

CERTIFICATE GENERATOR TOOL

FINAL REVIEW REPORT

Submitted by

Rupin Patel (19BCB0030)

Raghav Anand (17BCE0430)

Sameer Bishnoi (19BCB0029)

Yash Agarwal (19BCE0691)

Prepared For

SOFTWARE DESIGN AND DEVELOPMENT (CSE1005)

PROJECT COMPONENT

Submitted To

Ragunthar.T

Assistant Professor (Senior)

School of Computer Science and Engineering



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

1. Introduction

2. Project Scope

3. Key contacts and stakeholders

4. Project resource requirements

4.1. Software resource requirements

5. Software Requirement Specifications

6. Design contents

6.1. Use case diagram

6.2. Work Breakdown Structure

6.3. Activity Diagrams

6.4. DFD Diagrams

6.5. Sequential Diagrams

6.6. Class Diagram

6.7. State Chart Diagrams

6.8. Component Diagram

6.9. Deployment Diagram

7. Test case

8. Implementation

Introduction

Design a certificate generator tool that automatically maps the contents of a given certificate. Retrieves the data of the certificate holder from a .csv file and prints the details accordingly at their intended location. The tool will generate a certificate by reading .csv file or user will enter the data manually and the generated certificate will be stored in a local storage as well as on online firebase server with unique has values. In existing system there was not any facility of auto mailing while generating certificates, no online backup and verification data is not proper.

So there are some features which had been implied in this tool to solve such problems.

1. PROJECT SCOPE

Certificate Generator Tool aims at automation of the following processes

- An organization who wishes to manage large number of certificates and on verify certificates on large scale.
- Certificate Generation with unique hash values
- Stored Generated certificates in local storage and also in firebase storage bucket.
- Auto email Generation about the generated certificates to candidate.
- Report Generation of certificates with unique id and stored it in a .txt file for referencing.

2. KEY CONTACTS AND STAKEHOLDERS

Name	Registration Number	Phone Number
Rupin Patel	19BCB0030	7069490002
Sameer Bishnoi	19BCB0029	8780764945
Raghav Anand	17BCE0430	8949911918
Yash Agarwal	19BCE0691	9667116877

3. PROJECT RESOURCE REQUIREMENTS

3.1 Software Resource Requirements

- Windows 10
- Python
- HTML 5
- CSS
- jQuery
- JavaScript
- Firebase
- LucidChart
- .exe reader

Software Requirements Specification

for

Certificate generator tool

4.1.1 Table of Contents

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms and Abbreviations
 - 1.4 Reference
- 2. Overall Description
 - 2.1 Product Perspective
 - 2.2 Product Functions
 - 2.3 User Characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and dependencies
- 3. External Interfaces Requirements
 - 3.1 System Interfaces
 - 3.2 User Interfaces
 - 3.3 Software Interfaces
 - 3.4 Communication Features
- 4. Functional Requirements
 - 4.1 Safety Requirements
 - 5.3 Security Requirements
 - 5.4 Software Quality Attributes
- 6. Other Requirements

1. Introduction:

The Project will allow user to create certificate by editing parameters of in-built templates. The project will read all parameters from .csv file and paste each parameter on respected coordinates of certificate template and store the generated .jpg format certificates in online storage bucket (Firebase) and automatically mail the generated certificates to respected participants with verification code.

1.1 Purpose:

The purpose of this document is to capture, in natural language and at a functional level, the description and requirements of certificate generator tool. The focus here is the users and admin needs in any competitions, trainings, workshops, etc. This is a functional description of those features required to address current certificate

generation system requirements. A short discussion accompanies each requirement, to add the background and framework necessary to explain the functionality. It also describes nonfunctional requirements and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

1.2 Scope:

This software system will be a GUI tool for any Organizations who wishes to manage large number of certificates and on verify certificates on large scale. More specifically to design and develop a simple and intuitive system which shall cater the certification needs of any institute. The system shall provide features to the user and admin of an institute to be part of different types of groups, such as competitions, internships, trainings various Hackathons and share data and documents to participants and concerned admins through auto mailing system.

1.3 Definitions, Acronyms and Abbreviations:

Certificate generator Tool:

The certificate generator tool that automatically maps the contents of a given certificate. Retrieves the data of the certificate holder from a file Writes/prints the details accordingly at their intended location, uploaded on online database and shared the certificates to concerned admins and participants through mail.

Admin:

Admin is user which has rights to manage certificates, verifications codes and signatures to be pasted on certificates.

System:

System refers to the existing certificate tool and has improved features and user interface in a new certificate generator tool.

1.4 References:

- 1) IEEE 830 Template
- 2) GlobalCert requirement Problem statement
- 3) GlobalCert certification process requirement documentation.

2. Overall Description:

2.1->Product Perspective:

The Product is based on image [processing by using openCV. We can observe image processing concept in our day to day life such as scanners, cameras, image filters etc. By using concept of image processing we can put some parameters on particular coordinates of image. By the help of this feature we can generate forms and certificates.

2.2->Product Functions:-

The tool will generate a certificate by reading .csv file or user will enter the data manually and the generated certificate will be stored in a specific folder. The generated certificate will be sent to the concerned participant, via mail. This is according to the participant's email id in the .csv file or email id entered in input box. The generated certificates will be stored in the Google firebase storage bucket for retrieving and referencing the data.

2.3->User Characteristics:-

The Admin:-

He/ She will generate certificates by running this tool. And get access to participant's credentials.

The Participants:-

The participants can check their certificates on web page and also get certificates information through mail with credential to validate it in future.

2.4->Constraints:-

User Interface Constraints:-Using this system is fairly simple and intuitive. A user familiar with basic browser and tool navigation skills should be able to understand all functionality provided by the system.

Hardware Constraints:-The system should work on most home desktop and laptop computers which support Python, JavaScript and HTML.

Software Constraint:-The system will be intended to run on Firefox 4 and above, Google Chrome 10 and above and Internet Explorer 8 and above. And Python3.7 and greater versions must be installed.

Data Management Constraints:-System shall be able to interface with other components according to their specifications.

Operational Constraints:-The system is limited by its operating server in terms of the maximum number of users it can support at a given time.

2.5->Assumptions and dependencies:-

Most of the certification portals and tool have a lot of redundant features which are rarely used. Our new system focuses on the features which are most important to the users of an institute or organization along with introduction of some new features which other portals lacks.

3. External Interface Requirements:

3.1->System Interface: Python Tkinter is used as interface for inputs from user and Firebase is used as an online storage. The Web page the user inputs data via the web server using HTML forms. The actual program that will perform the operations is written in JavaScript.

3.2->User Interface: The new system shall provide a very intuitive and simple interface to the user and the administrator, so that the user can easily navigate through pages, and generate certificates and easily choose required template. The interface will allow user to easily select required parameters to generate required certificates.

3.3->Software Interface:

Tools side: Python libraries and python-backend (Pyrebase) server to upload certificates online and Python SMTP to send mail to each participant.

Online server: Firebase storage bucket connected with parameters and API keys to Tool side server.

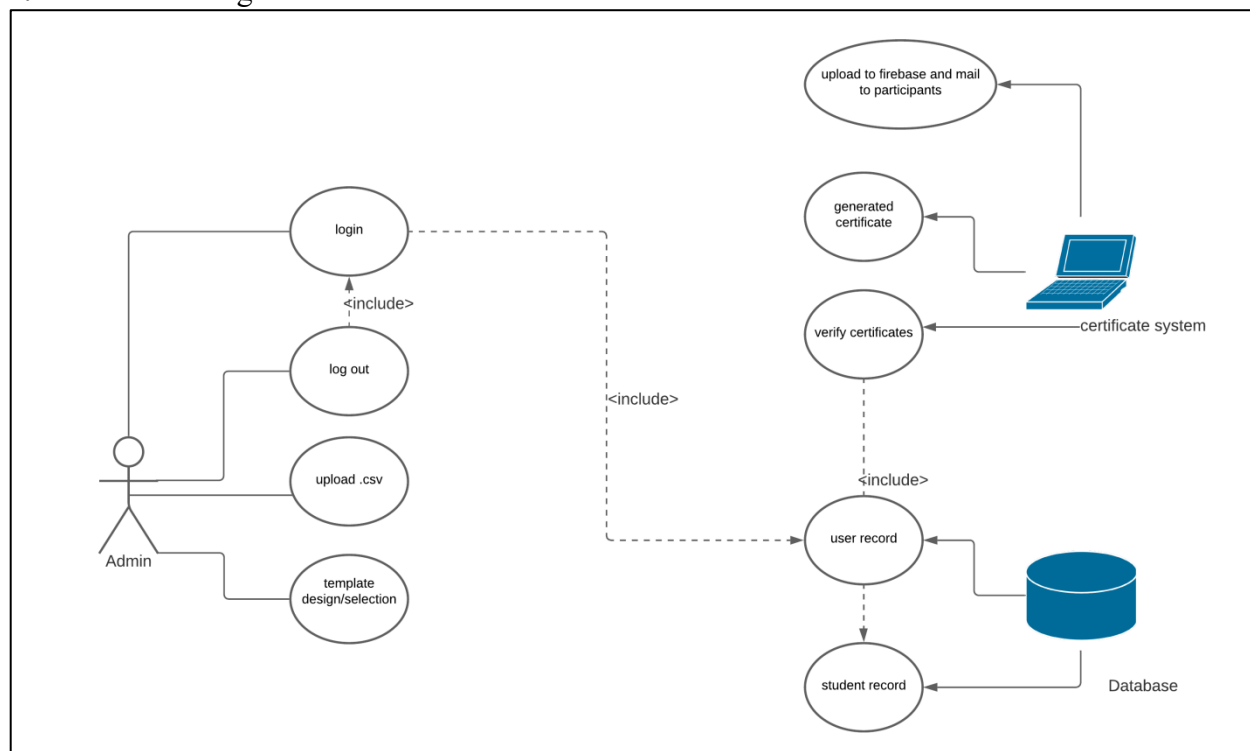
Client side: The website based on HTML and Javascript fetching certificates from Online server.

3.4->Communication Interfaces:

The HTTP or HTTPS protocol(s) will be used to facilitate communication between the client and server. And SMTP protocol for auto mailing.

4. Functional Requirements:-

4.1->Use case diagram:



4.2->Use case description:-

- 1) A simple authentication page Login->Logout will be initial Route
- 2) A system will ask Admin to upload .csv file
- 3) Then Admin will select required templates and parameter
- 4) System will generate certificates upload to online server and mail the generated certificates to participants
- 5) The Database will store credentials of each participant certificate.

4.3->System features:-

- ▶ The generated certificate will be sent to the concerned participant, via mail. This is according to the participant's email id in the .csv file or email id entered in input box.

Auto Mailing System

In existing system there was not any facility of auto mailing while generating certificates

- ▶ The generated certificates will be stored in the Google firebase storage bucket for retrieving and referencing the data

The generated certificate is uploaded in the Google Firebase, according to the name of the participant in .jpg format

- ▶ Verification of the certificate by any organisation or authority

We will provided a unique Code for each certificate

When this Code is entered in online backup website, it redirects to a page where you can see particular certificate

This webpage will show whether the credentials are valid or not

9. Non Functional Requirements:-

9.1->Usability:-

The system can be easily implemented anywhere any time to generate instant certificates by uploading .csv file. The user can easily verify the generated certificates on web page anytime. And the data will be stored in database.

9.2->Security:-

1. Passwords will be saved encrypted in the database in order to ensure the user's privacy.
2. The user's IP will be logged.
3. The system will be protected against vulnerabilities such as database not found ,less secure app permission not given to mailing system.

9.3->Performance:-

The system should support at least 200 concurrent users. This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of users be able to access the system at the same time, since the certification portal is important credentials system. The times when the system will be under the most stress are likely during generation and storing of certificates. Therefore, it must be able to handle at least 200 concurrent users.

9.4->Availability:-

The system should be available at all times, meaning the user can access it using a web browser and tool , only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved with the firebase server and saved by the administrator.

9.5->Reliability:-

The reliability of the overall program depends on the reliability of the separate components.

9.6->Maintainability:-

MySQL and firebase is used for maintaining the database and the web page server takes care of the site. In case of a failure, a re-initialization of the program is recommended.

9.7->Probability:-

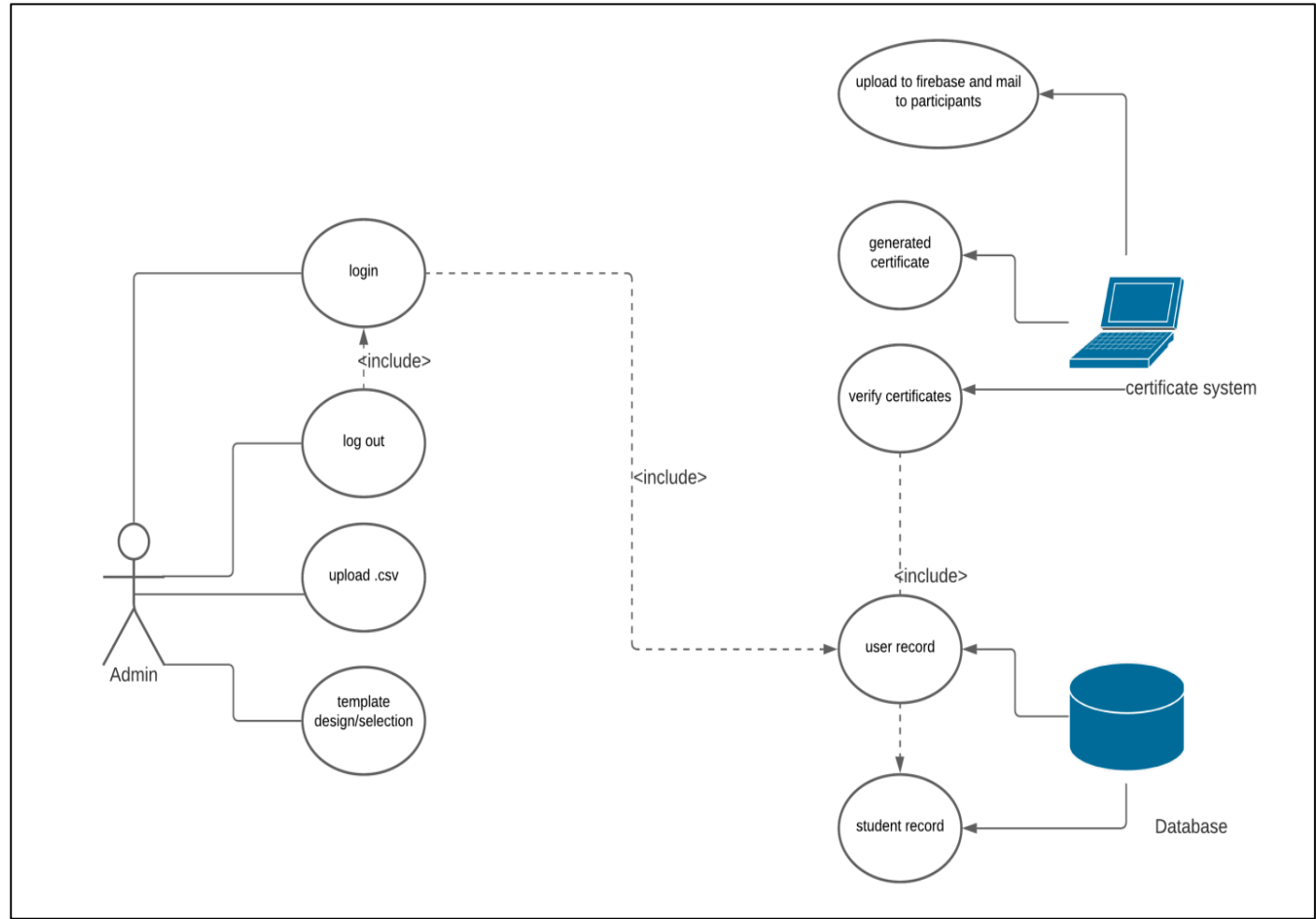
The application should be compatible with other systems. Node-js, python, firebase programs are practically independent of the OS-system which they communicate with. The end-user part is fully portable and any system using any web browser should be able to use the features of the application.

