Oregon State University
College of Engineering

CS Capstone Design Document

December 1, 2017

ISS Barometer App

Prepared for

NASA

Don Pettit _____ _____
Signature *Date*

Prepared by

Group20

Cade Raichart _____ _____
Signature *Date*

Daniel Kato _____ _____
Signature *Date*

Nathan Shepherd _____ _____
Signature *Date*

Abstract

This document focuses on the individual components of our project and how they will be implemented. Following the template laid out by IEEE std 1016-2009, we have provided viewpoints and concerns in relation to our technology choices. We chose to focus on the design viewpoints provided by N. Rozanski and E. Woods for software design description, and have chosen 6 view points as applicable to our project.

CONTENTS

1 INTRODUCTION

1.1 Purpose

The ISS Barometer App will provide crew members of the International Space Station with a readout of barometric pressure, as read from the iPads internal sensor. This application will supplement the current method of air pressure reading, the MANOVACUMETER, a mechanical barometer. As our application is targeted toward the crew members, it needs to be highly readable, clear, and good at quickly conveying data. Along with ease of use, the need for readability and efficiency of use are the main decision factors in our production, as well as our design [1].

1.2 Scope

Listed below are the final product requirements of this product; version one will be provided at the end of December 2017, version two will be provided before the end of May. The optional research, referred to as O1, will be completed if time allows, and is not considered necessary for completion of the product.

V1: Data Screen with initial pressure, current pressure, start/stop button, time stamp, pressure change (measured in sec / mmHg)
V1: Graph Screen with plot pressure as a function of time
V1: Additional settings menu that includes plot scale, scroll functionality, orientation, decimal point, as well as other other units of measure for pressure
V1: All items will be optimized for efficient reading and clear interface
V2: Graph will have pinch to zoom functionality, scaling options (to be set in settings)
V2: Graph will be able to be exported as a csv, excel readable file
V2: Graph can be compared to t-Res table, to measure time remaining
O1: Research can be done to identify low pressure accuracy and resistance

As approved by our client, these are the goals of our project, with unlisted items being out of scope. This may be changed in the future, depending on available time, for instance, the research portion may not be feasible, due to time restrictions. Our version one application will be given to the client, who will then be able to give it to the ISS crew members. Although they are the intended target user, giving them version one will allow our team to improve upon any issues that the find in using the application. This means that our version two application may have changes not mentioned in the list, as they may be specific to the users and will only be apparent after use.

1.3 Context

This application is being developed for use on the ISS. The stakeholders of the ISS Barometer App include the client, Donald Pettit, and the current and future crew members of the ISS, as well as any NASA employees that will benefit from this application. That includes Ground Control members, as well as testers that may authorize our application for installation. Although there are other applications that read and record air pressure data, they have been deemed unfit for the following reasons.

