**Personal Planner Application**

## Programming Assignment - Documentation

# Specification

## Main aims and objectives

The main aims of the personal planner program are to allow the user of the program to create plans, view what plans are happening on the current date, view plans, and edit plans. This is so that a user can write down small event/reminders that they can look back on and will remind them of something they have to do so that they do not forget or that they can plan out their day.

At first the user will be presented with a menu which they will be given a choice of what they would want to do, after each task (e.g. create plan) the user will be returned to the menu. The personal planner program is meant for multiple users, each user will have their own individual file. To create a plan the program will ask the user the plan title, plan date, plan time and the plan location. To make sure the plan date and plan time are in the right format the program will do a check first. These will then be entered into the file (the file is not meant to been seen by the user, so the format of the data entered into the file only matters for the program). To view a plan the user will enter the title of the plan they want to view the program will then search the file for the value the user entered and then output the file they requested to view. To edit plans the program will search the file that the user wants to edit it will then ask the user to re-enter the plan, this will then be put back into the file as an edited plan. The date today section will ask the user for the date they would like to search for, the program will then search the file for the plan, like the view a plan section, and then display this back to the user. A view all function will also be able to show all of the plans the user has in their file, this is so that a user can look at all of their files at once.

## Programming Requirements

|  |  |
| --- | --- |
| No. | Requirement description |
| 1 | The main programming part of the program will be file handling this will include creating, opening, reading and writing. the program should do this effectively. |
| 2 | The program will use a class, this class will hold the details, input functions and output functions for the plans the user creates. |
| 3 | Different aspects of the program will be split into functions. |
| 4 | Functions that have no part of the class can be split into header files, this makes the programs structure better and it is easier to understand. |
| 5 | The program will make use of vectors to store files into. By storing a file into a vector, it makes reading and editing the plan details easier. |
| 6 | For validation the program should make use of loops such as for loops and while loops as well as if statements. |
| 7 | To output and input data the program will use cout and cin. |

## User requirements

|  |  |
| --- | --- |
| No. | Requirement description |
| 1 | The program is required to run without errors. |
| 2 | The program should be easy to understand, the user should be able to instantly know what to press or type. This can be accomplished with proper instructions. |
| 3 | At no given time should to screen be cluttered, this would make reading the text on screen more difficult. The screen should be cleared after each section is completed. |
| 4 | The main menu section should have access to all sections of the program. This should be the only menu screen of the program. |
| 5 | The user should be asked their name so that the program can open their specific file which contains their plans. |
| 6 | The create plan section of the program should do just that. It is required to allow the user to create a plan, it should not do anything different to achieve anything else. This is to make it easy for the user to use and understand. |
| 7 | The view plan section of the program should do just that. It is required to allow the user to view a plan, it should get the user to enter a plan title and then find the plan that the date belongs to this plan should then be displayed to the user of the program. This section should not do anything different to achieve anything else. This is to make it easy for the user to use and understand. |
| 8 | If multiple plans have the same title the program will show the user both of the plans. |
| 9 | The view all section of the program must display all the plans in the users file. This is its only function. |
| 10 | The edit plan section of the program should not do other that getting the users plan and allowing them to edit it. This is to make it easy for the user to use and understand. |
| 11 | The what’s on today section of the program should get the user to enter a date and then find the plan that the date belongs to, this plan should then be displayed to the user of the program. |
| 12 | If the user has incorrectly typed in a value the program should give them another opportunity to enter it, this is to make the program to avoid frustration. |

# Design and Implementation

## Design brief

When the program is first loaded the menu will be displayed to the user, this will give the options: create a plan, view a plan, view all, edit a plan, what’s on today and exit. The menu will be made using a switch statement. If the user clicks on exit the program will end. To open a file in this program the user will enter their name, this will then open or create the file(if not already been created) called ‘name’.txt, the program will ask the user for their name every time they go to another section of the program that requires the file. This is so that multiple uses can use the same running program, this will be in a separate function to be called every time its needed. The create plan section of the program will open the text file and ask the users to enter the details of the plan they want to enter, the date and time will be subjected to validation to make sure they are in the right format and are ‘legal’ this will be done by separate functions. The details will then be added to the file. The view plan function will save the file into a vector this is to make handling the data easier and more efficient as all the lines in the file are saved as known lines in the vector. The user will then enter the title of the plan they want to view, the vector will then be searched for the plan the user has requested, if the plan is found the program will display it to the user otherwise the user can enter the title again. Viewing all plans will save the file as a vector but will instead display the whole file line by line to the user so that they can see all the plans they have made. The edit plan section of the program will first save the contents of the file to a vector, then ask the user for the plan title they want to edit (similar to the view plan section), the program will then ask the user to re enter the plan which will then be saved to the vector to be written back to the file. The what’s on today section of the program will work like the view a plan section but the user will search for their requested plan using the date instead of the plan title.

