

Camp Scout

by

John Adkerson

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology/Cybersecurity

© Copyright 2022 Insert Your Last Names Here

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

<u>John Adkerson</u>	<u>11/29/2021</u>
Insert Your Name Here	Date
<u>Vineela Kunapareddi</u>	<u>11/29/2021</u>
Insert Faculty Advisor's Name Here, Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2022

Table of Contents

Abstract	3
Introduction	4
<i>Project Summary:.....</i>	<i>4</i>
<i>Problem Statement:.....</i>	<i>4</i>
<i>Solution:</i>	<i>4</i>
<i>Project Source:</i>	<i>6</i>
Discussion	7
<i>Project Objectives/Goals:</i>	<i>7</i>
<i>Project Scope:</i>	<i>7</i>
<i>Quick Project Timeline:</i>	<i>7</i>
<i>Technologies Used:.....</i>	<i>8</i>
<i>Technical Architecture Diagram:.....</i>	<i>11</i>
<i>User Personas:</i>	<i>13</i>
<i>Use Cases:</i>	<i>13</i>
<i>Use Case Diagram:</i>	<i>19</i>
Testing Plan:	19
Overview	Error! Bookmark not defined.
Methodology.....	Error! Bookmark not defined.
Scope.....	Error! Bookmark not defined.
Objectives.....	Error! Bookmark not defined.
Test Logs and Procedures	Error! Bookmark not defined.
Testing Review	Error! Bookmark not defined.
<i>Change Management Plan:.....</i>	<i>23</i>
<i>Budget:.....</i>	<i>24</i>
<i>Problems Encountered and Analysis of Problems Solved:.....</i>	<i>25</i>
Conclusion.....	26
References.....	<i>Error! Bookmark not defined.</i>

Abstract

This project started out as a personal challenge posed by extended family members and has now turned into something fully functional and useful. Camp Scout is multifaceted platform where campers can monitor campsite availabilities, share their ideas, experiences, and interact with others. Using state of the art front end technology, a robust API library, and simple back-end code to do the leg work, Camp Scout succeeds in its ventures. With only four small pieces of information, a user can opt in to be notified via text message if their campsite becomes available. These results mean that the future of Camp Scout is a bright one, and the roadmap of scalability is always being observed. Further developments into cloud security, persistence mechanisms, and qualified hosting will be executed in the coming months, and the potential for Camp Scout to serve as a functional social media community will proceed to blossom.

Introduction

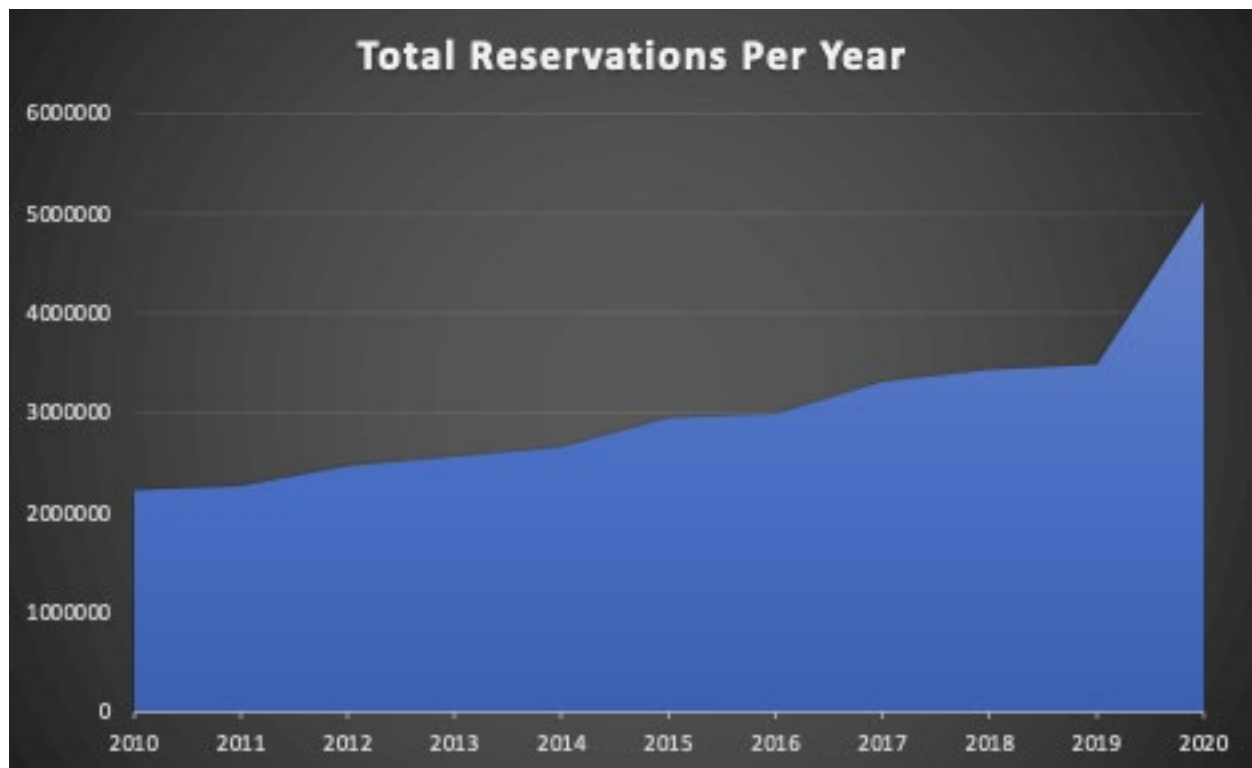
Project Summary:

