

InfraDrone Android Application

DESIGN DOCUMENT

Team Number 07

InfraDrone

Mitra, Simanta

Evan Snitkey, Blake Agey, Yangxiao Wang, David Schmadeke

sdmay18-07@iastate.edu

<https://sdmay18-07.sd.ece.iastate.edu>

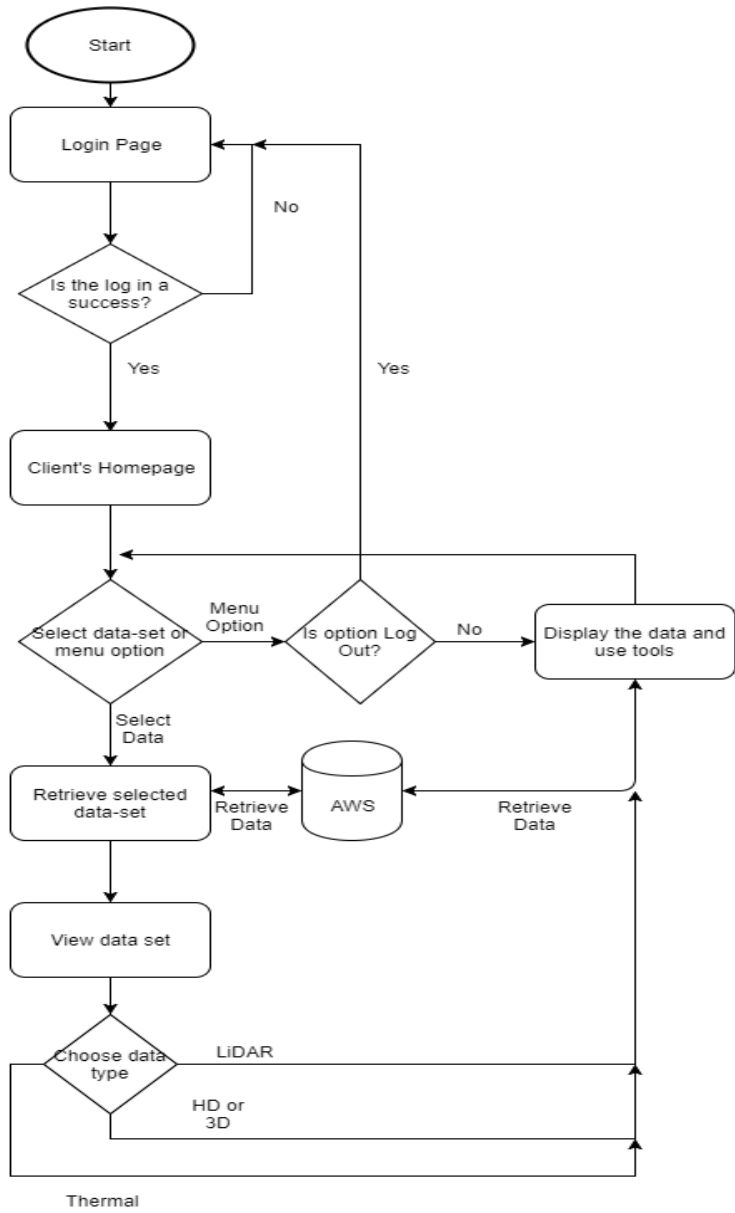
Revised: 12/3/17 Version 2

Table of Contents

1 Introduction	3
1.1 Acknowledgement	3
1.2 Problem and Project Statement	3
1.3 operational Environment	3
1.4 Intended Users and uses	4
1.5 Assumptions and Limitations	4
1.6 Expected End Product and Deliverables	4
2. Specifications and Analysis	5
2.1 Proposed Design	5
2.2 Design Analysis	5
3. Testing and Implementation	6
3.1 Interface Specifications	7
3.2 Hardware and software	7
3.3 Process	7
3.4 Results	7
4 Closing Material	8
4.1 Conclusion	8
4.2 References	8

List of figures

Figure 1



List of symbols

AWS - Amazon Web Services: <https://aws.amazon.com/>

VR - Virtual reality

OBJ- object file

MTL- object material file

LiDAR- Light Detection and Ranging

1 Introduction

1.1 ACKNOWLEDGEMENT

This project was created and supported by InfraDrone and its employees. We would especially like to thank Akash Vidyadharan, Samuel Eue, Rakesh Chowdari Maddineni, and Mitra Simanta for taking the time to help guide us and give us advice on this project.

