Soufan Weather Station

Soufan Weather Station Mobile App

Majd Soufan

December 11, 2017

Project sponsor and advisor Dr. Hwang

ABSTRACT

Soufan Weather Station is a complete weather system that can be installed almost everywhere (houses, farms, research facilities, etc). This project forms the font-end of the Soufan weather station, which is the mobile application. The essence of this product is to allow the user to interact with the weather station and analyze the resulting weather data. A designated server will receive the data from the station and send it to the mobile application utilizing an API. The mobile application will present the data to the user in various ways. Also, the mobile app will provide access to previous data that has been saved on the server. This application could be very beneficial for researchers and farmers to keep track of weather data.
ACKNOWLEDGMENTS

This project would have never come to reality without the help of Dr. Hwang throughout the semester. She always insisted on meeting with me to keep track of my progress. She supported me and directed me on the right path especially when I was behind.

LIST OF FIGURES

1. Figure 1: System Overview Architecture
2. Figure 2: Front-End
3. Figure 3: First data presentation method
4. Figure 4: Second data presentation method
5. Figure 5: Splash Screen
6. Figure 6: Login Screen
7. Figure 7: User View
8. Figure 8: Results

INTRODUCTION

Making people's lives easier is always the purpose of software engineering. In order to fulfill this purpose, software engineers develop software of all kinds like social media applications, weather applications, device cleaning applications, etc. However, every piece of software is built to solve a problem or to modify an existing solution. An excellent example is Skype. Skype is a software application that allows users from all over the world to make phone and video calls. Therefore, Skype solved a connection problem among people from all over the world. Likewise, a weather station application could be very beneficial for many people.

Keeping track of weather data is an essential issue to a lot of people. Therefore, users could implement a personal weather station to track the weather data. The overall project is a
personal weather station that fetches weather parameters and sends them to a mobile application. The mobile app presents the weather data to the user. This report is related to the front-end of this system, which is a mobile application that receives the data from a server and displays it to the user. The server communicates with the mobile app through an API. The mobile application can utilize the weather information to create weather analysis and display it in various user-friendly manners.

I. PROBLEM STATEMENT

Thousands of people consider weather as a basic factor in determining their daily duties. For more than a decade, people have been using newspapers, the radio, and the Internet as their main resources for collecting data about the weather in their city. The problems with these resources are inaccuracy and inconsistency. For example, if the consumer needs to find how the weather is going to be the following day in Chicago, the information from these kinds of resources could indicate a possibility of rain in the city. Due to the enormous size of Chicago, however, this information may not apply to the entire city. Therefore, the weather information could be false. People need to gain access to accurate weather data easily, and the traditional methods of receiving weather data are not very efficient in a world that is ruled by speed and technology.

The two main categories of people that need to access weather data regularly are farmers and researchers. Generally, farmers check the weather on a daily basis because they have to create their own daily reports to measure the productivity. Therefore, farmers are in a deep need to access accurate weather information associated with their fields. Nevertheless, normally farmers usually do not have a high level of education, so they will prefer to work with simple and efficient tools. Simplicity is not a feature of most traditional methods of retrieving personal weather status.
On the other hand, researchers that utilize an exposed environment in their experiments are also involved in this problem. Normally, people who work in research facilities or experimental laboratories highly consider the weather as a main factor in their experiments, especially if the experiment is being operated in an exposed environment. In these cases, every piece of information regarding the weather is very important, and it is included in the study. Therefore, new methods should be created to fulfil the drawbacks of the traditional systems that are being used.

II. REQUIREMENTS AND SPECIFICATIONS

One of the most essential parts of this project is the mobile application. It is the link between the user and the station. The importance of this mobile app prevails in the ease of using it. Due to this era of advanced technology, almost everyone possesses a smartphone nowadays. Therefore, the idea of developing such an application on a daily used platform is very beneficial to the project. Moreover, this mobile software presents the data in several ways and styles. This feature adds more professionalism to the project and make it more efficient, especially for researchers.

