DSA: ICS 2105

MULTI-METHOD STUDENT CHECK-IN SYSTEM DOCUMENTATION

July 7, 2025

AUTHORS:GROUP4

1. **IRERI LINUS M. SCM211-0373/2024**
2. **EDWIN MWANGI SCM211-0332/2024**
3. **LOUIS MULAA SCM211-0375/2024**
4. **MOHAMED MUNGAI SCM211-0329/2024**
5. **BENEDICT MAKAU SCM211-0302/2024**

Contents

1 Introduction 3

2 Literature Review 4

2.1 Student ID Verification 4

2.2 Biometric Authentication 4

2.3 Face Recognition Techniques 4

3 System Assumptions 4

4 System Flow(Taking Face recognition as the chosen method) 5

4.1 For instance: Time Complexity Analysis of the system flow above (Taking the Face Recognition as the chosen method) 5

5 Flowcharts and Pseudocodes 6

5.1 System Overview 6

5.2 System Overview 7

5.3 Student ID flowchart 8

5.4 Student ID pseudocode 9

5.5 Student Biometric flowchart 10

5.6 Student Biometric pseudocode 11

5.7 Face Recognition flowchart 12

5.8 Face Recognition pseudocode 13

6 Time Complexity 14

6.1 System Overview Time Complexity 14

6.1.1 Worst-Case Complexity 15

6.1.2 Final Complexity 15

6.2 Student ID Time Complexity 16

6.3 Student Biometric Time Complexity 17

6.4 Student Face Recognition Time Complexity 19

7 Conclusion 20

Introduction

In modern institutions, advanced face recognition solutions empower secure, seamless, and efficient user authentication processes daily worldwide.
These cutting-edge methods utilize sophisticated algorithms for extracting unique facial features, ensuring robust verification accuracy consistently worldwide.
If matches fail repeatedly, the system transitions to default Student ID checks or alerts staff for assistance promptly.
The retry mechanism accommodates up to three attempts, ensuring adequate opportunities for successful verification without unnecessary disruption.
Administrators can monitor performance metrics continuously, detecting anomalies early and applying corrective measures to sustain reliability overall.
Modern hardware integration ensures real-time image processing, minimizing wait times while preserving accuracy under various environmental conditions.
All captured data is securely stored and encrypted, preventing unauthorized access and protecting sensitive personal information effectively.
Image preprocessing includes normalization, face detection, and feature extraction, facilitating robust comparisons despite lighting or angle variations.
After feature vectors are obtained, the system matches them against a registered database, confirming identities upon alignment seamlessly.
Each verification attempt updates internal counters, enabling the system to track retries and trigger fallback strategies appropriately.
If retries exceed three, the mechanism defaults to student ID verification or prompts staff intervention for resolution immediately.
Performance metrics, such as average processing speed and successful match rate, are periodically evaluated to ensure quality consistently.
System scalability is paramount, supporting larger databases without compromising accuracy or significantly increasing verification latency excessively today.
A modular architecture permits seamless upgrades, ensuring that future improvements in hardware or algorithms integrate smoothly anytime.
User experience remains central, emphasizing extremely straightforward interactions, minimal wait times, and highly reliable authentication outcomes consistently.
Data privacy measures include encryption and strict access controls, safeguarding personal information from unauthorized retrieval attempts persistently.
Periodic system audits evaluate security protocols, ensuring compliance with established standards and fostering confidence among stakeholders worldwide.
Fallback procedures minimize service disruptions, guaranteeing that users can complete check-ins promptly even under unexpected conditions repeatedly.
Alerts are automatically generated if failures persist, directing support personnel to address issues and maintain operational continuity.
This documentation explains core concepts, system assumptions, time complexity, and fallback strategies ensuring maximum effectiveness overall consistently.
Through diligent implementation, regular monitoring, and iterative refinement, face recognition check-in systems achieve enduring reliability worldwide successfully.

