Software Requirements Specification

for

SaveCation

Revision 1.0 approved

Prepared by Group 5 (James, Savanna, Thomas, Trevor)

Morehead State University

CS 380-001

29 April 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Repo change	4/29/25	AI interfacing delayed, and web scraping methods changed.	Rev 1.0

1. Introduction

1.1 Purpose

This document is intended to describe the function, design, and development of the Budget Vacation Planner Rev. 1.0 (henceforth referred to as SaveCation) to be created by Group 5 for the Software Engineering (CS 380) class at Morehead State University. This document will be a guide to the group members and instructor on the requirements for input and output on the individual components of this group's project including the user interface, web scraping, and search algorithm. Additionally, it will outline the potential expanded functionality of the application.

1.2 Document Conventions

Priorities for higher level requirements are assumed to be inherited by detailed requirements unless otherwise mentioned. However, there are levels to the priorities, which we have split into high, medium, and low. The priority for each functional requirement will be specified. Should any term be especially important, it will be typed in **bold**. Names that are subject to change will be written in red text.

1.3 Intended Audience and Reading Suggestions

Since this document is designed for our Software Engineering class, each reader may have a different reason to analyze this document. It is recommended that every class of reader view section 1 and 2 before moving to other parts of this document, as these sections provide an important overview of the system as a whole. Since this project is relatively small, we have included potential additional functionalities in this document. However, developers part of this group are not encouraged to read deeply into the requirements marked as medium or low priority as these may be developed later. A finished product will be delivered before resources will be spent developing non-essential functions.

1.4 Product Scope

The vacation planner outlined in this document (SaveCation) is intended to help average people plan for a vacation without the hassle of finding a place to stay, activities to do, things to eat, and attractions to visit, and does so by providing a single application that suggests a complete itinerary. Creating this software has corporate potential as it would be an invaluable website for different establishments to advertise on. On a simplified level, the user should input a budget and be shown a list of possible places to stay near their destination as well as possible entertainment opportunities nearby. Additional functionality will be built on and yet more could be added.

1.5 References

Use case diagram, Gantt chart, development graph, and system overview available in appendix.

2. Overall Description

2.1 Product Perspective

SaveCation will be built from scratch without the use of preexisting components but will utilize the aid of existing databases through web scraping. The website skiplagged.com will be used to gather information regarding airline ticket prices, and vacasa.com will provide information regarding housing expenses. Additionally, eventbrite.com will provide suggestions for events and corresponding ticket prices to be factored into the estimated vacation cost. The following graphic will illustrate a simplified version of the systems:

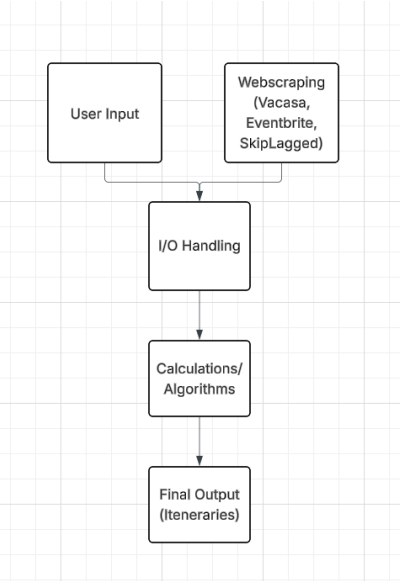


Figure I: System Overview

2.2 Product Functions

The primary operations that SaveCation will execute will be useful for determining **high quality suggestions** for destinations and activities within user-specified parameters. Such operations will be:

1) Accept user input specifying:
   a) Budget
   b) U.S. State
   c) Dates of Departure/Return
   d) Number of travelers
   e) If a car will be rented
   f) Departure location
2) Scrape the web for data involving relevant costs
3) Implement appropriate algorithms to perform calculations with scraped/input data
4) Provide suggestions for a vacation itinerary that satisfy all user-specified parameters (see operation #1)

2.3 User Classes and Characteristics

SaveCation will almost exclusively be used by individuals looking to plan their next vacation. It is possible that others may use it, for example businesses that want to see if they will be suggested by the system, or perhaps parties interested in advertising their services within the web app, but only the needs of those using SaveCation as a tool for planning vacations will be considered in the design process. Therefore, this system will be designed strictly as a tool, and only features that satisfy the requirements for which this tool will be created will be designed. As this is the case, all users will be assumed to possess the same level of technical expertise, which will be assumed to be low, and all will be granted the same privileges and access to all functions.

