CHAPTER4

METHODOLOGY, SYSTEM TESTING, IMPLEMENTATION

4.1 METHODOLOGY

This study seeks to develop a real-time facial recognition attendance system for Kantanka Financial Co-operative Society using deep learning algorithms. Traditional attendance systems often rely on manual input or identity cards, both of which are prone to errors, impersonation, and time theft. To address these issues, this project introduces a smart, automated solution capable of verifying and recording attendance through facial recognition.

Since real-time recognition demands quick processing and accurate identification, deep learning provides a suitable approach. Deep neural networks, particularly convolutional neural networks (CNNs), are effective in learning facial patterns and features from images. These models improve over time by training on more facial data, making them ideal for dynamic organizational environments.

This system aims to classify input faces as either recognized (present) or unrecognized (absent/unauthorized) by comparing them against pre-registered employee datasets. The choice of a deep learning model over traditional algorithms is justified by its superior accuracy in image-based classification tasks and its ability to adapt to different lighting conditions, facial expressions, and angles. Furthermore, its layered architecture allows it to extract deeper features for more robust face detection and verification.

4.1.1Dataset

The dataset for this study is the publicly available **Labeled Faces in the Wild (LFW)** dataset developed by the University of Massachusetts. It is accessible through various machine learning repositories and is widely used for benchmarking facial recognition systems. The dataset contains over 13,000 labeled images of faces collected from the web, with variations in pose, lighting, expression, and background. Each image is labeled with the name of the person pictured, and many individuals have multiple images in the dataset.

Due to the nature of real-time facial recognition, this dataset is suitable for training and testing the deep learning model for the attendance system. The variety in lighting and angles helps simulate real-world scenarios likely to be encountered in organizational environments such as Kantanka Financial Co-operative Society. Additionally, the dataset provides a solid foundation for building and evaluating the accuracy of the recognition system before deployment with locally collected staff images.

4.1.2 Data Preprocessing

In this project, image data is captured in real-time using the webcam. The face recognition model handles necessary normalization and preparation for facial comparison internally, ensuring the images are properly processed before they are used for recognition. The process includes transforming the captured images into a suitable format for accurate matching with stored user data. This step is crucial for ensuring that the system performs efficiently in real-time attendance tracking without requiring extensive manual preprocessing.

4.1.3 Feature Selection

To optimize the accuracy and efficiency of the facial recognition system, a feature selection method based on relevance scoring will be implemented. This process will analyze key facial landmarks (such as the eyes, nose, and mouth) and evaluate their contribution to distinguishing between individuals. By utilizing statistical techniques to assess the importance of these features, we can ensure that the model focuses on the most discriminative attributes. This will enhance the system's ability to handle variations in pose, lighting, and individual appearance, making it more robust and adaptable to real-time attendance tracking.

4.1.4 Programming environment

The system is implemented in visual studio code using Python 3 on a machine equipped with an Intel i3, and 8GB of RAM. Python is well-suited for the project due to its strong support for image processing and facial recognition through libraries such as OpenCV and face recognition. These libraries provide the necessary tools for facial recognition processing directly from the webcam feed.

The implementation uses a robust framework for detecting and recognizing faces, ensuring accurate identification by leveraging efficient image processing techniques. The system integrates seamlessly with the webcam for real-time face capture, making it an ideal solution for attendance tracking.

4.2 TESTING OF THE NEW SYSTEM

4.2.1 Unit Testing:

Unit testing ensures that each functional component of the system behaves as expected. It helps catch bugs early, improve code reliability, and promote maintainable development. This study adopts unit testing practices to verify the correctness of core operations and modules. It also aligns with Olan's (n.d) emphasis on quality and correctness in software design.

4.2.2 Integration Testing:

Integration testing focuses on verifying that individual modules or components interact correctly when combined. It ensures that data flows smoothly across the system and that different parts of the application work together as intended. This type of testing helps uncover interface mismatches, broken data links, and communication issues that unit testing might miss. It is especially useful in systems like ours, where multiple operations such as image capture, user interaction, and face verification must function seamlessly together

4.2.3 Functional Testing:

Functional testing is a black box testing method that verifies software behavior against specified requirements. It uses test scenarios to simulate real-world user interactions, ensuring the system functions correctly in each case. This enhances system reliability, reduces risk by identifying issues before deployment, and improves the overall user experience. Functional testing plays a key role in confirming that the software meets user expectations and behaves as intended (Functional Testing, 2022).