Literature Review

2.1 Student ID Verification

Student ID verification is the traditional method used in many institutions. It involves swiping or scanning a physical ID, which is then verified against a database. While cost-effective and widely implemented, Student ID systems can be susceptible to human error and are sometimes less secure against fraudulent use.

2.2 Biometric Authentication

Biometric authentication has been widely studied in both academic and industrial contexts. Common biometrics include fingerprint, iris, and facial recognition. According to various studies, face recognition is appealing due to its contactless nature and improvements in camera technology.

2.3 Face Recognition Techniques

Classical face recognition approaches involved techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Modern solutions often leverage deep learning with Convolutional Neural Networks (CNNs). These approaches achieve high accuracy but can be computationally intensive depending on the network architecture and database size.

System Assumptions

- A functional camera is available to capture the user's face.
- The system has a pre-enrolled database of facial features for verification.
- The environment has adequate lighting and minimal occlusion to ensure accurate face captures.
- Up to three retry attempts are allowed before the system falls back to ID verification or alerts staff.
- Network connectivity and backend servers are operational for data processing and storage.
- Each student has their own Identity card.

System Flow(Taking Face recognition as the chosen method)

1. **Start:** The user initiates the check-in process.
2. **Camera captures student's face:** A camera feed captures the user's face in real time.
3. **System processes:** Preliminary image enhancements or normalization.
4. **System extracts features:** Facial features are extracted using a dedicated algorithm (e.g., a CNN).
5. **System compares with stored data:** Extracted features are matched against a database of enrolled faces.
6. **Match Found?**
   - **Yes:** The user proceeds to check-in, and relevant details are updated (entry time, check-in time). Optionally, a notification can be sent.
   - **No:** The system retries capturing and processing up to three times. If all attempts fail, the system falls back to ID verification or alerts staff.
7. **End:** The process concludes either by successful check-in or escalation to an alternative check-in method or staff assistance.

4.1 For instance: Time Complexity Analysis of the system flow above (Taking the Face Recognition as the chosen method)

Let:

- p = time to process the captured image.
- k = time to extract features.
- m = time to compare features with stored data.
- **Capture face:** $O(1)$
- **Process data:** $O(p)$
- **Extract features:** $O(k)$
- **Compare with database:** $O(m)$
- **If match found:** proceed to check-in in $O(1)$
- **If no match:** retry up to 3 times $\rightarrow O(3(p+k+m)) = O(p+k+m)$
- **If all fail:** fallback to ID check or alert staff in $O(1)$

