AirCaddy – Final Report

Created by grant folgate, Eric byrd, madison wiecorzek, and ryan vero

2017

**Table of Contents: Pg. #**

1. **MODULE: ORGANIZATION DESCRIPTION**
   1. Organization Description
      1. Mission, Objectives, Industry 4
      2. Information System Culture of Organization 4
   2. Overview of Business 5
   3. Business Area Analysis of the Unit 5
2. **MODULE: PROJECT DEFINITION**
   1. General Application Goal
      1. Original Proposal 6
      2. Project Goal 6
      3. Environment and Boundaries 7
   2. Statement of Work
      1. Objectives of the Project 7
      2. Success Criteria 7
      3. Risk Analysis 7,8
3. **MODULE: PROCESS MODELLING**
   1. General Application Goal 8
   2. Statement of Work 9,10
4. **MODULE: DATA MODELING**
   1. Initial ERD for the Application 11
   2. Data Dictionary 12-14
5. **MODULE: DEVELOPMENT STRUCTURE**
   1. Work Breakdown Structure 15,16
   2. Planned Development Platform and Required Resources 17
6. **MODULE: TIME FRAME for IMPLEMENTATION**
   1. Timeline Plan – Gantt Chart 18
   2. Activity Estimates 18,19
   3. Division of Labor Plan 19,20
7. **MODULE: APPENDICIES**

7.1 Terms 20

**8 Exhibits**

MODULE 1 – ORGANIZATION DESCRIPTION

* 1. Organization Description
     1. Mission, Objectives, Industry

AirCaddy provides a concentrated hub for golf advice and golf related reviews for the beginner to intermediate golfer and participating golf courses across the country. The project will provide a superior outreach for the showcased course. The application will offer a form of advertisement for an audience in search of an appealing golf course including appeal that is relative to course popularity (gathered via user reviews), landscape (drone footage previews), or geographical location. Additionally, application users can expand their knowledge of the game and other users fine-tuned for each course showcased in the project. Knowledge becomes more precise in all aspects of the game; both technical and social. This application, with the user of drone technology, provides a platform for drone industry growth. The current marketplace does not have any web applications that provide drone footage of golf courses that breaks them down into individual holes and provide user comments on each of the holes. This system will help users who are unfamiliar with a golf course and wish to get a better feel for it by virtually walking through each of the holes to find trouble spots and obstacles that are not immediately visible form a scorecard of aerial view.

* + 1. Information Systems Culture of the Organization

Elk Valley Golf Course, the organization that our team will be collaborating with initially, is not a technology-based company rather a golf course.

* 1. Overview Description of the Organization Business Unit

As mentioned in our proposal, we are deciding to have three types of users that pertain to the business for our application. The first type of user is a general user that does not have an account with our application. This person will be able to view golf course ratings, see drone footage of each hole on the golf course, and see what other people have to say about each hole on a course. The second type of user is one who has created an account with our application. This type of user can do everything a non-authenticated user can do, but he or she will also be able to create comments on each hole of a course, and up and down vote comments. The last type of user will be one that is able to post golf courses and drone footage for each hole of that course. To gain this type of access, an administrator of the application will have to approve them to do so.

* 1. Business Area Analysis of the Unit

The current high-level information systems architecture layout of the system highly involves the Microsoft .NET Environment. Since this is a web application using .NET technologies, we will be using a suitable server that has Microsoft Windows Server 2012 or greater. The application itself will be hosted on the server using IIS (Internet Information Services). The application will interact with a relational database, we plan on using Microsoft SQL Server 2012 or greater for our implementation.

The application itself will be using the ASP.NET framework. The backend will be developed in C#, and the front end will be developed in HTML, CSS, and JavaScript. Some potential frameworks that may help the front-end development include Knockout.js or AngularJS to help with creating a smoother, more maintainable front-end.

To store our videos that will be captured by the drones, we will use YouTube’s API to interact with our channel. These videos will be stored on YouTube’s servers.

In the future, if this business continues to grow, we will be looking at moving to cloud deployment using Microsoft Azure. This way our database, and application will not have to be internally managed by us. The down time can be taken care of and reduced by Microsoft themselves.

MODULE 2 – PROJECT DEFINITION

* 1. **General Application Goal**

2.1.01 Original Proposal

There are various golf based application areas that are not established yet and for which new ideas and models are presently under development. For this project, the primary user is assumed to have some interest in the sport of golf and the strategic process of the game itself. Many avid golfers currently experience difficulties when navigating a new course. They have to rely on a map at each tee box with minimal information on each hole. A golfer using this application can gain a valuable perspective in terms of blind spots, depth perception, and a feel for the terrain they are about to encounter.

2.1.02 Project Goal

Provide an application that gives the user a preview of the course hole with a drone view to improve overall golf experience and allow for a social media platform for users to share feedback on course/hole experiences.

2.1.03 Environment and Boundaries

The application environment will be primarily at the golf course of choice by the user, although the application would work anywhere that an internet connection is available, either through cell data or through Wi-Fi.

* 1. **Statement of Work**

***2.2.01 Objectives of the Project***

Our objective for this project is after a year of being on the market, we plan to have a growing user rate of 20% quarterly and increase the number of golf courses available in the application by 55% by the end of the year.

***2.2.02 Success Criteria***

How many users of the application and how often the users actually use the application will determine success. Usage will be tracked through footage view counters and commented reviews. The most important success factor for this project is solidifying a functionality that does not exist in any other golf application. The unique feature in our golf application is the use of state of the art drone footage that will be provided at each hole of the course that will be broadcasted in our application. Externally, the more courses and users that apply to add footage and comments on our application, the greater the change of market growth.