4.2.4 Performance Testing :

Performance testing evaluates how well the system performs under expected or high workloads. It measures key aspects like response time, processing speed, and resource usage to ensure the application remains efficient and responsive. This type of testing is crucial for identifying bottlenecks and ensuring the software remains stable under pressure. For a system involving real-time image capture and facial recognition, performance testing ensures that recognition remains fast and accurate without delays or crashes.

4.2.5 Acceptance Testing:

Acceptance testing is defined as a type of software testing that evaluates whether a system meets its business and user requirements. It is the final stage before the software is released to production. An acceptance test is a formal description of the behavior of a software product, generally expressed as an example or a usage scenario. Acceptance testing ensures user validation as software functions as the user expect, providing the confidence that it meets their needs and expectation. It serves as a final quality check before deployment as it acts as the ultimate validation step. Acceptance testing also combats the likelihood of costly post release fixes and customer dissatisfaction, hence saving time and costs.

4.2.6 Visual (Live) Testing:
Live testing will involve a manual assessment procedure where testers interact directly with the graphical user interface of the facial recognition attendance system. This includes examining GUI components such as buttons and image displays, evaluating camera feed responsiveness, and testing navigation between login and registration windows. The process ensures usability, accessibility, and system compatibility across different hardware setups. Testers will also assess how smoothly the system identifies users and handles failed recognition attempts. This hands-on approach ensures the interface meets quality expectations, enhances user experience, and promotes overall system acceptance.

4.2.7 Security Testing:
Security testing will assess the measures implemented to protect user data captured by the facial recognition attendance system. This includes verifying the use of access controls to restrict unauthorized use, ensuring secure handling and storage of facial data, and evaluating the system for vulnerabilities. Penetration testing will also be conducted to identify and address potential threats such as unauthorized access to the database or manipulation of attendance logs. These tests ensure the system maintains data confidentiality, integrity, and security across all user interactions.

4.3 IMPLEMENTATION OF A NEW SYSTEM

Getting a group of people to learn and use a new system can be challenging. Introducing a new application and its procedures can have a big impact on an organization, so it’s important to choose the right implementation strategy. System implementation is a crucial part of the development process, and there are four main strategies to consider: parallel, pilot, phased out, and direct implementation. To pick the best approach, the team needs to carefully evaluate each method and decide which one fits the situation best.

4.3.1 Parallel Implementation

To ensure a smooth rollout and give room for real-time validation, the facial recognition attendance system will be deployed alongside the current method of attendance for a period. This temporary dual approach allows the institution to test the accuracy and stability of the new system without disrupting existing routines.

Setting up the system involves installing facial recognition software and connecting it with webcam-enabled devices on-site. The software runs in the background while the traditional attendance method continues as usual, making it easy to compare both systems and resolve any issues early on.

During this time, attendance data from both sources are collected and monitored simultaneously. This helps verify that the system accurately matches faces with registered profiles and ensures that records are reliable and time-stamped correctly. Minor inconsistencies can be spotted and corrected before the system becomes the primary method.

Users, especially administrators and instructors will be given short training sessions to understand how to use the system, track logs, and resolve any issues. Their early feedback will play a big role in making improvements.

Performance and usage will be monitored throughout the trial phase. Any lags, misreads, or system flaws will be documented and used to fine-tune the software. Only after confirming the system works reliably in the real-world setting will the transition be made permanent.

4.3.2 Phased Implementation

In a phased implementation approach, the facial recognition attendance system will be deployed step-by-step, starting with a specific user group or feature set and expanding until the entire system is operational. This method allows for careful testing and minimizes the risk of widespread disruption. The following outlines the stages involved in the phased implementation:

Firstly, the **planning phase** focuses on setting clear goals and timelines for each stage of deployment. Key factors such as the intended user base, hardware availability, and system complexity are considered. A detailed strategy will be developed, outlining the specific tasks, deadlines, and milestones for each phase of the rollout.

Next, during the **execution phase**, the system will be introduced in smaller, manageable steps. It will first be implemented in select classrooms or departments to test its effectiveness in real-world conditions. The core functionalities such as capturing attendance through facial recognition will be activated initially, and further features or user groups will be added incrementally. Sufficient training will be provided for both faculty and administrative staff, ensuring they are equipped to use the system efficiently.

