Personal Diary Management System

Md Abul Bashar Nirob - 2022198042
Jannatul Islam Eshita - 2524186042
Tashin Binte Taiba - 2522283642
Sara Tasnim - 2522727042
Tasnia Afrin - 2523855642
ECE Department Major in CSE, Course Code: CSE-115, Section: 4, Group: 7
July 05, 2025

Abstract—This project presents a Personal Diary Management System developed in C programming language. The system provides a secure digital platform for users to record personal thoughts and memories. Key features include creating, viewing, editing, and deleting diary entries, along with advanced functionalities like password-protected encryption, search capabilities, and automated backups. The program uses file handling for data storage and offers a user-friendly, menu-driven interface with a colorful console design. This report documents the system's design, implementation, results, and potential future enhancements.

I. Introduction

A. Background

Diaries have long been used as personal journals to record thoughts, experiences, and memories. Traditional paper diaries, however, face risks such as loss, damage, or unauthorized access. Digital diary systems address these issues by providing secure, easily accessible, and organized platforms for personal record-keeping.

B. Problem Statement

Existing digital diary solutions often lack robust security features or are overly complex for casual users. Our project aims to develop a simple yet secure Personal Diary Management System that balances functionality with ease of use.

C. Objectives

- Develop a menu-driven interface for easy navigation.
- Implement CRUD (Create, Read, Update, Delete) operations for diary entries.
- Ensure data security through file encryption.
- Provide search and backup functionalities.
- Enhance user experience with a colorful console interface.

D. Scope

The system is designed for individual users who need a private and organized way to maintain their diaries. It is implemented in C and runs on Windows-based systems.

II. Literature Survey

Several existing digital diary applications were reviewed during the project's planning phase:

1) Standard Text Editors (Notepad, WordPad):
   - Lack specialized features for diary management.
   - No encryption or search functionality.
2) Third-Party Diary Apps (Day One, Journey):
   - Offer cloud sync and multimedia support but require subscriptions.
   - May raise privacy concerns due to third-party data storage.
3) Open-Source Solutions:
   - Often complex for non-technical users.
   - Limited customization options.

Our system fills the gap by providing a lightweight, secure, and user-friendly solution with essential diary management features.

III. Methodology

A. System Architecture

The system follows a modular design, with each functionality implemented as a separate module:

1) User Interface Module:
   - Displays a color-coded menu for options like adding, viewing, or editing entries.
2) File Handling Module:
   - Stores entries in diary.txt using fopen(), fprintf(), and fclose().
3) Encryption Module:
   - Uses XOR-based encryption with a user-defined password.
   - Encrypted files are saved as diary$_{e}ncrypted.txt$.
4) Search & Edit Module:
   - Allows searching by date and modifies entries via temporary files.
5) Backup Module:
   - Creates timestamped backups (e.g., diary$_{b}ackup_{2}0240705.txt$).

B. Implementation Details

Key Functions:

- addEntry(): Appends new entries with dates.
- viewEntries(): Displays all saved entries.
- searchEntry(): Finds entries by date.
- editEntry() and deleteEntry(): Modifies or removes entries safely.
- encryptFile() and decryptFile(): Secures data with password protection.
- backupDiary(): Generates automated backups.

Technologies Used:

- Programming Language: C
- Tools: GCC Compiler, Windows Console API (for colors)

IV. Results & Discussion

A. Functional Testing

- Adding Entries: Successfully saves entries with dates.
- Viewing Entries: Displays all entries in a structured format.
- Search Function: Accurately retrieves entries by date.
- Encryption/Decryption: Works as intended with password protection.
- Backup System: Creates restorable backups without data loss.

B. Limitations

- No Database: Relies on text files, limiting scalability.
- Basic Encryption: XOR encryption is not highly secure against advanced attacks.
- Platform Dependency: Designed primarily for Windows.

C. User Feedback

Testers found the system easy to use but suggested:

- Adding a reminder feature.
- Improving multi-user support.

V. Conclusion

This project developed a secure and user-friendly Diary Management System in C. It successfully implemented core features like adding, viewing, editing, and encrypting diary entries. The system provides an organized way to maintain personal records with password protection and backup functionality. The colorful menu interface enhances user experience. While using basic file storage and encryption, the program effectively serves as a practical digital diary solution. The project demonstrates successful application of file handling and modular programming concepts in C.

References

[1] B. Kernighan and D. Ritchie, The C Programming Language, 2nd ed.

[2] IEEE Editorial Style Manual for Authors.

[3] C Programming tutorials on file handling, available at https://www.w3schools.com/c/c$_f$*iles.php*.