9.8->Privacy:

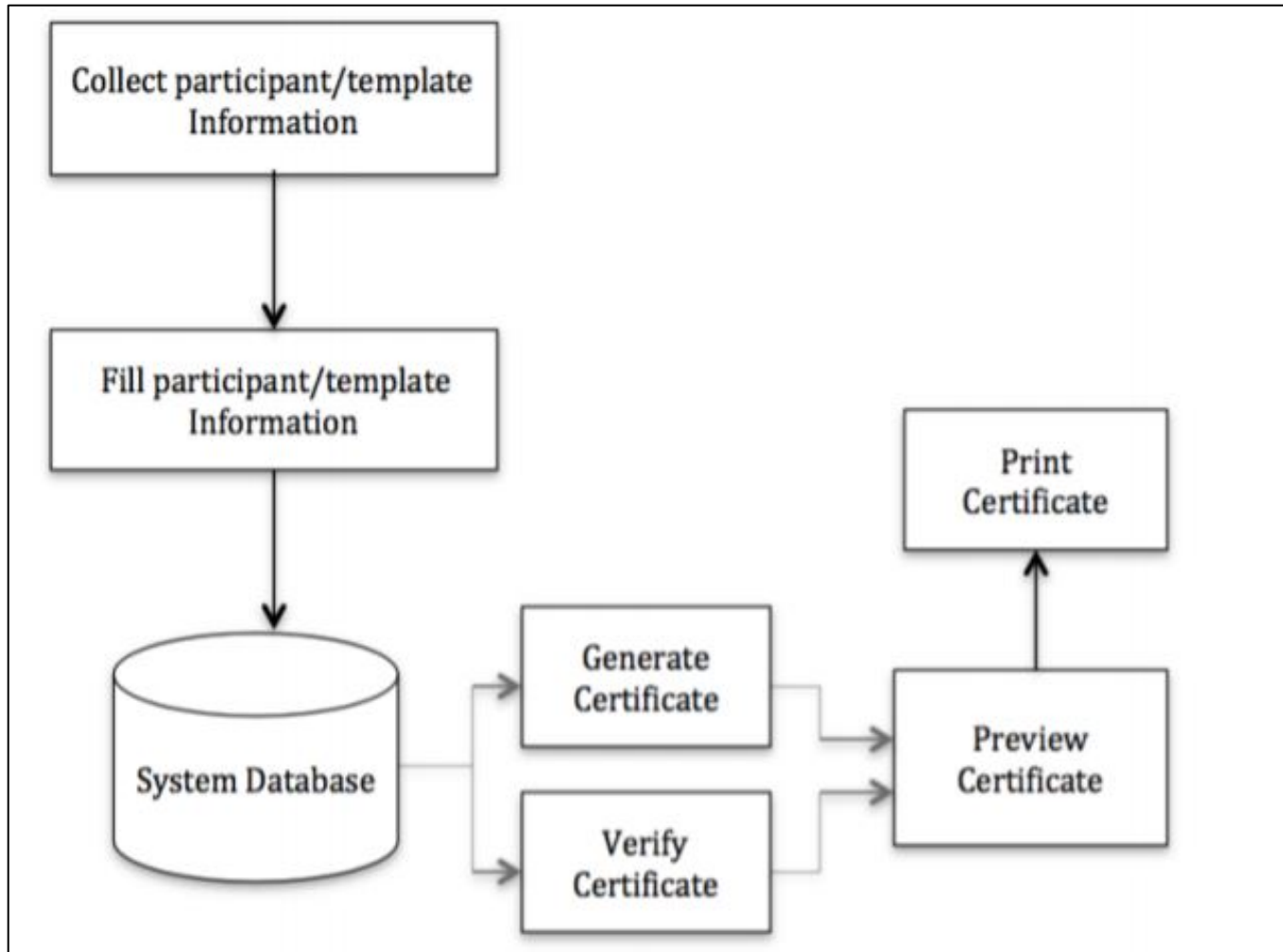
The system will authenticate user and admin before entering and no other data and certificates will be shared outside the user and admin profile ,each of them have unique token in firebase database.

5. COPY DESIGN CONTENTS OF YOUR TEAM HERE

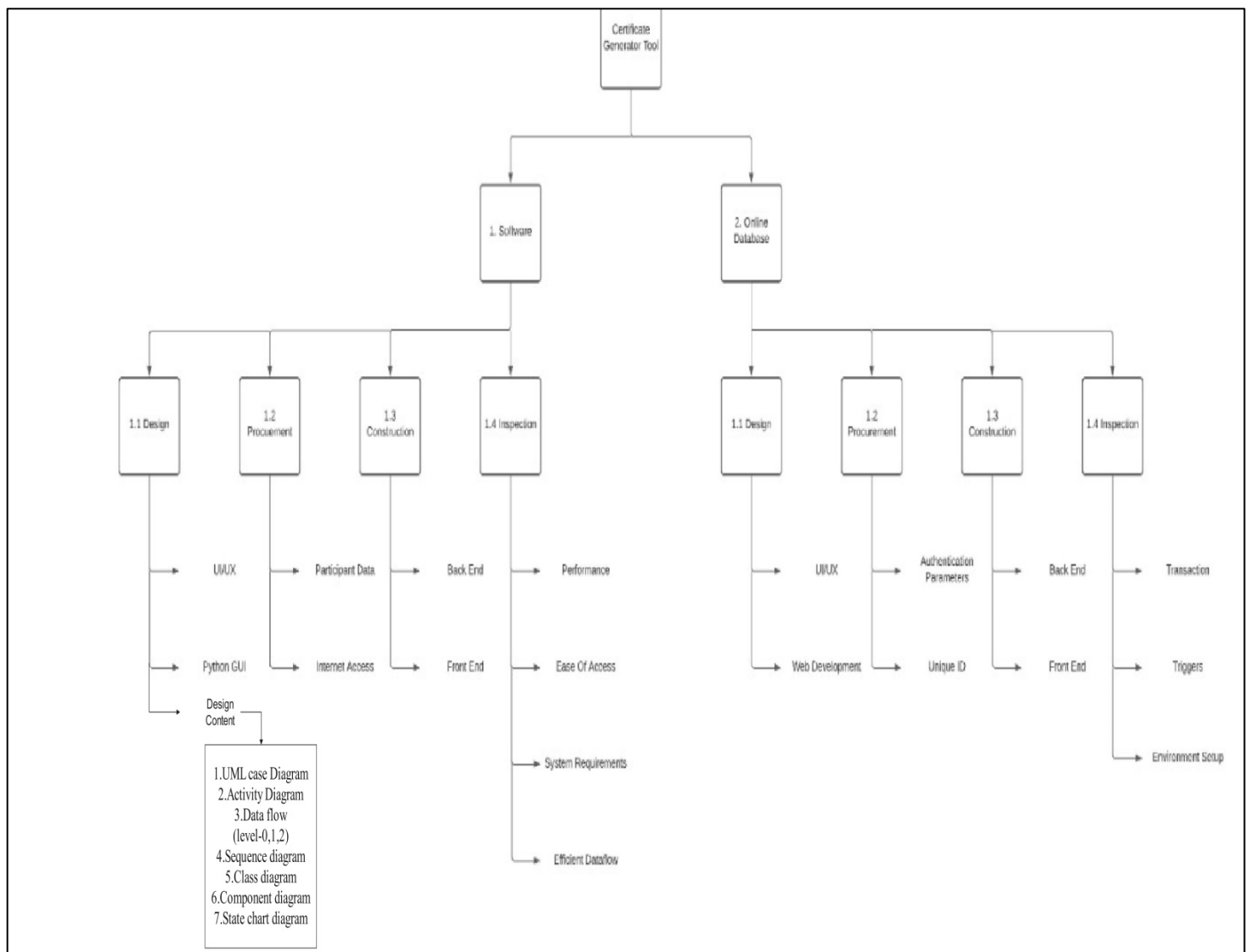
UML CASE DIAGRAM:



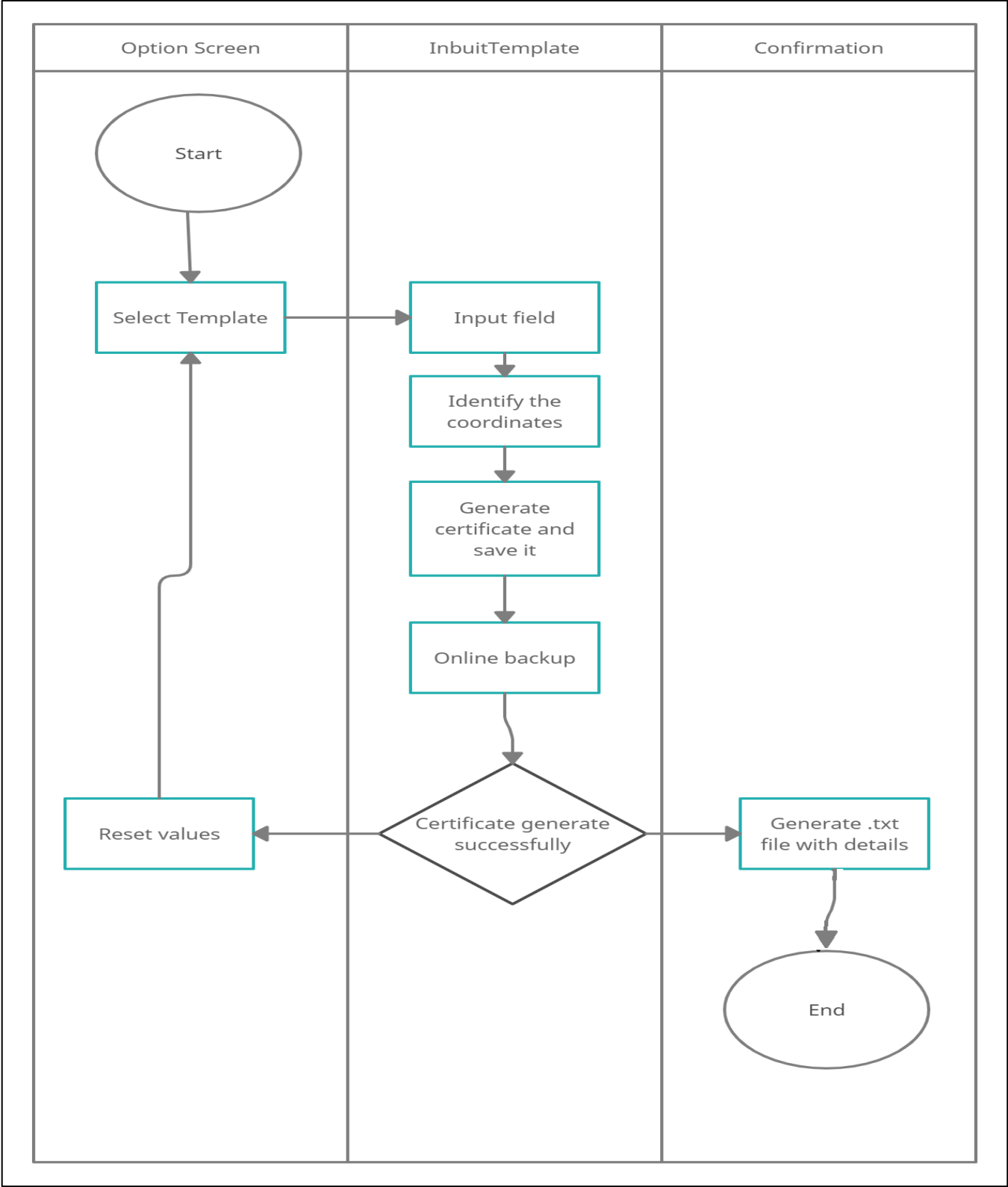
SYSTEM ARCHITECTURE:

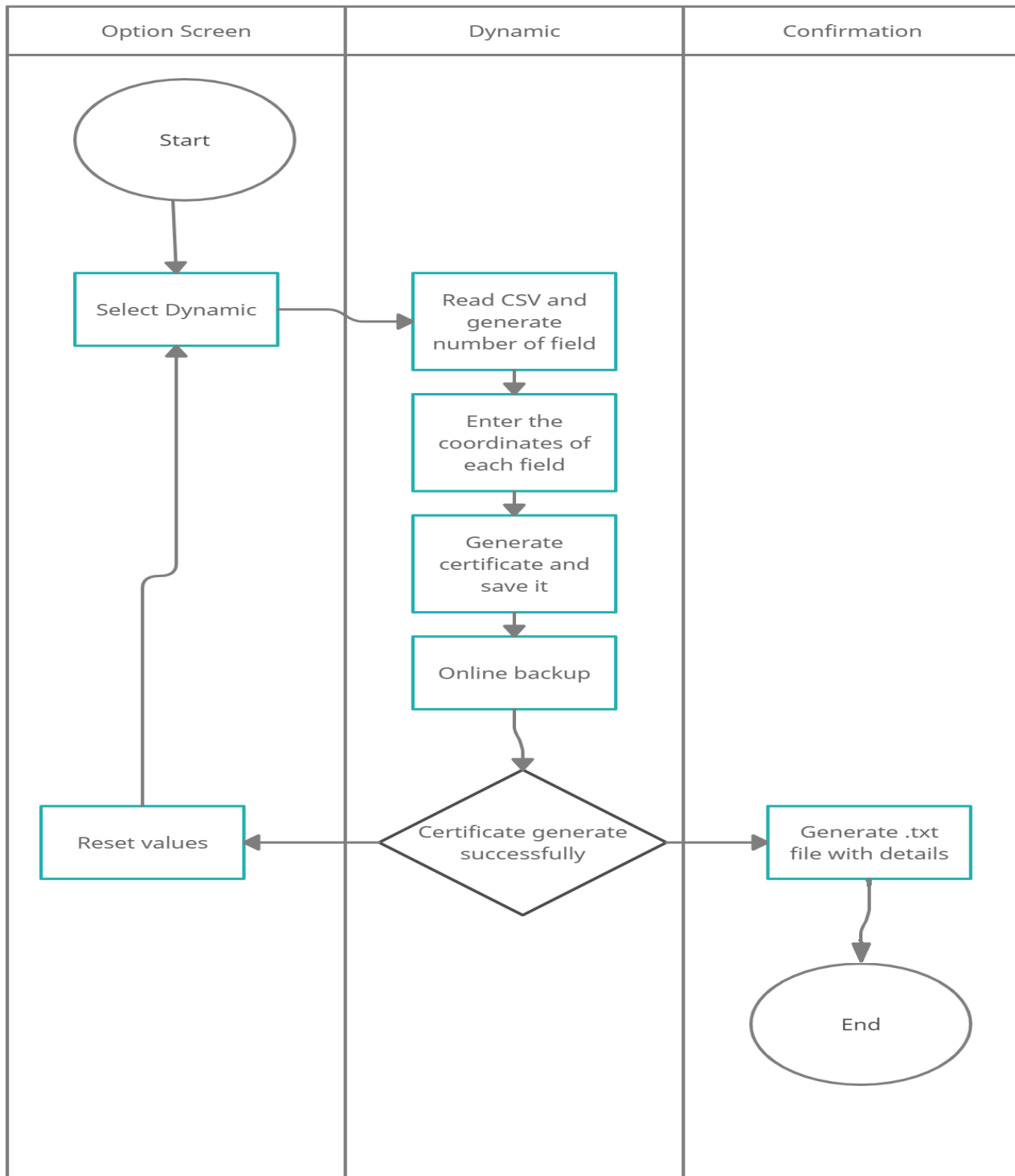


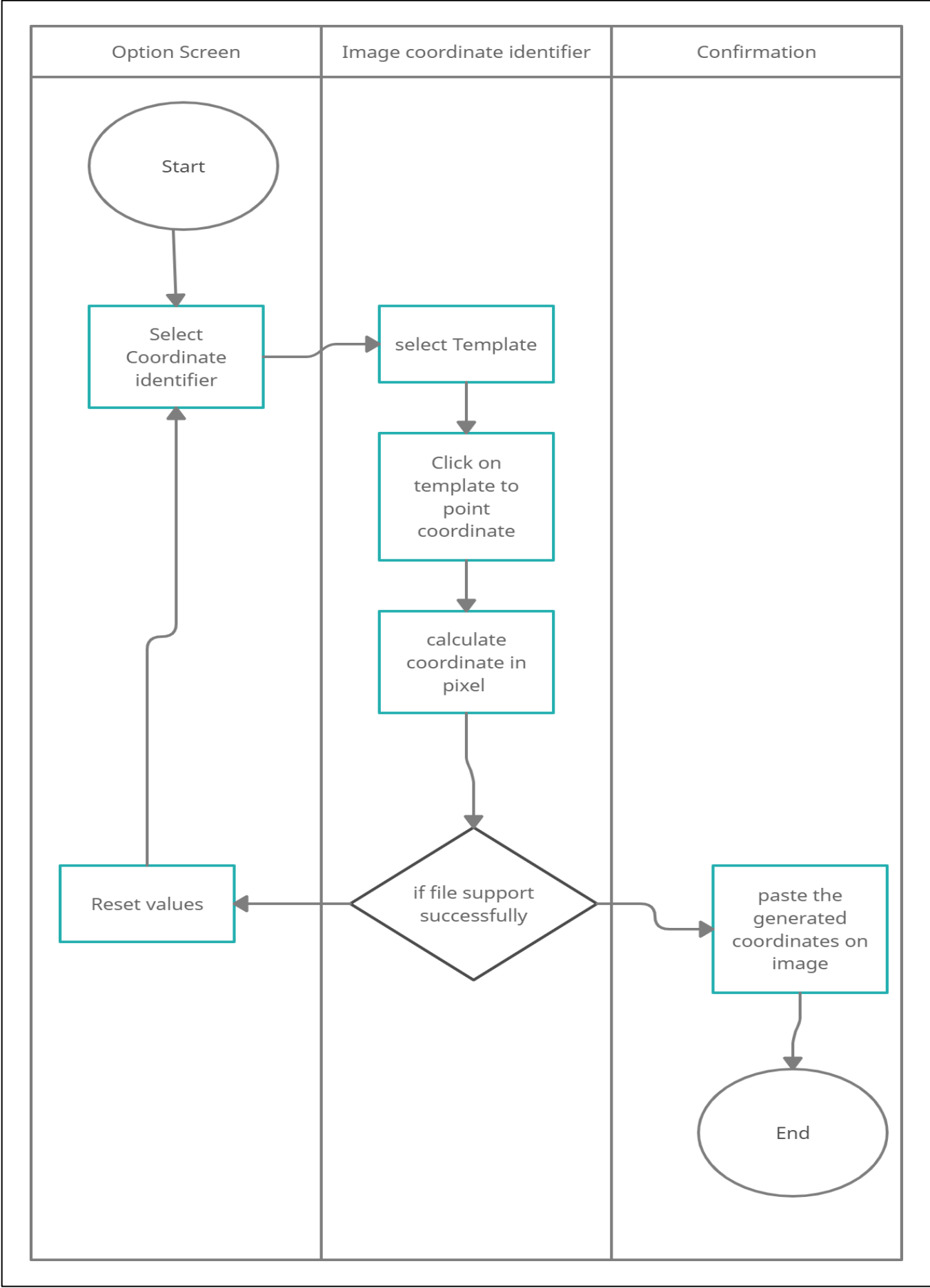
WORK BREAKDOWN STRUCTURE:

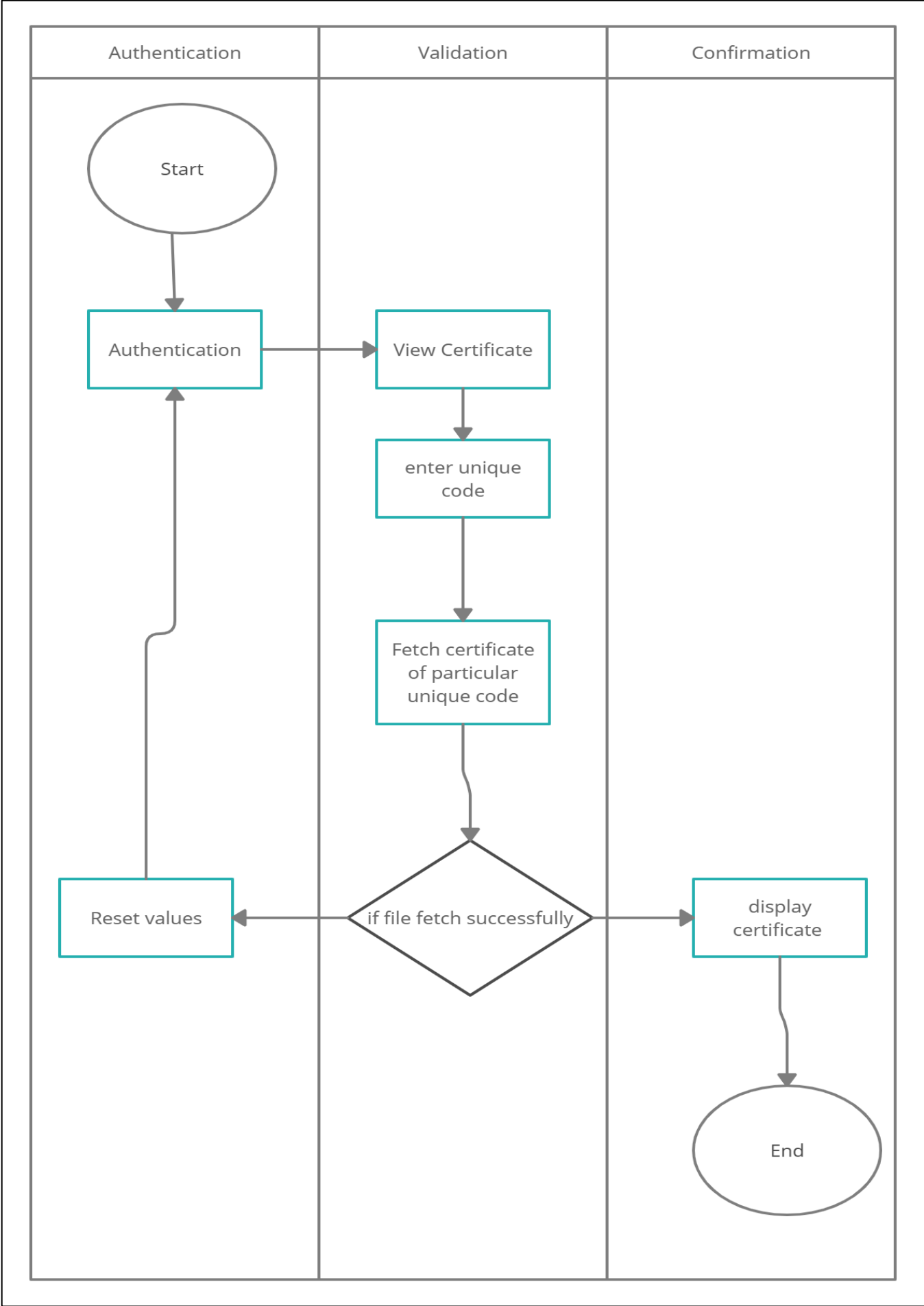


ACTIVITY DIAGRAM:



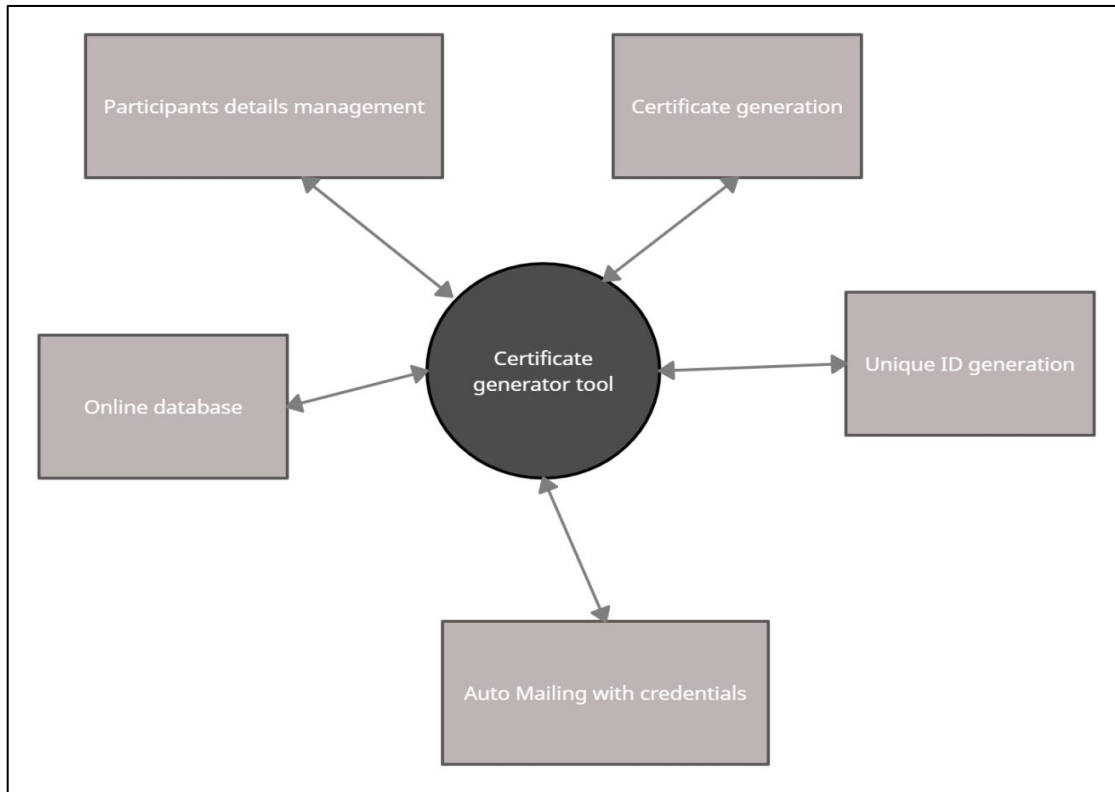




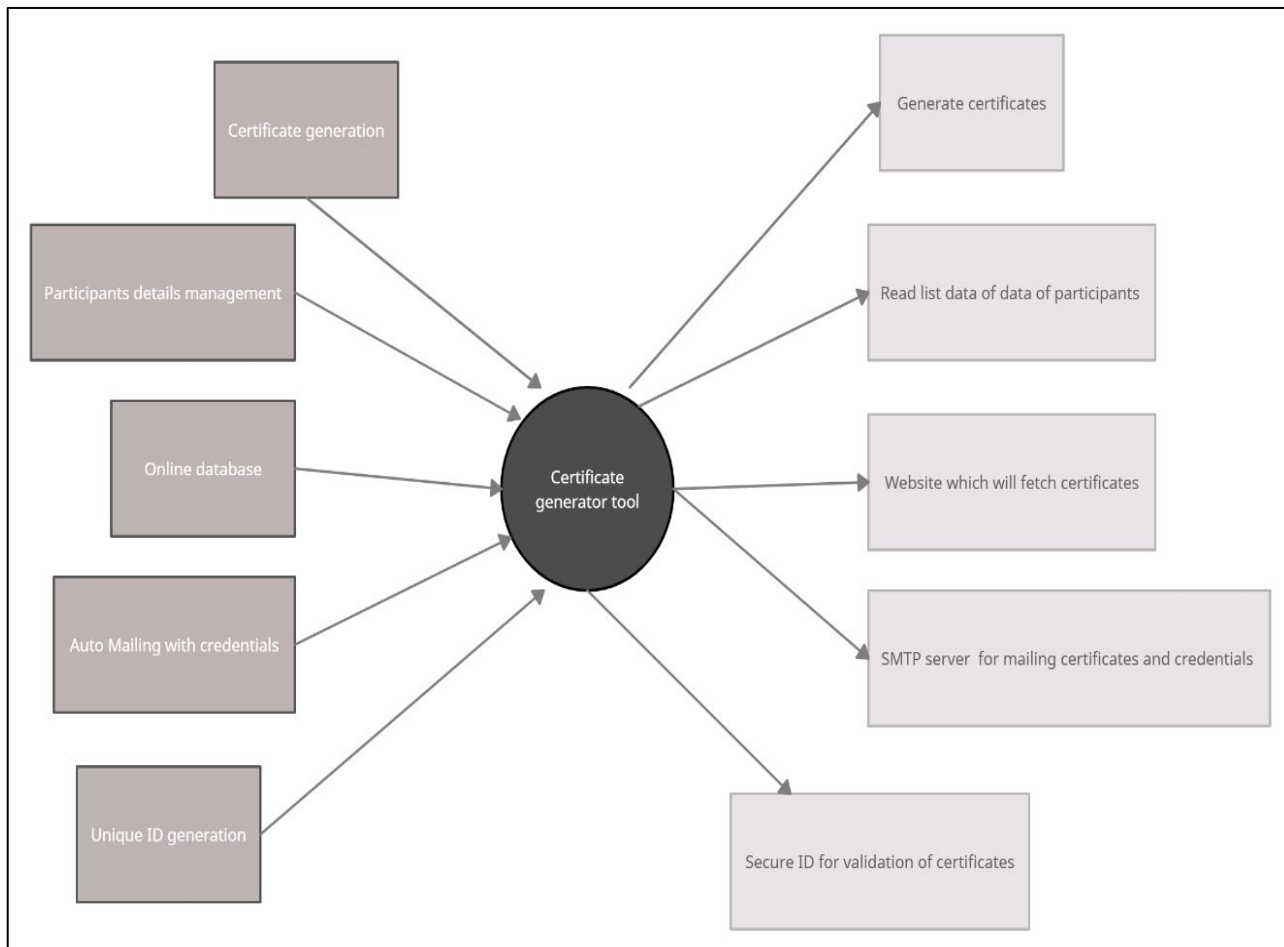


DATA FLOW

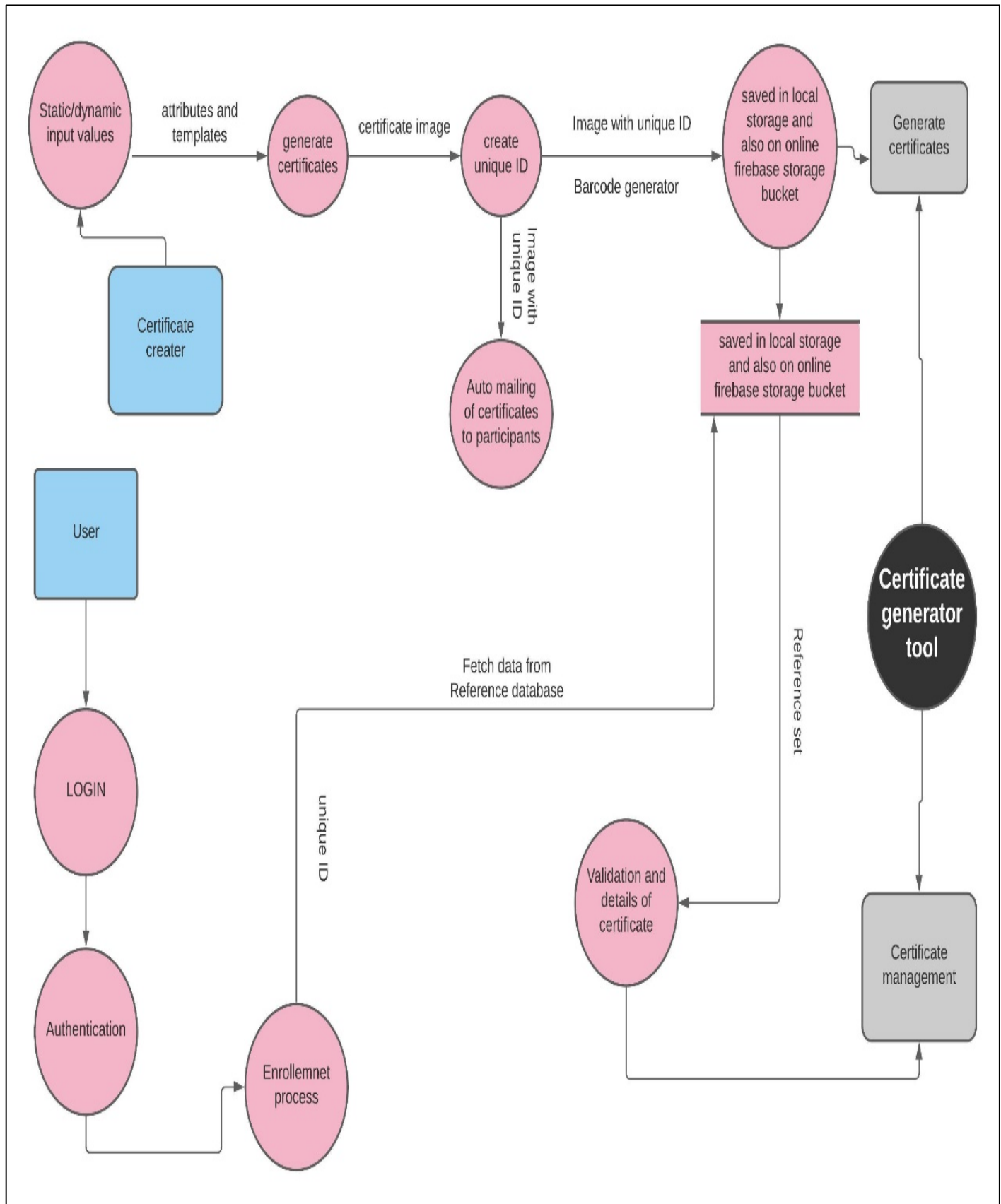
Level-0:



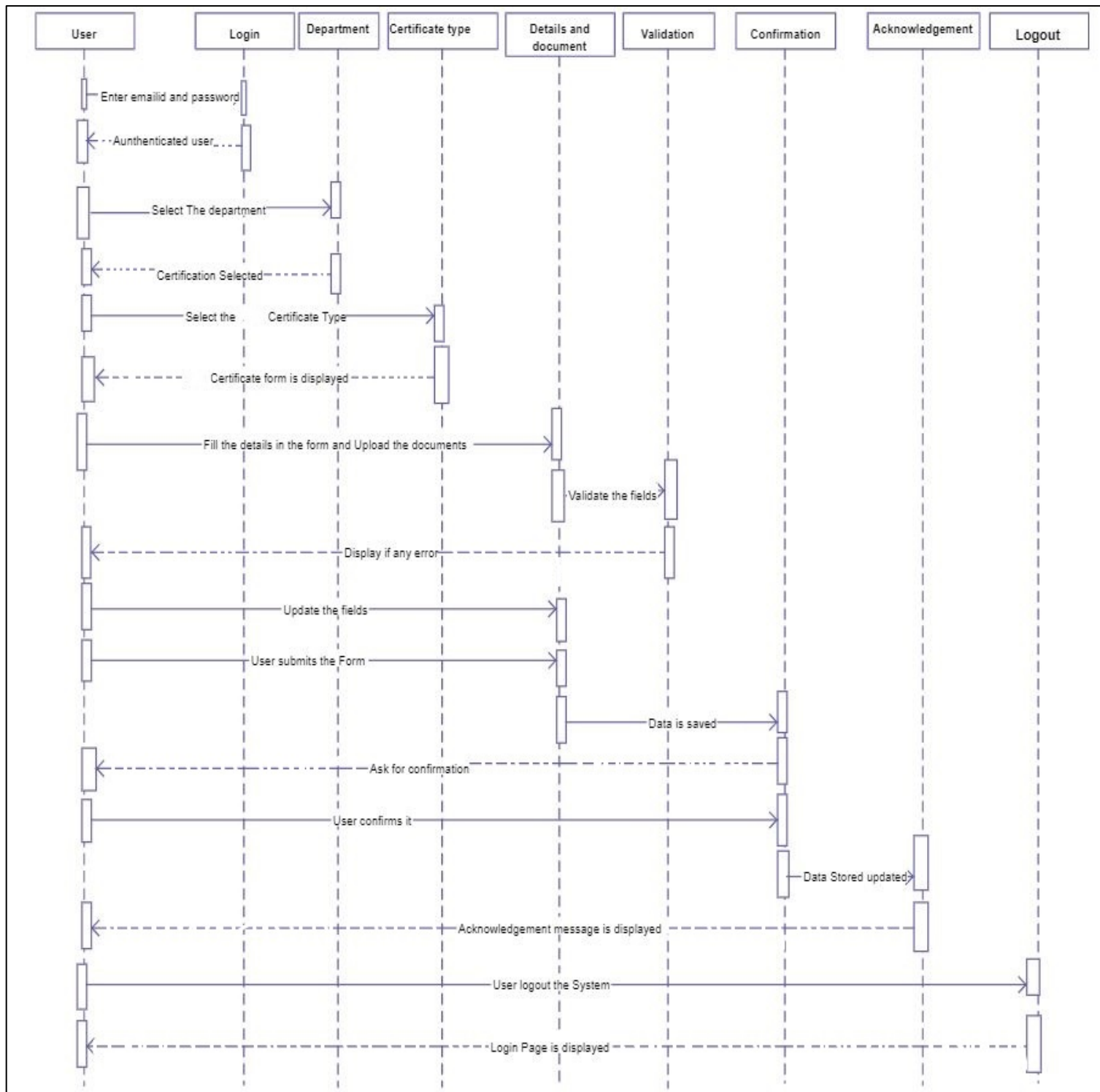
Level-1:



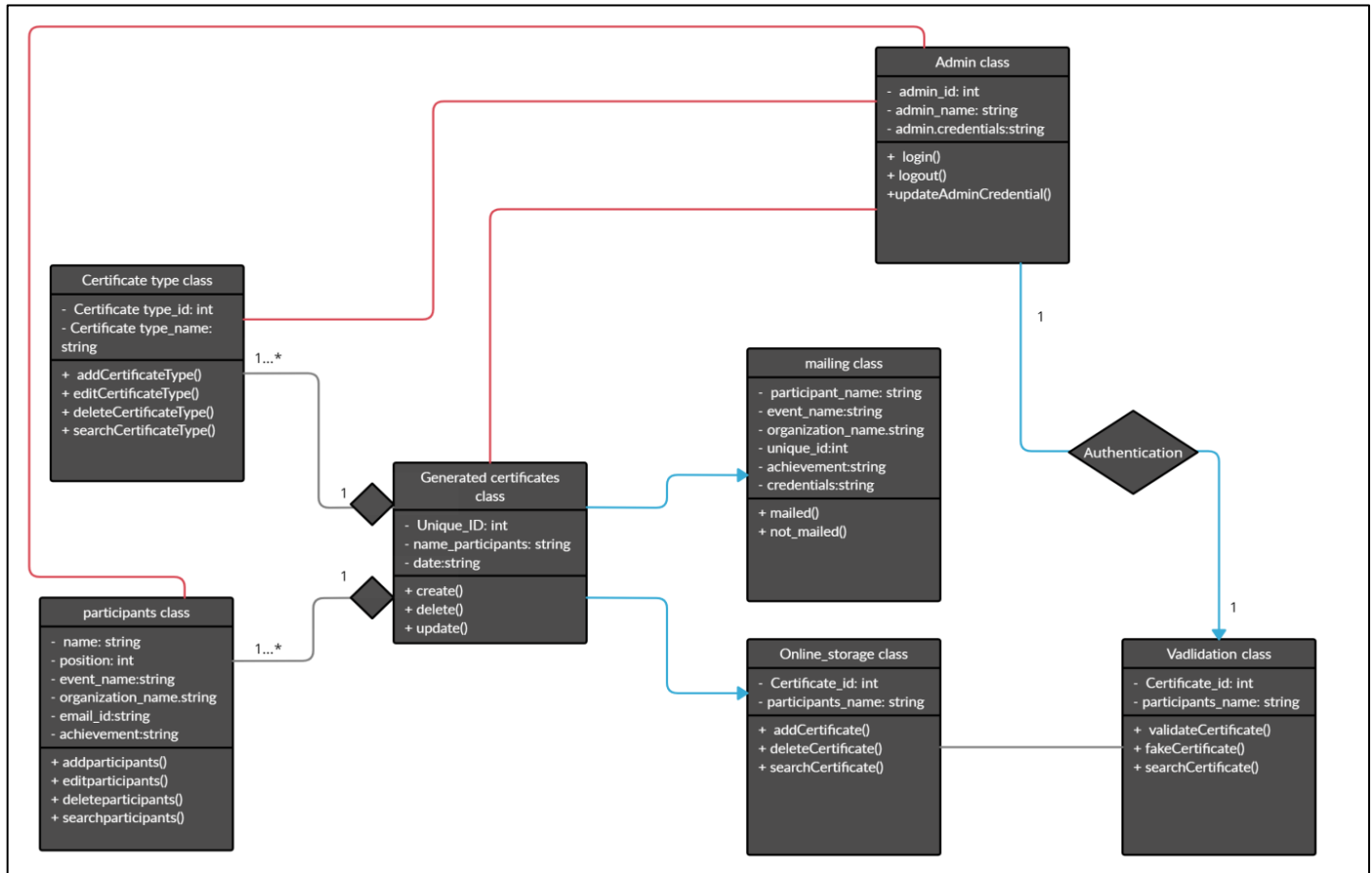
Level-2:



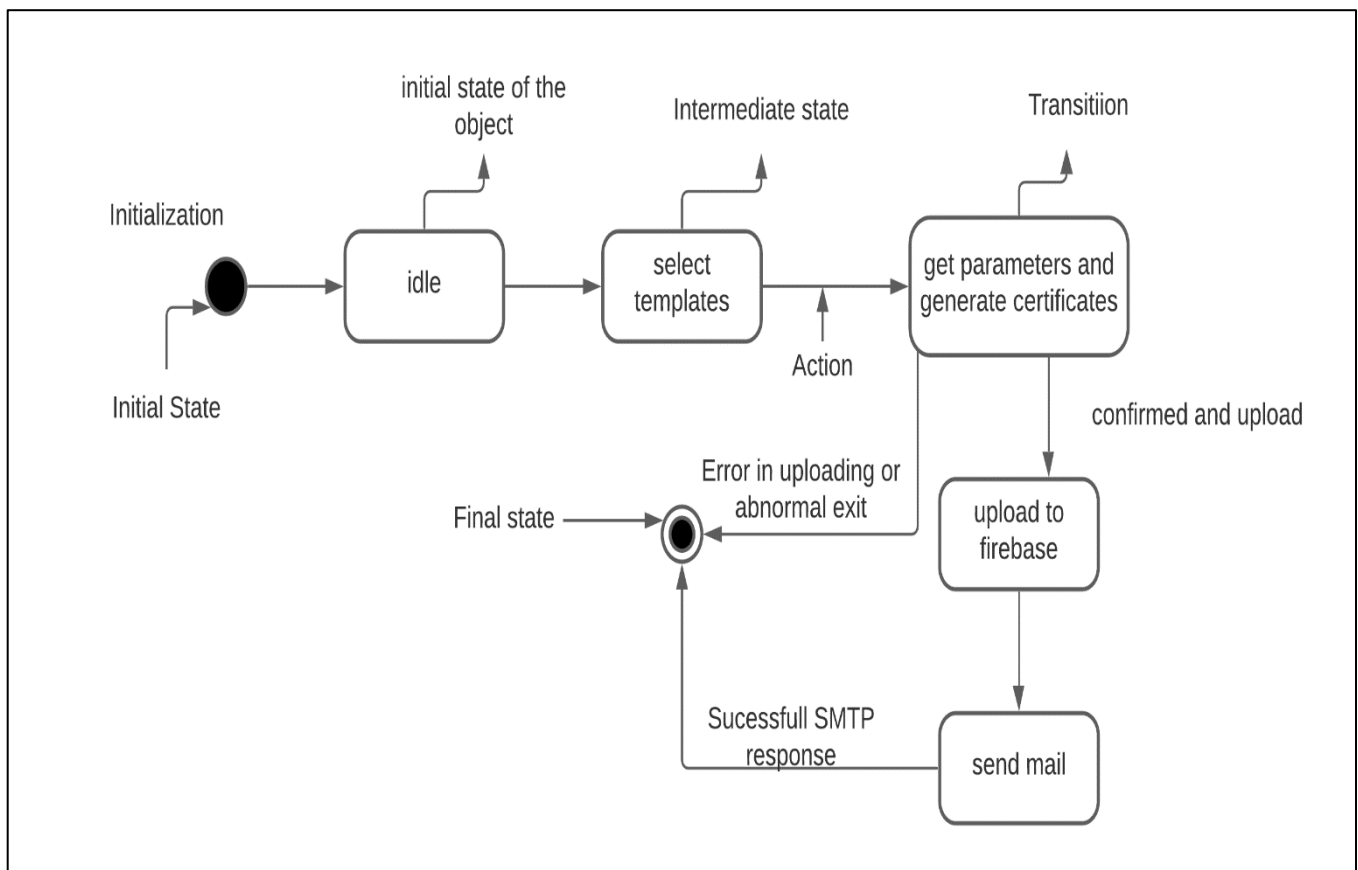
SEQUENCE DIAGRAM:



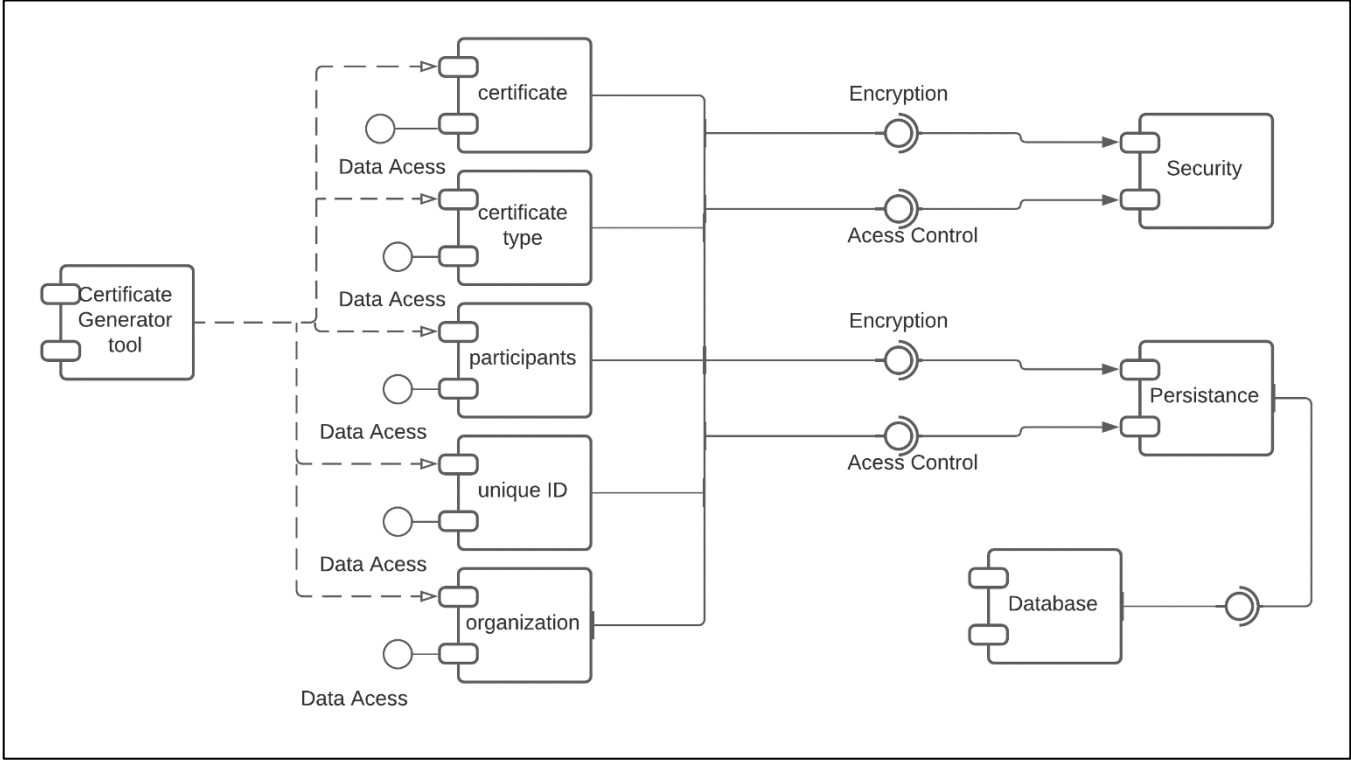
CLASS DIAGRAM:



STATE CHART DIAGRAM:

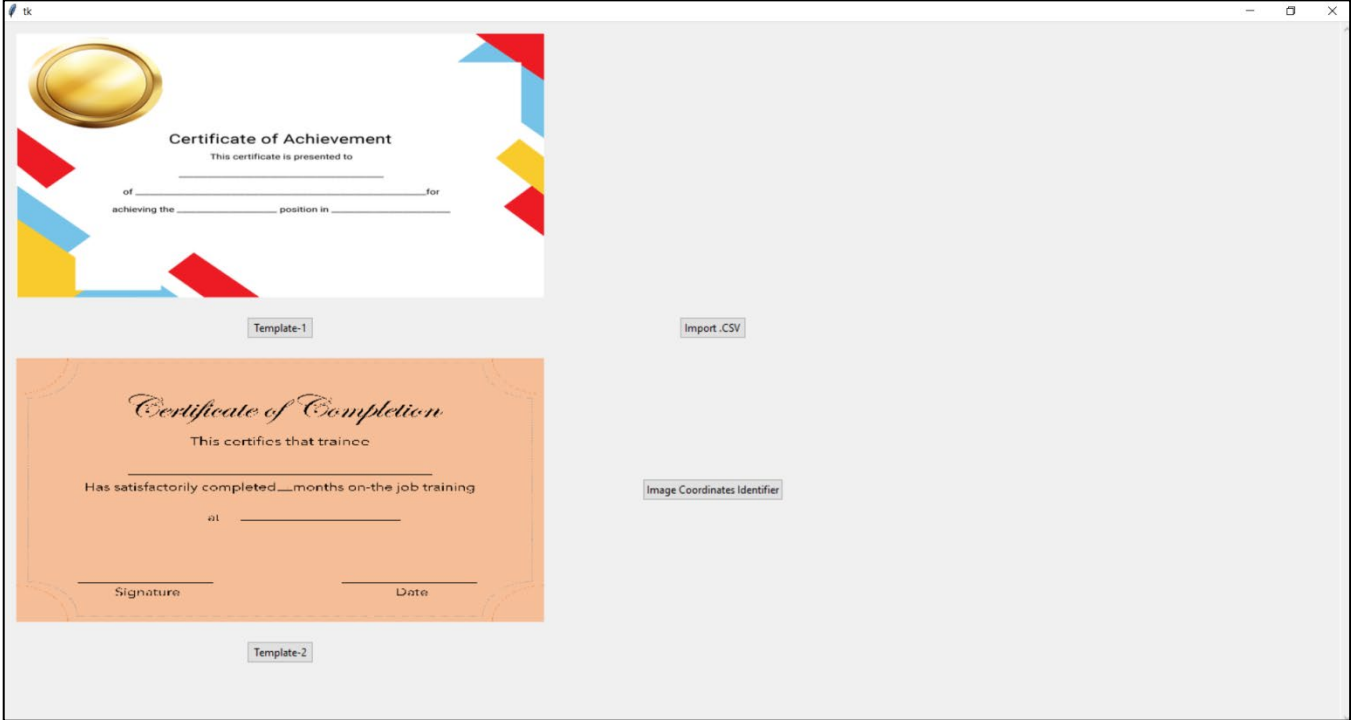


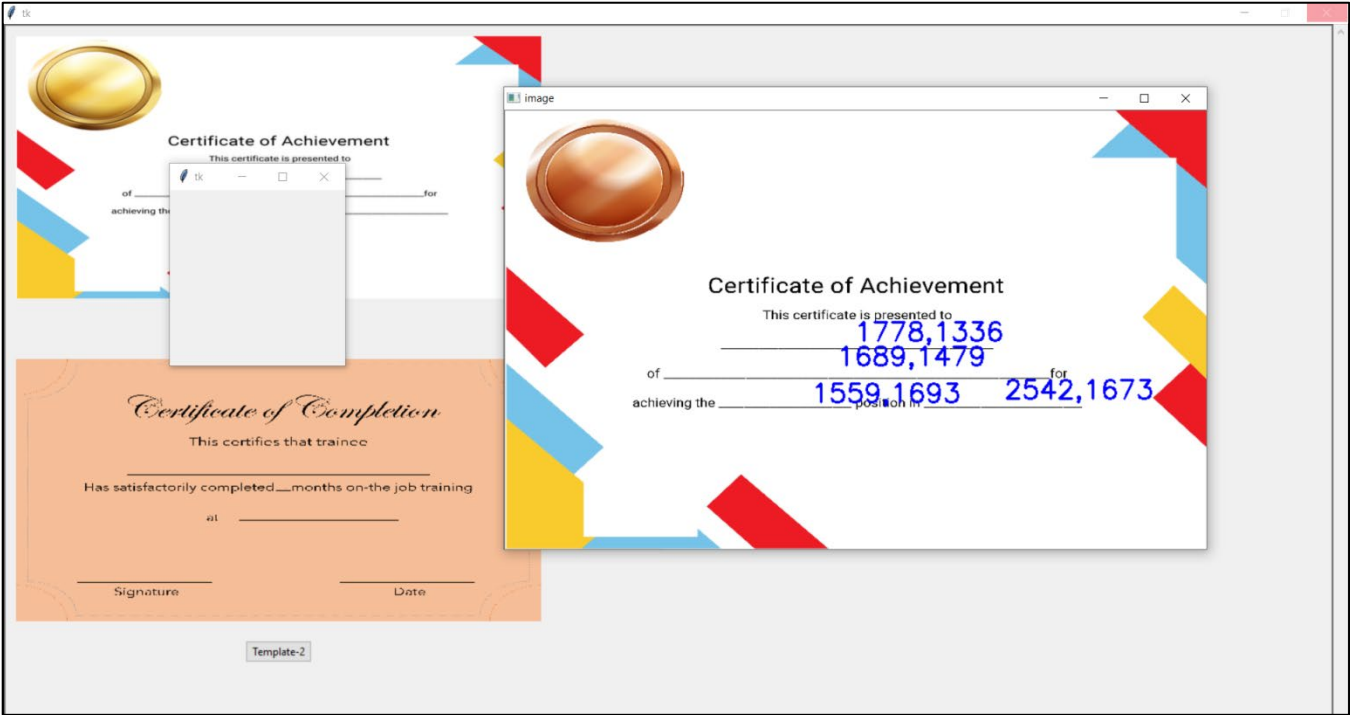
COMPONENT DIAGRAM:



DEPLOYMENT

Software:





tk

Name

College

Position

Event

Email Id

Entry Boxes

Destination folder


C:/Users/Rupin Patel/Desktop/Certificate generator tool/Dynamic.csv

C:/Users/Rupin Patel/Desktop/Certificate generator tool/bronze.jpg

Website

Certificate Generator Tool

How to use ??



Enter Your unique id -1868699909

Certificate of Completion

This certifies that trainee

Tracey Hewell

Has satisfactorily completed __months on-the job training

at Twitternation

Signature

5-Nov-21

Date

6. TEST CASE EXECUTED ON YOUR PROJECT

UNIT TESTING EXECUTED BY:-

Rupin Patel (19BCB0030)

Raghav Anand (17BCE0430)

Sameer Bishnoi (19BCB0029)

Yash Agarwal (19BCE0691)

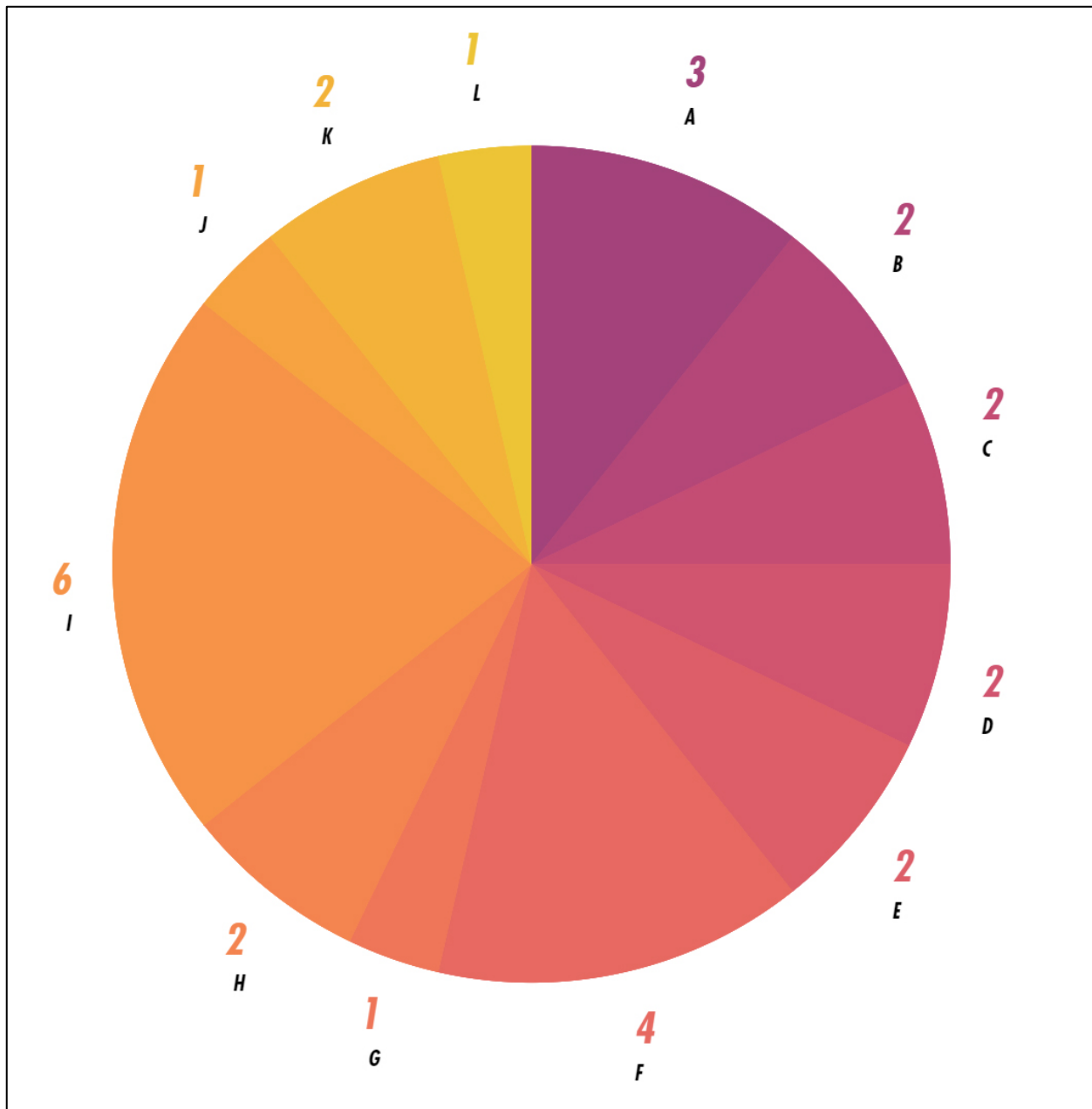
Test Cases#	Test case description	Test case Data	Result
1	while entering Manual data in Template-1 and 2 if any field get missing	Example in Template-1 Name: Rupin Patel College: VIT Position: First Event: Email:patelrupin63@gmail.com	The certificate will be generated without event name printed on it.
2	while entering Manual data in Template-1 and 2	Example in Template-1	Error in debug