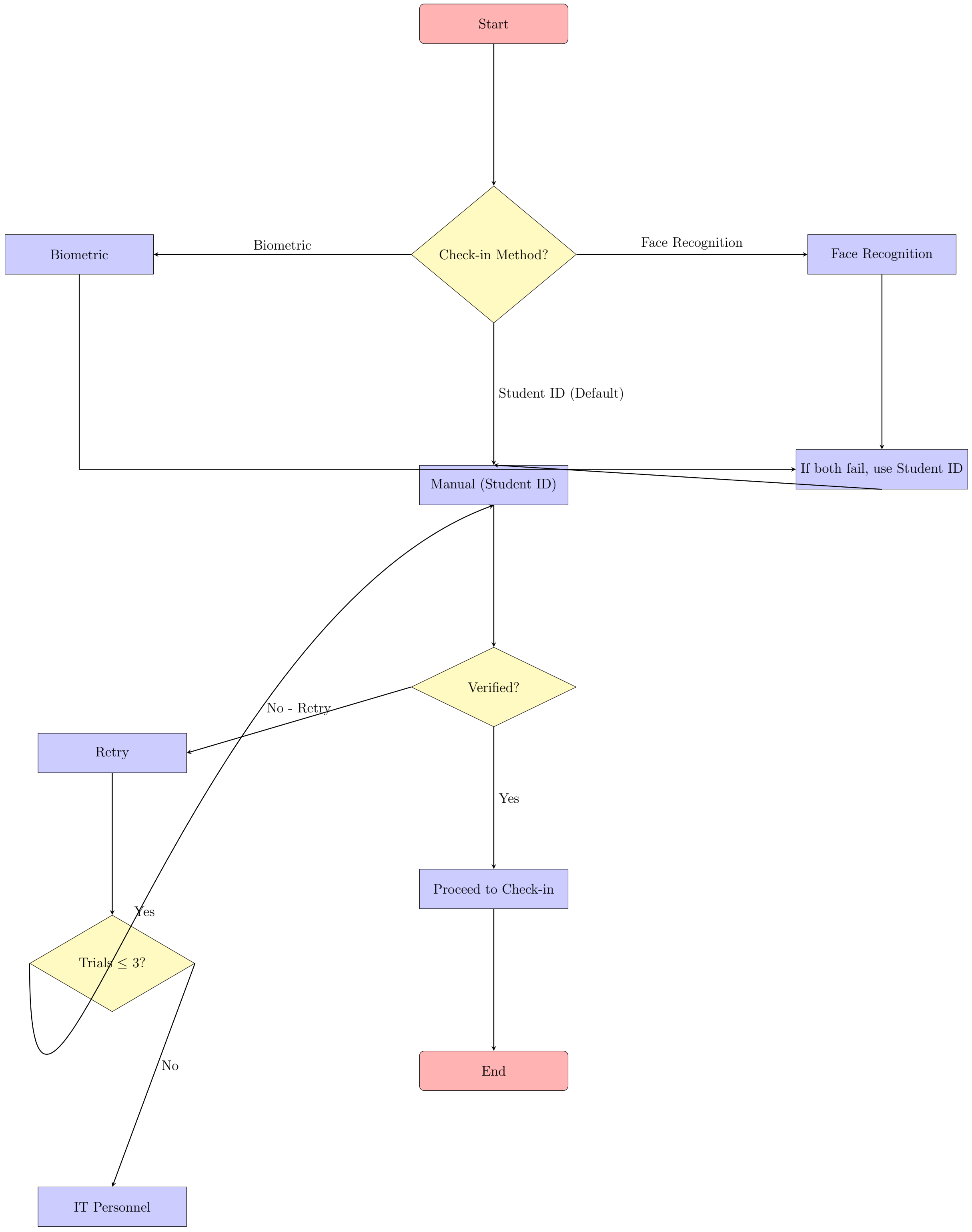
Worst-Case Complexity

$$O(p+k+m)$$

assuming 3 retries is a constant factor.

