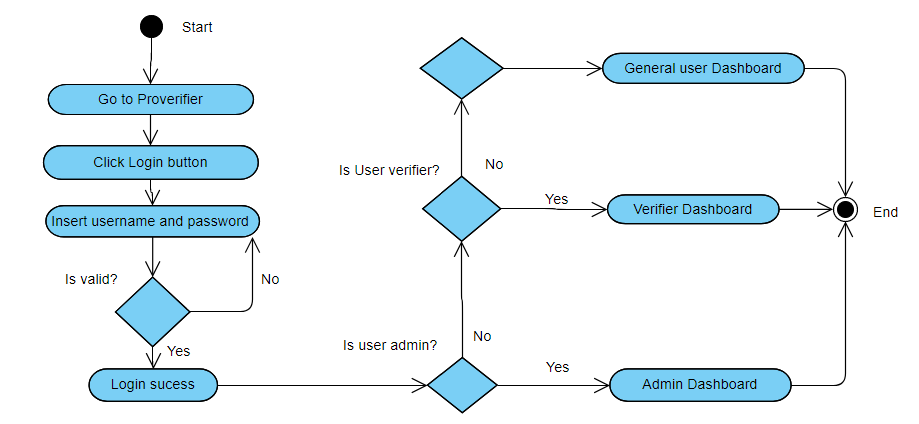
**4.4.6 Signup Activity Diagram**

****

**Figure-** 4.4.6: Signup Activity Diagram

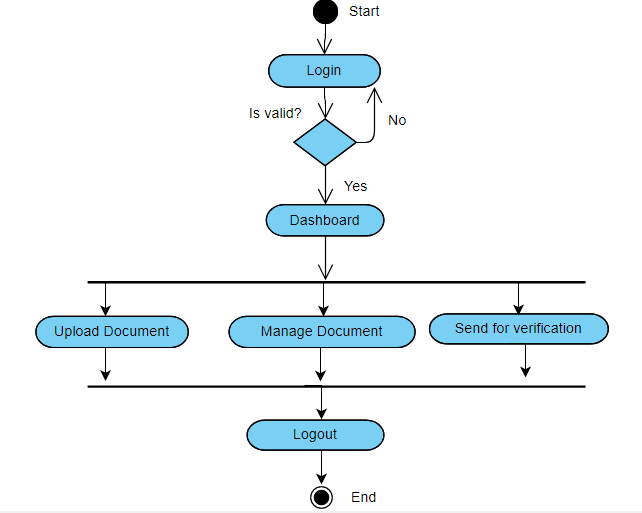
**4.4.7 Login Activity Diagram**

**18**

****

**Figure-4.4.7:** Login Activity Diagram

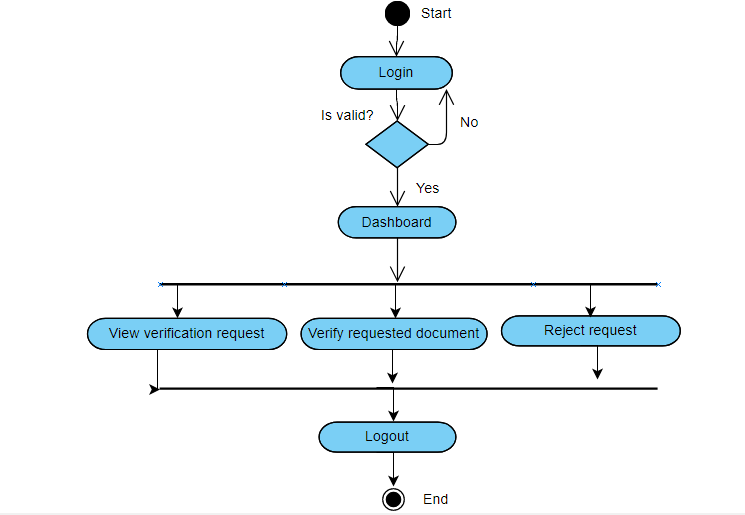
