

# NextGen Playbook

by


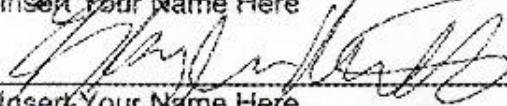
Ryan Cash and Skylar Mills



Submitted to:  
the Faculty of the School of Information Technology  
in Partial Fulfillment of the Requirements for  
the Degree of Bachelor of Science  
in Information Technology

© Copyright 2017 Ryan Cash

The author grants to the School of Information Technology permission to reproduce and distribute copies of this document in whole or in part.

 _____ Insert Your Name Here	<u>4/3/2017</u> Date
 _____ Insert Your Name Here	<u>4/2/2017</u> Date
<u>Russell E Mc Melton</u> _____ Insert Faculty Advisor's Name Here, Faculty Advisor	<u>4/15/2017</u> Date

University of Cincinnati  
College of  
Education, Criminal Justice, and Human Services

April 2017

## Table of Contents

<b>Abstract</b> .....	1
<b>Problem Statement</b> .....	2
1.1 Introduction	
1.2 Project Description.....	3
1.3 Problem	
1.4 User Profile.....	4
1.4.1 Application Title	
1.4.2 Potential Users	
1.4.3 Software and Interface Experience	
1.4.4 Experience with Similar Applications	
1.4.5 Task Experience	
1.4.6 Frequency of Use	
1.4.7 Key Interface Design Requirements	
1.5 Technical Elements.....	7
1.5.1 Software	
1.5.2 Network	
1.5.3 Server	
1.5.4 Backup	
1.6 Objectives/Deliverables.....	9
1.7 Budget.....	12
1.8 Use Case.....	13
1.9 Testing Plans and Results.....	14
1.10 Future Plans.....	17
<b>Conclusion</b> .....	18
References.....	20
Appendix A: Code Snippets.....	21
Appendix B: Presentation Poster.....	22
Appendix C: Application Screenshots.....	23
Appendix D: System Architecture Layout and Work Flow Model.....	25

## Figures

Figure 1: User Profile.....	5
Figure 2: Screen Shots.....	6
Figure 3: Architecture Diagram.....	8
Figure 4.1 & 4.2 Objectives/Deliverables.....	9
Figure 5: Budget Table.....	12
Figure 6: Use Case Diagram.....	13

**Abstract:**

NextGen Playbook (NGP) is a secure application that revolutionizes the way coordinators can call and deliver plays to their players. Utilizing a Microsoft Surface tablet to call the plays, NGP will then use a secured Wi-Fi network to send a picture of the play chosen to the android wrist devices worn by all of the offensive players. This program houses pictures of the plays in the SQL database and allows the pictures to be sent to the players on the field to their digital wristband device. NextGen Playbook increases the ability to communicate plays between the sideline and those on the field while improving game speed by eliminating the need to huddle. NGP is designed to decrease the communication issues with radio and current helmet devices; however, it is designed to work alongside those technologies instead of replacing them. NGP answers growing concerns for the NFL's television ratings dropping as well. This is accomplished by speeding up the game, while allowing more action with the ball in play more often.

## Problem Statement

### 1.1 Introduction

In both college and professional football, studies have shown that the speed of the game is well connected to the popularity of the sport. The Commissioner of the National Football League (NFL), Roger Goodell, met with The New York Times in November of 2016 and explained that they are concerned that the speed of game play is what is causing the drop in TV ratings for the NFL (NY Times). However, football remains to be the number one watched sport in America per a report from *therichest.com* published in February of 2015 that reviewed television viewer numbers from 2014. The NFL accounts for 32% of the sports viewers each year, doubling the next closest sport of baseball at 16%. College and professional football are two of the top three watched sporting groups in America, accounting for 42% of tv viewers (*therichest.com*). However, football has seen an 11% drop in television ratings from 2015 to 2016. Football teams at all levels (high school, college, and professional) are experimenting with different offensive schemes to help speed up the game. Offense designs, like the spread, are a hot topic and used for football at all levels. Spread offenses are limited by the ability to call plays effectively at a fast and continuous rate over the entire course of the game. With completely eliminating the need to huddle as much and perfecting the way to immediately communicate the play without error, NextGen Playbook will create the perfect spread offense. NGP will not replace the radio communication; rather, it is intended to work alongside radio, while improving several other aspects of the game. Faster offensive play calls with this innovative tool will allow less time for the defense to react, causing a change to how the game is played and coached. This will also allow, the ability to have clear, constant, accurate, and easy communication to all the players on the field for every offensive play call.

