PenParks

by

Maxwell Vest & Sheldon Wheeler

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

Insert Your Name Here	Date 4/17/17
Insert Your Name Here	Date 4/17/2017
Insert Faculty Advisor's Name Here, Faculty Advisor	Date 4/17/17

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

April 2017

Table of Contents

Tables and Figures

Abstract

There are apps that cover national parks and let the user create their own experiences within them. However, there isn't an app as versatile, user friendly and more compatible then PenParks for local parks. With PenParks, you the user has ability to customize your own park experience. The user can discover the variety of local parks close to the them. The app lets the user craft their own experience by creating trails, exploring the different actives that are offered within it, and lets the user identify the various wildlife within the park. This app transforms the way a user looks at a park experience and helps educate them on the park itself. So, the next time you decide to discover a park use PenParks to become your own pathfinder, explorer and naturalist.

Introduction

There are a lot of parks and nature preserves in the United States. These parks range from huge iconic National Parks like Yellowstone to small neighborhood parks. We want to augment a user's park experience with the technology of a mobile app.

Project Description

To design a mobile application for experiencing local parks that will be available to the public. The application will enhance user's park experience. Allowing users to mark interesting sights and share them for others to see. Data on individual parks can be downloaded to the user's device.

Problem Statement

Most parks have limited information available about them. The average visitor doesn't have an outlet to add to that information for future visitors. Pen Parks wants to let visitors using the application add information to the park. This is information will be points of interest and animal sightings in the park. that can be shared so others can access it.

User Profile

The user will find a list of parks when opening the application. The user profile can be accessed which will send the user to their profile. This will contain options to view their uploaded data and data that has not been uploaded yet. If the user is not logged in it will redirect them to a screen to login with a Google account.

Application

Cross platform mobile app that will navigate experiences through the park.

Potential Customers

Customers will be people that want to explore every corner of a local or national park.

Software, Interface and Related Experience

This app will be constructed in a way to discover more information about a park. Therefore, the user will not need to have any technological experience when using the app. They will only need an understanding of how to login and save park information.

Experiences with Similar Apps

The process of using this app will be similar to accessing Google or creating accounts for other apps. If the user can follow instructions, the app will be easy to use for them.

Task Experience

The only task experience required for this app will be when the user wants to create trails or update account information.

Frequency of use

The user can use this app for a multitude of reasons from discovering parks to mapping out trails.

Key Interface Design Requirements that the Profile Suggests

We are using an interactive map that will let the user, find a desired park that they wish to customize.

Use Case Diagram

Figure 1 is a Use Case diagram showing the flow of a user's experience.