As the system rolls out, **feedback and iteration** will play a crucial role. Continuous input from end users will be collected to identify issues, assess system performance, and improve user experience. The system will be refined based on real-time observations and suggestions, ensuring that it meets the dynamic needs of its users.

Finally, in the **completion and expansion phase**, the system will gradually extend to other departments or larger user bases, ensuring that it can scale effectively. Transition management will be a priority, and any challenges or obstacles faced during the rollout will be addressed promptly. The goal is to ensure smooth integration of the facial recognition system across the institution, fully replacing the old method of attendance tracking once all issues have been resolved.

4.3.3 Pilot Implementation

"In a direct cutover strategy, the new system replaces the old one on a designated turn-on date" (Laudon & Laudon, 2020, p. 412). Before the full-scale deployment of the facial recognition attendance system, a **pilot implementation** will be carried out in a controlled environment, such as a single department or classroom, to assess the system's effectiveness, user acceptance, and potential issues. The following steps are part of the pilot implementation:

Pilot Site Selection: The first step is to carefully select the pilot site for deployment. Factors such as the department size, technical readiness, and willingness to participate are considered. Involvement from key stakeholders including faculty, IT staff, and administrative personnel is crucial to ensure the site is suitable for testing the system and gathering meaningful feedback.

Pilot Planning: A detailed implementation strategy is developed for the pilot phase, outlining the goals, timeline, resources, and success criteria. The scope of the pilot, including which features will be tested and the number of users involved, is defined. Clear guidelines are set for participants to ensure the pilot runs smoothly and the evaluation process is well-structured.

Pilot Deployment: The facial recognition attendance system is deployed at the selected pilot site according to the plan. Users receive training on how to use the system effectively, and technical support is available to address any issues. Throughout the pilot, system performance is continuously monitored, and feedback from users both faculty and students is collected to assess usability, reliability, and any challenges encountered.

Evaluation and Decision-Making: After the pilot deployment, an evaluation is conducted to assess the system's success based on the predefined criteria. Key metrics such as user satisfaction, accuracy of attendance tracking, system performance, and ease of integration into existing processes are analyzed. The feedback and performance data will guide decisions on refining the system, fixing issues, and determining whether the system is ready for a broader rollout.

4.3.4 Direct Implentation

Direct Implementation involves a swift transition to the facial recognition attendance system without a gradual phase-in period, ensuring all users start using the system immediately after deployment. The following steps are part of the direct implementation process:

Readiness and Preparation Evaluation: The first step is to ensure all prerequisites are met, including system configuration, data integration, and user training. A thorough evaluation of the organization's preparedness is conducted to identify any potential challenges or barriers that could affect the smooth deployment of the system.

Deployment and Training: The facial recognition attendance system is made available to all targeted users, and comprehensive training is provided to ensure they are proficient in using the system effectively. This dual approach of deployment and training aims to minimize disruptions and ensure users can seamlessly transition to using the new system.

Monitoring and Support: Close monitoring of the deployment process is essential to identify any issues that arise and provide prompt resolution. Continuous support is available to users to address technical challenges and assist with any questions or difficulties encountered during the initial stages of usage.

Evaluation and Optimization: After the system is fully implemented, an evaluation is conducted to assess its performance and effectiveness. Feedback is gathered from users and stakeholders to identify areas for improvement and optimization. This process ensures the system remains aligned with user needs and is continuously refined based on real-world usage.

4.4 System Documentation

4.4.1 About the system

Real-Time Facial Recognition Attendance System for Kantanka Financial Co-Operative Society using Deep Learning Algorithms

The Real-Time Facial Recognition Attendance System for Kantanka Financial Co-Operative Society utilizes advanced deep learning algorithms to automate and streamline attendance tracking. This system employs sophisticated deep learning techniques, specifically designed to process and analyze facial features, enabling accurate and reliable recognition of individuals for attendance purposes.

The system operates in real-time, capturing facial data via a camera interface and comparing it with pre-registered images in the database. Upon successful recognition, the system logs attendance automatically, eliminating the need for manual entry and reducing the potential for errors or fraud. The facial recognition model is built using a deep learning framework that ensures high precision in detecting and verifying identities even in varying lighting conditions or angles.

