AutoCertiGen: Revolutionizing the Automation of Electronic Batch Certificate Generation using Apache Libraries

Tina Sachdeva[1*], Akansha Jain[1], Khushi Srivastava[1] and Meenakshi Bharadwaj[1]

ABSTRACT

The production of digital documents such as certificates for various individuals has been an arduous problem for every organization, particularly educational institutions with thousands of students enrolled and wherein it is required to issue certificates to hundreds of them on a regular basis. This task, particularly, crops up during cultural and technical festivals but is also required for council position holders, project appreciators, society members to name a few. We live in a modern era of technology where individually entering candidate information and then certifying the documents is not a realistic choice. This served as the motivation for us to take this up as a project and hence, we automated this crucial and time taking process, by proposing and building an offline Android application named 'AutoCertiGen' that has the potential to generate thousands of certificates in a matter of minutes, complete with signatures if needed. The only thing the user needs to do is enter the number of certificates to be produced, upload the input details in the form of an Excel file, select the certificate from a list of predefined templates in the application, and add the signatories' relevant data. This is significant because previously no platform in the form of an Android application has been dedicated towards automating this task. The application has been built using Android Studio which is the official Integrated Development Environment (IDE) for Google's Android Operating System, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. Java is used for the backend development of the application while XML is used to design its layout and its Graphic User Interface (GUI). We used Apache POI and Apache PDFBox Java libraries, modified to be Android-compatible, for all development purposes. Moreover, the application is lightweight and does not require internet connectivity for operation. It addresses the problem at hand with just a few taps on a touchscreen.

Keywords: *Certificate, electronic, automate, Android, signature, Excel, template, Apache*

1. Introduction

Information Technology has been remarkably scaling new heights in the past few decades, from the invention of the first computer in the early 1800s to the beginning of artificial intelligence in the 1950s, and now various breakthroughs in human-computer interactions, high-performance computing, compiler optimization, etc. It is beyond doubt that it forms an indispensable part of our lives now. This development may also be observed in the fact that digital document management now controls

1 *Department of Computer Science, Shaheed Rajguru College of Applied Sciences for Women, University of Delhi*

* *Corresponding Author ✉ tina.sachdeva@rajguru.du.ac.in*

the majority, if not all, enterprises in the business sector and institutes in both public and private sectors. Recent global events have provided additional momentum to this. The COVID-19 pandemic, which began in early 2020, also marked the beginning of an era in which digitalization of every possible piece of information was not merely a convenience, but a necessity. Even if it weren't a mandate, the benefits of digital documentation make it a good decision. Upgrading to digital documentation allows all of the company's records to be readily searchable, saving time and money, strengthening employee and customer collaboration, and ensuring the protection of sensitive information.

However, the task of digital document production is quite burdensome. Despite all of the advancements in technology and digital services available to the public, the lack of a facility to mediate and simplify the mass issuance of certifications has not been effectively addressed. It can take several hours to fill out the fields in each file individually for hundreds of copies. And when considering educational institutions like schools, colleges, there is an ever-increasing need for the production of thousands of documents like academic, cultural, or sports certificates and appreciation.

This work aims to make electronic development of certificates for a hefty number of students in one go and this has been possible with an automatic certificate generation system. This automated system places the corresponding details of a given student neatly on each certificate, alongside authorization in the form of an electronic signature. To make the process simpler and convenient, the system has been developed in the form of an offline Android application called 'AutoCertiGen'. The certificate templates are pre-defined in the application and the data is uploaded by the user in the form of an Excel sheet file.

The development of this system in the form of an Android application was considered because Android smartphones are the most popular electronic devices used by the global population. In India itself, the number of smartphone users has been projected to be 748 million in 2020, and this figure is expected to rise to 1,341 million during the next ten years (Sun, 2021). Android is developed by Google, and it is based on the Linux kernel (Singh, 2014). Since Android is open-source, developers can use it to build applications that can then be sold on the Android market. As a result, its applications are affordable for developers to create and for customers to utilize as well. It is the dominating mobile operating system today, accounting for more than 80% of the market (Magas and Zadorozhnyi, 2018). As a result, our application could be accessed by as many people as possible.Being an offline application provides the convenience of being able to utilize it anytime, anywhere i.e., even in the absence of internet connectivity.

During the formation of this application, many unexpected complications arose, for which inventive solutions were devised to attain desired results. Out of many approaches that were considered, some could be implemented in Java but not in Android programming. Other solutions required APIs and libraries that were not open source. The final method employed Apache POI and Apache PDFBox Java libraries that were modified to be Android-compatible. These libraries are used to create PDF files and insert the data from the input Excel sheet file. So, with this application, thousands of certificates that previously had to be filled out manually and validated individually may now be created in a matter of minutes and only with a few clicks.

2. Related Work

There has been a lot of recent work into automated certificate generation systems. Many authors toiled away for hours trying to come up with a solution that proved to be useful to a wide range of users. Srushti A. Shimpi*et al.* worked on a certificate generation system that is flexible in terms of generating student mark sheets. Their system concentrated primarily on database technology and a credit-based grading system

(CBGS) (Shimpi *et al.*,2014). They used PHP as their server-side scripting language for the implementation of their website. Similar work was done by Bharti Chikankar *et al.* in which they propose an application called Automated Batch Certificate Generation and Verification System ABCGVS, which is based on client-server technology. Their method focused mostly on database technologies for retrieving student data and a credit-based grading system (CBGS). They also created a website with the bootstrap four CSS frameworks and their associated components like the mobile responsive navigation bar, modals, cards, and carousels (Chikankar and Jaiswal, 2020). Taking into consideration the problems encountered by many institutes in maintaining students' data manually, V. Thusyanthy et al. developed a system named College of Technology Management System (COTMS). Their main objective was to formulate a computer software application dedicated to Jaffna College of Technology only. They developed COTMS using C#.Net programming language and also designed a relational database management system to construct a secure database with specific access privileges (Thusyanthy *et al.*,2016). To enhance the existing automated batch certificate generation system, Ahmed Dalhatu Yusuf et al. designed a web application that allows an end-user to define certificate template and template format without requiring XML knowledge by clicking a few buttons and typing from the system GUI, verifying the certificate, and generating one or more certificates simultaneously in an instantaneous manner (Yusuf *et al.*, 2017).

These papers were all attempting to solve the same problem, but each of them developed a web-based application to implement their proposed system using different technologies that work well on a laptop or a desktop.

