The World's National Parks

by

Max Mukimov, Ryan Jordan, & Malik Aktleuov

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

Max Mukimov, Ryan Jordan, Malik Aktleuov	04.08.2020
	Date
Abdou Fall	04.08.2020
Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2020

DEDICATION

This project is dedicated to all the National Park Rangers and volunteers.

TABLE OF CONTENTS

List of Illustration

ACRONYMS AND ABBREVIATIONS

API	a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.
AWS	is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow. ... Using managed databases like MySQL, PostgreSQL, Oracle or SQL Server to store information.
DBeaver	is a SQL client software application and a database administration tool. For relational databases it uses the JDBC application programming interface (API) to interact with databases via a JDBC driver.
EC2	Amazon **EC2's** simple web service interface allows you to obtain and configure capacity with minimal friction.
Elastic Beanstalk	**AWS Elastic Beanstalk** is a compute service which makes it easier for the developers to quickly deploy and manage applications which you upload to the **AWS** cloud.
GeoLocator	the process or technique of identifying the geographical location of a person or device by means of digital information processed via the Internet.
GIS	A geographic information system (**GIS**) is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographical data.
IDE	An integrated development environment (**IDE**) is a software application that provides comprehensive facilities to computer programmers for software development.
IUCN	International Union for Conservation of Nature
JSON	JavaScript Object Notation, **JSON** is a lightweight data-interchange format that is easy for humans to read and write, and for machines to parse and generate. **JSON** is based on the object notation of the JavaScript language.
MySQL	**MySQL** is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.
PHP	PHP is a script language and interpreter that is freely available and used primarily on Linux Web servers. PHP, originally derived from **Personal Home Page** Tools, now stands for PHP: Hypertext Preprocessor, which the PHP FAQ describes as a "recursive acronym."
RSS	Really Simple Syndication, a standardized system for the distribution of content from an online publisher to Internet users.
RESTful	**REST** stands for Representational State Transfer. (It is sometimes spelled "**ReST**".) It relies on a stateless, client-server, cacheable communications protocol -- and in virtually all cases, the HTTP protocol is used. **REST** is an architecture style for designing networked applications.

UI	short for user interface.
UNESCO	is the United Nations Educational, Scientific and Cultural Organization
UX	short for user experience.
WCPA	World Commission on Protected Areas (**WCPA**) World Database of Protected Areas (WDPA/Protected Planet)
XML	a metalanguage which allows users to define their own customized markup languages, especially in order to display documents on the Internet.

ABSTRACT

There are around 2700 locations with National Park in their title on our planet, and more than 6500 locations with Category II type of protected areas that defines “National Park” per International Union for Conservation of Nature (IUCN), and its World Commission on Protected Areas (WCPA). So far, there is not a single website or mobile app dedicated to the World’s National Parks where all the information about the parks is available on a map. If any user wants to search information about the world’s national parks, he or she will have to rely on Wikipedia, TripAdvisor, travel blogs, or find the park’s official website. Searching for a list of World National Parks on online search engines usually returns an incomplete Wikipedia page or short list of well-known or well publicized National Parks. That makes the search very difficult and doesn’t give all the information. Also, there is not a single database or API with all the world’s national parks on the Internet. We’ve created The World’s National Parks website and mobile application that provides a brief information and a map of all the National Parks. Mobile Application has features like finding parks nearby and directions.

1. PROBLEM STATEMENT

1.1. Introduction

National Parks are locations with some of the most unique, beautiful, and rare form of flora and fauna, and with best scenic views our planet has to offer. Most of them are considered as UNESCO's World Heritage sites or Biosphere Reserve locations. The goal of preserving the National Parks is so the future generations will enjoy them too. Almost every country in the world has National Parks that feature unique beauty and biodiversity of the region. Not surprisingly, there has been increase of National Parks visitation. For example, Great Smoky Mountain had more than 11 million visitors in 2017 which is the world record among National Parks. In Costa Rica, from 1985 to 1999, tourism in parks increased by 400% and according to Nacaragua.com, tourists prefer more to visit parks than beaches and cities. Yet, it's hard to find information about all the National Parks in single resource. Users will have to refer to multiple pages and apps. The World's National Parks Web and Mobile app will try to fix this issue.

1.2. Project Description

The World's National Parks Web and Mobile app will let users to view our planet's National Park in a single resource. The user will not only see the locations of all the National Parks but also be able to create their own list, search with filters, comment on a National Park, rank, make suggestions, and upload one photo. Team members of this project will continue on working on it after graduation as a side project, similar to the World Waterfall Database project www.worldwaterfalldatabase.com. In addition to features mentioned above, The World's National Parks mobile application will have features like Nearby Parks, which will allow a user to locate a National Park within a certain distance of them. Also, the mobile app

will be equipped with RSS function, which will collect new data on National Parks. Both apps will use the database our team created for this project, to get information about locations and brief information on each park.

1.3. Problem

In order to find a National Park's location in the world and see how they spread out, a user will have to rely on multiple resources. A simple search will take a user to a Wikipedia page where a National Parks' list is available for each country, but 55 countries pages are missing. In addition, some countries have listed National Parks pages, but with minimal or unrelated information (Niger, Paraguay for examples). Another tool would be Google Maps which relies on user input but can have discrepancy and inaccurate information. Next, the user can refer to TripAdvisor or travelers' blogs, which mostly contains information about personal experience with National Parks. Most countries have an official website dedicated to National Parks, but not all of them are in English, and the user will be limited to one country only. Some hobbyist and ecotour companies have created very informative webpages on specific countries' National Parks, but again they have information only about one country and with tour advertising purpose. None of these options will give a user the ability to see all the National Parks' locations at the same time, nor will it allow the user to customize the data in any way. Nevertheless, the user has an option to create a customized personal Google Maps map and manually mark the parks but this is time consuming. One of our team members started the endeavor but stopped after 126 entries. To illustrate the situation with analogy, one can imagine someone who wants to see all the states of the United States, but is limited to Wikipedia, Books, blogs, and TripAdvisor. In order, to make the situation closer to our project, we must imagine that the United States map doesn't exist. In this situation, the user can still get some information, but it's a completely different experience if the user sees the

map of the United States with all the states together. This makes it obvious for need of a Web and Mobile App, not only with all the National Parks' location but also with a customization option.

1.4. Solution

The World's National Parks website will provide information and a map of all the National Parks. A user will be able to search for National Parks and filter search criteria. The website will enable user to see how National Parks are spread-out throughout the continents and be able to see basic information about each one of them. In addition, users can rate and leave a comment about a specific National Park. Since our team will apply the Bootstrap framework it will enable the mobile view, so users can visit the website on mobile devices. One part of the project will be creating a database with each park's information such as country it belongs to, area, established year, official website, image, latitude and longitude, brief description and if it is considered a World Heritage site. Since database is collected using the MySQL system, we'll be able to create several database files types: JSON, XML, GIS. This database will enable a RESTful API to be created for the website and mobile application.

Additionally to creating the website, our team will work on an Android application of the World's National Parks. This will let a user to choose viewing the website in mobile view or download the application. The mobile application will have a Google Maps view of National Parks' locations and they will be pulled from our database. The Android application user will be able to sign in and mark the location of National Parks as visited. For marking the location, the user will earn points, and app will have verification feature where the app will be able to confirm user's location. If a user gets verified, he or she will be able to earn more points. The app will have additional features like users' rank, bucket list, find nearby National Parks, and

RSS feed with news on National Parks. Both website and Android app will use the same data, so a user will be able to see their own profile on both versions.

Even though the database that our team is creating is only one tenth of the project, it is an essential and crucial part of it. Since such a collection if information does not exist in database form, this makes the solution unique for the time being.

2. PROJECT MANAGEMENT

2.1. Budget

Table 1 Budget table presents the budgets for project. It shows the cost of different types of duties we've completed based on current prices. This included labor for data entry, developing Web and Mobile applications and other expenses. It does not reflect what it will actually cost since we are doing it for free.

ITEM	UNIT #	COST	TOTAL
Labor for Data Entry	600 hours	$12	$7,200
Web App development	800	$35	$28,000
Android App development	1000	$35	$35,000
Certificates and Domain registration	2	$35	$70
TOTAL			$70,270

Table 1.

2.2. Project Objective/Goals.

This project will create a dynamic and interactive website where users will be able to see all of the National Parks locations on a Google Maps map. In addition, users will be able to see basic information on each National Park by clicking on it: area, year established, link to official website, coordinates. Search option will allow user to filter search criteria and save the results. The National Parks database, created by the team will be used with Google Maps API for the website and Android App, and the data will be available in XML, JSON and GIS formats. The Android application will let the user find the closest National Park based on location and mark that National Park as visited. Users will earn points for visited National Parks and verified visitors will get more points. The application will use Facebook or Google

logins, so users won't have to create additional account. Also, the Android app and website will display a chart of users who visited most of the parks.

The features of The World National Parks will include:

- Users will be able to view all the National Parks on a Google Maps map, and see basic information about each National Park. Besides the map, user can view National Parks as a list and filter the list.
- Users will have the ability to login to The World National Parks website and Android app with Facebook and Google accounts. No separate account needs to be created.
- Each user will have the ability to save the search results and make reviews on any National Parks. The website will have the option to view in mobile view, and the Android application will let the install app version.
- The Android application will allow the users to mark National Parks as visited and the app will also let the user find the closest National Park from their current location.
- Database file will be available in XML, JSON and GIS formats, and will be free to use in noncommercial purposes.

These features will let the user to find the basic information about National Parks, find them on the map and leave a review about them.

2.3. **Methodology**.

Our team used both Waterfall and Agile methodology for this project. Since it's a school project and we have a strict schedule on deliveries and outcomes, at the same time we used Agile tools like Scrum meetings and Sprints. Every Saturday, our team had Scrum meetings at Blue Ash campus. Meeting were scheduled, even if there are no deliveries for that week, this helped to keep the team on pace. During our Scrum meetings, we had to

update our Design requirements for UI and back end development. We had encountered obstacles and came up with new solutions to improve our project. Big changes like changing the data entry application, different delivery to search results, adding an Android Application, changing database connections, and conducting surveys were applied as a result of Scrum meeting discussions.

