Securing ATM Transactions

Soubhik Sinha[1] , Prishita Raj[2] , Anisha Agarwal[3]

[1,2,3]Department of Information Technology , VIT University , Vellore , Tamil Nadu , India

1. ABSTRACT

ATMs allow you to make deposits and withdraw money and you can even print a statement, view your account balance and even transfer money between your accounts. ATMs, if properly secured, are a safe and most convenient way to manage our money. To protect our money and transactions we need to safeguard them from different types of attacks. Nowadays due to development in technology, new ATM machines are being built up with more and more security. But to destroy this security level, threats are being imposed. Regardless of enhancement in the automation, still ATMs are prone to thefts and frauds. This project shows some of the trending technologies to reduce transaction flaws, which enables the user only to do the transaction(s) , without much hassle. The hybrid model depicted will have Fingerprint scanner, QR code scanning method, 3D Facial recognition system and the GSM module based OTP authentication.

2. INTRODUCTION

We all carry debit / credit cards, do online shopping via net-banking, pay to milkmen using UPI based mobile applications. Apart from all of the aforementioned paying methods, payment by liquid cash cannot be denied. Though in this fast paced technological era, usage of paper money is getting diminished day-by-day, Liquid cash still holds a great position, today, in terms of money transfers, informal payments, and of course, to buy something from the foot market. But where in this world can someone get paper money – the ATM (Automated Teller Machine). Though the introduction of this sublime creation was done in the late 1980s in London, it has been developed a lot – from the introduction of interactive and voice enabled touch screens to satellite connectivity, to notify the user how much amount has been withdrawn. But, cybercrimes have now often become a threat to one's hard earned money. From card cloning machines in the card inlets, to fixing a spy camera on the keypad, hackers won't spare a single chance to steal. Thus, ATM transactions are quite vulnerable. ATM cards are protected by a 4-digit PIN number, which is only known to users – but those 4-digits won't give a guarantee of safeguarding one's bank account. Nowadays, one may find 2 kinds of Debit / Credit Cards available to the users – one is having a magnetic black strip embedded on the card's back side, and the other – the safer version, having a chip inside; both have all the information linking to one's bank account. The former can easily be compromised, if deliberately swiped / inserted through a cloning machine (that can possibly be attached to an ATM's card insertion slot). Rather, the latter can be safe up to some extent. Introduction of the finger-print scanner, which can never be manipulated, and will always remain unique – can be one of the most powerful keys for successful and noiseless transactions. But non-manipulation doesn't mean it cannot be copied. There are special papers / transparent plastic pieces , which can save the imprint of the finger-prints. Thus, the finger-print scanner alone cannot fight against the intruders. Retinal scanners were introduced in mid 1990s in New York – now one can even find them in the latest iPhones. Retinal scanners are quite accurate, but can be slow, because an ATM is not an iPhone or just a normal hand-held machine. Thus, the layers of security might slow down the process, maybe 15 – 30 seconds – not a time efficient way to withdraw cash. Another famous barricade to secure ATM transactions is 'Facial recognition'. Facial recognitions can be faster, if done in ambient light. But face detection cameras can easily be fooled – if another person wears a 3D mask of the user and also has the credentials to access the bank account. So, face detection also won't work. Also, a user's face changes with time (naturally , due to aging), it might happen that the face detection camera is not embedded with a Machine Learning Algorithm, which shall record and update the user's face in the database, by comparing with the previously stored image. Same is the issue for voice activated systems. One of the safest methods for authentication is OTP (One-Time Password). You just get this temporary numerical PIN,

and then enter into the ATM machine and then withdraw cash. But, OTPs are only safe, if your mobile phone is not accessed by any unknown external agent. Overall, if we compare all the security measures, there will be some pros and cons for each. Thus, a 'hybrid' of many of the available methods can surely guarantee for securing transactions of the ATMs. It can be a little time taking, but shall be a near - flawless and hassle-free method to access bank accounts. We are not using Iris OR Retinal Scanner , because if someone is too close to the iris scanner (less than 2 cm distance) , the continuous radiation of IR (Infra-red) rays , shall damage the eye. The same applies for retinal scanners.

3. LITERATURE SURVEY

Well, the 'Hybrid' system can only be created if the proposed system can out show the existing ones. Let us have a look at some existing works – which are also being implemented.

S. No.	Title & Author	Year of Publication	Methodology & Techniques Used	Strengths / Advantages	Limitations
1	Enhanced security for ATM machine with OTP and Facial recognition features by Mohsin Karovaliyaa, Saifali Karediab, Sharad Ozac, Dr. D.R.Kalbanded	2015	1. Face Recognition 2. OTP (One-Time Password)	1. Helps the machine to identify users easily 2. OTP generated, can act as a random temporary PIN – no hassle to remember account specific PIN , which can be vulnerable	1. Damaged Camera shall hinder facial recognition procedure 2. OTP will only be available for a very short period of time 3. If server goes down, OTP will not be generated – user will be handicapped
2	Designing a Biometric Strategy (Fingerprint) Measure for Enhancing ATM Security in Indian E-banking System by Shimal Das , Jhunu Debbarma	2011	Fingerprint based identification	Being having unique fingerprint of every user - can safeguard transactions	1. Fingerprint scanner may get damaged by over usage 2. Can be slow to match even near 100% of fingerprint via the database
3	Enhancing ATM Security using Fingerprint and GSM Technology by V. Padmapriya , S. Pakasam	2013	1. PIN verification 2. Fingerprint Recognition technology 3. GSM modem	Sending a GSM modem generated PIN to user's mobile phone – changes for every request	1. If GSM modem is down , PIN will not be generated – user will become handicapped
4	Face Recognition Technique : Enhanced Safety Approach for ATM by Deepa Malviya	2014	Face recognition from 3 angles	Introduction of face grid algorithm strengthens the linking with PIN	If user's face is not placed precisely according to grid, then the face will not be detected – after a certain

					number of trials , it shall not allow for further scanning
5	Enhanced ATM security with OTP Based Authentication by Prashant Kumar Yadav , Akshtar Husain , Surjeet Kumar	2020	1. Face recognition (PCA – Principal Component Analysis) (LDA – Linear Discriminant Analysis) 2. Fingerprint detection 3. OTP	Random 6 digit OTP generation will authenticate user via mobile phone	1. If GSM module stops working , randomly generated OTP will not be reached to user's phone 2. OTP generated will live for a very short period of time , after which if used , will be invalidated
6	Improving Security levels in ATM using Multi Factor Authenticator by Frimpong Twum , Kofi Nti , Michael Asante	2016	1. PIN verification 2. Fingerprint based authentication	Opted the Three tier design structure – verification and authorization at every tier using database	Failing of Fingerprint scanner shall not allow the user to carry out transaction(s)
7	QR based Card-less ATM Transactions by Meenu Jacob , Nikhil Mathew , Rose Merin Jose , Seba Siby , Neethu C Sekhar	2016	QR Code	Being card less transaction , usage of ATM cards will become obsolete – handheld phones shall be used to get random PIN.	1. If QR code gets damaged / unreadable by phone's camera , transaction shall not happen 2. Random PIN generator, which is connected to GSM module – shall get deactivated on power failure
8	ATM Transaction Security Using Fingerprint / OTP by Krishna Nand Pandey , Md. Masoom , Supriya Kumari , Preeti Dhiman	2015	1. Fingerprint scanner 2. OTP verification	Fingerprint , being unique to every person – is a strong key to access account , also OTP verification via GSM module is flawless for user verification	1. Three successive wrong attempts will lock the ATM card for 24 hours 2. The glass surface of the scanner if damaged – will not read
9	An Enhanced ATM Security System using Second-Level Authentication by Muhammad-Bello B.L. , Alhassan M.E. , Ganiyu S.O.	2015	1. SMS service 2. Token device	Collaboration of SMS API and Bank database can eradicate the faults over PIN verification	Bank database , if is under maintenance OR updations – the SMS API system will not proceed for verification , as it fetches data from the database itself

10	Palm Vein Biometric Technology : An approach to upgrade security in ATM transactions by B.V. Prasanthi , S Mahboob Hussain , A.S.N. Chakravarthy , Prathyusha Kanakam	2015	1. Palm vein technology 2. Unique Identification Number (UIN)	Vein pattern in human palms are unique , unpredictable, novel to every individual – and also being contactless is a plus point from hygiene point of view	Palm veins among young people are difficult to locate by low resolution cameras – only possible if the palm is close enough , that too under enough lighting
11	Securing ATM Transactions Using QR Code based Secure PIN Authentication by Sumanth C M	2019	1. QR code (SAPQ) 2. OTP (One-time password)	1. Ensures the security of ATM transactions by making use of three-level verification. 2. It also makes SPAQ service simple and user-friendly.	1. If QR code gets damaged / unreadable by the phone's camera , transaction shall not happen. 2. OTP will only be available for a very short period of time.
12	Secure Authentication for ATM transactions using NFC technology by Divyansh Mahansaria, Uttam Kumar Roy	2019	1. OTP (One-time password) 2. NFC (Near Field Communication)	1. NFC tag doesn’t need a power source – it is passive and is simply read or written to by the powered terminal. 2. It also allows a quick and safe transaction.	1. The threats could be on the components, communication channels and the authentication protocol of the system. 2. It would be a problem if the user’s phone gets stolen.
13	Secure Card-less ATM Transactions by Khushboo Yadav, Suhani Mattas, Lipika Saini, Poonam Jindal	2020	OTP (One time password)	This system provides a three level security, first when the user's identity is verified while logging in the system, second through user-id, password and the code present in the mobile app – when entered in the ATM machine and last via the reference number.	OTP generated will live for a very short period of time , after which if used, will be invalidated
14	Design of a Customer-Centric Surveillance System for ATM Banking	2020	SMS messaging packets	It’s visible to the banking security systems/authority, and	For temporary network failure, which possesses very low latency for messaging services for

	Transactions using Remote Certification Technique by Olugbemiga Solomon POPOỌLA, Ibraheem Temitọpẹ JIMOH, Adebayọ Olusọla ADETUNMBI, Kayọde Boniface ALESE, Chukwuemeka Christian UGWU			simultaneously visible to the bank account owners	multiple users, with supports for international roaming
15	ATM Shield: Analysis of Multi Tier Security Issues of ATM in the Context of Bangladesh by Md. Raqibul Hasan Rumman, Atish Sarker, Md. Majharul Islam, Md. Imdadul Hoque, Robin Kuri, Md. Babar Ali Bhuyan, Nayeem Al-Tamzid Bhuiyan	2020	1. Fingerprint verification (Biometrics) 2. QR Code Authentication (using GSM smartphones)	1. Multilevel security at client side using cryptography algorithm and user's biometric features. 2. Biometric features cannot be easily hack because of its unique identification 3. QR code generation via mobile phone 4. At the server side the steganography algorithm is used for hiding the encrypted information.	1. If QR code gets damaged / unreadable by the phone's camera , transaction shall not happen. 2. If the fingerprint scanner gets damaged, it may read false data and the real users may not be able to access their account.
16	Towards A Secured Financial Transaction: A Multi-Factor Authentication Model by Lala, O.G., Aworinde, H.O., Ekpe, S.I.	2020	Facial Recognition (using the LDA and PCA algorithm)	No issues like wear and tear or the possibility of easily acquiring the authentication method.	Damaged cameras shall hinder facial recognition procedure along with poor lights in the room.
17	A Novel Technique for ATM Security by Image Processing by Pranesh Kulkarni, Dr. Raghavendra S.P	2019	Face Recognition Systems (FRS)	No issues like wear and tear or the possibility of easily acquiring the authentication method.	Damaged cameras shall hinder facial recognition procedure along with poor lights in the room.
18	ATM Transaction Security Using Fingerprint Recognition by Mithun Dutta,	2017	1. Fingerprint (Biometric) 2. PIN	1. Initially captured fingerprint images are	If the fingerprint scanner gets damaged, it may read false data and the real users may not be

