**National Park Journal for Outdoor Enthusiasts: Project Spec**

**General Description**

To help users capture memories from their adventures in national parks, I plan to create an application that enables a user to journal about their travels to national parks in the US. The primary interaction users would complete includes creating journal entries by selecting a national park they’ve visited, logging details about their experiences and memories at the selected national park, recording hikes and activities they’ve completed at the selected park, and rating the national park. Additionally, I envision this app also providing the capability to help a user learn information about different national parks, so the user could decide which national park(s) to visit next!

The primary users of this interface will be individuals who enjoy traveling to and visiting national parks in the US and want a way to digitally capture, or log, details of their memories at the national parks they visit that can be accessed later. Additionally, users of this application will include individuals seeking to learn about various national parks. The application will help users digitally record and store their memories in one place, preventing the need to record their experiences by hand on paper and store the record somewhere with the potential to be lost or misplaced.

This application will be programmed as a desktop GUI application using Tkinter, which will support much of the heavy lifting, within a jupyter notebook. Data for this application project will ideally be stored in a CSV file, but as a backup, data could also be in the form of nested lists or dictionaries. Lists/dictionaries might not be as clean as having a CSV file, but this is still a doable option since the dataset will be fairly small (less than 100 rows). For the primary interaction involved with this project, I envision, at a minimum, needing a dataset that includes a list of US national parks and their location by state. For the interaction of learning about national parks, I would also have additional data on national park descriptions and costs, as well as a link that would take a user to the National Park Service website page dedicated to the specific park (example: <https://www.nps.gov/zion/index.htm>).

Possible feature enhancements, if time and skill level allows (in order of importance):

* Enable users to upload photos as part of journal entries, since photos would really enhance the capturing of their experiences and memories at the national parks they have visited.
* Present users with additional information when they indicate they want to learn about various national parks, as a user would likely expect more information beyond a description, cost, and an external link.
* Display images of national parks when users learn about various national parks in the US.
* Allow users to track which national parks they have visited. Maybe this is a visual chart or a percentage calculated and presented to the user within the interface.

**Task Vignettes**

Task 1: Journal about a visit to a national park

Blake loves the outdoors and frequently visits US national parks. At the conclusion of his visits to national parks, he often jots down memories from his experience on paper, which he stores in a binder at home to be reviewed later if he wishes to reminisce on his experience. Recently, Blake wanted to relive his experience at Zion National Park, but he misplaced his record and couldn’t locate the file. Blake decides he needs to stop being so old-school and start digitally recording his memories in one place. Blake does some research and stumbles upon my GUI application, which he decides to use. The application is launched and Blake begins by selecting the option to create a new journal entry. He is then able to select a US national park from a dropdown menu (example: Zion National Park, UT), type out details of his experience in a textbox (example: “Best national park I’ve visited. Hiking Angels Landing was gnarly.”), and save and store his entry.

Details:

* User launches application
* Application split between two tabs (Journal and Learn) but opens on the Journal tab by default
* User presses a UI element (button) to “Create a new journal entry”
* An updated interface is displayed to the user where they can identify a national park and jot down details of their experience
  + Possible instruction prompt: “To complete a new journal entry and log the details of your national park experience, select the appropriate national park from the dropdown menu and then record your memories in the text field. When finished, save your entry by clicking the ‘save’ button.”
* User selects the appropriate national park from the dropdown menu
* User navigates to the text entry field, where they type in the desired details of their experience
* [Task 2 completed - see below]
* User navigates to the UI element (button) that reads “Save” at the bottom of the interface
* Journal tab updates with Blake’s new entry displayed for him to see and view later, if desired