The modular design of the system includes components for facial data preprocessing, recognition, and real-time logging, as well as user-friendly reporting interfaces. This setup ensures that the system can be easily integrated into existing infrastructure at Kantanka Financial Co-Operative Society. The system is highly adaptable, offering scalability and customization to meet the specific needs of the organization. With its advanced functionality, the system ensures efficient attendance tracking while enhancing security and reducing administrative overhead.

4.4.2 User Access Level

To ensure that data is secure, and that the system is managed effectively, the facial recognition attendance system for Kantanka Financial Co-Operative Society incorporates various user roles with different levels of access. The roles are defined as follows:

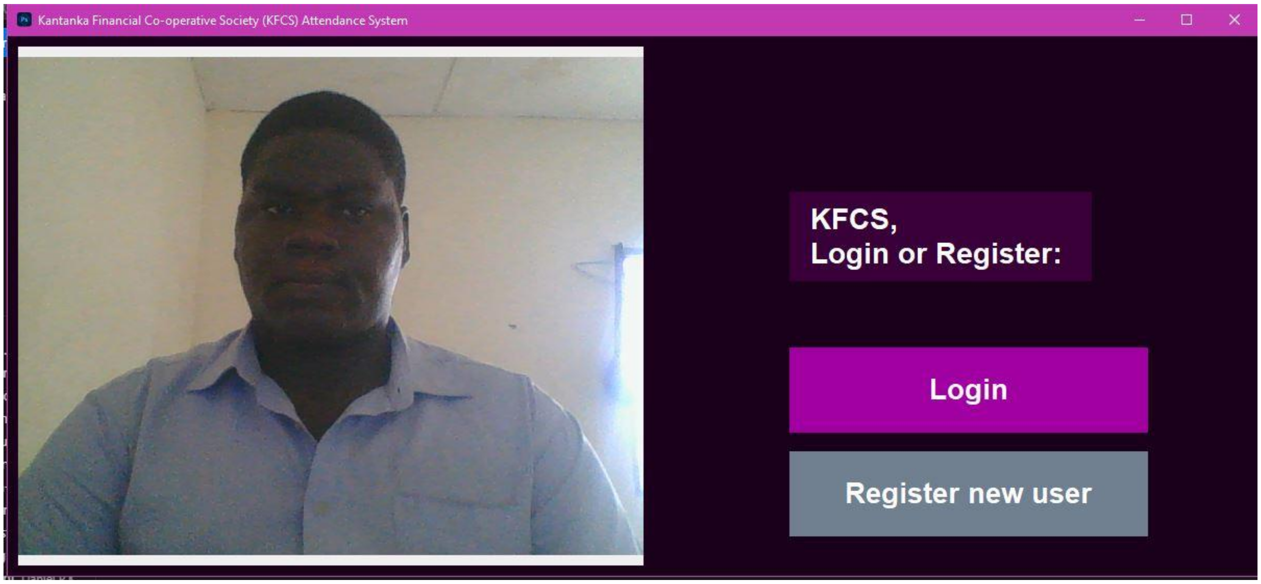
1. **Administrator:** Administrators have full access to all system functionalities, including user management, configuration settings, and system maintenance. They are responsible for setting up the system, managing user permissions, configuring database settings, and ensuring that the system operates smoothly.
2. **Attendance Manager:** This role is responsible for monitoring attendance data. They can access reports, view real-time attendance logs, and generate summaries for the management team. They may also oversee the registration of new users and ensure the accuracy of the attendance data.
3. **Employee/User:** Employees or regular users can only access their own attendance records. They can register for the system, update their facial data (if necessary), and view their attendance history. They cannot access administrative or system settings, ensuring their data remains private and secure.

By segmenting access according to role, the system ensures that sensitive data is only available to those with the proper permissions, maintaining both security and efficiency in tracking attendance.