With the advancement of mobile phone technology, the world is shrinking. As the number of customers grows, so does the amount of facilities available. Mobile phones have changed and become an indispensable part of our life, starting with simple ordinary handsets that were only used for making phone calls to a phone being used as a camera, music player, tablet PC, television, and a web browser, among other things (Magas and Zadorozhnyi, 2018). Mobile phones have practically placed the world at our fingertips and due to this reason, we collectively decided to implement the certificate generation process in the form of an offline Android application. With just a few clicks, the application will be able to generate numerous certificates at the same time. It will be widely used because it will save time when entering participant information into the certificate generation system. It will use the data from an Excel sheet to automate the process. This application may be used anywhere because it is built on Android, and users do not need to be connected to the internet to use it. We've also incorporated functionality that allows one to add signatures very easily.

3. Methodology and Implementation

In this section are discussed the system flowchart, use cases, and application's implementation, wherein we deeply discuss the working along with all interface screenshots and challenges faced while achieving the basic functionality of our application.

1. *Technology used*

The application has been built using Android Studio which is the official Integrated Development Environment (IDE) for Google's Android Operating System, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. Java is used for the backend development of the application. According to Table 5 of the publication (Pereira *et al.*, 2017), JAVA is listed among the top five languages for various combinations of objectives such as 'Time & Memory', 'Energy & Time', 'Energy & Memory', and 'Energy & Time & Memory'. This demonstrates that the solutions need less energy and time to implement with Java. We have used XML to design the layout and Graphical User Interface (GUI) of the application.

Two API libraries, namely Apache POI and Apache PDFBox, were the main pillars for the working of this application. Apache POI is an open-source library developed and distributed by Apache Software Foundation that allows programmers to create, modify, and display Microsoft Office files using Java programming. Our application uses the Android-compatible version of this library available at the GitHub repository 'poi-android' of Alex Saveau(Saveau, 2018) to extract data from the Excel sheet provided by the user. Apache PDFBox is a free and open-source Java library that facilitates the development and conversion of PDF documents. The Android-compatible version for this library was taken from the GitHub repository 'PdfBox-Android' of Tom Roush (Roush, 2021) to create copies of the certificate templates and then add the details in the certificate. Table 1 provides the specs. of the device used for testing and debugging purposes.

Table 1: Specifications of device used for testing and debugging

Processor	Octa-core Max 2.02GHz
RAM	32 GB, 64 GB
Operating System	Android 8, 9, 10, 11

2. *System Flowchart*

The working of the application is clearly shown in the system flowchart (see Figure 1).

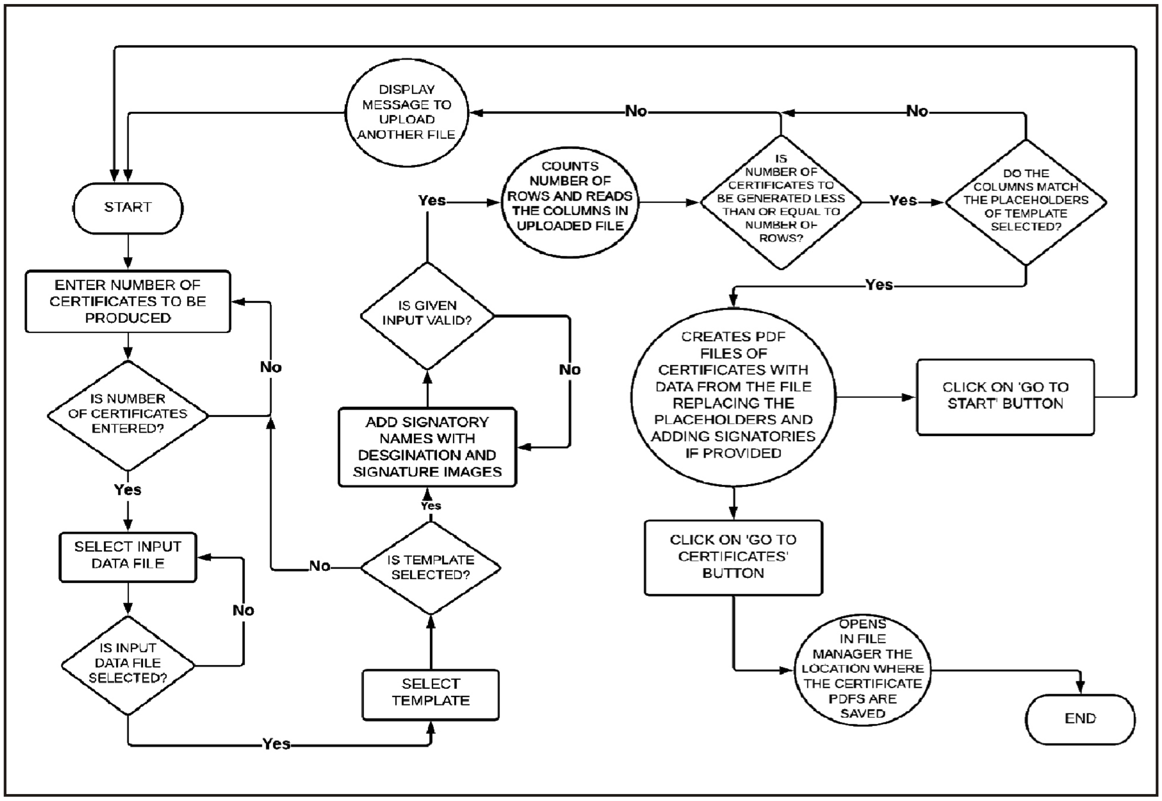

START
ENTER NUMBER OF CERTIFICATES TO BE PRODUCED
IS NUMBER OF CERTIFICATES ENTERED?
Yes
No
SELECT INPUT DATA FILE
IS INPUT DATA FILE SELECTED?
No
Yes
SELECT TEMPLATE
IS TEMPLATE SELECTED?
No
Yes
ADD SIGNATORY NAMES WITH DESGINATION AND SIGNATURE IMAGES
IS GIVEN INPUT VALID?
No
Yes
DISPLAY MESSAGE TO UPLOAD ANOTHER FILE
COUNTS NUMBER OF ROWS AND READS THE COLUMNS IN UPLOADED FILE
IS NUMBER OF CERTIFICATES TO BE GENERATED LESS THAN OR EQUAL TO NUMBER OF ROWS?
No
Yes
DO THE COLUMNS MATCH THE PLACEHOLDERS OF TEMPLATE SELECTED?
No
Yes
CREATES PDF FILES OF CERTIFICATES WITH DATA FROM THE FILE REPLACING THE PLACEHOLDERS AND ADDING SIGNATORIES IF PROVIDED
CLICK ON 'GO TO START' BUTTON
CLICK ON 'GO TO CERTIFICATES' BUTTON
OPENS IN FILE MANAGER THE LOCATION WHERE THE CERTIFICATE PDFS ARE SAVED
END