Camp Scout is an application that will revolutionize campsite reservation process. Never again, will increased demand for reservations cost prospective campground goers excessive time and energy. With the rise of camping popularity emerging from the pandemic, Camp scout employs new innovative technologies to help users track and acquire their desired campsite reservations.

Problem Statement:

During the process of campsite reservation, via the government supported website [recreation.gov](https://www.recreation.gov), a peculiar bind presents itself to us. Since recent socioeconomic changes have increased the net amount of campground goers, thus demand for select campsites has followed in sympathy. Recreation.gov serves as a market bottleneck reservation portal, and the only location on the internet where U.S. national park campsite reservations can be made.

With the amount of prospective campsite reservation buyers far outnumbering the amount of available campsite reservations, it can be quite difficult to scour for a particular desired reservation. With no viable notification system in place, patrons default to refreshing their browser indefinitely, until their desired campsite reservation's availability status updates to "true". If a notification system were to be installed, it would save users lots of time and effort that they would otherwise spend refreshing recreation.gov pages.



1 Total Yearly Reservations

Depicted above is the ongoing growth rate of reservations per year as recorded by reservation.gov. Between the years of 2010 and 2019, the maximum growth rate exhibited between individual years is 11.11%. Between 2019 and 2020, when the global COVID-19 pandemic began, the growth rate increased to 46.99%. Based on this observation, we can depict that the global pandemic is driving demand for campgrounds and campsites alike.

Solution:

Camp Scout will serve as a notification sign-up system for recreation.gov users. Camp Scout will be accessed by users via a cutting-edge web portal designed with React. Users will create an account to initiate a search, and in the process implant themselves into a rich social community surrounding camping. The social media aspect of camping has yet to be pioneered in the current market and there is lots of missing campsite pictures on recreation.gov. Camp Scout aims to take an innovative approach to camping by allowing users to post photos of their campsites they attend. Once posted, users can choose to link them into a public news feed where other Camp Scout users can see their adventures. Using the built-in API that recreation.gov affords to the public, Camp Scout will accept select information from the user regarding their desired campsite information, and then hit the API endpoint with select parameters to retrieve an availability status. Once an availability status of “true” is received for a given user reservation, Camp Scout will then send a text message to the associated user’s phone number. The notification system allows users to go about their lives while waiting for their campsites to become available.

Project Source:

During yearly annual camping trips, it has become evident that booking a campsite has become a process of greater difficulty. Campsites that have been previously easy to reserve are now being swiped off the shelves as soon as they are put on sale. After witnessing the complaints of many campsite goers on public forums, it became evident that implementing a notification system would help resolve the issue.