4.4.2 Getting Started

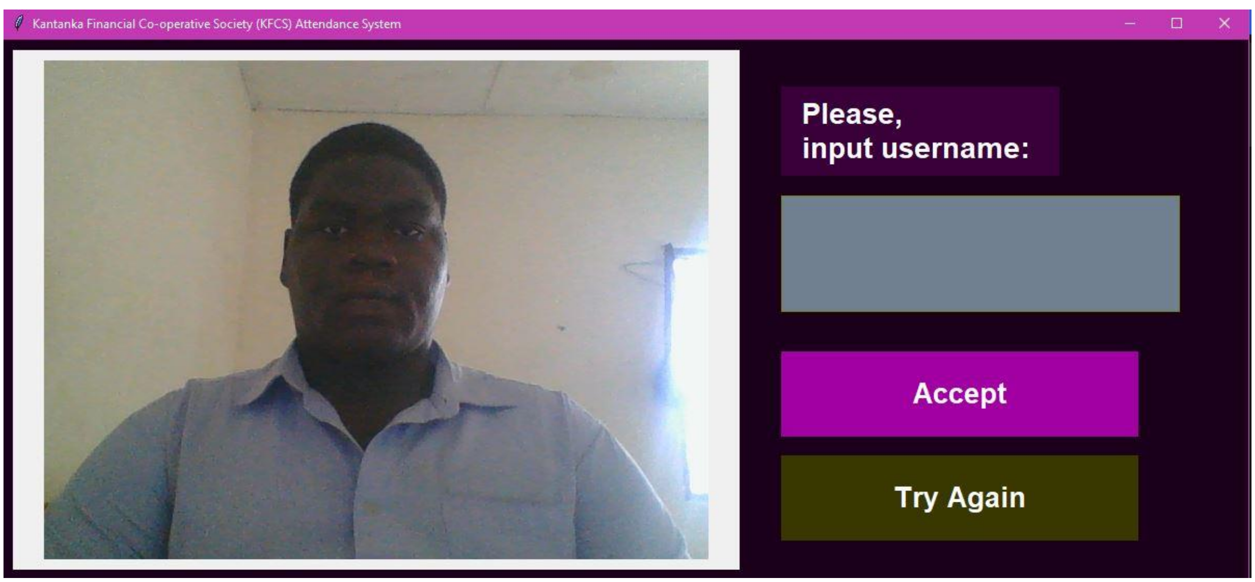
To use the attendance system and have a pleasant and error free experience, there are a few things to note and some procedures to follow. All users and employees as such would be required to register and have their information stored in the database in order to use the Attendance system. Upon launching the app, the first page you would encounter will enable you to register.

1.