2.4 Operating Environment

This product is intended to operate as a web application, and therefore must be **compatible with the most common modern browsers**, including: Firefox, Opera/Opera GX, Google Chrome, Microsoft Edge, and Safari.

2.5 Design and Implementation Constraints

This product will be programmed in Python, and will need to be able to interface with **Vacasa, Eventbrite, and Skiplagged** through web scraping. Skiplagged will provide our program with information on airline ticket prices and dates, EventBrite on local costs and expenses to be expected in order to help provide a cost estimate for the desired vacation plan. Vacasa will provide housing information. Due to the nature of being a web application, the performance standards must meet a higher bar than a typical desktop application, due to limitations imposed by browsers' processing capabilities and network limitations.

2.6 User Documentation

SaveCation will be delivered to the general public as a web app, and not to any particular organization. This being the case, the documentation concerning the operation and use of the system will be provided in the form of an FAQ, which will be created using questions posed by potential users and developers alike. The FAQ will be presented as a web page, accessible through a hypertext link that a user in need of help may click.

2.7 Assumptions and Dependencies

The success of this product relies on our ability to interface with Vacasa, EventBrite, and Skiplagged. We assume that these resources will not require an unreasonable amount of effort to integrate, and that we will not face any legal challenges or takedown requests. Another assumption is that all of the functions we have described will be **simple** enough to implement in a web application, and will meet the performance requirements specified earlier in this document. Should any of these assumptions be incorrect, the cost of recovery would be incredibly high, but it is with a high degree of confidence that the authors of this document believe these assumptions to be sound, and find it very unlikely for these issues to arise.

3. External Interface Requirements

3.1 User Interfaces

This product will consist of a single homepage, with search results presented in an expandable/collapsible format. On the left-hand side, search functions will be available in a hamburger menu. The design is intended to be **minimalistic** and **user-friendly** to appeal to a wider range of age groups. Design decisions and feature presentation will be directed with this minimalism in mind.

3.2 Hardware Interfaces

This system will be designed to allow typical mouse-and-keyboard style input for making selections and typing strings.

3.3 Software Interfaces

The SaveCation system will be developed using Streamlit, a Python-based framework that allows for rapid development of interactive web applications. This application will be able to run standard web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, which will allow accessibility across desktop and mobile. Streamlit will handle both the user interface and the backend logic, allowing the user to input their destination and budget, view recommended attractions, dining options, and receive a complete estimated trip cost. In addition, this system will interface with SkipLagged to gather flight date and cost information and EventBrite/Vacasa to gather information regarding travel costs specific to the suggested areas, both of which will be used to calculate an estimated trip cost, an internal database containing dates of nationally recognized holidays in order to warn the user of potentially high levels of traffic during their trip, and browser softwares, specified in section 2.4 of this document (to allow simple distribution and execution of the system).

3.4 Communications Interfaces

In compliance with modern security standards, HTTP will be employed to ensure a secure connection to the host site. A HTTPS certification may be pursued in the future. Since this product will be designed as a web app, it will be hosted by a personal server. The system must be able to interface with EventBrite, Vacasa, and SkipLagged through a secure network connection. The primary issue of the web app implementation is that, at the time of writing, only temporary networking measures are available. However, migrating the server to a more permanent solution in the future would take little effort.

4. System Features

4.1 Housing, Transportation, and Entertainment Cost and Location Data Collection

4.1.1 Description and Priority

This product will collect data regarding the cost of potential activities and travel costs, as well as information on the location of said activities and costs. Since this feature is the most appealing and useful part of this product, its importance is of the highest degree, and will make up the bulk of the effort required in implementation. In particular, information regarding housing, local options for activities, flight costs and dates, and expected gas expenses will be necessary.

4.1.2 Stimulus/Response Sequences

The User will complete a search field with a filter, and provide information concerning the User's budget, optional destination parameters, trip start and end dates, number of travelers, departure location, and whether a car will be rented. The system will then utilize this information to provide several suggestions on potential vacation itineraries that satisfy all user-specified parameters. Functions that gather necessary information regarding costs and locations, as well as functions that can calculate necessary values with that information will be implemented.