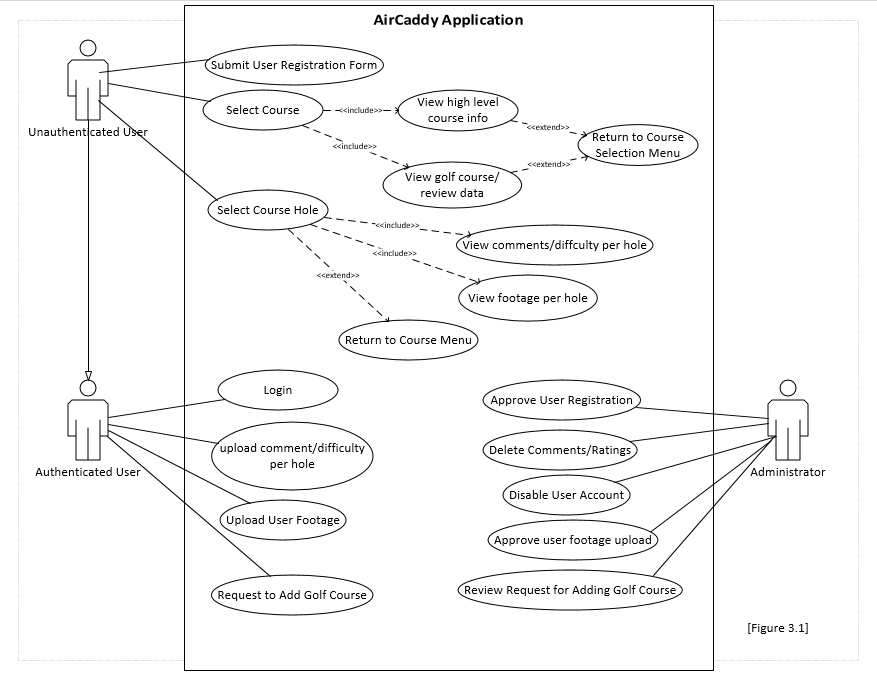
***2.2.03 Risk Analysis***

There will be some marketplace competition including the following existing applications 18Birdies, GolfNow, and Ping. These applications do not provide the in-person view or depth charts but they do offer an aerial view of each hole and the yardage to the hole; this could be enough information for some users. User acceptance can be another issue because some users will want to use the application exclusively while they are on the course; there is no guarantee that the golf course area will have adequate 4G coverage to access the application. The highest risk we have is staff illness, since we have a small team of developers, if any member became ill this could potentially setback the delivery date.

MODULE 3 – PROCESS MODELING

* 1. ***General Application Goal***

To help better describe this section, we will provide a user story diagram to give a better picture of how the users correspond to the functionalities offered by the system. With each of these functionalities, a system sequence diagram will be provided to better explain the flow of how the system will respond as seen below in **Figure 3.1**.



* 1. **Statement of Work**

In this section we will define all our listed functionality described in the user story diagram. In each functional requirement we will describe the inputs that go into them, the outputs, and the flow of data for each of them.

1. ***Request ability to be able to manage course footage ( [Exhibit 1] for more details)***

The authorized type of user will fill out a form that will require them to provide the name of the course they are trying to upload for, their phone number, their email, their name, and justification on why they would like to upload course footage. By doing this we can reduce the risk of bad video data being uploaded to our application. This information will be available for an administrator to approve. The outputs of this functional requirement will either be a rejection and the user will be unable to upload course footage or they will be granted authorization to upload video footage of their course.

1. ***See golf course review data ( [Exhibit 2] for more details)***

Any type of user will be able to view course review data that each course uploaded to the application may or may not have. To generate this data, once a course is uploaded to the application; our application will use the Yelp Fusion API to gather high level review data about the course which will then be displayed to the user to give better insight to them on whether it is worth exploring that course in more detail.

1. ***See high level course information ( [Exhibit 3] for more details)***

Any type of user will be able to view high level course information that includes the operational hours of the golf course, address, and the phone number. To generate this data, once a course is uploaded to the application; our application will use the Yelp Fusion API to gather general business information which will then be displayed to the user which will allow the user to contact the course owner if they have more questions.

1. ***See footage per hole of golf course ( [Exhibit 4] for more details)***

Unauthorized and authorized users will want to have this capability to give them better insight on what a particular hole is like on that golf course. To do this, the user will click on a golf course and select a hole they would like to see. To generate the video for the user to see, our application will use YouTube’s API to show the particular hole on the particular course the user would like to view.

1. ***See difficulty/comments per hole ( [Exhibit 5] for more details)***

Any type of user can view this data and it will be generated by authenticated users to post what they think about that course by giving it a difficulty rating and providing any useful feedback that could help the user. This type of data will be obtained from the application’s database which is a Microsoft SQL Server instance.

1. ***Upload difficulty/comments per hole ( [Exhibit 6] for more details)***

Authorized users will have privilege to do this. This type of user will fill out a form that requires them to give the hole a difficulty rating (1 being very easy, and 10 being very difficult) and some general comments about the hole. This information will be served in our application’s Microsoft SQL Server database instance.

1. ***Approve authorized users to upload course footage and become Golf Course Owner type user ( [Exhibit 7] for more details)***

An administrator will be able to review all of these types of requests and have the ability to reject or accept that type of user to do those kinds of actions. This information will be obtained from our application’s Microsoft SQL Server database instance.