	Kanchita Keam Psyche, Shamima Yasmin			converted to templates instead of storing anywhere which makes misuse of the system totally impossible. 2. This system is easy to install, less time consuming and mostly approved biometric methods.	able to access their account.
19	Modified SHA256 for Securing Online Transactions based on Blockchain Mechanism by Maria Rona L. Perez, Dr. Bobby Gerardo, Ruji Medina	2018	Blockchain	High assurance of data protections against intrusive users.	
20	A Study of Possible Biometric Solution to Curb Frauds in ATM Transaction by Ahmad Tasnim Siddiqui, Mohd. Muntjir	2013	1. Biometric (Fingerprint, Iris scan) 2. Face recognition	It's very authentic and cannot be copied or stolen from the user.	If there's damage in the iris / fingerprint scanner device, then matching the exact data would be a problem.
21	Enhanced security for ATM machines with OTP and Facial recognition features by Mohsin Karovaliya , Saifali Karedia , Sharad Oza , D.R. Kalbande	2015	1. Face recognition 2. OTP	LIVE facial image comparison on the basis of stored database via webcam , which will trigger afterwards to verify using OTP	1. Triggered OTP will be received by the user if network services are inactive 2. Damaged webcam will give distorted user facial image – incomparable to the one stored in the db.
22	New Approach in Biometrics to Combat the Automated Teller Machine Frauds : Facial Recognition by Priyanka Mahajan	2016	3-D Facial recognition	Facial images obtained from all the three perspectives via webcams will provide precise authentication	1. Viewing angle w.r.t. every user is different 2. Poor lighting will give dark facial images – not suitable if the user is wearing sunglasses , or any cloth covering the user's face.
23	Design of Highly Secured Automatic Teller Machine System by using Aadhar Card and Fingerprint by Abhijeet S. Kale , Sunpreet Kaur Nanda	2014	1. Fingerprint recognition 2. Mobile number verification	Aadhar card number , being unique to every citizen , is also linked to the user's mobile number as well as fingerprint scans	Aadhar card number , if linked to any other mobile number – the mobile verification shall not be carried out.

			3. Aadhar card number verification	– thus , there will be no hassle in verification , from the system's point of view.	
24	Card-less Automatic Teller Machine (ATM) : Biometric Security System Design using Human Fingerprints by Madhuri More , Sudarshan Kankal , Akshaykumar Kharat , Rupali Adhau	2018	1. Fingerprint verification 2. Mobile number verification using randomly generated code (PIN)	As fingerprint (unique to everyone) can be enough to get connected to a user's account – only mobile number verification and random code input can make the withdrawal easy	If the fingerprint sensor surface (made of glass) , is damaged , the precision will be lost in terms of scanning. Hence, verification will not be done.
25	Survey of Security of ATM Machines by Prachi More , S.D. Markande	2016	1. Fingerprint scanning 2. Facial recognition 3. RFID technology	Among all the mentioned methods , Fingerprint scanning have been proved more accurate and safer	Fingerprint sensors have a glass surface – if it gets even a scratch OR dirt gets built up , the scanner will not be able to function.
26	ATM Security Using Fingerprint Biometric Identifier: An Investigative Study by Moses Okechukwu Onye Olu , Ignatius Majesty Ezeani	2012	Fingerprint Biometric Identifier	User Authentication is improved apart from just using PIN	If the database fails / is inactive – only using PIN also won't allow the user to withdraw
27	Comparison of biometric identification methods by Csaba OTTI	2016	1. Fingerprint 2. Iris 3. Retina 4. Face (2D , 3D) 5.Hand Geometry 6. Face Heatmap 7. Vein patterns (Hand veins , Finger veins) 8. Voice	No proper / highly appreciable technology can be assumed full proof for ATM transaction security	Every technology has numerous disadvantages , it's up to the developer which to be used under what scenario keeping in mind / according to the demands

28	Automated Teller Machines in India : A Literature Review from Key Stakeholders Perspectives by Jyotiranjan Hota , Saboohi Nasim , Sasmita Mishra	2013	CRM technology	1. Leverage a 360-degree view of every customer 2. Improve Customer Retention 3. Enable Quicker Processes 4. Using Insights to Improve Sales and marketing Efforts 5. Productive personals	1. CRM is costly 2. Poor Communication
29	Biometrics and Smart Cards in Identity Management by Bart Jacobs, Erik Poll	2010	1. Biometrics 2. RFID - enabled smartcards 3. e-passports	The card can protect the information, it cannot easily be cloned, and even if a card is lost or stolen, the protection it provides remains in place.	After eavesdropping on communication between e-passports and readers, an attacker can mount a brute force attack trying out all the possible keys
30	Practical Attacks on Proximity Identification Systems by Gerhard P.Hancke	2006	RFID tokens	An attacker executing a relay attack cannot avoid causing a delay in the system.	
31	Comparison of Various Biometric Method by Rupinder Saini , Narinder Rana	2014	1. Face Biometric 2. Iris Biometric 3. Fingerprint Biometric 4. Finger Vein 5. Voice Biometric 6. Lips Biometric	Can easily identify a person in a crowd and so we can verify their identity.	1. The systems are usually less efficient if facial expressions vary. 2. Iris scanners tend to be more expensive in comparison with additional biometrics. 3. Cuts, marks transform fingerprints which often has a negative effect on performance. 4. In case an accident causes a user to lose his/her finger then it can be a problem during the verification process.

					5. May hacked with prerecorded voice messages. 6. A big smile might cause difficulty in recognition of a person with respect to the same person with a neutral appearance as before.
32	Secret data communication system using Steganography, AES and RSA by Septimiu Fabian Mare, Mircea Vladutiu, Lucian Prodan	2011	Biometric (Fingerprint)	Fingerprint biometric technique is fused with the ATM for person authentication to ameliorate the security level.	Since a fingerprint scanner only scans one section of a person's finger, it may be susceptible to error.
33	Towards understanding ATM security: a field study of real world ATM use by Alexander De Luca, Marc Langheinrich, Heinrich Hussmann	2010	PIN	Enhanced the security features which are“built in” into the authentication mechanism, i.e, the security of a system should not rely on active secure behavior of a user.	
34	A New Vision for ATM Security Management by Claudio Porretti, Denis Kolev, Raoul Lahaije	2016	GAMMA concept	They described a new vision for ATM Security Management as a gamma project that is implemented by the federated architecture of the Security Management Platforms.	
35	Improving Security Levels In Automatic Teller Machines (ATM) Using Multifactor Authentication by Frimpong Twum, Kofi Nti, Michael Asante	2016	A multifactor (PIN and Fingerprint) based authentication.	1. Security arrangement to enhance the security and safety of the ATM and its users. 2. The proposed system is a good cost effective measure for implementing a well secure ATM	

				transactions to protect ATM users from fraudsters.	
36	Cardless Automatic Teller Machine (ATM) Biometric Security System Design Using Human Fingerprints by Madhuri More, Sudarshan Kankal, Akshaykumar Kharat, Rupali Adhau	2018	1. QR Code 2. ATM PIN 3. OTP	It proposed an enhanced feature to improve the service of ATM cash withdrawal in less time with more level of security.	
37	ATM Security by Kavita Hooda	2016	Biometric (Face recognition)	1. Biometric ATM systems are highly secure as it provides authentication with the information of body parts. 2. It is a viable approach, as it is easy to maintain and operate with lower cost.	Poor lighting will give dark facial images – not suitable if the user is wearing sunglasses , or any cloth covering the user's face.
38	A Study of Possible Biometric Solution to Curb Frauds in ATM Transaction by Ahmad Tasnim Siddiqui, Mohd. Muntjir	2013	1. Biometric (Fingerprint, Iris scan) 2. Face recognition	It's very authentic and cannot be copied or stolen from the user.	If there's damage in the iris / fingerprint scanner device, then matching the exact data would be a problem.
39	Securing ATM with OTP and Biometric by Mohammed Hamid Khan	2015	1. OTP (one time password) 2. Biometric	1. By using biometric security, the alternative security will be the same as OTP. 2. Biometric will use fingerprint scanning.	Changes in the Hardware part will be required, that is one fingerprint scanner is required to be attached to ATM machines.
40	Real Time SMS-Based hashing scheme for securing financial	2011	A hash code using the customer PIN number and phone	Provides additional security layer and	Safe only until mobile phone is not accessed by

	transactions on ATM systems by Ugochukwu Onwudebelu; Olumide Longe; Sanjo Fasola; Ndidi C. Obi; Olumuyiwa B. Alaba		number. The generated hash key is then used to decrypt messages requesting for transactions from the customer.	fortify existing PIN access thus safeguarding customer accounts and account information	any unknown external agent
41	A Survey for Securing Online Payment Transaction Using Biometrics Authentication by M. Hari Priya and N. Lalithamani	2017	Biometric	Quicker transaction in the most secure way of biometrics features involved that cannot be forged	If there's damage in the iris/fingerprint scanner device, then matching the exact data would be a problem.
42	Smart ATM security system using FPR, GSM, GPS by Bharati M Nelligani, N V Uma Reddy, Nithin Awasti	2016	1. RFID cards are used as ATM card, IR sensors in order to sense the presence of the card holders and to turn on Fan and Light. 2. GPS is used to track the location in case the cash box is robbed. 3. Fingerprint is used to identify and verify authorized bank personnel.	Ensures a secured, authenticated transaction and gives an idea about the major security issues in the ATM system.	Modification in the existing ATM algorithm by having Iris Recognition, Vein Pattern Recognition, Face recognition and using Multibiometrics for authentication.
43.	Securing Cardless Automated Teller Machine Transactions Using Bimodal Authentication System By Ameh Innocent Ameh, Olayemi Mikail Olanyi & Olumide Sunday Adewale	2016	Fingerprint authentication model for a cardless ATM using Principal Component Analysis (PCA), with a PIN as a second factor of authentication.	The proposed technique would be profitable to both banks and customers in the areas of security, and eradication of the recurrent cost of card acquisition and maintenance	Failing of Fingerprint scanner shall not allow the user to carry out transaction(s)

44.	Enhanced way of securing automated teller machine to track the misusers using secure monitor tracking analysis by Jayakumar Sadhasivam, M Alamelu, R Radhika, S Ramya, K Dharani and Senthil Jayavel	2017	Scan the iris known (a part or movement of our eye) and fingerprint of the customer.	Provides a three-way security	If there's damage in the iris/fingerprint scanner device, then matching the exact data would be a problem.
45.	Securing ATM and Card Transactions using SMS-Based Security by Kevin Alex Sam, Liya Mary Antony, Reenu Xavier, Remitha Rahim	2016	Customer sends a message to the bank server before he actually performs an actual banking transaction.	Helps reduce the effects of frauds and attacks by adding a layer of security.	Consumes extra time and is burdensome because SMS has to be sent for all payments.
46.	Securing Card Transaction Against Shoulder Surfing Attack by Dr. M.P. Dale, Shruti R. Gogawale, Twinkle S. Deore	2018	System designed is based on the Microcontroller LPC 2138.	Gives more security to PIN entering process for common users.	Time consuming.
47.	Securing ATM Pins and Passwords Using Fingerprint Based Fuzzy Vault System by Sweedle, Machado, Prajyoti D'silva, Snehal D'mello, Supriya Solaskar, Priya Chaudhari	2018	Fuzzy vault system (bio-cryptosystem that combines cryptography and biometrics)	Aims to modify a fuzzy vault system to secure ATM pins and passwords with user's fingerprint data such that only the genuine user can access the pins and passwords by providing the valid fingerprint.	If there's damage in the fingerprint scanner device, then matching the exact data would be a problem.
48.	Fingerprint watermarking and steganography for ATM transaction using LSB-RSA and 3-DWT algorithm by Mayank Garg; Shashikant Gupta; Pallavi Khatri	2015	Combination of fingerprint verification methods with watermarking technology.	Provides copyright protection and authentication of digital images.	Failing of Fingerprint scanner shall not allow the user to carry out transaction(s).