4.1.3 Functional Requirements

This feature will use a combination of web scraping and AI for data collection, so input verification is only necessary for the search parameters. In regard to this feature, it must be verified that the information is of the correct data type, specified for each case, and within a reasonable set of values. Each expense should cost less than $10,000 dollars per person per day, and at least $0.

COSTD-1: Ticket cost and location data is gathered through web scraping
COSTD-2: Localized entertainment and gas price estimates collected through Web Scraping
COSTD-3: Combines user search parameters with collected data to determine suitable options for later presentation in an itinerary

4.2 Natural Disaster and Holiday Alert

4.2.1 Description and Priority

This feature will present the user with a pop-up window notifying the user about potential weather conditions or holidays that may lead to high volumes of traffic during the trip. Since this system is designed to suggest trips to its users, this feature will be considered high priority. The ease of implementation was also considered, as it will have a relatively low cost.

4.2.2 Stimulus/Response Sequences

Whenever a user performs a search, a pop-up window will provide a disclaimer stating that when evaluating the system's suggestions, the user should consider factors such as weather advisories and natural disasters, and that the system does not consider these factors when generating suggestions. Additionally, if the user's vacation date falls within a week of a nationally recognized holiday, red text will be displayed alongside the search results that this

holiday is close in date relative to the trip and may result in higher levels of traffic than usual, and that the cost estimate may be higher than usual due to increased housing costs. The user will be presented with a checkbox that allows the user to prevent the weather popup from returning indefinitely.

4.2.3 Functional Requirements

This feature will rely on an internal database of holidays that will be compared to the start and end date of the user's trip to determine when the user is planning a trip within 1 week of a known holiday. Additionally, the system must store a value to check if the user has selected the option to hide the pop-up.

NOTI-1: Store and access internal database of nationally-recognized holidays
NOTI-2: Provide user option to hide weather popup

4.3 Car Rental, Taxi Service, and Uber Fare Cost Calculations

4.3.1 Description and Priority

This feature will be dedicated to estimating the cost of transportation during the trip, excluding the cost to reach the destination and cost to return home. More specifically, the cost to travel between activities, services, and housing during the trip. We assume for this that the user will rely on car rental (including gas), taxi, or Uber (or similar service such as Lyft). In future versions, the scope may be expanded to include nautical forms of transport such as ferries. This feature is of low priority, and will be added only once all higher priority features are complete.

4.3.2 Stimulus/Response Sequences

This feature's stimuli and response sequences are identical to those specified in 4.1.2, the additional functionality simply including these intermediate transport costs in the final estimate, improving accuracy. This feature is considered to be of medium priority because the developers of SaveCation are confident the accuracy requirements specified in section 5.4 will be met without its inclusion.

4.3.3 Functional Requirements

Future Interfacing with an LLM is necessary to maintain up-to-date estimates on fares and car rental prices.

FARE-1: AI receives input and provides suitable output regarding fares and car rental prices for use in system calculations.

4.4 Proximity-Based Calculations For Activities/Services/Transportation

4.4.1 Description and Priority

Proximity-based calculations could be included to prioritize similarly priced housing options that are located closer to the suggested activities and optional services that the user may find of interest. This feature would also make it possible for a user to personalize the desired ratio of cost-to-convenience. If multiple housing options for a particular itinerary are less than $5 apart in total cost per night (considering number of travelers), the option with the shortest average distance from suggested activities and services will be selected for the system's suggestion on that particular itinerary. This feature is considered to be of medium priority, as it is desirable, but not necessary for a functional product that satisfies the requirements outlined in this document.

4.4.2 Stimulus/Response Sequences

When the system receives the user inputs specified in section 4.1.2, the system will compare similarly priced housing options based on average distance from the suggested activities and services generated by the system, and select the closest option to suggest.

4.4.3 Functional Requirements

Cost of each housing option in the suggested area must be stored and compared, so that when options are similarly priced, only more convenient options are presented.