Figure 1. 'AutoCertiGen' application - System Flowchart

As the application starts, an activity appears asking the user to 'enter the number of certificates to be produced'. The 'number of certificates' input field must be mandatorily filled; else the application won't proceed. Also in the same activity, there is a button to 'Browse' the data file (basically an Excel sheet file) that can be in the format .xlsx only. The user is required to browse the data file to proceed to the next activity. When selected,

Table 2: Use Cases

Actors	Use Cases
User	Upload Excel Files Select Template Add Signatories Go to Certificates Go to Start
Application System	Generate Certificates

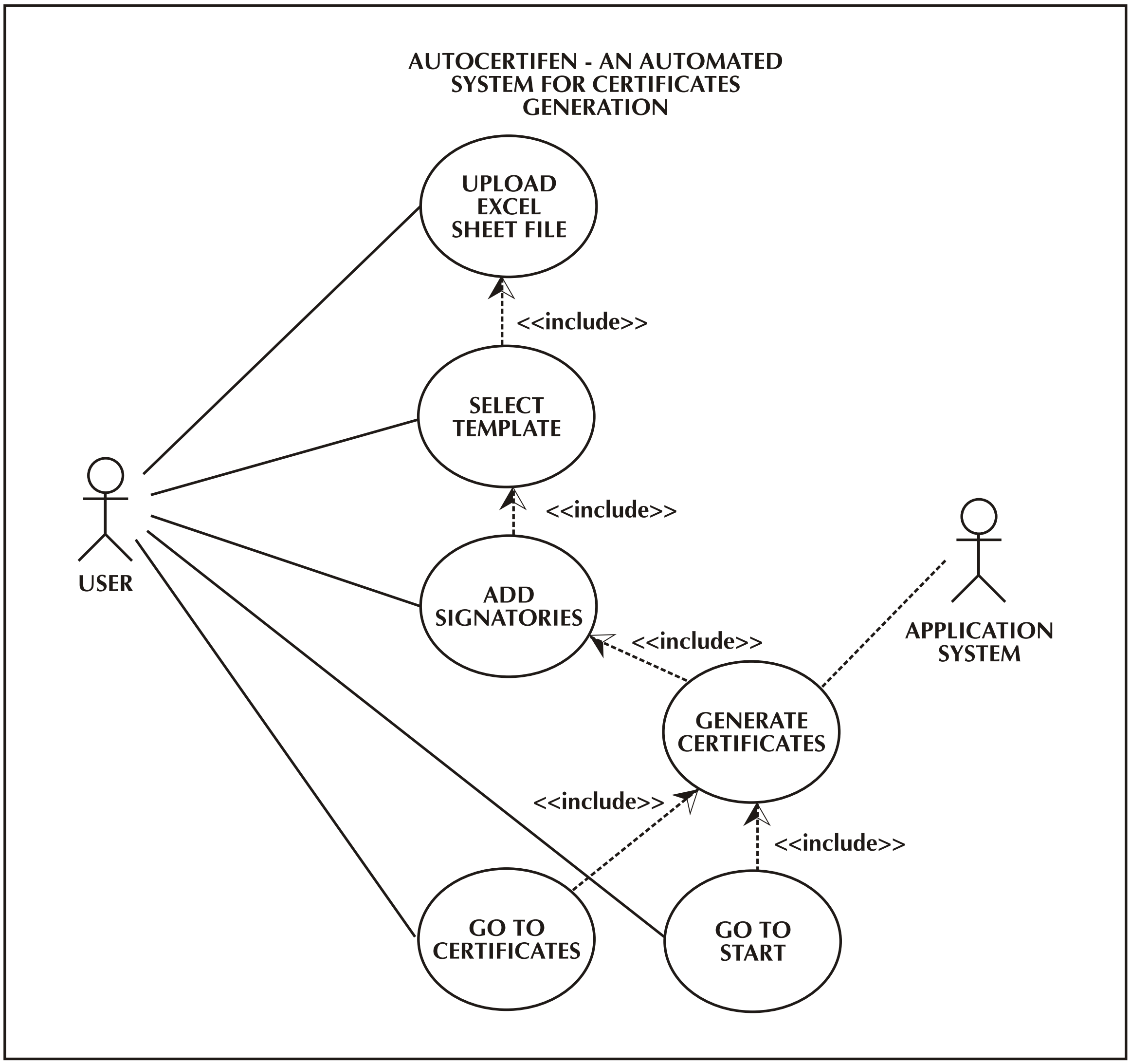
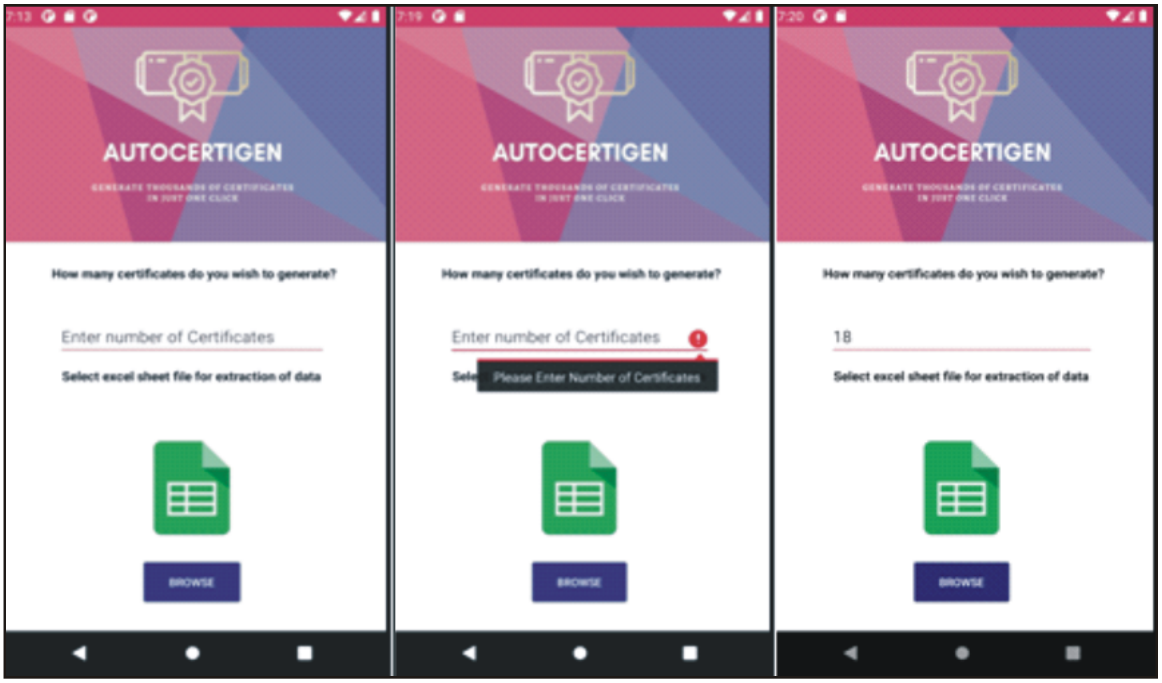
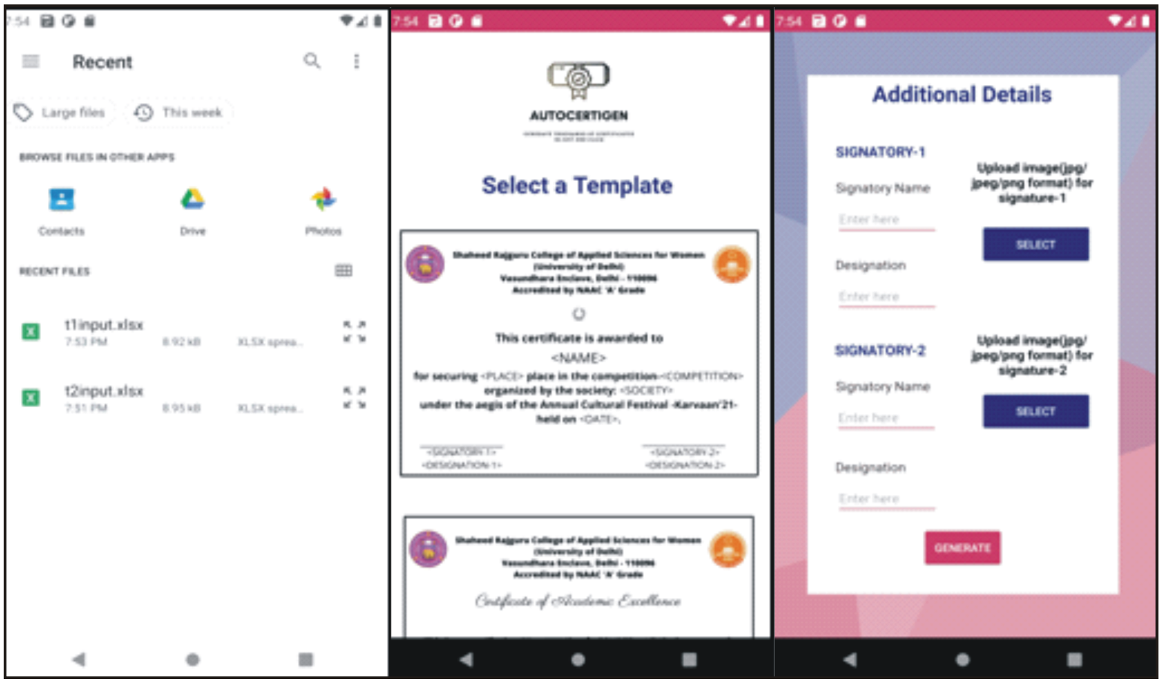