Flowcharts and Pseudocodes

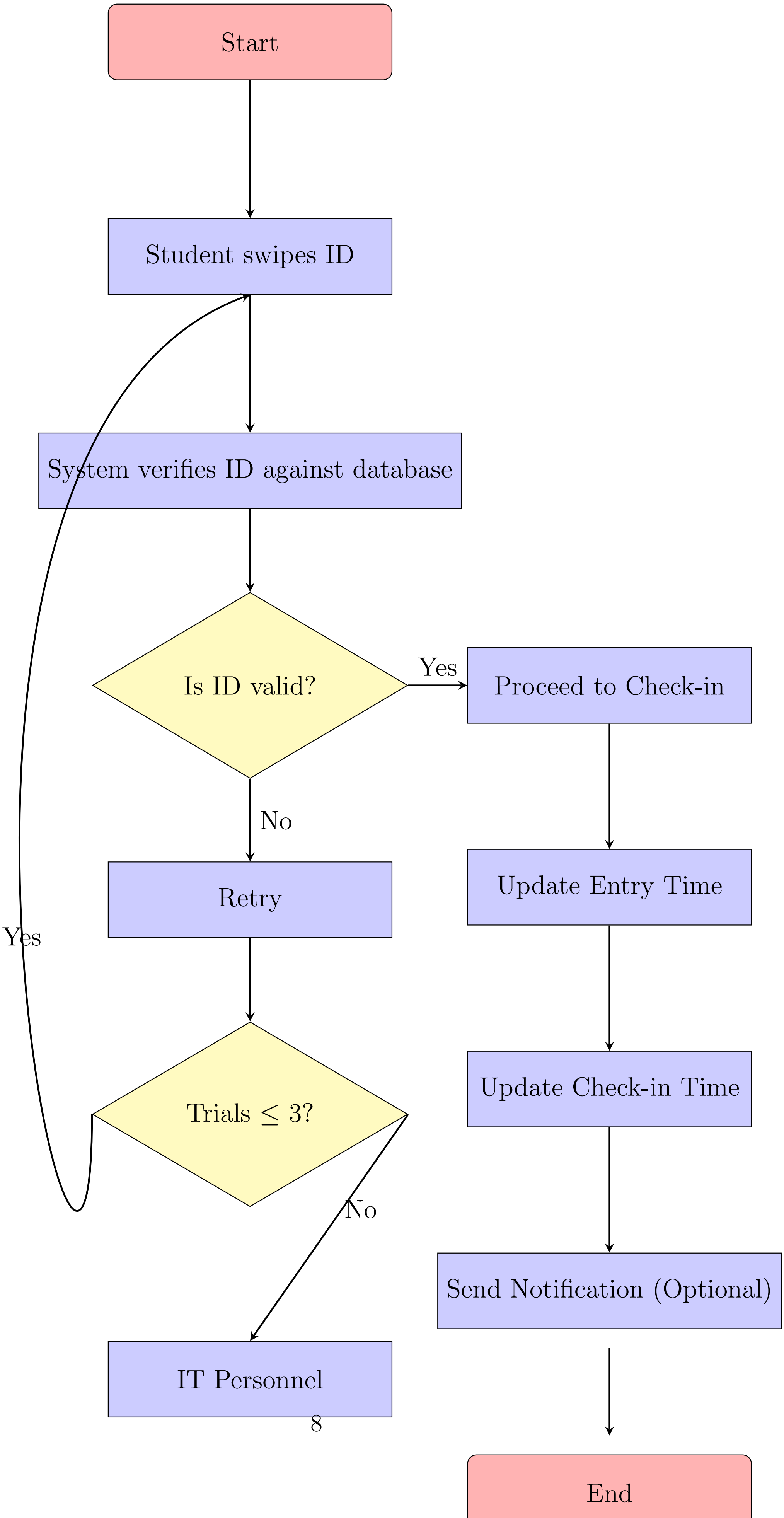
5.1 System Overview



5.2 System Overview

```
Start
Default Check-in Method: Student ID
Optional Methods: Biometric or Face Recognition
Choose Check-in Method
if Check-in Method is Biometric then
    Capture Biometric Data
    if Biometric Data is Verified then
        Proceed to Check-in
        End
    else
        Retry Biometric (Up to 3 times)
        if Retries $> 3$ then
            Fallback to Student ID
        end if
    end if
else if Check-in Method is Face Recognition then
    Capture Face Data
    if Face Data is Verified then
        Proceed to Check-in
        End
    else
        Retry Face Recognition (Up to 3 times)
        if Retries $> 3$ then
            Fallback to Student ID
        end if
    end if
end if
Proceed to Default Student ID Verification
Manual Student ID Check
if ID is Verified then
    Proceed to Check-in
    End
else
    Retry Verification
    Increase Retry Count
    if Retries $\leq 3$ then
        Repeat Verification Process
    else
        Direct to IT Personnel
        End
    end if
end if
```

