Project Deliverable G: Prototype 2

GNG 2101 – Intro. to Product Dev. and Mgmt. for Engineers

Faculty of Engineering – University of Ottawa

Submitted by

[B34]

[Michael Kagnew, 30011347]

[Fatimah Vakily, 300125671]

[Gabriel Cordovado, 300110852]

[Joseph Francis, 300116730]

[Jasmine Kokkat, 300115249]

[Emma Ballantyne, 300115563]

November 5, 2020

University of Ottawa

Table of Contents

List of Tables 2
List of Figures 2
Introduction 3
Client Feedback Summary 3
Prototype 3
Prototype Testing 5
Project Plan Update 7
Conclusion 8

Table of Figures

Table 1: Target Specifications Prototype Test 6

List of Figures

Figure 1: Initial Prototype 2 3
Figure 2: Screenshot Functions 4
Figure 3: Android Prototype 4

Introduction

In this deliverable the feedback received from our client on our first prototype is summarized and considered for the creation of our second prototype. The prototype we have created is displayed and the testing that occurred is documented with the expected and experimental results recorded. Difficulties with the second prototype led to our new design, the basics of which are explained. Finally, our project plan is updated again, outlining the tasks and deadlines for our next deliverable, as well as the final product.

Client Feedback Summary

The focus of this meeting was to clarify to our client the limitations we faced when trying to meet his needs. We explained that we were focusing solely on notifications and nothing related to his other interactions with his phone. We also described our first prototype and the display screen. The client gave helpful notes on the display that would be optimal. He advised using bright colours such as red or blue and avoiding greens and yellows. We were also informed that our client is undergoing a medical procedure and will be unavailable for any contact from November 5th to November 15th. Knowing this, we set ourselves a goal of having our second prototype complete before November 5th, in the case that additional discussion needed to happen with the client about how we would move forward with our final deliverable.