**4.4.8 General User Activity Diagram**



**Figure-**4.4.8: General User Activity Diagram

**19**

**4.4.9 Verifier Activity Diagram**



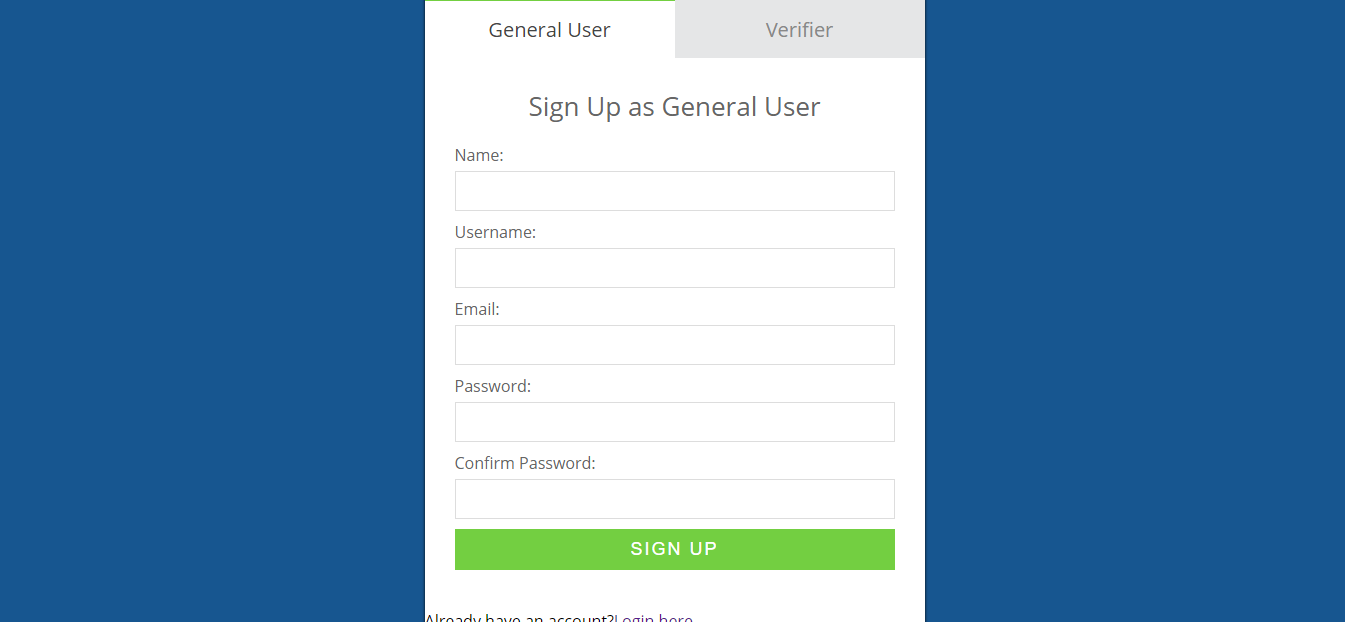
**Figure-**4.4.9: Verifier Activity Diagram

