Planner4Life – Project Manual
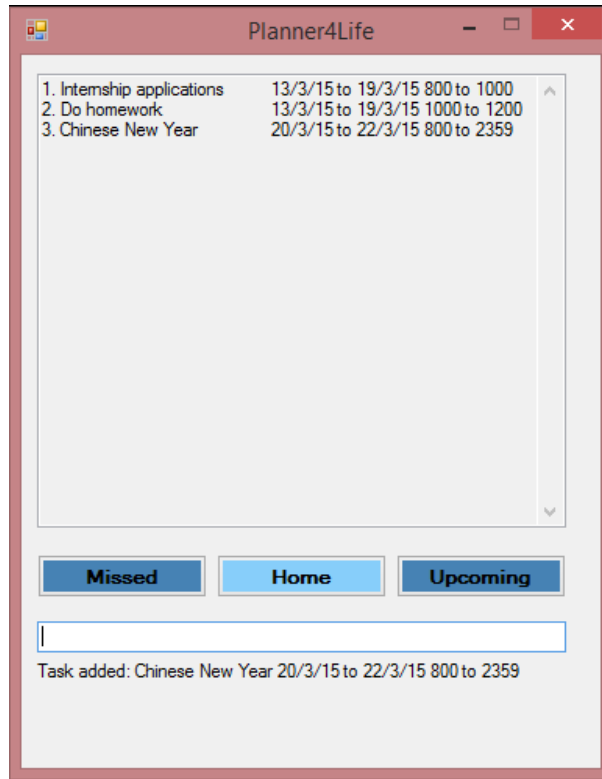
Supervisor: Huang Da   **Extra feature:** Recurring Tasks

Sakib Bin Farooque Rahmatullah Team lead, Deliverables and Deadlines	Le Minh Thu Documentation, Integration	Karthikeyan s/o Shanmugam Testing, Code Quality	Lee Thye Jie Scheduling and Tracking, Testing

Planner4Life
User Guide V0.1

Congratulations on downloading Planner4Life User Guide! Planner4Life is your new companion for life. It keeps track of your various tasks and deadlines. This User Guide provides you with detailed descriptions of Planner4Life's functionalities to enable you to use it effectively.

1 What is Planner4Life?

Planner4Life's primary function is to organize your tasks and deadlines. As a user you can add a Task on a particular date and attach the desired information to that task.

Planner4Life will then display these tasks in chronological order so that you always stay on top of your work!

2 Anatomy of K5

2.1 Types of Views

Planner4Life consists of 3 Views for uncluttered access: Home, Upcoming and Missed. The current View is indicated by a highlighted button with its View name. Users may switch between Views by selecting the corresponding buttons, or by keying in <home>, <upcoming> or <missed> for their respective views.

Home View
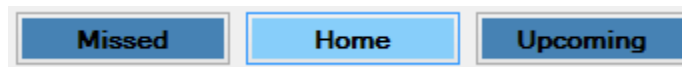
The Home View displays all the tasks for the current day and the next consecutive 7 days. Tasks that have no date indicated are also shown in the Home View.

Upcoming View
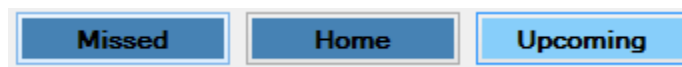
The Upcoming View displays all the tasks with dates past the coverage of the Home View. Users are able to browse distant and recurring tasks in this View.

Missed View

The Missed View displays all the tasks with a due date that has passed. When a task has passed without completion, the Missed button will turn red to notify the user that he has missed a deadline.
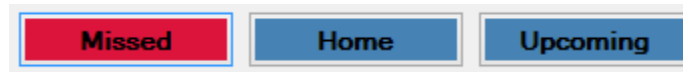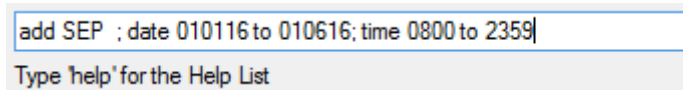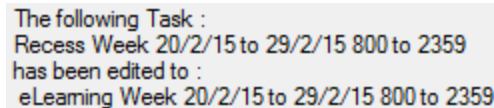
3 Keying in Tasks

add SEP ; date 010116 to 010616; time 0800 to 2359

Type 'help' for the Help List

Users key in tasks in the text bar that can be found below the View buttons. Tasks are processed when the <Enter> key is pressed.

