

Crit B Design

Input Data

To maximise the versatility of my planner, it is important to offer various input options for information regarding the plan - Implementing only a singular text field intended for the plan to be written would feel clunky and minimise versatility of my product. I want to ensure my client can input whatever data would best aid them, and provide the necessary space to do so. To ensure the user can maximise usage from this planner, I will create multiple editable text fields labelled with their corresponding intended function.

The input options will be:

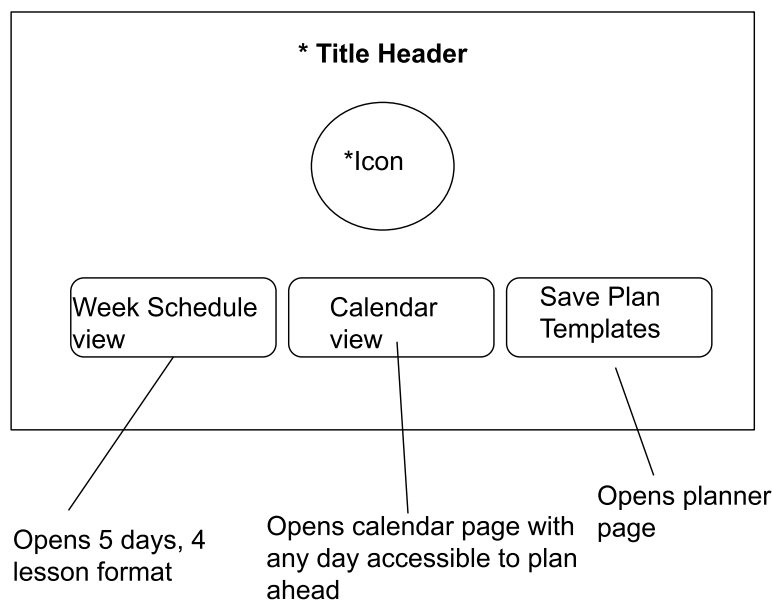
- Subject
- Lesson Target/Aim
- Duration
- Required materials
- Plan Details
- Additional Notes

All of these will be text fields of varying size (The Duration field will be comparatively smaller than the Plan Details field) accessible and editable on the same page.

The text in each of these fields will be recorded if the “save” button is clicked. All the text will then automatically load into the corresponding text field the next time the plan page is reopened.

Software Interface: GUI designs

Main Window:



Clicking on “Week Schedule view”:

	Monday	Tuesday	Wednesday	Thursday	Friday
Period 1	...	...	...	...	...
Period 2	...	...	...	...	...
Period 3	...	...	...	...	...
Period 4	Japanese	...	...	...	...

Button names could change according to “Subject” input in Planner page

Clicking on button opens Planner page specific to that Lesson e.g. Thursday, Period 4

Clicking on a particular lesson will open the Planner page:

Monday, Period 4

Subject

Lesson goal: Length:

Materials:	Plan:	Notes:
...	...	...

Clear Save Load

Clears the text fields as well as the save

Saved data automatically loaded in fields when opened

All text fields editable

Saves text in each field

Allows user to load template of choice

Clicking on the “Calendar view” button on the *main page* will open the Calendar page:

Drop down options
for year and month

Year: ▼ Month: ▼

Mon	Tues	Wed	Thur	Fri	Sat	Sun
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Wednesday, Jan 22

Period 1

Period 2

Period 3

Period 4

Clicking on a Day button will open a unique Day view page →

Clicking on a specific period will then open the Planner Page again.

Clicking on the “Save templates” button on the *main page* will open a Planner page, but without the “Load” button:

Template

Subject: File name: Determines the .txt file name

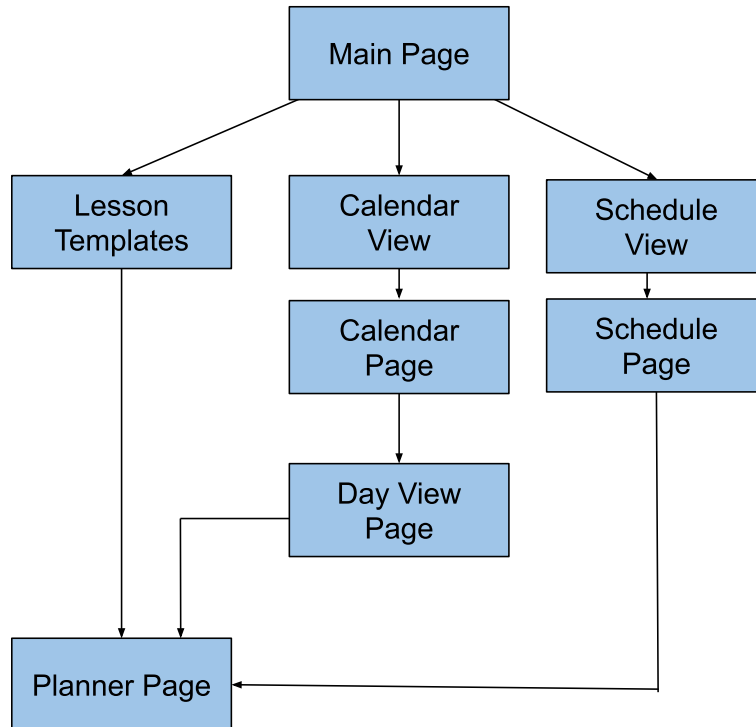
Lesson goal: Length:

Materials: Plan: Notes:

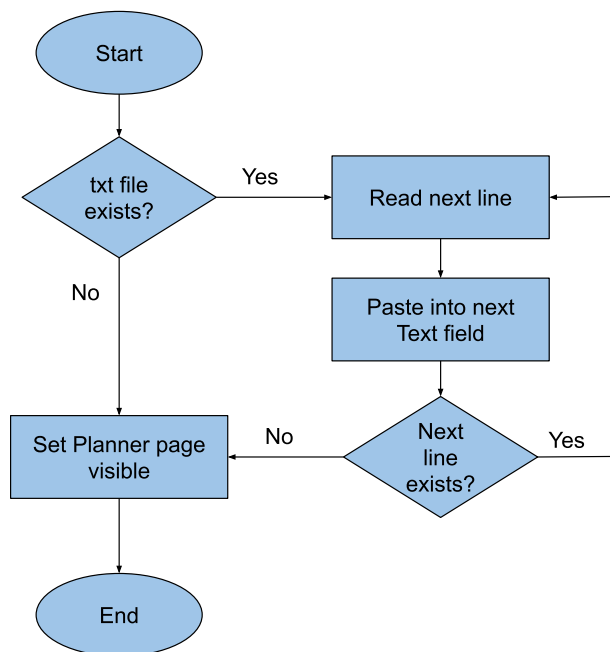
Clear button will only
clear text fields

Diagrams and Flowcharts:

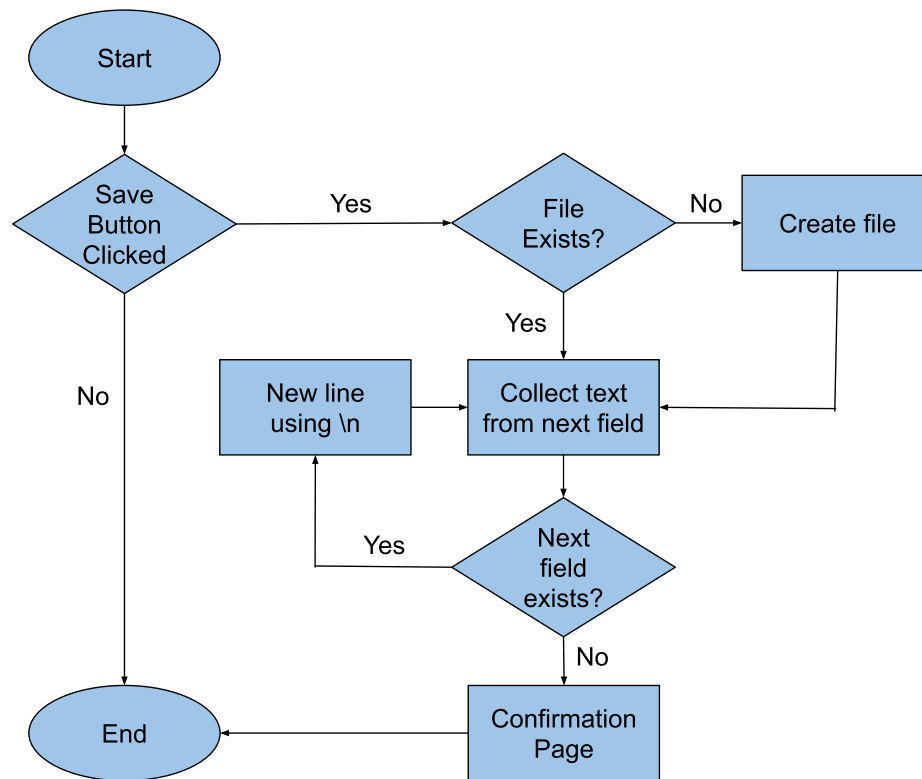
To represent the relationship between the various pages of my lesson planner program I constructed a structure diagram to clarify the sequencing:



Opening the Planner Page will first run a search to check if a .txt file exists. If it does, line by line, the text will be pasted into each text field.



To ensure the above works smoothly, the data needs to be saved in such a way that the text from each text field is placed in a new line each time.



Important Variables

Main Page:

In order to create a GUI, I will likely be using Java Swing library class. This means that I will be using data types such as JFrames, JPanels and JButtons. For the homework icon on the mainpage I will probably use the Image Icon data type, also from Java swing. The main page will mostly just contain GUI components as no data processing occurs here.

Variable	Type
MainFrame	JFrame
Panel	JPanel
CalendarButton	JButton
ScheduleButton	JButton
TemplateButton	JButton
Icon	ImageIcon

Schedule Page

I will try to initialise the 5x4 grid of buttons using a loop to decrease the amount of coding - writing a new line of code for every button would not be productive. Day names could appear on the top of the column, while lesson numbers would appear to the left of each row.

Variable	Type
DayButtons	JButton
Daynames	JLabel
Lessonnum	JLabel

Planner Page/Template Page:

I know that this page will require many textfield variables to record the different aspects of the plan, length, aim, materials, etc. As for the save functionality, I am not particularly familiar with File handling in Java at the moment, but I know it will be necessary. This is something I think I will learn during the process of making the product.

Variable	Type
SubjectField	JTextArea
GoalField	JTextArea
LengthField	JTextArea
MaterialsField	JTextArea
PlanField	JTextArea
NotesField	JTextArea
TemplateNameField *only for save template*	JTextArea
File *name depends on planner page object name*	File