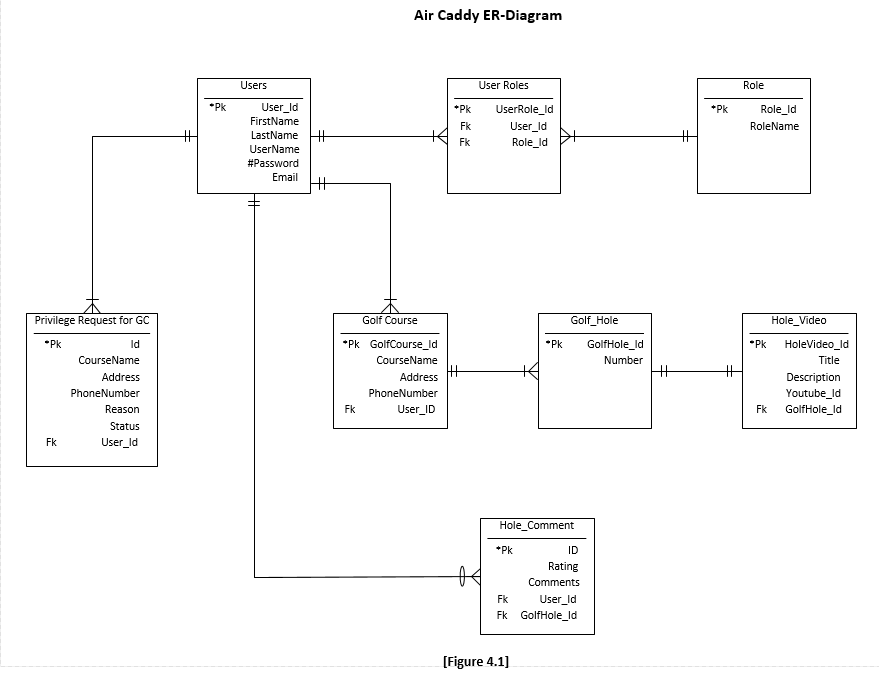
1. ***Manage course footage for their course ( [Exhibit 8] for more details)***

A Golf Course Owner type of user will be able to upload footage per course simply by clicking on the golf course they own, selecting a hole, and then uploading a video file. The metadata of the file will be stored in our application’s Microsoft SQL Server database instance (File name, YouTube Id to obtain from YouTube), and the video itself will be stored on YouTube utilizing their API.

MODULE 4 – DATA MODELING

4.1 Initial ERD for the Application

- See **[Figure 4.1]** listed below



4.2 Data Dictionary

- See **[Figure 4.2]** listed below

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table | Field Name | Data Type | Data Format | Field Size | Description | Example |
| User | User\_Id | Integer | NNNNNN | 6 | Unique number to associate with a user | 123456 |
| User | First Name | Text |  | 15 | First name for user | Jim |
| User | Last Name | Text |  | 25 | Last name for user | Smith |
| User | Username | Text |  | 15 | Username for user | JimBob101 |
| User | Password | Hashed Text (NVARCHAR(  MAX)) |  | 256 | SHA-1 Hashed Password |  |
| User | Email | Text |  | 30 | Email unique to user | [text@text.com](mailto:text@text.com) |
| User Roles | UserRole\_Id | Integer | NNNNNN | 6 | Unique number to associate each entry in user roles table | 123456 |
| Roles | Role\_Id | Integer | NNNNNN | 6 | Unique number to associate user roles with each entry in table | 123456 |
| Role | Role Name | Text |  | 10 | Role to outline user privileges | Admin, Verified, Regular, Owner |
| Golf Course | GoldCourse\_Id | Integer | NNNNNNNN | 8 | Unique number to identify approved golf course | 12345678 |
| Golf Course | Course\_Name | Text |  | 20 | Name of golf course | Elk Valley Golf Course |
| Golf Course | Address | Text |  | 30 | Street number, street name, city, state, zip code of golf course | 123 Green Rd Erie, Pa, 16509 |
| Golf Course | PhoneNumber | Integer | 7249998877 | 10 | Phone number of golf course | 8172737263 |
| Golf Hole | GolfHole\_Id | Integer | NNNN | 4 | Unique number to identify holes on courses | 1234 |
| Golf Hole | Number | Integer | 1-18 | 2 | Number associated with hole on the course | 7 |
| Hole Video | HoleVideo\_Id | Integer | NNNNNN | 6 | Unique number to identify footage of a hole | 134567 |
| Hole Video | Title | Text |  | 20 | Title of footage for each hole | Hole 9 Footage |
| Hole Video | Description | Text |  | 120 | Description of footage |  |
| Hole Video | YouTube\_Id | Integer |  | Unknown | Unique identifier connecting to Youtube | Unknown |
| Request | Reason | Text |  | 120 | Reason for application of golf course involvement |  |
| Request | Status | Text |  | 10 | Status of gold course involvement | Approved/Pending/Denied |
| Hole\_Comment | HoleComment\_Id | Integer | NNNNNN | 6 | Unique number associate with comment on a hole | 123456 |
| Hole\_Comment | Rating | Integer | 1-10 | 2 | Rating of hole based on footage | 9 |
| Hole\_Comment | Comments | Text |  | 120 | Users comment on holes in terms of difficulty |  |

MODULE 5 – DEVELOPMENT STRUCTURE

**5.1 Work Breakdown Structure**

|  |  |
| --- | --- |
| **Request ability to be able to manage course footage** | * Create web form for administrator approval * Provide UI to upload video |
| **See golf course review data** | * Create Yelp Fusion API Client * Create UI to show golf course review rating * On UI show user’s reviews |
| **See high level course information** | * Display course information from Yelp API |
| **See footage per hole of golf course**  **See difficulty/comments per hole** | * Display footage of selected hole from YouTube API * Display user comments from SQL database |
| **Upload difficulty/comments per hole** | * Display form to user, information collected will be stored in SQL database |
| **Approve authorized users to upload course footage and become Golf Course Owner type user** | * Retrieve user requests from SQL database, approve or deny |
| **Manage course footage for their course** | * Display form, ability to edit Database information and overwrite video file per hole of their own course. |
| **User login, registration, and roles** | * Need to implement registration, login, and roles logic before any further logic can be developed |

**Work Process Breakdown**

Our team will be following the SCRUM development life cycle. And will be working in two to three week iterations. Each iteration will consist of a sprint planning meeting, a sprint review meeting, and a sprint retrospective meeting.

During the sprint planning meeting, the technical and business team will meet and review our sprint backlog which will contain all of the user stories with an associated point value in them. During this meeting the business and technical team will come to an agreement on which product backlog items need to be worked on. The technical team will then work on the items chosen and will get those done before the sprint review meeting at the end of the iteration.