2.4. Gantt Chart.

The following Gant charts illustrate timeline of the project delivery and allocated time for each task. Each Gantt Chart in Figure 1 and Figure 2 represent one of each semester.

Figure 1: Fall Semester Gantt Chart shows Fall project schedule with the major milestones listed for Fall semester.

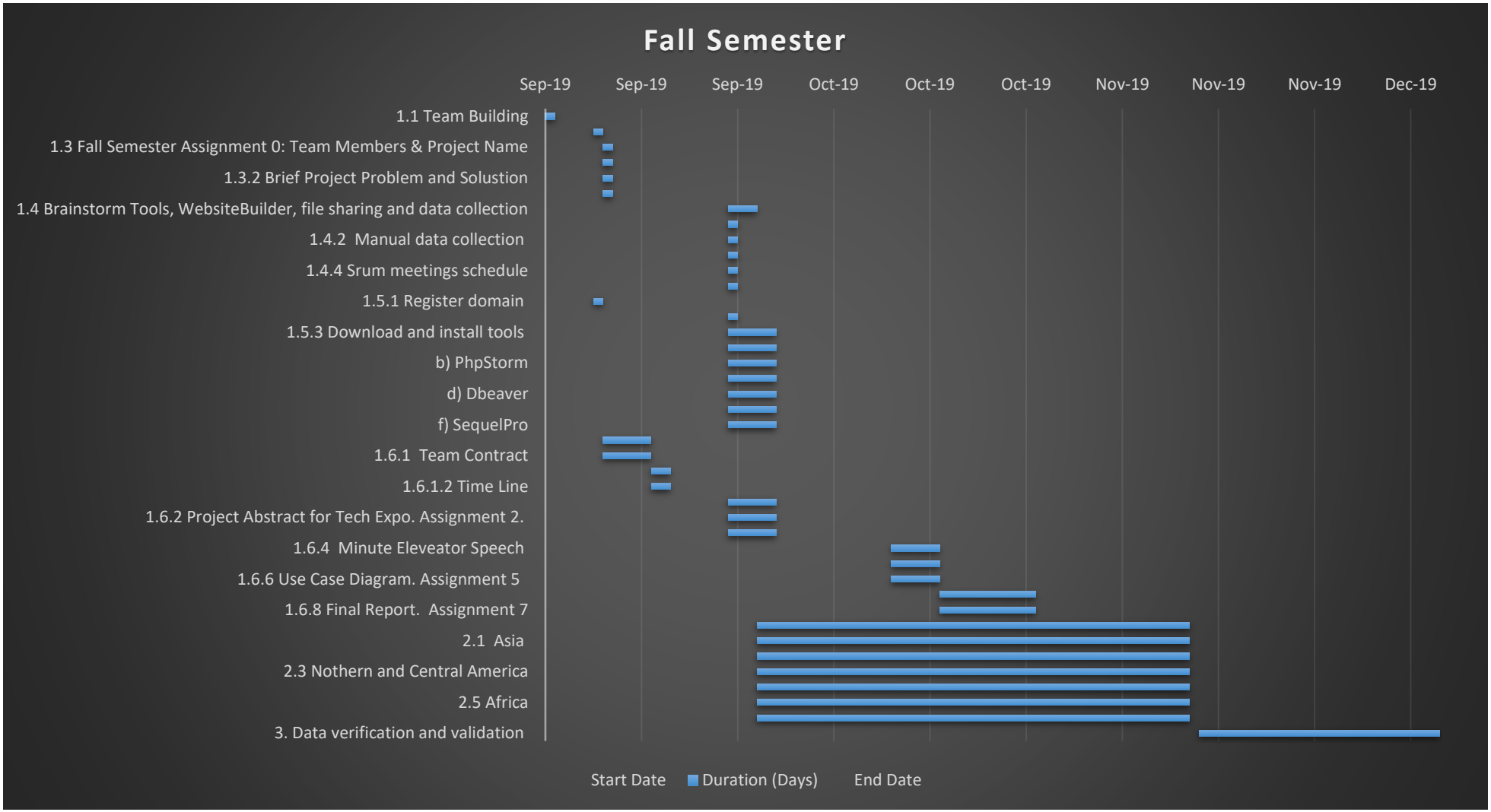

Figure 1.

Figure 2: Fall Semester Gantt Chart shows Spring project schedule with the major milestones listed Spring Semester.

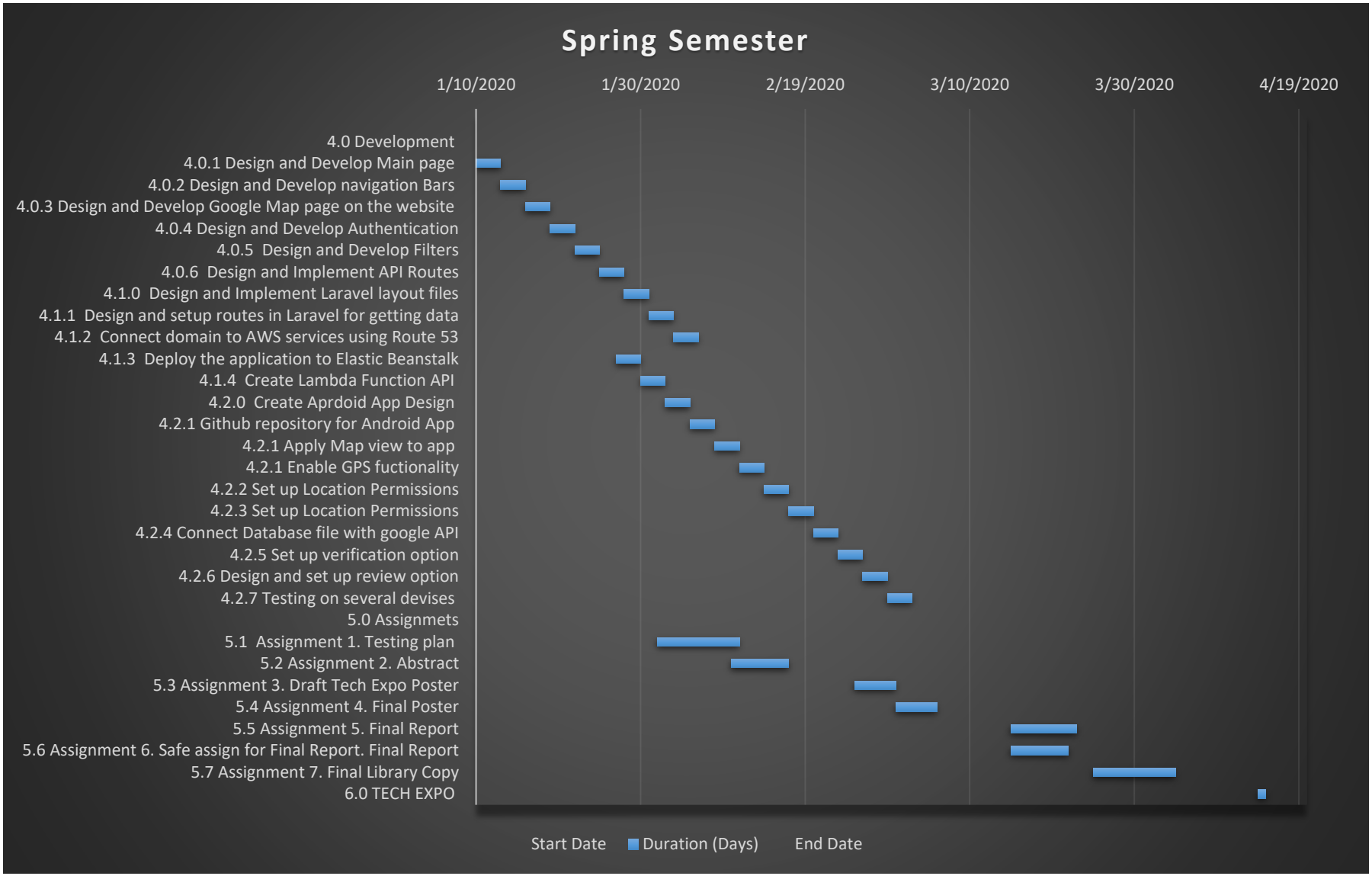

Figure 2.

2.5 Work Breakdown Structure.

WBS in ***Figure 3: Work Breakdown Structure*** represents a deliverable-oriented breakdown of the project into smaller components. Components are color marked, tasks in green color have been completed, yellow tasks are in progress, and orange tasks are to be completed by the end of the year and next semester.