4. PROPOSED ALGORITHM(S)

Explanation of Various Technologies used:

Now as we have gone through numerous articles and existing works, which are subjected to implementation5, we shall now discuss the technologies we might be including in our project.

1. **ATM (AUTOMATED TELLER MACHINE)**

   Let us just explain with a one line definition - "An electronic telecommunications device that enables customers of financial institutions to perform financial transactions, such as cash withdrawals, deposits, funds transfers, balance inquiries or account information inquiries, at any time and without the need for direct interaction with bank staff ". Though the main objective of this project is to implement new technologies to improve ATM transactions, it should also be known what an ATM actually is. Below are some pictures of the ATM and a close up view of the operational panel.



Though, there is a camera present, as one can see - it is only active for monitoring the users. The present ATM's operational panel has many disadvantages. First, the camera itself - it doesn't support facial recognition and also, the Iris scanner / Retinal scanner is not embedded. Secondly, the keypad is also unprotected - anyone can attach a low resolution camera to the floor, just above the keypad, thus, enabling hackers to read the PIN. The card insertion slot - might have a cloning machine (which looks like the insertion slot) - which can clone all the information of the Magnetic card. These are some physical anomalies one should take care when paying the next visit to any ATM. We shall not discuss the internal working of the ATM as it's out of this project's scope.

2. FINGERPRINT SCANNER

Fingerprint of every user is unique. Thus, authentication can become easy if a fingerprint scanner is installed in an ATM. It may happen, users may not have to input PIN - just put the thumb, withdraw cash - that's it. But the fingerprint scanner has some drawbacks. These scanners have a small glass surface, on which the thumb (or any other finger) should be placed. Overuse can either make some scratches on the glass surface, because of which, the infra-red light won't be able to detect / scan the fingerprint clearly. Apart from that, with respect to security concerns, these scanners are also capable of detecting non-prominent fingerprints - even from an Acetate sheet consisting of someone's fingerprints being kept on the glass surface - thus, creating a threat to the user's bank account.

3. FACIAL RECOGNITION VIA ON-BOARD CAMERA

This is a very simple stage - the user just has to align his / her face to the camera, so that the whole face is visible on the screen. The facial detection algorithm will try to detect and match the image recorded through the bank database. After the detection procedure is completed, an OTP will be sent to the user's registered mobile number (which may also be the PIN, if we are talking about Card-less transactions). But here is the crux - if more than one face is detected by the camera, the system will be temporarily locked and a message will be floating on the screen - saying, "ONE USER AT A TIME". Nowadays, 3D facial recognition is becoming a trend, which uses 3 cameras, set at different angles to read the user's face and authenticate.

4. OTP GENERATOR USING GSM MODULE

OTP (One-Time Password) has been in use for quite a long time. Whether you are performing Net-Banking, doing online shopping, creating an account in Netflix OR Amazon Prime OR Hotstar, OTP Authentication is a must. The same can be used in ATMs. Many users tend to forget their ATM PIN, resulting in changing the PIN whenever visiting the ATM. Well that's a tedious job. Rather, after they insert their Debit / Credit card, the machine will try to read the card's information and let the user get a random PIN generated by the bank server using the GSM module (modem) (which is connected over mobile networks) when the user wants to perform cash withdrawal. The random PIN can be sent to the user's registered mobile number, which is linked to the bank account. Below is the diagrammatic approach to the

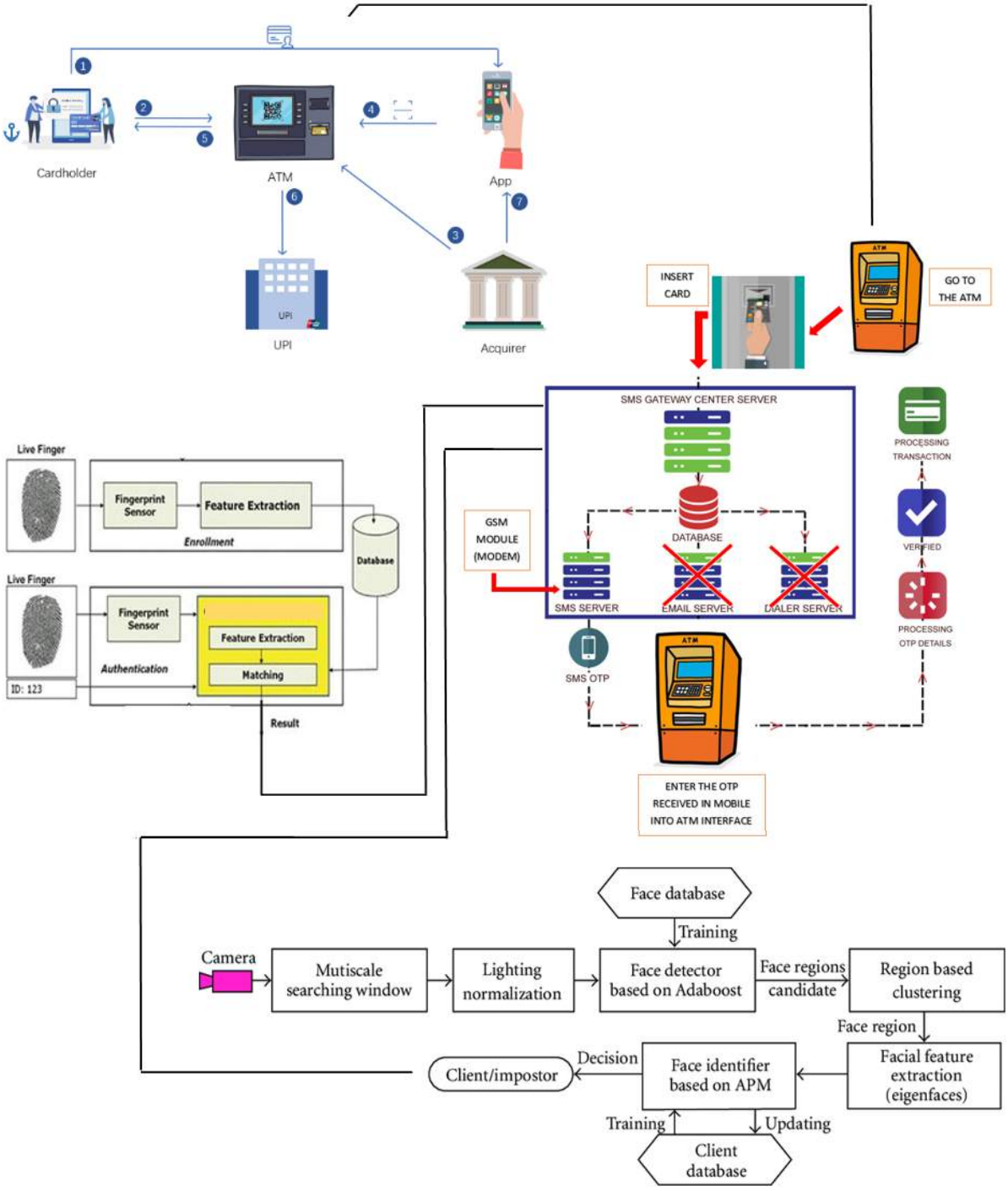
aforementioned (NOTE : We are not including Email OTP or Voice Call Authentication , as they are not time efficient methods and it can be irritating for the users as well).

5. SECURE TRANSACTION USING QR CODE

Nowadays, many Mobile applications like - Paytm, BHIM, Amazon Pay, Google PhonePe etc. have made buying goods and services easy. There is no hassle for swiping the card through the card machine, counting currency notes whether extra is not paid, etc. These instant transaction applications use the combination of QR code and UPI ID, both are unique to every person. The concept is - if someone wants to pay for a commodity, he / she will scan the QR code via the mobile camera - enabled by the application or just enter the UPI ID of the seller and then TRANSACTION IS DONE! Thus, the usage of ATM cards plays no role here. Similar (not exact) approach can be with an ATM. The user will have to scan a QR code - shown on the ATM's screen. The user, after scanning, will enter the amount to be withdrawn. The transaction is safeguarded via a PIN (can also be received by OTP, if not mentioning a fixed PIN but a random combination of fixed length number), after which the ATM cash withdrawing outlet will be ready with the cash. This method is considered one of the safest methods today because there is no need to insert the card (So no cloning of the card). The QR code is not static - it changes with time OR after every transaction - this is called **" dynamic QR code "**.

SOUBHIK SINHA PRISHITA RAJ ANISHA AGARWAL

PROPOSED ARCHITECTURE



19BIT0303 19BIT0288 19BIT0317

5. EXPERIMENTAL SETUP AND RESULT ANALYSIS

Fingerprint Scanner

ISAA Project - J Component

Team ▶ SOUBHIK SINHA [19BIT0303]
ANISHA AGARWAL [19BIT0317]
PRISHITA RAJ [19BIT0288]

Topic : Securing ATM Transactions

Component : FINGERPRINT SCANNER (RECOGNITION)

```
[ ] # Let us import all the necessary python libraries

    import cv2
    import os
    import sys
    import numpy
    import matplotlib.pyplot as plt
    from skimage.morphology import skeletonize, thin
```

```
[ ] # We need to install an important library - enhance

    !pip install enhance

    Requirement already satisfied: enhance in /usr/local/lib/python3.7/dist-packages (1.2)

[ ] from enhance import *
```

```
# Let us commence with the functions


    ''' Now , as the fingerprint scanned can be a distorted image
    (distorted because , the fingerprint scanner is installed on a
    public device - so there is an obvious reason od getting bad image),
    the images have to be enhanced '''

    def removedot(invertThin):
        temp0 = numpy.array(invertThin[:])
        temp0 = numpy.array(temp0)
        temp1 = temp0/255
        temp2 = numpy.array(temp1)
        temp3 = numpy.array(temp2)

        enhanced_img = numpy.array(temp0)
        filter0 = numpy.zeros((10,10))
        W,H = temp0.shape[:2]
        filtersize = 6

        for i in range(W - filtersize):
            for j in range(H - filtersize):
                filter0 = temp1[i:i + filtersize , j:j + filtersize]
```

```
                flag = 0
                if sum(filter0[: , 0]) == 0:
                    flag +=1
                if sum(filter0[: , filtersize - 1]) == 0:
                    flag +=1
                if sum(filter0[0 , :]) == 0:
                    flag +=1
                if sum(filter0[filtersize - 1 , :]) == 0:
                    flag +=1
                if flag > 3:
                    temp2[i:i + filtersize , j:j + filtersize] = numpy.zeros((filtersize , filtersize))

        return temp2
```

```
def get_descriptors(img):
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    img = clahe.apply(img)
    '''img = image_enhance.image_enhance(img)'''
    img = numpy.array(img, dtype=numpy.uint8)

    # Threshold
    ret, img = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)

    # Normalize to 0 and 1 range
    img[img == 255] = 1

    #Thinning
    skeleton = skeletonize(img)
    skeleton = numpy.array(skeleton, dtype=numpy.uint8)
    skeleton = removedot(skeleton)

    # Harris corners
    harris_corners = cv2.cornerHarris(img, 3, 3, 0.04)
    harris_normalized = cv2.normalize(harris_corners, 0, 255, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32FC1)
    threshold_harris = 125
```

```
    # Extract keypoints
    keypoints = []
    for x in range(0, harris_normalized.shape[0]):
      for y in range(0, harris_normalized.shape[1]):
        if harris_normalized[x][y] > threshold_harris:
          keypoints.append(cv2.KeyPoint(y, x, 1))

    # Define descriptor
    orb = cv2.ORB_create()

    # Compute descriptors
    _, des = orb.compute(img, keypoints)
    return (keypoints, des);


  def main():
    image_name = sys.argv[1]
    img1 = cv2.imread("database/" + image_name, cv2.IMREAD_GRAYSCALE)
    kp1, des1 = get_descriptors(img1)

    image_name = sys.argv[2]
    img2 = cv2.imread("database/" + image_name, cv2.IMREAD_GRAYSCALE)
    kp2, des2 = get_descriptors(img2)

    # Matching between descriptors
    bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
    matches = sorted(bf.match(des1, des2), key= lambda match:match.distance)
```

```
    # Plot keypoints
    img4 = cv2.drawKeypoints(img1, kp1, outImage=None)
    img5 = cv2.drawKeypoints(img2, kp2, outImage=None)
    f, axarr = plt.subplots(1,2)
    axarr[0].imshow(img4)
    axarr[1].imshow(img5)
    plt.show()

    # Plot matches
    img3 = cv2.drawMatches(img1, kp1, img2, kp2, matches, flags=2, outImg=None)
    plt.imshow(img3)
    plt.show()

    # Calculate score
    score = 0;
    for match in matches:
      score += match.distance
    score_threshold = 33

    if score/len(matches) < score_threshold:
      print("FINGERPRINT MATCHED !")
    else:
      print("FINGERPRINT NOT MATCHED !")
```

```
  if __name__ == "__main__":
    try:
      main()
    except:
      raise

    # Due to unaccessibility of hardware components (the fingerprint scanner)
    # we will not be able to provide input for our program
    # hence , that shall be left for the future work of the project
```