DIST-1: Store and compare price and proximity data for each housing option
DIST-2: Eliminate less convenient yet similarly priced options

4.5 Budget-Based Calculations For Local Services

4.5.1 Description and Priority

Some users may require certain services to be available during a trip. For example, some may need access to childcare, and some may need or prefer to be near health services. Since proximity to these particular services would be added to the search parameters in these cases, the cost of housing and transportation would likely be impacted, as many services are only available closer to the more developed sections of cities. This feature would introduce the option to include these parameters in the system's search and final estimation. This feature is considered to be of low priority, since the vast majority of users are not expected to utilize it.

4.5.2 Stimulus/Response Sequences

During the user's entry of the search parameters, they would be provided with the option to specify mandatory services from a predefined list. Should the user select one or more of these options, any housing options exceeding a distance of 5 miles from any selected service will be discarded. Should no housing options fulfill all parameters, the user will be notified and asked to adjust their parameters.

4.5.3 Functional Requirements

Local services that fall under the categories of childcare, health services, grocery stores, and shopping centers will be located based on user-specified requirements and only housing options that are within 5 miles of the specified services will be presented.

SERV-1: Services must be identified and categorized utilizing web scraping
SERV-2: Services proximity to available housing must be determined, unsuitable options eliminated
SERV-3: User must be presented optional services to select from

4.6 Cost-to-Convenience Slider

4.6.1 Description and Priority

A slider may be implemented to allow the user to have a simple way to visually demonstrate the importance of the convenience of housing options (proximity to suggested activities and services) as opposed to the affordability of them. This is considered to be of low priority because it is a convenience feature, and relies on the functionality of another optional feature (4.4), acting as an extension of it.

4.6.2 Stimulus/Response Sequences

The user will drag a point across a line (as a slider) that will indicate greater convenience on one end, and greater affordability on another end. The point will rest in one of 5 preset spots along the line, in increments of 25% (0-100). At the extreme end of convenience (100%) of the housing options within budget, only the location with the lowest average distance to suggested activities/services will be considered in the output. At the extreme end of affordability (0%), the cheapest option will be selected, ignoring proximity. For calculating the intermediate options, each housing option will be organized from lowest to highest cost and assigned a value for its average distance from suggested activities/services. Any option that has a higher value for distance than any of the options preceding it will be eliminated (higher cost and higher distance makes it an unsuitable choice). Then, the option that correlates closest to the user's preference (25th, 50th, or 75th percentile determined by position) will be selected. In the result that two options are suitable (even number of options), the cheaper option will be selected.

4.6.3 Functional Requirements

An algorithm will be implemented to relate the user's relative selection to a quantity that represents their choice accurately. The user provides input that is controlled and error is minimized by implementing extreme simplicity in presentation, that input is quantified by an algorithm, and suitable housing options are selected based on user input.

SLID-1: User adjusts affordability-to-convenience slider to preference
SLID-2: User selection is quantified and compared to housing options
SLID-3: Suitable options are selected and used in final output

4.7 Output: Suggested Trip Itineraries

4.7.1 Description and Priority

This feature represents the primary function that makes the product appealing to the user, the combination of several features to provide an output that satisfies all user-defined requirements and all developer determined standards that will provide to the user multiple suggestions regarding locations and activities near said locations to aid in the selection of a user-planned trip. This is considered to be of the highest degree of importance as it solves the very problem the system was designed to address, which is a user that desires ideas for trips that are tailored to the user’s needs.

4.7.2 Stimulus/Response Sequences

After inputting all required values to perform a search, the user will be offered multiple itineraries with suggestions for locations and activities within the user-selected state that satisfies the user-specified budget and all optional parameters.

4.7.3 Functional Requirements

Functions for acquiring user inputs, externally collected data (web scraping and AI), acquiring and presenting images of suggested areas, and algorithms for processing data and performing calculations.

OUT-1: User input is acquired for required fields and optional fields or default values assigned for optional fields (Inputs include: budget, desired state, number of people, if a car will be rented, and dates of travel).
OUT-2: External data is collected (typical cost of housing, entertainment, and travel, all specific to the area)
OUT-3: Images of suggested destinations are acquired and presented alongside them utilizing web scraping.
OUT-4: Store cost and location data is used to determine output through calculations

5. Other Nonfunctional Requirements

5.1 Performance Requirements

As stated in section 2.5, since the system will be implemented as a web app, performance speed is an important constraint considering browsers do not execute tasks as quickly as most native operations, and network connectivity can be a limiting factor. Considering this, a user search should take no longer than 5 seconds on average to yield results. Beyond 10 seconds, a timeout should be implemented to terminate the search. Additionally, the ping time between the system's host and the informational resources (Vacasa, EventBrite, SkipLagged) should not exceed 150 ms for a period of time longer than 3 seconds.