	if email id is in wrong format					Name: Rupin Patel College: VIT Position: First Event:Football Email:patelrupin63gmailcom	window Something went wrong.										
3	In template-1 while Importing .csv file the excel format must be					The right format of entries and right format of emailId	Number of certificates generates/total entries without any error										
	Name	College	Position	Event	EmailId												
	XXX X	XXXX	XXXX	XXX X	XXXX												
4	In template-1 while Importing .csv file the excel format must be					If any field is empty such as name	Generation of certificate without any name print on it.										
	Name	College	Position	Event	EmailId												
		XXXX	XXXX	XXX X	XXXX												
5	In template-1 while Importing .csv file the excel format must be					Invalid email id in excel	Generation of certificate Mailing errors With error Something went wrong										
	Name	College	Position	Event	EmailId												
	XXX X	XXXX	XXXX	XXX X	XXXX												
6	Internet issue or No internet					no connection to firebase and SMTP	Generation of certificate But no online backup and mailing With error something went wrong										
7	While Generating certificate by any method if user try to generate or migrate to other page within App					Deadlock between process	App not responding										
8	While using template-2 compulsory select signature format of .png					Adding signature .jpg	Invalid Format										
9	while entering Manual data in Template-1 and 2 if all field are correctly with proper format of email id					Correct insertion of values	Generation of certificate without any errors										
10	<table><tr><td>Name</td><td>Duration</td><td>company</td><td>Date</td><td>Email id</td></tr><tr><td>XXXX</td><td>XXXX</td><td>XXXX</td><td>XXXX</td><td>XXXX</td></tr></table>					Name	Duration	company	Date	Email id	XXXX	XXXX	XXXX	XXXX	XXXX	Correct for .csv import in template	Generation of certificate without any errors
	Name	Duration	company	Date	Email id												
	XXXX	XXXX	XXXX	XXXX	XXXX												
Import .csv file in template-2																	
11	Using Image Identifier coordinate feature file format must be .jpg or .png					Correct file format	Gives correct coordinate seen on image as well as on debug screen										
12	Using Image Identifier coordinate feature file if other .jpg or .png is been selected					Incorrect file format	Give an error terminate process										
13	While using dynamic generation if user has enter all field correctly but not entered destination folder name					Destination is unknown	Error something went wrong										
14	While using dynamic generation if user has enter					Destination:<correct>	Error										

	destination name next step for importing .csv		Import .csv:<incorrect>	something went wrong			
15	After importing .csv entered correct coordinates in pixel with comma seprated <width , height>		Destination:<correct> Import .csv:<correct> Co.ord:<Incorrect format>	Error something went wrong			
16	After importing .csv import your template in .jpg format		Destination:<correct> Import .csv:<correct> Co.ord: <correct> Template:<incorrect>	Error something went wrong			
17	Inserting all values correctly		Destination:<correct> Import .csv:<correct> Co.ord: <correct> Template:<correct>	Generation of certificate without any errors			
18	While importing .csv in dynamic option <table><tr><td>Field-1</td><td>Field-2</td><td>Field-3.....</td></tr></table>		Field-1	Field-2	Field-3.....	Add only those column in .csv which must be printed on certificate	Automatically create number of coordinate field which are required
Field-1	Field-2	Field-3.....					
19	The coordinate boxes will be correspond to field		If number entries and less or more corresponding to .csv no. of columns	Internal error or user must check .csv again			
20	Website Authentication Simple email Id and password or sign in with google		Valid and correct inputs	Migrate to next page			
21	Website Authentication signup		New users must signup before sign in	Add user to firebase authentication			
22	New user to website		Try to sign in without signup	Error in authentication			
23	Validation of certificate		Enter correct unique Id	Fetch photo from firebase and display			
24	Invalidation of certificate		Enter incorrect unique Id or incorrect format of unique Id	Error incorrect entry			
25	Invalid inputs while authentication		Invalid user entries	Error in authentication			
26	While using template-2 compulsory select signature format of .png		Adding signature .png	Valid format			

Outputs possibilities:

1. No certificate generation-A
2. Successful Certificate generation but no mailing-B
3. Successful Certificate generation but no online backup and mailing-C
4. Authentication error-D
5. Certificate Validation error-E
6. Invalid format error-F
7. Not responding error-G
8. Certificate with empty fields-H
9. Successful generation of certificate without any error and with all file format correctly -I
10. Successful validation of certificate on website without any error-J
11. Successful authentication-K
12. Correct coordinates in coordinates identifier function-L



USABILITY TESTING – CERTIFICATE GENERATOR TOOL

The usability evaluation of the **Certificate Generator Tool** was conducted by **Group Number: 6** in **Google Meet** on **27.05.2021**.

During the usability evaluation, **four** participants, matching the user profile, were asked to spend one hour with the site. During this hour, participants:

- Completed a user background questionnaire
- Answered questions about initial site impressions
- Performed real-world tasks on the site while thinking aloud
- Answered questions about their overall satisfaction

Participants, having the following profile characteristics, evaluated **Certificate Generator Tool**

7. TEAM NUMBER: 6

□ Audience Type

Participant	Participant Name
1	Arnav Sunil Deshpande
2	Karandeep Singh
3	Sanskrita Saha
4	Yash
Total Participants – 4	

□ Age:

Category	Number of Participants
18-25	4
26-39	0
40-59	0
60-74	0

□ Gender:

Gender	Number of Participants
Female	1
Male	3

Following is a summary of the participants' computing environment:

URL of tested website:	https://cert-43ffe.web.app/
Computer platforms:	Microsoft Windows
Browser tested:	Google Chrome

Screen resolution:	1600 x 1200 pixels
Operating system:	Windows X
Connection speed:	1 MBPS

The following tasks were identified from user data collection efforts and assistance from Team
Number: 4.

#	Task
1	Verify if the user as a student will be able to login with a valid email and valid password.
2	Verify if the user as a student won't be able to login with an invalid email and invalid password.
3	Verify If the user is able to sign up without facing any issue or not?
4	Verify if the user is able to see the Home page or not?.
5	Verify if the user is able to see the View Certificate Option or not?.
6	Verify if the user is able to view the download option or not?.
7	Verify if the user is able to download software without any issue or not?.
8	Verify if the user is able to install software without any bugs or not?.
9	Verify if the user is able to see a folder where the path of installation was given.
10	Verify if the user can see the main.exe among the packages in the folder.
11	Verify if the user after running the main file is able to see Graphical User Interface.
12	Verify if the user is able to see all the four buttons with 2 inbuilt templates.
13	Verify all the fields in the form are functioning well or not for the generation of one certificate?
14	Verify if the user is able to import the csv file from his/her computer.
15	Verify if the user is able to see the debug frame or not?.
16	Verify if the user is able to see the progress of generating certificates.
17	Verify if the user is able to see the file named result template in the folder in which information of all generated certificates is stored along with their hash value.
18	Verify if the user is able to see the folder named teams in which the certificates will be present and all their credentials will be correct.

19	Verify if the certificates with their hash code are mailed to the registered id.
20	Verify that the user is able to download the certificate after entering hash code.
21	Verify whether the user is able to import signatures or not?.
22	Verify whether the user is able to design his/her own template by clicking on the import.csv button.
23	Verify whether the user is able to see different coordinates for different locations present on the certificate template after using image coordinate identifier.
24	Verify if the user is able to generate certificates by entering correct coordinates and even import images or not?.
25	Verify if the user is able to logout from the system.

#	Participant1 (19BCB0046 , YASH) Observer (19BCB0030 ,RUPIN PATIL)	Passed or failed	Time taken (s)	Interface is good / bad / ok
1	Verify if the user as a student will be able to login with a valid email and valid password.	Pass	6	Ok
2	Verify if the user as a student won't be able to login with an invalid username and invalid password.	Pass	5	Ok
3	Verify If the user is able to sign up without facing any issue or not?	Pass	5	Good
4	Verify if the user is able to see the Home page or not?	Pass	4	Good
5	Verify if the user is able to see the View Certificate Option or not?	Pass	4	Good
6	Verify if the user is able to view the download option or not?	Pass	4	Good
7	Verify if the user is able to download software without any issue or not?	Pass	20	Good
8	Verify if the user is able to install software without any bugs or not?	Pass	150	Ok
9	Verify if the user is able to see a folder where the path of installation was given.	Pass	5	Ok
10	Verify if the user can see the main.exe among the packages in the folder.	Fail	4	Bad
11	Verify if the user after running the main file is able to see Graphical User Interface.	Fail	5	Bad
12	Verify if the user is able to see all the four buttons with 2 inbuilt templates.	Pass	5	Good
13	Verify all the fields in the form are functioning well or not for the generation of one certificate?	Pass	10	Good
14	Verify if the user is able to import the csv file from his/her computer	Fail	10	Ok
15	Verify if the user is able to see the debug frame or not?	Pass	5	Good
16	Verify if the user is able to see the progress of generating certificates.	Fail	40	Ok
17	Verify if the user is able to see the file named result template in the folder in which information	Pass	5	Good

18	of all generated certificates is stored along with their hash value.			
	Verify if the user is able to see the folder named teams in which the certificates will be present and all their credentials will be correct.	Fail	5	Bad
19	Verify if the certificates with their hash code are mailed to the registered id.	Paas	10	Good
20	Verify that the user is able to download the certificate after entering hash code.	Paas	30	Ok
21	Verify whether the user is able to import signatures or not?.	Fail	10	Bad
22	Verify whether the user is able to design his/her own template by clicking on the import.csv button.	Fail	30	Ok
23	Verify whether the user is able to see different coordinates for different locations present on the certificate template after using image coordinate identifier.	Paas	4	Good
24	Verify if the user is able to generate certificates by entering correct coordinates and even import images or not?.	Paas	10	Good
25	Verify if the user is able to logout from the system.	Paas	2	Good

#	Participant 2 (19BCB0055 , KARANDEEP SINGH) Observer (19BCB0029 ,SAMEER BISHNOI)	Passed or failed	Time taken	Interface is good / bad / ok
1	Verify if the user as a student will be able to login with a valid email and valid password.	Pass	10	Good
2	Verify if the user as a student won't be able to login with an invalid username and invalid password.	Pass	3	Ok
3	Verify If the user is able to sign up without facing any issue or not?	Pass	6	Good
4	Verify if the user is able to see the Home page or not?	Pass	4	Good
5	Verify if the user is able to see the View Certificate Option or not?	Pass	8	Good
6	Verify if the user is able to view the download option or not?	Pass	4	Good
7	Verify if the user is able to download software without any issue or not?	Pass	35	Good
8	Verify if the user is able to install software without any bugs or not?	Pass	80	Ok
9	Verify if the user is able to see a folder where the path of installation was given.	Fail	3	Bad
10	Verify if the user can see the main.exe among the packages in the folder.	Fail	7	Bad
11	Verify if the user after running the main file is	Pass	5	Ok

	able to see Graphical User Interface.			
12	Verify if the user is able to see all the four buttons with 2 inbuilt templates.	Pass	5	Good
13	Verify all the fields in the form are functioning well or not for the generation of one certificate?	Pass	10	Good
14	Verify if the user is able to import the csv file from his/her computer	Fail	10	Ok
15	Verify if the user is able to see the debug frame or not?	Pass	5	Good
16	Verify if the user is able to see the progress of generating certificates.	Fail	25	Bad
17	Verify if the user is able to see the file named result template in the folder in which information of all generated certificates is stored along with their hash value.	Pass	6	Ok
18	Verify if the user is able to see the folder named teams in which the certificates will be present and all their credentials will be correct.	Fail	8	Bad
19	Verify if the certificates with their hash code are mailed to the registered id.	Pass	10	Good
20	Verify that the user is able to download the certificate after entering hash code.	Pass	30	Ok
21	Verify whether the user is able to import signatures or not?.	Fail	10	Bad
22	Verify whether the user is able to design his/her own template by clicking on the import.csv button.	Fail	25	Bad
23	Verify whether the user is able to see different coordinates for different locations present on the certificate template after using image coordinate identifier.	Paas	4	Good
24	Verify if the user is able to generate certificates by entering correct coordinates and even import images or not?.	Pass	14	Good
25	Verify if the user is able to logout from the system.	Pass	5	Good

#	Participant3 (19BCB0065 , DESHPANDE ARNAV SUNIL) Observer (19BCE0691 , YASH AGGRAWAL)	Passed or failed	Time taken	Interface is good / bad / ok
1	Verify if the user as a student will be able to login with a valid email and valid password.	Pass	6	Ok
2	Verify if the user as a student won't be able to login with an invalid username and invalid password.	Pass	3	Good
3	Verify If the user is able to sign up without facing any issue or not?	Pass	5	Good
4	Verify if the user is able to see the Home page or not?	Pass	4	Ok

5	Verify if the user is able to see the View Certificate Option or not?	Pass	4	Ok
6	Verify if the user is able to view the download option or not?	Pass	5	Good
7	Verify if the user is able to download software without any issue or not?	Pass	20	Good
8	Verify if the user is able to install software without any bugs or not?	Pass	100	Ok
9	Verify if the user is able to see a folder where the path of installation was given.	Pass	5	Good
10	Verify if the user can see the main.exe among the packages in the folder.	Fail	4	Bad
11	Verify if the user after running the main file is able to see Graphical User Interface.	Pass	4	Ok
12	Verify if the user is able to see all the four buttons with 2 inbuilt templates.	Fail	5	Bad
13	Verify all the fields in the form are functioning well or not for the generation of one certificate?	Pass	16	Good
14	Verify if the user is able to import the csv file from his/her computer	Fail	10	Ok
15	Verify if the user is able to see the debug frame or not?	Pass	5	Good
16	Verify if the user is able to see the progress of generating certificates.	Fail	40	Ok
17	Verify if the user is able to see the file named result template in the folder in which information of all generated certificates is stored along with their hash value.	Pass	5	Good
18	Verify if the user is able to see the folder named teams in which the certificates will be present and all their credentials will be correct.	Fail	5	Bad
19	Verify if the certificates with their hash code are mailed to the registered id.	Pass	30	Ok
20	Verify that the user is able to download the certificate after entering hash code.	Pass	20	Ok
21	Verify whether the user is able to import signatures or not?.	Fail	10	Bad

22	Verify whether the user is able to design his/her own template by clicking on the import.csv button.	Pass	30	Ok
23	Verify whether the user is able to see different coordinates for different locations present on the certificate template after using image coordinate identifier.	Fail	4	Bad
24	Verify if the user is able to generate certificates by entering correct coordinates and even import images or not?.	Pass	10	Ok
25	Verify if the user is able to logout from the system.	Pass	2	Good