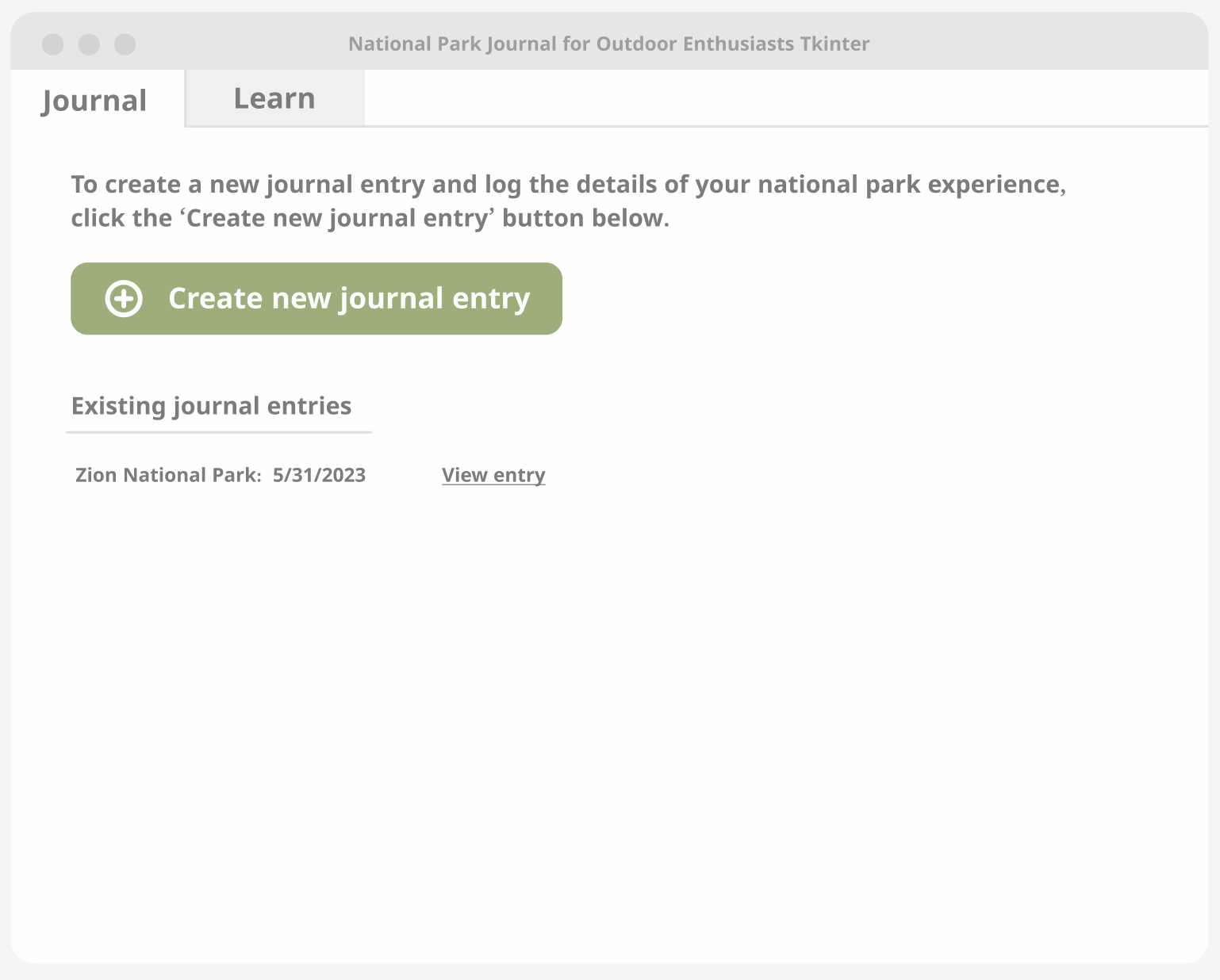
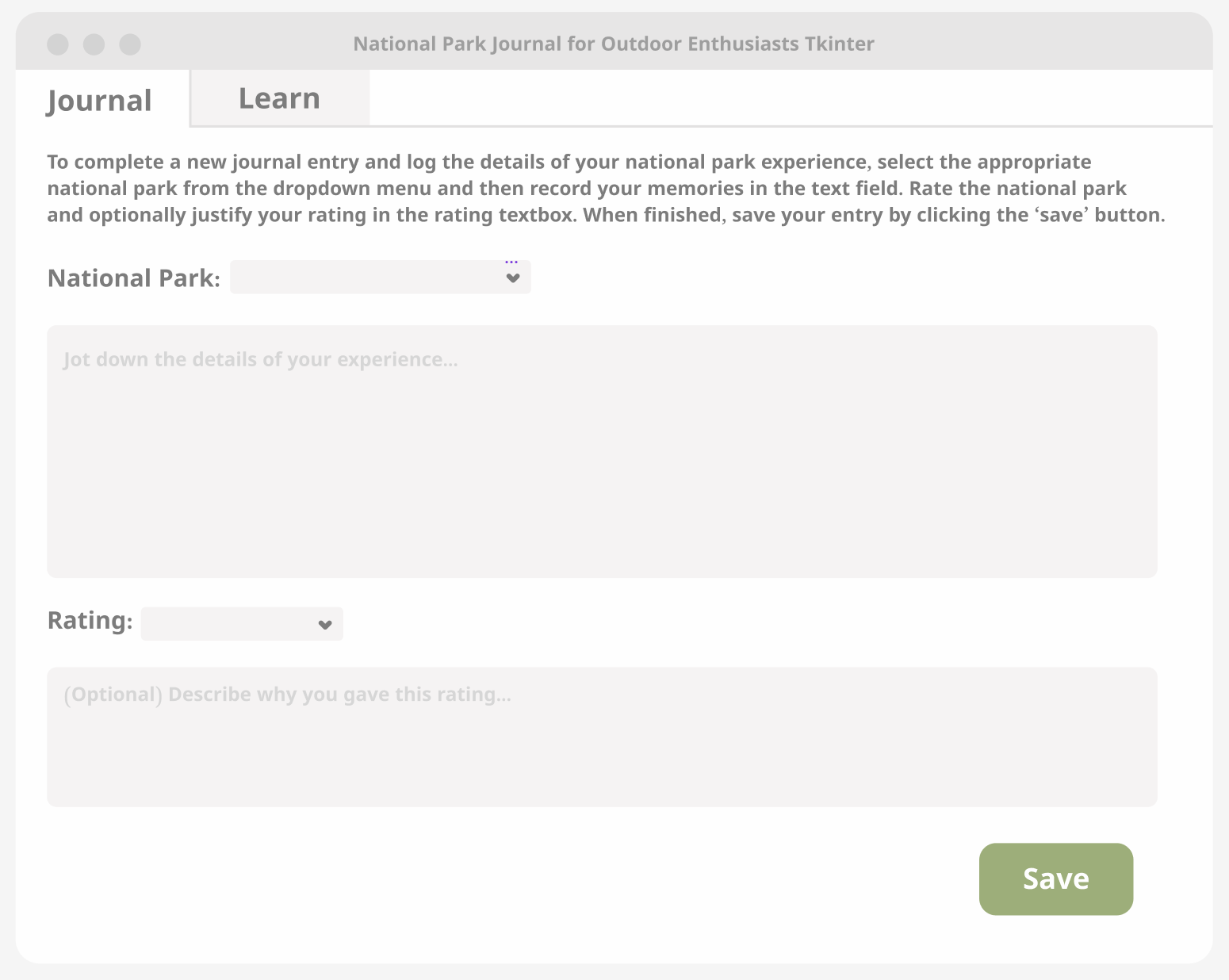