1.2 PROBLEM AND PROJECT STATEMENT

Drones are becoming more advanced and capable. Some drones are capable of carrying heavy equipment such as high tech imaging devices that can collect data from hard to reach places.. The data collected then needs to be easily accessible, especially from mobile devices so that it can be viewed effectively. The purpose of this project is to design a mobile application for InfraDrone that will quickly, and easily display collected data from one of their drones. We will also incorporate VR into our application as another option to view data. This application will be designed around InfraDrone's already existing web portal.

The solution that our team has chosen is to use an Integrated Development Environment (IDE) called Android Studio to design an application to display collected data from InfraDrone. We plan to use the customers already existing credentials to collect the correct data from Amazon Web Services and display it to the user. Once we have the application correctly pulling and displaying data we will use Xamarin to develop cross platform functionality. Our application will be designed around InfraDrone's current website.

1.3 OPERATIONAL ENVIRONMENT

The end product will be an Android mobile application, so it will be used inside of a phone that could be used anywhere. The physical and weather environmental effects aren't necessarily useful for designing the product because it does not consist of any hardware. Because the product is completely software, other operational concerns exist; such as security and usability. In terms of security, we will be using the same design that the current existing web portal uses. This means that if there were to be a security issue, it would not be due to the application, it would be due to the Amazon AWS service. For usability, we will be conducting in depth tests to ensure that our application runs smoothly on any current Android device.

1.4 INTENDED USERS AND USES

This application will be used by any users that are currently using the InfraDrone web portal application. This audience consists of any InfraDrone employees and any of its clients. The application will be used exactly the same way as the currently existing web portal. The only differences between the web portal and new Android application are screen scalability and VR functionality (on the Android platform only). The main use for this application that differs from the desktop client is portability. Users can easily view structures and validate integrity via VR within the application on site. The point of this is to be able to view a structure from a view that one would not be able to by themselves.

1.5 ASSUMPTIONS AND LIMITATIONS

Assumptions

1. The security and database connections will work the same way as the current web portal structure. All usernames and passwords will be identical.
2. The design of the application will be identical to the web portal. This includes navigation, user interface, and functionality of the interface.
3. All client data will be pulled from Amazon AWS. This includes all different types of images and 3D objects.

Limitations

1. The application will only be expected to run optimally on current mobile devices. We will not be developing for older devices.
2. Android is the only platform that we are required to finish development on. iOS and Windows mobile platforms are stretch goals.
3. If we do implement cross-platform development, Android will be the only platform to include VR functionality.

1.6 EXPECTED END PRODUCT AND DELIVERABLES

Deliverables

1. November 21st (Demo 1) - First demo of the Android mobile application. Expected deliverables for this date include basic GUI layout representing the current web portal application and successful data links to Amazon web services for data collection and client login functionality.

2. December 5th (Demo 2) - End of first semester presentation. Second demo of the Android mobile application. Expected deliverables for this date include fully functioning Android application that works the same way as the current web portal. In depth code coverage tests are a requirement for this due date as well.
3. May 2018 (Demo 3) - Commercialized end product. This includes everything so far as well as fully functioning VR implemented into the application for 3D object viewing.