Discussion

Project Objectives/Goals:

Below is a bulleted list of the primary, secondary, and tertiary features planned to be implemented with the Camp Scout development cycle.

- Provides a user notification signup service that will notify users of campsite availabilities
- Provides the ability to authenticate
- Provides campsite/campground bookmarking
- Provides a database of camp spots
- Provides a beautiful front end that will captivate the user
- Provides a centralized Job handling Server
- Provides cloud-secure Infrastructure

Project Scope:

Camp Scout will allow users to monitor campsite reservation availability. Providing campsite goes with an online space via which they can interface with others who share their interests, users create a unique account via which they can log in and out of as they please. After creating an account, each user is afforded a profile space in which they can view their previously favorited campsites, uploaded photos, and added friends. When creating a search for a specified campsite reservation, the front-end website will provide users with an autocomplete search box that will suggest campgrounds by querying a National Park Service API. Once the necessary search information is accepted by a user on the front end, a job is created and associated with the user's ID. The centralized job processing server then receives that job and adds it to its queue to be processed in a serialized fashion. Camp Scout will utilize Google Firebase to ensure its users

are protected and cloud security is enforced when retrieving, deleting, or updating data therein.

The front end will be built with React and will be aesthetically tuned to captivate the user.

Quick Project Timeline:

Task #	Task Name	Duration	Start Date	End Date
1	Create a splash screen wireframe	1 Week	9-27-2021	10-3-2021
2	Create a user profile page wireframe	1 Week	10-4-2021	10-10-2021
3	Enable Authentication and Sign-in via Google Firebase	1 Week	10-11-2021	10-17-2021
4	Create a job creation screen where users can start looking for campsites	1 Week	10-18-2021	10-24-2021
5	Create a news feed via which users can see other users' activity	1 Week	10-25-2021	10-31-2021
6	Create a navigation bar including an autocomplete search box to look for campgrounds/other users.	1 Week	11-1-2021	11-7-2021
7	Develop a JSON schema that will enable communication between client/server.	1 Week	11-8-2021	11-14-2021
8	Create a "user creation" REST Endpoint in the centralized server that will consume JSON.	1 Week	11-15-2021	11-21-2021
9	Create a "day checker" REST Endpoint in the centralized server that will consume "job" JSON.	1 Week	11-22-2021	11-28-2021
10	Create a UserDAO via Google Firebase and integrate previously	1 Week	11-29-2021	12-5-2021

	created “user creation” REST endpoint.			
11	Create a UserJobDAO with Google Firebase.	1 Week	12-6-2021	12-12-2021
12	Configure UserJobDAO to send/consume JSON for communication with “job queue” REST endpoint.	1 Week	12-13-2021	12-19-2021
13	Create a Job queue Implementation within the centralized server.	1 Week	12-20-2021	12-26-2021
14	Implement job queue rotation logic within centralized server.	1 Week	12-27-2021	1-2-2022
15	Create campground availability cache that “day checker” can reference periodically within the centralized server.	1 Week	1-3-2022	1-9-2022
16	Create a user context object that will store the current searching status of associated ongoing jobs.	1 Week	1-10-2022	1-16-2022
17	Create a “Search all availabilities” feature within job creation implementation.	1 Week	1-17-2022	1-23-2022
18	Create search frequency optimization logic within the centralized server.	1 Week	1-24-2022	1-30-2022
19	Create auto-delete job logic within firebase upon job completion.	1 Week	1-31-2022	2-6-2022
20	Create auto-create job logic in firebase and set to consume JSON from centralized server.	1 Week	2-7-2022	2-13-2022
21	Create auto-modify job logic in firebase and set to consume JSON from centralized server.	1 Week	2-14-2022	2-20-2022

22	Create a SMS notification REST endpoint and set to consume JSON from centralized server.	1 Week	2-21-2022	2-27-2022
23	Create a Message Event REST endpoint and set to produce JSON from centralized server.	1 Week	2-28-2022	3-6-2022
24	Create a Google Firebase MessageEventDAO to record message events. Set to consume JSON from the centralized server.	1 Week	3-7-2022	3-13-2022
25	Integrate Twilio account into Notification logic within the centralized server.	1 Week	3-14-2022	3-20-2022
26	Create SMTP notification logic and REST endpoint within the centralized server. Set to consume JSON from the centralized server.	1 Week	3-21-2022	3-27-2022
27	Integrate the front-end client website structures with the centralized server backend logic.	1 Week	3-28-2022	4-3-2022
28	Perform user testing a hone in the aesthetic design of the front end website.	1 Week	4-4-2022	4-10-2022

Technologies Used:

Below is a brief description of the capabilities that each chosen technology has to offer, along with why it was chosen.