Figure 2: Use Case Diagram: 'AutoCertiGen' Application


AUTOCERTIGEN
GENERATE THOUSANDS OF CERTIFICATES IN JUST ONE CLICK
How many certificates do you wish to generate?
Enter number of Certificates
Select excel sheet file for extraction of data
BROWSE
Please Enter Number of Certificates
18

3 a) 3 b) 3 c)

Figure 3: Activity interface to enter the number of certificates and browse data files



Shaheed Rajguru College of Applied Sciences for Women
(University of Delhi)
Vasundhara Enclave, Delhi - 110096
Accredited by NAAC 'A' Grade

This certificate is awarded to
<NAME>
for securing <PLACE> place in the competition <COMPETITION>
organized by the society: <SOCIETY>
under the aegis of the Annual Cultural Festival -Karvaan'21-
held on <DATE>.

<SIGNATORY-1> <DESIGNATION-1>

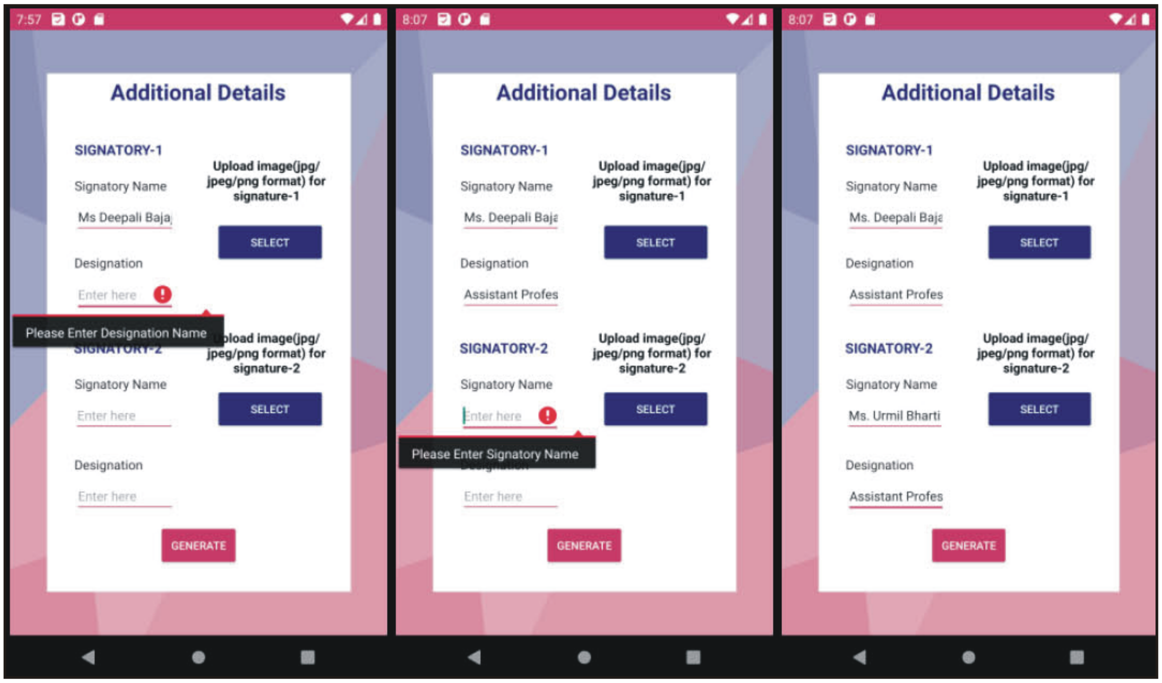
<SIGNATORY-2> <DESIGNATION-2>

Shaheed Rajguru College of Applied Sciences for Women
(University of Delhi)
Vasundhara Enclave, Delhi - 110096
Accredited by NAAC 'A' Grade