Prototype

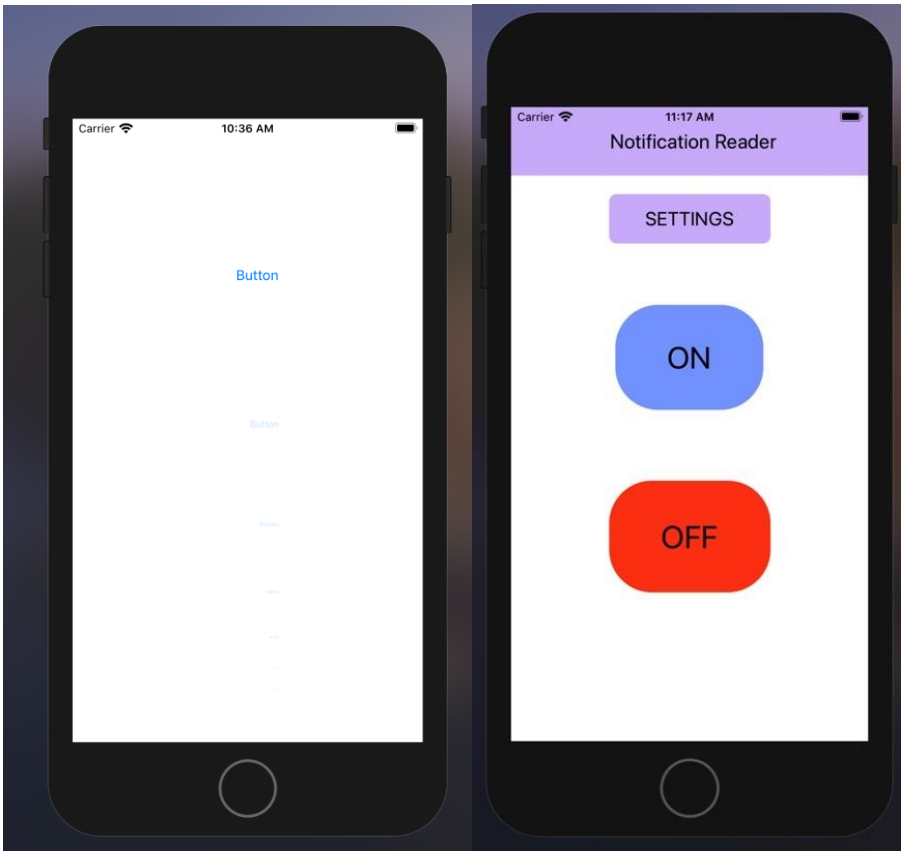


Figure 1: Initial Prototype 2

For our initial prototype, we used Xamarin.iOS to create an application which would take screenshots of the notification screen to save on Firebase. We were able to set up Firebase for the iOS application but the screenshots, as shown in Figure 1, were not clear to see. A section of our code for the screenshots we took is shown below in Figure 2, with the bytes being then converted to an image to display on the screen. The quality of the screenshot was low, and it only took screenshots of the UI of our application rather than the notification screen. Due to Apple's restrictions on taking screenshots of the main screen, we were not able to overcome this challenge.

```
using System;
using UIKit;
using System.Runtime.InteropServices;
using Foundation;

namespace AccessibleMessaging.Services
{
    public class ScreenshotService
    {
        public ScreenshotService()
        {
        }

        public static byte[] Capture()
        {
            var capture = UIScreen.MainScreen.Capture();
            using (NSData data = capture.AsPNG())
            {
                var bytes = new byte[data.Length];
                Marshal.Copy(data.Bytes, bytes, 0, Convert.ToInt32(data.Length));
                return bytes;
            }
        }

        public static UIImage convertToUIImage(byte[] myByteArray)
        {
            return UIImage.LoadFromData(NSData.FromArray(myByteArray));
        }
    }
}
```

Figure 2: Screenshot Functions

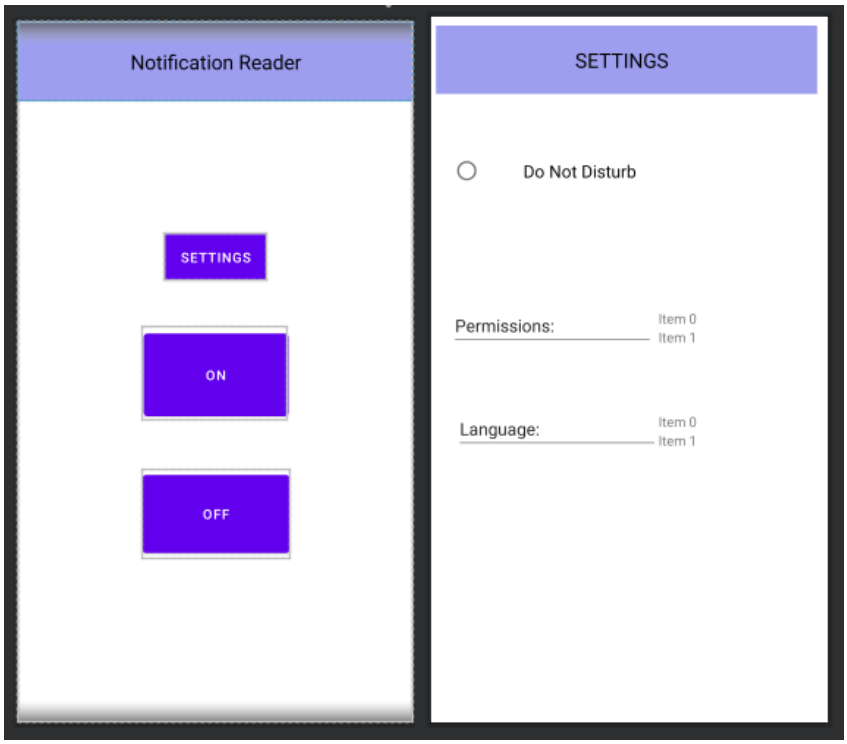


Figure 3: Android Prototype

The next prototype we attempted was an Android prototype made using Android Studio and the Java coding language. The interface is shown in the figure above and includes the main screen and the settings page. We once again implemented a Firebase database to track user information with the app.

Prototype Testing

Table 1: Expected vs actual results for the target specifications

Metric #	Metric	Expected results	Actual Results
1	Ease of gesture navigation	Limited number of screens to help user quickly find the information with reduced swiping	Passed since there were only three large buttons with most of the work done by the application rather than the user
2	Intuitive button usage to activate/deactivate	Have very visible buttons in identifiable colours for the client to turn on or disable the application with minimal steps	Passed with a very simplistic UI design to avoid accidentally clicking on the wrong button
3	Voice controls display information from recent notification	Intended to read out who the sender is, the message and application the message was from using Siri	Failed because SiriKit was not implemented since notifications were not accessible
4	Multi-language use (English and Spanish)	Can detect the text's language from being given the notification to read it with	Failed since the ML kit was set up in application to offer this functionality but the images

		the correct pronunciation.	could not be processed
5	Non disruptive functionality to phone usage	Allows the user to interact with the phone without interference from the application in speed, storage capacity, batter power, and navigation methods	Failed since it takes up client's phone storage and application could time out if used for too long
6	Reads only notification-related information	Can read the entirety of the message shown on the main screen of the iPhone by identifying	Failed because the application was unable to access notifications
7	Customer Satisfaction	Uses an iOS platform to make the application to accommodate his preferences	Failed since we were not able to use his OS
8	Time to load application	Can load the application in 2-3 seconds so the notifications are close to instantaneous	Passed with a reasonable speed of about 3 seconds
9	Ease of installation and setup on phone	Can be installed on the App Store with no authentication required	Failed since it has not been put on the App Store at this stage

