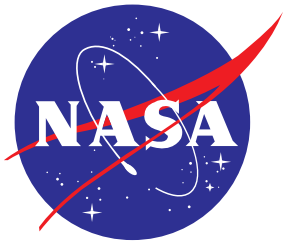

CS Capstone Technology Review

November 21, 2017

ISS Barometer

Prepared for

NASA

Don Pettit _____ _____
Signature *Date*

Prepared by

Group 20
ISS Barometer

Daniel Kato _____ _____
Signature *Date*

Abstract

When the International Space Station (ISS) has a leak, they use barometers and lookup tables to determine the amount of time they have before they need to evacuate. Unfortunately, the ISS is down to 2 barometers, but have 8 - 10 iPad Air 2's on board. Our goal is to create an application that can aid the astronauts on the ISS in calculating the amount of time they have before they need to evacuate the ISS in the event of a leak.

There are many different technologies that can be used to create this application including Development Framework, Chart Library, and Organization of the UI. The decision of which technology to use for each component will be made based on comparing three different options. Which ever option best suits the clients needs will be selected.

CONTENTS

1 OBJECTIVE

The International Space Station(ISS) serves as laboratory for crew members to conduct experiments in microgravity. The ISS can experience breaches in the hull for many reasons, but when the hull is breached, the pressure of the environment begins approaching unsafe levels. To aid in the fixing of breaches the crew members currently use 1960's era mechanical barometers called *MANOVACOMETERS* to measure the pressure of the cabin and record the rate it in which it is decreasing. Because these are no longer being manufactured, the numbers of these barometers are dwindling.

Our objective is to create an iOS application that replaces these mechanical barometers. The application will display the current pressure like the *MANOVACOMETERS*, but will also display other useful pieces of information such as the current change in pressure as a function of time($\frac{\Delta p}{\Delta t}$), and a graph displaying the progression of $\frac{\Delta p}{\Delta t}$.

My role in the ISS Barometer team will be to choose the development platform, focus on the implementation of the chart that plots $\frac{\Delta p}{\Delta t}$, and choose the overall organization of the user interface. This graph will have 2 modes: a mode in which the data flows off the left side of the graph and a mode where the data is compacted to allow the entire frame of data to fit in the graph. The x axis will represent time and the y axis will represent mm/Hg.

2 DEVELOPMENT FRAMEWORK

2.1 Xcode/Swift

Xcode is the development platform provided by Apple for engineers to create apps for their iOS products. This platform allows the developer to write code and connect it to User Interface elements to create an application. This the most commonly used framework for iOS app development. Because Xcode and Swift are the proprietary software development tools from Apple, these tools provide the developer with the full capability to interface with Apple's hardware. This will allow the developer to have full access to the iPad's onboard sensors through Apple's APIs.

The way the user interface is created and manipulated in Xcode is through a drag-and-drop style interface called Interface Builder. This can make it easy to quickly turn an idea into an application, but when it comes to collaborating and having multiple people make changes to the UI, the Xcode implementation falls short. When interacting with Interface Builder, each change is saved to a .xib file, which is obfuscated away from the developer. This becomes a problem when certain UI elements that are linked to the code are changed, because the reference to the UI element that is linked to the Swift code may be different after the UI change. This can cause errors that can only be fixed by opening up the xib file and manually changing the reference, or deleting the code and re creating the reference, which can cause its own set of problems.

2.2 Unity

Unity is a game development platform and game engine that can be used to create iOS applications and games. This framework is Because Unity is primarily used to create games, the UI and transitions between pages could be enhanced by taking advantage of Unities computer graphics APIs. Because Unity is a game engine, there is likely to be a massive amount of unnecessary overhead when creating an application that is not graphics intensive. It also does not have the capability to interface with the iPad's built in barometer.

2.3 React Native

React Native is a software development framework that allows developers to create apps for mobile devices using javascript, CSS, and HTML. Because the UI in a React Native app is controlled by CSS and HTML, every creation or modification of an element will be executed via code and not by Xcode's Interface Builder, it will be a lot easier to collaborate and make changes to the code base. React native was built to allow the rapid creation of cross platform apps. As a result, it does not offer the ability to interface with the built in barometer.

2.4 Decision

Because the ability to interface with the onboard barometer is required for this project, we will be using Xcode and swift to develop the ISS Barometer application.

3 Chart Library

3.1 Criteria

A good chart library will be one that is easy to use, visually appealing, and have the ability to update a graph elegantly in real time. The chart will need to be configurable in a way that allows the data to either flow off the side of the graph, or shrink the width between points for every added data point. It will also need to be a library that is compatible with the iPad Air 2 and iOS 10+.

3.2 Charts

Charts is an open source library by Daniel Cohen Gindi for building charts in iOS applications. Charts' latest commit at the time is 6 days ago, which means that it is still being maintained to some degree. It also specifically supports Xcode 9.0 and Swift 4.0, making it likely that we wont run into compatibility issues. This API also has specific methods for adding realtime and dynamic data to the graph. One downside is that the graphs are not particularly appealing visually

3.3 Dr Charts

Dr Charts is an open source library by Zomato for building charts in iOS applications. The charts created using Dr Charts can be animated to create a more pleasant user experience without compromising visibility of data. A few downsides to this API are that it only mentions support for iOS 7, 8, and 9 and this could cause compatibility issues. Additionally, there is no built in method for adding a data point to the graph and this could make it difficult to create a live graph with updating data.

3.4 PNCharts

PNCharts is an open source library by Kevin Zhow for building charts in iOS applications. This API provides animations for several different types of graphs, which could contribute to a better user experience. The API also makes it pretty simple to construct a graph. Instead of having class methods, you just set certain member variables to the data you want to represent in the graph. A downside is that this API hasn't made a change since June 7th of 2017, so it is unclear as to if this repo will continue to be maintained.

3.5 Decision

Because reliability is of the utmost importance, and Charts offers support for dynamic graphs, we will be using the Charts API for the ISS Barometer App. The other 2 libraries don't have the consistent support present in Charts. Also the Charts API is makes it very easy to add points to a dynamic graph, making it perfect for the clients needs.

4 Organization of UI

4.1 Tabs

A tab based application is one that has multiple views that are accessed via a tab bar at the bottom of the screen. This is advantageous because each page is only one click away at all times. Also, just by reading the tabs you know all of the possible pages you can use to interact with the application. The tab bar is cemented to the bottom of the screen at all times, which takes up quite a bit of screen real estate, and also detracts from the focus of the page. Also some pages don't need to be displayed at all times such as the settings page.

4.2 Single Page

A single page application will have the graph and the statistics displayed on the main page. With this view, you can double click on the graph to make it fullscreen, and access frequently used settings via gestures. All other settings will be accessible from the app's tab in the setting application on the device. This layout will allow for more of a focus on the content, and reduce the number of infrequently pressed buttons. By employing gestures, it will end up taking more time to learn how to use the application comfortably .

4.3 Homepage With Sub Pages

The homepage will have informative icons which when pressed will take the user to a view with either the statistics, the graph, or settings. The focus will be on the content, and users will be able to explicitly select what content they see and don't see by choosing a view. by having pages linked to a main page, a button to go home will need to be present on the screen to allow the user to go back and select another page.

4.4 Decision

Because settings shouldn't need to be accessed frequently, and both statistics and the graph will need to be accessed when ever the astronaut is using the application, we will be using the single page layout.