Certificate of Academic Excellence

4. a) 4. b) 4. c)

Figure 4: Activities to browse Excel files and to select a template



Additional Details
SIGNATORY-1
Signatory Name
Designation
Upload image(jpg/jpeg/png format) for signature-1
SELECT
SIGNATORY-2
Upload image(jpg/jpeg/png format) for signature-2
GENERATE

Ms Deepali Baja

Enter here

Please Enter Designation Name

Ms. Deepali Baja

Assistant Profes

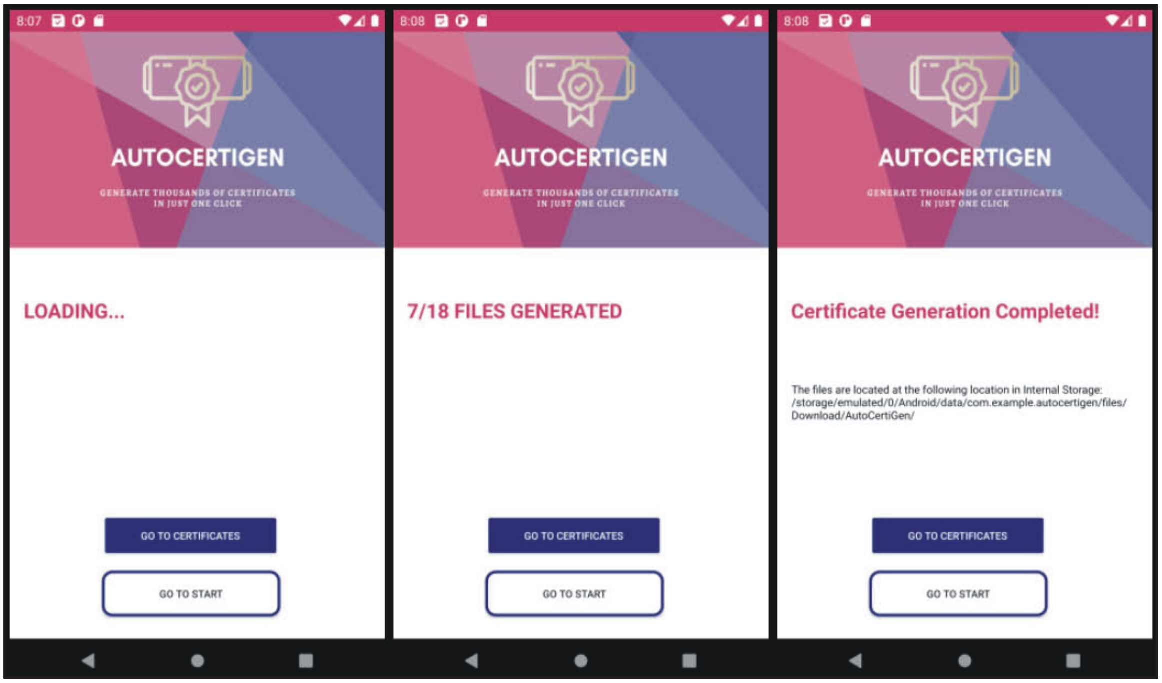
Please Enter Signatory Name

Ms. Urmil Bharti

Assistant Profes

5. a) *5. b)* *5. c)*

Figure 5. Activity interface for adding



LOADING...

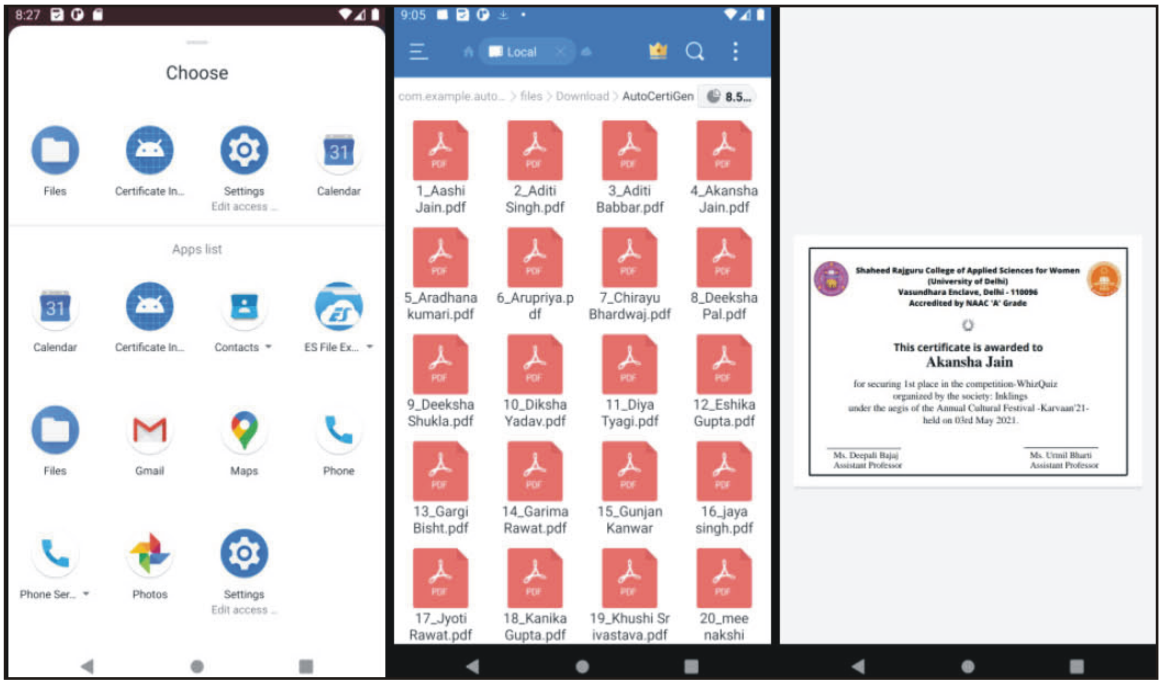
7/18 FILES GENERATED

Certificate Generation Completed!