**20**

Chapter 5

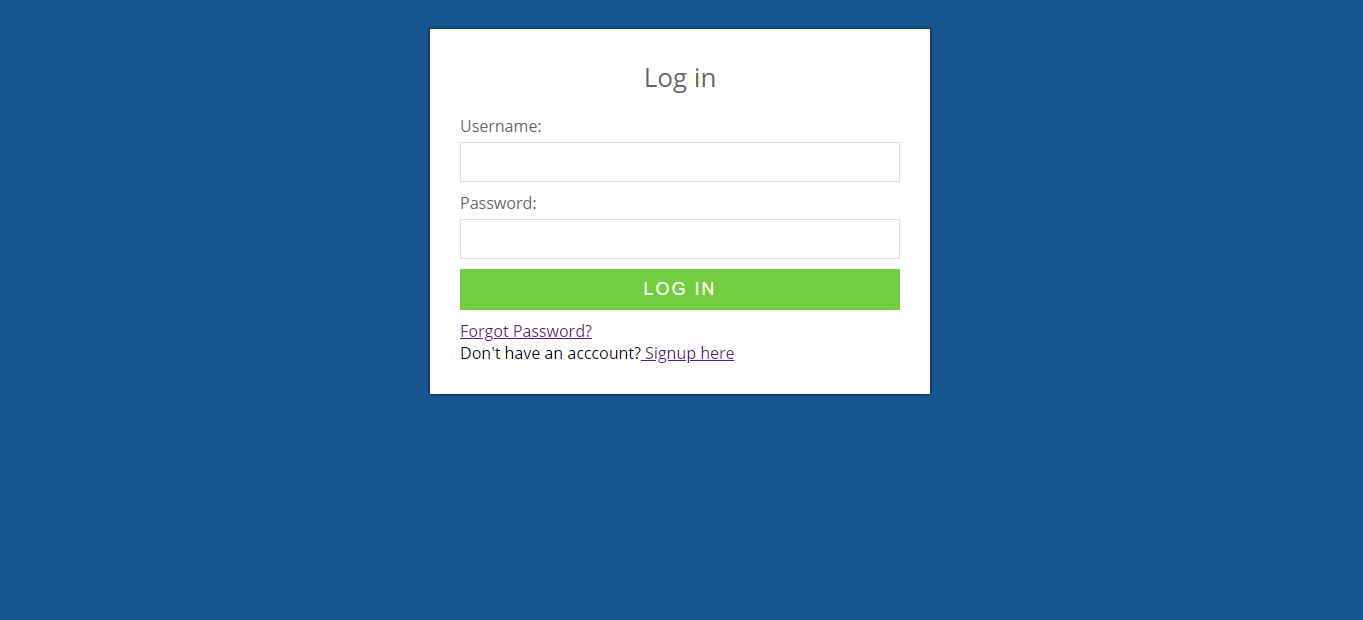
System Implementation

**5.1 Signup Page**

****

**Figure-5.1:** Signup page

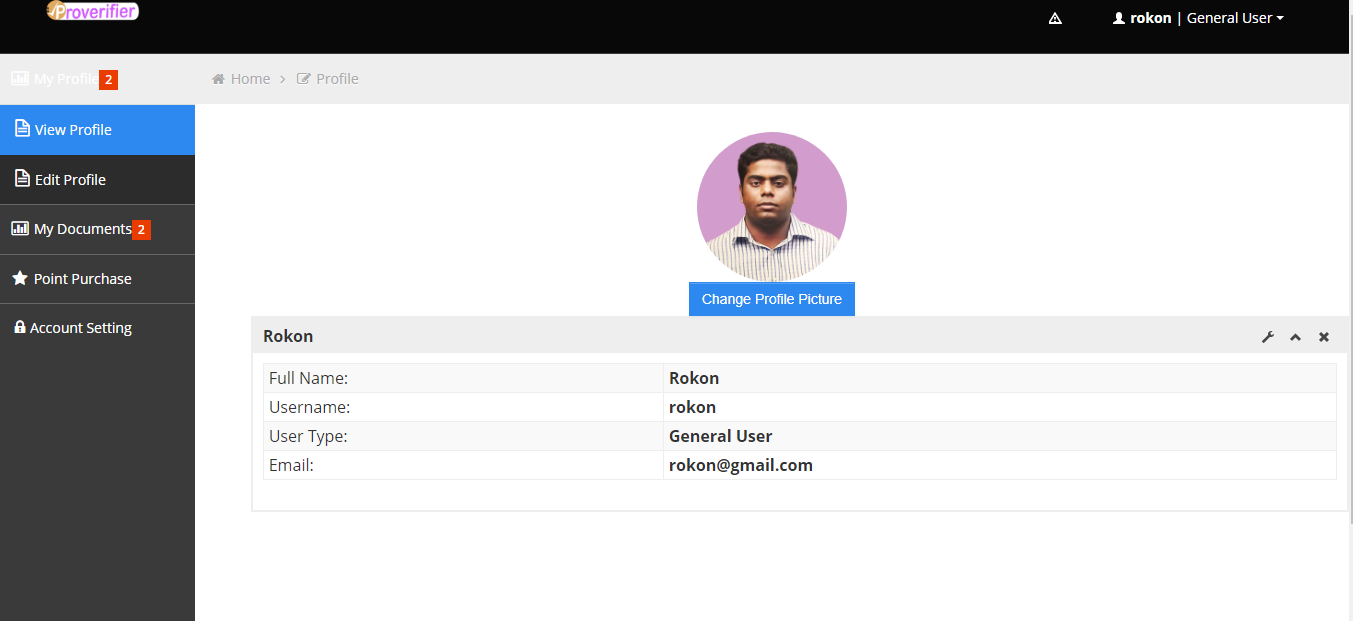
**5.2 Login Page**

****

**Figure-5.2:** Login page

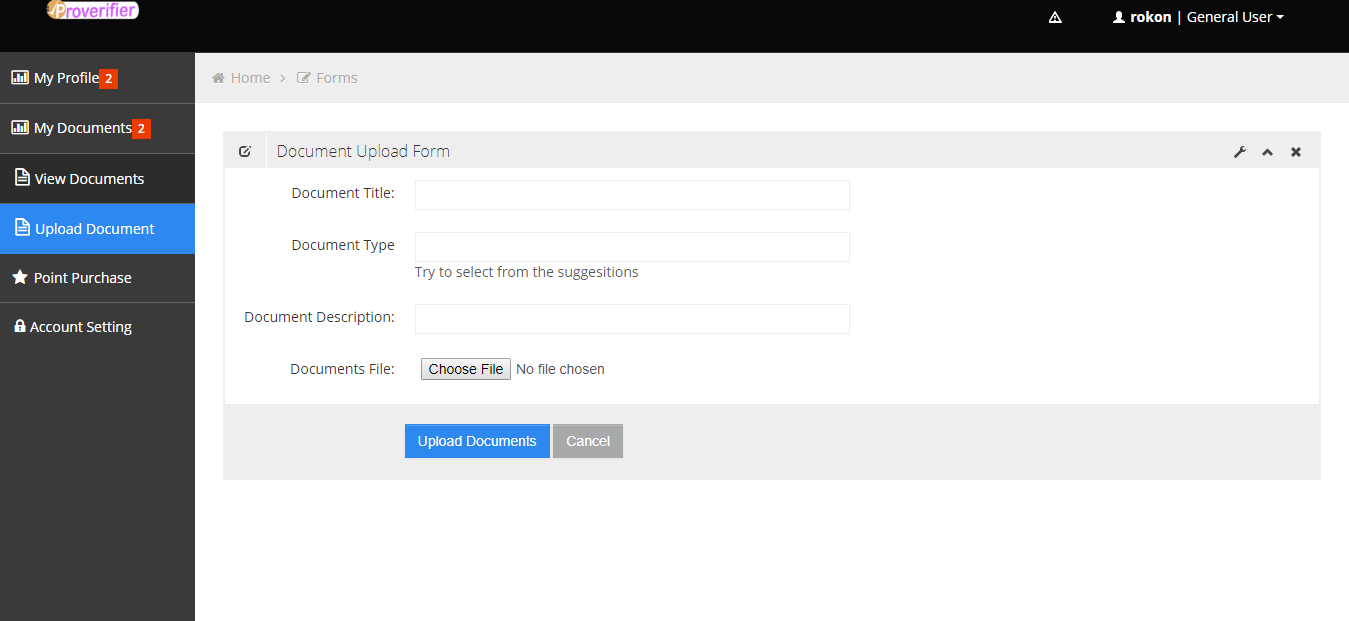
**21**

**5.3 Dashboard**

****

**Figure-5.3:** General User Dashboard

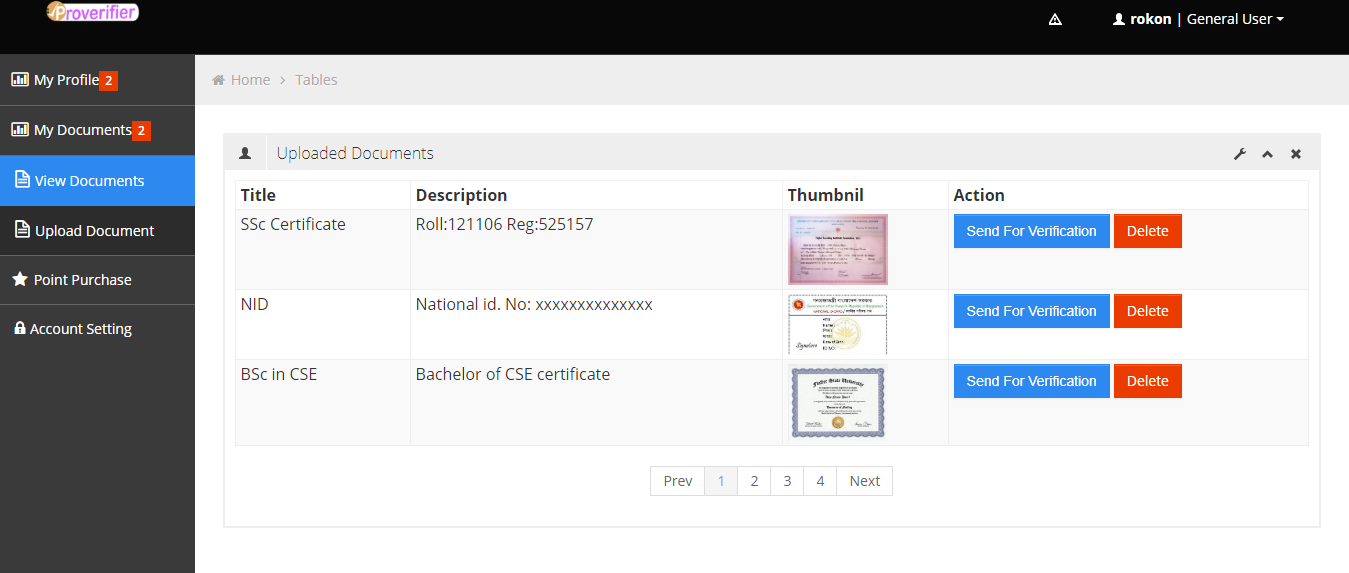