#	Participant4 (19BCB0059 , SANSKRITA SAHA) Observer (17BCE0430 , RAGHAV ANAND)	Passed or failed	Time taken	Interface is good / bad / ok
1	Verify if the user as a student will be able to login with a valid email and valid password.	Pass	4	Good
2	Verify if the user as a student won't be able to login with an invalid username and invalid password.	Pass	3	Ok
3	Verify If the user is able to sign up without facing any issue or not?	Pass	6	Good
4	Verify if the user is able to see the Home page or not?	Fail	3	Ok
5	Verify if the user is able to see the View Certificate Option or not?	Pass	5	Good
6	Verify if the user is able to view the download option or not?	Pass	4	Good
7	Verify if the user is able to download software without any issue or not?	Pass	25	Good
8	Verify if the user is able to install software without any bugs or not?	Pass	152	Ok
9	Verify if the user is able to see a folder where the path of installation was given.	Fail	8	Ok
10	Verify if the user can see the main.exe among the packages in the folder.	Fail	5	Bad
11	Verify if the user after running the main file is able to see Graphical User Interface.	Paas	9	Good
12	Verify if the user is able to see all the four buttons with 2 inbuilt templates.	Pass	8	Bad
13	Verify all the fields in the form are functioning well or not for the generation of one certificate?	Pass	12	Good
14	Verify if the user is able to import the csv file from	Paas	13	Ok

	his/her computer			
15	Verify if the user is able to see the debug frame or not?	Pass	7	Good
16	Verify if the user is able to see the progress of generating certificates.	Fail	43	Ok
17	Verify if the user is able to see the file named result template in the folder in which information of all generated certificates is stored along with their hash value.	Pass	6	Good
18	Verify if the user is able to see the folder named teams in which the certificates will be present and all their credentials will be correct.	Paas	3	Good
19	Verify if the certificates with their hash code are mailed to the registered id.	Fail	13	Bad
20	Verify that the user is able to download the certificate after entering hash code.	Paas	32	Ok
21	Verify whether the user is able to import signatures or not?.	Fail	8	Bad
22	Verify whether the user is able to design his/her own template by clicking on the import.csv button.	Fail	31	Ok
23	Verify whether the user is able to see different coordinates for different locations present on the certificate template after using image coordinate identifier.	Paas	4	Ok
24	Verify if the user is able to generate certificates by entering correct coordinates and even import images or not?.	Fail	11	Bad
25	Verify if the user is able to logout from the system.	Paas	5	Good

EXIT QUESTIONS/USER IMPRESSIONS

At the end of each session, we asked participants these questions:

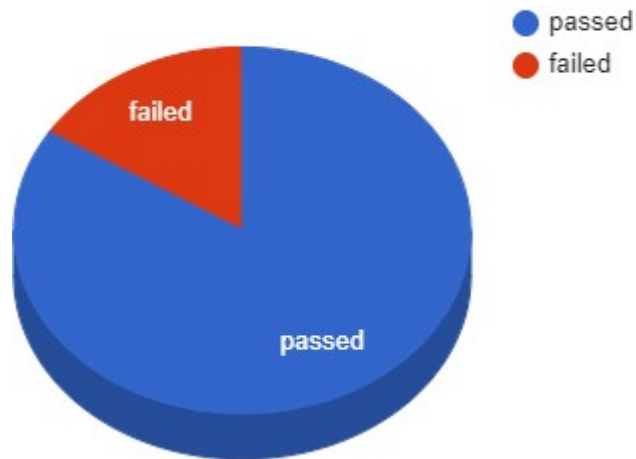
- What is your overall impression of the site?
- What is your impression of the search capability?
- Do you feel this site is current? Why?
- What did you like best about the site?
- What did you like least about the site?
- If you were the website developer, what would be the first thing you would do to improve the website?
- Is there anything that you feel is missing on this site? (Probe: content or site features/functions)
- If you were to describe this site to a colleague in a sentence or two, what would you say?
- Do you have any other final comments or questions?

Questions	Participant 1: Yash	Participant 2: Karandeep Singh Chandi	Participant 3: Arnav Deshpande	Participant 4: Sanskrita Saha
What is your overall impression of the site?	Good	Best	Average	Nice
What is your impression of the search capability?	Average	Good	Best	Good
Do you feel this site is current? Why?	Yes, it is storing the answers in real time.	Yes, because it takes Immediate inputs and outputs.	Yes, it takes the input and updates it immediately.	Yes, the website as a whole updates immediately.
What did you like best about the site?	Certificate Generation By Use Of Coordinates.	Database handling	Preparing Quiz	Time correctness
What did you like least about the site?	GUI representation.	Feedback form	User Buttons	Some interface
If you were the website developer, what would be the first thing you would do to improve the website?	Frontend in some areas.	As a whole, it is good.	Representation of Buttons	Interface
Is there anything that you feel is missing on this site? (Probe: content or site features/functions)	Interaction feature because of Less Interactive Frontend Design.	Nothing	Online proctoring	Nothing much.
If you were to describe this site to a colleague in a sentence or two, what would you say?	Good, Real Life Implementable. Nice Idea Implementation.	It is working well with little bugs so you can prefer this site over the others.	User friendly website and you can use it easily	Good interface and working well with all the inputs.
Do you have any other final comments or questions?	Interface can be improved a little bit. Although a Nice Project.	Overall well maintained and good website.	Easy and user compatibility website.	Good website.

▣ METRICS:

Test Cases Planned	Test Cases Executed	Test Cases: Pass	Test Cases: Fail
28	25	21 (84%)	4 (16%)

RESULTS



#	Task	Average time taken (s)
1	Verify if the user as a student will be able to login with a valid email and valid password.	6.5
2	Verify if the user as a student won't be able to login with an invalid email and invalid password.	3.5
3	Verify If the user is able to sign up without facing any issue or not?	5.5
4	Verify if the user is able to see the Home page or not?.	3.75
5	Verify if the user is able to see the View Certificate Option or not?.	12
6	Verify if the user is able to view the download option or not?.	5.25
7	Verify if the user is able to download software without any issue or not?.	25
8	Verify if the user is able to install software without any bugs or not?.	120.5
9	Verify if the user is able to see a folder where the path of installation was given.	5.25
10	Verify if the user can see the main.exe among the packages in the folder.	5
11	Verify if the user after running the main file is able to see Graphical User Interface.	5.75
12	Verify if the user is able to see all the four buttons with 2 inbuilt templates.	5.75
13	Verify all the fields in the form are functioning well or not for the generation of one certificate?	7.25
14	Verify if the user is able to import the csv file from his/her computer.	8.75
15	Verify if the user is able to see the debug frame or not?.	5.5
16	Verify if the user is able to see the progress of generating certificates.	37
17	Verify if the user is able to see the file named result template in the folder in which information of all generated certificates is stored along with their hash value.	5.5
18	Verify if the user is able to see the folder named teams in which the certificates will be present and all their credentials will be correct.	5.25

19	Verify if the certificates with their hash code are mailed to the registered id.	15.75
20	Verify that the user is able to download the certificate after entering hash code.	28
21	Verify whether the user is able to import signatures or not?.	9.5
22	Verify whether the user is able to design his/her own template by clicking on the import.csv button.	29
23	Verify whether the user is able to see different coordinates for different locations present on the certificate template after using image coordinate identifier.	5.5
24	Verify if the user is able to generate certificates by entering correct coordinates and even import images or not?.	11.25
25	Verify if the user is able to logout from the system.	3.5

8.IMPLEMENTATION

19BCB0030

Rupin Patel

```
def import_csv_data_1():

    csv_file_path = askopenfilename()
    print(csv_file_path)
    print("????????????????????????????????????????????????????????????")
    arr.set(csv_file_path)
    csv_text= Label(root, text=arr.get())
    csv_text.grid(row=9, column=0, pady=20)
    global my_e
    with open(arr.get()) as file:
        data = csv.reader(file, delimiter=',')

        ncol = len(next(data))
        ncol=ncol-1
        print(ncol)

        file.seek(0)
    with open(arr.get(),"r") as file:
        reader = csv.reader(file, delimiter=',')
        data = list(reader)
        row_count = len(data)
        print(row_count)

        file.seek(0)
    for x in range(ncol+1):
        my = Entry(root)
        my.grid(row=10, column=x, pady=20, padx=5)
        my_e.append(my)

def image():
    image_path = askopenfilename()
    print(image_path)
```

```

print("????????????????????????????????????????????????????????????????????????????????????")
arr_image.set(image_path)
csv = Label(root, text=arr_image.get())
csv.grid(row=12, column=0, pady=20)

```

```

def generate():
    config = {
        <Firebase configuration>
    }
    firebase = pyrebase.initialize_app(config)
    storage = firebase.storage()
    def calc_start(center, a, size_of_font):
        length = len(a)
        center[0] = int(center[0] - size_of_font * (length / 2))
        return center
    print(arr.get())
    print(base_destination.get())
    for i in range(0,1):
        try:
            group=arr.get()
            print(group)
            print(':::::::::::::::::::::::::::::::::::::')
            with open(group)as file:
                data =csv.reader(file,delimiter=',')
                ncol = len(next(data))
                d = list(data)
                row_count = len(d)
                print(row_count)
                # Read first line and count columns
                file.seek(0)
                print((ncol))

                count=0
                l=[]
                f=[]

                for row in data:
                    count+=1
                    if(count==1):
                        continue
                    for i in range(0, ncol):
                        l.append(row[i])
            print(l)

            for j in range(0,len(l),ncol):
                f.append(l[j:ncol+j])

            print(f)
            q=[]
            for i in range(0,len(f)):

                q=f[i]
                print(q)
                for k in range(0,len(q),4):
                    print(q[k])
                    try:

```

```

destination = base_destination.get() + q[k] + '/'
if not os.path.exists(destination):
    os.makedirs(destination)
img = cv.imread(arr_image.get())
m_q=[]
for en in my_e:
    l_q = []
    l_q = list(map(int, str(en.get()).split(',')))
    m_q.append(l_q)

font = cv.FONT_HERSHEY_TRIPLEX

size_of_font = 40
for a in range(0, len(m_q)):
    start_1 = calc_start(m_q[k+a], q[k+a], size_of_font)

    cv.putText(img, q[k+a], tuple(start_1), font, 2, (0, 0, 0), 2,
cv.LINE_AA)

cv.imwrite(destination + q[k] + '_' + q[k+1] + '_' + q[k+2] + '.jpg',
img)

path = q[k] + " " + q[k + 1] + " " + q[k + 2] + '/' + '.jpg'
uniq_id=str(hash(path))
path_on_cloud = "certificates/" + q[k] + " " + q[k+1] + " " + q[k+2]
+ ".jpg"

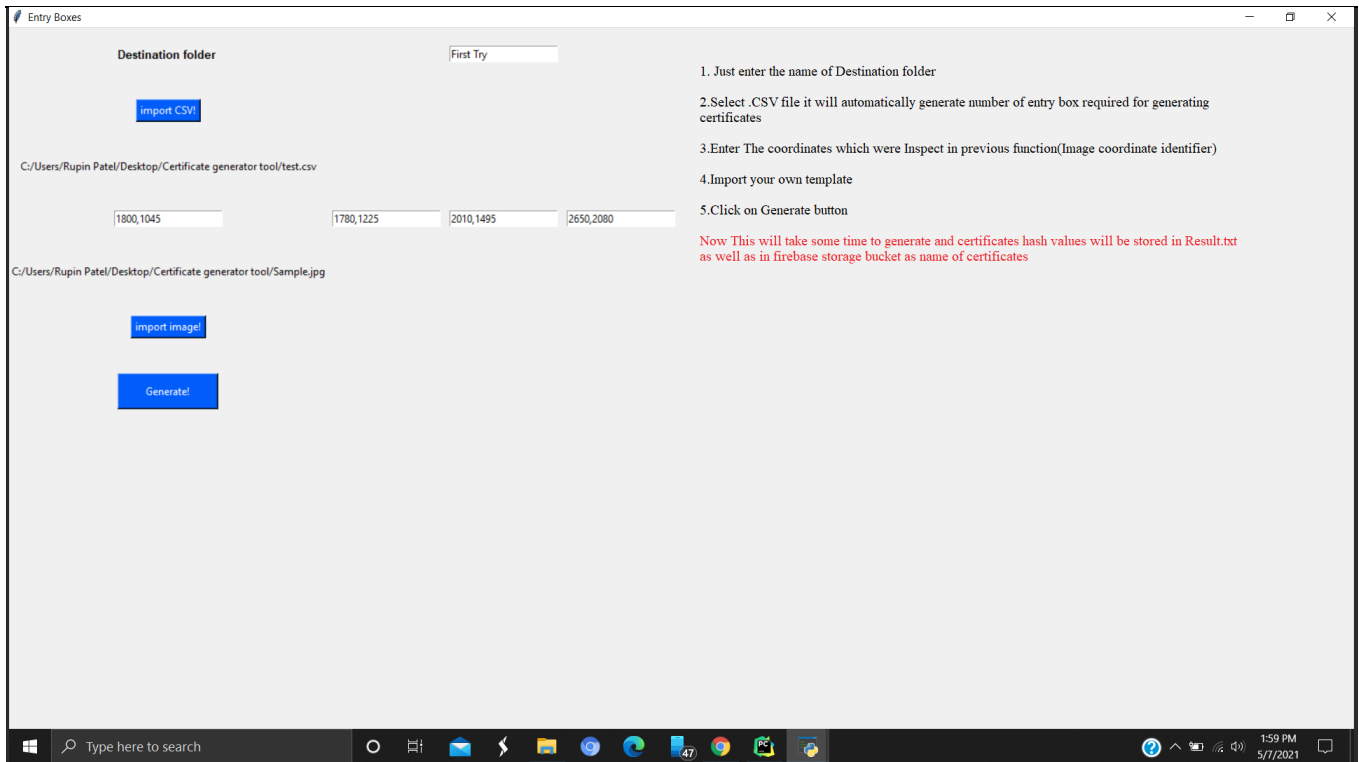
path_local = destination + q[k] + ' ' + q[k+1] + "_" + q[k+2] +

storage.child(path_on_cloud).put(path_local)
path_cloud = uniq_id + '/' + '.jpg'
storage.child(path_cloud).put(path_local)
#f_txt = open("Result.txt", "w")
with open('Result.txt', 'a') as the_file:
    the_file.write(q[k]+" "+"sucessfull"+" "+"+ uniq_id+'\n')

    print(q[k]+" "+"sucessfull"+" "+"+ uniq_id)
except:
    print("Something went wrong")

except:
    print("file is not proper format: {}".format(arr))