4 Operation Feedback

The following Task :
Recess Week 20/2/15 to 29/2/15 800 to 2359
has been edited to :
eLearning Week 20/2/15 to 29/2/15 800 to 2359

After each successful operation, Planner4Life will display feedback below the user input bar to let the user know what has happened. A useful prompt is also always present to remind the user to the type of input that is expected. Displayed above is the feedback for *edit*.

5 Command Formats

There are 7 basic commands: *add, delete, undo, search, clear, edit* and *help*.

5.1 Add a task

Input format: *add* <task description>; date <start date> to <end date>; time <start time> to <end time> #

The *add* function allows you to enter information fields into Planner4Life and store your tasks. Separate information fields are separated with the < ; > symbol. Information fields are input with a preceding keyword. The full list of keywords can be found below.

The order of the information fields entered does not matter. Irrelevant information fields can also be left out as well.

If the operation is successful, a prompt will appear below the input bar as confirmation.

Keywords	Function
date <start date> to <end date>	To specify start and end dates. The date input format is ddmmyy.
time <start time> to <end time>	To specify time period i.e. time 0800 to 1300. Time is input in 2400hr format.
#	Marks the task as important.

Examples

add meeting at office; time 0930 to 1030; #impt

add concert at esplanade

add assignment; date 020515 to 270215

5.2 Edit a task

Input format: *edit* <task index >; <key in new task in *add* format>

The *edit* function allows the user to edit specific details of an existing task.

Each task shown on the screen is numbered in increasing order. To edit a task, identify the desired task number and re-enter the task to make the change. A detailed example is shown below.

If the operation is successful, a prompt will appear below the input bar as confirmation.

Example

Original task: 2. meeting 02/02/15

Enter: *edit* 2; add meeting; date 030315

Final task: 2. meeting 03/03/15

5.3 Delete a task:

Input format: *delete* <task index>

The *delete* function allows the user to delete tasks identified by its corresponding task number.

If the operation is successful, a prompt will appear below the input bar as confirmation.

Example:

Task to delete: 3. Meeting in office 0900

Command: delete 3

5.4 Clear tasks:

Input format: *clear* followed by <Y> or <N>

This command clears all the tasks in K5 Planner. Upon pressing entering *clear*, a dialog bar will appear asking you to confirm your operation. Enter <Y> for Yes or <N> for No. Do note that *clear* cannot be undone and will delete all your information from the database.

5.5 Undo:

Input format: *undo*

This command undoes accidental deletes, edits or adds.

This command can only be executed immediately after a successful *add, delete* or *edit* command.

Function	Effect of *undo* on function
Add	Removes the previously added task
Delete	Replaces the previously deleted task
Edit	Restores the original task before the commit

5.6 Search:

Input format: *search* <target word>

The search function allows the user to search for a <target word> in all tasks in the planner. All tasks that contain the <target word> will be displayed in the Display Window. There will be an index number associated with each search result and users can reference this number for future operations.
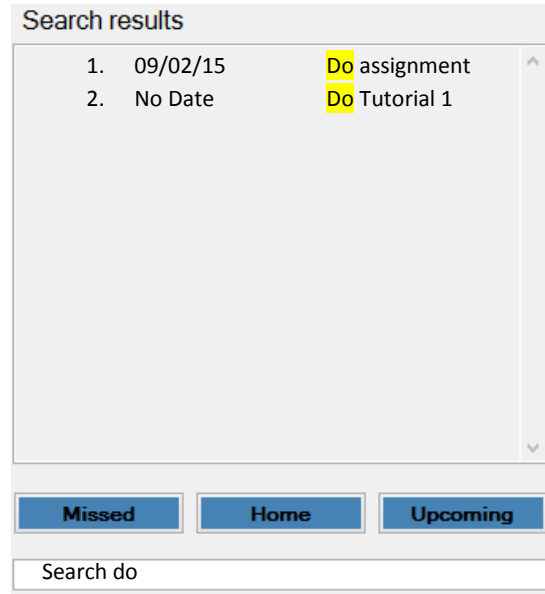
5.7 Help:

Input format: *help*

This command displays a help list in the Display Window that lists down the various commands and their functions.
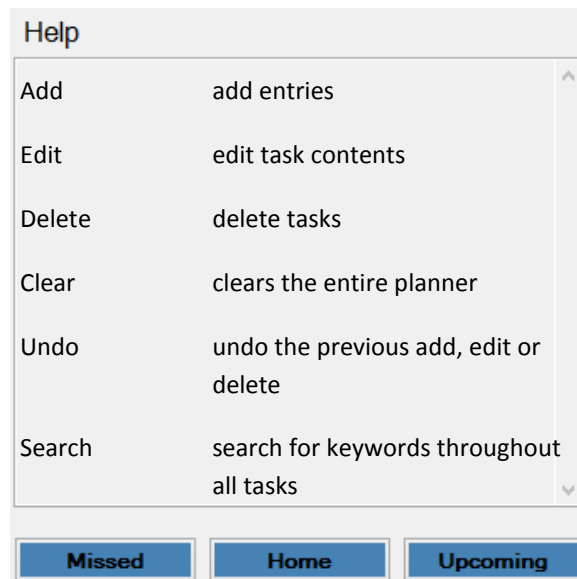
Appendix A: User stories - As a user, I can…

High Priority

No.	Functionality (As a user I can…)	Purpose
1	add tasks in a single command	so that I work faster
2	delete tasks in a single command	so that I work faster
3	undo changes quickly	so that my mistakes are not permanent
4	use the search feature	to search for any tasks from the list
5	can change specific details in tasks	so that I don't need to delete the old entry and enter a completely new one
6	view my tasks categorized by importance	so that I know which one needs more attention
7	view my tasks in chronological order	so I know which to do first
8	mark my tasks as completed	so I know which are done
9	add start and end tasks for a task	so that I know when to start and end the task
10	can receive notifications for upcoming deadlines	so that I know a task needs to be completed soon
11	I can record: Tasks, Time, Venue, Remarks	so I can view and edit them individually
12	I know that there is a save feature	so that when I close and reopen the app, my information is saved
13	I can navigate through the tasks easily	so I can view the list easily
14	I can view past tasks and events	if I ever need to recall anything
15	I can scroll down the list of items	to view the list easily
16	I can click on help if I am unsure	so that I can refer to the commands and shortcuts
17	I can enter recurring tasks easily	so that I don't have to enter them one by one
18	I can save the storage file anywhere	so that I can access it on different computers
19	I can use the application even	so that I can use it anywhere

	without Internet	

Medium Priority

No.	Functionality (As a user I can…)	Purpose
1	receive notification for missed deadlines	so that I know that a deadline has been missed
2	hide outdated tasks	so that if I don't want to view them, they won't be on the screen
3	use hotkeys to access the application	so that it is easily accessible
4	choose to have the application open in one corner of the screen	so I can easily view upcoming tasks

Low priority

No.	Functionality (As a user I can…)	Purpose
1	enter tasks using short form	so that it is faster
2	be notified if the task being entered has a duplicate	so that I can avoid duplication
3	can add checklists to multi-stage tasks	so that I can keep track of smaller tasks that fall under a major task
4	I can block multiple time slots if unsure about a task and the application will automatically delete the other time slots when one is confirmed	so that I can place task in planner even without a confirmed timing and prevent clashes
5	I can sync my tasks to other devices	so that I can use the planner from other devices
6	I can have suggested words that I commonly use	so that I can type faster
7	I can sync my planner with Google Calendar	so that I can use both the application and google calendar

Appendix B: Non-Functional Requirements

Requirement

K5 should automatically save any changes made after each action performed

K5 should be able to save calendars as readable text files so that the user can easily edit the contents directly if he/she so wishes

K5 should be able to run as a standalone .exe file

K5 should be able to perform CRUD functions with no perceivable lag time

K5 should be compatible with computers that run Windows XP, Windows Vista, Windows 7 and Windows 8.1 Operating System with Microsoft .NET framework 4.0 installed

K5 should maintain a clean UI that can be easily read at all times

K5 should conform to C++ coding standards so that other developers can pick up the project

Development of K5 should be documented diligently so that the development process is accountable

Appendix C: Product survey

Product: Any.do **Documented by**: Karthikeyan s/o Shanmugam
Strengths:
- Easy to add new tasks – just click add and type in task and hit enter
- Easy to mark as done – just mouse over task, click on the tick icon that appears
- Easy to update and delete
- Tasks marked as done continue to appear at the bottom with a strikethrough allowing user to keep track of work that is done.
- Can sync with phone and computer as long as there is internet.
- Good categorisation - can sort according to "today, tomorrow, upcoming, someday" or into groups "personal, work, <add own name> etc"
- Can add a text note (picture, video, camera, attachment, voice recording)
- Can add subtasks
- Can add Reminders for tasks
- Can collaborate tasks with other people using same application
- Can mark tasks as important or special attention
- Can order tasks according to importance/priority
- GUI is simple, sleek and easy to use
- Has good flexibility since it is not really a command input type organiser. Just input exactly what the task is, and it appears as so.
- Can search for tasks by typing in keywords
- Supports recurring tasks