## **1.2 Project Description**

We have designed an application with an easy to use user interface that instantaneously sends plays from a Microsoft Surface used by the coaches to an Android end device, similar to a smart watch, worn by players. This program will house all of a team's plays for offense and will keep pictures of each play in the form of a BLOB in a SQL database. As soon as the play is selected by the coach, it will be sent to the players' devices, decreasing or eliminating the amount of times needed to huddle. This device will speed up the game, allowing for more action that may increase people's interest, and NGP also will almost completely eliminate common communication issues that occur today.

Before the start of each game, the coaches will be able to add, remove, or edit plays on the program to prepare for the upcoming games. With the proper data available, play prediction could be added to the program that would base the the predictions off of input play history and be based off tendencies of the coaches. Overtime, this feature would allow for the coaches to quickly select the plays they wish to use at any given point in time.

## **1.3 Problem**

The problem that NextGen Playbook faces is the issue of securing all facets of the technology. NGP will require security on each team's server, network, devices, databases, and signal used to send the plays to the end devices. For this application to be used in a real-world scenario, it must have proven security on each of these levels. Not having all levels properly secured could result in an opponent stealing plays as they

are called or attacking the network to prevent the device from properly working. In order to prevent this from happening, penetration testing this application will be critical and contribute to the overall success of the product.

#### **1.4 User Profile**

The potential users of NPG are a very narrow group of people who are not in the information technology field. For that reason, this program will be easy to use for the coaches and coordinators. If the technology is too complicated to use, the coaches will want to stay with their current ways of doing the things this program is designed to replace and enhance.

##### **Coaches and Coordinators**

Coaches and coordinators will use this program to enhance their play-calling ability and decrease the time between plays. It will also allow them to easily communicate when a timeout is needed by sending a message in the alerts the players and the referee.

##### **Players and Referees**

The other group impacted by this program will be the players and referees. It will allow them to receive the play call or the request for a timeout from the coach without any confusion. Players will not have the ability to select a play; they only receive the plays. Referees will only receive the timeouts when called. This application targets offensive players but will indirectly impact all players in a game, whether they are on offense, defense, or special teams. It does this by changing the pace of the game and how substitutions are handled.

Figure 1 below represents the potential users of the application.

<b>User Profile Form</b>	
Application:	NextGen Playbook
Potential Users	Coaches, Coordinators, Referees, and Players of a football team at the college and professional level
Software and Interface Experience:	Windows based program, User interface is easily navigated Similar layout to a football video game Android-based smartwatch, User Interface quickly displays new play
Experience with Similar Applications:	Play design programs on a computer Video Games such as Madden
Task Experience:	Allow quicker play call Allow more efficient play call Allow for clear, easy communication while not sacrificing comfort.
Frequency of Use:	Daily with practice and during all weekly games during the season Updating in real time per input information Suggested annual training
Key Interface Design Requirements that the Profile Suggests:	Making this program user-friendly and easy to operate to allow for a small learning curve

Figure 1: User Profile

Figure 2 below depicts the application as it appears on the Microsoft Surface.