5.2 Safety Requirements

In future versions of SaveCation, everytime the system is used, a notice will be provided regarding potential inclement weather and natural disasters. The user will be reminded that it is their responsibility to ensure safe travel conditions before departing, and that SaveCation is not able to account for unsafe conditions in real-time, and it is possible that an unsafe destination may be recommended, for example, a trip to a Florida beach during a hurricane would be undesirable, but it is possible that the system will suggest such a trip since detecting hurricanes is beyond its boundaries. The user will also be informed that the system does not select locations based on perceived neighborhood safety or crime rates, and only factors in cost and proximity.

5.3 Security Requirements

SaveCation will be available to the public, and a system for managing user accounts will not be designed. To protect user privacy and data integrity, no information related to identity will be requested– nor will personal information.

5.4 Software Quality Attributes

Above all other qualities, the system must be correct. A margin of error no greater than 2% for underestimates, and no greater than 20% for overestimates will be tolerated. The first priority of the developers regarding cost estimates is that of preventing overspending, or underallocation of budget. In the opinion of the authors of this document, few outcomes are as undesirable as a user bringing an insufficient budget on a trip suggested by SaveCation. For this reason, almost no leniency has been granted towards underestimation, as the suggested budget should err on the side of caution, as overpreparation is preferable. The system should also be relatively easy to use, as the typical user is expected to be, on average, thirty years of age or older. It should take the average user no longer than five minutes to generate their first results. Additionally, as a web application, an acceptable degree of compatibility must be achieved to ensure functionality with all browsers listed at the beginning of this document.

5.5 Business Rules

All functions will be available to all users. SaveCation will be used for the single-minded purpose of suggesting suitable travel itineraries for users, and therefore no functionality for things such as

moderation will be required as there are no social elements, and no functionality for limited privileges will be necessary due to the lack of transmission of sensitive information. This product will not be used by any particular centralized organization, and therefore all functionality may be made public.

6. Other Requirements

An internal database of national holidays must be generated for this system.

