Report on the Development of the Department Diary System

1. Introduction

The 'Department Diary' is a department activity monitoring system designed to keep track of all activities conducted during a semester. The system enables users to add, view, edit, and delete records of activities, providing a structured way to maintain a record of department events. File handling is effectively utilized to ensure data persistence, enabling users to save and retrieve records as needed. This application was developed using the C programming language due to its efficiency, portability, and suitability for file-handling operations.

2. Objectives

The primary objectives of developing the Department Diary system are:
- To provide a centralized platform for storing and managing department activities.
- To enable efficient addition, viewing, editing, and deletion of records.
- To utilize file handling for data persistence.
- To facilitate follow-up of departmental activities through organized records.

3. System Features

The application supports the following features:
1. Add Activity: Users can add records to the diary, including details like Activity ID, Task Name, Address, Date, Time, and Duration (in minutes).
2. View Activities: Users can view all the activities stored in the diary.
3. Edit Activity: Users can modify the details of a specific activity by specifying its ID.
4. Delete Activity: Users can delete an activity from the diary based on its ID.
5. Data Persistence: All records are saved in a file (department_diary.txt) to ensure data is preserved across application sessions.

4. Design and Implementation

a. Data Structure

The system uses a struct to store activity details:

```c
typedef struct {
   int id;
   char name[100];
   char address[100];
   char date[20];
   char time[10];
   int duration; // Duration in minutes
} Activity;
```

b. File Handling

The file department_diary.txt is used to store the records. Data is saved in a structured format using delimiters (|) to separate fields for easy parsing.

c. Menu-Driven Interface

A menu-driven approach is used to allow users to interact with the application. Users select options for adding, viewing, editing, or deleting records, or for exiting the program.

d. Key Functions

The application includes the following functions: addActivity(), viewActivities(), editActivity(), deleteActivity(), saveToFile(), and loadFromFile().

5. Sample Outputs

Adding an Activity:

```
Enter Activity ID: 1
Enter Task Name: Seminar on AI
Enter Address: Room 101
Enter Date (YYYY-MM-DD): 2024-11-15
Enter Time (HH:MM): 10:00
Enter Duration (in minutes): 120
Activity added successfully.
```

Viewing Activities:

```
ID: 1
Task Name: Seminar on AI
Address: Room 101
Date: 2024-11-15
Time: 10:00
Duration: 120 minutes
```

Editing an Activity:

```
Enter Activity ID to edit: 1
Enter New Task Name: Workshop on AI
Enter New Address: Room 102
Enter New Date (YYYY-MM-DD): 2024-11-16
Enter New Time (HH:MM): 11:00
Enter New Duration (in minutes): 150
Activity updated successfully.
```

Deleting an Activity:

```
Enter Activity ID to delete: 1
Activity deleted successfully.
```

```

## 6. Challenges Encountered

During development, the following challenges were faced:
- Handling user input, particularly string manipulation and validation, required careful implementation.
- Ensuring data consistency during editing and deletion operations, especially in file handling.
- Designing a simple yet effective file format for storing and retrieving activity details.

## 7. Future Improvements

The application can be enhanced by:
- Adding a search feature to locate specific records based on keywords.
- Implementing password protection for secure access.
- Supporting additional data formats such as JSON or binary for efficient storage.
- Introducing a graphical user interface (GUI) for a better user experience.

## 8. Conclusion

The Department Diary system successfully meets its objectives, providing a functional and efficient way to monitor departmental activities. By leveraging file handling in C, the system ensures data persistence and reliability. With further enhancements, the application can be extended to support more features and offer greater usability.

## 9. Appendices

File Format:
```
ID|Task Name|Address|Date|Time|Duration
1|Seminar on AI|Room 101|2024-11-15|10:00|120
```
```