At the sprint review meeting, both the technical team and business team will meet and go over the work that has been done during the iteration. If the business team thinks what has been developed suffices, then those items in the backlog will be marked as complete. However, if the business team does not like what has been developed during the iteration, then those items will be worked on again in the next iteration. A new set of product backlog items will also be added for the technical team to develop.

The sprint retrospective meeting will take place immediately after the sprint review meeting, the technical team will discuss ways to improve the efficiency within the team and make sure everyone is on the same page before further work is done in the next iteration.

During each iteration the technical team will have bi-weekly stand ups to discuss any road blocks team members are dealing with, and check in to see general progress of the iteration.

**5.2 Planned Development Platform and Required Resources**

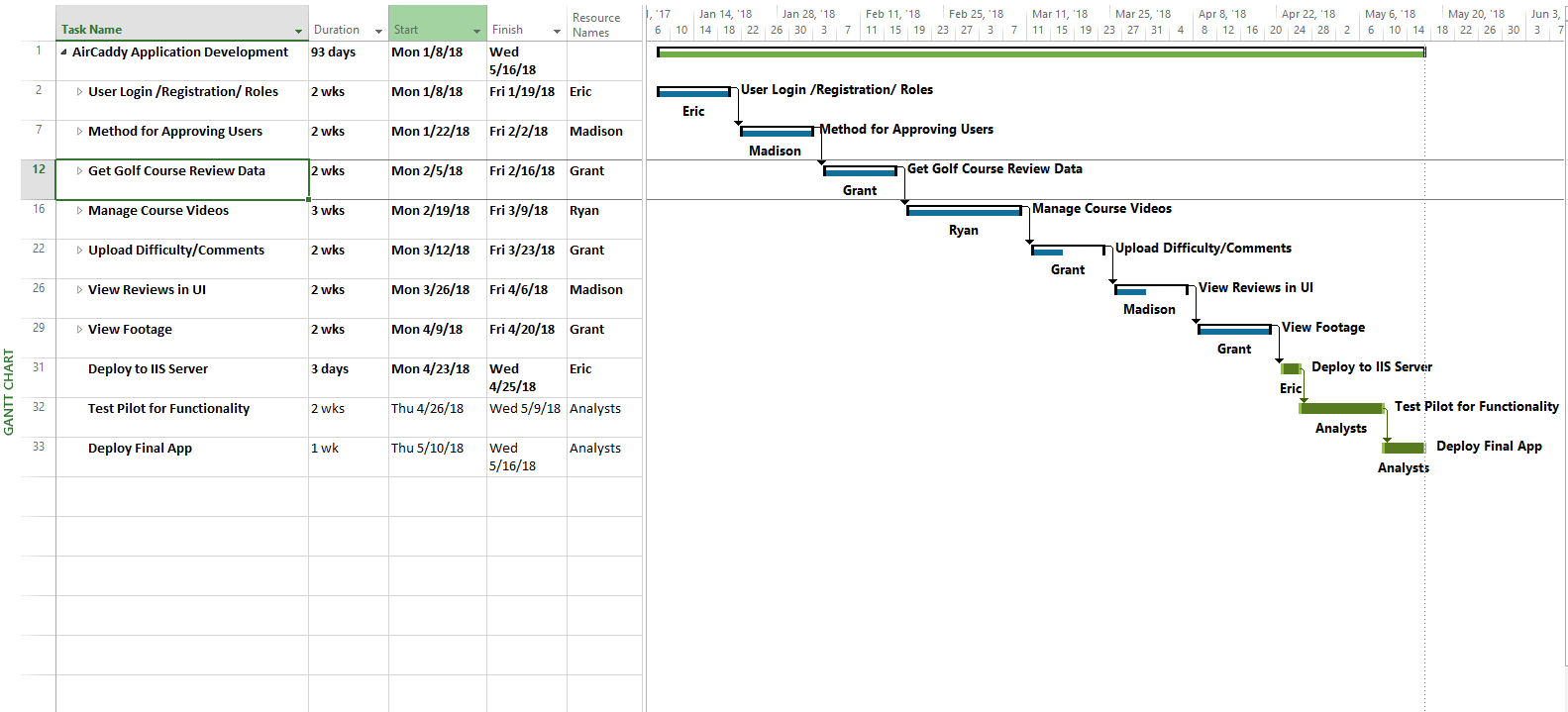
For this application we plan on developing in the .NET Environment in which we will specifically be using ASP.NET to develop this web application. The backend which will be providing the logic to interact with our database and external web services will be created in the C# programming language. Our front end which will be used to display data and allow the user to interact with our application will be made using HTML, CSS, and JavaScript. To help speed up the CSS development we will be using the Bootstrap framework to provide a readily available set of styling to our application. To help provide a more robust user interface we will be using Knockout.JS which will help us provide two-way databinding.

Our application will also require several more services to run correctly. A SQL Server database instance which will store our data necessary for the application to run. Our application will also interact with two external web services; one of which being the Yelp Fusion API which will provide our application with review data for each golf course entered into the system; the second web service we will be using is the YouTube API which will allow us to upload our videos to our private YouTube channel and allow us to play these videos within our application.

Lastly, this application will be deployed as a prototype on one of the servers held at Gannon that has at least Windows Server 2012 on it. On that server, we will need Internet Information Services enabled on it and it will need it to support version 4.5 of .NET or higher so our application can run correctly on the server. One of the servers that is a potential for our group to use is the pendragon server that is maintained by Professor Cannell and Professor Coffman. Another potential server that can be available for us to use as a prototype is one of the servers maintained by ITS. Carol Kugler is a point of contact for this.