Facial recognition

```
'''
ISAA - J Component Project

TEAM : SOUBHIK SINHA (19BIT0303)
       ANISHA AGARWAL (19BIT0317)
       PRISHITA RAJ (19BIT0288)

Topic : Securing ATM Transactions
'''

''' Component : 3D FACE - RECOGNITION '''

from Detector import main_app
from create_classifier import train_classifer
from create_dataset import start_capture
import tkinter as tk
from tkinter import font as tkfont
from tkinter import messagebox,PhotoImage

#from PIL import ImageTk, Image
#from gender_prediction import emotion,ageAndgender

names = set()


class MainUI(tk.Tk):

    def __init__(self, *args, **kwargs):
        tk.Tk.__init__(self, *args, **kwargs)
        global names
        with open("nameslist.txt", "r") as f:
            x = f.read()
            z = x.rstrip().split(" ")
            for i in z:
                names.add(i)

        self.title_font = tkfont.Font(family='Helvetica', size=16, weight="bold")
```

SOUBHIK SINHA PRISHITA RAJ ANISHA AGARWAL

```
        self.title("Face Recognizer")
        self.resizable(False, False)
        self.geometry("500x250")
        self.protocol("WM_DELETE_WINDOW", self.on_closing)
        self.active_name = None
        container = tk.Frame(self)
        container.grid(sticky="nsew")
        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)
        self.frames = {}

        for F in (StartPage, PageOne, PageTwo, PageThree, PageFour):
            page_name = F.__name__
            frame = F(parent=container, controller=self)
            self.frames[page_name] = frame
            frame.grid(row=0, column=0, sticky="nsew")

        self.show_frame("StartPage")



    def show_frame(self, page_name):
            frame = self.frames[page_name]
            frame.tkraise()
```

```
    def on_closing(self):

        if messagebox.askokcancel("Quit", "Are you sure?"):
            global names
            f =  open("nameslist.txt", "a+")
            for i in names:
                    f.write(i+" ")
            self.destroy()



class StartPage(tk.Frame):

        def __init__(self, parent, controller):
            tk.Frame.__init__(self, parent)
            self.controller = controller

            #load = Image.open("homepagepic.png")
            #load = load.resize((250, 250), Image.ANTIALIAS)

            render = PhotoImage(file='homepagepic.png')
            img = tk.Label(self, image=render)
            img.image = render
            img.grid(row=0, column=1, rowspan=4, sticky="nsew")
            label = tk.Label(self, text="        Home Page        ", font=self.controller.title_font,fg="#263942")
            label.grid(row=0, sticky="ew")
            button1 = tk.Button(self, text="   Add a User  ", fg="#ffffff", bg="#263942",command=lambda: self.controller.show_frame("PageOne"))
            button2 = tk.Button(self, text="   Check a User  ", fg="#ffffff", bg="#263942",command=lambda: self.controller.show_frame("PageTwo"))
            button3 = tk.Button(self, text="Quit", fg="#263942", bg="#ffffff", command=self.on_closing)
            button1.grid(row=1, column=0, ipady=3, ipadx=7)
            button2.grid(row=2, column=0, ipady=3, ipadx=2)
            button3.grid(row=3, column=0, ipady=3, ipadx=32)
```

```
        def on_closing(self):
            if messagebox.askokcancel("Quit", "Are you sure?"):
                global names
                with open("nameslist.txt", "w") as f:
                    for i in names:
                        f.write(i + " ")
                self.controller.destroy()



class PageOne(tk.Frame):
    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        self.controller = controller
        tk.Label(self, text="Enter the name", fg="#263942", font='Helvetica 12 bold').grid(row=0, column=0, pady=10, padx=5)
        self.user_name = tk.Entry(self, borderwidth=3, bg="lightgrey", font='Helvetica 11')
        self.user_name.grid(row=0, column=1, pady=10, padx=10)
        self.buttoncanc = tk.Button(self, text="Cancel", bg="#ffffff", fg="#263942", command=lambda: controller.show_frame("StartPage"))
        self.buttonext = tk.Button(self, text="Next", fg="#ffffff", bg="#263942", command=self.start_training)
        self.buttoncanc.grid(row=1, column=0, pady=10, ipadx=5, ipady=4)
        self.buttonext.grid(row=1, column=1, pady=10, ipadx=5, ipady=4)


    def start_training(self):
        global names
        if self.user_name.get() == "None":
            messagebox.showerror("Error", "Name cannot be 'None'")
            return
        elif self.user_name.get() in names:
            messagebox.showerror("Error", "User already exists!")
            return
        elif len(self.user_name.get()) == 0:
            messagebox.showerror("Error", "Name cannot be empty!")
            return
```

19BIT0303 19BIT0288 19BIT0317

```
        name = self.user_name.get()
        names.add(name)
        self.controller.active_name = name
        self.controller.frames["PageTwo"].refresh_names()
        self.controller.show_frame("PageThree")


class PageTwo(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        global names
        self.controller = controller
        tk.Label(self, text="Select user", fg="#263942", font='Helvetica 12 bold').grid(row=0, column=0, padx=10, pady=10)
        self.buttoncanc = tk.Button(self, text="Cancel", command=lambda: controller.show_frame("StartPage"), bg="#ffffff", fg="#263942")
        self.menuvar = tk.StringVar(self)
        self.dropdown = tk.OptionMenu(self, self.menuvar, *names)
        self.dropdown.config(bg="lightgrey")
        self.dropdown["menu"].config(bg="lightgrey")
        self.buttonext = tk.Button(self, text="Next", command=self.nextfoo, fg="#ffffff", bg="#263942")
        self.dropdown.grid(row=0, column=1, ipadx=8, padx=10, pady=10)
        self.buttoncanc.grid(row=1, ipadx=5, ipady=4, column=0, pady=10)
        self.buttonext.grid(row=1, ipadx=5, ipady=4, column=1, pady=10)

    def nextfoo(self):
        if self.menuvar.get() == "None":
            messagebox.showerror("ERROR", "Name cannot be 'None'")
            return
        self.controller.active_name = self.menuvar.get()
        self.controller.show_frame("PageFour")
```

```
    def refresh_names(self):
        global names
        self.menuvar.set('')
        self.dropdown['menu'].delete(0, 'end')
        for name in names:
            self.dropdown['menu'].add_command(label=name, command=tk._setit(self.menuvar, name))

class PageThree(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        self.controller = controller
        self.numimglabel = tk.Label(self, text="Number of images captured = 0", font='Helvetica 12 bold', fg="#263942")
        self.numimglabel.grid(row=0, column=0, columnspan=2, sticky="ew", pady=10)
        self.capturebutton = tk.Button(self, text="Capture Data Set", fg="#ffffff", bg="#263942", command=self.capimg)
        self.trainbutton = tk.Button(self, text="Train The Model", fg="#ffffff", bg="#263942",command=self.trainmodel)
        self.capturebutton.grid(row=1, column=0, ipadx=5, ipady=4, padx=10, pady=20)
        self.trainbutton.grid(row=1, column=1, ipadx=5, ipady=4, padx=10, pady=20)

    def capimg(self):
        self.numimglabel.config(text=str("Captured Images = 0 "))
        messagebox.showinfo("INSTRUCTIONS", "We will Capture 300 pic of your Face.")
        x = start_capture(self.controller.active_name)
        self.controller.num_of_images = x
        self.numimglabel.config(text=str("Number of images captured = "+str(x)))

    def trainmodel(self):
        if self.controller.num_of_images < 300:
            messagebox.showerror("ERROR", "No enough Data, Capture at least 300 images!")
            return

        train_classifer(self.controller.active_name)
        messagebox.showinfo("SUCCESS", "The modele has been successfully trained!")
        self.controller.show_frame("PageFour")
```

```
class PageFour(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        self.controller = controller

        label = tk.Label(self, text="Face Recognition", font='Helvetica 16 bold')
        label.grid(row=0,column=0, sticky="ew")
        button1 = tk.Button(self, text="Face Recognition", command=self.openwebcam, fg="#ffffff", bg="#263942")

        #button2 = tk.Button(self, text="Emotion Detection", command=self.emot, fg="#ffffff", bg="#263942")
        #button3 = tk.Button(self, text="Gender and Age Prediction", command=self.gender_age_pred, fg="#ffffff", bg="#263942")

        button4 = tk.Button(self, text="Go to Home Page", command=lambda: self.controller.show_frame("StartPage"), bg="#ffffff", fg="#263942")
        button1.grid(row=1,column=0, sticky="ew", ipadx=5, ipady=4, padx=10, pady=10)

        #button2.grid(row=1,column=1, sticky="ew", ipadx=5, ipady=4, padx=10, pady=10)
        #button3.grid(row=2,column=0, sticky="ew", ipadx=5, ipady=4, padx=10, pady=10)

        button4.grid(row=1,column=1, sticky="ew", ipadx=5, ipady=4, padx=10, pady=10)

    def openwebcam(self):
        main_app(self.controller.active_name)

    # def gender_age_pred(self):
    #  ageAndgender()

    # def emot(self):
    #  emotion()
```

```
app = MainUI()
app.iconphoto(False, tk.PhotoImage(file='icon.ico'))
app.mainloop()
```

```
'''
ISAA - J Component Project

TEAM : SOUBHIK SINHA (19BIT0303)
       ANISHA AGARWAL (19BIT0317)
       PRISHITA RAJ (19BIT0288)

Topic : 'Securing ATM Transactions'

Component : 3D - Facial Recognition (Classifier)

'''


import numpy as np
from PIL import Image
import os, cv2



# Method to train custom classifier to recognize face
def train_classifer(name):
    # Read all the images in custom data-set
    path = os.path.join(os.getcwd()+"/data/"+name+"/")

    faces = []
    ids = []
    labels = []
    pictures = {}


    # Store images in a numpy format and ids of the user on the same index in imageNp and id lists

    for root,dirs,files in os.walk(path):
            pictures = files
```

```
    for pic in pictures :

            imgpath = path+pic
            img = Image.open(imgpath).convert('L')
            imageNp = np.array(img, 'uint8')
            id = int(pic.split(name)[0])
            #names[name].append(id)
            faces.append(imageNp)
            ids.append(id)

    ids = np.array(ids)

    #Train and save classifier
    clf = cv2.face.LBPHFaceRecognizer_create()
    clf.train(faces, ids)
    clf.write("./data/classifiers/"+name+"_classifier.xml")
```

```
'''
ISAA - J Component Project

TEAM : SOUBHIK SINHA (19BIT0303)
       ANISHA AGARWAL (19BIT0317)
       PRISHITA RAJ (19BIT0288)

Topic : 'Securing ATM Transactions'

Component : 3D - Facial Recognition (Dataset Creation)

'''

import cv2
import os

def start_capture(name):
        path = "./data/" + name
        num_of_images = 0
        detector = cv2.CascadeClassifier("./data/haarcascade_frontalface_default.xml")
        try:
            os.makedirs(path)
        except:
            print('Directory Already Created')
        vid = cv2.VideoCapture(0)
        while True:

            ret, img = vid.read()
            new_img = None
            grayimg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            face = detector.detectMultiScale(image=grayimg, scaleFactor=1.1, minNeighbors=5)
            for x, y, w, h in face:
                cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 0), 2)
                cv2.putText(img, "Face Detected", (x, y-5), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255))
                cv2.putText(img, str(str(num_of_images)+" images captured"), (x, y+h+20), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255))
                new_img = img[y:y+h, x:x+w]
            cv2.imshow("FaceDetection", img)
```