**5.4 Document Upload Form**

****

**Figure-5.4:** Document Upload Form

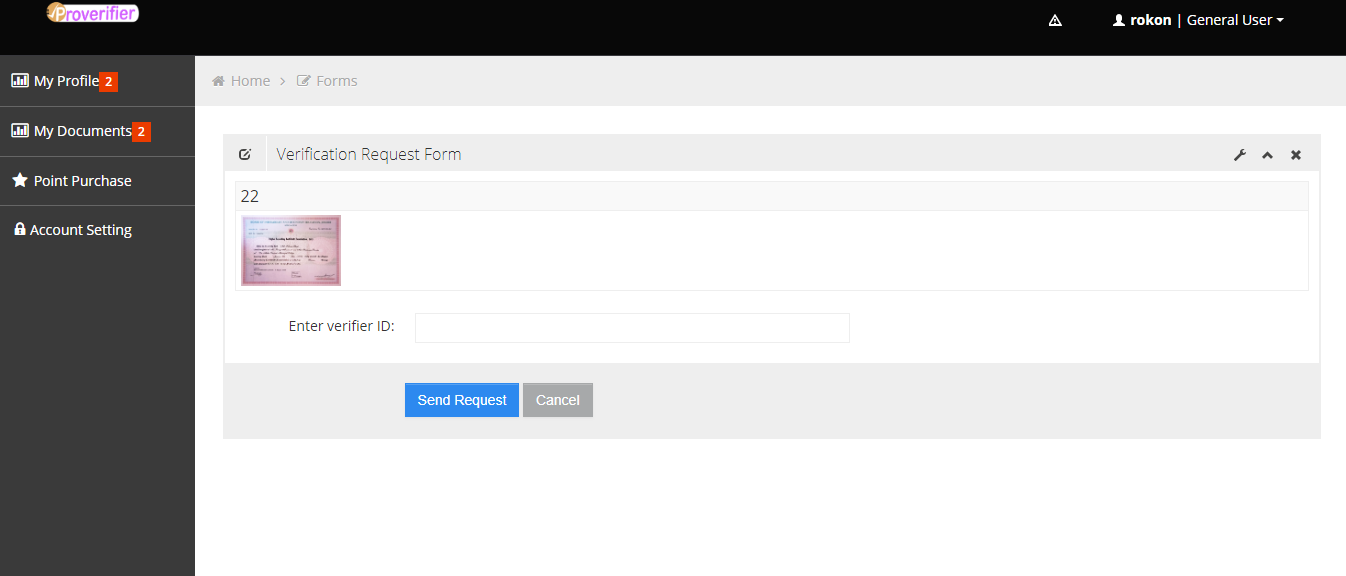
**22**

**5.5 View Uploaded Documents**

****

**Figure-5.5:** View Uploaded Documents

**5.6 Send Document for Verification**

****

**Figure- 5.6:** Send Document for Verification

**23**

Chapter 6

Testing

**6.1 Introduction**

Testing is an important part of any software product. It helps to understand the capability of the system to survive under worst case scenario. Software testing is an integral part of the software development life cycle which identifies the defects, flaws or the errors in the application. Proper testing ensures the quality of the product and without testing software cannot be acceptable to the client.