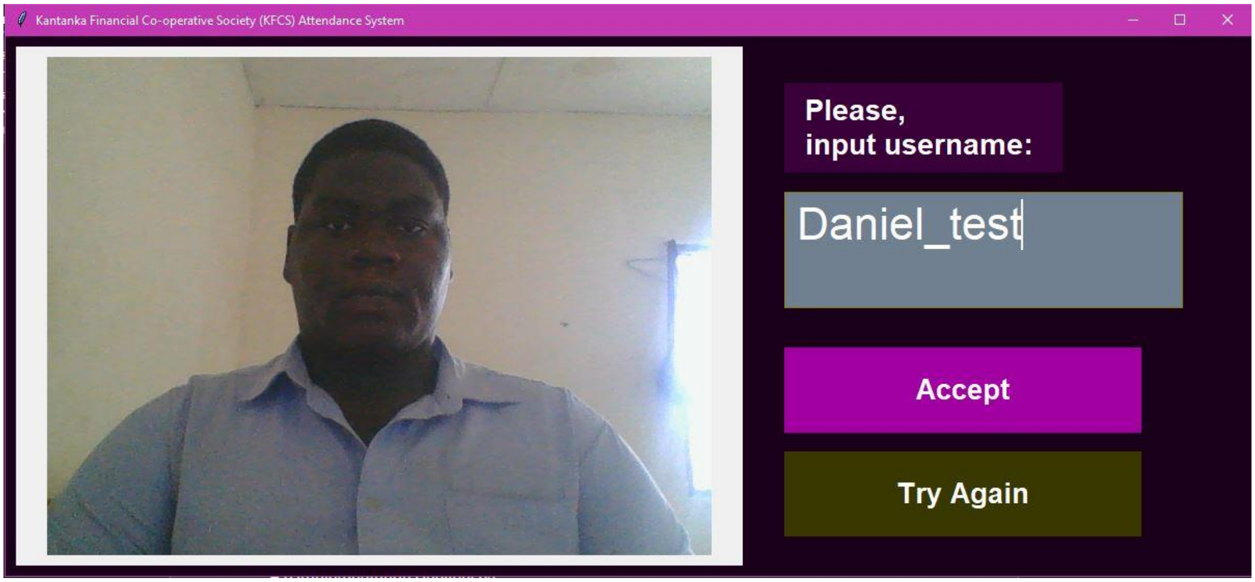


Click the “Register New User button to begin the registration process

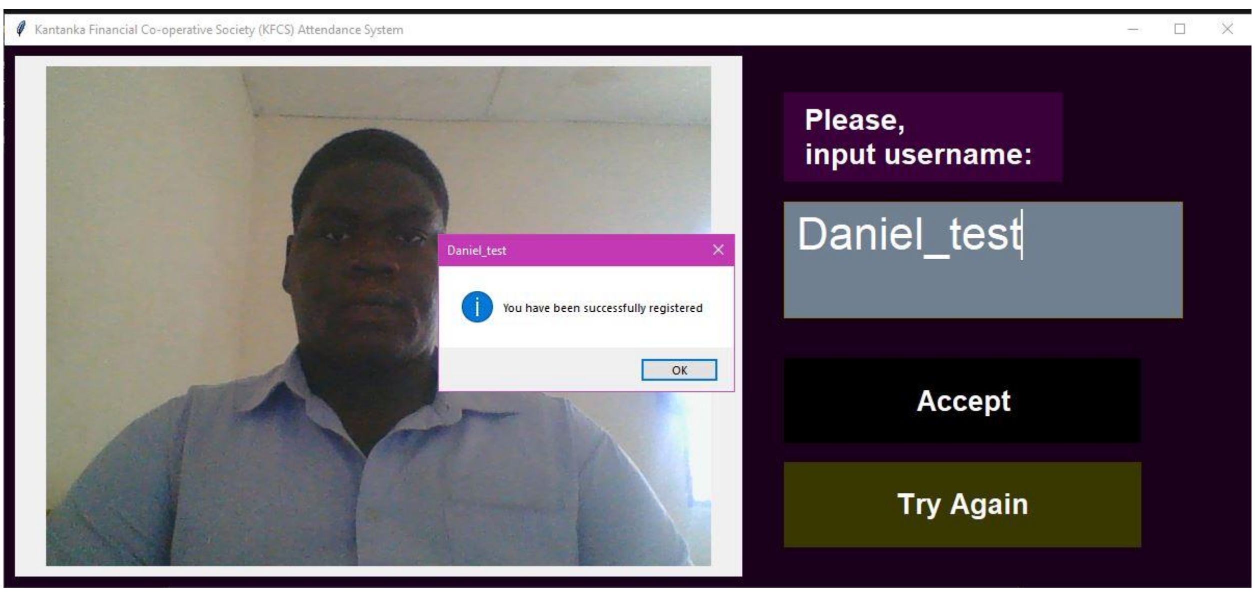
2. On the registration page, you would be required to input a username to store you in the database and enable you log in later.