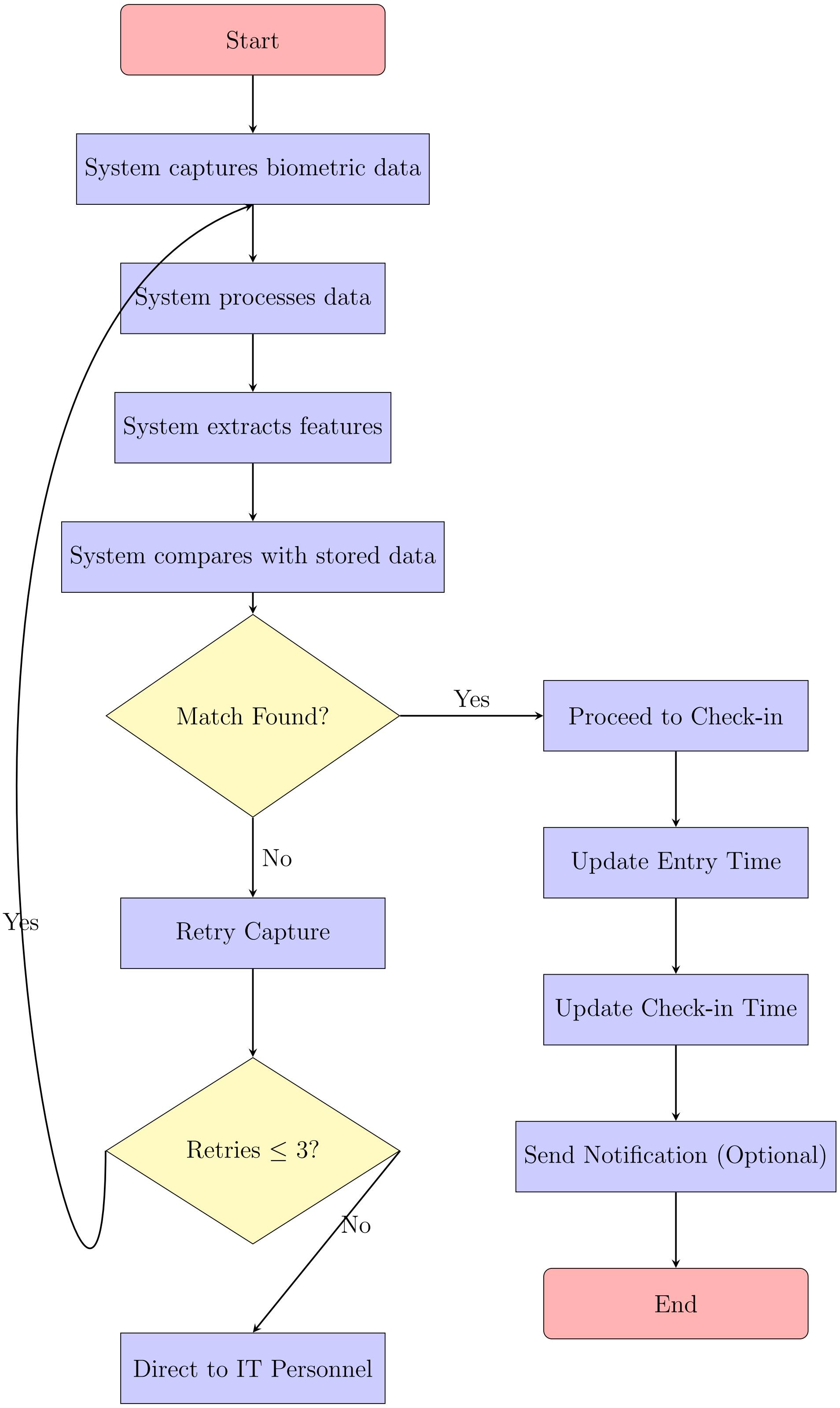
5.3 Student ID flowchart



5.4 Student ID pseudocode

```
Start
Student swipes ID
System verifies ID against database
if ID is valid then
    Proceed to Check-in
    Update Entry Time
    Update Check-in Time
    Send Notification (Optional)
    End
else
    Retry
    Increase trial count
    if Trials $\leq 3$ then
        Repeat verification
    else
        Direct to IT Personnel
        End
    end if
end if
```