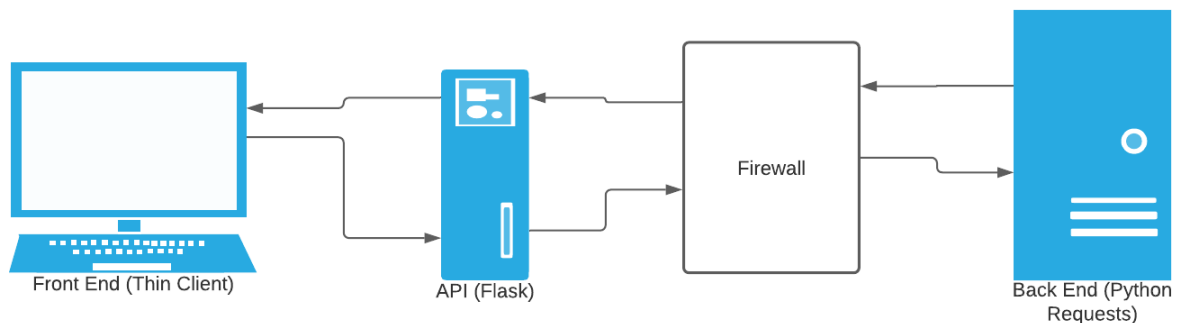
Code Stack Used:

- React
 - To build a beautiful front end
 - Controller that handles event listeners
 - Pass values to RESTful services via JSON
 - Was chosen because it offers a sleek contemporary look and is easier to integrate with Node.js than Vue.js, or Angular.
- Node.js
 - User DTO
 - CAMPSITE JOB DTO
 - Serves to house all business logic
 - Centralized job handling server
 - Was chosen because it allows for offline JavaScript code compilation which facilitates effective testing strategies
- Twilio
 - SMS messaging capabilities
 - SMTP messaging capabilities
 - Was chosen because of elegant Python integration and cheap messaging pricing at \$0.0040 per message sent.

- Python Requests
 - Initiate GET requests to recreation.gov API endpoint
 - Retrieve response JSON
 - Was chosen because it easily allows for modular parameter/header integration when calling an endpoint.
- MongoDB
 - Authentication
 - User Information
 - Search Information
 - 1 to many relationships using NOSQL (Users: one, Searches: many)

Technical Architecture Diagram:

Below is a depiction of the technical architecture providing the resources essential to Camp Scout's effective functionality. A four-component foundation can be observed. The front-end thin client is interacted with by the end user, which then communicates through our hosted API via JSON, then from the API, the JSON is redirected through the firewall. Lastly the Backend consumes the JSON schema for use. Once the Backend has finished its processing it then formats a response JSON schema and reverses the processes until the end user sees the result populate their front end.



User Personas:



Job Title

Project Manager

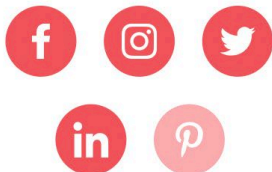
Age

25 to 34 years

Highest Level of Education

Bachelor's degree (e.g. BA, BS)

Social Networks



Industry

Technology

Organization Size

11-50 employees

The New Camper

Preferred Method of Communication

- Email

Tools They Need to Do Their Job

- Project Management
- Email

Job Responsibilities

Enter text here

Their Job Is Measured By

Team Productivity

Reports to

Director

Goals or Objectives

Leads

They Gain Information By

Conferences

Use Case 1

The Late Bloomer

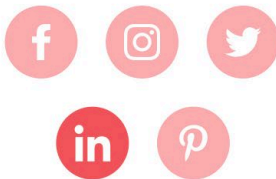


Job Title
Broker

Age
Age 65 or older

Highest Level of Education
Bachelor's degree (e.g. BA, BS)