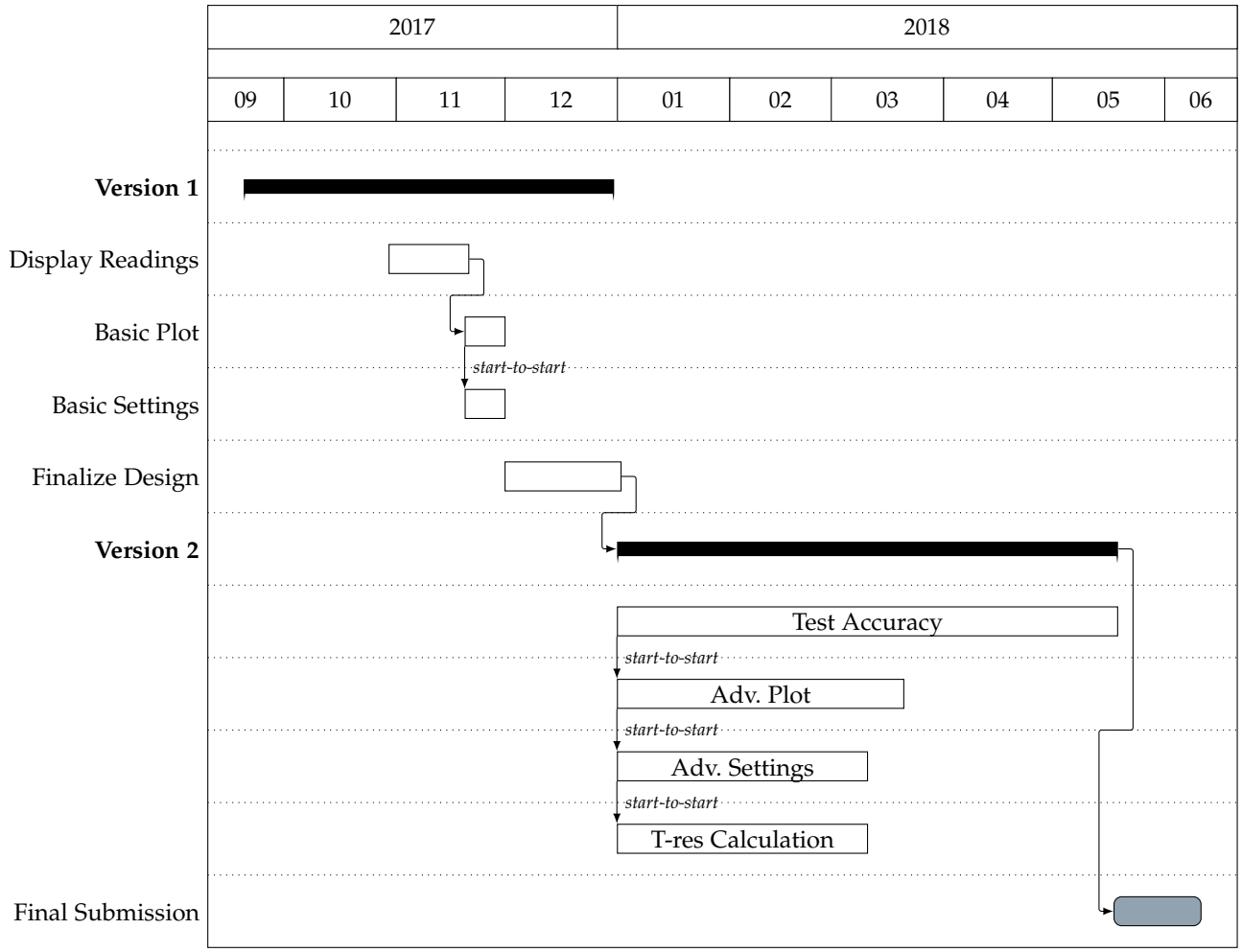
- Complex and hard-to-read UI
- Imprecise readings or recordings
- Lack of customizability involving units

- Optimized for altitude or weather

These needs prompted this project, and serve as simplified concerns that we seek to avoid. The users of the application, crew members of the ISS, require that the application meet their needs, as listed in the Scope. Because we will be avoiding pitfalls of other invalid applications, our application will be tailored specifically to serve the crew members. This means that their feedback, after version one is supplied, will be invaluable in helping guide changes to be made in version two. As a result, we should produce an application that is designed from the bottom up, for the crew members and NASA to use in the future.

2 TIMELINE

The following is a chart that displays our projected timeline [2].



3 GLOSSARY OF TERMS

2.1 csv: A file format readable by Excel and other spread sheet software

2.2 iOS: The internal operating system of Apple products, specifically iPads

2.3 iPad: A touch screen tablet device with barometer capabilities

2.4 ISS: International Space Station

2.5 MANOVACUMETER: The current mechanical air pressure gauge aboard the ISS

2.6 mmHg: Millimeters of Mercury, the standard pressure unit

2.7 NASA: National Aeronautics and Space Administration

2.8 t-Res: The chart that NASA uses to calculate residual time left on station

2.9 UI: Acronym for user interface

2.10 Segue: A transition form one view to another

2.11 API: Application Programer Interface; Third party code to aid in development

4 References

[1] D. Kato, C. Raichart, and N. Shepherd, "Problem statement: Iss barometer app," 2017.

[2] ——, "Requirements document: Iss barometer app," 2017.

[3] C. Raichart, "Technology review," 2017.

[4] N. Shepherd, "Technology review," 2017.