**6.2 Web Application Testing**

As our system is web based, so we have followed web application testing strategies. The main goal of testing a web application is to run the application using combinations of input and state to discover failures. A failure is the manifested inability of a system or component to perform a required function within specified performance requirements [5]. For web application it need to tests functional requirements, non-functional requirements, unit integration and system testing [6]. We basically tested non-functional requirements and unit integration and system.

**6.3 Non-functionality Requirement Testing**

Non-functional requirements includes-

1. Load testing
2. Stress testing
3. Compatibility testing
4. Usability testing
5. Accessibility testing
6. Security testing
7. Performance testing

**6.3.1 Load Testing**

Load testing refers time needed to perform several tasks and functions under predefined conditions. These predefined conditions include the minimum configuration and the maximum activity levels of the running application. It also indicates how many user requests it can handle at a time for a same page. It will depend on the server configuration that our system can take how much load.

**6.3.2 Stress Testing**

Stress testing is used to verify the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. Using stress test it can be understand that how the system recovers from crashes. Stresses are usually given on different input fields like log in and sign up areas.

**24**

Failures found by stress testing are mainly due to faults in the running environment.

**6.3.3 Compatibility Testing**

Compatibility testing is carried out to determine if an application runs as expected on a running environment that has various combinations of hardware, software, and middleware. It is an important aspect of testing. We have tested the following compatibility testing:

* **Cross Browser Test:** Our system has tested on different operating environment and different browser. It runs in every browser very fluently.
* **Responsiveness:** Our system interface is responsive. It fitted well with smartphone, tablet and desktop screens by changing its layout device to device.
* **Printing Option:** Page printing option executed very well with proper page alignment, fonts, page graphics etc. Page is fit for printing or can be adapted using browser printing option.

**6.3.4 Usability Testing**

Usability testing aims to verify to what extent an application is easy to use. Usually, design and implementation of the user interface both affect usability. Thus, usability testing is mainly centered around testing the user interface: issues concerning the correct content rendering (e.g. graphics, text editing format) as well as the clarity of messages, prompts, and commands that are to be considered and verified.

**6.3.5 Security Testing**

We have tested the following cases for security testing:

* Web application directories can’t be accessed directly using browser.
* Dashboard URL is entered into the browser address bar without logging in. Access was forbidden. This process also done for several internal pages. Result was positive.
* If user is logged in using username and password and browsing internal pages then try changing URL option directly.

**6.3.6 Accessibility Testing**

Accessibility testing can be considered a particular type of usability testing whose aim is to verify that the access to an application’s content is allowed even in the presence of reduced hardware and software configurations on the client side (e.g. browser configurations disabling graphical visualization, or scripting execution), or in the presence of users with disabilities, such as visual impairment.

**6.3.7 Performance Testing**

Application performance testing is the way to measure your Web application’s ability to conduct multiple transactions simultaneously while maintaining adequate response times. It also helps you isolate bottlenecks in any component of the network infrastructure to accurately pinpoint potential problems. We have tested our system performance on a local server. The result was satisfactory.

**25**

**6.4 Functionality Requirement Testing**

Functional requirement testing is a type of software testing whereby the system is tested against the functional requirements/specifications.

* **Testing levels:** which specify the different scope of the tests to be carried out, i.e. the collections of components to be tested.
* **Test strategies:** which define heuristics or algorithms to create test cases from software representation models, implementation models, or test models.
* **Test models:** The models which represent the relationships between a representation’s elements or a component’s implementation.
* **Testing processes:**  which define the flow of testing activities, and other decisions such as when to start testing, who is to perform the testing, how much effort should be used, etc.

**6.5 Unit Integration and System Testing**

The flow of activities of a software testing process usually begins with unit testing and proceeds with integration and system test. The aim of unit testing is to verify each application’s individual source code component, while integration testing considers combined parts of an application to verify how they function together. Finally, system testing aims at discovering defects that are properties of the entire system rather than of its individual components.

**6.5.1 Unit Testing**

In unit testing individual units or components of software are tested. Units refer the smallest testable part of software. It is important to choose which part of the application components to be tested individually.