MODULE 6 – DEVELOPMENT STRUCTURE

* 1. Timeline Plan – Gantt Chart
* See **[Figure 6.1]** below for details



* 1. Activity Estimates

There are eight major work items that will need to get done over the course of the semester and there will be about 2 minor work items that will need to get done. In this section, more detail will be discussed about the work items. The first two things that will need to get done will be the creation of the user roles and authentication functionality; once that is finished a form in which users who would like to create footage for a specific golf course will be created. These two items will require about four weeks of work. It is important that these two things get done first is because any work related to uploading footage for a golf course, getting golf course review data, having user comments and difficulty ratings per hole of the golf course can not be done unless there are user roles added to our system that allow the golf course data to enter our system. Our team has decided that it will be harder to implement user roles later into the application when it is more developed, so we would like to get that done first while the application is very loosely coupled. The next wave of features will take about six weeks to get complete and tested. These features are the bulk of the application and will allow users to upload golf course footage per hole, post comments and difficulty ratings of a golf hole, and get golf course review data from Yelp. The reason that these features will take this long is because most of our team does not have experience with writing code to interact with the YouTube and Yelp API’s. The last wave of features includes a UI to view golf course footage, a UI to view user ratings and comments about a hole, and a UI to see golf course review data. This set of features should take around four weeks because the amount of testing that goes into these features will be less intensive.

Once the major work items have been completed, the next two will be more straightforward and will require little to no development work. The first work item that will need to get accomplished is deploying our application to a test server here at Gannon as a prototype which will take about a week. The second work item that will need to get accomplished will be an initial first wave of testing in which we will write and perform integration tests for our system. This testing will take around two weeks to complete.

* 1. Division of Labor Plan

In the beginning of the development of the application. We will be doing mostly pair programming with Grant Folgate taking the lead. Since most of the members on this team are inexperienced with .NET development, pair programming will get them up to speed. Once the developers on the team get comfortable with the coding standards, roles will be assigned to each of the developers. The target timeframe for this is by the first week of February. Grant Folgate will specialize in the front-end development of the app but will assist in the back end development for more complicated tasks such as the development of the code to interact with both YouTube’s API and Yelp’s API. Madison Wiecorzek and Eric Byrd will do most of the back-end development that interacts with our relational database. Eric has experience in troubleshooting and fixing technical problems as well. This will be advantageous to eliminate downtime and keep the development team efficient. Ryan Vero will assist Madison and Eric in the back-end but will also help me with more complicated front-end development such as the views to display the footage for the golf course and having a view to show golf course hole ratings and comments.

This division can be further explained by viewing the Gantt Chart listed in **[Figure 6.1]** of section 6.1**.**

MODULE 7 – APPENDICIES

7.1 Terms

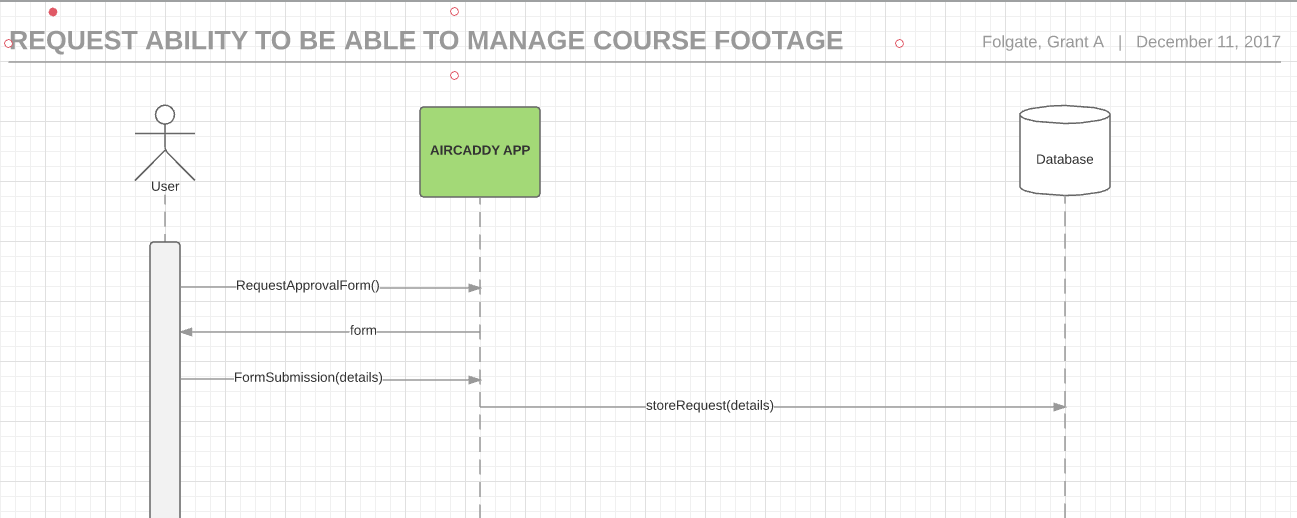
*API* – Application Programming Interface