```



19BCB0029
Sameer Bishnoi

```
def click_event(event, x, y, flags, params):

    # checking for left mouse clicks
    if event == cv2.EVENT_LBUTTONDOWN:
        img_1 = PIL.Image.open(image_path)

        # fetching the dimensions
        wid, hgt = img_1.size

        a=(wid/800) * x
        b=(hgt/500) * y
        # displaying the coordinates
        # on the Shell
        a1=int(a)
        b1=int(b)
        print(a1, ' ', b1)

        # displaying the coordinates
        # on the image window
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(img, str(a1) + ', ' +
                    str(b1), (x, y), font,
                    1, (255, 0, 0), 2)
```

```

cv2.imshow('image', img)

# driver function
if __name__ == "__main__":
    image_path = askopenfilename()
    print(image_path)
    print("????????????????????????????????????????????????????????????")

    # reading the image
    img = cv2.imread(image_path, 1)
    img_1 = PIL.Image.open(image_path)

    # fetching the dimensions
    wid, hgt = img_1.size

    # displaying the dimensions
    print(str(wid) + "x" + str(hgt))

    img = cv2.resize(img, (800 , 500))

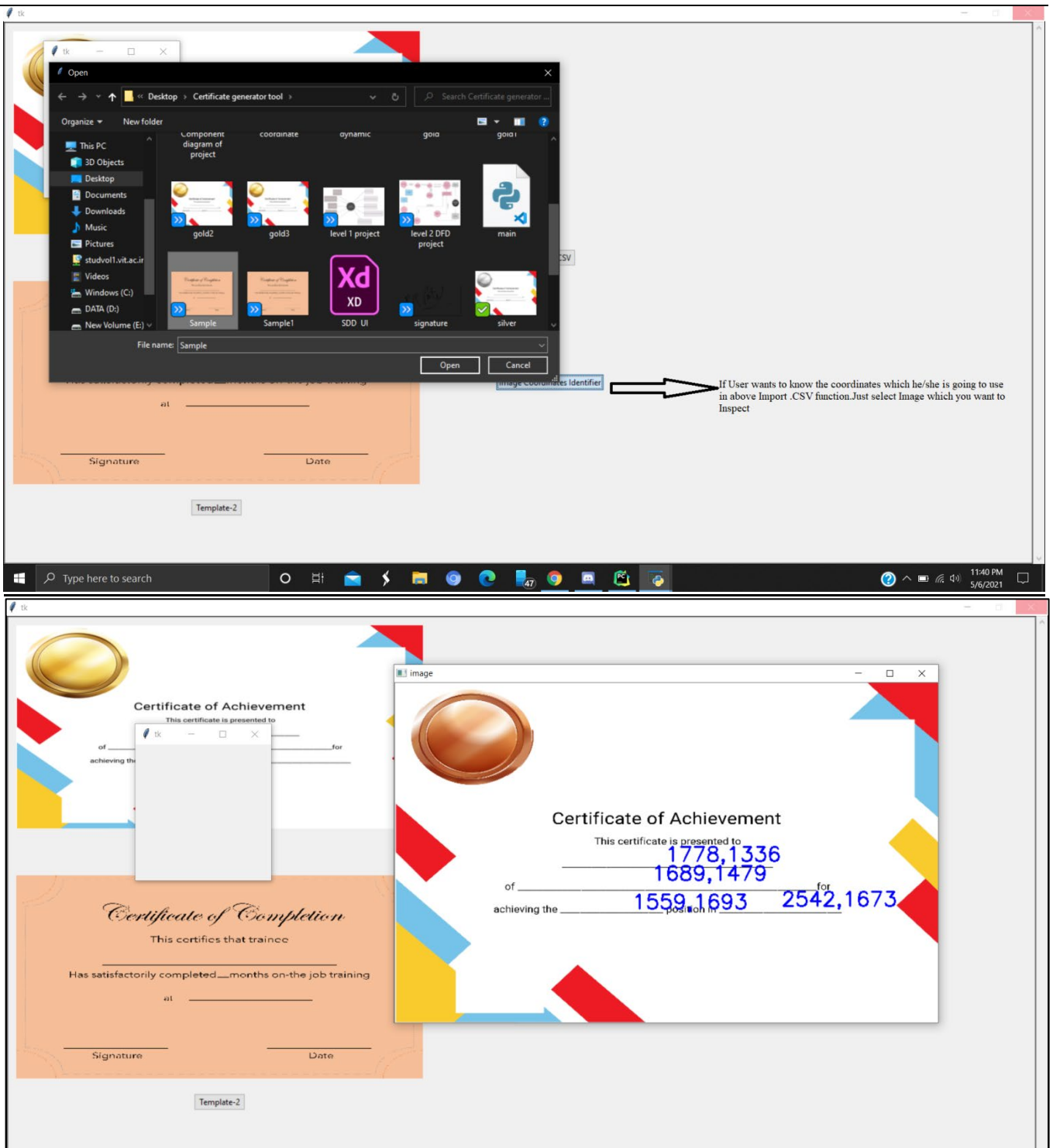
    cv2.imshow('image', img)

    # setting mouse handler for the image
    # and calling the click_event() function
    cv2.setMouseCallback('image', click_event)

    # wait for a key to be pressed to exit
    cv2.waitKey(0)

    # close the window
    cv2.destroyAllWindows()

```



17BCE00430

Raghav Anand

```

def openfile():

    self.filename = filedialog.askopenfilename(title="Select Signature")
    ttk.Label(self,text='Signature Path',font=('calibre',10,'bold')).grid(row=9,column=0)
    ttk.Label(self,text=self.filename, font=('calibre', 10, 'bold')).grid(row=9,column=1)
    print(self.filename)


def submit():

    name = name_var.get()
    duration = duration_var.get()
    company = company_var.get()
    date=date_var.get()
    email_id = email_var.get()


    def calc_start(center, a, size_of_font):
        length = len(a)
        center[0] = int(center[0] - size_of_font * (length / 2))
        return center


    print("The name is : " + name)
    print("The duration is : " + duration)
    print("The company is : " + company)
    print("The date is : "+ date)
    print(self.filename)


    base_destination = 'company/'
    #included_ext = ['csv']
    # arr = [fn for fn in os.listdir() if any(fn.endswith(ext) for ext in included_ext)]
    # no_of_files = len(arr)
    im0=self.filename


    for i in range(0, 1):

        try:

            print("::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::")

            for i in range(0, 1):
                try:

                    destination = base_destination + company + '/' + duration + '/'
                    if not os.path.exists(destination):
                        os.makedirs(destination)
                    im1 = Image.open('Sample.jpg')

                    im3 = Image.open(im0)
                    back_im = im1.copy()
                    back_im.paste(im3, (500, 1800),mask=im3)
                    back_im.save('company/_ .jpg', quality=100)

```

```

img = cv.imread("company/_.jpg")

center_1 = [1800, 1045]
center_2 = [1780, 1225]
center_3 = [2010, 1495]
center_4= [2650, 2080]

font = cv.FONT_HERSHEY_TRIPLEX

size_of_font = 40

start_1 = calc_start(center_1, name, size_of_font)
start_2 = calc_start(center_2, duration, size_of_font)
start_3 = calc_start(center_3, company, size_of_font)
start_4 = calc_start(center_4, date, size_of_font)

cv.putText(img, name, tuple(start_1), font, 2, (0, 0, 0), 2, cv.LINE_AA)
cv.putText(img, duration, tuple(start_2), font, 2, (0, 0, 0), 2, cv.LINE_AA)
cv.putText(img, company, tuple(start_3), font, 2, (0, 0, 0), 2, cv.LINE_AA)
cv.putText(img, date, tuple(start_4), font, 2, (0, 0, 0), 2, cv.LINE_AA)

cv.imwrite(destination + name + '_' + company + "_" + duration + '.jpg',
img)

path=name +company+duration+ '/' + '.jpg'
uniq_id=str(hash(path))
path_on_cloud = "certificates/" + name + " " +company+" " +duration+
".jpg"

path_local = destination + name + '_' + company + "_" +duration +'.jpg'
storage.child(path_on_cloud).put(path_local)
path_cloud = uniq_id+ '/' + '.jpg'
storage.child(path_cloud).put(path_local)
from_addr = 'patel.rupin56@gmail.com'
to_addr = email_id
msg = MIMEMultipart()
msg['From'] = from_addr
msg['To'] = to_addr
msg['subject'] = 'Your Certificate id is ' + uniq_id

filename = path_local
attachment = open(filename, 'rb')

part = MIMEBase('application', 'octet-stream')
part.set_payload((attachment).read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', "attachment; filename= " +
filename)

msg.attach(part)

email = 'patel.rupin56@gmail.com'
password = '+++++++'
recipients = 'patel.rupin56@gmail.com'

```

```

mail = smtplib.SMTP('smtp.gmail.com', 587)
mail.ehlo()
mail.starttls()
mail.login(email, password)
text = msg.as_string()
mail.sendmail(from_addr, to_addr, text)
mail.quit()
with open('Result Template-2 .txt', 'a') as the_file:
    the_file.write(name + " " + " sucessfull" + " " + uniq_id +
'\n')

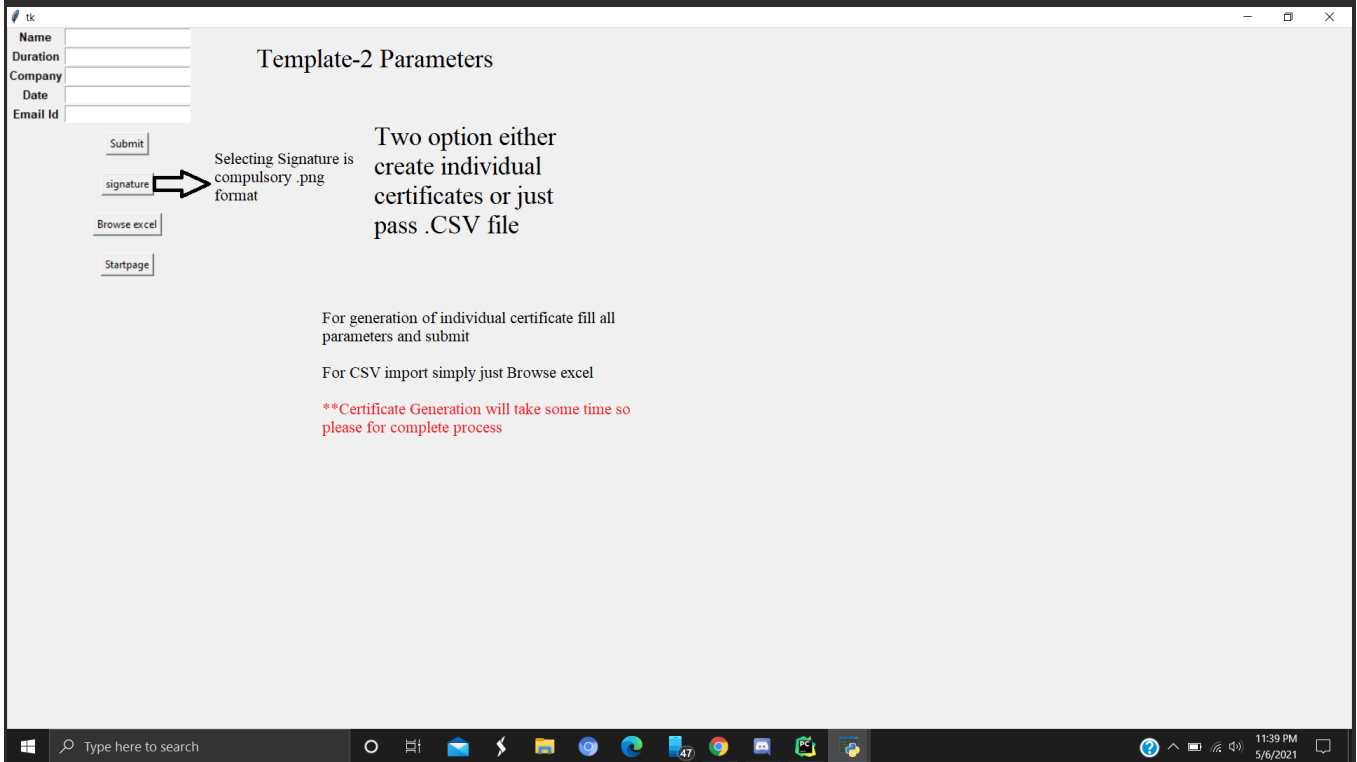
print(name + " " + uniq_id)

except:
    print("Something went wrong")

except:
    print("Error")

name_var.set("")
duration_var.set("")
company_var.set("")
date_var.set("")
email_var.set("")

```



19BCE0691

Yash Agarwal

```

def submit():
    name = name_var.get()
    college = college_var.get()
    position = position_var.get()
    event = event_var.get()
    email_id = email_var.get()

def calc_start(center, a, size_of_font):
    length = len(a)
    center[0] = int(center[0] - size_of_font * (length / 2))
    return center

print("The name is : " + name)
print("The college is : " + college)
print("The position is : " + position)
print("The event is : " + event)
print("The email id is : " + email_id)
base_destination = 'teams/'
included_ext = ['csv']
# arr = [fn for fn in os.listdir() if any(fn.endswith(ext) for ext in included_ext)]
# no_of_files = len(arr)

for i in range(0, 1):

    try:

        print("::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::")

        for i in range(0, 1):
            try:

                destination = base_destination + college + '/' + event + '/'
                if not os.path.exists(destination):
                    os.makedirs(destination)

                if (position == "First"):
                    img = cv.imread("gold.jpg")
                elif (position == "Second"):
                    img = cv.imread("silver.jpg")
                else:
                    img = cv.imread("bronze.jpg")

                center_1 = [1800, 1375]
                center_2 = [1800, 1540]
                center_3 = [1430, 1715]
                center_4 = [2535, 1715]

                font = cv.FONT_HERSHEY_TRIPLEX

                size_of_font = 40

                start_1 = calc_start(center_1, name, size_of_font)
                start_2 = calc_start(center_2, college, size_of_font)
                start_3 = calc_start(center_3, position, size_of_font)
                start_4 = calc_start(center_4, event, size_of_font)

                cv.putText(img, name, tuple(start_1), font, 2, (0, 0, 0), 2,
cv.LINE_AA)

                cv.putText(img, college, tuple(start_2), font, 2, (0, 0, 0), 2,
cv.LINE_AA)

                cv.putText(img, position, tuple(start_3), font, 2, (0, 0, 0), 2,

```

```

cv.LINE_AA)
    cv.putText(img, event, tuple(start_4), font, 2, (0, 0, 0), 2,
cv.LINE_AA)

    cv.putText(img, event, (1500, 500), font, 2, (0, 0, 0), 2)
    cv.imwrite(destination + name + '_' + event + '.jpg', img)
    path=name + position+ event + '/' + '.jpg'
    uniq_id=str(hash(path))
    path_on_cloud = "certificates/" + name + ".jpg"
    path_local = destination + name + '_' + event + '.jpg'
    storage.child(path_on_cloud).put(path_local)
    path_cloud = uniq_id + '/' + '.jpg'
    storage.child(path_cloud).put(path_local)
    from_addr = 'patel.rupin56@gmail.com'
    to_addr = email_id
    msg = MIMEMultipart()
    msg['From'] = from_addr
    msg['To'] = to_addr
    msg['subject'] = 'Your Certificate id is ' +uniq_id

    filename = path_local
    attachment = open(filename, 'rb')

    part = MIMEBase('application', 'octet-stream')
    part.set_payload((attachment).read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition', "attachment; filename= " +
filename)

    msg.attach(part)

    email = 'patel.rupin56@gmail.com'
    password = "+++++++"
    recipients = 'patel.rupin56@gmail.com'

    mail = smtplib.SMTP('smtp.gmail.com', 587)
    mail.ehlo()
    mail.starttls()
    mail.login(email, password)
    text = msg.as_string()
    mail.sendmail(from_addr, to_addr, text)
    mail.quit()
    with open('Result Template-1 .txt', 'a') as the_file:
        the_file.write(name + " " + " sucessfull" + " " + uniq_id +
'\n')

    print(name)

except:
    print("Something went wrong")

except:
    print("Error")

```