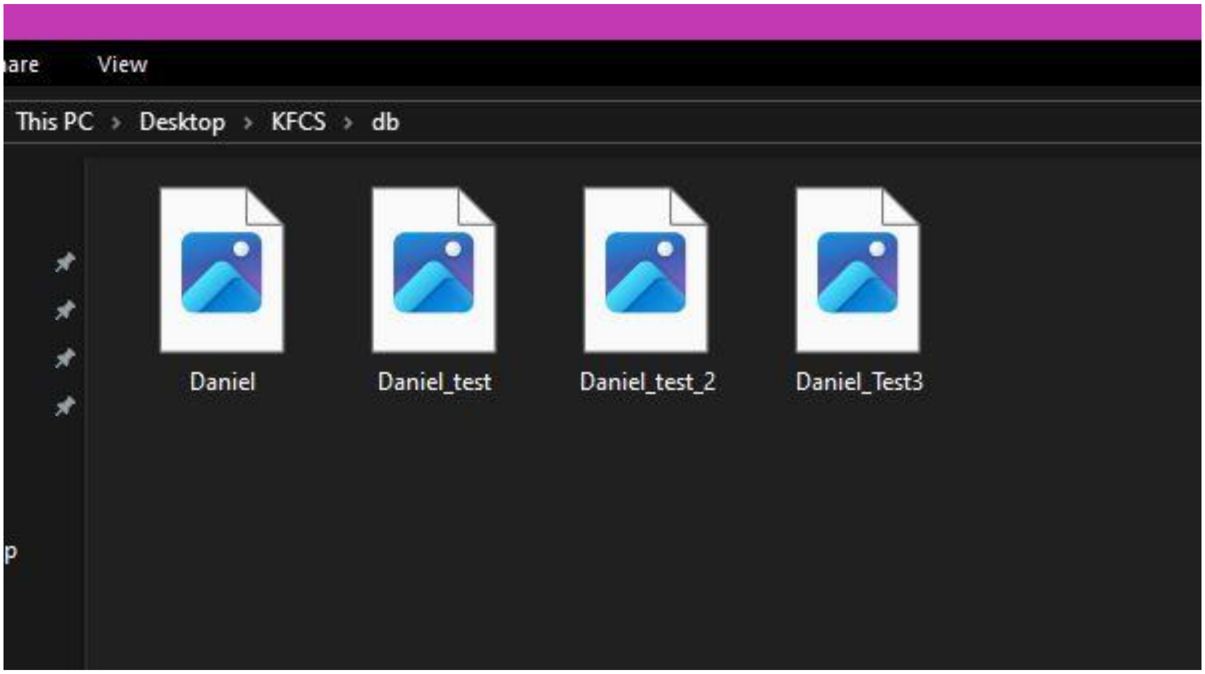
Be sure that the frame the system captured is clear and concise, if u feel it’s not, you can hit the “Try again” button.