The audience of this project will have different levels of education. People like farmers, most likely, are not highly educated. Therefore, using a complicated mobile application to access their weather data can demonstrate a real problem for them. In order to overcome such a problem, this mobile application is simple and easy to navigate. Moreover, by simplifying this product, the application can get more attention and more success. The application receives the data from a server maintained by the weather station. Finally, the application should always be connected to the server in order to receive data.
III. DESIGN

People can have their own weather stations in their own backyards. This weather station is supported with all kinds of sensors that receive and send weather data. This project forms only the front-end part of the Soufan weather station, which is the mobile application. The mobile app presents this data in a very user-friendly method utilizing a mobile application. Building a weather station with a connected mobile application would provide people with valuable personal information. The value of the mobile application is contained in the ease of accessing the weather station at the specific location of a weather station for example, at a consumer's house. People can check all kinds of weather data around their houses by using a mobile application on their personal cell phones. Moreover, this application saves the data for a long time, and it also analyzes all the previous data. This product is also supported with an online server that maximizes the storage capacity. Plus, the mobile app has a very appealing design that is simple and efficient at the same time.

The overall architecture of this project will provide some sort of API to the weather station server to receive the data. The back-end part of the project will call certain API functions that perform some actions like increasing and decreasing the sensing range of the weather station. It has a user interface side that interacts with the user and presents the data that is coming from the server. This organization is very dynamic, and it requires a checking system that handles all the connection errors. Moreover, the mobile application includes a security system that prevents hackers and frauds. The circled part of Figure 1 represents this project, which is the mobile application that interacts with the Soufan Weather. Figure 2 shows a brief overview about the mobile application, the user interface and the service it supports, which is displaying weather data.
In order to fulfill the specifications listed above, the project is required to use certain technologies that can accomplish these requirements. First, to develop an iOS mobile application in an efficient way, the application will be written in C# using the Xamarin framework. The
Xamarin framework is a new method to build an iOS application by utilizing C# for writing the back-end code for it. This technology will save time while writing code and will make the job easier. Moreover, Xamarin uses the native mobile APIs for iOS platform. Therefore, creating UI controllers on iOS is natural and simple. For the front-end, the development will be accomplished by using the native programming language for iOS which is Swift and the Xcode storyboard. In order to represent the data in different appealing ways, the mobile application contained multiple pages and have access to the native APIs that iOS provides. The SQL server is a very crucial element in this project. The SQL server will host the database, and it interacts with both the weather station and the mobile application. MySQL can be used as the main RDBMS for this server. It is efficient and very useful. So, finding resources can be easy for such RDBMS.

This mobile application is divided into two main parts:

1. Front-end: The ways and methods of displaying the data.
2. Back-end: using the API, it fetches data from the server and delivers it to the front-end

In general, every mobile application contains back-end and front-end parts. However, the back-end side of this project is programmed by using only C#. The back-end side essentially calls the API that interacts with the weather station server. The SQL web server is the data resource. All the weather data that has been collected by the physical weather station will be sent to the SQL server. The information is also stored on the server for long periods of time. However, the back-end calls the API that provides all the methods and properties that contribute in two main functionalities, which are receiving and analyzing the data.

Receiving the data requires some specific C# libraries or frameworks like the .NET. Using the .NET framework allows the application to connect to a specific server. After
connecting the program with the server, the application can receive data from it. As mentioned above, the data is stored in a SQL database on the web server. Therefore, the back-end can send and receive SQL queries to the API. Furthermore, it must include database libraries that can convert the C# code into SQL queries and fire them. However, receiving data from the web server depends on the user’s input. Therefore, the API will read the input from the front-end side, then will implement the right code with the right SQL queries. In this part, the system does a continuous checking on the connection between the server and the application.

The second function of the API will be analyzing the received data. At this point, the user will be expecting some way of presenting their weather data. Therefore, the API interacts with the front-end part of this mobile application. The API is responsible for analyzing the data. For example, the user can request data from the past month, but they would only want to include specific types and dates. Therefore, the back-end, by calling the API, must receive the data from the SQL server and process it, then deliver it to the front-end part. The interaction between the front-end and back-end is contained in a way that the back-end will deliver the data that has been received from the server. For example, the data structure that is used in this part of the program is a list of points, if the data representation method was a chart. However, the data structure is different throughout the project due to the diversity that the front-end contains.

The Xamarin technology provides several ways to convert the C# code from the back-end side into native code. The API delivers the analyzed data to the front-end code, and then the front-end code processes the data and convert it into bit maps or other kind of pixel representation. Visual Studio can be used as an IDE for any Xamarin application. Therefore, all the features that the Visual Studio provides will add to the development process of the mobile application. One of the features that Visual Studio supports is debugging. Visual Studio has one
of the most powerful debugging tools. This feature will save time and help the development and testing process a lot. Another feature that Visual Studio provides is the Live Synch Emulator for iPhone, which can test and examine the front-end of the project in a continuous manner. Therefore, the developer can make changes in the code that can affect the visual views, and the emulator will catch these changes and implement them while the program is running. The Visual Studio is a very beneficial tool especially in developing software applications that contains front-end parts.