At first, I was struggling with a method of handling the data inside the file. I remembered about using vectors, specifically the ‘push\_back()’ function, which are very helpful when holding large amounts of data that can be changed.

## Flowchart

Program Start

Display Menu - give the user an option to create, view (by title or date) or edit

User picks an option

Create

view

edit

Asks the user for their name in order to open the file.

Asks the user for their name in order to open the file.

Asks the user for their name in order to open the file.

User inputs the new plan details: title, date, time and location.

User inputs the new plan details: title, date, time and location.

Valid date and time?

No

Yes

File is opened, if file is not present then it is created.

New plan is written to the file

Program end

exit

Returns to menu

File is opened into a vector

File is opened into a vector

User inputs the name (or date) of the plan they want to view

Plan found?

No

Yes

Full plan is displayed to the user

Returns to menu

User inputs the name (or date) of the plan they want to edit

Plan found?

No

Yes

User inputs the new plan details: title, date, time and location.

Valid date and time?

No

Yes

New plan is written to the file

Program flow chart

Flow Chart (View all)

Program Start

Display Menu - give the user an option to create, view (by title or date) or edit

User picks an option

View all

Asks the user for their name in order to open the file.

Program end

Full plan is displayed to the user

Returns to menu

Exit

## Use case diagram

Personal Planner Program

User

User file

Include

Include

Include

Include

Include

Include

Include

Include

Include

Include

Extend

# Testing

## Test Table

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test description | Test Input | Expected outcome |
| 1 | Test to see if the entry menu works as expected. Break point placed after each | Value from the menu (e.g. 1) | The program will open the section corresponding to the number entered (if 1 then create plan will open) |
| 2 | Test to see if the entry menu works as expected if the wrong value has been entered. | Value that is not stated to be an option by the menu (e.g. test) | The program will ask the user to re-enter the value until a valid on is entered. |
| 3 | Test to find out if the program creates a file if one is not present. | User enters their name and a file called ‘name’.txt isn’t present. | The program will create the file. |
| 4 | Check that the validate date function doesn’t allow for an invalid date to be inputted. | 31 02 2019 | The date is invalid, and the user is asked to enter the date again |
| 5 | If the user enters a word value for the date the program should tell them to re-enter the date the same as if a wrong number is entered | ‘Test’ is inputted when the plan date is asked for in the create plan section | The program will ask the user to input the date again |
| 6 | If the user enters a word value for the time the program should tell them to re-enter the time the same as if a wrong number is entered | ‘Test’ is inputted when the plan time is asked for in the create plan section | The program will ask the user to input the time again |
| 7 | Check that the validate time function doesn’t allow for an invalid time to be inputted. | 24 30 | The time is invalid, and the user is asked to enter the time again |
| 8 | Test to check if the date entered is put into date format after the user enters the date, they want | 2 4 2011 | The program should turn these values into 02/04/2011 |
| 9 | Test to check if the time entered is put into time format after the user enters the time, they want | 3 0 | The program should turn these values into 03:00 |
| 10 | Test to find out if the plan the users entered plan is properly inputted into the txt file. | User completes the create plan section of the program | The plan will be imputed into the text file with each detail being on a separate line. |
| 11 | Test to find out if the program makes the user enter the name of the plan, they want to view again if they enter a plan that is not present in the file. In the view plan section. | User enters a plan title that is not present in the txt file they have opened | The program will get the user to enter the name of the plan again |
| 12 | Test to find out if the program displays multiple plans to the user if there are plans present that have the same name in a file once searched for. | User creates multiple plans with the same name and then searches for them in the view plan section. | The program will display the plans one after another. |
| 13 | Test to find out if the program replaces the old plan with the edited plan once edited. | Complete edit plan section and then view all plans to check if the plan has changed. | The plan will be edited and will be in the same position as the old plan |
| 14 | Test to check if the view all section of the program outputs all the plans in the file. | User completes the view all section. | All the plans will be outputted |
| 15 | Test to see if the user enters a title instead of a date in the whats on today section, will affect the program. | User types in the plan title instead of the date | The validation on the data shouldn’t allow the user to enter the title. |