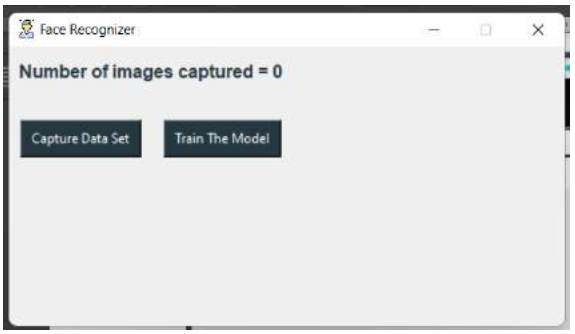
```
            key = cv2.waitKey(1) & 0xFF


            try :
                cv2.imwrite(str(path+"/"+str(num_of_images)+name+".jpg"), new_img)
                num_of_images += 1
            except :

                pass
            if key == ord("q") or key == 27 or num_of_images > 310:
                break
        cv2.destroyAllWindows()
        return num_of_images
```

```
'''
ISAA - J Component Project

TEAM : SOUBHIK SINHA (19BIT0303)
       ANISHA AGARWAL (19BIT0317)
       PRISHITA RAJ (19BIT0288)

Topic : 'Securing ATM Transactions'

Component : 3D - Facial Recognition (Face Detector)

'''

import cv2
from time import sleep
from PIL import Image

def main_app(name):

        face_cascade = cv2.CascadeClassifier('./data/haarcascade_frontalface_default.xml')
        recognizer = cv2.face.LBPHFaceRecognizer_create()
        recognizer.read(f"./data/classifiers/{name}_classifier.xml")
        cap = cv2.VideoCapture(0)
        pred = 0
        while True:
            ret, frame = cap.read()
            #default_img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray,1.3,5)

            for (x,y,w,h) in faces:


                roi_gray = gray[y:y+h,x:x+w]

                id,confidence = recognizer.predict(roi_gray)
                confidence = 100 - int(confidence)
```

```
                pred = 0
                if confidence > 50:
                    #if u want to print confidence level
                            #confidence = 100 - int(confidence)
                            pred += +1
                            text = name.upper()
                            font = cv2.FONT_HERSHEY_PLAIN
                            frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
                            frame = cv2.putText(frame, text, (x, y-4), font, 1, (0, 255, 0), 1, cv2.LINE_AA)

                else:
                            pred += -1
                            text = "UnknownFace"
                            font = cv2.FONT_HERSHEY_PLAIN
                            frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
                            frame = cv2.putText(frame, text, (x, y-4), font, 1, (0, 0,255), 1, cv2.LINE_AA)

            cv2.imshow("image", frame)


            if cv2.waitKey(20) & 0xFF == ord('q'):
                print(pred)
                if pred > 0 :
                    dim =(124,124)
                    img = cv2.imread(f".\\data\\{name}\\{pred}{name}.jpg", cv2.IMREAD_UNCHANGED)
                    resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
                    cv2.imwrite(f".\\data\\{name}\\50{name}.jpg", resized)
                    Image1 = Image.open(f".\\2.png")

                    # make a copy the image so that the
                    # original image does not get affected
                    Image1copy = Image1.copy()
                    Image2 = Image.open(f".\\data\\{name}\\50{name}.jpg")
                    Image2copy = Image2.copy()
```

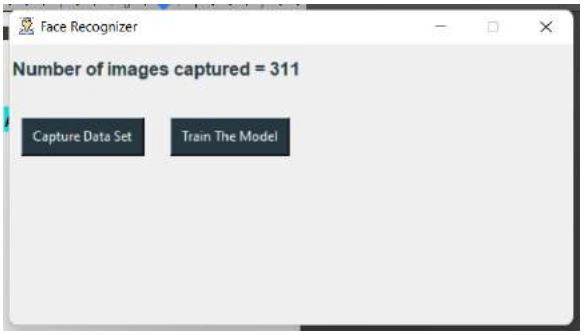
```
            # paste image giving dimensions
            Image1copy.paste(Image2copy, (195, 114))

            # save the image
            Image1copy.save("end.png")
            frame = cv2.imread("end.png", 1)

            cv2.imshow("Result",frame)
            cv2.waitKey(5000)
        break


cap.release()
cv2.destroyAllWindows()
```

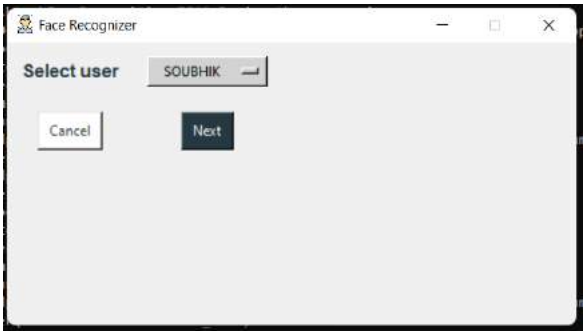
OUTPUT

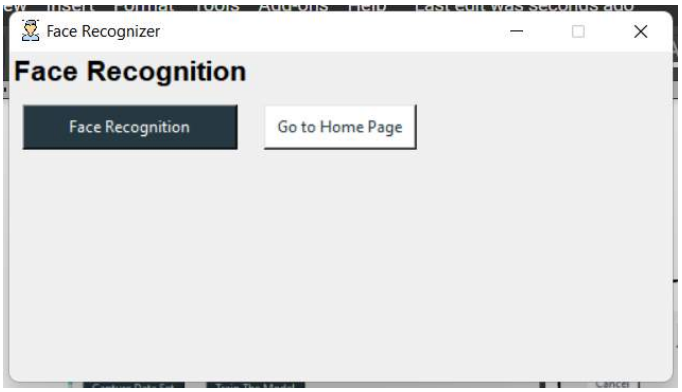


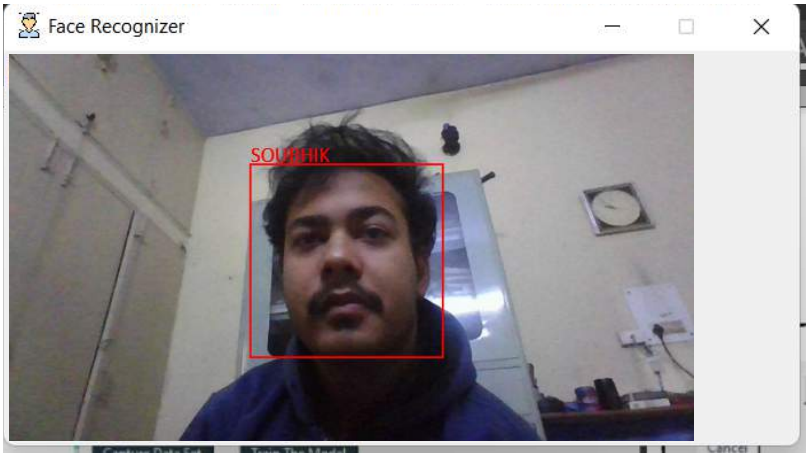
NOTE : At this stage , the facial recognition application will take 300+ images. Thus , giving the user the provision for turning her / his head left AND right , up AND down. Hence , the application can be derived as 3D - Facial Recognition application.

SOUBHIK SINHA PRISHITA RAJ ANISHA AGARWAL









19BIT0303 19BIT0288 19BIT0317

OTP (via GSM Module)

ISAA Project - J Component

Team ▶ PRISHITA RAJ [19BIT0288]
SOUBHIK SINHA [19BIT0303]
ANISHA AGARWAL [19BIT0317]

Topic : Securing ATM Transactions

Component : GSM MODULE (OTP - GENERATION)

```
# Importing the necessary library needed

import math, random
```

+ Code + Text

```
# Now , there is no library which has a ready functionality
# to generate OTP , thus we have to write a function to do it

# As we know , that the GSM module is responsible for generating the OTP
# so , for the time being , let us consider this local system as the
# GSM module
```

```
def OTP_generation() :

    # Though we have seen OTP as a string of numbers only,
    # Here , we will be generating apha-numeric OTPs

    # For generating a random , Alpha-Numeric OTP ,
    # there should be a range of characters to choose from

    char_choice = 'abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    generated_OTP = ""

    # Length is important , afterall , we should not create an OTP which is more
    # than 12 characters long (Some experts say) - LONGER THE OTP , LONGER THE TIME IT WILL TAKE
    # FOR AUTHENTICATION

    OTP_length = len(char_choice)

    # Lets create the OTP
    for i in range(8) :
      # 8 characters should be enough
      generated_OTP += char_choice[math.floor(random.random() * OTP_length)]

    return generated_OTP

# Being a function , it has to be called from a God-father function
if __name__ == "__main__" :

    print("8 character OTP Generation : ", OTP_generation())
```

```
8 character OTP Generation :  R2TF55sd
```

QR CODE SCANNER

jupyter QR CODE GENERATION Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Raw NBConvert

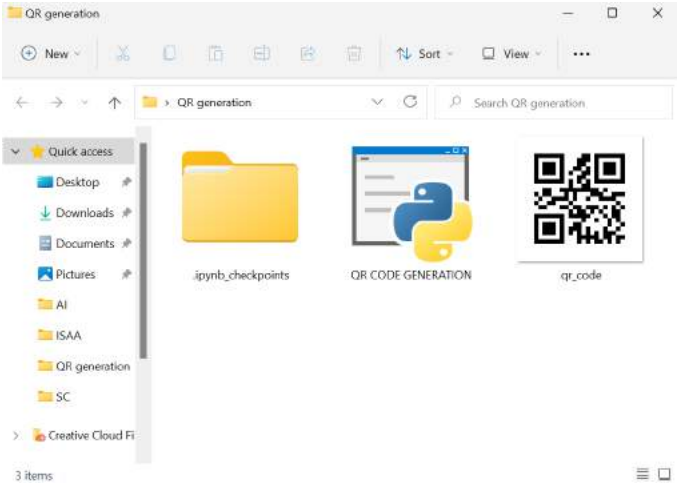
```
PRISHITA RAJ   19BIT0288
ANISHA AGARWAL 19BIT0317
SOUBHIK SINHA  19BIT0303
```