*Front-end* – Pertaining to what the user sees in the application

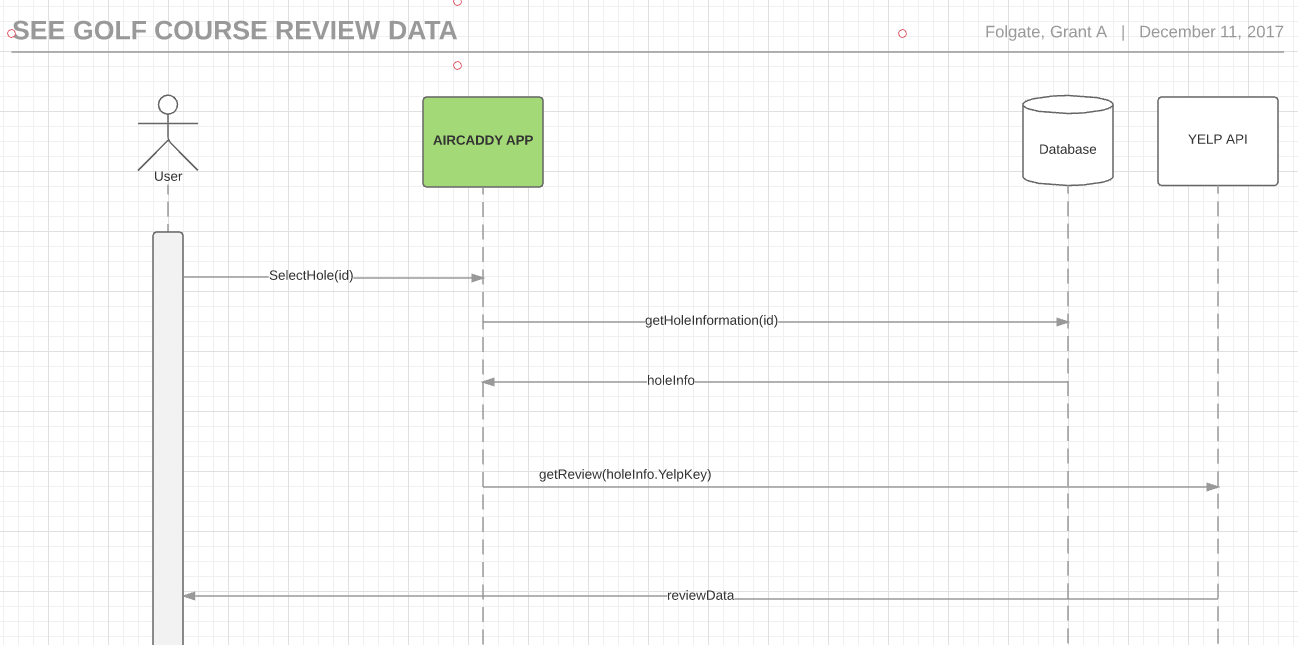
*Back-end* – Pertaining to the business logic that happens behind the scenes when a user performs an action on a page

*Two-way databinding*- Binds html properties to data elements in a particular JavaScript object.