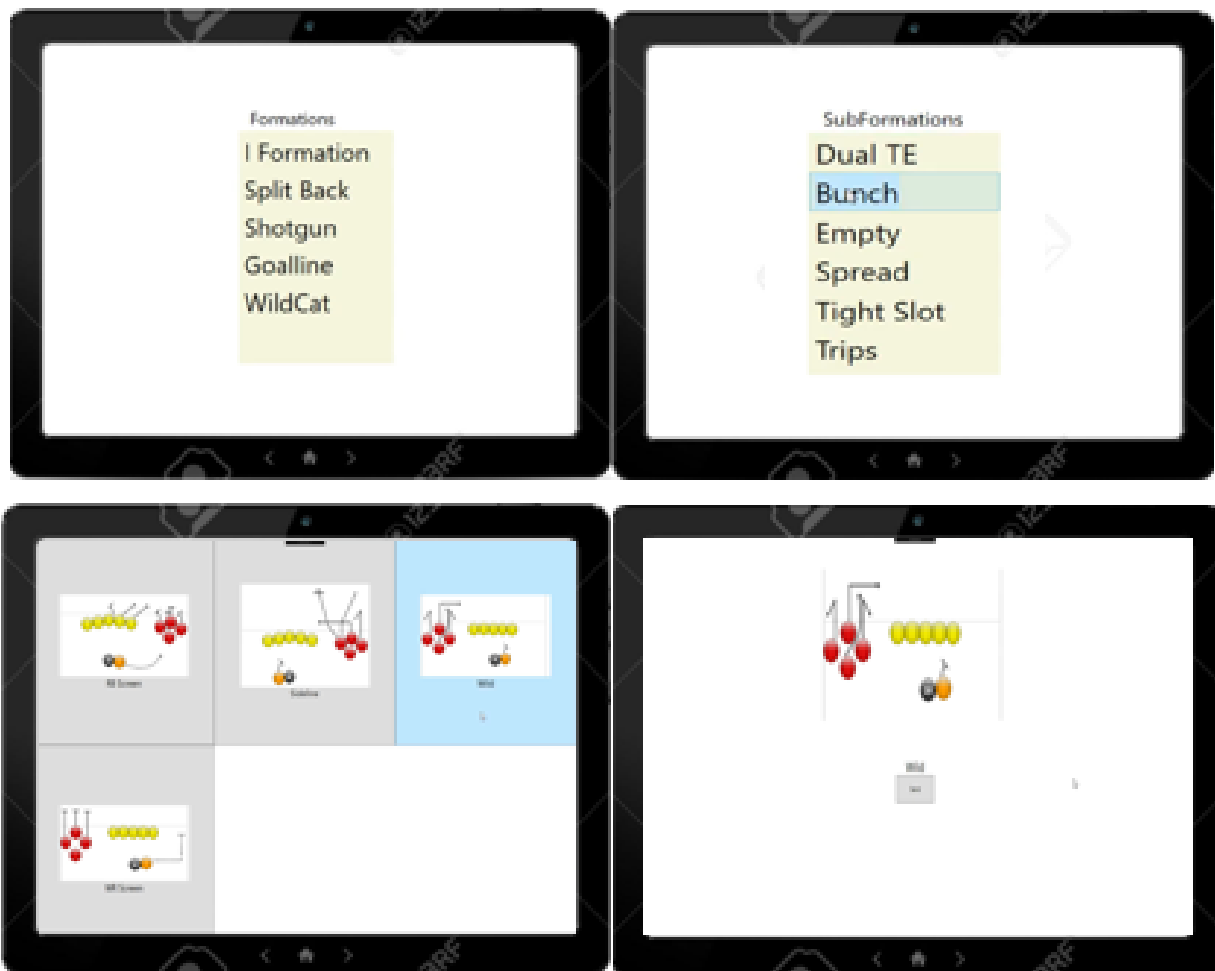


Figure 2: Tablet Screen Shots

## **1.5 Technical Elements**

### **a. Software**

NGP's software will be a custom Android app for receiving play data and a custom C# Windows application for Microsoft Surface for allowing coaches to receive and send play data. ASP.Net Web Services will also be used for transferring data.

### **b. Network**

This will be on an isolated network housed in stadiums wherever it is used in college or at the professional level. The team's mobile server will be connected and established for the devices to communicate with. Separate routers for each team will be on the field to allow the devices to communicate quickly and easily. There will be 12 routers on the field for each team that provides complete signal of the entire field.

### **c. Server**

A Microsoft server will travel with the team to each stadium and will house the database used for this application. There will be a document and process for who can handle this server to ensure needed security level is met. This will connect with the remaining devices needed to allow proper use of NextGen Playbook.