2. Specifications and Analysis

2.1 PROPOSED DESIGN

1. IDE and Platform
In this project, we will create a cross platform application that works on Android and IOS devices. Therefore, the initial approach will be using some cross platform development tools to develop Android and IOS app simultaneously.
2. UI Design
Once we have decided what we will be using for cross platform development, we will start working on the initial UI design. For this project, our client required us to make the mobile app similar with their webapp. Therefore, we will basically copy their webapp design to the mobile app.
3. Backend
After the UI design process, we will start backend development. Our client is using Amazon Web Service (AWS) to store their data and user information. Then we will also create our backend frameworks based on AWS.
4. Data Viewing
When the backend development process has been finished, we will implement the feature that user can access their data and view their data in our app. There are multiple types of data: images, PDF files, 3D models. For the 3D models, the client provided a demo that how SketchFab works and how their web app implement this feature. We plan to use some existing library/api to create view for the 3D models.
5. VR
At the final stage for the first semester, we should have a working application without VR, and start the development of VR. We plan to use some existing library/api to parse the 3D object and display it with Google VR SDK.

2.2 DESIGN ANALYSIS

1. After we did some research and obtained some advice from our client. We decided to use Xamarin to develop our application for IOS and Android simultaneously. We followed some tutorial to create a simple app by using Xamarin. However, we ran into some problem starting our own app.
 - a. Strengths
Xamarin provided cross platform development based on the one language C#. That means we can just use the same IDE, language, and APIs for IOS, Android, and Windows phone apps.
 - b. Weaknesses
The most difficult part for us is the different language and platform. Xamarin is using C# and all of our team member are familiar with Java instead of C#. And Xamarin does not work as we expected. We thought we can write only one app for multiple platform, however, we need to create IOS and Android app individually. Although there are some parts like UI and backend can be shared.
 - c. Results
Also we do not have the access to the Apple Developer Program, we cannot test any IOS related things. With the various problems and difficulties, we decided to create an working Android app first. As result, we are developing our app on Android Studio with Java.
2. We planed copy the web app's UI to our mobile app as the client required. When we jumped into the UI design stage, we realized that we cannot just copy the design from web app. The mobile devices have limited screen compared to a computer. We may redesign the UI and keep the web app's color and logo design.
 - a. Strengths
It can save a lot time for us to have an existing UI design, we do not have to start over from scratch.
 - b. Weaknesses
We will have many restrictions and limitations to match the web app's design.
 - c. Results
After re-thinking the UI design, we decided to use navigation bar to replace the section button on the web app.
3. The backend development is a tough task. There were many problems we faced and solved. The most important problem would be that we do not have direct access to client's AWS account. We created our own AWS account for testing. But

the client does not have the same setting as we do. To solve this problem, we contacted our client and asked them the settings they did on their AWS account.

- a. Strengths
Using the AWS is more convenient than deploy our own server and database.
- b. Weaknesses
Our team do not have the experience with AWS. And the AWS is already set up for client's web app. And we do not have direct access to client's AWS account.
- c. Results
Although our clients changed the structure of their data storage during our development process, we successfully pull the data and create our own data structure for the app.

4. Data viewing and VR

- a. Strengths
External library helped us parse the files and display it on mobile devices.
- b. Weaknesses
We should start this part earlier with sample data instead of waiting for completion of backend.
- c. Results
We did finish the PDF and image part of data viewing, but we had some problem with parsing 3D object files, especially mtl files. Our VR implementation had same problem, to display 3D object in VR, we also need the parsing function which is similar to 3D object parsing. Our plan is that try to make the external library work, otherwise, we would create our own algorithm for parsing mtl files.

3. Testing and Implementation

3.1 INTERFACE SPECIFICATIONS

We are currently developing on the Android platform. To program in this platform, we are using the IDE Android Studio. The project will also be using Amazon Web Services and its S3 product for object storage. This interface is used as a database storage that we can pull from and download objects, pdfs, and other types of files. We also use an API called Rajawali to parse object files and their respective material files. We also have an API called ModelViewer-o8o, which is what we use to create a model for the

object file and load it into Google VR. To create a basic VR component, we used Google VR API in Java for Android.

3.2 HARDWARE AND SOFTWARE

The software being used for the testing phase is the Android emulator that comes with Android Studios. This is useful for testing because it allows us to view the UI design and if they are aesthetically pleasing. It also allows us to run through the different parts of our project to check how the different components are working, and if any new components create errors or bugs in themselves or previously built components.