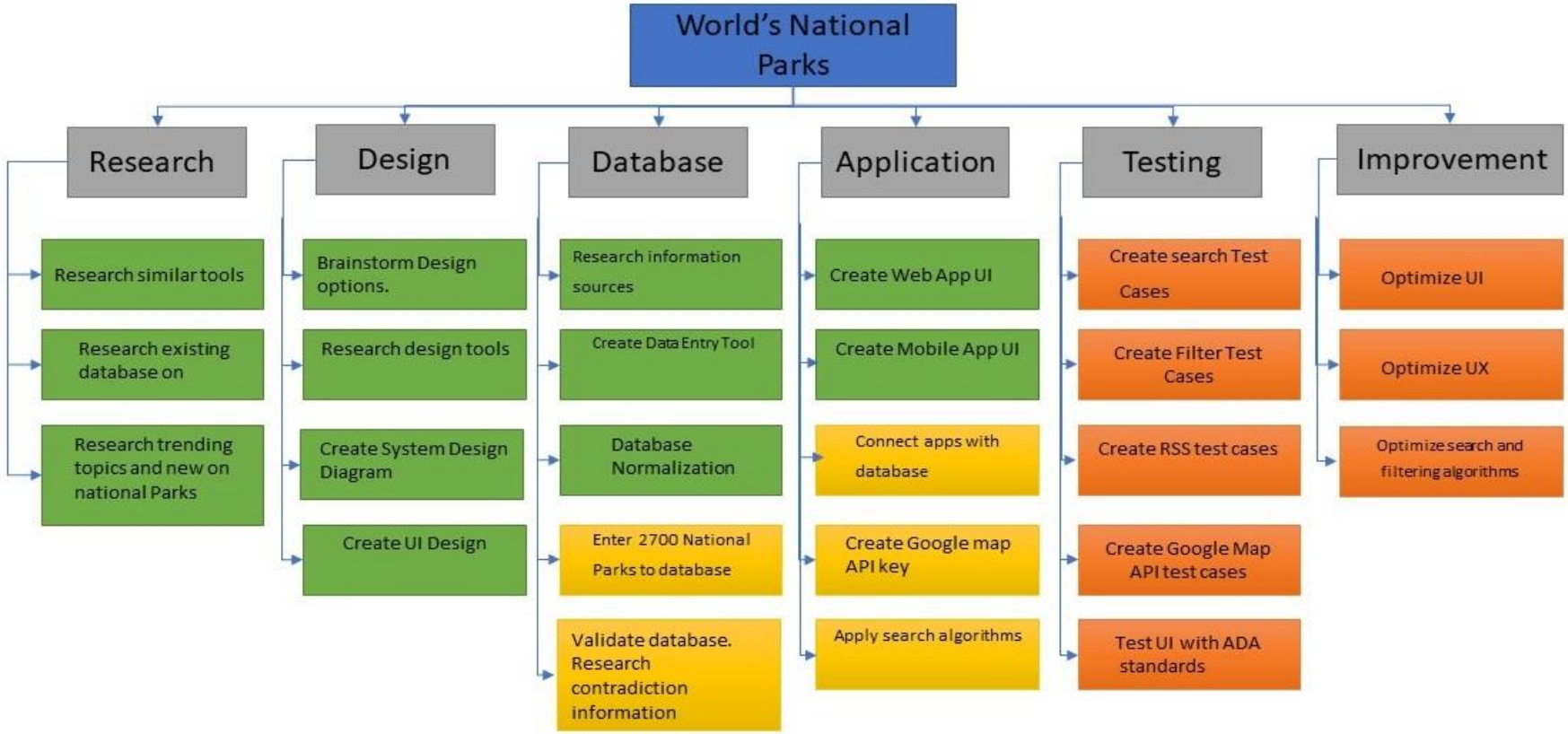

Figure 3

2.6 User Profile

Figure 4: User profile: Application user illustrates user's profile for The World's National Parks Web and Mobile application. It shows who can be potential users of applications, similar application and user's task experience.

USER PROFILE: Application User
PROJECT: **The World National Parks**
POTENTIAL USERS: - People interested in National Parks - Travelers - Landscape Photographers - International Organizations members that deal with nature and wildlife protection.
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: This project will be primarily targeted towards users that are comfortable using Web and Mobile applications. Both website and mobile application will be intuitive and easy to use. Only a basic knowledge of geography is required.
EXPERIENCE WITH SIMILAR APPLICATIONS: Since there is not a single database on all the National Parks, users can find information on them on several websites: - Wikipedia - Government or travel websites

- Manually search on Google Maps.
- Separate mobile app on each country (US National Parks, Australia National Parks, India National Parks, etc.)

TASK EXPERIENCE:

At initial opening of the application, the user will be able to find National Parks by region, country or National Parks' name. User will have the ability to save the search results. The search results can be displayed as a list or on the Google Maps map. By clicking on search results on the map, user can find basic information on each park. Users can leave a review on a National Park and mark the park as visited.

FREQUENCY OF USE:

Web app URL and mobile app will be used whenever a user wants park information.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Visually fluid and easy to navigate UI
- Easy search function.
- Tips to get started
- Quick start/setup

Figure 4.

Figure 5: User profile: Application Admin shows interactions of The World's National Parks applications with admins. It shows how the admin interacts with applications on different level.

USER PROFILE: Application Admin
PROJECT: **The World National Parks**
POTENTIAL USERS: - Web and Mobile Application developers. - Business Analyst.
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: App admins will have all access to Front and Back End of the applications. In addition, they have full control of database and support its integrity and consistency.
EXPERIENCE WITH SIMILAR APPLICATIONS: In order to validate the database and update the information, the app admin can utilize following resources: - Wikipedia - Government or travel websites - Google Maps - Separate mobile app on each county (US National Parks, Australia National Parks, India National Parks)

- Travel Books
TASK EXPERIENCE: Admins will use custom data entry Web app for new park entry, DBeaver for database management, Sublime Text for Web app development, Android Studio for mobile app development and AWS S3 bucket as photo storage.
FREQUENCY OF USE: Admins will update database every time new information comes in. Web and mobile application will be constantly updated and improved based on new available libraries that can improve user experience or based on user suggestions.

Figure 5.

2.7 Use Case

Figure 6: Use Case Diagram presents the project use case diagram. Share use cases consist of 2 parties in total. For the most part, the same level of functionality is available to both parties. These parties consist of the group creator and the group member. Group creators have the expanded ability to invite members, although any group member can create their own group and invite members to it.

Figure 6

2.8 Problem Encountered and Analysis of Problem Solved.

2.8.1 Database and Data Entry.

Our team created the database using MySQL and a data entry Web App which enabled us to do the data entry very quickly. The data entry process itself got complicated with information not being available on many National Parks in English. Lists of countries' National Parks didn't have pages on 55 countries in Wikipedia. Our team had to refer to other sources, or pages in different languages, since members of our team are fluent in Russian, Uzbek and Kyrgyz. In addition, our team members speak in entry level Spanish, Polish, Chinese, Japanese, Turkish and German which helped with information gathering. Also, volunteers who helped us with data entry are fluent in Khmer, Spanish and Chinese. Most of the time we had to rely on government sites published in native languages.

2.8.2 Web Application

Since we are using the Laravel PHP framework, it makes it very simple to connect a database to a Web application. The problem is that this is all done server side, not client side. This means once we use PHP to pull the data and put it into an array, we can't use it with the client side Google Maps API. The solution to this was to ditch using Laravel to get the data directly and instead create an API in AWS. We wrote an AWS Lambda function to get data when it is requested by AWS API Gateway. This means we could use the data from our own API to put through the Google Maps API. We will still be using Laravel for working with user accounts.

2.8.3 Mobile Application.

The main challenge with Android App was to connect our database stored in AWS to Google Maps. While researching the issue of applying REST API on Android App to load the data, a temporary remediation was to download the database from DBeaver in JSON format and load the data as ‘local data’ within the app. One of the advantages of using ‘local data’ is that it allows the app to load faster since the app doesn’t have to pull the data AWS at every load. Since other function on the app use the same ‘local data’, the overall reply time of every click is much faster.

2.8.4 National Parks’ images.

Part of the data entry process was to collect one image of each National Park. Since the project will need more than 2700 images, we’ve tried to reach out to several resources. One of the team members is a Google Local Guide Level 8, so he reached out to a local guide members forum and asked suggestions about using images from Google Maps. Most of the Local guides suggested that using images directly from Google Maps may violate the Use Terms of Google Maps, even by giving the credit and with photographer’s permission. Next our team reached out to several travel photographers, most of them replied that their images are owned by publishers like travel journals or National Geographic. One of the photographer’s assistants suggested that since our project is for school, we should be able to use the images under ‘Fair Use” act. Due to the fact that our team will probably continue working on the project after graduation this option was not suitable. Our team decided to use images from Wikipedia. By attributing the author and providing a link to the license, photos from Wikipedia can be used freely. For the parks where images are not available on Wikipedia, our application will note that Image is missing and will give an option to upload one image.

3. TESTING

3.1 Overview and Methodology

In this section we'll discuss how testing was organized and performed for our project. The World's National Parks project has three main components: database, Web App and Android App. Each component will be tested separately based on its functionality by using the most suitable tools. The apps of our project were built by using Agile Methodology, since this methodology allows to handle constant changes easily. During the building process our team had to implement a lot of changes in the database, UI, app's functionality, and implementing our project by small iterations eliminated necessity of going back to the beginning and start all over. Since we applied Agile Methodology, testing was part of each iteration.

3.2 Scope and Level of Testing

Our tests will cover all major features in the app. In the database, all the tables, columns and rows will be subject for structural, functional and non-functional testing. In order to evaluate apps' functionality different level of test will be performed: easy, medium and large.

Figure 7 Testing Pyramid illustrates how different level of test are used in for our apps.

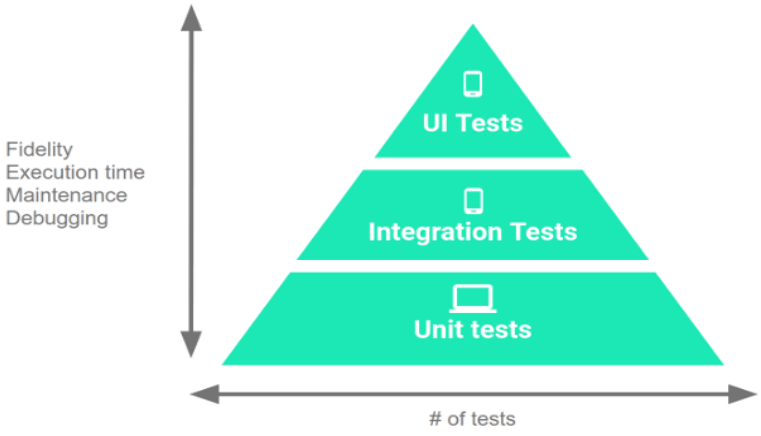

Figure 7.

- Small tests are unit tests that validate our app's behavior one class at a time.
- Medium tests are integration tests that validate either interactions between levels of the stack within a module, or interactions between related modules.
- Large tests are end-to-end tests that validate user journeys spanning multiple modules of our app.