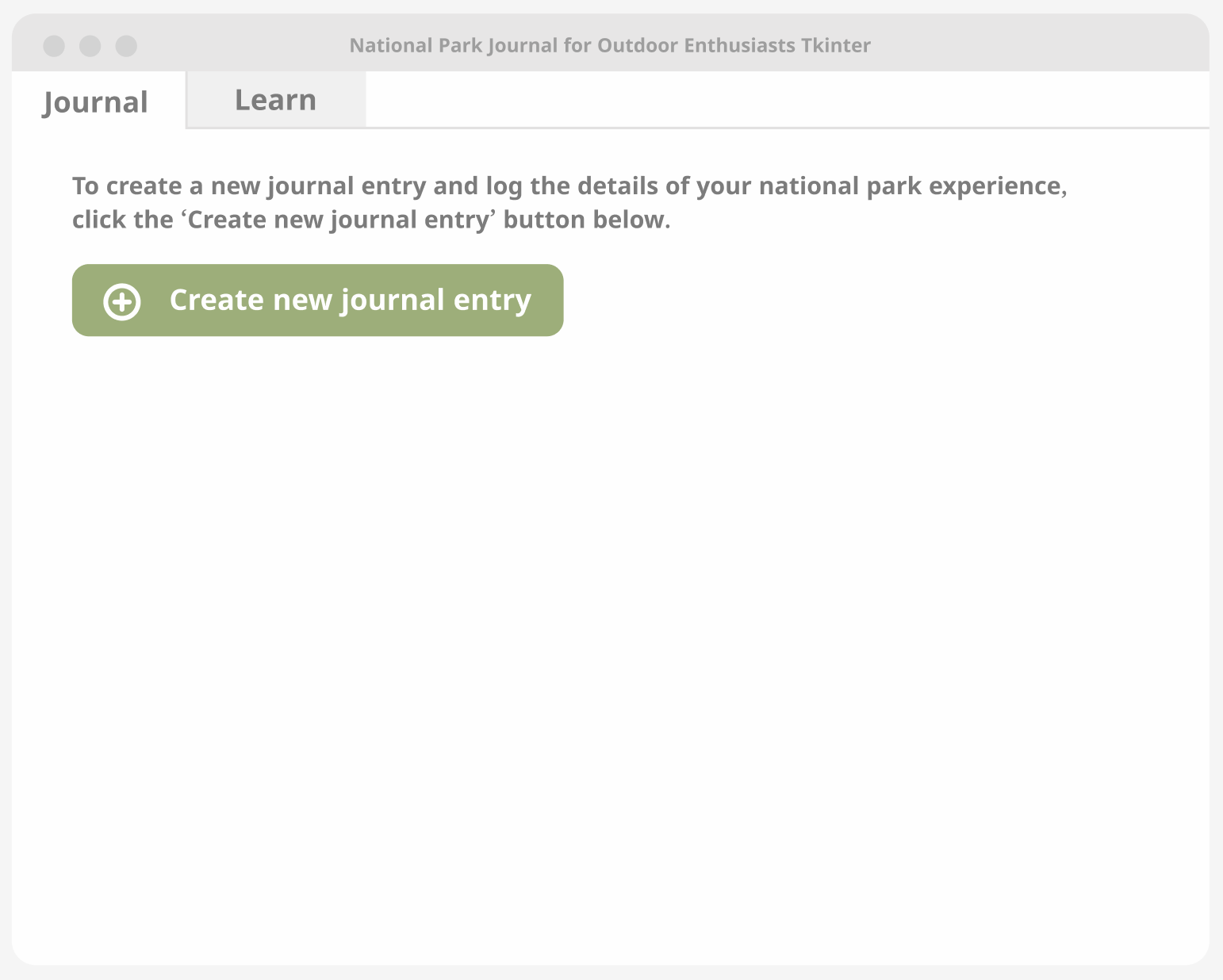
Task 2: Rate the national park visited