[5] N. Hanan. (2015) Segue between swift view controllers. Coding Explorer Blog. [Online]. Available: http://www.codingexplorer.com/segue-swift-view-controllers/

[6] developer.apple.com. (2017) Core data programming guide. Apple Inc. [Online]. Available: https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/CoreData/index.html

5 Context Viewpoint

5.1 Design Concerns

The context viewpoint is a project viewpoint that looks at the application in respect to its relationship with the environment. For this viewpoint, the major design concerns are related to the applications use, its users, and its location. The stakeholders of this perspective are the end users, the client, and NASA as a whole. All of them rely on the ability of our application to be clear in how it interacts through its UI. The end users require that the ISS Barometer App provide accurate data clearly and quickly, with easily understood interactivity. The users will be reliant on this application in potentially dire situations, leaks in the ISS, meaning it must not be unduly difficult to understand or operate. As mentioned in context section, the problem of complicated UI and difficult to understand displays are the reasons that other applications are not used. NASA requires that this application not introduce any security problems, and that any data displayed can be trusted. NASA also requires that the application be functional and clear of critical bugs, which could jeopardize mission safety. The latter requirement is lessened by the fact that NASA will keep the MANOVACUMETERS aboard the ISS, as well as other redundancies, but should still be considered. The setting of use, the ISS, introduces some interesting considerations to the design. For instance the accelerometer will not work as it does in 1g, and requires shaking to reorient the iPad. Also the iPad may be exposed to extreme low pressures, and should be trusted to remain functional and accurate. These considerations do not affect the design of the application, but may be tested if the optional goals are achieved. In addition to low pressure and gravity, the stations remoteness may also affect design, in that updates and communications will be sent through NASA and their network.

5.2 Design Elements

In response to the design concerns, several design elements have been chosen to alleviate the dilemmas. As a result of the crew members need for clear, usable design, we have decided to use Swift Charts as our plot library, and a

combination of Cocoa controls, FlatUIKit, and CRPageViewController [3]. In addition to specific UI libraries, we plan to design the UI with simplicity and readability in mind. This will be judged in part by our client, and eventually by the crew members once it is deployed, as version one. The plot library will provide the functionality that is needed, in addition to being a supported and trusted plotter. Using trusted and recommended libraries and packages will also help provide NASA with understandable and trusted code. This means that their approval and authorization will be quicker and therefore allow faster turn around on things such as updates. Accelerometer issues caused by low gravity is an easily solvable issue, requiring only the addition of a button to switch orientation. This has been a design choice, proposed by the client, since the beginning, and would solve the issue for the end users. The other problem caused by the environment, that of low pressure accuracy, will not be as easily solved. As our application is not able to alter the hardwares capabilities in withstanding pressure, the only option would be to alter the platform. The options that we have considered include external Arduino applications, the iPhone 6, or the iPad Air. The iPad has been identified as our required platform, by our client, and backed by our research [3]. This means our only option for ensuring the reliability of the application is to test it in a controlled pressure container. This would require things outside the scope of this project, but may be achieved as a stretch goal, as seen in the Scope section. The added difficulty of transferring data through NASA to the ISS means that updates will be sparse, and return data or remote data storage is infeasible. These factors guided our decision to use CoreData and NSUserDefaults as our storage packages, instead of a remote server package, like SQLite [4]. The choices made for the Context Viewpoint will help optimize our application for the location and users involved.

6 Functional Viewpoint

6.1 Design Concerns

The functional viewpoint is a project viewpoint that deals with the capabilities, external interfaces, and internal structure of each component of the application. For the ISS Barometer the main concerns are in regards to the maintainability of the code and the way each element of the application interacts with one another. Each of the main components of the application will need to pass data around in a manner that is easy follow and manipulate. The stakeholders for this viewpoint are the client, the ISS crew members, our team, and NASA as the ability of our application to be updated and maintained relies on a good interface design and internal structure. It is necessary that our application be well modularized so that it can be reworked to meet any future needs of NASA and the crew on the ISS. One concern is that the settings page and the main page will need to be able to communicate and pass data to one another. Another concern is that the graph will need continual access to an updating array of pressure data points. It is important that the graph is updated each time a new barometer reading is taken to provide the user with as much data as possible.