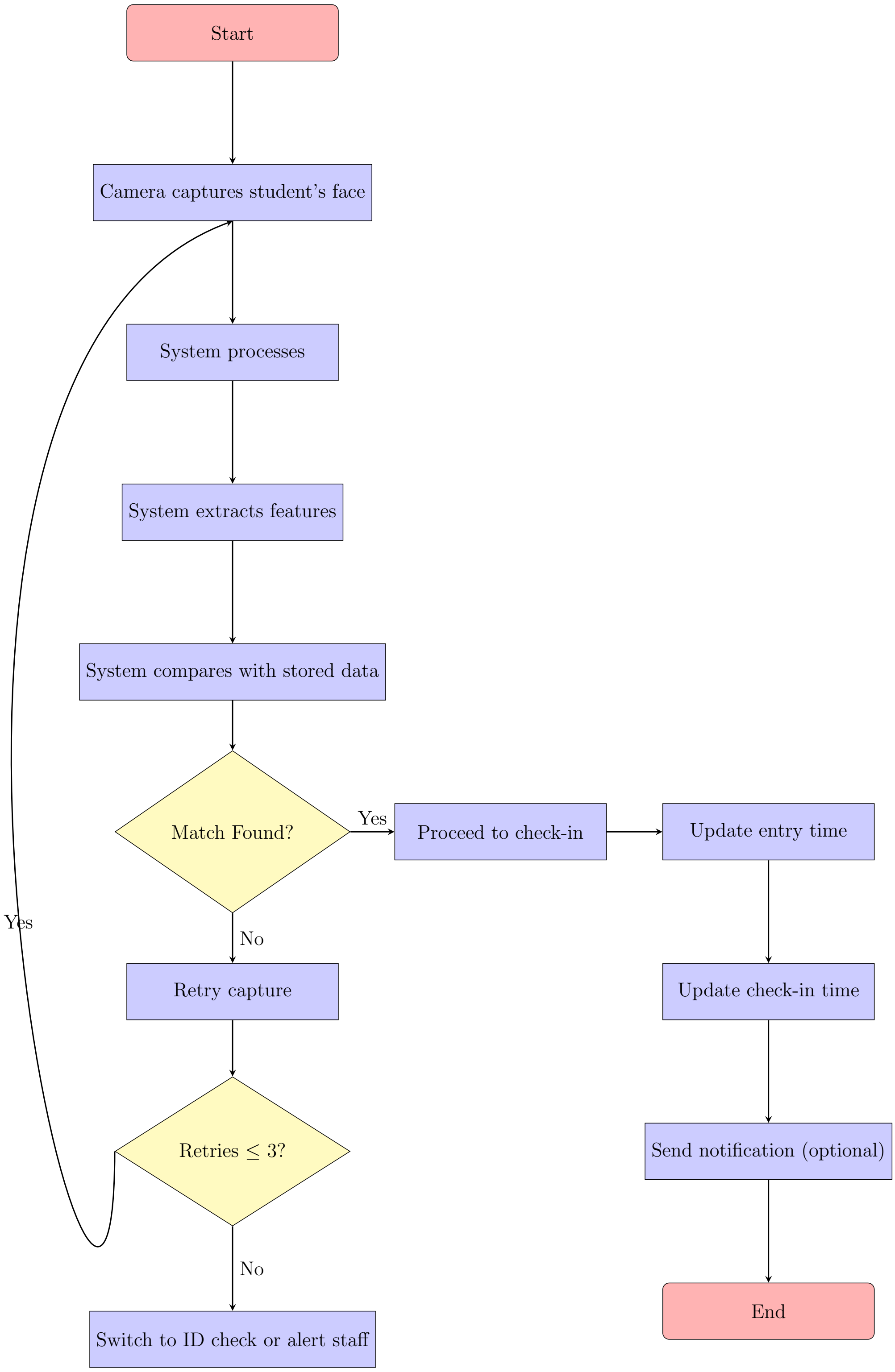
5.5 Student Biometric flowchart



5.6 Student Biometric pseudocode

```
Begin
Display "Start Biometric Check-in"
$attempts \leftarrow 0$
$maxAttempts \leftarrow 3$
while $attempts \leq maxAttempts$ do
    Capture biometric data
    Process biometric data
    Extract features
    Compare with stored data
    if Match Found then
        Display "Proceed to Check-in"
        Update entry time
        Update check-in time
        Send notification (optional)
        Display "Check-in Successful"
        Exit
    else
        Display "Retry Capture"
        $attempts \leftarrow attempts + 1$
    end if
end while
if $attempts > maxAttempts$ then
    Display "Direct to IT Personnel"
end if
Display "End Biometric Check-in"
```