The iOS main programming language is Swift programming language. As mentioned above, the data is presented in different manners. One of the presentation method of the data is a bar chart. The way to convert the data into bitmaps or any other pixel representation is being done by the front-end part of this project. The API delivers the data to the front-end and then Swift code will convert it into visual views that the user can interact with.

Developing in Swift and Xcode storyboard gives full permission and control over the iPhone hardware and software. Therefore, by developing in Swift the application has access to native APIs. Plus, it could easily implement native iOS UI controls.

One critical element in this project is data representation methods. This feature is embedded in the front-end side of this project. In order to increase the quality of this product, the data is being displayed in several ways. The back-end, utilizing some API functions, analyzes the data and sends it to the front-end views as a list of points. The views convert this collection of points into bitmaps or any other type of pixels, and present it on the screen. The data figuration will be presented in charts like in Figure 3, in diagrams like in Figure 4. Here are some examples of data representation:
**Figure 3: Data Representation 1**

**Figure 4: Data Representation 3**
One of the most essential part of this mobile application is the simplicity of the design. As mentioned above, the target audience of this application is farmers and researchers. Therefore, the users’ educational level will vary, and people like farmers will need a simple practical mobile application that will help them in getting access to their weather data. Down below is a prototype of the app’s design with navigational indication. Figure 5 shows the splash screen for this app. Figure 6 presents a simple log-in page. Follow on, Figure 7 includes an initial prototype to the main page and the data page.

Figure 5: Splash Screen
Figure 6: Login Page

Figure 7: Application Overview
One of the most important features that this project will implement is log-in system. Every user has a unique user ID and a password that connects them to their own personal accounts. Users accounts are created utilizing the back-end part of the project. The profiles are being sent to the SQL server by calling the API database functions, and stored on the server. When the user logs in to their account the front-end passes the information to the back-end, which checks if the data is associated with the profiles that are saved on the server. This checking process requires firing some SQL queries and sending them to the server. This product implements a reliable security system. This system is a part of the back-end portion of this project. The system covers all the user-login common problems. It checks for multiple failed logins and all other security issues related to it. Plus, this system covers all the database constraints and keys. The security system increases the reliability of this product and make it more practical.
RESULTS

At the end, developing the mobile application in Xamarin provided the front-end part of this mobile application with several methods of representing data, and that was one of the most important specification of this project. In addition, the mobile application’s design satisfies the simplicity requirements. Therefore, the number of users with low educational level will increase.

Finally, the back-end to this program interacts with an API that includes a collection of functions. These functions are responsible for communicating with the server to complete certain methods. This entire project, with these supported features, is able to provide daily weather report to the user in various presentation. It also allows the user to gain easy access to their old weather information and keep track of it. This system as whole unit can overcome all the drawbacks that the traditional weather tracking systems have.

Figure 8: Results
IV. CONCLUSION

After I completed this project and finished working on the Soufan Weather Station Mobile app, I clearly gained significant knowledge in mobile application and C#. Hopefully, this experience will help me in my future career and the upcoming opportunities. However, for any future work that can be done on the Soufan Weather Station Mobile app; the app can obtain an analyzing part that can compare the data from different time periods. Also, it can generate xml files that can be compared and analyzed. Some other idea could be adding more weather data type to the software so that the application could have more potential.
References

   <https://arqspin.com/tag/api/>

   <https://clipartfest.com/categories/view/f3215aa4cb7b88dab94aae26dc2fd4a0e57af773/web-server-clipart.html>

   <http://madisonbeckerman.com/Weather-App>

   <http://www.clipartkid.com/cell-phone-cartoons-cliparts/>


   Acurite. Web. 29 Apr. 2017
BIOGRAPHY

My name is Majd Soufan. I am computer science student at UE. My passion about computer started in high school when I decided to know more about video games and how they are created. I was very determined on studying computer science at college and on programming in general. I have continued chasing my dream throughout the university. Therefore, I have obtained several software engineering internships during my college life and finally secured a full-time job as a software engineer in Indianapolis. Hopefully, I will try to continue my education on a master’s level in the upcoming years.