As we work up the pyramid, more tests will be created as a unit test, each test increases in fidelity but also increases in execution time and effort to maintain and debug. Overall 70 percent of a test will be performed in Small tests as a unit test. The Scope of a Small Test will be each method and class used in apps.

3.3 Testing Objectives

The main objective of our testing is to sustain the functionality of apps and database. Since the database and each application work differently, each one will have different testing objectives. The objective of database testing will be performing Structural, Functional and Non-Functional testing of the database.

- Structural testing involves the validation of all of the elements inside the data repository. Part of the Structural testing is schema testing or mapping testing where we make sure that the World’s National Parks Web and Mobile App will be connected to the database the same way. For example, if a new national park is added to database both apps should show the new park entered in search results.
- Functional testing as specified by the requirements specification needs to ensure operations performed by app and database align with what was required. For example, if an end user asks to pull up national parks in Italy, the apps should both give accurate result.

- Non-Functional Testing can be load testing, stress testing, security testing and so on. For example, in order to perform stress testing of our database, we can restart or reload database many times to check if it breaks under heavy load.

Mobile Application's testing objectives are app compatibility with different Android devices, UI load test, earlier failure detection, safer code refactoring and etc. Android Studio allows us to run applications under different emulators where the developer can choose Samsung, Google or LG phones. That will help us to make sure that the UI is loading the same way on different devices. In addition, UI testing will be performed on actual devices.

Another objective of Mobile App testing is to make sure that all the functions are tested as separate test units. Once all of the unit tests have completed, we can move to medium tests where the app's views, view models, validation of XML layout, interaction with database and DAO's will be tested.

Objects of the Web Application's testing are browser compatibility where we make sure that the page is opening with the same layout.

3.4 Testing Procedures

Testing in the database is performed in different levels of software development. Successful data entry process is confirmed after every data entry like image shown in Figure 8.

Figure 8: Data Entry illustrates successful data entry process confirmation.

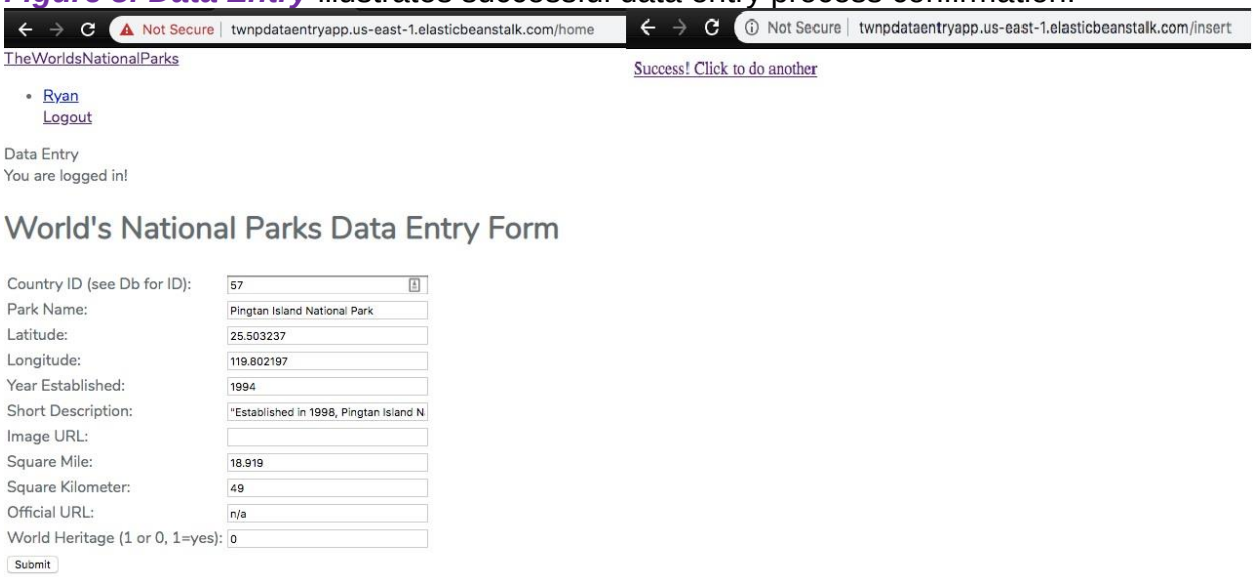

Figure 8.

Testing on database validation is performed using Views created on DBeaver software, where each View was created based on possible scenarios. Figure 8 illustrates how Views are stored in DBeaver software.

Figure 9: Views in DBeaver shows an example of Views in Word National Parks database.

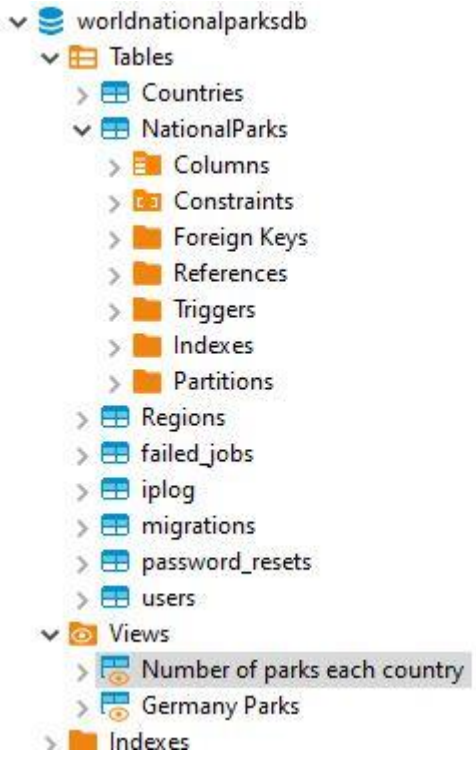

Figure 9.

Android Studio allows you to easily create unit tests for each method and separate tests for infrastructure tests which is more like medium tests. All the unit tests will be stored in two separate folders: test and Android test. Android Studio allows you to run all the test cases at the same time or perform separate unit tests. For example, we can run the test for the National Parks News page where a class executes with a new window loading and at the same time RSS feeds are pulled. Android Studio allows you to run unit tests for each method in a specific class or run all the tests after each development iteration that reduces issues and bugs during regression tests.

Figure 10: Creating a Unit Test in Android Studio shows how unit test can be created.

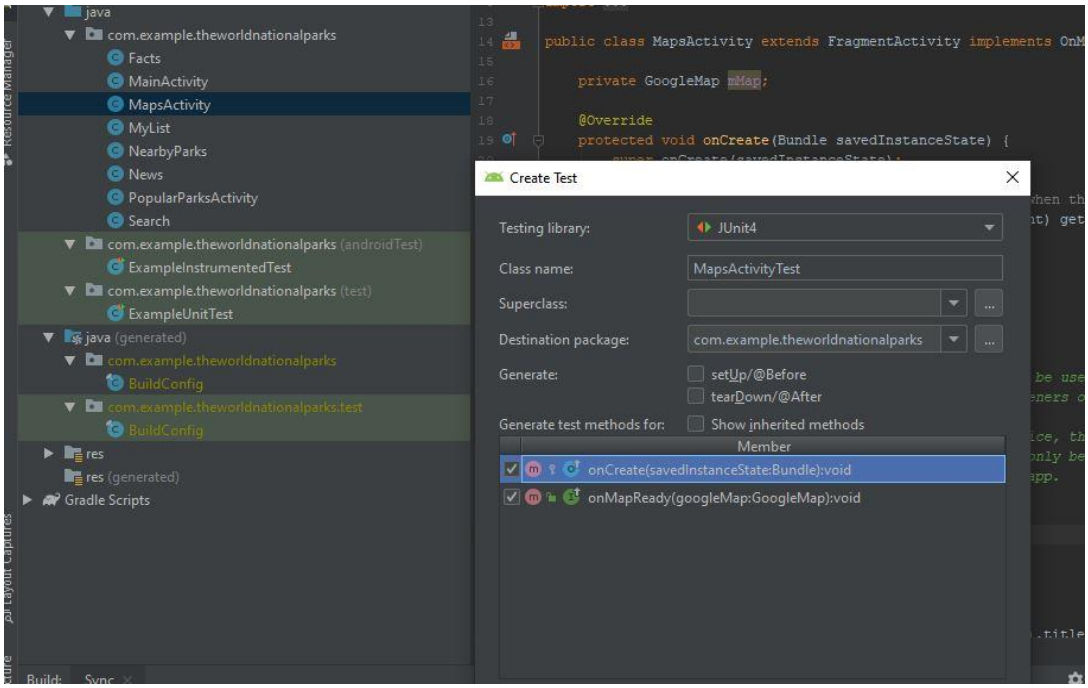

Figure 10.

Figure 11: Unit test examples illustrated how separate tests can be run. In this example, Android Studio tests successful page creation and load of Google Map's API.

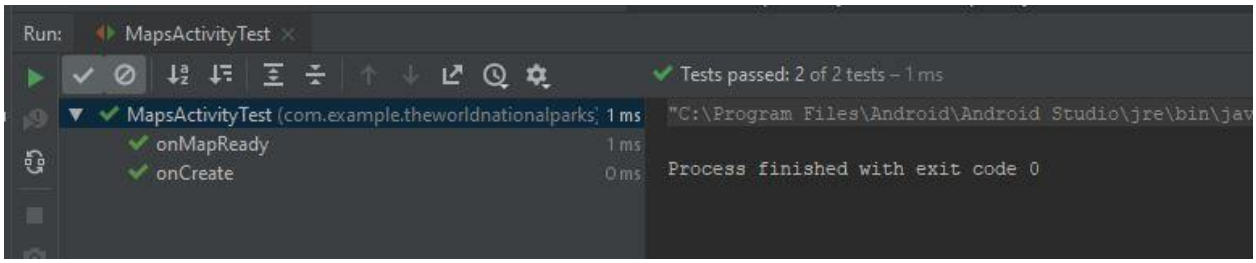

Figure 11.