Weaknesses:
- Web-based application – so there is a need to go to the site to use it. Not a desktop application
- Since its web based, it cannot be accessed on computer without internet. Possible to access in on phone app without internet access.
- Cannot undo changes
- No support for timed task. Since its purely text based, not like a calendar where it is possible to block out time, it will not be able to tell if there is a clash in timing of task

Product: S Planner **Documented by**: Le Minh Thu
Strengths:
- Ability to sort tasks in chronological order of due dates
- Ability to sync tasks to Google Calendar, which can be accessed using computer
- Ability to differentiate tasks of different priorities
- Ability to notify conflicting events
- Ability to use the program without internet

Weaknesses:
- Cannot categorize tasks according to user's preferences
- Cannot clear the entire list of tasks

Product: Outlook Calendar **Documented by**: Lee Thye Jie
Strengths:
- Allows syncing across devices
- Colour-coded entries for different categories
- Ability to import/export new calendars
- Clear indication of the current day
- Extensive information fields for individual entries e.g. Location, Reminder etc.
- Simple, clear interface

Weaknesses:
- No search function
- Unable to add tags to entries
- If there are too many entries on a day, the newer entries are not visible until the view mode is changed

Product: Google Calendar **Documented by**: Sakib Bin Farooque Rahmatullah
Strengths:
- Ability to take in info by single command
- Ability to colour code items
- Ability to rank items by priority
- Ability to set reminders
- Ability to add locations
- Ability to add remarks
- Ability to set durations
- Ability to recurring events

Weaknesses:
- Settings page is very large and troublesome to change items one by one
- Need internet to sync

Planner4Life Developer Guide V0.1

Planner4Life Developer Guide

Planner4Life is a standalone .exe file that performs the functions of a calendar planner. It allows users to input multiple tasks with their description, date and time, with which Planner4Life will create a running schedule. Planner4Life also contains functions that allow the user to perform CRUD (Create, Read, Update and Delete) processes on the schedule, as well as to organize the content into easily readable formats.

1 Challenges

- Implementing persistent memory for Planner4Life using a standalone .exe file
- Creating an architecture that is as modular as possible to facilitate future development and easy testing

2 Principles

Ease of Use

We believe in creating a planner application that is easy for the user to learn and use efficiently.

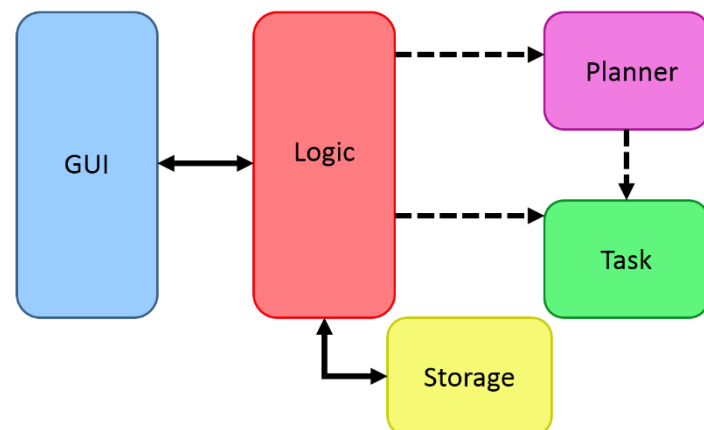*Clean code*

Writing code that is both as concise as possible and easy to read.

Safe code

Application of defensive coding; comprehensive exception handling capabilities that provides consistent feedback to the user.

3 Product Architecture

14

Given below is an overview of the main components.

Task

Contains variables for storage and organization of the various information fields of a particular task entry. It also includes their getter and setter functions.

Planner

Organizes and arranges Task objects into their proper orders and lists (eg. Missed list, upcoming list). Contains functions to manipulate the Task lists as well as getter and setter functions. It is also able to convert Task lists into readable data that will displayed.