Social Networks



Industry
Investment

Organization Size
1-10 employees

Preferred Method of Communication

- Phone
- Email
- Face-To-face

Tools They Need to Do Their Job

- Business Intelligence Dashboards

Their Job Is Measured By

Revenue Generated

Job Responsibilities

Enter text here

Goals or Objectives

Revenue

Reports to

CEO

Use Case 2

The Experienced Camper



Job Title
Student

Age
18 to 24 years

Highest Level of Education
Some college, no degree

Social Networks



Industry
Marketing

Organization Size
Self-employed

Preferred Method of Communication

- Phone
- Email
- Text Messaging
- Social Media

Tools They Need to Do Their Job

- Content Management Systems
- Cloud-Based Storage & File Sharing Applications
- Email

Job Responsibilities

Enter text here

Their Job Is Measured By

Grades

Reports to

Professors

Goals or Objectives

To earn a degree

Use Case 3

Use Cases:

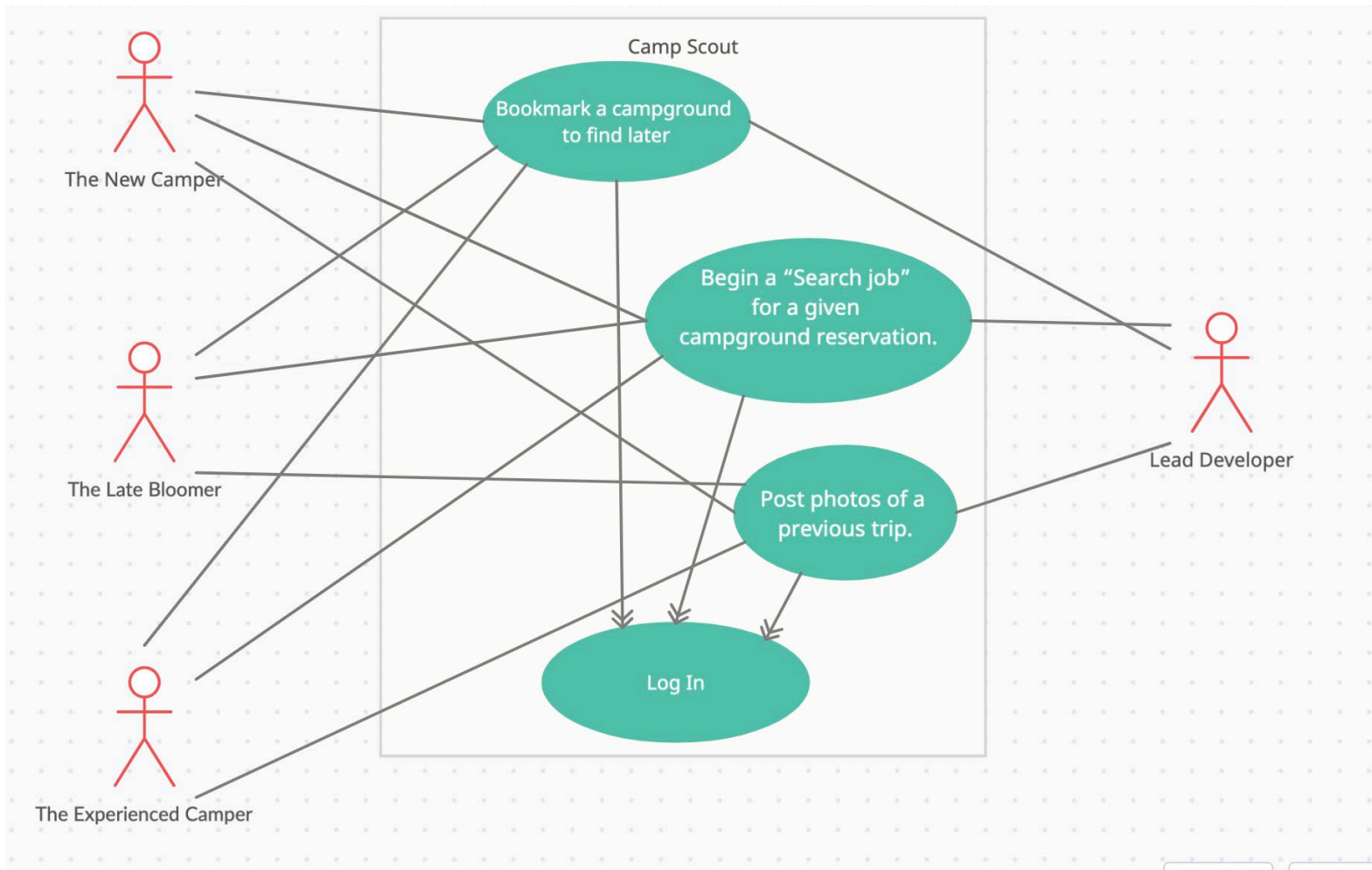
Name	Bookmark a campground to find later.
ID	CS_001
Description	While browsing through campgrounds on the website, a user finds a campground that they particularly enjoy but aren't ready to find a reservation yet. They want to save it to a list so that they can find the campground later and thus make the reservation they were originally interested in.
Actors	Camp Scout customer
Organizational Benefits	Increase sales by helping the customer remember campgrounds/reservations they were previously interested in.
Frequency of Use	20% of users save an item to be bought later each time they visit Camp Scout. 50% of saved items are purchased within one year of the saved date.
Triggers	The user selects the option to bookmark a campground.
Preconditions	User is viewing a campground in the catalog
Postconditions	The campground selected to be saved is visible to the user when they view their bookmarked campgrounds. The campground selected to be saved is reflected as a saved campground when the user views their search/browser results.
Main Course	1.) System prompts the user to confirm saving the selected campground instead of purchasing right away. 2.) User confirms to save the campground now. 3.) System determines user is not logged in and redirects user to log in to their account. 4.) User logs on. 5.) System stores the saved campground 6.) System redirects the user to their bookmarked campgrounds to view the list in full.
Alternate Courses	AC1: The system detects that the user is already logged on 1.) Return to Main Course step 5. AC2: The user logs off again 1.) Return user to Main Course step 3. AC3: User does not have an account already 1.) User creates an account. 2.) System confirms account creation. 3.) Return user to Main Course step 4.
Exceptions	EX1: User decides to purchase the item now 1.) See "Look for reservation" Use Case EX2: System fails on saving item to list 1.) System notifies user that an error has occurred. 2.) Return user to Main Course step 1.