Figure 12: Testing the reorder sorting button shows that the JavaScript code executes when the button is clicked. This made it easier to diagnose button issues when the sorting would not work.

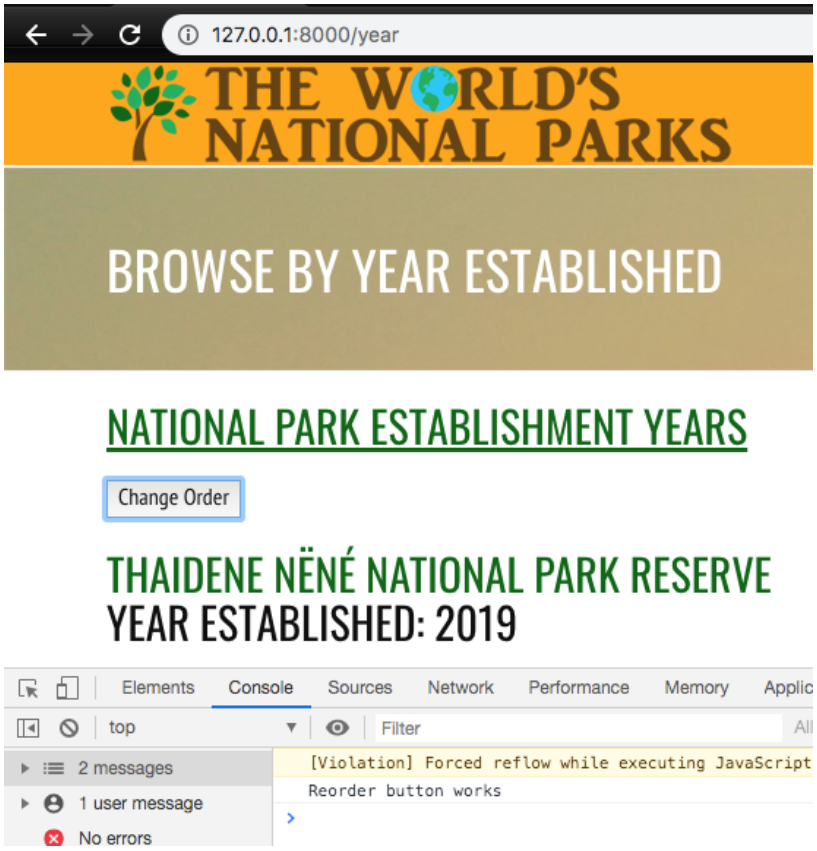

Figure 12.

In addition to tests performed by our team members, separate Black Box and UX Tests will be performed by friends, peer students, coworkers and family members. They will be asked to answer following questions by using the World's National Parks Web and Mobile applications.

- How intuitive was the applications?
- How easy was it to navigate?
- How appealing was the color choice?
- What would you change?
- What would you add?

3.5 Testing Scenarios Report Table.

Figure 13: Testing Scenarios Report Table shows examples of test scenarios for Database, Web and Android Apps

Test Scenarios for World Nationa Park Database	**Description**	**Performaed by**	**Results**	**Date**
Database availability. Stress Test of database.	Call the database repeatedly 50 more times with 10 mins intervals.	Malik	Pass	3/4/2020
Data entry test	Check the data entry and update to confirm that database is updating real time	Malik	Pass	10/10/2019
Search result test	Search for specific National Park, check the results. Report errors.	Malik	Pass	2/15/2020
Create search criterias by using Views in Dbeaver	Views for sorting by largest National Parks	Malik	Pass	1/10/2020
	Views for sorting by countries	Malik	Pass	1/10/2020
	Views for sorting by regions	Malik	Pass	1/10/2020
	Views for sorting by alphabetical orders. Ascending	Malik	Pass	1/10/2020
	Views for sorting by alphabetical orders. Descending	Malik	Pass	1/10/2020
Test database images storage	Check if database is always on and url is available .	Malik	Pass	11/15/2019
Test database images availablity	Check if images saved as public	Malik	Pass	11/18/2019
Test Scenarios for World Nationa Park Web App	**Description**	**Performaed by**	**Results**	**Date**
Browser compatibility	Check if the page can be opened in IE, Chrome, Firefox and Safari.	Ryan	Pass	1/5/2020
Overall speed test	Check latency issues.	Ryan	Pass	1/5/2020
Check search functions	Search for random names and check the results.	Ryan	Pass	4/1/2020
Check Map functionality	check if map loads with National Parks icons	Ryan	Pass	2/2/2020
	check if National Parks page can be oprned through map	Ryan	Pass	2/2/2020
	check if national Parks images are visible on map	Ryan	Pass	2/2/2020
Check Website's UI load	check if CSS is loading properly	Ryan	Pass	2/2/2020
Chek browsing fuction	Check browsing option by Regions	Ryan	Pass	2/19/2020
	Check browsing option by Countries	Ryan	Pass	2/19/2020
	Check browsing option by size of the park	Ryan	Pass	2/19/2020
	Check browsing option by World Heritage status	Ryan	Pass	2/22/2020
Check all the buttons	Check if all the buttons are working properly.	Ryan	Pass	3/5/2020
Check Mobile Version of webb app	Check if Mobile version is loading in Android and iPhone.	Ryan	Pass	3/28/2020
Test Scenarios for World Nationa Park Android App	**Description**	**Performaed by**	**Results**	**Date**
Check app load on Android Device	Load the apk package on differnet android devices.	Max	Pass	2/2/2020
Check Buttons functionality	Check all the buttons and confirm functionality	Max	Pass	2/5/2020
Check UI visibility	Ensure if app's UI is loading accordingly	Max	Pass	2/5/2020
Check Navigation.	Navigate through the app to ensure proper pages are loading.	Max	Pass	2/9/2020
Load the Map	Ensure that National Parks are loading on Google Map.	Max	Pass	2/15/2020
Test image load	Check if images of National Parks are loading	Max	Pass	2/17/2020
Test Page load	Check if pages of National Parks are loading properly.	Max	Pass	2/18/2020
Test RSS feed	Ensure if RSS feeds are loading with newest dates and linsk are working	Max	Pass	3/1/2020
Check Favorites option	Ensure that user can save the park as favorite	Max	Pass	3/1/2020
Chek Visited option	Ensure that Visited option is working	Max	Pass	3/9/2020
Check My List Function	Confirm that Favorite Parks are showing up in MyList	Max	Pass	3/9/2020
Check GeoLocator function	Check 'Find my location'	Max	Pass	3/11/2020
Check Driving option from Map.	Confirm that driving button navigating to Google Map	Max	Pass	3/15/2020
Near Me function.	Confirm that 'Near me' option is finding closest parks.	Max	Pass	3/18/2020
REST API connection	Ensure that Maps are loading using REST API	Max	Fail	3/22/2020

Figure 13

3.6 Feedback from Manual and UX testing.

Manual test of Android app was done by providing the .apk file for the app. Several users from different continents were able to test it. The main suggestions can be categorized as the following:

a) When searching for National Parks list in Search results or in My List option, having the country flag would be more useful. This suggestion mostly came from testers from Europe.
b) Being able to see more pictures for each National Park.
c) Being able to leave comments.
d) Adding travel guide information on each National Parks page.

Overall feedback was that app is interesting, easy to use, and navigate. Additional suggestions were to create an iOS version of it.

3.7 What We Learned During Testing.

We learned that using Agile Methodology where testing is conducted during each iteration, helped us face less issues while creating the app and database. Bugs and defects were found in earlier stages and fixed right away without delaying the issue. While doing our testing we consulted with developers who work on testing and they helped with creating the testing for our apps and database.

We were suggested to implement Red-Green-Refactor testing method in our apps. It's part of the Test-Driven Development where tests are written before writing the code, and developer has to write a code based on the test's requirements. If any of the tests fails, most of the IDE tools throw red error messages. Once bugs are fixed and all the tests are passed with a green success message, the app can be refactored where all the new updates are implemented.

Figure 14: Test Driven Development Cycle shows in graphic chart how Red-Green-Refactor works where refactoring is done after successful test completion.

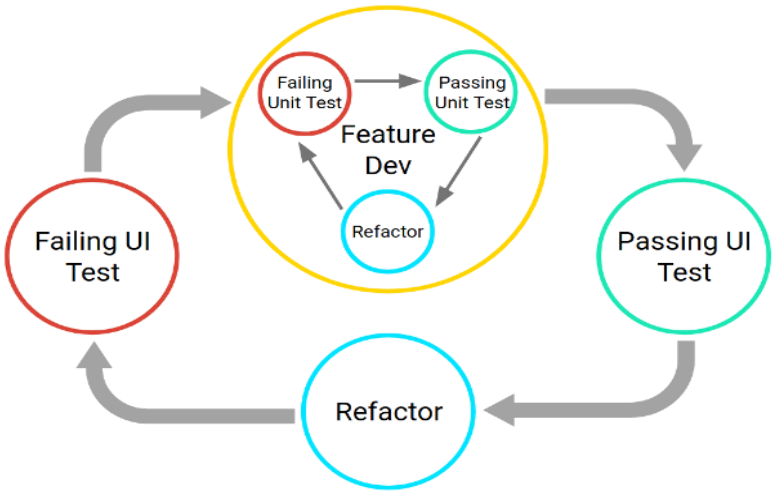

Figure 14.

4. TECHNICAL DISCUSSION

4.1 System Diagram.

4.1.1 Web App Diagram. We created the web application using the Laravel PHP framework for the backend, while the frontend uses the Bootstrap framework. This makes it very responsive to different screen dimensions and devices. We decided that there are features like "Nearby Parks" that would be best suited to a mobile device, so we also created an Android application using Java. This is all hosted on Amazon Web Services using Elastic Beanstalk which is comprised of EC2, S3, and RDS.