The two pieces of hardware that have been used in the testing phase is Android phones and a Google Cardboard. The Android phone is useful because it allows us to test how the UI design looks and use integrative testing on the various components of our project. It also confirms that our touch controls working correctly, unlike the emulator. On the other hand, the Google Cardboard allows us to take our VR part of the application and render the view to appear as a 3D object. We then tested how the object reacts with to the Cardboard with different views, head tilts, and positions.

3.3 PROCESS

We are currently developing the product using an agile based project management model called Scrum. This management style allows us to separate milestones into small deadlines and split those among the team members to do in short sprints that lead up to the completion of a long sprint. We are doing weekly short sprints where we will meet with the clients to end each sprint and start the next. We also meet with our faculty advisor once monthly, unless a team member has a problem and none of the team knows how to solve it.

To test the UI Design, we examined the application's pages for various phones using the emulator of Android Studios. We also will test the tilt views to ensure that each part of the UI fit the screen and does not mess with other parts of the design. Testing the back-end involved making sure that the data comes to the front-end correctly and in a relatively fast time. We also have to implement and test some sort of security so that hackers cannot use our queries in the back-end to access data that does not belong to them. Developers tested sample data in the database then ensure that it is properly reflected on the front end.

To test the VR and data viewing of components of the application, we tested VR with a simple object file as a proof of concept for next semester. We went through the various tools and touch actions that should be working when a object file is run in VR. We also used hardware like Google Cardboard or Daydream System to test other parts of the VR. The data viewing component will be tested by pulling the sample data from the back-end that we were given access to and displaying the various data types. We will go through the pages and work on page loads and seeing the correct data for each data type.

All these testing process will be done integratively to ensure that older parts to a component are not broken when we push new code to the project. To see a flow diagram of the application see Figure 1.

3.4 RESULTS

We are currently working on the testing for route navigation and pulling data from the database in Amazon Web Services. For navigation testing, we have been able to log in to the sample we created to test out the UI. The menu was successfully implemented as a sidebar and we can log out correctly from the sample. We were able to access the data from the S3 storage in Amazon Web Services. This gave us the ability to see and download the files that we have the credentials for.

Other testing with the UI interface has been completely successfully so far. We were able to make sure that the different screen sizes and versions of phones did not affect the user experience using the emulator that comes with Android Studios. We are also able to download and view the PDFs and JPGs that are clicked. When it came to viewing object files, we were able to render the base objects but the mtl files were causing fragment shader errors because of illegal characters in the filenames. When we tried to change the filenames inside the application, it still looked for the previous filenames.

4 Closing Material

4.1 CONCLUSION

Our goal for this project is to design a fully functioning application, similar in design to InfraDrone's website that effectively displays data collected by a drone. We currently have the ability to collect a client's data so that we can display it. We also have a basic design and layout of our application as well as the ability to display images and PDFs. Moving forward into the next semester of the project we plan to try to follow our schedule created in scrum to stay on track and utilize the whole team.

As far as our end goal is concerned we plan to develop the VR during the second semester as well as refine our current application so that it looks more like the current web portal. Once we have developed a working application for Android we then plan to develop a working application for IOS by using Xamarin as a stretch goal. The whole time we will meet with our advisor and client on a regular basis with any problems that we have, as well as to confirm that we are on the right track.

4.2 REFERENCES

1. Vidyadharan, Akash. "InfraDrone VR Phone Application." Iowa State University. Ames, IA.
2. InfraDrone. "InfraDrone Web Portal." *Login*,
ec2-34-206-210-136.compute-1.amazonaws.com/Active/index.php.
3. "Rajawali." *GitHub*, github.com/Rajawa
4. Google. "Google VR." *Google VR*, Google, vr.google.com/.
5. "Amazon Web Services (AWS) - Cloud Computing Services." *Amazon Web Services, Inc.*, aws.amazon.com/.