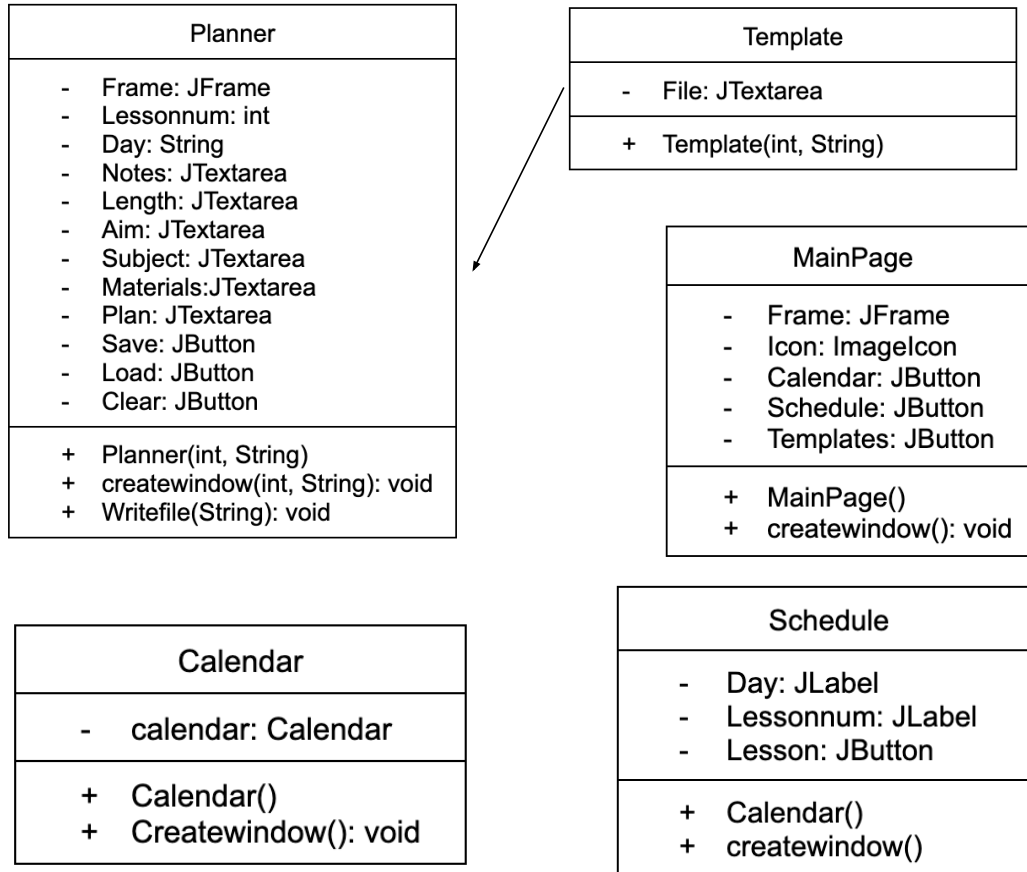
Calendar Page

While researching online, I found a calendar library class, with all the code and methods already created. Rather than making my own, which will be time-consuming and tedious, I can simply use this open-source code.

Variable	Type
Calendar	Calendar

UML Diagrams

Given the various pages in my software, it might be beneficial to separate all of them in classes, with their own variables and methods. To formulate a rough idea of what these classes would contain, I constructed some UML diagrams.



Firstly regarding the Planner class, as each planner page would correspond to a button on the schedule page, each planner object needs to have a unique identifier. For this, I made the Day and Lessonnum variables be passed into the constructor method as parameters.

Note also this is simply a prototype of my classes - the final product will most certainly differ. I can't plan ahead perfectly what variables are needed and what should be passed as parameters. For example, rather than defining them as class variables, GUI components (JButtons, Textareas, JFrames) could simply all be initialised and used in the constructor methods or the Createwindow method. Such variations in class structure could occur.

As the template class is simply the Planner page with a slight variation (an additional "File name" textfield), I decided to incorporate inheritance, indicated by the arrow from the Template UML diagram to the Planner UML diagram. With all the methods for creating the page already prewritten, this would greatly cut down on time when coding the template class.

Testing Process

Success Criteria	Testing Method
Main Page opens on Start, with loaded image and 3 buttons, "Schedule View", "Calendar view" and "Save Template"	Start the program on IntelliJ, check if all loads accordingly
Once "Schedule" Button is clicked, it opens a grid of 4x5 buttons to correspond to lessons.	Click on "Schedule", check if buttons load as expected.
Planner page opens without complications, with all editable text fields and 3 buttons, "Save", "Clear" and "Load".	Click on a random lesson button on the grid, see if the Planner page opens accordingly. Check with different lesson buttons as well.
Lesson plans are saved on the computer as .txt files, when the "Save" button is clicked	Click Save, check if the confirmation window opens. Navigate to the project folder and see if .txt files appear.
Saved lesson plans are automatically loaded onto the text field once a lesson plan has been opened	Save a plan, close the program then restart. Navigating back to the same lesson button, the saved plan should automatically load.
If "Load" button is clicked, it allows the user to choose a .txt file plan. Selected plan text will then load onto the various text fields.	Press "Load", check that a window opens with the correct file path to where the .txt files are stored.
If "Clear" button is clicked, it clears the text fields and clears the corresponding lesson plan file	Press "Clear" and see if all text is erased. Then navigate to the corresponding .txt file, check that this file is empty.
Clicking "Templates" will open a Planner page without "Clear" button. An additional text field will also appear called "file name"	Press "Templates", check if the Planner page loads accordingly with the variations present.
Saving templates will save the .txt file in an accessible directory, using the specified file name identifies the file.	Navigate to the folder "Templates" and check if the file is there
When "Calendar" is clicked from the Main page, it opens a Calendar with drop down options for Year and Month.	Press "Calendar" and see if the JCalendar object loads accordingly. Drop down menus for months and years should be interactable.
Clicking a day from the Calendar will open options for periods, which open a Planner page when clicked.	Press "Calendar", check that the calendar loads properly, with all the drop down options interactable. Click on a day, see the Dayframe opens and the Planner page opens accordingly.