GUI

The GUI is seen by users. It is the platform of communication between the program and users. It takes in users' commands, provide users with feedback on the outcome of the commands and displays different views (each view is a different Task list) of events depending on users' preference.

Logic

Logic is the central dispatcher of the program which receives user input from GUI and requests relevant components (Planner, Task and Storage) to carry out appropriate operations on the user input. The content to be displayed as well as outcome of operations are dispatched from Logic to GUI to be viewed by users.

3.1 GUI

The GUI is a CLR Windows form that takes in the user input and displays the output from Planner4Life. Its primary objectives are to display the information in a well-organized manner to the user. By maintaining semantic meaningfulness of the information displayed, the GUI allows the user to understand his schedule.

All GUI pages are displayed in the Display Window. The GUI consists of two categories of pages:

Schedule Content Display

Schedule Content Display windows are accessible through text commands or button clicks.

- <Home> displays the tasks for the next 7 days.
- <Upcoming> displays all the tasks past the next 7 days till the last entry.

- <Missed> displays tasks that have concluded prior to the current date.
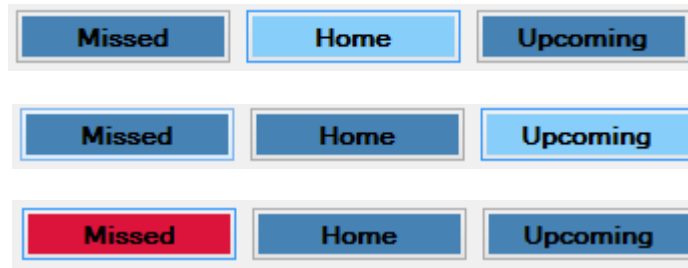
All pages are indicated by their respective highlighted button.

Program information pages

Program Information pages are only accessible through text commands.

- <Help> displays a help window.
- <Search> displays the search results that match the search keyword.
- <All> displays every task entered into Planner4Life, including past tasks.
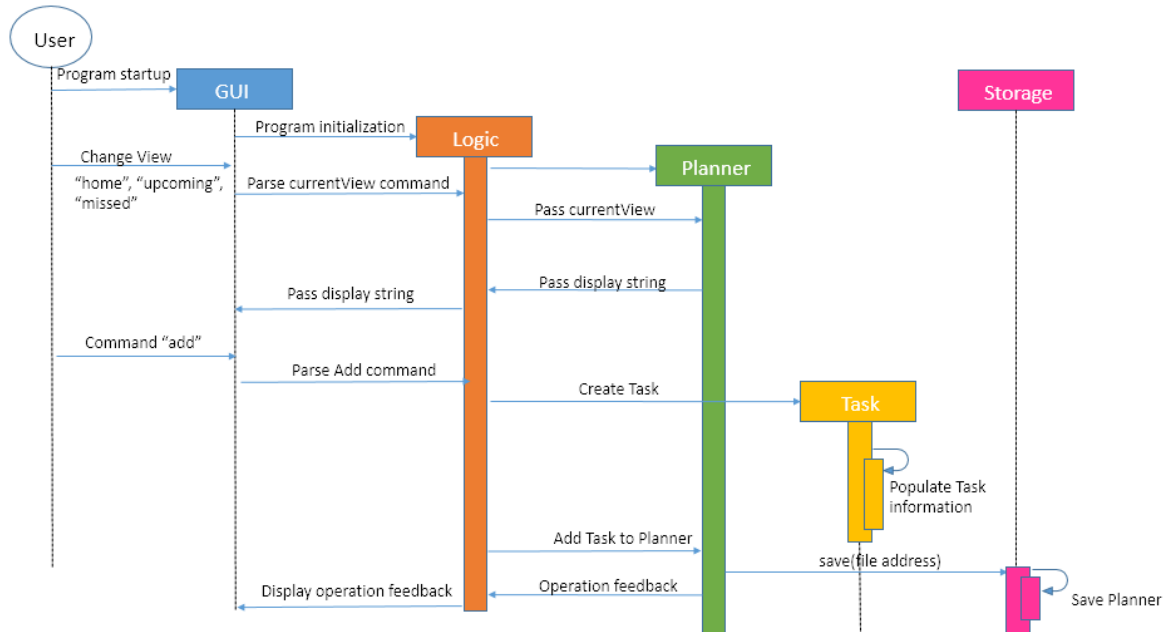
Request flow

