Case Study: MyCalendar – Training Mgmt System

Group 2

Team member Emp ID	**Team member Name**	**Team Member Email Id**
46289282	**Shrishti Singh**	**shrishti.singh@capgemini.com**
46290767	Devansh	DEVANSH.A.DEVANSH@CAPGEMINI.COM
46289312	Sejal Shrivastava	sejal.shrivastava@capgemini.com
46283140	Preeti Jha	preeti.a.jha@capgemini.com
46292699	Ankit Kumar	ankit.as.kumar@capgemini.com
46283097	Preetilisa Ray	preetilisa.ray@capgemini.com

PROBLEM STATEMENT

1.2 OBJECTIVE

To create meeting scheduling application MyCalendar ike Outlook calendar. The application is to be develope as Executable file compiled on Linux. There is one entity Employee who acts like meeting organizer or participant.

1.3 ABSTRACT OF THE PROJECT

1. Employee should be able to login through application.
2. Employee should be able to search other employee by name or email.
3. Employee should be able to schedule a meeting with other employees.
4. Other employees should receive notification for the meeting.
5. Employees should be able to perform action like Attend, Reject, Maybe.
6. The response should be send to the organizer employee.
7. Displays the scheduled meeting on that date.
8. The organizer should be able to cancel the meeting.
9. Set reminder for the employees to attend meeting.
10. Handle data and errors properly. Show appropriate messages to user.
11. Display good input, output messages and reports in proper format.
12. Security features should be implemented whereever possible. For example user passwords can be stored in encrytped format.

1.4 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

1. When MyCalendar starts it displays Following Screen -

```
-------------------------Login Screen--------------------------
Enter Employee ID     <EmpID>
Enter password        <password>
```

Information about employees is available in “employees.txt” file where one line correspond to one employee. It is comma separated file containing following information about employee -

Employee ID	number(4)
Employee Password	string(10)
Employee Name	string(50)
Department	string(10)
Mobile Number	number(10)
email	string(30)
Skype ID	string(20)

Create “employees.txt” in notepad having information about at least 20 employees from different departments.

When Employee ID and password is entered it is checked against entry in “employees.txt”.

- If match is found “Calender Menu” is displayed.
- If match is not found then message “Invalid Employee ID or password” is displayed and system exits.

2. Calender Menu

```
-------------------------Calender Menu--------------------------
    1. Schedule Meeting
    2. Send Notifications
    3. Send Reply
    4. Display Calender
    5. Display Meeting
    6. Cancel Meeting
    0. Quit

    Enter your option : <option>
```

option = 1 (Schedule Meeting)

Information about scheduled meetings is stored in “meetings.txt” file where one line correspond to one meeting. It is comma separated file containing following information about meeting -

Meeting ID	number(4) auto increment
Organizer Employee ID	number(4) = Employee ID of login Employee
Meeting Description	string(50)
Meeting Date	date (DD/MM/YYYY format) = today’s date
Meeting Duration	number(1) Number of hours meeting will be held
Meeting Status	char(1) = ‘S’ (for Scheduled)
Meeting Location	number(4) = selected from entries in “locations.txt” file.
Number of participants	number(3)

Information about meeting participants is stored in “participants.txt” file where one line correspond to one meeting participant. If number of participants = 3, three lines will be written in “participants.txt” file. It is comma separated file containing following information about participant -

Meeting ID	number(4)
Employee ID	number(4) = participant’s Employee ID (selected from “employees.txt” file)
Notification sent	char(1) = ‘N’
Response received	char(1) = ‘N’
Response Type	string(6) = ‘None’ initially, values = Attend, Reject, Maybe

In this option all above information regarding meeting will asked and stored in corresponding files.

option = 2 (Send Notifications)

Once meeting is scheduled orgnized can send notifications to participants about meeting.

List of meetings organized by login employee for which notifications are not sent will be displayed from “meetings.txt”.

Enter Meeting ID to send notifications <MeetingID>

These are is stored in “notifications.txt” file where one line correspond to one notification. It is comma separated file containing following information about notification -

Meeting ID	<MeetingID>
Notification ID	number(4) auto increment
Sender Employee ID	number(4) = organizer Employee ID
Receiver Employee ID	number(4) = participant Employee ID
Notification Text	string(50) = This will be formed by combining meeting ID, Date, Location

In “participants.txt” field “Notification sent” will be marked as “Y”.

option = 3 (Send Reply)

Notifications where “Receiver Employee ID” = “login Employee ID” and “Response received” = ‘N’ will be displayed from “participants.txt” and “notifications.txt”. User will select “Notification ID” to send reply.

Enter reply for <Meeting ID> to <Organizer Employee ID> (Attend/Reject/Maybe) <ReplyText>

Update “Response received” = ‘Y’ and “Response Type” = <ReplyText> in corresponding line in “participants.txt”.

Notification for which reply is sent will be deleted from “notifications.txt” file.

option = 4 (Display Calender)

Enter Start Date <StartDate>
Enter End Date <EndDate>

Meetings scheduled in range <StartDate> to <EndDate> will be displayed ascending order of Meeting Date.
Only meetings organized by login Employee or where login Employee is participant will be displayed.

```
Meeting Date : 20/02/2020

Meeting ID     0100
Time          10:00
Description    Meeting to discuss deliverables of phase 1
Location       Conference Room 1


Meeting ID    0105
Time          14:00
Description    Meeting to do code review
Location       Conference Room 10

Meeting Date : 01/03/2020

Meeting ID     0110
Time          09:00
Description    Meeting to do performance review
Location       Manager Cabin 202

Meeting ID     0112
Time          12:00
Description    Agile team meeting
Location       Conference Room 5
```

option = 5 (Display Meeting)

Only meetings organized by login Employee will be displayed.
User will enter meeting ID from above list -

Enter Meeting ID <MeetingID>

All information regrading meeting from “meetings.txt”, “participants.txt”, “locations.txt” will be displayed in proper format.

6. Cancel Meeting

Only meetings organized by login Employee will be displayed.
User will enter meeting ID from above list -

Enter Meeting ID to cancel <MeetingID>

Are you sure you want to cancel this meeting (Y/N)? Y

All notifications corresponding to <MeetingID> will be deleted. "Meeting is cancelled" notifications will be added for all participants.

If user selects 'Y' all entries corresponding to <MeetingID> will be deleted from "meetings.txt", "participants.txt" files.

Assumptions: <Write assumptions made>

Technical Requirements -

1) C programming language
2) Use file input/output operations to read and write data.
3) Use multiple Linked Lists to read data from files at the beginning and write updated data to files before applicaton ends.
4) Use dynamic memory allocation.

Non Functional Requirements

1) Multi-file multi-directory solution is expected. Modular and maintainable code (comments) and all coding standards should be followed.
2) makefile to build application. Two-step compilation process - .o and then executable should be generated.
3) Use valgrind tool on application executable to detect memory leak. Final valgrind report to be submitted in "reports" directory.
4) Level 0 DFD (context diagram), Level 1 DFD, Flow diagram and pseudocode for 2 complex functions logic.
5) SRS in pdf format, RTM, Plan, Presentation. MOMs
6) HLD_LLD Document (optional)
7) Unit test cases and Integration test cases in UT_IT document. Both types of test cases i.e. sunny and rainy should be present in this document

Set Up Checklist for Project

Software Requirement:

Vi Editor, ctags, splint, valgrind, gcc, make, git account

Minimum System / Hardware Requirements:

Laptop with access to internet and Linux OS