Blake wants to remember which national parks he enjoyed visiting the most and, similarly, which national parks he thought didn’t live up to their hype. While logging the details of his experiences at a national park, Blake has the option to rate the national park on a scale from 0-5 and justify his rating (example: 5, “Zion is the best national park in the US”).

Details:

* After the user records their memories in the primary textbox, the user has the option to rate the selected national park on a scale from 0-5, perhaps in the form of another dropdown menu or sliding bar (to be determined later)
  + Possible instruction prompt: “Rate your experience at this national park by selecting a number from 0-5 from the dropdown menu. Optionally, justify your rating by describing it in the rating textbox.”
* User selects a rating from the dropdown menu
* User justifies the rating by inputting a rating text description in the rating text field

Task 1 + 2 Mockup:

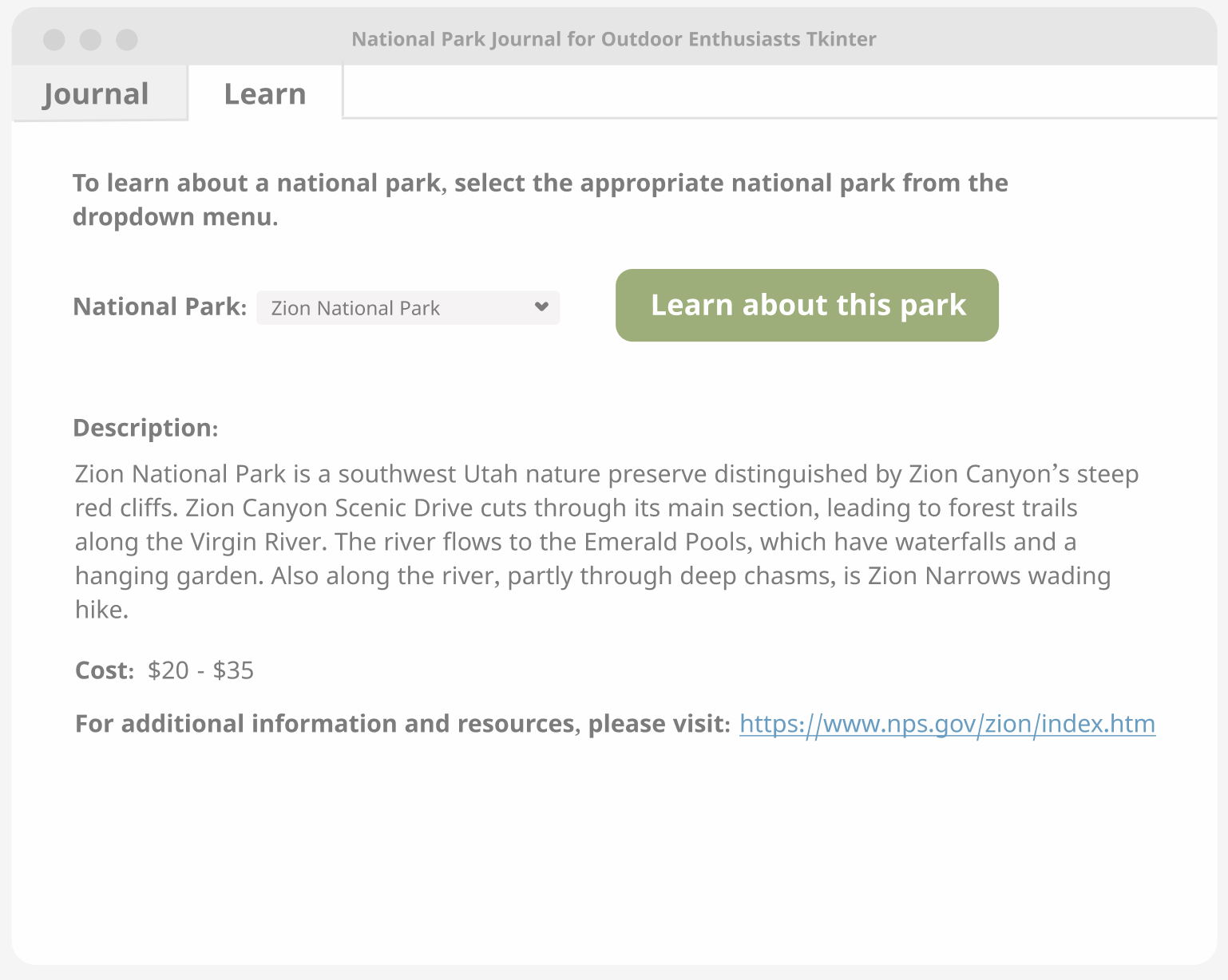
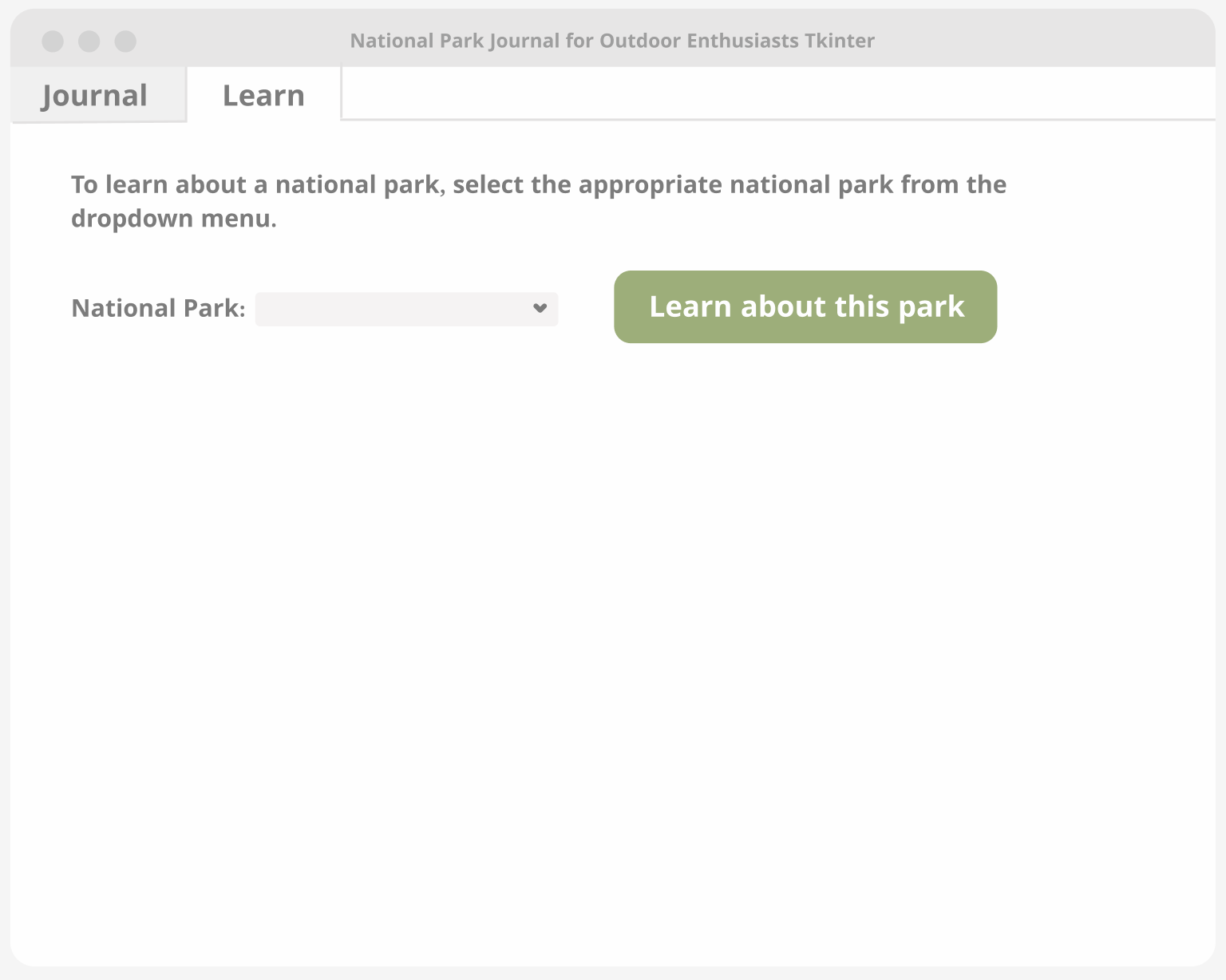