### **d. Backup**

The server will be in a raid to allow for redundancy and to prevent any issues resulting from a loss of data or a drive. A backup mobile server should be on hand at all times to prevent an accident from causing total loss of information and use.

Backups should be taken regularly, and drives should be cycled. Backup drives should be stored in a separate location than the rest of the equipment to proper security.

Figure 3 depicts the architecture diagram for the system.

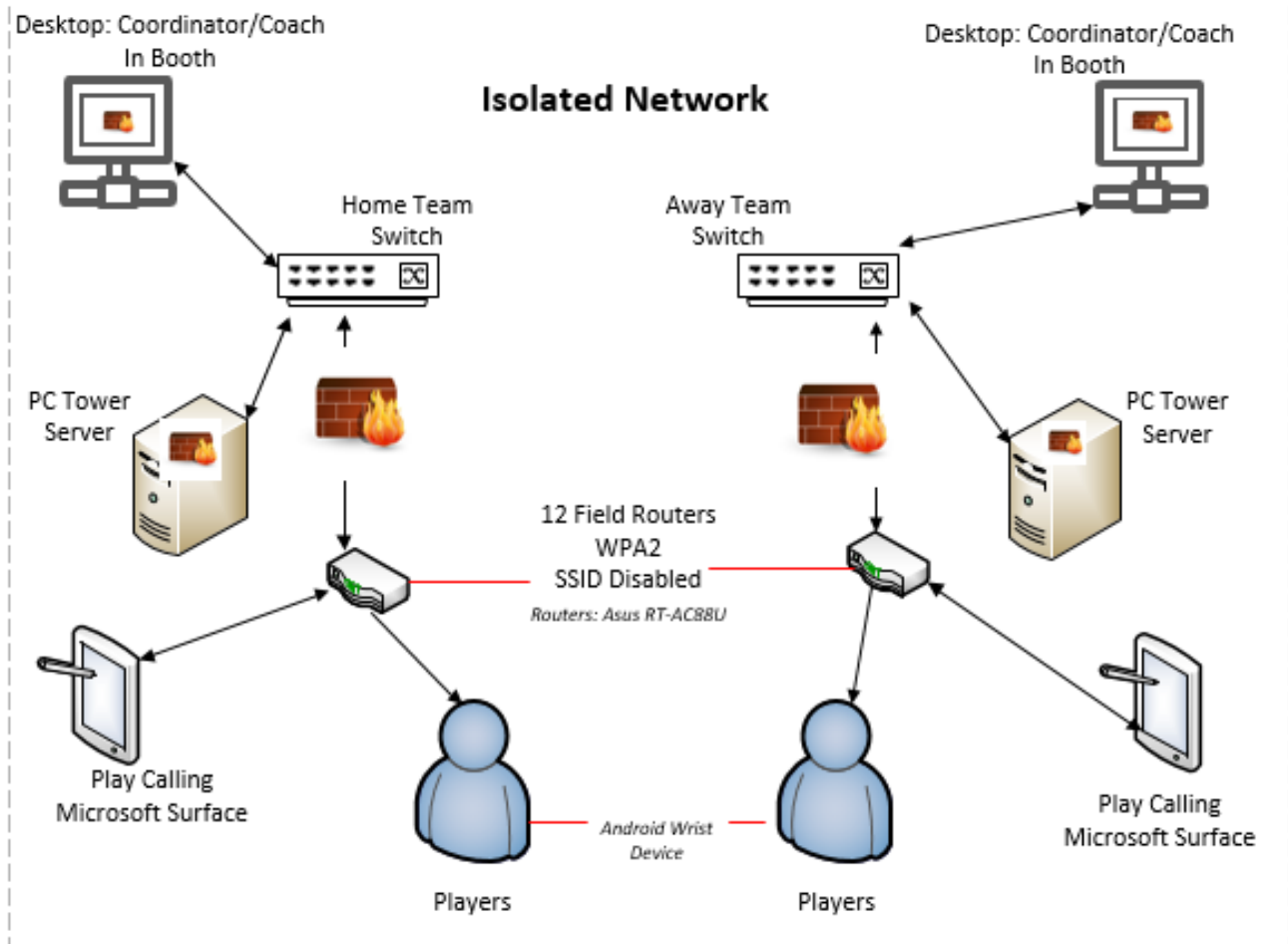


Figure 3: Architecture Diagram

## 1.6 Objectives/Deliverables

Figure 4.1 and Figure 4.2 below depicts the Objectives/Deliverables for the entire project.