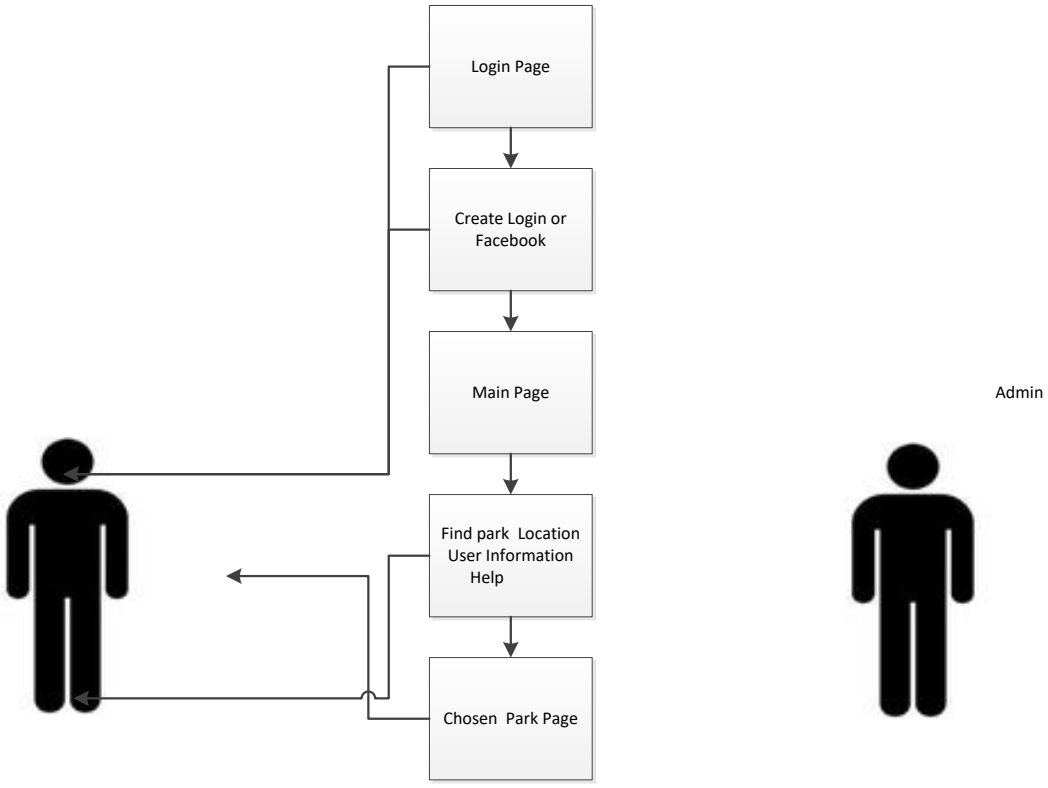

Figure 1: Use Case Diagram.

Budget

This budget is for if we are working for a company. We need to store all the park names with data about them so we chose to buy thirty servers to contain all of that data. We need to buy visual studios to use Xamarin. Finally, we can install the Xamarin software for free.

Number	Item	# of Units	Price Per Unit	Total Item Price(Rounded)
Software				
1	AWS SQL Server Express	1	0.00	$0.00
2	Microsoft Visual Studio	1	472.86	$473.00
3	AWS Web Hosting	1	0.00	0.00
Subtotal				$473.00

Figure 2: Budget

Gantt Chart

The following Gantt chart shows the timeline of the project. Figure 3 shows the various steps needed to complete the project. The chart is split into the three categories of planning, prototyping, and testing.

Task Name	Status	Start Date	End Date
Task Group 1 - Planning		**09/15/16**	**10/30/16**
Mapping Core Features		09/15/16	10/06/16
Mapping Extra Features		09/22/16	10/10/16
Research Technologies		09/22/16	10/10/16
Implementing Technologies		10/01/16	10/30/16
Contact Sponsor Brandon Jones		09/26/16	09/26/16
Researching Competition		09/15/16	09/20/16
Examine Scope of Audience		09/15/16	09/20/16
Buying Technologies		09/22/16	09/26/16
Task Group 2 - Prototyping		**09/22/16**	**11/30/16**
Building Infastructure		09/22/16	10/30/16
Programming App		09/30/16	11/30/16
Designing App		09/30/16	11/30/16
Preparing for Presentation		11/05/16	11/30/16
Task Group 3 - Testing		**11/03/16**	**11/30/16**
Testing App		11/03/16	11/30/16
Fix Bugs		11/08/16	11/30/16
Fulfill Requirements		11/03/16	11/30/16

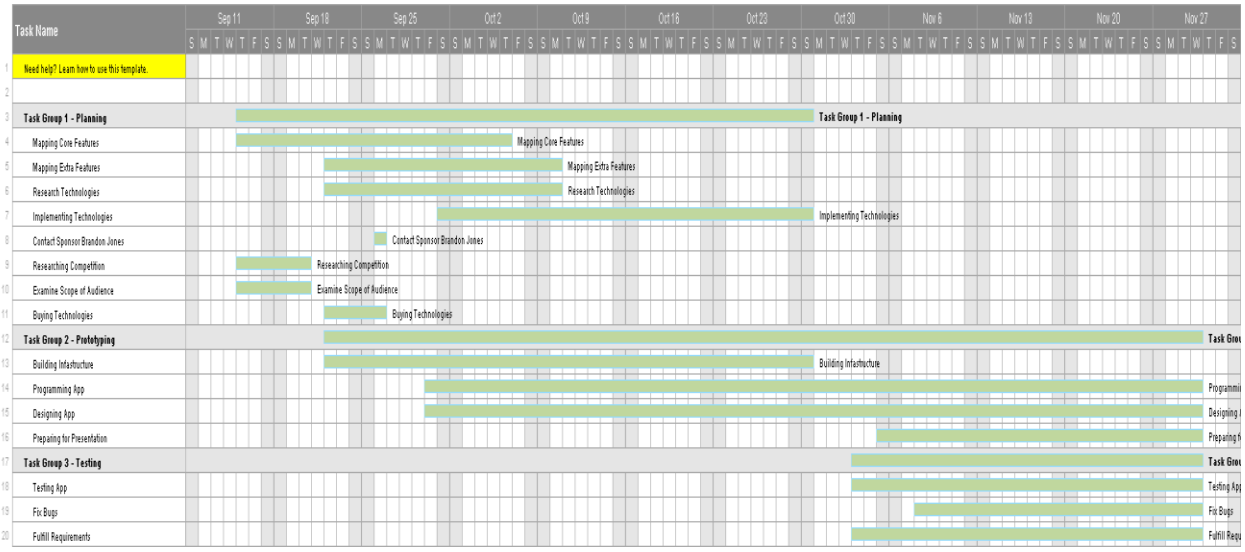

Figure 3: Gantt Chart Semester One

Gantt chart for the second semester of the project.