5.7 Face Recognition flowchart



5.8 Face Recognition pseudocode

```
Start
Student swipes ID
System verifies ID against database
if ID is valid then
    Proceed to Check-in
    Update Entry Time
    Update Check-in Time
    Send Notification (Optional)
    End
else
    Retry
    Increase retry count
    if Retries $\leq 3$ then
        Repeat verification
    else
        Direct to IT Personnel
        End
    end if
end if
```

Time Complexity

6.1 System Overview Time Complexity

- **Selecting the check-in method:** $O(1)$ (constant time decision)
- **Student ID Verification (Default)**:
  - Scan Student ID: $O(1)$.
  - Verify against database: $O(p)$, where p is the number of stored student IDs.
  - If verified: Proceed to check-in $O(1)$.
  - If failed after 3 retries, escalate to IT personnel: $O(1)$.
- **Biometric Verification:**
  - Capture biometric data: $O(1)$.
  - Process biometric data: $O(k)$, where k is the number of features checked.
  - Extract features: $O(k)$.
  - Compare with stored records: $O(m)$, where m is the number of stored biometric templates.
  - If match found: Proceed to check-in $O(1)$.
  - If failed, retry up to 3 times: $O(3(k+m)) = O(k+m)$ (constant factor ignored).
  - If unsuccessful after 3 retries, fallback to **Face Recognition**.
- **Face Recognition (if Biometric Fails)**:
  - Capture face data: $O(1)$.
  - Extract face features: $O(f)$.
  - Compare with stored records: $O(n)$, where n is the number of face templates.
  - If match found: Proceed to check-in $O(1)$.
  - If failed, retry up to 3 times: $O(3(f + n)) = O(f + n)$.
  - If unsuccessful after 3 retries, fallback to **Student ID Verification**.
- **Final Fallback to Student ID (if all else fails)**:
  - Scan Student ID: $O(1)$.
  - Verify against database: $O(p)$.
  - If verified: Proceed to check-in $O(1)$.
  - If verification fails, alert IT personnel: $O(1)$.
- **Successful Check-in Steps:**
  - Update entry time: $O(1)$.
  - Update check-in time: $O(1)$.
  - Send notification (optional): $O(1)$.

6.1.1 Worst-Case Complexity

The worst case occurs when both **Biometrics** and **Face Recognition** fail, forcing the system to fall back to **Student ID Verification**, which also fails, leading to escalation to IT personnel.

$$O(k+m)+O(f+n)+O(p)=O(k+m+f+n+p)$$

Since retries are limited to 3 times, they do not affect the asymptotic complexity.

6.1.2 Final Complexity

- If database lookups (m, n, p) are fast $(O(1))$, the process runs in **$O(k+f)$**.
- If database lookups are large $(O(m+n+p))$, the worst-case complexity is **$O(k+m+f+n+p)$**.

Thus, in practical scenarios, **biometric and face recognition processing times** (k, f) dominate, making the worst-case complexity:

$$O(k+f)$$

If all methods fail, IT personnel intervention occurs in $O(1)$.

6.2 Student ID Time Complexity

This algorithm verifies a student's ID against a database, allows up to three retry attempts, and escalates to IT personnel if all attempts fail.

Time Complexity Analysis

- **Swipe ID:** $O(1)$.
- **Verify ID against database:** $O(p)$, where p is the number of stored IDs.
- **If ID is valid:**
  - Proceed to check-in, update entry/check-in time, and send notification, each in $O(1)$.
- **If ID is invalid:**
  - Retry up to 3 times, resulting in at most $3 \times O(p) = O(p)$.
  - If still invalid after 3 retries, direct to IT personnel: $O(1)$.

