LOST STUDENT ID RECOVERY SYSTEM

Introduction to Computing Project

Presented to the

Faculty of the College of Information Technology

Cagayan de Oro College

Cagayan de Oro City

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Information Technology

NAME OF RESEARCHERS

AGBU, FRANKK LOYD S.

CAPINIG, MARLOU M.

DAGOOC, MANUEL B. III

NACALABAN, JUDYNA A.

UNGON, KEICELINE A.

NOVEMBER, 2025

1.1 Introduction

In schools and universities nowadays, one of the common struggles for students and administrators is the disappearance of student identification cards. IDs are essential because they are used to access school facilities, borrow books from the library, and serve as proof of student identity. However, every year, many students lose or misplace their IDs, which causes inconvenience and sometimes leads to security problems on campus. According to a study by the National Association of Student Personnel Administrators (NASPA), between 10 % and 20% of college students report losing at least one item, including IDs, which leads to interruptions in day-to-day campus life. The replacement process also takes time and effort, and institutions often spend around ₱200 to ₱300 per card for reprinting and administration, which puts additional financial strain on both students and schools. In addition to financial expenses, misplaced IDs lead to security threats, including identity theft or illegal entry to restricted areas , especially on crowded campuses where IDs act as resource gatekeepers.

Furthermore, the issue is that schools continue to process lost and found cases by hand, which leads to unclaimed IDs and inefficient tracking. In response to these recurring issues, the Lost Student ID Recovery System was developed, a console-based Python program that securely records, updates, and tracks lost ID information. Featuring an admin login and CRUD functions, it allows for organized usage of in-memory data. The goal of this system is to track information, reduce the administrative workload, and increase the recovery rate of lost student identification cards.

1.2 Statement of the Problem

General Problem

Without an effective centralized way to handle lost and found student identification cards, tracking is poor, data handling inefficient, and the window for timely recovery is frequently missed, which leads to administrative difficulty and inconvenience for students and staff.

Specific Problems

- **Limited Tracking and Visibility**

Without a system, discovered IDs might not be properly logged or searchable, it makes difficult to notify owners or verify claims. Confusion and delays in retrieving misplaced IDs to students are caused by this lack of visibility.

- **Data Handling Inefficient**

Paper logs and other manual recording procedures are vulnerable to human error and loss. This causes disarray and inefficiency by making it difficult to keep up-to-date, accurate records of lost or recovered IDs.

- **Missed Recovery Opportunities**

Administrators are unable to promptly locate and return misplaced IDs in the absence of an appropriate search and monitoring system. Recovery is slowed down as a result, and many IDs go unclaimed.

1.3 General Objective

This study's goal is to create a Python console-based Lost Student ID Recovery System with in-memory storage. By enabling an administrator to effectively add, view, search, edit, and delete information, the system seeks to enhance the administration, tracking, and recovery of misplaced student IDs.

1.4 Specific Objectives

- **To enhance tracking and visibility**

Develop a centralized digital system that makes it simple for administrators to enter, look up, and amend found ID data. This guarantees openness, prompt owner identification, and expedited recovery of misplaced identification documents.

- **To improve data handling efficiency**

Implement a structured database using CRUD functions to store and manage ID records accurately. This minimizes human error, prevents data loss, and ensures consistent and reliable documentation.

- **To increase recovery opportunities**

Establish an automated or well-monitored process that alerts administrators or students when an ID is found. This feature improves communication, speeds up recovery time, and reduces the number of unclaimed IDs.

Scope and Limitation

1.5 Scope

The system focuses on basic CRUD operations via a text-based console interface with admin login:

- Add new found ID records.
- View all records with full details.
- Search records by student name.
- Update claim status.
- Delete records with confirmation.
- Require admin login (hardcoded username "admin" and password "12345") to access the system.
- Data is stored in memory (using a list of dictionaries) during the session.

1.6 Limitation

- Console-only interface; no graphical user interface (GUI).
- Basic user authentication via hardcoded admin login; no advanced security or multi-user support.
- Data is stored in memory and not persistent across sessions (records are lost when the program exits).
- No synchronization with external systems or databases.
- Limited input validation; assumes correct data entry.
- No integration with email or notifications for automated alerts.

1.7 System Requirements

Hardware Requirements:

- Processor: Intel Core i3 or above.
- Memory (RAM): At least 2 GB.
- Storage: Minimum 100 MB of free disk space.
- Monitor: Any display supporting text output.
- Keyboard: For input commands.