Task Name	Start Date	End Date
Task Group 4 - Adding More Features	**01/09/17**	**03/13/17**
Connecting API to external database	01/09/17	02/13/17
Creating Login Page	01/09/17	03/13/17
Trail Mapping Feature	02/05/17	03/13/17
Contact Sponsor Kevin Ghantous	01/28/17	01/29/17
Task Group 5 - Deliverables	**01/09/17**	**02/13/17**
Gantt chart	01/22/17	01/22/17
Testing Plan Reports	01/24/17	01/24/17
Final Abstract	01/22/17	02/13/17
Preparing for Presentation	01/09/17	02/13/17
Create Poster	03/06/17	03/06/17
Task Group 6 - Testing	**01/09/17**	**03/13/17**
Testing App	03/01/17	03/13/17
Fix Bugs	01/09/17	03/13/17
Collecting Park Data	03/01/17	03/13/17

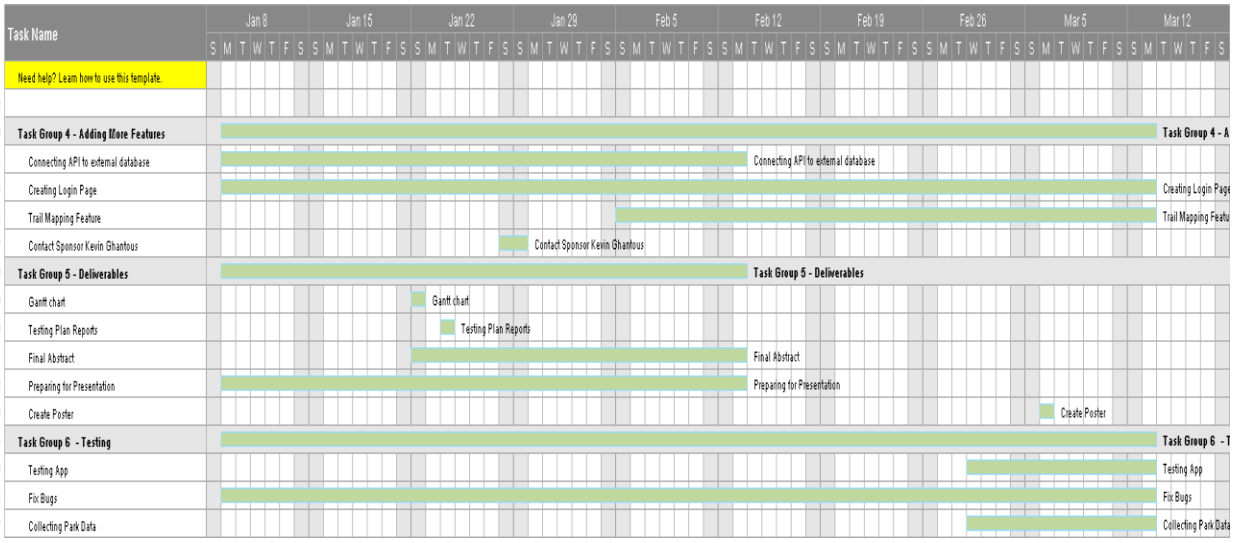

Figure 4: Gantt Chart Semester Two

Methods

The following sections will give details on the steps and technologies needed to build the project.

Building Project with Xamarin

The project will be created using the cross mobile platform framework Xamarin. From an article written by Ariel Ben Horesh, "Using Xamarin, developers can target Android, iOS and Windows (Mobile) 10, using a single industry standard language, C#. Certain aspects of the code base are platform specific, for example the UI layer, forcing programmers to develop them repeatedly for each platform." (Horesh 2016) So with Xamarin cross platform mobile development can require less coding by not having to entirely rewrite the application for each targeted platform. PenParks will specifically be developed with Xamarin Forms. Xamarin Forms allows for sharing of UI elements across targets. Very little platform specific code needs to be written using Xamarin Forms.

Managing Data

The project will require two databases. SQLite will be used to handle data on a user's device. This will allow the storage of a user's data without having to upload it to a remote server. It will also be used to make the application work more efficiently by downloading data from the central server to the SQLite database to reduce calls to the remote database. An SQLite database will also make offline usage possible.