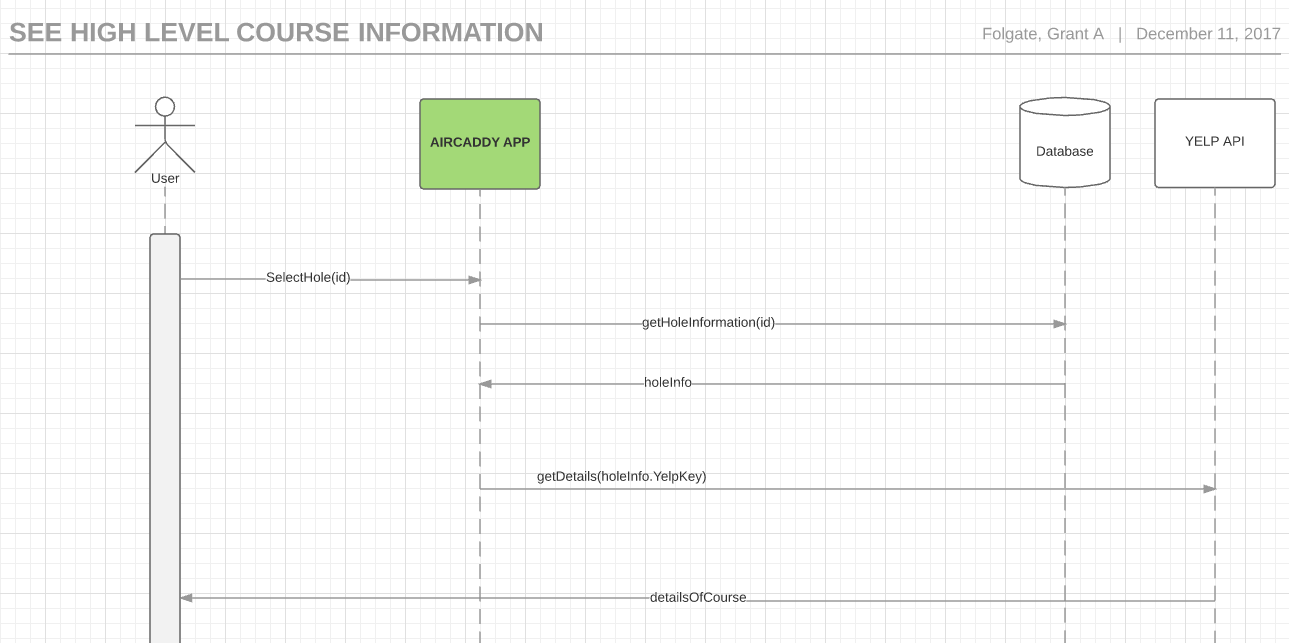
**[Exhibit 1]**

****

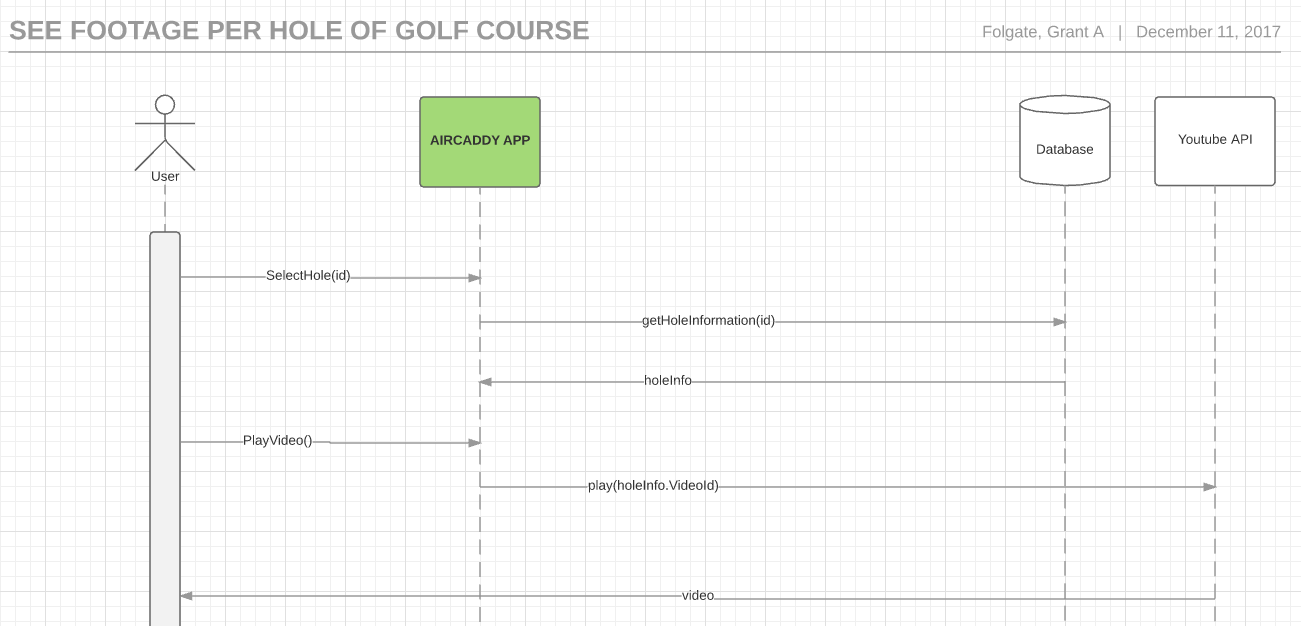
**[Exhibit 2]**

****

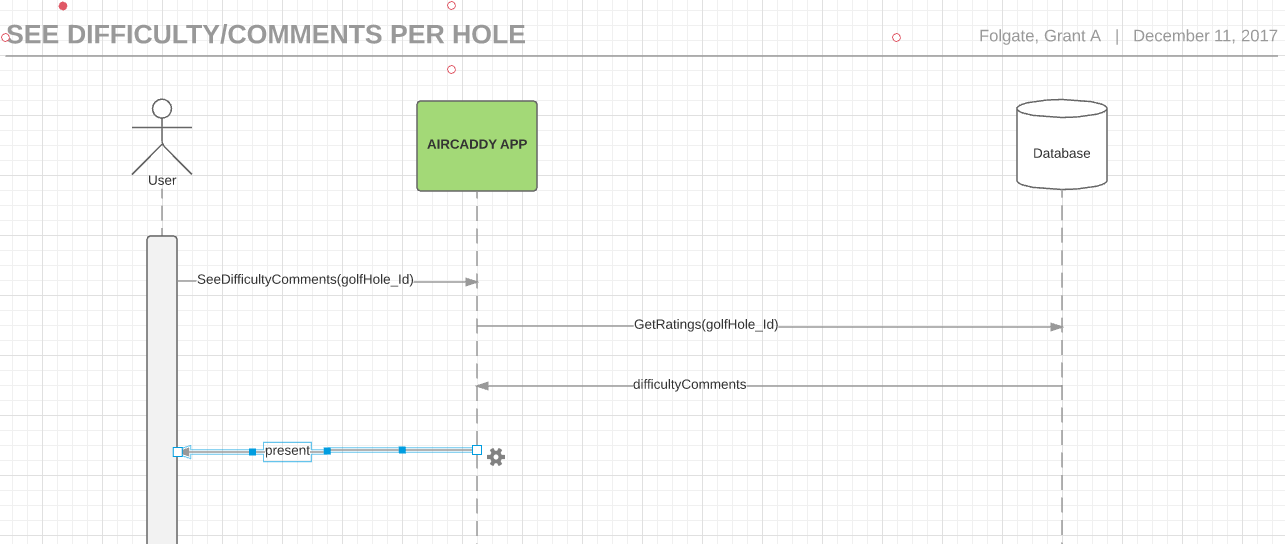
**[Exhibit 3]**

****

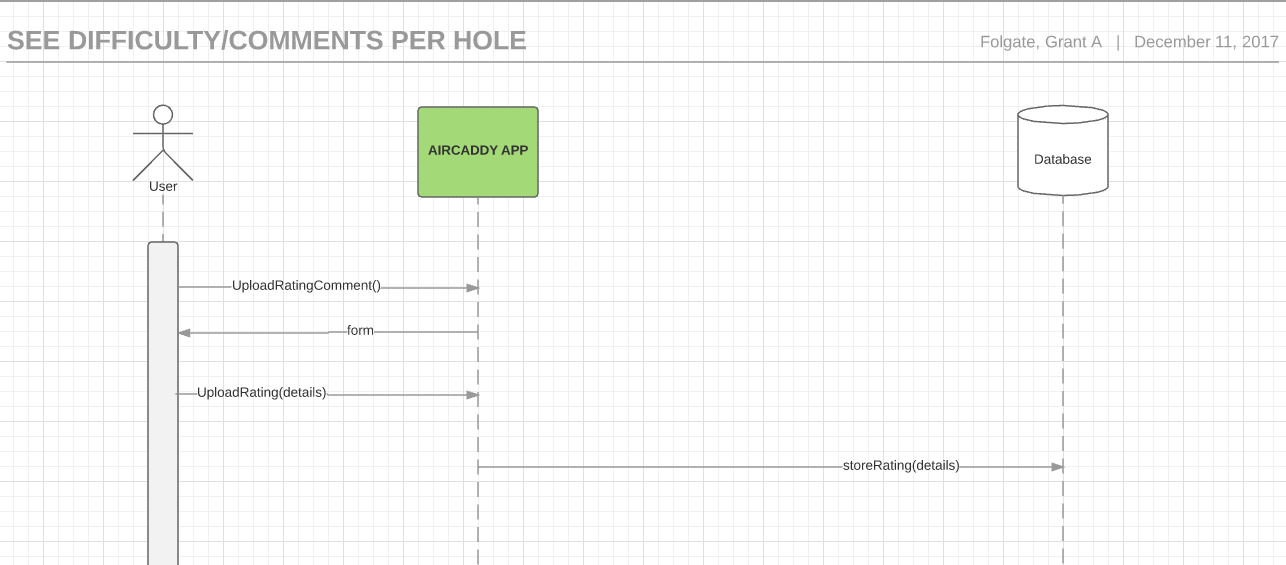