Name	Begin a "Search job" for a given campground reservation.
ID	CS 002
Description	While browsing through campgrounds on the website, a user finds a campground that they particularly enjoy and are ready to start searching for an open availability. They want to be notified when the desired reservation becomes available.
Actors	Camp Scout customer
Organizational Benefits	Increase sales by helping the customer easily purchase reservation "Search jobs".
Frequency of Use	90% of users who visit the site make a purchase on their first visit.
Triggers	The user selects the option to start looking for a reservation availability.
Preconditions	User is viewing a campground reservation in the catalog.
Postconditions	The campground reservation selected to be purchased is visible to the user when they view their purchased "Search jobs". The campground selected to be purchased is reflected as a previously visited campground when the user views their search/browser results.
Main Course	1.) System prompts the user to confirm purchasing the selected campground reservation. 2.) User confirms to purchase the campground reservation now. 3.) System determines user is not logged in and redirects user to log in to their account. 4.) User logs on. 5.) System stores the purchased campground reservation "Search job" 6.) System redirects the user to their purchased campground reservation "Search jobs" to view the list in full.
Alternate Courses	AC1: The system detects that the user is already logged on 2.) Return to Main Course step 5. AC2: The user logs off again 2.) Return user to Main Course step 3. AC3: User does not have an account already 4.) User creates an account. 5.) System confirms account creation. 6.) Return user to Main Course step 4.
Exceptions	EX1: User decides to save the item now instead of buying 2.) See "Bookmark a campground to find later" Use Case EX2: System fails on saving item to list 3.) System notifies user that an error has occurred. 4.) Return user to Main Course step 1.