The files are located at the following location in Internal Storage: /storage/emulated/0/Android/data/com.example.autocertigen/files/Download/AutoCertiGen/

6. a) *6. b)* *6. c)*

Figure 6:Activity interface for certificate generation



Shaheed Rajguru College of Applied Sciences for Women
(University of Delhi)
Vasundhara Enclave, Delhi - 110096
Accredited by NAAC 'A' Grade

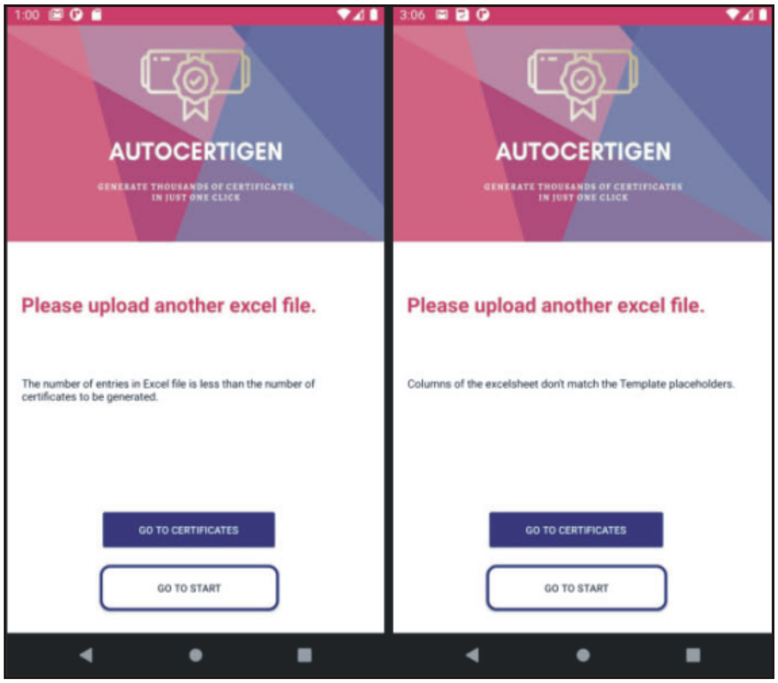
This certificate is awarded to
Akansha Jain

for securing 1st place in the competition-WhizQuiz
organized by the society: Inklings
under the aegis of the Annual Cultural Festival -Karvaan'21-
held on 03rd May 2021.

Ms. Deepali Bajaj
Assistant Professor

Ms. Urmil Bharti
Assistant Professor

7. a) 7. b) 7. c)
Figure 7: Navigating to the folder where the generated certificates are saved



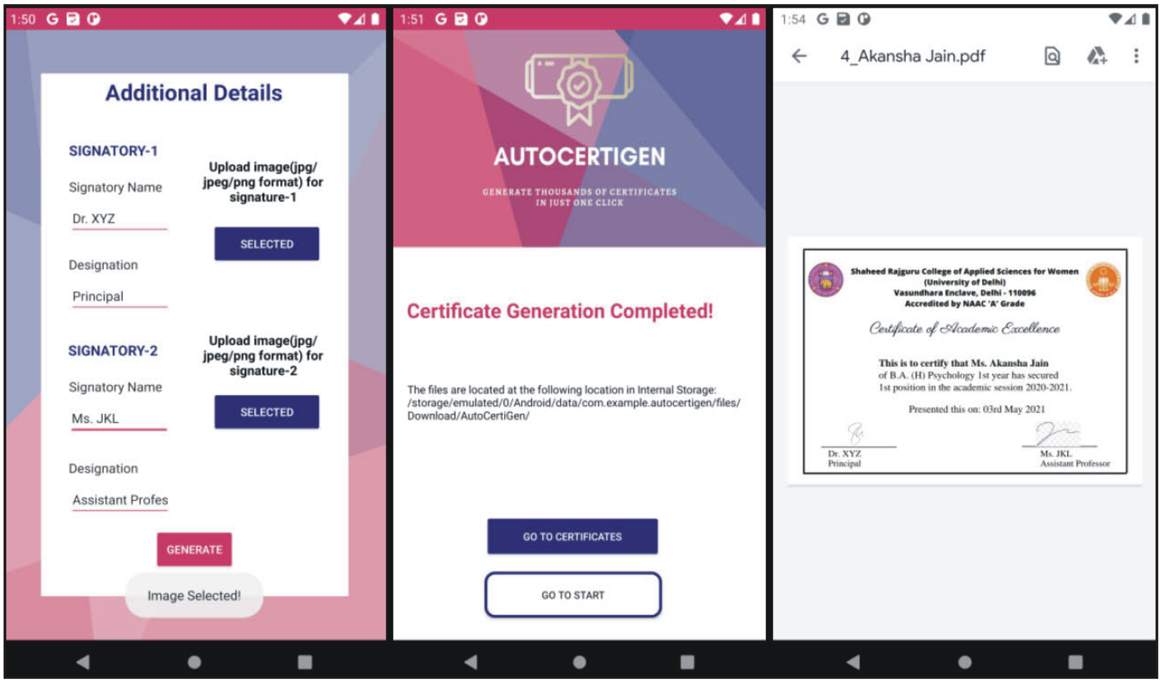
Please upload another excel file.

The number of entries in Excel file is less than the number of certificates to be generated.

Please upload another excel file.

Columns of the excelsheet don't match the Template placeholders.

8. a) 8. b)
Figure 8:Invalid input handling



1:50

Additional Details

SIGNATORY-1

Signatory Name

Dr. XYZ

Upload image(jpg/jpeg/png format) for signature-1

SELECTED

Designation

Principal

SIGNATORY-2

Upload image(jpg/jpeg/png format) for signature-2

Signatory Name

Ms. JKL

SELECTED

Designation

Assistant Profes

GENERATE

Image Selected!

1:51

AUTOCERTIGEN

GENERATE THOUSANDS OF CERTIFICATES IN JUST ONE CLICK

Certificate Generation Completed!

The files are located at the following location in Internal Storage: /storage/emulated/0/Android/data/com.example.autocertigen/files/Download/AutoCertiGen/

GO TO CERTIFICATES

GO TO START

1:54

4_Akansha Jain.pdf

Shaheed Rajguru College of Applied Sciences for Women (University of Delhi) Vasundhara Enclave, Delhi - 110096 Accredited by NAAC 'A' Grade

Certificate of Academic Excellence

This is to certify that Ms. Akansha Jain of B.A. (H) Psychology 1st year has secured 1st position in the academic session 2020-2021.