4.1.2 Android App Diagram. The Android App is written in Java using Android Studio. The app uses a database in JSON format stored within the app as ‘local data’. For the Google Map, it applies Map Activity to display National Parks location on the app, where location information is pulled from the JSON file. While using the Google Maps map, the app applies GeoLocator for finding the user’s location that enables them to see the closest National Parks from the user’s current location.

Figure 15: System Architecture illustrates how our application are built and connected with each other.

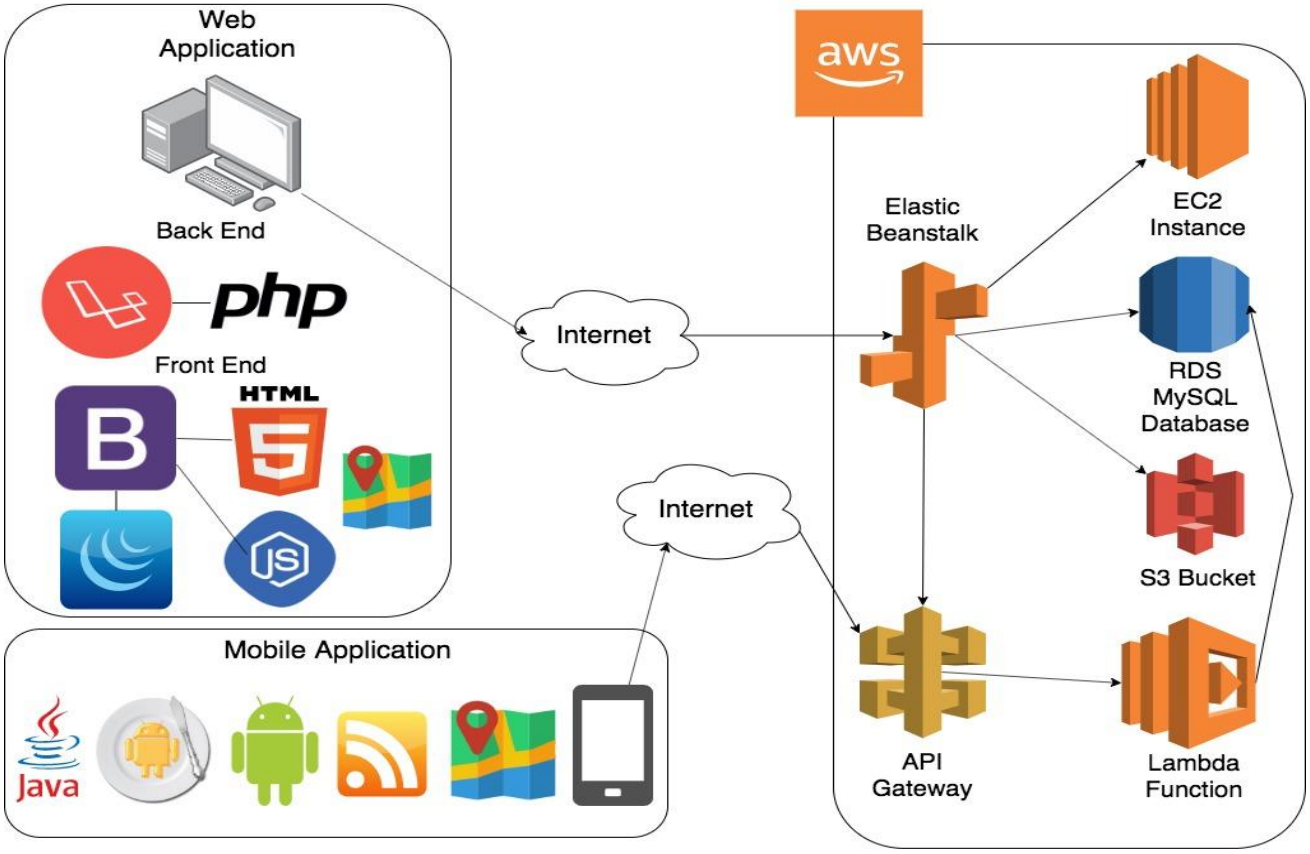

Figure 15.

4.2 Network

Because Elastic Beanstalk was used to deploy this application, it is running on an Amazon Linux EC2 Instance. A classic type load balancer was setup to route all HTTP request traffic (Port 80) to Port 443 and to use our SSL certificate. Our domain was purchased through Google Domains so DNS records were modified to point to AWS DNS servers.

4.3 Security Elements

The GitHub repositories with the code for this project are private with access only given to group members. The data entry application has a login form which takes an email address and a password. The set passwords are complex so they cannot be easily guessed. These passwords are securely stored hashed in our database. After too many failed login attempts it will not let someone try again for a period of time. The Web applications have both been tested for XSS and SQL injection vulnerabilities and none were found.

4.4 Data Entry App and Database.

For the database our team used DBeaver software where scripts were written using the MySQL engine. We've created three tables: regions, countries and national parks. Data entry was performed by making a data entry Web application in the cloud, so our team was able to collect the data and access the data entry site from anywhere. Our team reserved http://twnpdataentryapp.us-east-1.elasticbeanstalk.com domain for this Web App.

Figure 16: Database Diagram shows how tables are organized and related.

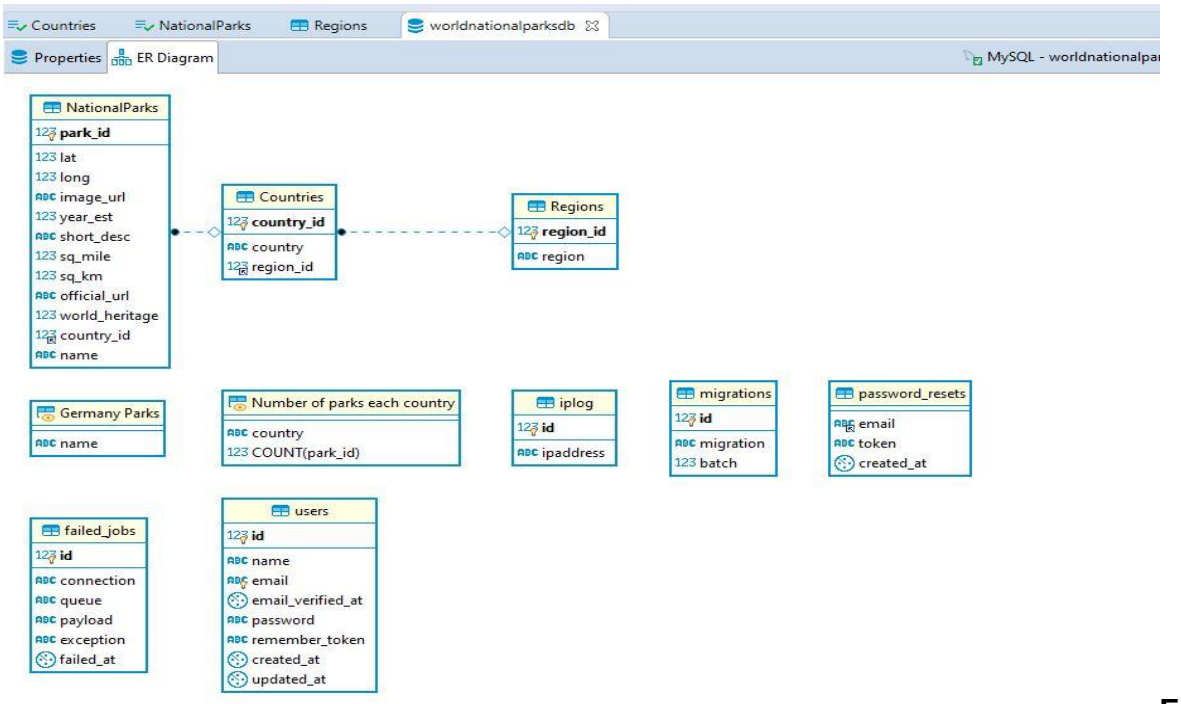

Figure 16

Figure 17: Data Entry Web App illustrates a Web page with data entry form. This data entry form is only for National Parks table in database and collects basic information on each national Park. Half of the page shows source where database connections is visible.