Name	Post photos of a previous trip.
ID	CS 003
Description	While browsing through campgrounds on the website, a user finds a campground that they previously visited and would like to upload photos of their experience.
Actors	Camp Scout customer
Organizational Benefits	Increase sales by helping customers integrate into a flourishing community environment where users can share their experiences with others.
Frequency of Use	70% of users who visit the site wish there was more photos available of any given campground they search for in the catalog.
Triggers	The user selects the option to upload photos.
Preconditions	User is viewing a campground reservation in the catalog.
Postconditions	The campground selected to upload photos to, is visible to the user when they view their "Photos" in their user profile page.
Main Course	1.) System prompts the user to confirm what image files they would like to upload from their computer. 2.) User confirms to upload their desired photos. 3.) System determines user is not logged in and redirects user to log in to their account. 4.) User logs on. 5.) System stores the uploaded photos and associates them with the given user profile. 6.) System redirects the user to their previously uploaded photos to view the list in full.
Alternate Courses	AC1: The system detects that the user is already logged on 3.) Return to Main Course step 5. AC2: The user logs off again 3.) Return user to Main Course step 3. AC3: User does not have an account already 7.) User creates an account. 8.) System confirms account creation. 9.) Return user to Main Course step 4.
Exceptions	EX1: User decides to keep their photos private 3.) User selects "Visible only to friends" option before their final photo upload. EX2: System fails on saving item to list 5.) System notifies user that an error has occurred. 6.) Return user to Main Course step 1.

Use Case Diagram:

Provide one use case diagram accounting for all use cases identified in the section above along with their user profile association. This can be a simple stick figure diagram. Make sure to introduce the section to the reader.



Use Case Diagram 1

Testing Plan:

Overview

During each weekly sprint within the project lifecycle, certain features will be implemented that integrate into the business logic layer of the application. These features rest upon the foundational logical analysis of given events, and the appropriate following response to said events. Whenever we encounter these situational decisions that needs to be made within our application, proper testing architecture should be designed before the construction of the feature ever begins. This is not quite a TDD (Test Driven Design) strategy, but a riff on the original idea. Instead of building all possible tests before construction, we incrementally build tests at the beginning of each weekly sprint, compounding/playing on the project lifecycle position relative to the prior week's progress.

Methodology

The approach Camp Scout has taken to testing is decision-based analysis. Instead of writing numerous tests on functionality for all foreseeable components, tests are only designed for elements of the application that actively “choose” a route for the application to take during use. Since these “choice” elements of the application are the crux of Camp Scouts functionality, testing these features will not only increase efficiency in time delegation, but also ensure that the most important components of Comp Scout are working as expected.

Scope

Below is a list of features subjected to the given testing strategy:

- Unique account credential entry validation within the sign-in page
- Password complexity validation within the sign-in page
- Unique job creation validation within the job-creation page

Objectives

Below is a series of axioms on which the testing strategy for Camp Scout rests upon:

- All major use cases must be tested and accounted for
- All UI bugs need to be resolved before IT Expo
- All decision-based business logic will work as expected before IT Expo
- All major features we need to be working as expected before IT Expo

Test Logs and Procedures

Item#	Test Case#	User	Role	Expected Output	Actual Output	Pass/Fail	Reason for Failure/Success	Date
1	1	John	Create User Account	Database will record the users' credentials in the user collection	The user credentials were successfully stored	Pass	Successful REST endpoint communication	1/22/2022
2	2	John	Create a duplicate user	User creation will be rejected	The user was able to be created	Fail	Logic that checks existing users for duplicates was misconfigured	2/03/2022
3	3	John	Create 10 simultaneous searches to be processed	Each search will be sequentially processed in the order that they were entered.	Some were processed correctly, but others were being processed incorrectly	Fail	The logic involved in deleting a job after it had been successfully processed was misconfigured.	2/16/2022
4	4	John	Create 3 users and then submit 5 searches per user	All jobs will be processed correctly, and each user will be notified to their successes	Each user was notified correctly when the searches they submitted were terminated	Pass	The job processor in the backend is performing as expected. Messaging service is successfully populating messages with search info.	3/04/2022
5	5	John	Submit one search each for all 50 states.	Each search will be processed in the same way as every other. No difference in functionality.	Select campgrounds were not processed correctly	Fail	Certain states structure their campground data differently.	3/24/2022

Testing Review

Conclude the testing section with a retrospective on what you learned during testing, if you reworked any parts of the project, what you would change in the future now that you have been through this exercise.