Worst-Case Scenario

In the worst case, the system checks the ID up to 3 times (all invalid) and then escalates to IT. Since 3 is a constant, the dominant factor is the database lookup $O(p)$:

$$O(3p) = O(p).$$

Conclusion

The overall time complexity of this Student ID check-in system is:

$$\boxed{O(p)}$$

where p is the size of the student ID database.

6.3 Student Biometric Time Complexity

The given biometric check-in process consists of multiple authentication methods: **Biometrics**, **Face Recognition**, and **Student ID (Default)**. If biometrics and face recognition fail, the system falls back to **Student ID** verification. If all methods fail, an alert is sent to IT staff.

Stepwise Complexity Breakdown

- **Biometric Verification:**
  - Capture biometric data: $O(1)$
  - Process biometric data: $O(k)$ (depends on data complexity)
  - Extract features: $O(k)$
  - Compare with stored records: $O(m)$, where m is the number of stored biometric templates
  - If match found: Proceed to check-in $O(1)$
  - If no match: Retry up to 3 times $O(3(k + m))$, then fall back to **Face Recognition**
- **Face Recognition (if Biometrics Fails):**
  - Capture face data: $O(1)$
  - Extract face features: $O(f)$
  - Compare with stored records: $O(n)$, where n is the number of face templates
  - If match found: Proceed to check-in $O(1)$
  - If no match: Retry up to 3 times $O(3(f + n))$, then fall back to **Student ID**
- **Student ID Verification (Final Fallback):**
  - Scan ID: $O(1)$
  - Verify against database: $O(p)$, where p is the number of stored student IDs
  - If verified: Proceed to check-in $O(1)$
  - If failed: Alert IT Staff $O(1)$

Worst-Case Complexity

If both **biometrics** and **face recognition** fail, the process falls back to **Student ID verification**. The worst case occurs when all authentication methods fail, leading to:

$$O(3(k+m)) + O(3(f+n)) + O(p) = O(k+m+f+n+p)$$

Since **3 is a constant**, it does not affect the asymptotic complexity.

Final Complexity

- If database lookups (m, n, p) are fast $(O(1))$, then the overall complexity is **$O(k+f)$**.
- If database lookups are large $(O(m+n+p))$, then the overall complexity is **$O(k+m+f+n+p)$**.

If all methods fail, an **IT alert is triggered** in **$O(1)$** time.

6.4 Student Face Recognition Time Complexity

This flowchart captures a student's face, processes and extracts features, compares them to stored data, and either proceeds to check-in or retries up to 3 times. If all attempts fail, the system switches to an alternate check (ID) or alerts staff.

Step-by-Step Complexity

- p = time to process the captured image
- k = time to extract features
- m = time to compare extracted features with stored data

1. **Camera captures student's face**: $O(1)$
2. **System processes**: $O(p)$
3. **System extracts features**: $O(k)$
4. **System compares with stored data**: $O(m)$
5. **Match Found?**
   - **Yes**: Proceed to check-in, update times, send notification (all $O(1)$).
   - **No**: Retry capture, up to 3 times.
6. **Retry Capture**:
   - Each retry includes processing, extracting features, and comparing again: $O(p + k + m)$
   - Maximum 3 retries: $3 \times (p + k + m) = O(p + k + m)$ (constant factors are ignored in Big-O).
   - If all retries fail, switch to ID check or alert staff: $O(1)$

Worst-Case Complexity

In the worst case, the system goes through 3 failed attempts before switching to an alternate method or alerting staff:

$$O(p + k + m).$$

If p, k, and m are constants (or bounded by a small database), the process is effectively $O(1)$. Otherwise, $p + k + m$ dominates.

Conclusion

Face recognition check-in systems offer a contactless and convenient user experience. By allowing up to three retries, the system provides flexibility while maintaining security. In the event of repeated failures, defaulting to an alternative verification method (e.g., student ID) or alerting staff ensures minimal disruption. The worst-case time complexity is dominated by the facial processing and comparison steps, making it $O(p+k+m)$. With continued advances in camera hardware and facial recognition algorithms, such systems can be further optimized for speed and accuracy.