1. Take in command-format String (User Input)
2. Convert String to string
3. Pass string to Logic
4. Logic processes the commands in the string
5. Logic return output-ready string
6. Convert string to String
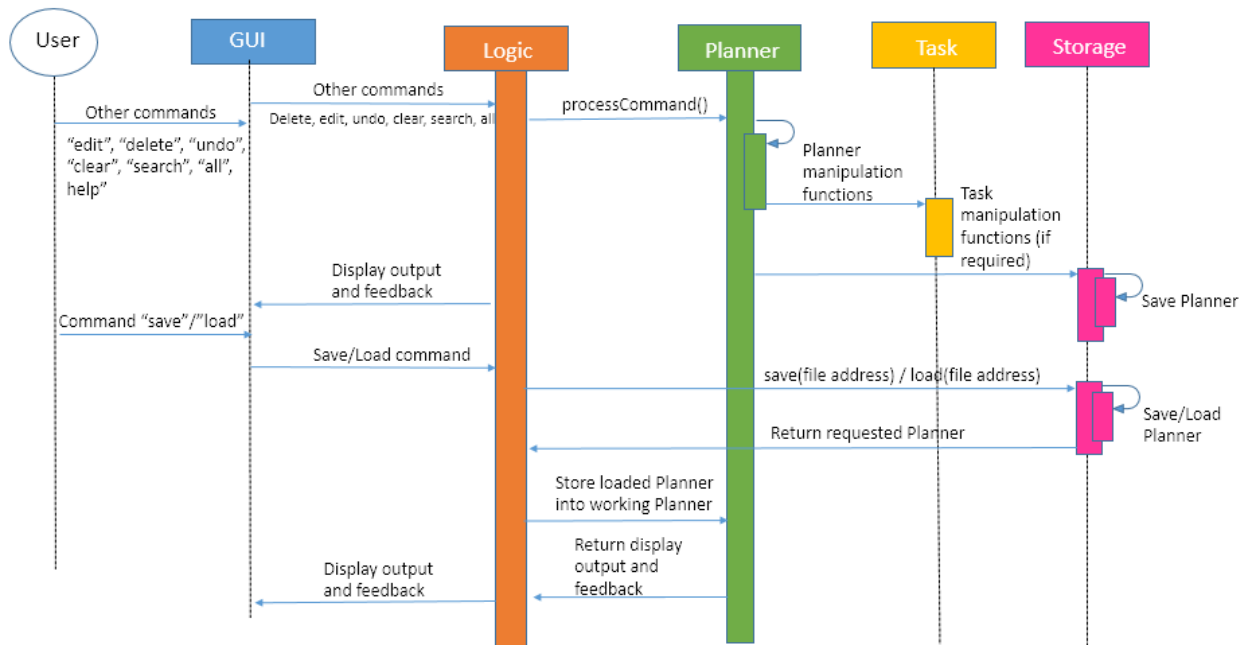7. Print output in Display Window
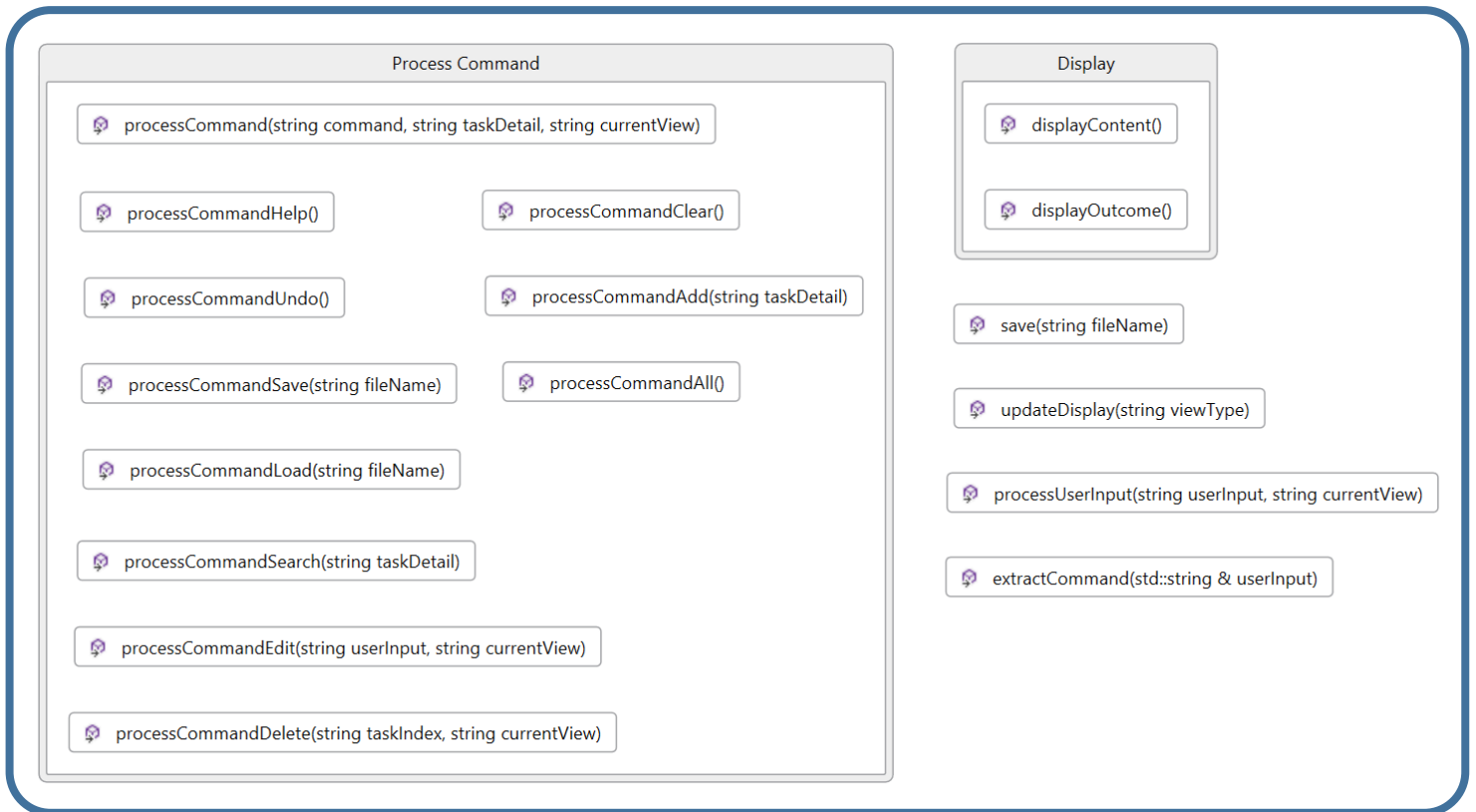8. Print output in feedback prompt