Change Management Plan:

During the occurrence of feature removal from Camp Scout's development lifecycle, the appropriate response will be initiated as follows:

- John Adkerson will request a change and John Adkerson will handle the change implementation
- Triage Strategy
 - Price the feature is question's reward in terms of Camp Scouts functionality versus the cost to Camp Scout's development lifecycle
 - If the cost outweighs the reward, in terms of what threshold John Adkerson has set, the feature will be deemed for removal
 - If a feature will set the project back one week but is still viable and important for Camp Scout's functionality, this feature will be labeled "Code Green"
 - If a feature will set the project back two weeks but is still viable and important for Camp Scout's functionality, this feature will be labeled "Code Orange"
 - If a feature will set the project back three weeks but is still viable and important for Camp Scout's functionality, this feature will be labeled "Code Red"

- The instant that a feature is deemed as condemned for removal, the feature removal log sheet will be updated with the information associated to the feature in question. Upon each weekly sprint meeting retrospective, John Adkerson will present this feature removal log to the advisor for review

Budget:

Below is a table displaying the transient and long-term costs associated with the work/resources necessary to complete the project. Team labor, as well as Amazon Web Service and Twilio subscriptions will serve as the ongoing costs, and hardware purchases will serve as the designated transient costs.

Table 1 Project Budget

	Rate Per/Hr	Work Effort (Hours)	1 X Costs
Labor - IT	20	600 hrs	\$ -
Labor - External	N/A	N/A	\$ -
Software - External	\$0.0075	5000 instances	
Hardware - External	0.048	6 instances	
Misc.	N/A	N/A	1x Server = \$1000
TOTAL	13037.776	12037.776	1000 +

Problems Encountered and Analysis of Problems Solved:

Below is a bulleted list of the problems encountered during Camp Scouts project development lifecycle. Following each listing is a brief description of how the team dealt with the problem.

- Difficulty reverse engineering common React templates for use within the application.
 - Our team ended up referencing a template while building a less complex framework from scratch within React.
- Difficulty animating and physically placing UX elements.
 - After much experimentation and many hours of tinkering, our team was able to place UX elements in a centered and organized manner.
- Difficulty implementing an autocomplete search bar where users can find desired campgrounds
 - Our team ended up pushing the feature backwards in the project lifecycle. Obtaining a live feed of every campground in America is surprisingly difficult.
- Difficulty developing an effective API technology that could serve as a translator between our concurrent JavaScript/Python processes.
 - After installing 3 different REST API technologies, and watching them all fail, our team luckily discovered Flask, which is a wonderfully simple and flexible API technology that met all the needs required by Camp Scout.

Conclusion

Below is a bulleted list of the biggest lessons I have learned so far during this project.

- Front End/UX design
- Database construction/integration
- API construction/integration
- JSON communication strategies
- Click listener implementation
- Xml http requests construction via JavaScript
- Business logic in handling campsite availability data
- System dashboard to record all system events/logs.
- Cloud security to preserve our user data.
- 3NF database schema within Google Firebase to serve as Camp Scout's persistence mechanism.
- Rotating queue of user "jobs" that the backend will process in a sequential fashion.
- Automatic update, create, delete, read commands within our API to remove jobs once they finish, and facilitate automated Camp Scout operations.

References

- Craig, C. A., & Karabas, I. (2021). Glamping after the coronavirus pandemic. *Tourism and Hospitality Research*, 21(2), 251–256. <https://doi.org/10.1177/1467358421993864>
- Craig, C. A. (2021). Camping, glamping, and coronavirus in the United States. *Annals of Tourism Research*, 89, 103071. <https://doi.org/10.1016/j.annals.2020.103071>
- Recreation.gov. (n.d.). *RIDB Recreation Data*. <https://Ridb.Recreation.Gov/>. Retrieved September 23, 2021, from <https://ridb.recreation.gov/download>