Table 1: Target Specifications Prototype Test

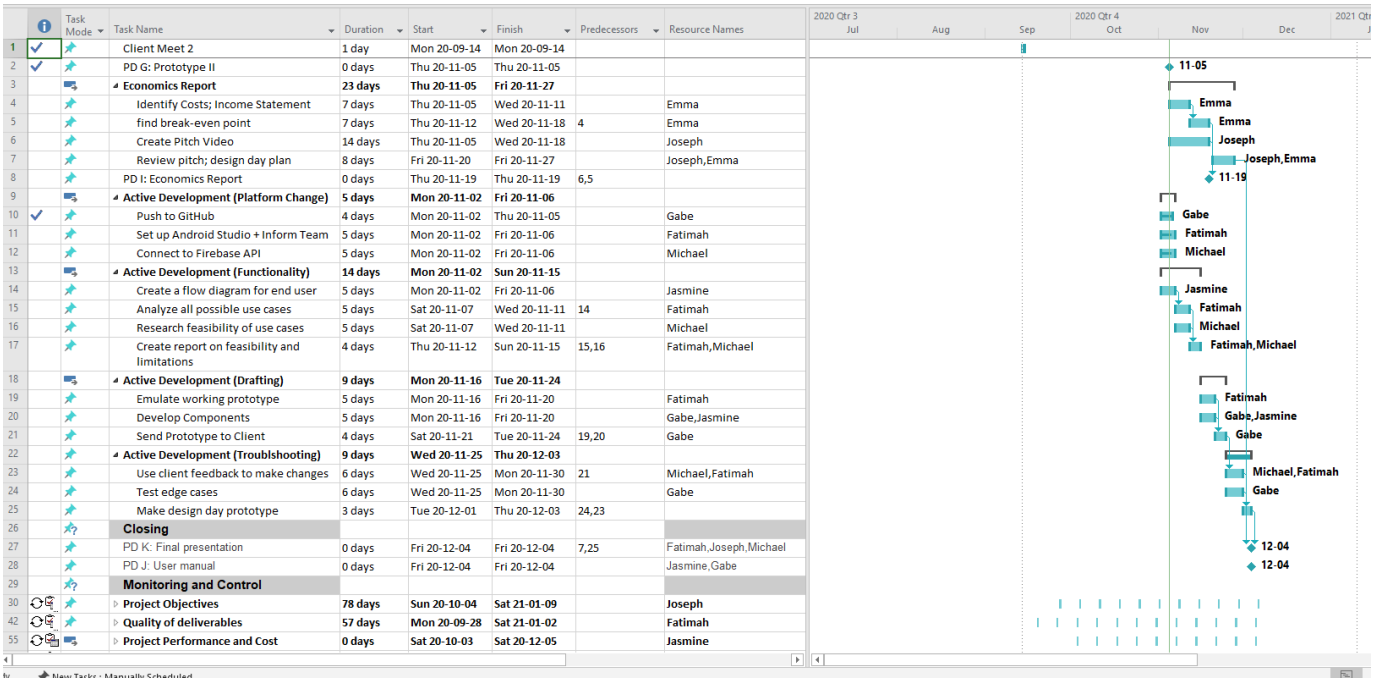
The above comparisons come from a problematic situation that was encountered while testing out the possible functionality with iOS. Unfortunately, after many attempts, it was concluded that the application could simply not work on an iOS device. This realization came later than desired because the team had limited knowledge about the limitations of the environments, we were working in. Unfortunately, planned 'work-arounds' would not be possible on iOS due to the restrictive nature of the system. These loopholes include plans such as keeping processes running in the background, taking screenshots of the screen automatically, listening for new notifications, accessing notifications, etc.

Apple's system is designed to limit an application's functionality once minimized. It makes an exception for app which connect to music, Bluetooth, location-services, or VOIP. This is a list for which our application does not fall within, hence, the iOS environment will terminate the application promptly within a 3 to 15 minutes window. Unlike the Android OS, iOS is a restrictive environment, which limits the sharing of information between proprietary and 3rd party licensed developers. As a result, *there is no open-API for the device notification center*. As a result, we are required to take screenshots programmatically to capture a possible notification when pushed to the screen by the OS.