Task 3: Learn about a national park to visit in the future

Blake hasn’t visited all the national parks in the US, but plans to before the end of his days. He wants to learn about other national parks in the US he hasn’t visited yet, so he navigates to the Learn tab of the application, where he is able to search for a national park he wishes to visit by selecting the park from a dropdown menu. For example, Blake selects “Zion National Park, UT” from the dropdown menu. Blake then clicks a “Learn about this park” button and the interface then updates with information about, for example, Zion National Park, including a park description, cost of entry, and a link to the National Park Service web page dedicated to Zion National Park.

Details:

* User navigates to Learn tab of application
* Application loads interface with a dropdown menu of US national parks
  + Possible instruction prompt: “To learn about a national park, select the appropriate national park from the dropdown menu.”
  + Possible feature enhancement (if time and skill level allows): option to filter first by state?
* User selects a national park from the dropdown menu and clicks the “Learn about this park” button
* Application interface updates with a description of the park and cost of entry
* Application interface will also present the user with a link to the National Park Service web page dedicated to the selected national park
  + Example: “For additional information and resources, please visit: <https://www.nps.gov/zion/index.htm>”

Task 3 Mockup:

**Technical Flow**

For this project GUI programmed using Tkinter, the necessary data should be stored in a CSV file, or alternatively in the form of nested lists or dictionaries. The data is to include a list of US national parks, their location by state, description, costs, and a link to the National Park Service website page dedicated to the specific park (example: <https://www.nps.gov/zion/index.htm>). To work with this data and enable it to fit the needs of this project and manage the data inputted by users, the pickle module may also be utilized, per the recommendation of Professor Harding. UI elements will include, at a minimum, a master window, frames, tabs, labels, text entry fields, buttons, and dropdown menus.