In [3]:

```
import pyqrcode # Imported pyqrcode Module
import random   #Imported random Module

data =  random.randint(1000,9999)# Declaring the Variables
image = pyqrcode.create(data) # Creating a Function
image.png("qr_code.png",scale=8) # Saving the Image
```

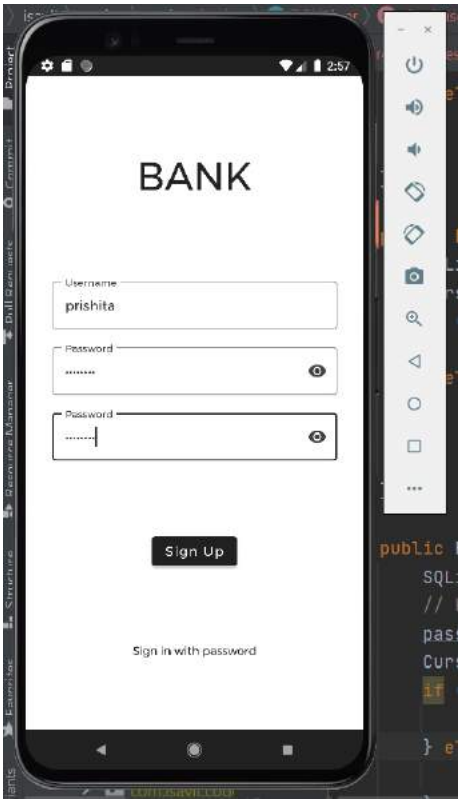
In [ ]:

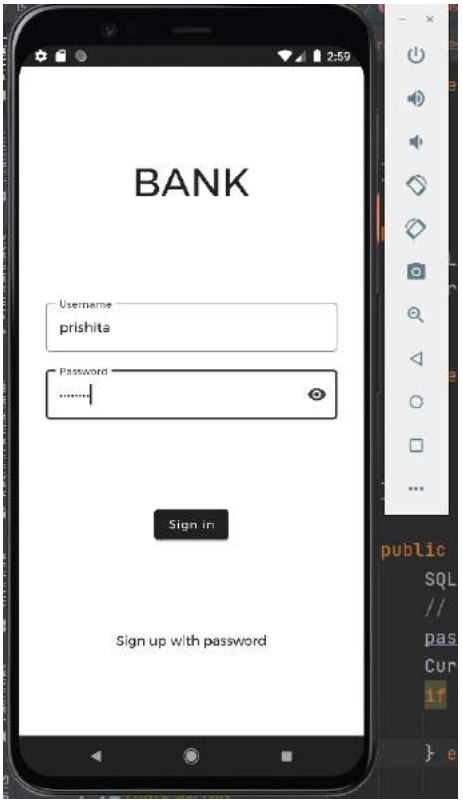
Generated QR Code:



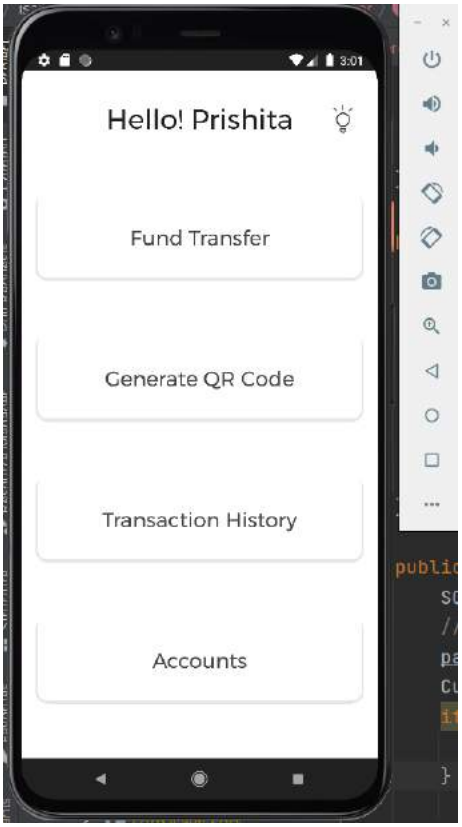
Bank Application:

Made in Android Studio & Below are the screenshots of the emulator.

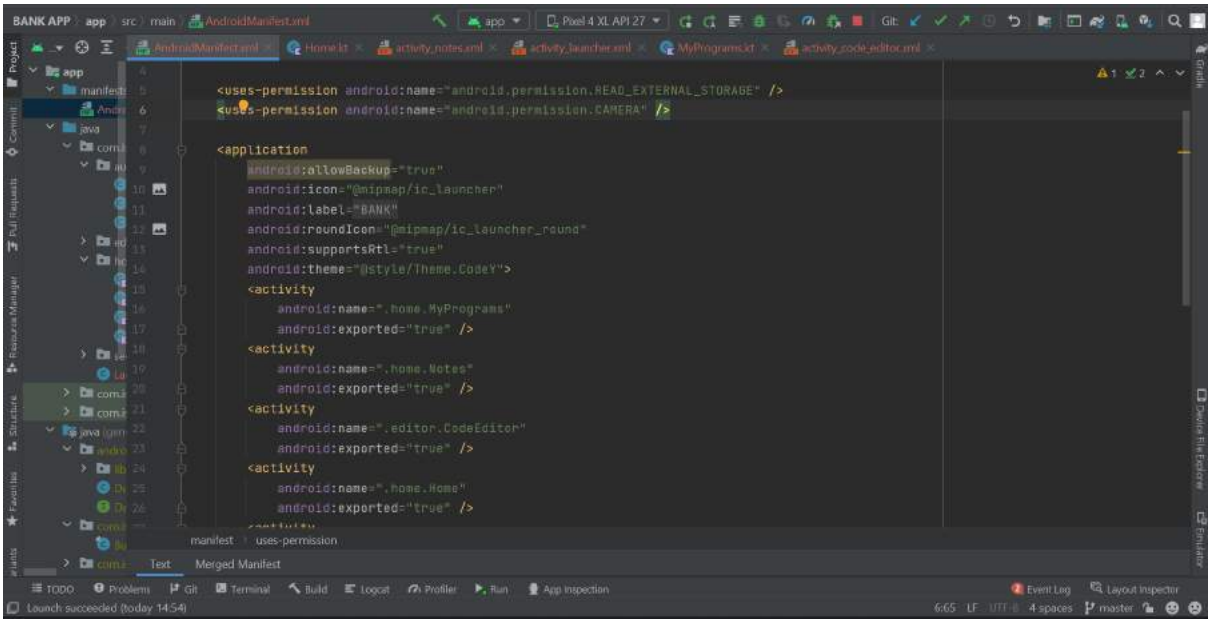




SOUBHIK SINHA PRISHITA RAJ ANISHA AGARWAL

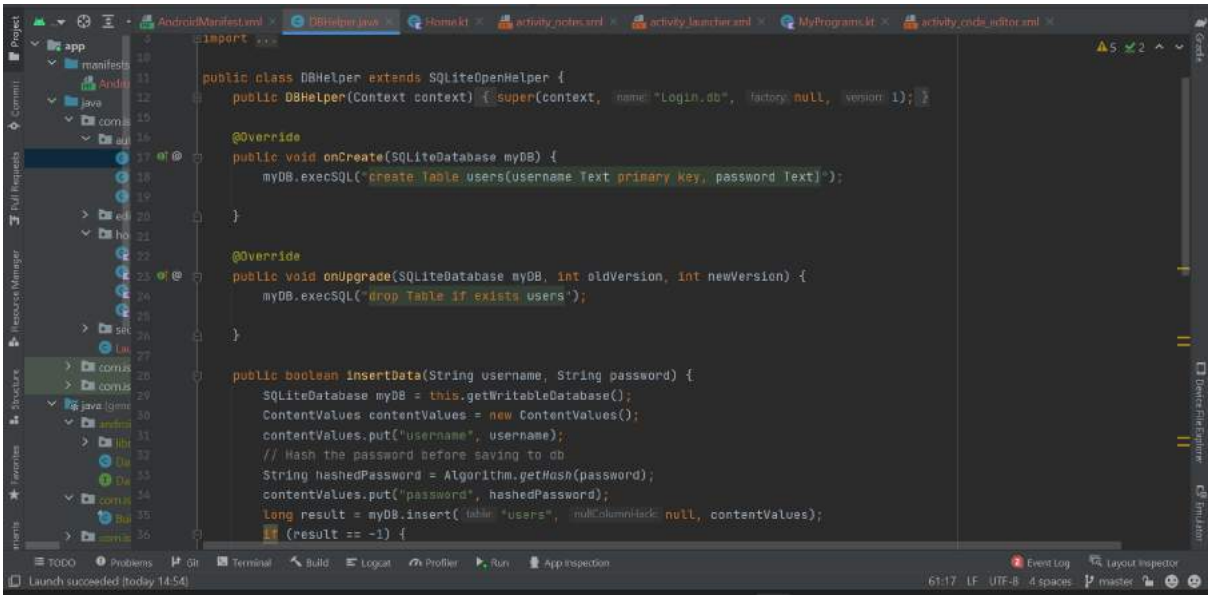


Code:

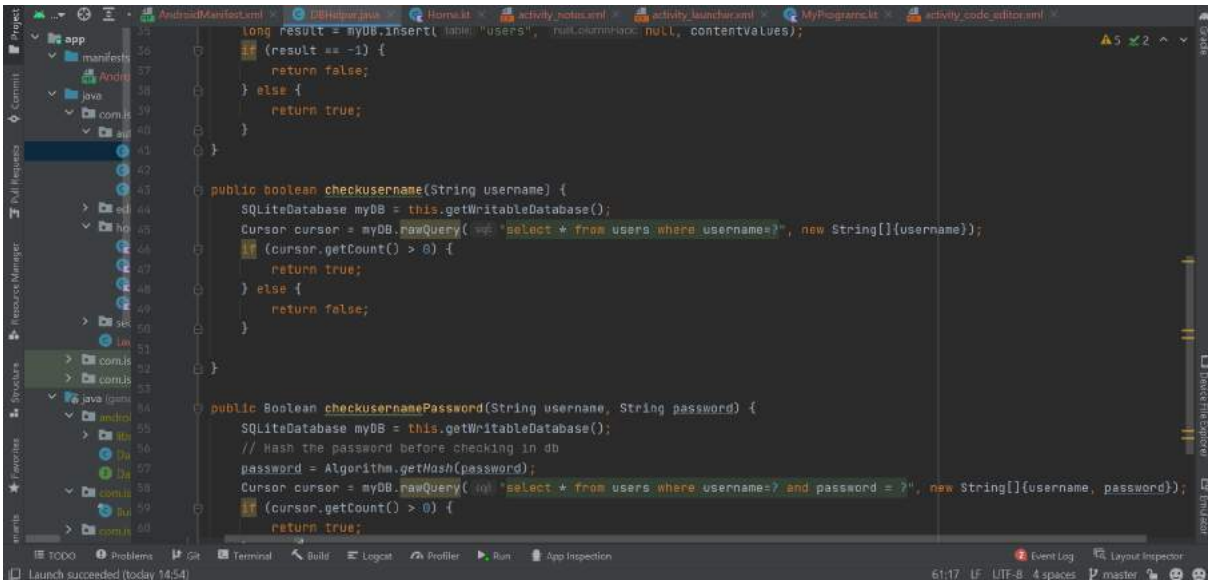


19BIT0303 19BIT0288 19BIT0317

SOUBHIK SINHA PRISHITA RAJ ANISHA AGARWAL



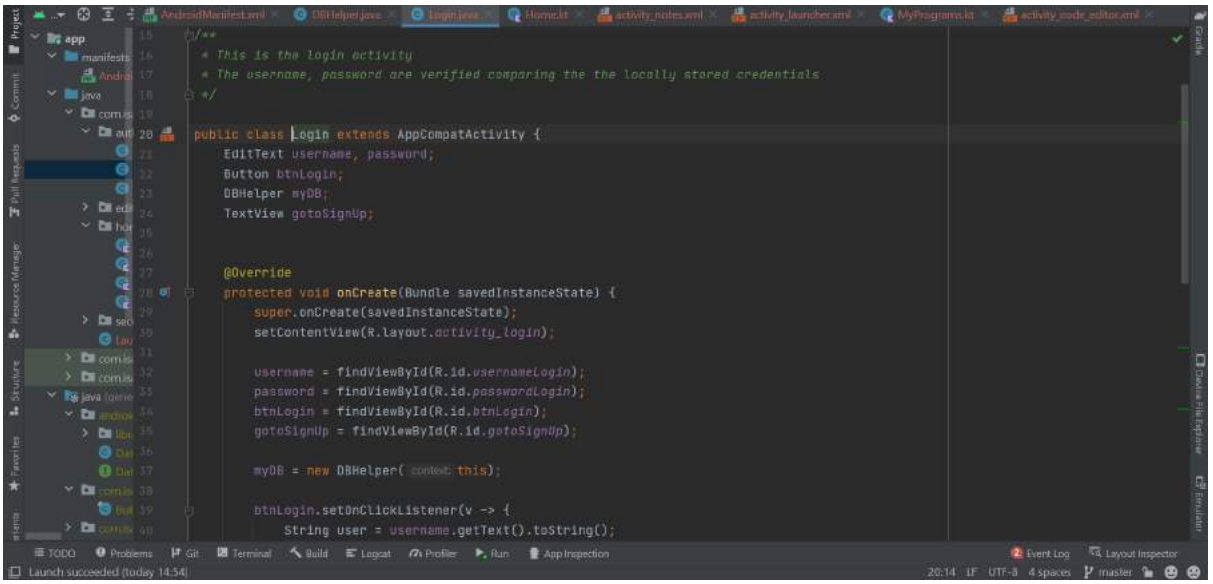
```
import ...

public class DBHelper extends SQLiteOpenHelper {
    public DBHelper(Context context) { super(context, "Login.db", null, 1); }

    @Override
    public void onCreate(SQLiteDatabase myDB) {
        myDB.execSQL("create Table users(username Text primary key, password Text)");

    }

    @Override
    public void onUpgrade(SQLiteDatabase myDB, int oldVersion, int newVersion) {
        myDB.execSQL("drop Table if exists users");

    }

    public boolean insertData(String username, String password) {
        SQLiteDatabase myDB = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("username", username);
        // Hash the password before saving to db
        String hashedPassword = Algorithm.getHash(password);
        contentValues.put("password", hashedPassword);
        long result = myDB.insert( table: "users", nullColumnHack: null, contentValues);
        if (result == -1) {
```



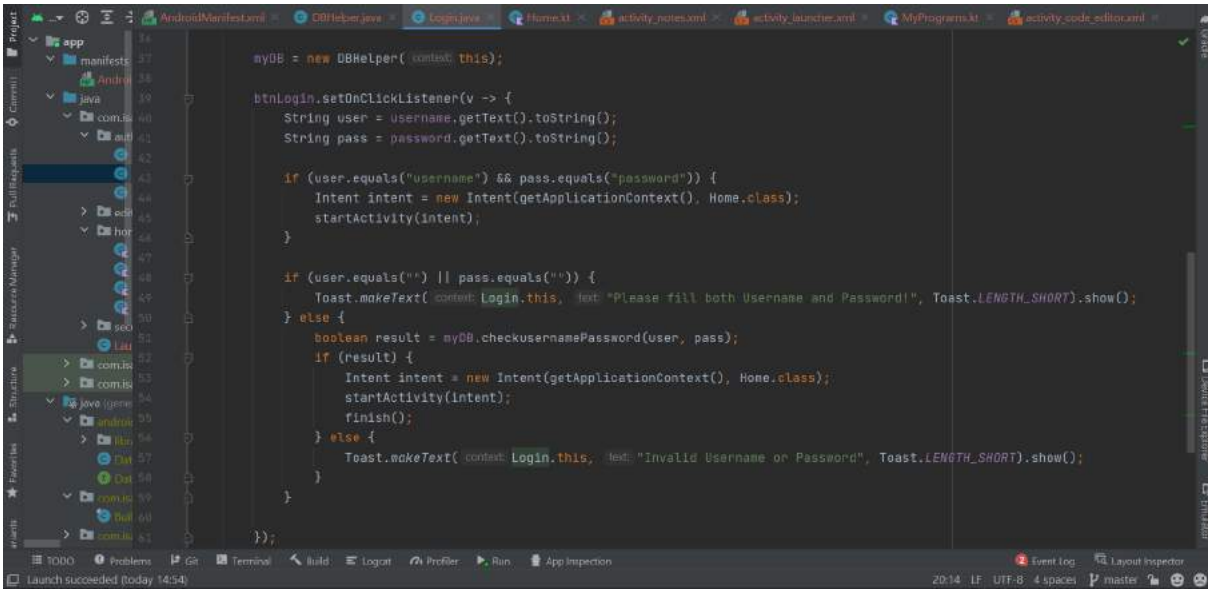
```
        long result = myDB.insert( table: "users", nullColumnHack: null, contentValues);
        if (result == -1) {
            return false;
        } else {
            return true;
        }
    }

    public boolean checkusername(String username) {
        SQLiteDatabase myDB = this.getWritableDatabase();
        Cursor cursor = myDB.rawQuery( sql: "select * from users where username=?", new String[]{username});
        if (cursor.getCount() > 0) {
            return true;
        } else {
            return false;
        }

    }

    public Boolean checkusernamePassword(String username, String password) {
        SQLiteDatabase myDB = this.getWritableDatabase();
        // Hash the password before checking in db
        password = Algorithm.getHash(password);
        Cursor cursor = myDB.rawQuery( sql: "select * from users where username=? and password = ?", new String[]{username, password});
        if (cursor.getCount() > 0) {
            return true;
```

19BIT0303 19BIT0288 19BIT0317

SOUBHIK SINHA PRISHITA RAJ ANISHA AGARWAL



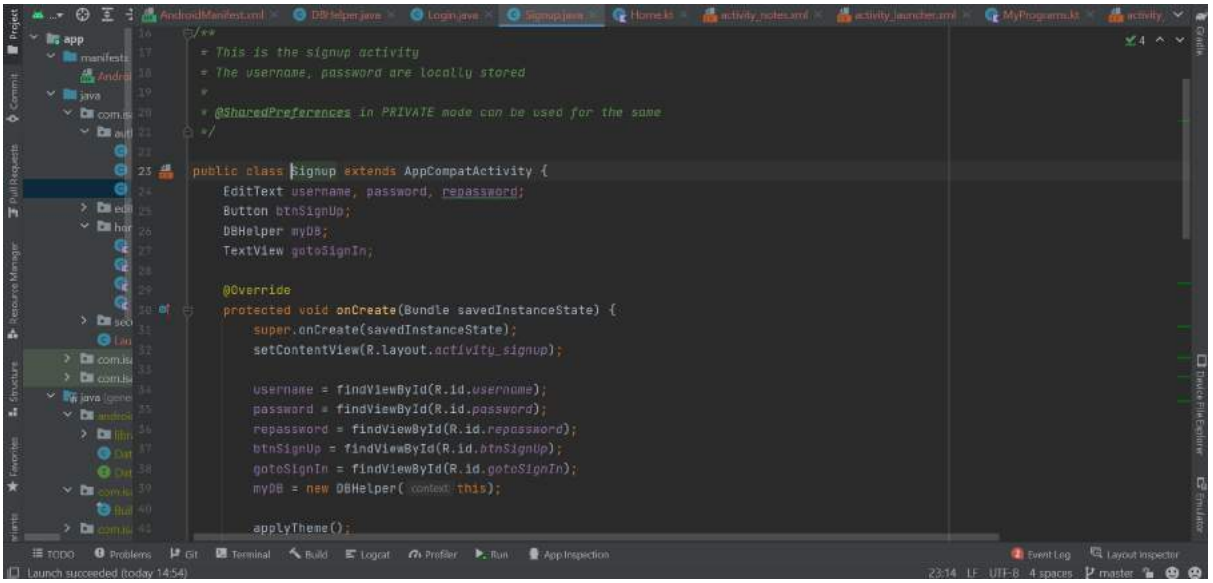
```
/**
 * This is the login activity
 * The username, password are verified comparing the the locally stored credentials
 */

public class Login extends AppCompatActivity {
    EditText username, password;
    Button btnLogin;
    DBHelper myDB;
    TextView gotoSignUp;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        username = findViewById(R.id.usernameLogin);
        password = findViewById(R.id.passwordLogin);
        btnLogin = findViewById(R.id.btnLogin);
        gotoSignUp = findViewById(R.id.gotoSignUp);

        myDB = new DBHelper( context: this);

        btnLogin.setOnClickListener(v -> {
            String user = username.getText().toString();
```



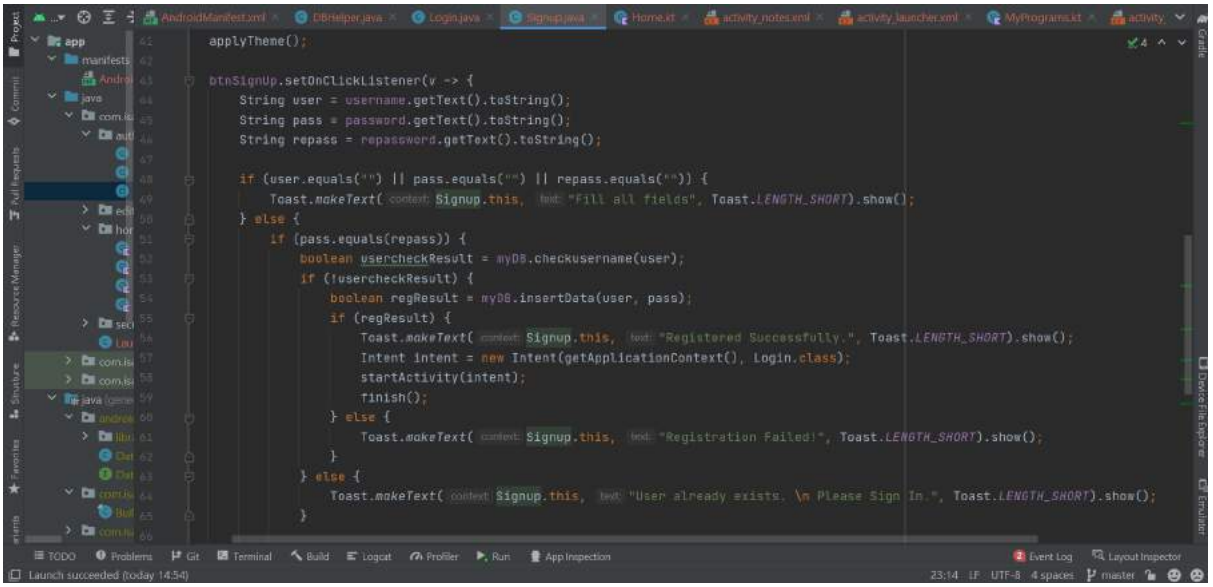
```
        myDB = new DBHelper( context: this);

        btnLogin.setOnClickListener(v -> {
            String user = username.getText().toString();
            String pass = password.getText().toString();

            if (user.equals("username") && pass.equals("password")) {
                Intent intent = new Intent(getApplicationContext(), Home.class);
                startActivity(intent);
            }

            if (user.equals("") || pass.equals("")) {
                Toast.makeText( context: Login.this, text: "Please fill both Username and Password!", Toast.LENGTH_SHORT).show();
            } else {
                boolean result = myDB.checkusernamePassword(user, pass);
                if (result) {
                    Intent intent = new Intent(getApplicationContext(), Home.class);
                    startActivity(intent);
                    finish();
                } else {
                    Toast.makeText( context: Login.this, text: "Invalid Username or Password", Toast.LENGTH_SHORT).show();
                }
            }

        });
```