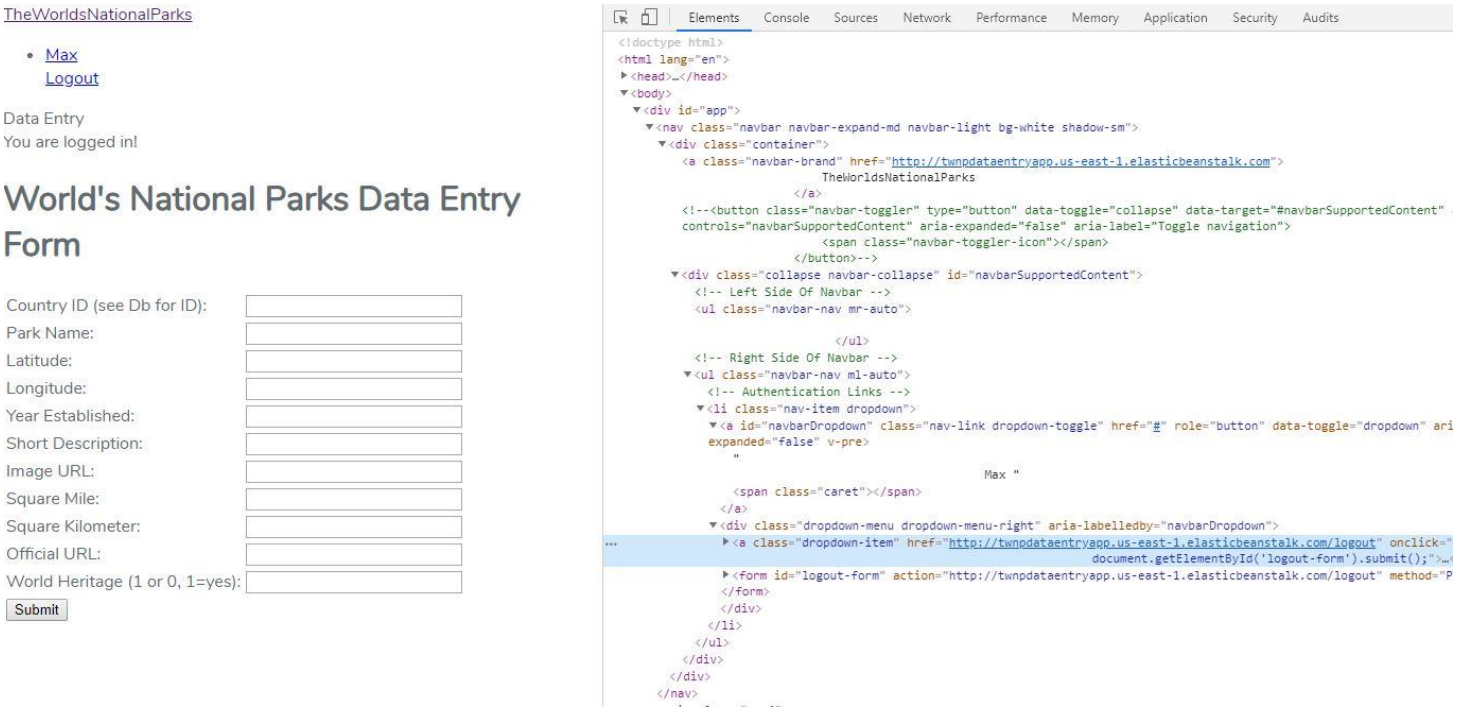

Figure 17

***Figure 18: DBeaver Database with all National Parks*.** Following figure shows our database in DBeaver, collected by our team. In this image, National Parks table is illustrated where columns like Park ID, Latitude, Longitude, image url and short description is visible.

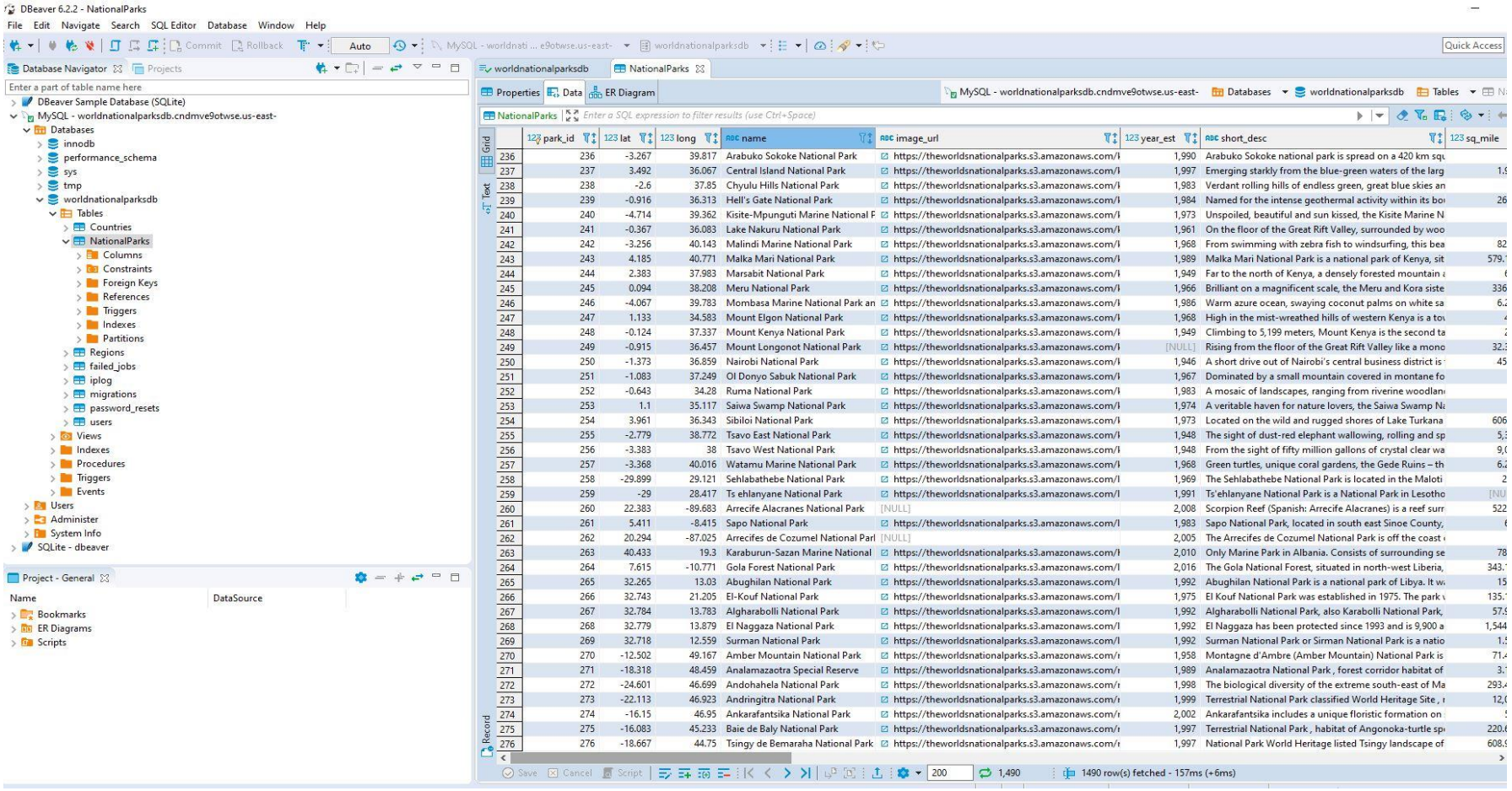

Figure 18.

4.5 Web Application

Figure 19: Homepage of Web Application (top). The Web application was developed using the Laravel PHP framework. The homepage view when user visits the website is a welcome page which also features a search box for finding national parks.

Figure 19

Figure 20: Homepage of Web Application (bottom). The bottom portion of the homepage allow user to browse parks by other categories.

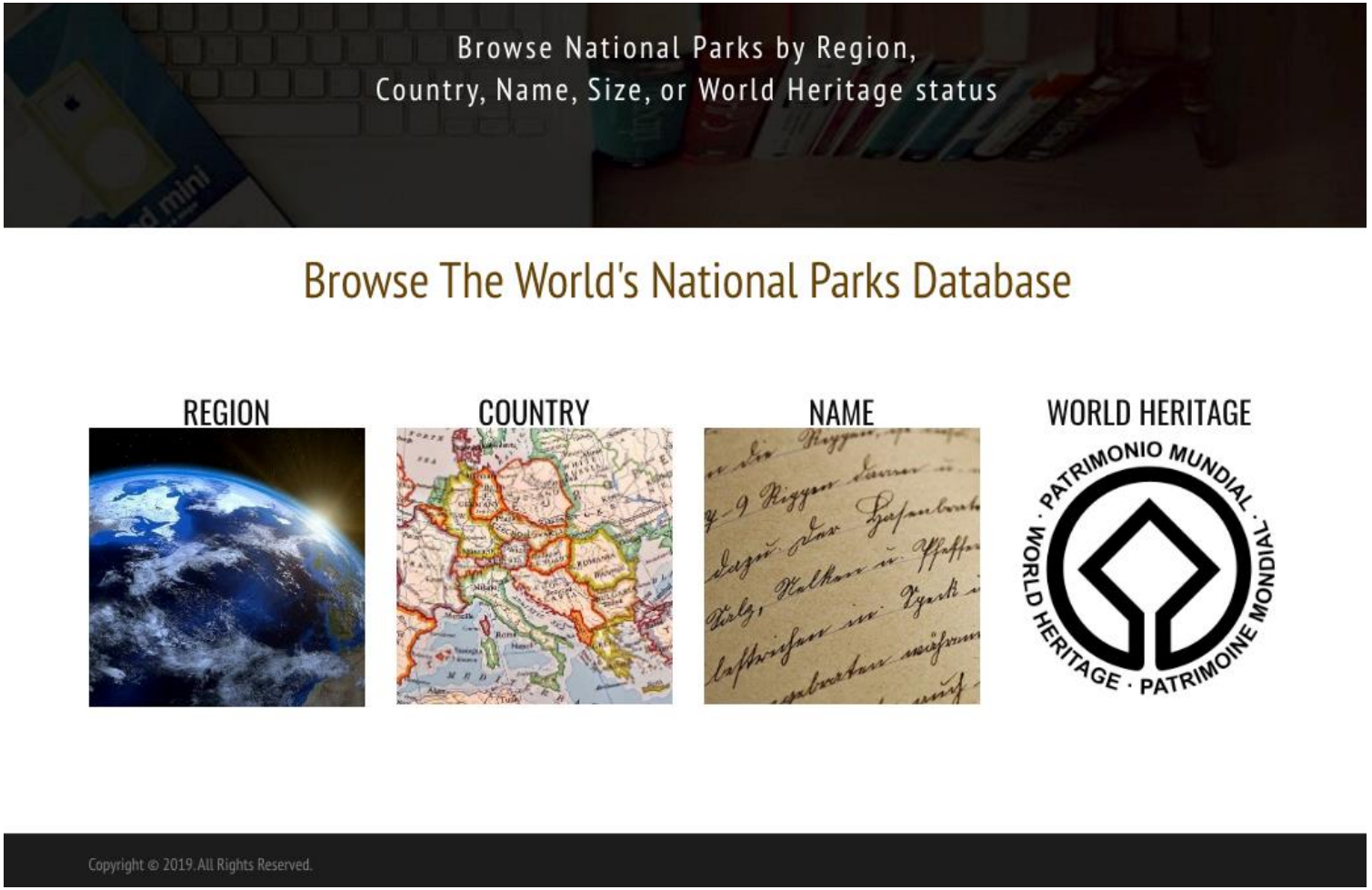

Figure 20

Figure 21: Maps Page of Web Application. This page allows a user to load a custom Google Maps map with pins at each national park. User can click on individual parks for more information about them.