3. Click accept to lock in the registration process

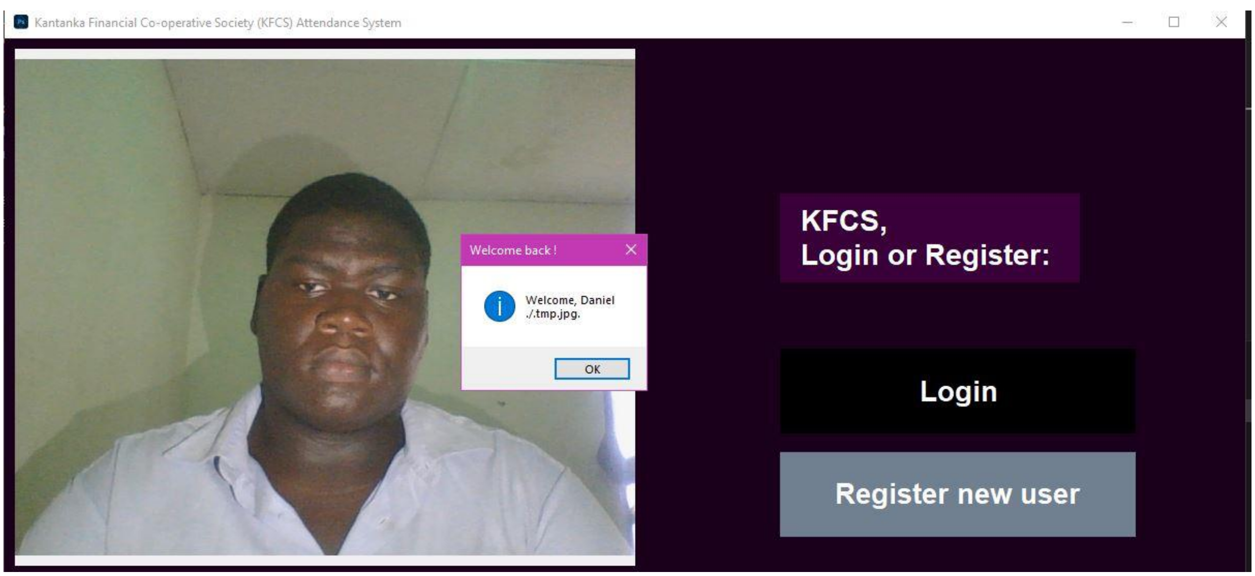


You have been successfully registered, and your details have been stored



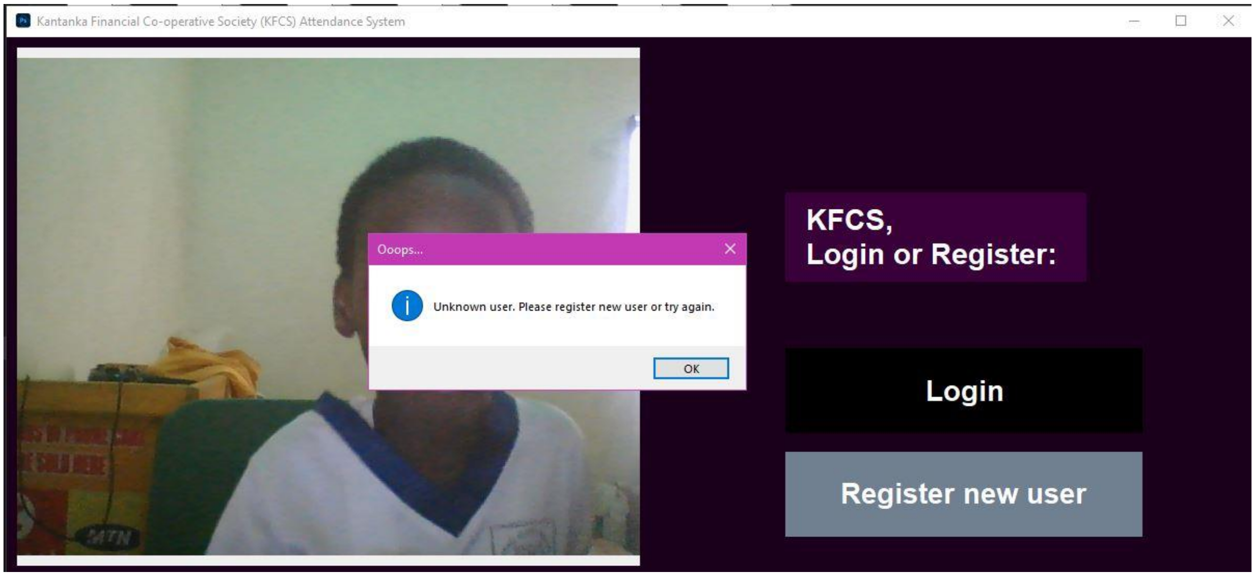
All the images collected.

4. Upon registration you would be taken back to the first page where you can attempt a login.



Successful log in attempt.

5. Unsuccessful login attempt.



6.Attendance Sheet generation post face verification

```
Daniel
./.tmp.jpg,2025-04-03 15:06:21.800870,in

                 ----------------------------------------
                 User: Daniel
./.tmp.jpg
                 Time: 2025-04-03 15:56:57
                 Action: Check-in
                 ----------------------------------------
```

a. The application was developed in a windows environment with the visual studio code editor.

b. Dependencies to look out for include cmake, dlib, opencv-python, Pillow, face_recogniton, tkinter. Key imports include os, datetime and subprocess.

c. Make sure to install the Visual studio C++ development bundle to avoid issues when installing cmake and dlib since dlib is built on C++.

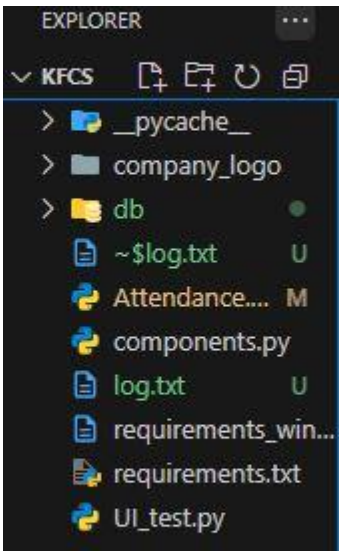
d. The system was developed by creating various objects through object-oriented programing.

e.Buttons, labels, messageboxes and the recognition function itself were initialized and then imported from our components script, for simplicity and readability.

f. Be sure to go through the file structure and our code appendixes below.

GitHub repository: https://github.com/Daniel-KK-world/KFC

g.File structure (test files included)



training process. I also highlighted the system's limitations, particularly its reliance on a limited dataset and its suitability for verification rather than large-scale recognition tasks.

Throughout the implementation phase, we encountered several challenges, such as selecting the appropriate technology stack, addressing data security and privacy concerns, ensuring scalability, and managing user adoption. Despite these hurdles, the system was successfully developed, tested, and prepared for deployment.