6.2 Design Elements

To address the concern of maintainability, we will implement 3 views: a main view, a graph view, and a settings view. The graph view will exist as a sub view of the main view. Each view will be controlled by a corresponding view controller class. Separating out each of the pages as views will allow us to keep the code separate and organized, as well as let us reuse the graph view to display it in full screen. To address the concern of transferring data between views, we will use segues as outlined in the Coding Explorer Blog [5]. This will involve overriding a function in the called `prepareForSegue()` in the sending end of the segue to assign a value to a variable that is passed into the receiving

end of the segue. This will allow the receiving end of the segue to to access the passed in data through the designated variable name. To store each barometer reading, a member variable of the main view controller will be created that holds an array of all of the collected data points in the current session. To address the concern of updating the graph with new readings, a member function of the graph view controller class will be called after each reading, adding the taken data point to the graph.

7 Information Viewpoint

7.1 Design Concerns

The stakeholders of the information viewpoint are those reliant on accurate and secure data storage. This is all of the main stakeholders: the client, the ISS crew members, our team, and NASA, as a whole. As the purpose of our application is to display information from the barometer module of the iPad, so it follows that it is necessary to display correct and timely data. Also necessary, is the ability to store the data between uses, so that a user may exit the application and open to the same view, complete with prior data. One additional concern may be the need to record data while the application is not running in the foreground of the iPad. This may be a concern, but from our understanding of the clients specifications, this is not within scope, and shouldn't be an issue. Once the user has a plot recorded they might want to export it as a csv file, which must also be secure from errors or tampering. It must be ensured that the data being exported to the csv is the data that was recorded. Once the csv is exported, that data is no longer our concern, as it is in the hands of the user. In addition to exporting and saving plot data between uses, our application will have some various settings, which will be stored between uses, so that certain settings are persistent. The settings that must remain persistent between uses are listed below [2]:

- Significant digits
- Plot scale
- Plot scroll functionality
- Orientation
- Units of pressure
- Sample rate of recording

7.2 Design Elements

The main security that will be implemented is done through using well defined and secure storage packages. We are not expecting intentional tampering, as the application will not be connected to the network, and the only users are using the application for self-preservation. That said, it is still possible that data be mangled, dropped, or altered as it is recorded or exported. By correctly using well maintained and trusted packages we will avoid many problems with data storage. As the data being stored is small, and relatively simple, the use of packages will simplify our storage, while allowing the users quick and reliable data. The packages that will store plot data and settings data will be two different packages. This method, as suggested by Apple[6], will separate user data, such as settings, from application data, like the recordings or the plot. The user data package will be NSUserDefaults, which is suggested for storing small portions of data. The settings stored will be integers or boolean data, and like other application data, will be stored locally on the iPad. The plot storage package will be CoreData, which will be able to handle larger sets of data, like the plot. This also will be stored locally.

7.3 Design Attributes

Data stored by the NSUserDefaults will be of the form of integers or booleans, as it will only be storing user choices. Data stored by CoreData, for the plot, will be in the form of paired floats and timestamps. Together each of these data points will form a data point on the plot, as the recording continues, and the graph implementation plots the points. If the data is recorded and then exported, the user will be prompted whether or not to remove the data stored in the application. If the user accepts then the plot data will be removed, and the screen will be returned to its prerecord state. The data must be removed from the application, either by deleting or exporting, before the user will be allowed to record again. This technique prevents overflow of data storage, which would cause the application to crash.

8 Concurrency Viewpoint

8.1 Design Concerns

For this viewpoint the major concerns are related to the speed of hardware, and timing. The stakeholders of this would be our client and the end users, the crew members aboard the ISS. The astronauts will need to rely on our app to correctly have all the timing work in unison, without race conditions causing out of sync displays. When using hardware and software systems together we run into some timing problems. Our app might have a timing concern with how fast the barometer records atmospheric pressure. The barometer might be reading the data slower than the graph can display it, or it might be reading it faster than we can display the data on the graph. This can cause problems with the end users. Looking at the graph with a older atmospheric pressure than the one the barometer can cause the user to not evacuate early, which could result in death of the crew. Our app will be doing many actions all at the same time, this all needs to be done simultaneously as to ensure the accuracy of the data. Astronauts need to have all data be precise and accurate to save lives.