A remote central server is also needed to be the repository for all data collected by users of PenParks. It will allow the users to receive the available data for a park on request. The server will be an SQL Server Express Edition. The database will be hosted by Amazon Web Services. Amazon Web Services will also be used to host the Web Service created for the project that is an

Asp.Net web service. Images are to be stored and accessed using Amazon Web Services S3, which is file storage service.

Managing Users

Logging into the application will be handled using Googles OAuth 2.0 API service. API credentials will be setup with Google and the user will logon with their own Google account to be authenticated. The app will only keep track of the user's authentication token received from Google and their Google User ID to attach to user created data.

Location Services

A major part of PenParks requires the use of GPS and Maps presented to the user. Location services will need to be accessed on devices, using a Xamarin plugin to access the location services. Maps will be rendered using Google Maps API. Google has APIs for maps and geocoding. The Maps API is free up to 20,000 calls every twenty-four hours.

Testing

This section will explain in detail how the testing methods for PenParks will be implemented. The testing language should be Xamarin (written in C#). Here is the individual that should use this testing method.

- Developers

Scope

The scope of the testing will involve the entire app. The major area of concentration will be securing the individuals information that they entered as an account. The testing will be organized in a way to determine the most important areas that have to have extensive security.

Objective

The objective of the testing is usability and stability. The whole app will be tested to ensure the app run smoothly. The developer will be the one to initiate the testing phase.

Testing Procedure

Here are the steps needed to complete the testing

- Create points of interest entries to test parts of the app

Entry and Exit Criteria

Entry Criteria

Build complete

Self-testing complete

Testing environment Is setup

Exit Criteria

All tests run

Bugs are documented

Logging Test and Reporting

When testing, if the situation of a bug comes up, the developer must document where they found it. After completing the test, the developer will determine if the bug is in an area that is being developed or a problem that will need a solution. If the developer finds that the bug is a problem, then he or she will fix that bug in order to ensure the app is secure.

System Testing

The testing method will compose of phases throughout the PenParks app. This will determine the issues that a bug can cause on a certain phase or if it will create a domino effect causing all phases to be defective.

Testing Procedure

The sequence of testing should follow a basic step

-create new testing scenarios and testing cases

This sequence must be completed in order to consider all of the testing a success for PenParks. If this sequence fails, then the developer will put the scenarios and cases that didn't work in a separate category.

Pass/Fail Conditions

If the scenarios and cases are successful for PenParks they will be put in a category that has all of the successful tests. If it works the opposite way, then the scenarios and cases will be put in the failed category.

Risks

The following risks that could potentially distract the testing phases

Data leakage

Lost Data

Time to fix bugs

Testing Results

Tested application in a real-world scenario on a Samsung Galaxy S6 phone. Data was collected at Burnet Woods and Mount Storm Park in Cincinnati, Ohio. Park entries were effectively collected and uploaded.

Bugs found were error handling issues related to lack of internet connectivity and image handling. Multiple images being loaded in the same session caused too much memory being used that needed to be cleared before exceeding the applications set heap size limit. This was solved with garbage collection being run before hitting that limit to free memory. Some issues still occur with internet connection problems.

Conclusion

PenParks is a cross-platform mobile application developed using Xamarin Forms. PenParks allows the user to augment their park experience by utilizing data uploaded by other users and the ability to create their own data and experience. The data will include animal sightings, recreational activities, and other points of interest. It also includes the ability to map trails within a park.

Works Cited

Horesh, Ariel Ben. "Build Cross-Platform Android and IOS UIs with Xamarin Forms." SitePoint. February 26, 2016. Accessed November 24, 2016. https://www.sitepoint.com/build-cross-platform-android-ios-uis-xamarin-forms/.

Advanced Xamarin Forms Apps. Produced by Housssem Dellai. Performed by Houssem Dellai. Youtube.com. January 3, 2017. Accessed February 12, 2017. https://www.youtube.com/playlist?list=PLpbcUe4chE7-uGCH1S0-qeuCWOMa2Tmam.

Login with Google in Xamarin Forms. Performed by Houssem Dellai. Youtube.com. September 10, 2016. Accessed March 15, 2017. https://www.youtube.com/watch?v=AgFIsVr26zg&index=3&list=PLpbcUe4chE79kfUAO_cSHZknJizvXodGQ.

Montemagno, James. "Jamesmontemagno/Xamarin.Plugins." GitHub. January 20, 2017. Accessed April 17, 2017. https://github.com/jamesmontemagno/Xamarin.Plugins.