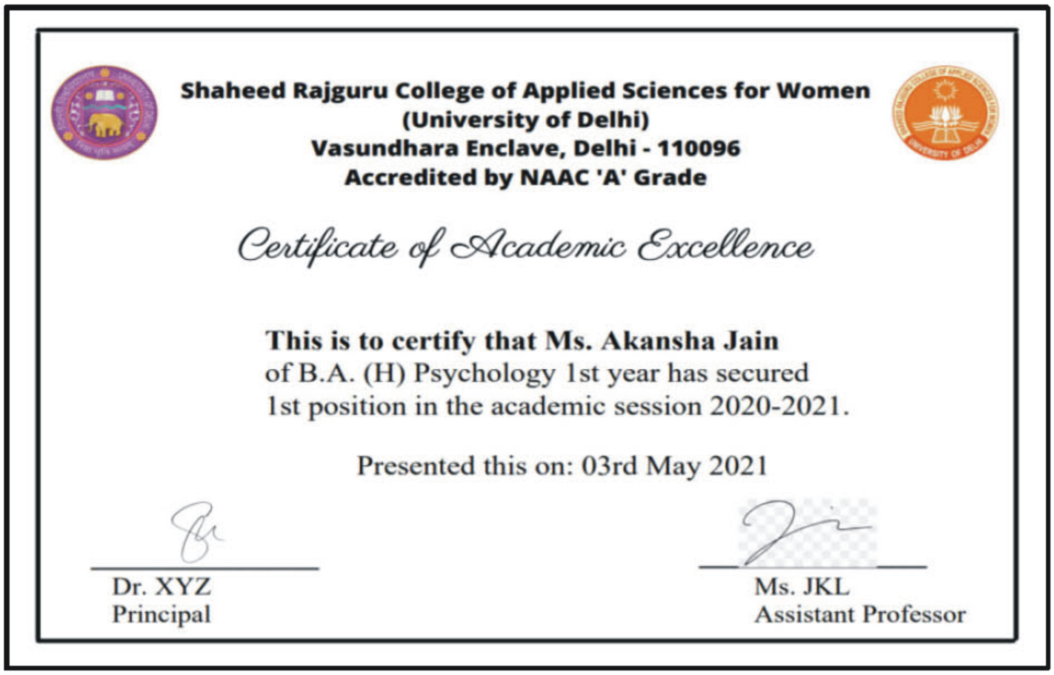
Presented this on: 03rd May 2021

Dr. XYZ
Principal

Ms. JKL
Assistant Professor

9. a) 9. b) 9. c)

Figure 9: Generating signed certificates.



Shaheed Rajguru College of Applied Sciences for Women
(University of Delhi)
Vasundhara Enclave, Delhi - 110096
Accredited by NAAC 'A' Grade

Certificate of Academic Excellence

This is to certify that Ms. Akansha Jain
of B.A. (H) Psychology 1st year has secured
1st position in the academic session 2020-2021.

Presented this on: 03rd May 2021

[illegible]
Dr. XYZ
Principal

[illegible]
Ms. JKL
Assistant Professor

Figure 10: Signed certificategenerated by 'AutoCertiGen'

After entering the number of certificates, the user is required to upload the Excel file using the 'Browse' button to proceed. Fig. 4. a) shows the file browsing from the internal storage after the button is clicked. As soon as the file is uploaded, the application opens the next activity shown in Fig. 4. b) i.e., the 'Select a Template' Activity. After selecting the template from the available ones, the application proceeds to the next activity where the user is asked to input some additional details related to the certificate. Fig. 4. c) shows the activity interface that has additional details with some input fields, it includes the mandatory fields like signatory name, and designation and includes an optional field to upload signature images for both the signatories. It is mandatory to enter valid details for both the signatories to proceed. After filling in all the correct details, the application proceeds to the certificate generation task by clicking the 'Generate' button.

c. *Adding signatory details*

In Fig. 5. a) and 5. b), the activity shows the message to fill details for mandatory fields if left empty. In Fig. 5. c), all mandatory fields are filled, and the user can proceed to generate certificates by pressing the 'Generate' button. Here, the user has an optional field to upload signature images for the signatories that lets the user upload signatures and produce signed certificates. If the signatures are not uploaded, then unsigned certificates will be generated which can be signed after getting printed. This application generates certificates in Portable Document Format i.e., PDFs.

d. *Certificates generated & file location displayed*

In Fig. 6. a), it is shown that the system is getting ready for the certificate generation process. In Fig. 6. b), the user gets updated by the certificate generation live count appearing on the screen. After this, the screen displays the path to the folder where the generated certificates are saved. In Fig. 6. c), the location to the directory of can be seen clearly, user can also jump to this by clicking on the 'Go To Certificates' button, now the user can either terminate the application or start generating another batch of certificates by clicking on to the 'Go To Start' button and repeat all the steps through start activity to the current activity as shown in Fig. 6. i.e., 'Generate Certificates' Activity.

e. *Navigating to the certificates folder*

In Fig. 7. a), the app chooser opens up after pressing the 'Go To Certificates' button in the activity as shown in Fig. 7. c). Since the default Files folder was not effective to navigate the user from the application to the folder location, we used another file manager application (here, ES File Explorer application [2]) to open the location, and it navigates the user to the required directory location as shown in Fig. 7. b). After opening any of the generated certificates, we can see the certificate shown in Fig. 7. c).

f. *Application response on invalid inputs*

There are two cases when the invalid input might be given by the user:

1. *User has entered an invalid count of certificates to be generated:*

As shown in Fig. 8. a), the user has entered the count more than the number of entries available in the Excel file. In this case, it displays a message saying, "The number of entries in the Excel file is less than the number of certificates to be generated" and the user is suggested to upload another Excel file.

2. *User has selected the Excel sheet incompatible with the placeholders of the chosen template:*

As shown in Fig. 8. b), when the user browses an Excel data file with the columns that are incompatible with that of the placeholders of the chosen template from the 'Select a Template' Activity, it displays a message saying, "Columns of Excel sheet don't match

Table 3: Test results for unsigned certificates

Number of certificates to be generated	Time taken in generating certificates (in milliseconds)	
	Template 1	Template 2
60	10023	6987
120	11973	10403
200	16510	15065
500	41044	35401
1000	78109	71982
1500	118055	102040

Table 4: Test results for signed certificates

Number of certificates to be generated	Time taken in generating certificates (in milliseconds)	
	Template 1	Template 2
60	9045	7014
120	14051	13809
200	27030	21008
500	57041	49050
1000	112670	105089
1500	169451	139110

the Template placeholders." and the user is suggested to upload another Excel file.

g. *Signed certificates generated using 'AutoCertiGen'*

For generating signed certificates, the user only has to upload the image by clicking on the 'Select' button in the 'Additional Details' Activity. After uploading the signature images, the button text updates to 'Selected' for the user's acknowledgment. The application then proceeds to the certificate generation after a click on the 'Generate' button and further navigates to the activity shown in Fig. 9. b). After successful certificate generation, the user gets the signed certificates as shown in Fig. 9. c). In Fig. 10, the signed certificate with signatures that are generated by the application 'AutoCertiGen' can be seen.