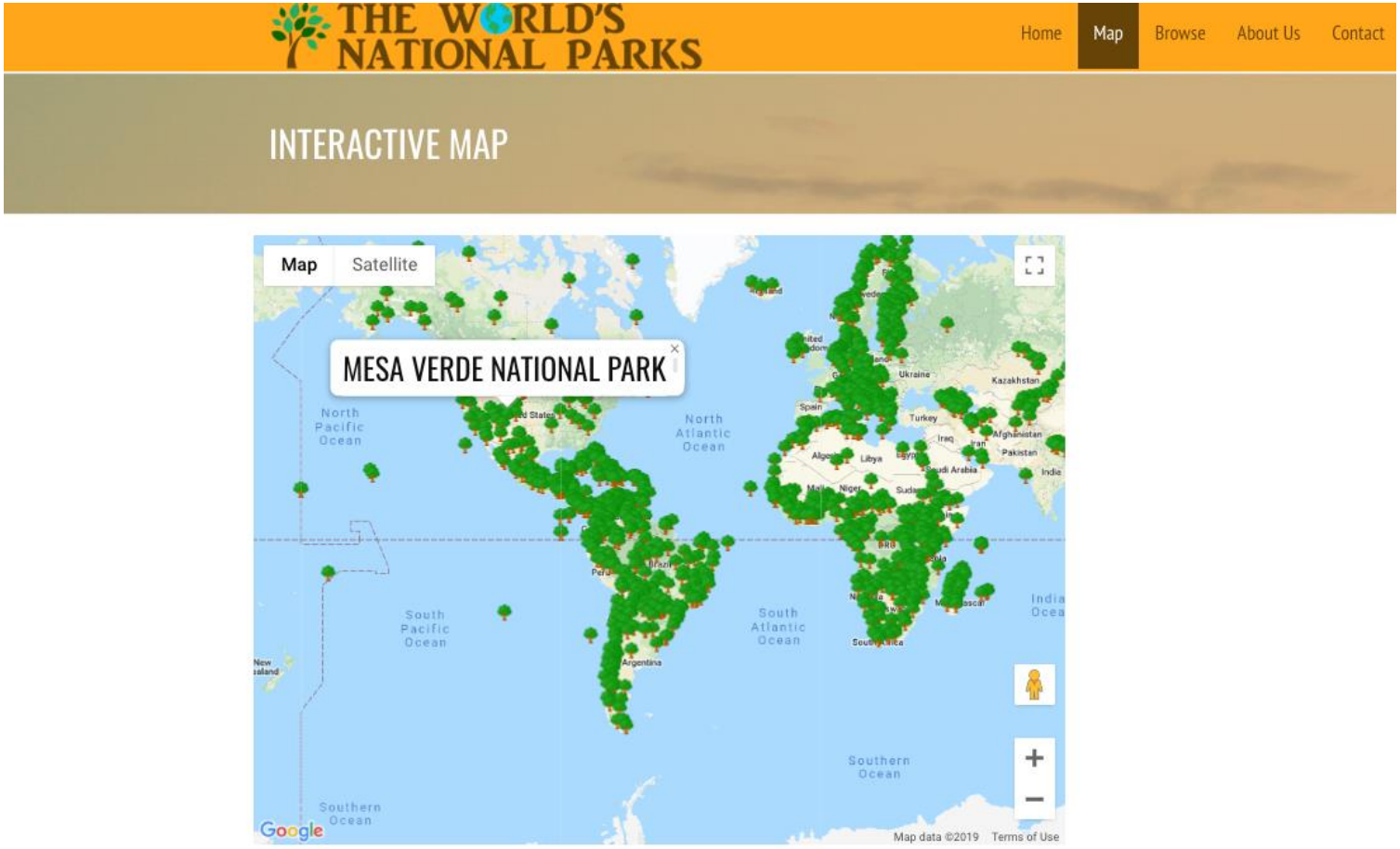

Figure 21

4.6 Mobile Application

The Mobile Application was developed using Android Studio and database was pulled directly from Firebase. DBeaver allowed to convert the information to JSON file, which was uploaded to Firebase. Mobile Application will have seven options to choose from on load page: Map of World's National Parks, Nearby, News, My List, Search by filter, Most Popular National Parks and Did You Know? (Interesting facts about National Parks).

Figure 22: Mobile App's start page. The following illustrations show the icons of the application, initial load page and the map app of all the National Parks.

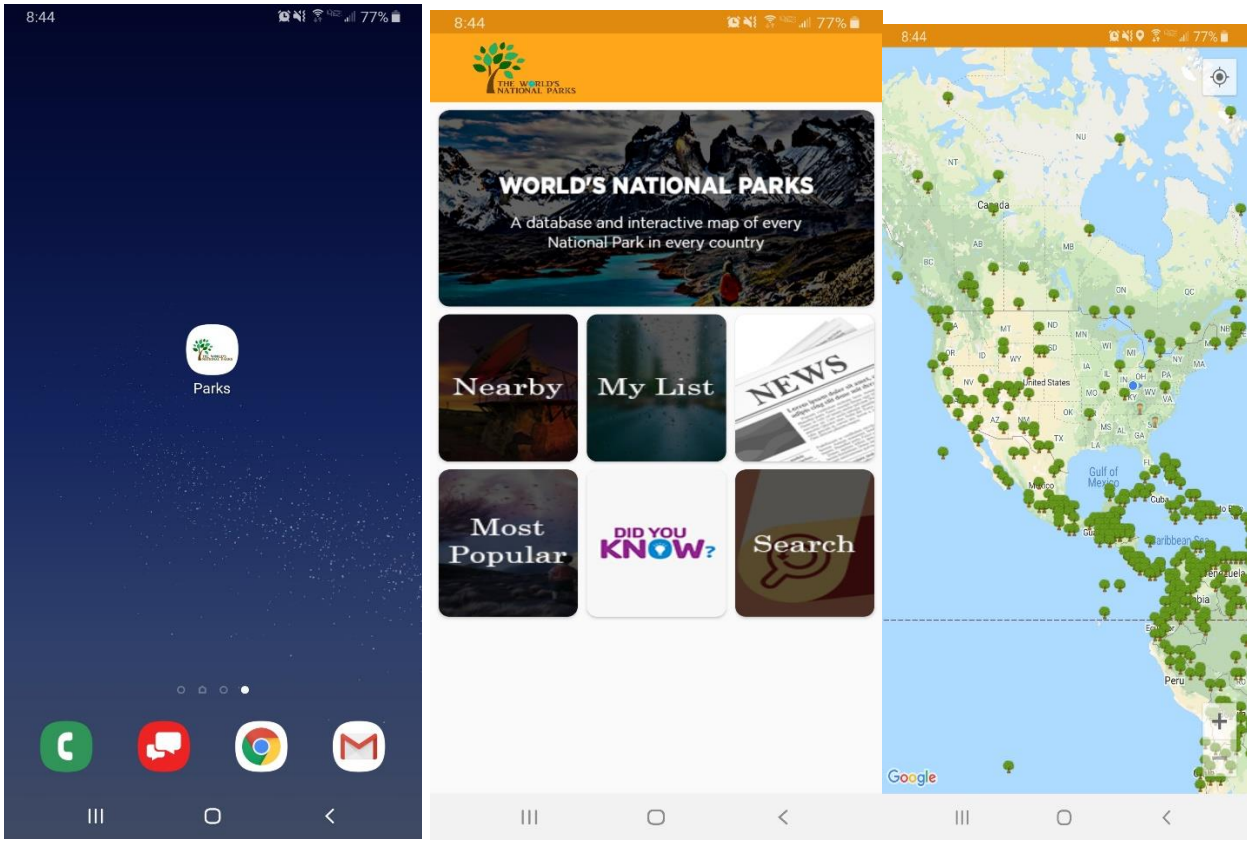

Figure 22

4.7 Initial Goals that had to be abandoned.

There were several goals we had to abandon since it required a longer time to develop and the Spring semester was not sufficient time or they were not important anymore. The following are the options and features that didn't get added to current version of the apps and database.

1. Option for users to login and leave comments on each National Parks they've visited on Web and Android Apps.
2. Option for users to upload images of National Parks on Web and Android Apps.
3. Develop iOS version of the Mobile App
4. Add short video link for each National Park in the database.
5. Add protection level on each National Park.
6. Create GIS file for each National Park that's visible on map.

4.8 Future Recommendations

Out team has decided to continue working on the apps and database as a side project. National Parks around the world get constantly changed by adding a new location or changing the status of current ones. Thus, this project will require continuous improvement of the database. While working on it as a side project, we are planning to implement the goals that didn't get added during the Fall and Spring semesters. Also, we will add features that Manual Tests suggested.

5. CONCLUSION

Our team was able to accomplish most of the goals that we set up for ourselves at the beginning of the project. We were able to create a unique database with 2700 National Parks' information that constantly gets updated. We've developed a Web Application that is registered at https://theworldsnationalparks.com/ and currently available. Our Mobile App is created for Android devices and we'll work on developing the app for iOS. Overall, this project was an interesting and challenging endeavor. During the process of creating our apps, we had chance to practice the Agile Development Process and get better understanding of the development process. Out team was able to work on each stage of the Software Development Life Cycle, where each member contributed to the success of this project. While creating the database, we learned skills on database creation, moderation, management, database integrity and security. Web and Mobile applications allowed us to practice on working with the Google Maps API, working with a relational database, practice UI and UX skills, initialize testing and overall working with N-tier architecture of software applications. All this was possible by collaboration and the teamwork attitude of our group.

REFERENCES

National Park Services. 2019 [Online]. Available:

https://www.nps.gov/grsm/learn/management/statistics.htm

Nicaragua 2019. [Online]. Available:

https://www.nicaragua.com/attractions/national-parks/

Lisa C. Chase, David R. Lee, William D. Schulze and Deborah J. "Ecotourism Demand and Differential Pricing of National Park Access in Costa Rica". *Land Economics*. Vol. 74, No. 4 (Nov, 1998), pp. 446-482