19BIT0303 19BIT0288 19BIT0317

SOUBHIK SINHA PRISHITA RAJ ANISHA AGARWAL



```
/**
 * This is the signup activity
 * The username, password are locally stored
 *
 * @SharedPreferences in PRIVATE mode can be used for the same
 */

public class Signup extends AppCompatActivity {
    EditText username, password, repassword;
    Button btnSignUp;
    DBHelper myDB;
    TextView gotoSignIn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);

        username = findViewById(R.id.username);
        password = findViewById(R.id.password);
        repassword = findViewById(R.id.repassword);
        btnSignUp = findViewById(R.id.btnSignUp);
        gotoSignIn = findViewById(R.id.gotoSignIn);
        myDB = new DBHelper( context: this);

        applyTheme();
```



```
    applyTheme();

    btnSignUp.setOnClickListener(v -> {
        String user = username.getText().toString();
        String pass = password.getText().toString();
        String repass = repassword.getText().toString();

        if (user.equals("") || pass.equals("") || repass.equals("")) {
            Toast.makeText( context: Signup.this, text: "Fill all fields", Toast.LENGTH_SHORT).show();
        } else {
            if (pass.equals(repass)) {
                boolean usercheckResult = myDB.checkusername(user);
                if (!usercheckResult) {
                    boolean regResult = myDB.insertData(user, pass);
                    if (regResult) {
                        Toast.makeText( context: Signup.this, text: "Registered Successfully.", Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(getApplicationContext(), Login.class);
                        startActivity(intent);
                        finish();
                    } else {
                        Toast.makeText( context: Signup.this, text: "Registration Failed!", Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText( context: Signup.this, text: "User already exists. \n Please Sign In.", Toast.LENGTH_SHORT).show();
                }
```

19BIT0303 19BIT0288 19BIT0317

6. CONCLUSION AND FUTURE ENHANCEMENTS

Earlier , the Camera installed / embedded on the ATM interface / console was only used for monitoring. But now , after the introduction of 3D-Facial Recognition , ATM transactions have become more secure , by detecting & enabling only one-user at a time. Moreover , the introduction of Finger-Print Scanner , is a reliable authentication tool , by precisely scanning the finger-prints , and removing any noise & disturbances - for quick and correct authentication. QR code is quite a new method , though exponentially adopted in many industries and applications , the banking industry may take some time to make it time efficient , although we have mentioned an efficient technique for the same. OTP generation using GSM Module , can be a rival to the QR code , as going through 2 mandatory checks and one-optional authentication , will definitely take less time , while maintaining the level of security , rather than opting for the usage of all the 4 modules , individually. The only shortcoming of using so many modules , all at one interface , is their synchronization with the central database , because one change can block the user from any activity. Future work shall include the implementation of an actual ATM interface and a Mobile Application (for interacting with the ATM interface) , embedding all the aforementioned modules , and testing on some real datasets.

7. LIST OF REFERENCES

[1] Mohsin Karovaliyaa, Saifali Karediab, Sharad Ozac, Dr. D.R.Kalbanded (2015) , Enhanced security for ATM machine with OTP and Facial recognition features

[2] Shimal Das , Jhunu Debbarma (2011) , Designing a Biometric Strategy (Fingerprint) Measure for Enhancing ATM Security in Indian E-banking System

[3] V. Padmapriya , S. Pakasam (2013) , Enhancing ATM Security using Fingerprint and GSM Technology

[4] Deepa Malviya (2014) , Face Recognition Technique : Enhanced Safety Approach for ATM

[5] Prashant Kumar Yadav , Akshtar Husain , Surjeet Kumar (2020) , Enhanced ATM security with OTP Based Authentication

[6] Frimpong Twum , Kofi Nti , Michael Asante (2016) , Improving Security levels in ATM using Multi Factor Authenticator

[7] Meenu Jacob , Nikhil Mathew , Rose Merin Jose , Seba Siby , Neethu C Sekhar (2016) , QR based Card-less ATM Transactions

[8] Krishna Nand Pandey , Md. Masoom , Supriya Kumari , Preeti Dhiman (2015) , ATM Transaction Security Using Fingerprint / OTP

[9] Muhammad-Bello B.L. , Alhassan M.E. , Ganiyu S.O. (2015) , An Enhanced ATM Security System using Second-Level Authentication

[10] B.V. Prasanthi , S Mahboob Hussain , A.S.N. Chakravarthy , Prathyusha Kanakam (2015) , Palm Vein Biometric Technology : An approach to upgrade security in ATM transactions

[11] Sumanth C M (2019) , Securing ATM Transactions Using QR Code based Secure PIN Authentication

[12] Divyansh Mahansaria, Uttam Kumar Roy (2019) , Secure Authentication for ATM transactions using NFC technology

[13] Khushboo Yadav, Suhani Mattas, Lipika Saini, Poonam Jindal (2020) , Secure Card-less ATM Transactions

[14] Olugbemiga Solomon POPOỌLA, Ibraheem Temitọpẹ JIMOH, Adebayọ Olusọla ADETUNMBI, Kayọde Boniface ALESE, Chukwuemeka Christian UGWU (2020) , Design of a Customer-Centric Surveillance System for ATM Banking Transactions using Remote Certification Technique

[15] Md. Raqibul Hasan Rumman, Atish Sarker, Md. Majharul Islam, Md. Imdadul Hoque, Robin Kuri, Md. Babar Ali Bhuyan, Nayeem Al-Tamzid Bhuiyan (2020) , ATM Shield: Analysis of Multi Tier Security Issues of ATM in the Context of Bangladesh

[16] Lala, O.G., Aworinde, H.O., Ekpe, S.I. (2020) , Towards A Secured Financial Transaction: A Multi-Factor Authentication Model

[17] Pranesh Kulkarni, Dr. Raghavendra S.P (2019) , A Novel Technique for ATM Security by Image Processing

[18] Mithun Dutta, Kanchita Keam Psyche, Shamima Yasmin (2017) , ATM Transaction Security Using Fingerprint Recognition

[19] Maria Rona L. Perez, Dr. Bobby Gerardo, Ruji Medina (2018) , Modified SHA256 for Securing Online Transactions based on Blockchain Mechanism

[20] Ahmad Tasnim Siddiqui, Mohd. Muntjir (2013) , A Study of Possible Biometric Solution to Curb Frauds in ATM Transaction

[21] Mohsin Karovaliya , Saifali Karedia , Sharad Oza , D.R. Kalbande (2015) , Enhanced security for ATM machines with OTP and Facial recognition features

[22] Priyanka Mahajan (2016) , New Approach in Biometrics to Combat the Automated Teller Machine Frauds : Facial Recognition

[23] Abhijeet S. Kale , Sunpreet Kaur Nanda (2014) , Design of Highly Secured Automatic Teller Machine System by using Aadhar Card and Fingerprint

[24] Madhuri More , Sudarshan Kankal , Akshaykumar Kharat , Rupali Adhau (2018) , Card-less Automatic Teller Machine (ATM) : Biometric Security System Design using Human Fingerprints

[25] Prachi More , S.D. Markande (2016) , Survey of Security of ATM Machines

[26] Moses Okechukwu Onye Olu , Ignatius Majesty Ezeani (2012) , ATM Security Using Fingerprint Biometric Identifier: An Investigative Study

[27] Csaba OTTI (2016) , Comparison of biometric identification methods

[28] Jyotiranjan Hota , Saboohi Nasim , Sasmita Mishra (2013) , Automated Teller Machines in India : A Literature Review from Key Stakeholders Perspectives

[29] Bart Jacobs, Erik Poll (2010) , Biometrics and Smart Cards in Identity Management

[30] Gerhard P.Hancke (2006) , Practical Attacks on Proximity Identification Systems

[31] Rupinder Saini , Narinder Rana (2014) , Comparison of Various Biometric Method

[32] Septimiu Fabian Mare, Mircea Vladutiu, Lucian Prodan (2011) , Secret data communication system using Steganography, AES and RSA

[33] Alexander De Luca, Marc Langheinrich, Heinrich Hussmann (2010) , Towards understanding ATM security: a field study of real world ATM use

[34] Claudio Porretti, Denis Kolev, Raoul Lahaije (2016) , A New Vision for ATM Security Management

[35] Frimpong Twum, Kofi Nti, Michael Asante (2016) , Improving Security Levels In Automatic Teller Machines (ATM) Using Multi Factor Authentication

[36] Madhuri More, Sudarshan Kankal, Akshaykumar Kharat, Rupali Adhau (2016) , Cardless Automatic Teller Machine (ATM) Biometric Security System Design Using Human Fingerprints

[37] Kavita Hooda (2016) , ATM Security

[38] Ahmad Tasnim Siddiqui, Mohd. Muntjir (2013) , A Study of Possible Biometric Solution to Curb Frauds in ATM Transaction

[39] Mohammed Hamid Khan (2015) , Securing ATM with OTP and Biometric

[40] Ugochukwu Onwudebelu; Olumide Longe; Sanjo Fasola; Ndidi C. Obi; Olumuyiwa B. Alaba (2011) , Real Time SMS-Based hashing scheme for securing financial transactions on ATM systems

[41] M. Hari Priya and N. Lalithamani (2017) , A Survey for Securing Online Payment Transaction Using Biometrics Authentication

[42] Bharati M Nelligani, N V Uma Reddy, Nithin Awasti (2016) , Smart ATM security system using FPR, GSM, GPS

[43] Ameh Innocent Ameh, Olayemi Mikail Olanyi & Olumide Sunday Adewale (2016) , Securing Cardless Automated Teller Machine Transactions Using Bimodal Authentication System

[44] Jayakumar Sadhasivam, M Alamelu, R Radhika, S Ramya, K Dharani and Senthil Jayavel (2017) , Enhanced way of securing automated teller machine to track the misusers using secure monitor tracking analysis

[45] Kevin Alex Sam, Liya Mary Antony, Reenu Xavier, Remitha Rahim (2016) , Securing ATM and Card Transactions using SMS-Based Security

[46] Dr. M.P. Dale, Shruti R. Gogawale, Twinkle S. Deore (2018) , Securing Card Transaction Against Shoulder Surfing Attack

[47] Sweedle, Machado, Prajyoti D'silva, Snehal D'mello, Supriya Solaskar, Priya Chaudhari (2018) , Securing ATM Pins and Passwords Using Fingerprint Based Fuzzy Vault System

[48] Mayank Garg; Shashikant Gupta; Pallavi Khatri (2015) , Fingerprint watermarking and steganography for ATM transaction using LSB-RSA and 3-DWT algorithm

==========