First Semester Starting 9/30/2016	9/19-9/25	9/26-10/2	10/3-10/9	10/10-10/16	10/17-10/23	10/24-10/30	10/31-11/6	11/7-11/13	11/14-11/20	11/21-11/27	11/28-12/4	12/5-12/11	12/12-12/31	
Feature and Technology Research	█													
Create Plan to Implement Features		█												
Development for Tablet Prototype		█	█	█	█									
Development for Watch Prototype				█	█	█	█							
Prepare for Presentations and Present								█	█	█	█	█		
Security Analysis & Improvement				Implement Bug fixes and Security Improvements as identified through frequent testing							█	█	█	
Bugfixing				Implement Bug fixes and Security Improvements as identified through frequent testing							█	█	█	
Wrapup Outstanding Items/ Identify Spring Deliverables													█	

Figure 4.1: Fall Objectives/Deliverables

Second Semester Starting 1/9/2017	12/15 - 1/8	1/9 - 1/15	1/16 - 1/22	1/23 - 1/29	1/30 - 2/5	2/6 - 2/12	2/13 - 2/19	2/20 - 2/26	2/27 - 3/5	3/6 - 3/12	3/13 - 3/19	3/20 - 3/26	3/27 - 4/2	4/3 - 4/9	4/10 - 4/16	4/17-4/23	
Update of Current Situation	█	█															
List of Expo Technology Requirements	█	█															
Ensure Weekly Backups are Recorded	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Development for Tablet Prototype	█	█	█	█	█												
Development for Watch Prototype	█	█	█	█	█												
Planning Overall Security Design	█	█	█	█	█												
Implementation of Security Plan			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Testing and Reports for Security Design							█	█	█	█	█	█	█	█	█	█	█
Build Out of Play Calling Options																	
Improving User Interface Layout	█	█	█	█	█												
Improve the Design and Quality of the GUI																	
Practice Presentations																	
Refresh Deliverables	█	█	█	█	█												
Finalize Deliverables																	
Encorporate Additional Feature (Time Permitting)																	
Finalize Overall Project - Practice For Expo																	
Expo Week																	█

Figure 4.2: Spring Objectives/Deliverables

### Fall 2016 Deliverables:

#### 1. Database

- a. Design database and create outline
- b. Create database based on outline

#### 2. Connectivity

- a. Determine best technology for the transmission of data from tablet to wrist
- b. Design and implement connectivity method for database and tablet

3. Program – Surface Tablet

- a. Design and implement GUI for tablet
- b. Design and implement business logic and database connectivity
- c. Develop push functionality to transmit to wrist device

4. Program – Wrist Device

- a. Determine proper wrist device to be used for receiving plays
- b. Design and implement GUI for device
- c. Design and implement receiving functionality

5. Security

- a. Analyze security risks of wireless play transmission
- b. Design solutions to combat security risks
- c. Implement solutions

**Spring 2017 Deliverables:**

1. Database

- a. Complete database and build out with a playbook of plays
- b. Security completely established and tested

2. Connectivity

- a. Connect all the devices on the same network
- b. Secure connection using several levels of security across each aspect

3. Program – Surface Tablet

- a. Finalize design and implementation GUI for tablet
- b. Design and implement business logic and database connectivity

c. Secure the information enclosed in the app

4. Program – Wrist Device (Android)

a. Design and implement GUI for device

b. Design and implement receiving functionality

5. Security

a. Analyze security risks of wireless play transmission

b. Design solutions to combat security risks

c. Implement solutions

d. Test the security of the overall project (repeat C and D as needed)

## 1.7 Budget

Figure 3 below represents the budget for what it would cost to build this project if we were charging for our labor.