8.2 Design Elements

In response to the design concerns, several hardware chips are being used correctly calculate the timing problems. The ISS Barometer App will be using the iPad Air 2 for its platform. The iPad uses a Apple A8x triple core 1.5 GHz Typhoon CPU. The triple core cpu allows for multiple action to be done simultaneously. This means that we can display the atmospheric pressure, $\frac{\Delta p}{\Delta t}$, calculate time to evacuation and display the graph simultaneously making it accurate and safe. With each core of the CPU running at 1.5 GHz we do not have to worry about the barometer recording data faster than it can be displayed. The choices that have been made regarding the graphing library, Chart, ensure that our plot will also be running at its optimized speed. Also, as the barometer module is internal the iPad uses the same CPU to poll it for barometric pressure. This means that, using this barometer, our application will avoid halting while waiting for data, as everything will be operating on the same structure. If we were to use an external barometer then this would be an issue, with wait times requiring that we factor in delay.

9 Development Viewpoint

9.1 Design Concerns

The development viewpoint is a project viewpoint that deals with the organization of modules, the standardization of testing, and common processing. For the ISS Barometer, the main concerns are in regards to testing, development framework, programming language, and third-party APIs. The stakeholders for this viewpoint are primarily the team,

as they are going to be the ones interacting with the development tools. Testing our application is necessary to ensure that each incremental change does not break the existing code base. The development framework will need to cater to the needs of the application which includes: direct access to the barometer, ability to incorporate a chart API, and ability to display separate views. Any third party APIs used will need to be compatible with iOS 10+, and positively contribute to the code base without creating any unnecessary burdens. The programming language used will have to be supported by the development framework, and the iPad.

9.2 Design Elements

To address the concern of development framework, we will use Xcode to write our application. This will allow us direct access to the barometer and any native features of the iPad. It will also allow us to take advantage of the native Xcode testing framework. To address the concern of module organization, we will use the package manager Cocoapods to install and manage all third party APIs. Cocoapods will also make it easy to keep our applications dependencies up to date to avoid security issues. We will be using the Swift 4 language to develop our application because it is a supported and feature rich language made by Apple that has received plenty of positive feedback regarding its usability and ease of development. We will be using 2 third party APIs: Charts by Daniel Daniel Cohen Gindi, and Cocoa Controls. These libraries were chosen due to their compatibility with the project as outlined in the technology review. To test our application's code and UI, we will use Xcode's built in testing library. This is the best solution as it is a full featured testing suit that is set up when the project is created, making it easy to test our program from the beginning.

10 Deployment and Post-Deployment Viewpoints

10.1 Design Concerns

This viewpoint is a project that looks at the environment that the system will be deployed in. The operational concerns are how it will be operated, administered and supported while in deployment. The second concern of this viewpoint is will the app help the crew while in use. The ISS Barometer App will be aboard the ISS space station. The space station is orbiting the Earth at a minimum of 330 km. The interior of the ISS has a pressure of 1 atmosphere, measured in millimeters of Mercury (mmHg), that must be maintained for the safety of the Astronauts aboard the ISS. If the atmospheric pressure drops below .540 atm it will endangering the crew. Leaks in the ISS can be caused by faulty seals in hatches or orbital debris striking the vessel, a severe leak consists of a loss of 1 mm per minute.

10.2 Design Elements

In response to the viewpoint concerns our app will be deployed to find these leaks. The astronauts will operate the app by going from room to room, closing hatch door behind them and seeing if the atmospheric pressure stabilizes. If it doesn't stabilize they will move on to the next room. Until they find the room/rooms with the leak or they must abandon ship. When the app is activated with the start button, it will record the atmospheric pressure and display it along with the appropriate time stamp. That data will be used to calculate the $\frac{\Delta p}{\Delta t}$. $\frac{\Delta p}{\Delta t}$ will be displayed by the first page, used as the slope of the graph, and it will be used with the comparison to the t-Res tables to display the time till evacuation. The App will be administered by uploading the files to the space station and downloading onto the iPads. Updates to the app will administered to the iPad in the same way. As the iPad's OS gets update so must we update the app.

11 Conclusion

As described in our Technology documents, we have sought out different options for the major pieces of this project. With the research done, and the viewpoints provided here, it has been shown that the ISS Barometer App is properly thought out and detailed. This application is simple in its infrastructure, but has the potential for quality design and efficient and reliable displays. In addition, testing this application, and the limitations of the iPad, if completed, may provide beneficial insight to the operating limitations and reliability of the readings. These improvements are planned for version two, as mentioned in the Scope. Using the viewpoints from this document, in addition to potential feedback from the users after version one, there may be many unforeseen improvements to be made. As those improvements are made, our application will become perfectly tailored to the users' needs.