Faced with no path forward, the team made the choice to use Android. This is of course, fundamentally not what the client requested; however, the original project is simply not feasible on iOS. This switch is a setback, therefore a truly working prototype II is not available, however the UI is able to be demonstrated. The goal is to have much more functionality be implemented within the coming days to make up for lost time in switching platforms.

Project Plan Update

(See MPP file for the full, updated project)



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Client Meet 2	1 day	Mon 20-09-14	Mon 20-09-14		
2		PD G: Prototype II	0 days	Thu 20-11-05	Thu 20-11-05		
3		**Economics Report**	**23 days**	**Thu 20-11-05**	**Fri 20-11-27**		
4		Identify Costs; Income Statement	7 days	Thu 20-11-05	Wed 20-11-11		Emma
5		find break-even point	7 days	Thu 20-11-12	Wed 20-11-18	4	Emma
6		Create Pitch Video	14 days	Thu 20-11-05	Wed 20-11-18		Joseph
7		Review pitch; design day plan	8 days	Fri 20-11-20	Fri 20-11-27		Joseph,Emma
8		PD I: Economics Report	0 days	Thu 20-11-19	Thu 20-11-19	6,5	
9		**Active Development (Platform Change)**	**5 days**	**Mon 20-11-02**	**Fri 20-11-06**		
10		Push to GitHub	4 days	Mon 20-11-02	Thu 20-11-05		Gabe
11		Set up Android Studio + Inform Team	5 days	Mon 20-11-02	Fri 20-11-06		Fatimah
12		Connect to Firebase API	5 days	Mon 20-11-02	Fri 20-11-06		Michael
13		**Active Development (Functionality)**	**14 days**	**Mon 20-11-02**	**Sun 20-11-15**		
14		Create a flow diagram for end user	5 days	Mon 20-11-02	Fri 20-11-06		Jasmine
15		Analyze all possible use cases	5 days	Sat 20-11-07	Wed 20-11-11	14	Fatimah
16		Research feasibility of use cases	5 days	Sat 20-11-07	Wed 20-11-11		Michael
17		Create report on feasibility and limitations	4 days	Thu 20-11-12	Sun 20-11-15	15,16	Fatimah,Michael
18		**Active Development (Drafting)**	**9 days**	**Mon 20-11-16**	**Tue 20-11-24**		
19		Emulate working prototype	5 days	Mon 20-11-16	Fri 20-11-20		Fatimah
20		Develop Components	5 days	Mon 20-11-16	Fri 20-11-20		Gabe,Jasmine
21		Send Prototype to Client	4 days	Sat 20-11-21	Tue 20-11-24	19,20	Gabe
22		**Active Development (Troublshooting)**	**9 days**	**Wed 20-11-25**	**Thu 20-12-03**		
23		Use client feedback to make changes	6 days	Wed 20-11-25	Mon 20-11-30	21	Michael,Fatimah
24		Test edge cases	6 days	Wed 20-11-25	Mon 20-11-30		Gabe
25		Make design day prototype	3 days	Tue 20-12-01	Thu 20-12-03	24,23	
26		**Closing**					
27		PD K: Final presentation	0 days	Fri 20-12-04	Fri 20-12-04	7,25	Fatimah,Joseph,Michael
28		PD J: User manual	0 days	Fri 20-12-04	Fri 20-12-04		Jasmine,Gabe
29		**Monitoring and Control**					
30		**Project Objectives**	**78 days**	**Sun 20-10-04**	**Sat 21-01-09**		**Joseph**
42		**Quality of deliverables**	**57 days**	**Mon 20-09-28**	**Sat 21-01-02**		**Fatimah**
55		**Project Performance and Cost**	**0 days**	**Sat 20-10-03**	**Sat 20-12-05**		**Jasmine**

Note that prototype II and design day prototype have been put together, given as "active development". This is because, due to our having to switch platforms late in the project, we cannot have a fully functional Prototype II and instead we will work towards the design day application.

Conclusion

In conclusion, though the team attempted multiple strategies to allow the app to function on the client's current iPhone while considering all the limitations imposed by Apple, the realization arose that such an application would only be able to be run if it was creating more problems to the client than it solved. Due to this, the team has—with the support of the client—since switched plans to begin developing the accessible messaging application as an android app instead. Though the development process so far has been full of setbacks and failures, we have gained a better understanding about the issues our client faces, experience designing a product from scratch, and comprehensive knowledge on how to organize our team together and keep trying despite the failures we faced. Currently, the project seems to be heading in a promising direction, as we are now able to utilize coding languages and technologies, we are much more experienced in (such as Android Studio and Java instead of Xamarin and Objective-C), and no longer have to work around all the barriers Apple imposes on iOS developers.