No.	Item	Hours Charged	Price Per Hour	Total
<b>Software</b>				
Labor	Application	350	\$100	\$35,000
Testing	Application	50	\$150	\$7,500
<b>Network</b>				
Labor Build	Server	200	\$150	\$30,000
Testing	Server	50	\$150	\$7,500
<b>Misc. Supplies</b>				
Equipment	Routers (20) and Armband (100)	-	-	\$10,500
Server	Hardware	2	\$1,500/each	\$3,000
<b>Licenses</b>				
SQL Server	SQL Server Standard	1	-	\$8,000
Visual Studio	License	1	-	\$13,300
Server 2016 Standard	License	1	-	\$880
<b>Total</b>				<b>\$115,680</b>

Figure 5: Budget

Our budget (Figure 3 above) for NextGen Playbook starts with the building of the teams' SQL server tower, that will perform at the teams' sites. As shown above, the budget table shows this cost of \$1,500 dollars for the server build. This includes the purchase of the tower, motherboard, RAM, hard drives, power supply, processor, and fan; they will purchase two servers to allow for redundancy making the cost \$3,000. The licenses to operate and design these servers, along with the labor to build the program and the network equipment with the labor, will bring the total to an estimated \$115,680.

### Use Case Diagram: NextGen Playbook

Figure 4 below shows how the users will interact with the application.