Sequence Diagram

- Functions: Startup, Change View and Add

- Functions: Other CRUD commands, Save/Load

3.2 Logic

The Logic component handles the business logic of Planner4Life. It takes in command input from GUI and processes the information using the Planner and Task Classes. It then receives the feedback from the Planner and Task Classes and passes it back to GUI for the user to view. Logic also makes use of the Storage in order to save and load data.

In particular, Logic is responsible for these:

- Directing the user commands to the correct functions
- Passing the correct information to be displayed
- Activating save and load features

Logic API

Important public functions:

void processUserInput(string userInput, string currentView):

Takes in user input and the current view from GUI and classifies it into the appropriate type of operations based on the command word

Update the content to be displayed on GUI

string extractCommand(std::string& userInput):

Takes out the first word in the original user input and identifies it as the command word

processCommand functions (add/edit/delete/help/all/search/load)

Dispatches task details and current view to the relevant function depending on the command word

displayOutcome() and displayContent() functions:

Provides GUI with outcomes of operations and content to be displayed

void save(string fileName):

Allows all tasks to be saved to a .txt file

3.3 Storage

The Storage Class manages the save and load features in Planner4Life. It receives the data to be saved from the Logic Class and saves it in a text file. It is also able to receive, through the Logic class, a filename and/or save location entered by the user and hence gives the user flexibility to choose the filename and location. Lastly, the Logic Class is able to obtain data from Storage on startup through the load functions.

In particular, Storage is responsible for these:

– Save data from the major list into a text file
– Load data from the text file in to the major list
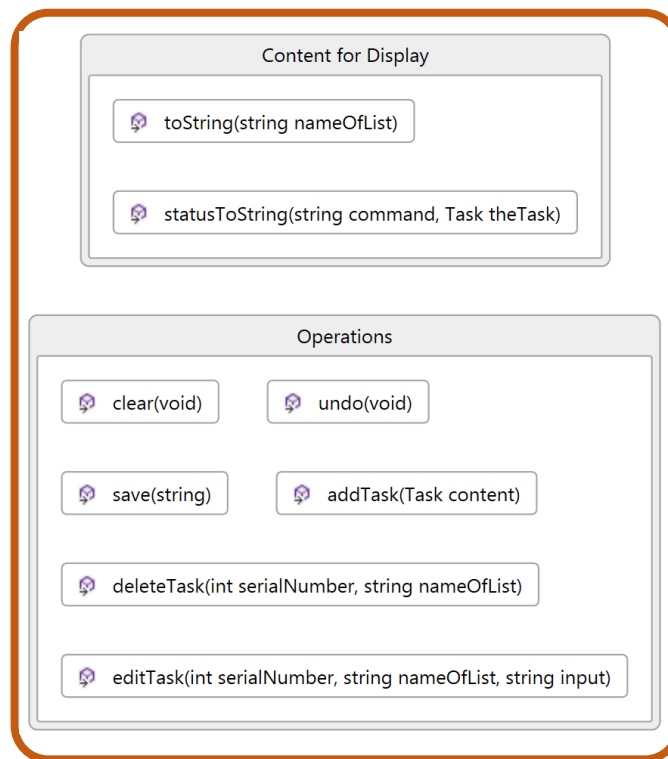– Set filename and file locations

Storage API

void save(string fileName, string fileLocation)

 Saves the files to the selected location as a .txt file and with a desired filename.

void load(string fileName, string fileLocation)

 Loads the files from the selected location as a .txt file and with the requested filename.

3.4 Planner

The Planner class manages all tasks created by users, by categorising them into the relevant lists. It receives new tasks from Logic and adds them to a master list of all tasks in chronological order, and the master list aids in the formation of the other lists. The Planner class methods (delete, edit, search, undo, clear and save tasks) are called by Logic to perform appropriate operations on the master list depending on the commands given by users. It also provides Logic with the content to be displayed in the different views of GUI as well as the status of operations.

In particular, Planner is responsible for these:

- Perform operations on the master list of all tasks
- Generate content to be displayed in the different views of GUI
- Generate status of operations to be displayed GUI

Planner API

Important Attributes

List of Tasks: All, Home, Missed, Upcoming

The All List contains all the tasks inserted into the planner. The Home, Missed, Upcoming Lists contain the various relevant tasks depending on the date.

Important public functions:

string <process>Task(Task content)   process: add, delete, edit, undo, clear, save

The above functions will manipulate the All List according to the command entered by the user.

string toString(string nameOfList);

The toString function passes the display of the various Lists to the Logic Component on request.

string statusToString(string command, Task theTask);

The statusToString function will return feedbacks and confirmations of the major functions being successful or unsuccessful.

3.5 Task
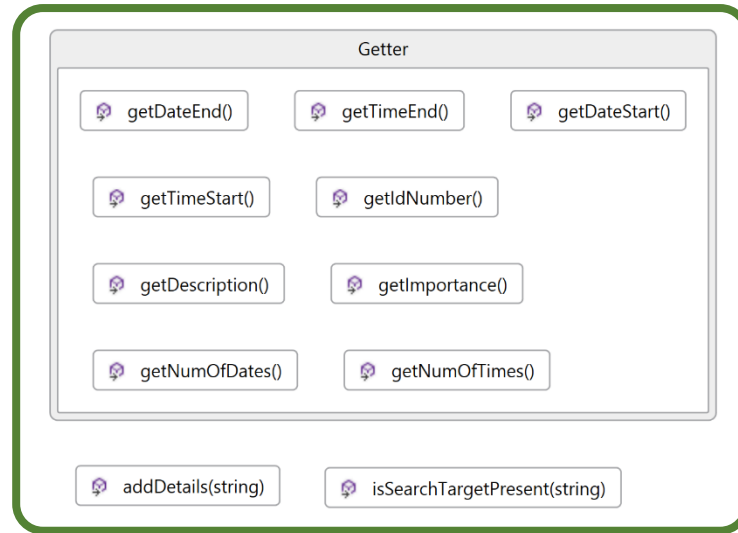
The Task class is in charge of creating new Task objects when prompted by Logic as well as providing Planner with information of Task objects.

In particular, Task is responsible for these:

- Create new Task object
- Retrieve information of existing Task object
- Search for target word

Task API

void addDetails(string):

Receive task details from Logic, parse the details and store the relevant information in a newly created Task object.

bool isSearchTargetPresent(string):

Search for keyword in the description attribute of the Task object

Getter functions

Retrieve the required information from the Task object