**6.5.2 Integration Testing**

Web application is consist of many units and pages. Combined them together the system work. Integration testing is a software testing technique where individual units of a program are combined and tested as a group. Integrating testing may detect errors when modules are integrated to build the overall system. We have tested different module separately. After integration we also tested the combined system. Both works fine.

**26**

**6.5.3 System Testing**

System testing is done to find out the defects in whole software. Two approaches are used for system testing to finding defects and bug in the system. One is black box testing and white box testing. However greybox testing is exists. But most of the cases we have used white box testing. We have done the following system tests on our software:

* **Functionality Testing:** All the functionalities of our system worked properly. Output was satisfactory. White box testing has been done for functionality testing.
* **Forms and Input Validation Testing:** Forms taking input without any error. Validation has been done for the input fields. Invalid input was not taken.

**27**

Chapter 7

Discussion

For this project the most challenging work was integrating two libraries with codeigniter. First library is ‘FPDI’, which is used to parsing the elements of an existing ‘pdf’ file and use elements as a template for another library. We hardly found that one library for PHP to parse pdf elements. Another one is ‘FPDF’, which is used to generate ‘pdf’ file on a web server. We have also learned lots of lesson for this project. During development period we faced some inconvenience, but meet them successfully. That helps us to enhance our problem solving skill and experience.

**7.1 Program Limitations**

* Hard copy of the documents needs to scan or takes photo before sending it for verification to the verifier.
* Only specific types of documents format is supported by the system.

**7.2 Future Scope**

This system is developed with multipurpose use concept. In future there is a scope to add the following modules:

* Adding some additional Functionality this system can be used as verified document hub. Where different types of documents can be stored by the users and make them verified through the verifier. Then user will be able to use the soft documents in many ways.
* Online certification can be possible using this system. An authorized person or institution will issue specific types of certificate to a user. Then the certificate will be added to his system profile.
* Any type property status verification using this system is another future scope of the project. It will ensure online ownership statement to the users.
* Copyright statement reservation and product registration also can be possible.

**7.3 Scope of Improvement**

* User interface need to improve for better user experience.
* Code optimization is another scope of improvement. Code optimization can speed up the site.
* Some database redundancy need to take aside to save space in database.
* Some neoteric security and authentication system can be integrated with it. Those will improve system security and make ‘Proverifier’ more acceptable to the users.

**28**

Chapter 8

Conclusion

Though ‘Proverifier’ is designed and developed for multipurpose verification or attestation work, but in future it can be used for better authentic document hub by adding some extra functionality. It existing advantages like harass less attestation, document authentication and document portability have make it unique and usable considering the present situation. This system also has some common advantage that might subsist on the most online based system like- it will be time saving, easily accessible and paperless documentation. So at the end, it can be said that ‘Proverifier’ is the optimum solution for online document attestation and verification.

Moreover, it is developed maintaining all aspects of Software Development Life Cycle. So this system can be suitable for government and non-government offices, their servants or authorized persons as well as general users.

**29**

REFERENCE

[1] Raymond H. Peterson (2002). *Accounting For Fixed Assets* (2nd Ed.). P-11

[2] Boote, D.N. & Beile, P. (2005). Scholars before researchers: On the centrality of the dissertation literature review in research preparation. Educational Researcher 34/6, pp.3-15.

# [3] [Nayan Ruparelia](https://www.researchgate.net/profile/Nayan_Ruparelia), “Software development lifecycle models”, <https://www.researchgate.net/publication/220631422>, May 2010 Volume 35, P-8.

# [4] Marijn Haverbeke (2018), *Eloquent JavaScript* (3rd ed.)*,* p-6.

[5] Hower R (2005),” Web Site Test Tools and Site Management Tools”, Software QA and Testing Resource Center. (accessed 5 June 2005).

[6] Giuseppe A. Di Lucca, Anna Rita Fasolino, “Web Application Testing”, https://www.researchgate.net/publication/226306110, 05 August 2014, pp. 222-233.

**30**

Appendix

|  |  |
| --- | --- |
| PDF | Portable Document File |
| HTML | Hypertext Markup Language |
| PHP | Hypertext Preprocessor |
| CSS | Cascading Style Sheets |
| UML | Unified Modeling Language |
| SDLC | Software Development Life Cycle |
| DFD | Data Flow Diagram |

**31**