4. Result

'AutoCertiGen' is an extremely efficient application that is capable of generating an average of 668 signature-less certificates per minute and 493 certificates with signatures. In addition to being time-efficient, the application focuses equally on the quality aspect of the certificates generated. Moreover, the generated certificates are of high image precision and occupy even less than 1 MB.

In order to prove the above claim, we recorded the time taken by the application in producing these certificates. Table 3 shows this data for unsigned certificates while Table 4 for signed ones. The first column is 'the number of certificates to be generated' and corresponding to that the 'time taken for completion' is noted for our two templates in the second column.

Table 5: Comparison with other tools

Parameters	d water. RusAutoCertiGen	AutoCertiGen
Platform	Most of the tools developed for certificate generation are created in the form of websites or web-based applications that employ various technologies.	AutoCertiGen is created in the form of an Android application because it can be used on a regular basis and is easily accessible in almost all scenarios (Summerfield, 2015).
Mode of Operation	Since the tools developed earlier were designed as websites, they require an internet connection to run on web servers.	The functionality of our application does not require an internet connection so developing an Android application was a better alternative.
Factors affecting Run-time	Aside from the system's internal characteristics, the runtime is also influenced by external factors, such as the speed of internet connection used by web-based applications and websites to reach the server.	Since the application is running in an offline mode, its runtime is solely determined by the system's internal factors.
Performance	Tools designed as websites use JavaScript to retrieve data from servers, making their functions slower than that of Android applications (Seo Team, 2020).	Android allows applications to carry out operations quickly and consistently, enhancing the user experience as they store data on the device directly (Seo Team, 2020).
Convenience	A responsive website can't leverage all smartphone features efficiently. Some websites can access features like camera, GPS, microphone, but the quality of interaction remains inferior to that of mobile applications (Tania, 2019).	AutoCertiGen can leverage the device's capability to optimize the user experience. It can access the mobile phone features to perform creative functions and engage users interactively (Seo Team, 2020).

From the above results, we can conclude that the application takes on average 89.796 seconds to produce 1000 certificates without signatures and 121.871 seconds to produce the same quantity with signatures.

To ensure the efficiency of our application on different parameters, a comparative study with other similar tools developed was carried out.

The application is also robust in the sense that it handles situations with erroneous inputs in a user-friendly manner by reporting appropriate messages. Like, if the user selects an incompatible data file as input, the application does not proceed and instead displays a message requesting the user to upload the correct file. In another case where the number of certificates to be generated is more than the number of entries in the Excel file then the application again prompts with a message. In both cases, the application does not compromise with respect to the final results.

5. Conclusion

We developed an automated offline Android application using Java to facilitate the process of creating e-certificates. The application is structured to accept user input in the form of an Excel sheet file, followed by entering the number of certificates to be produced, and then asking the user to choose a predefined certificate template. Finally, the user is prompted to provide the signatories' details, with the option of attaching their electronic signatures. The customized Android-compatible versions of two Java libraries, Apache POI and Apache PDFBox, were used to create the PDF files and add the content extracted from the input Excel file. The application has been programmed to notify the user if any incorrect input is provided from the user's end. We were able to deliver our focal objective of automating this entire operation and thus produce a huge number of certificates within minutes with just a few clicks.

6. Future Scope

This application has immense potential for expansion in the future. We aim to add an extremely important and useful functionality wherein a user can upload her own templates while still keeping the process automated. This will make the application easily customizable and hence, adapt to a variety of events and activities for which certificates are required. Another important aspect for extending this application would be an interface that allows institutions to verify the awarded certificates to candidates by just entering basic minimal information. This external certificate validation system feature will be a tool for popularizing the application by connecting more and more institutes to it. One more crucial direction in which we wish to work is the possibility of uploading files to the user's cloud storage which will help the application to work with minimal storage constraints.

7. References

1. Chikankar, B., & Jaiswal, S. (2020). Certificate Generation System. *International Journal of Research in Engineering, Science and Management, 3*(8), 570-573.
2. ES Applications (2021). *ESx File Explorer* Ver. 1.5.6) [Mobile app]. Google Play Store.
3. Magas, L. M., & Zadorozhnyi, V. M. (2018). *Android is the most popular mobile operating system* (Doctoral dissertation, ВНТУ).
4. Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J. P., & Saraiva, J. (2017). Energy Efficiency across Programming Languages: How Do Energy, Time, and Memory Relate? *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*, 256–267.
5. Roush, T. (2021, September). *PDFBox Android*. GitHub.
6. Saveau, A. (2018, April 10). *POI Android*. GitHub.
7. SEO Team. (2020, August 13). *Mobile Application vs. Mobile Website: A UX Comparison – Better Option?* Saffron Tech.
8. Shimpi, S. A., Mandare, S., Trivedi, A., &Sonawane, T. (2014). Certificate generation system. *International Journal on Recent and Innovation Trends in Computing and Communication, 2*(2), 380-383.
9. Singh, R. (2014). An overview of the Android operating system and its security. *Int. Journal of Engineering Research and Applications, 4*(2), 519-521.
10. Sun, S. (2021, September 3). *Smartphone users in India 2010-2040.* Statista.
11. Summerfield, J. (2015). Mobile website vs. mobile app: Which is best for your organization. *Human Service Solutions.*
12. Tania, H. (2019, September 1). *Pros and Cons of Mobile Websites and Mobile Apps.* Ruby Garage.
13. Thusyanthy, V., Thiruthanigesan, K., Wanninayake, W. M. P. P., & Thiruchchelvan, N. (2016). Automated Certificate Issuing and Students' Management System for College of Technology, Jaffna: Sri Lanka. *Middle-East Journal of Scientific Research, 24*(7), 2204-2208.
14. Yusuf, A. D., Boukar, M. M., & Shamiluulu, S. (2017, November). Automated batch certificate generation and verification system. *In 2017 13th International Conference on Electronics, Computer and Computation (ICECCO)* (pp. 1-5). IEEE.