Core functions + the rough flow in which they will be needed:

* Class App(Frame):
  + Frame for all widgets below
* def \_\_init\_\_(self, master):
  + This will be the primary window to host the application’s two tabs: Journal and Learn
    - Not sure yet how to split this main window into two tabs to allow for tabular navigation between Journal and Learn functionalities, but per a quick google search and browsing the course resources page, it seems doable
  + The app will launch with the Journal tab displayed by default
  + This function will also host the “Create new journal entry” button (Journal tab), with the command registered with a left button click callback to the enter\_new\_journal\_entry function below
  + Text widgets or labels will also be included to provide the user with guidance on entering a new journal entry (Journal tab) and how to learn about various national parks (Learn tab)
* def enter\_new\_journal\_entry():
  + When the “Create new journal entry” button is clicked by the user on the Journal tab, this function is called
  + The interface will update with the dropdown menu list of US national parks and a text entry field for the user to input details about their visit
  + The national park rating dropdown menu or slider, as well as the optional rating text entry field, will also be displayed to the user at this stage
  + The “Save” button will need to be incorporated here as well, processing the data saving method and prompting the system to store the user’s inputted data, which ideally will be stored and can be accessed later
    - Not sure yet how to set up this functionality, but perhaps the Journal tab of the primary window will update with a list of existing journal entries once entries are actually inputted and saved OR the entries are saved a .txt file on the users computer
  + The interface will also display instructions (in the form of labels) to prompt the user to select the national park, input details, rate the park, justify their rating, and save their entry
* def get\_natpark\_data():
  + I envision this function possibly being needed to assemble the dropdown menu containing the list of US national parks a user chooses from when entering a new journal entry
  + Maybe there is a way to wrap this into the enter\_new\_journal\_entry function above
  + Functionality would include grabbing the data from the CSV file or list/dictionary containing the national park data
* def new\_learn\_inquiry():
  + When the user navigates to the Learn tab of the primary window, this function is called and the interface displays to the user another dropdown menu of US national parks, allowing the user to select one from the list
  + This function could also host the “Learn about this park” button, with the command registered with a left button click callback to the get\_natpark\_learn\_data function below
  + The interface will also display instructions (in the form of a label) to prompt the user to select a park and click the “Learn about this park” button in order to learn about the appropriate park
* def get\_natpark\_learn\_data():
  + When the “Learn about this park” button is clicked on the Learn tab after selecting a national park from the dropdown list, this function is called
  + The interface’s Learn tab will update and display a description of the selected national park, the cost of entry details, and a link to the National Park Service park-specific web page for the user to review

**Self Assessment**

What was the biggest, most unexpected change you had to make from your sketch?

* After thinking through this GUI application in more detail, I realized that it would be necessary to incorporate tabular navigation as part of the interface, to provide users a (hopefully) smooth way to transition between the two areas of functionality within the GUI: journaling about visits to national parks and learning about various national parks.

How confident do you feel that you can implement the spec as it's written right now?

* I’m fairly new to programming (I’m feeling very thankful I took HCI 574), so after working through the details provided in this project spec, I am still a bit nervous about implementing everything, but I do think it is doable and will just require some necessary learning on my part! I think there will certainly be some frustration experienced as I learn how to program everything I need, but I am comforted by the fact that I will have visual aids along the way and the outcome will be really rewarding! I think between what I know right now, the plethora of information available online, and being able to obtain help from Professor Harding, I should be able to achieve the details outlined in this spec document.

What is the biggest potential problem that you NEED to solve (or you’ll fail)?

* I am having a hard time thinking through how I would store user inputs (journal text entries and rating justification text entries) and how this inputted data could be accessed later within the GUI application. A simpler idea would be to store these entries as text (.txt) files on the user’s computer, but I think it would be neat to be able to access existing journal entries from within the GUI application. Professor Harding recommended the pickle module, which I plan to research more and utilize as needed.

What parts are you least familiar with and might need Professor Harding’s help?

* Some other planned functionalities I also feel pretty unfamiliar with include implementing tabular navigation and setting up some of the UI elements (dropdown menus and sliders), but I think I will be able to figure these out. I will probably need some help connecting all the pieces/functions together, so everything operates smoothly. I think this entire project is going to be a major learning experience for me and while I probably will need help from Professor Harding, I plan on doing research on my own in an effort to implement everything I can before reaching out for help.