**[Exhibit 4]**

****

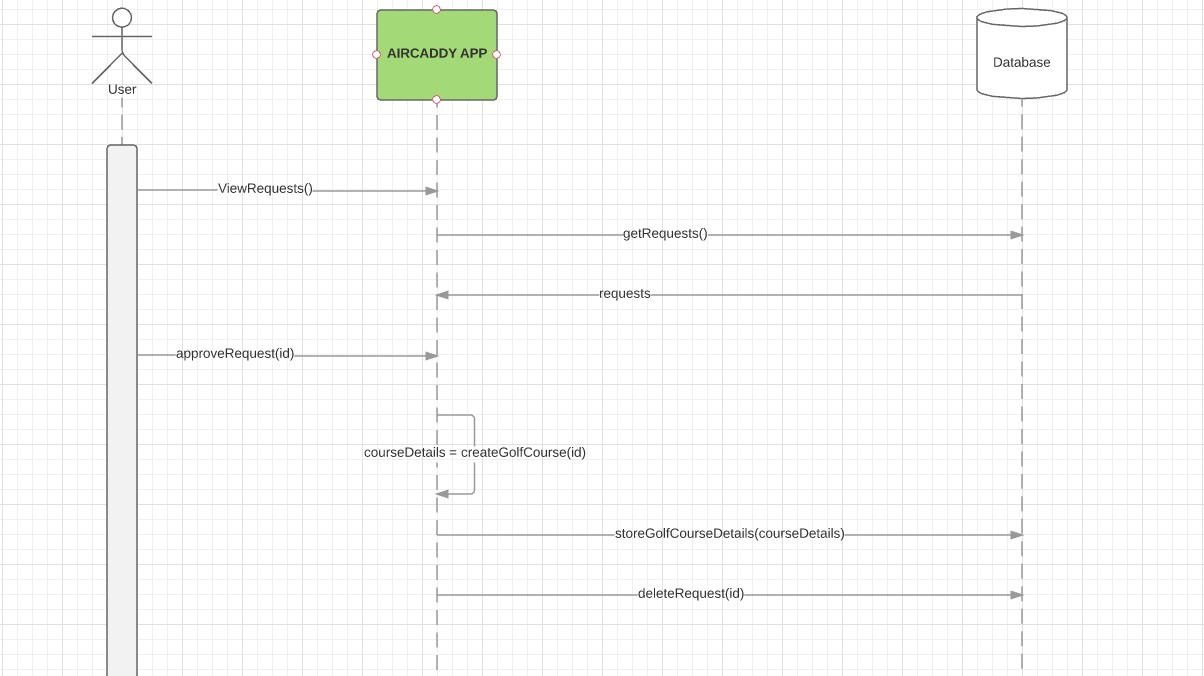
**[Exhibit 5]**

****

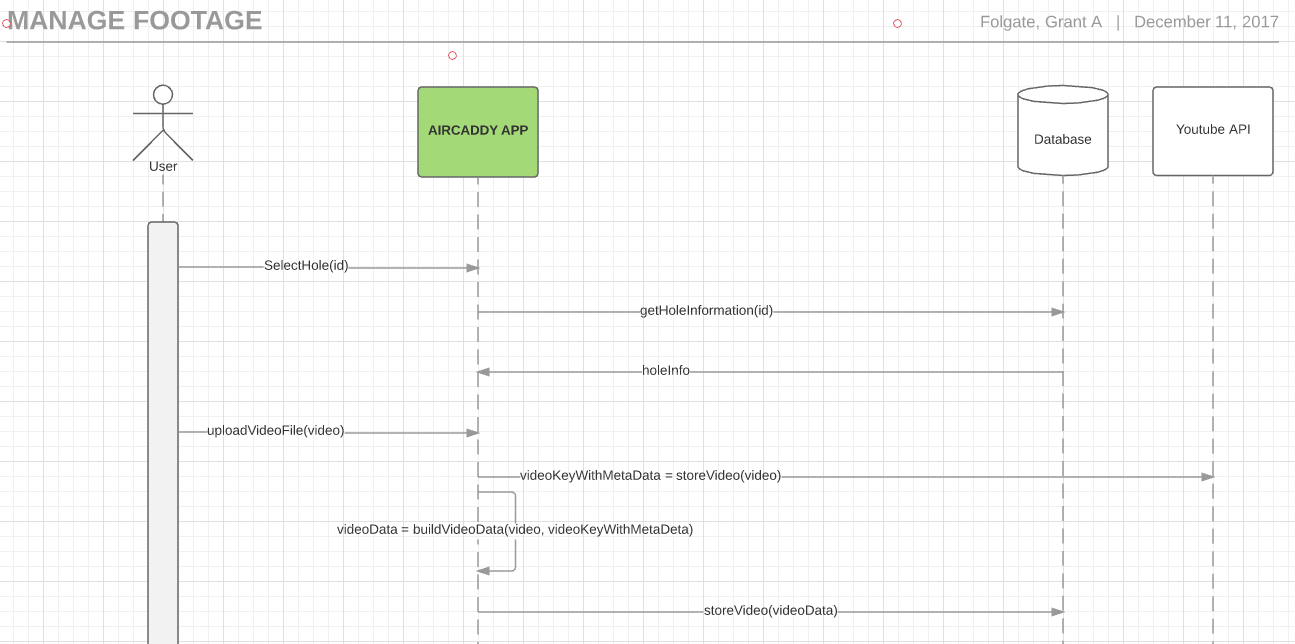
**[Exhibit 6]**

****

**[Exhibit 7]**

****

**[Exhibit 8]**

****