Appendix A: Analysis Models

Figure 1: System Overview

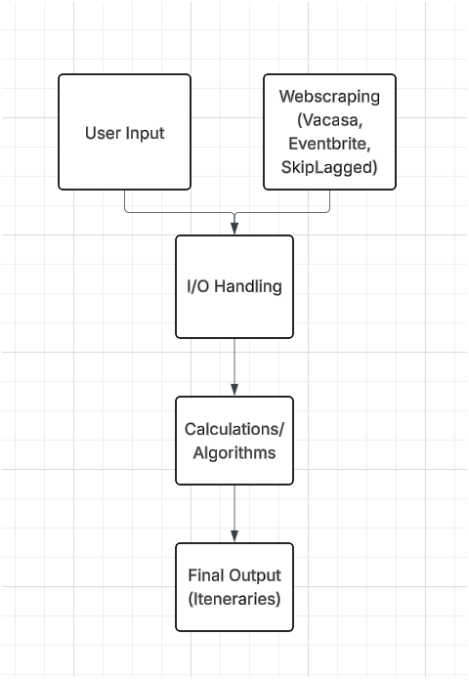


Figure 2: Use Case Diagram

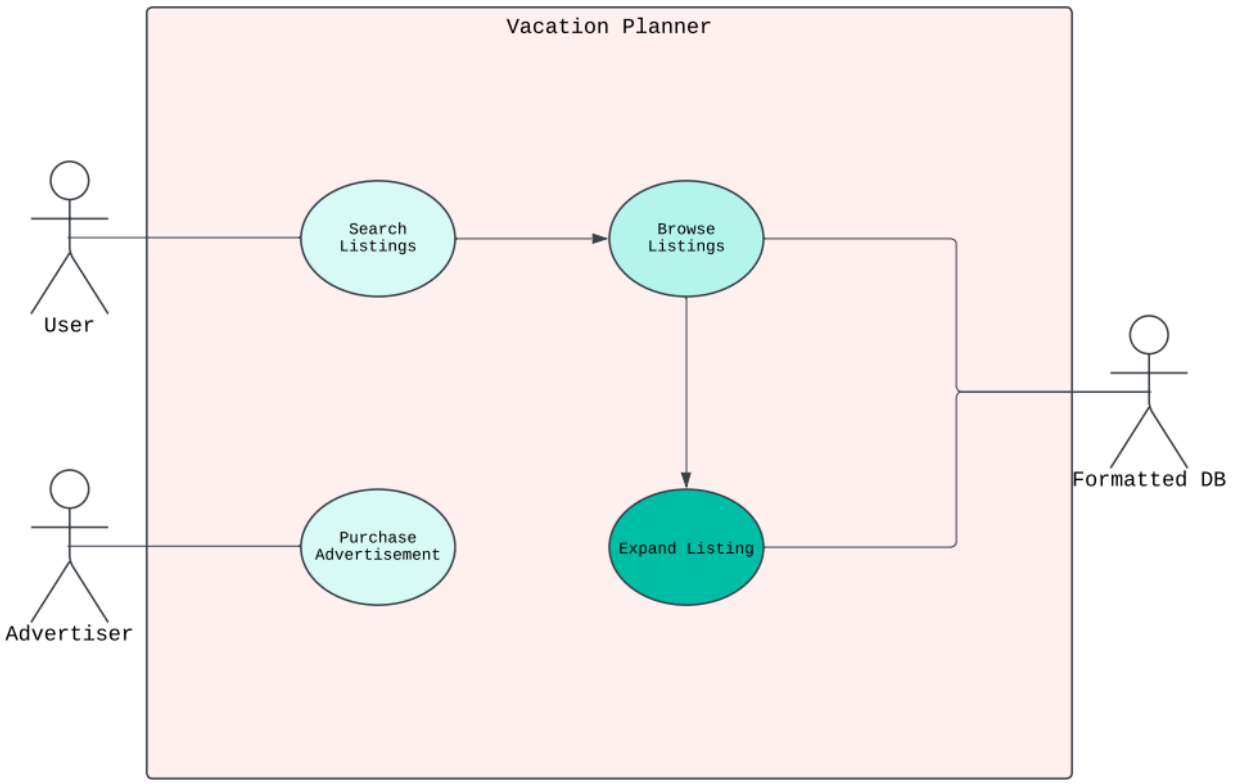


Figure 3: Gantt Chart

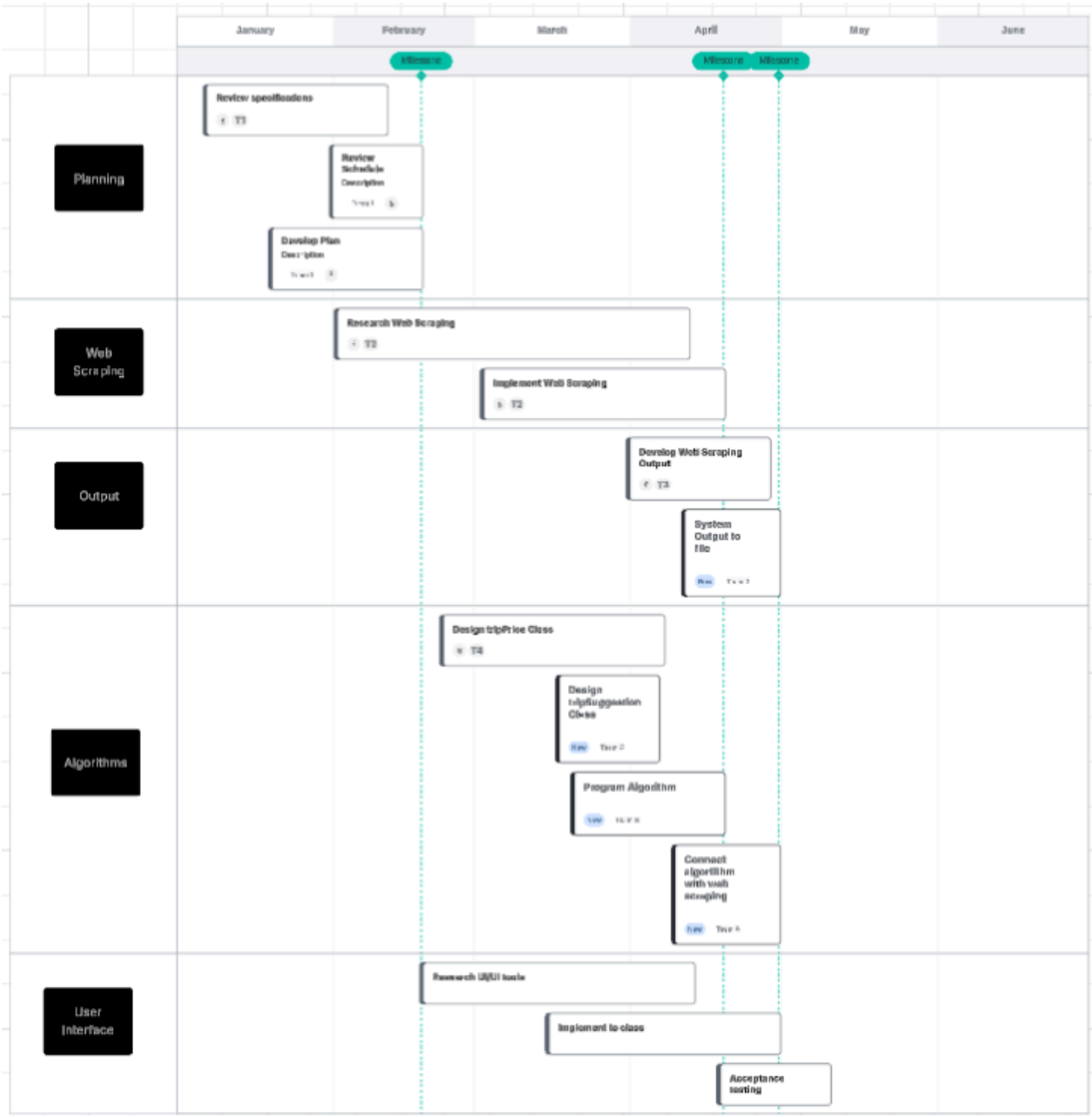
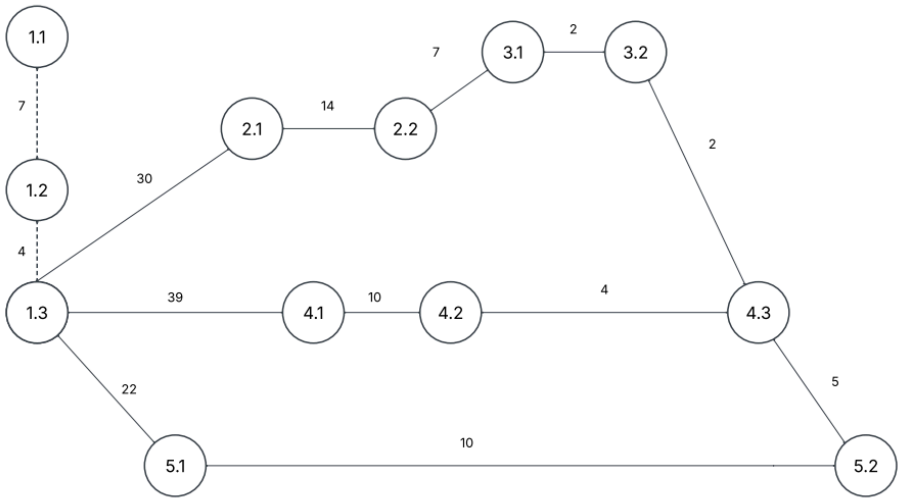


Figure 4: Development Graph



1.1 Review Specifications
1.2 Review Schedule
1.3 Develop Implementation Plan
2.1 Web Scraping Research
2.2 Web Scraping Prototype
3.1 AI-collected Web Scraping Output
3.2 Formatted and Finalized Web Scraping Output
4.1 Design tripSuggestion and tripPrice class
4.2 Program Algorithm
4.3 Connect Algorithm and Web Scraping
5.1 Research UI
5.2 Implement UI

Appendix B: To Be Determined List

No TBD’s at this time.