Programming Project / C++ - Meeting Management System

Student 1: Stanzel Robert Eduard

Student 1 is responsible for managing the meetings:

- Adding a new meeting (name, date, time, duration)

- Viewing scheduled meetings

- Deleting a meeting by name

- Storing meetings in a linked list and persisting to file

- Managing participant RSVPs and attendance status

Student 2: Vacant

Student 2 position is currently unfilled. When assigned, responsibilities may include:

- Enhancing the user interface

- Managing advanced reporting or reminder features

- Supporting multi-user or networked usage

II. Data Structures Used by the Team

The following structures are used to represent meeting and participant data:

```
typedef struct Meeting {

   char name[100];

   char date[11];

   char time[6];

   int duration;

   Participant* participants;

   struct Meeting* next;

} Meeting;
```

```
typedef struct Participant {

    char name[100];

    char email[100];

    RSVPStatus status; // 0: No Response, 1: Accepted, 2: Declined

    struct Participant* next;

} Participant;
```

III. File Structure

meetings.txt

Stores all meeting and participant data in the following format:

```
MEETING|<name>|<date>|<time>|<duration>

PARTICIPANT|<name>|<email>|<status>

... (repeated for each participant and meeting)
```

IV. Interacting with Executables

This application is menu-driven through the terminal. Options include:

1. Add Meeting - Create a new meeting

2. View Meetings - Display all scheduled meetings and participants

3. Delete Meeting - Remove a meeting by name

4. Add Participant - Add a participant to a specific meeting

5. Update RSVP - Change RSVP status for a participant

6. Exit - Save and quit the program