**Abstraction - Design Activity**

**Journal:**

Behaviors:

* We will have a Display method that it can show the journal entry
* We will have a Save method so that it can save the journal to the computer
* We will have a load method that will let us be able to load it into the main program
* All three methods will return void

Attributes:

* We will have a list of entries that is taken from the Entry class
* We will have a filename that is a string so that we will be able to save and load it

**Entry:**

Behaviors:

* We have a random prompt method that returns void to choose a random prompt to the user
* We have a save entry function that saves what the user responds with

Attributes:

* We have a date string to keep the date of the entry
* We have a prompt variable that is a string that gives the user a prompt
* We have a response variable that is a string where the user will respond

**Saving/Loading:**

* The Save and load function will be in the journal section so it can save and load new entries. We will have a save entry function but don’t know if we need a load entry in case we want to see one entry.

**Prompt Generation:**

* We are going to have two prompt sections, one section is for the main function of the program where it gives random prompt entries, but we will also have a series of prompts for each of the menu options.

**Interaction:**

* The Program class is going to have the three classes in it so that is where most of the abstractions is going to happen. We also will have abstraction with the journal getting the list of entries from the entry class. In the entry class we will use the prompts that were created in the prompt generator class.A list of tasks on a notebook

  AI-generated content may be incorrect.A group of white boxes with text

  AI-generated content may be incorrect.