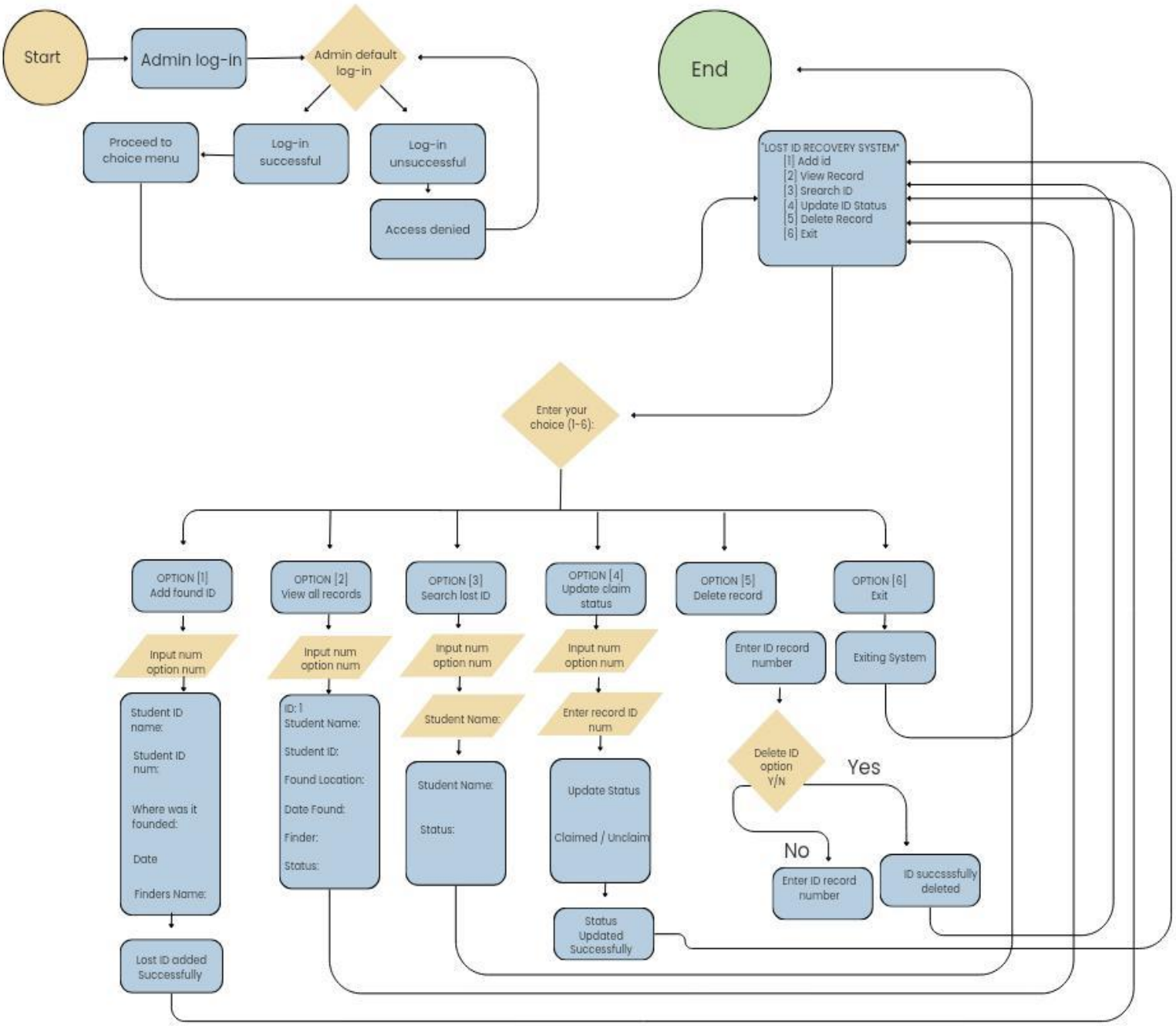
Software Requirements

- Operating System: Windows, macOS, or Linux.
- Programming Language: Python 3.x.
- Code Editor: Any text editor (e.g., Visual Studio Code).

Functional Requirements:

- Admin Login: Authenticate users with a hardcoded username and password before granting access.
- Add Record: Input and store details of a found ID in memory.
- View Records: Display all records in a formatted list.
- Search Records: Query by student name with partial matches.
- Update Status: Modify claim status for a specific record.
- Delete Record: Remove a record after confirmation.
- Exit System: Terminate the program.

1.8 Flowchart



The Lost Student ID System's procedure is shown in the flowchart. The first step is to connect or create a database and confirm that the required table is present. The main menu with six options then appears: add found ID, view all records, search lost ID, update claim status, delete record, and exit. When the user chooses an option, the application carries out the selected action, which could be deleting a record, reading or searching records, adding ID details, or modifying the status of a claim. If the user chooses to exit, the program ends. All the things considered, the flowchart illustrates how the system effectively handles lost ID information through a straightforward, sequential procedure.

1.9