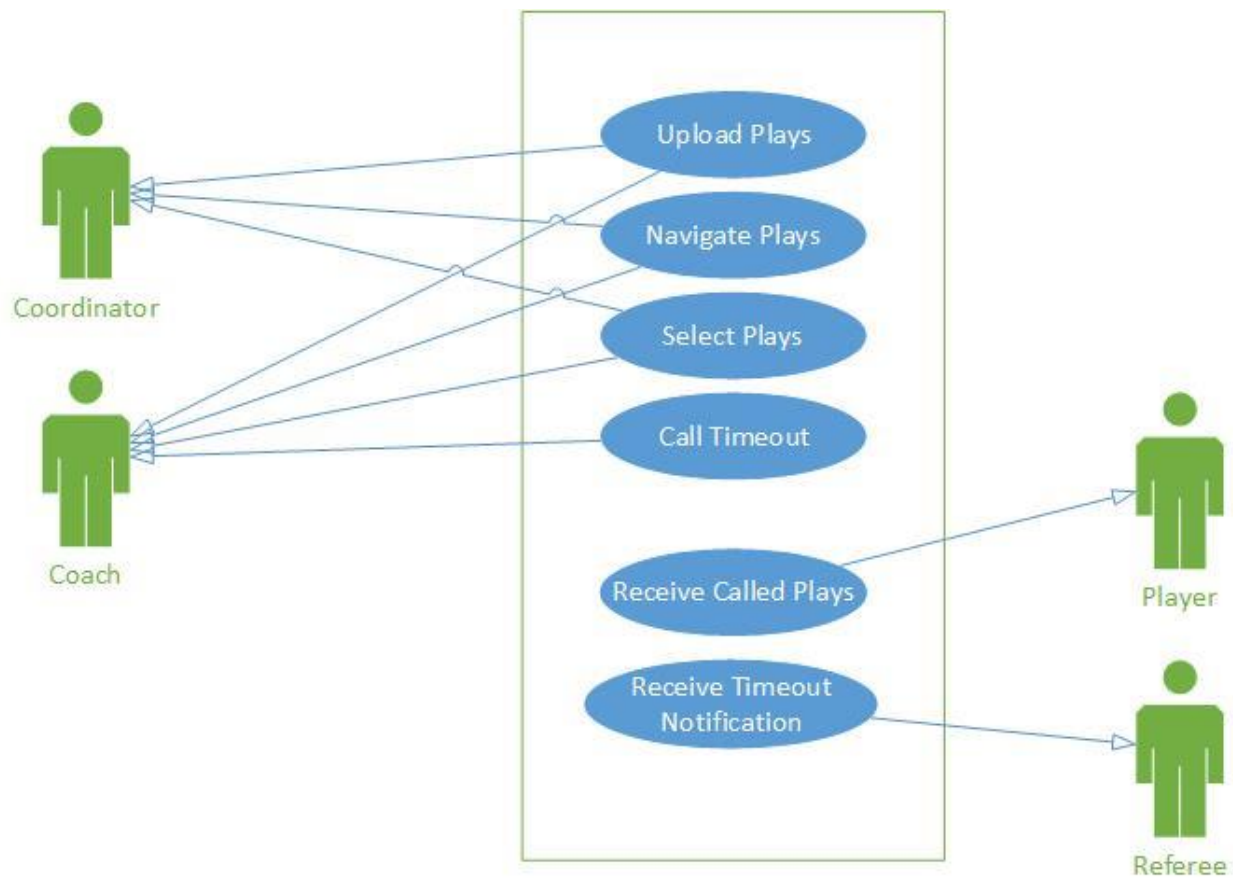


Figure 6: Use Case Diagram

#### 1.8 Use Case Diagram

In the use case diagram shown at Figure 4, we show the relationship between the users and the program. The coaches and coordinators will be able to upload new plays into the program and select plays to send out to the players. The players and referee will be on the receiving end of NextGen Playbook. The player will be receiving the play as it is called, and the referee will be able to be notified if a team calls a timeout.

## **1.9 Testing Plan and Results**

Our testing plan consists of 3 different stages: functionality, security, and user experience. NextGen Playbook has several parts that will require all three of these stages to be implemented. Testing will be carried out on the following: NextGen Playbook server/network, database, devices, and signal for transmission. The devices include the Microsoft Surface, Android watch, and router. To allow for the possibility for this project to make it to the market, we will certainly ensure we answer all questions before their asked through our testing preparation.

### **Plan**

Our testing plan started by ensuring that we will have a secured network before putting in place any other data. Once accomplished, we uploaded our database and program both to the server. After our data is fully established, we need to encrypt this information and put security precautions in place on both the program and the database. We then need to bring in the devices into the program and into the network, but only allow the appropriate devices to connect. Finally, once everything is secure and properly in place, we need to ensure the encrypted signal allows for good performance while simultaneously, maintaining a high level of security. We will then perform a penetration test to all of our settings and analyze how the entire network is working together and check for any security weaknesses. Our testing for functionality was something that is constantly being tweaked throughout the project. The functionality and user experience go hand-in-hand together, and we plan to test these ourselves, as well as have a few outside sources give us feedback on the overall product.

## Report

After we established a private network and were monitoring the items on the network, the server was established and finalized to assist in controlling the devices on the network and the network overall. During our testing to guess both the login and password to both the router and the server were done but not successful; and trying to discover the IP's needed for the server and to search out information to access the server was not possible as well. We then tried to login with random credentials to the program, server, and router and checked falsely created credentials to ensure their roles were working correctly. Failing to gain access to any of the devices, the program, router, or server, and failure to intercept plays as they occurred, left us with understanding that what we had established was a secured overall system. I then decided to allow myself into the system for testing and attempted to do SQL injection on both the database and the application. This came with good results as I was not able to successfully disrupt the application or cause any issues within our database. Furthermore, with the application, I was not able to execute a buffer overflow attack. Denial of service is a hard attack to prevent, however the system should be designed with enough bandwidth to sustain a substantial attack, as well have an IDS implemented to detect any issues, and firewalls to help prevent access. This isolated network will allow for tight detections on the network as everything on the network should be known and in a range of IP's allowed to be used. I was not able to produce a Ddos attack that caused issues.

Test Name	Actions Taken	Results
Wi-Fi Discovery	Used Winhotspot to discover the SSID of the network.	Could not obtain an IP address with this test. Since I know the IP address, I continued with other tests.
IP Exploit	Using Jack the Ripper on Kali Linux, I used several name and password lists to attempt to gain access to the network. I tried using Armitage to gain access to the network using the IP address.	I was not able to gain access through these testing methods. However, since I know the access credentials I continued with additional tests.
SQL Injection	After letting myself into the network with the credentials required, I attempted to use SQLMAP to do SQL injection on our database.	I had taken steps prior to this attempt to prevent SQL injection. These steps did work effectively in this case, as it prevented me from successfully doing the SQL injection.
Admin or Control Take Over	Using Armitage, I continued to try to get admin control or any access.	I was unsuccessful with these attempts.

After performing these series of tests, I found that our network design and our application are effective. I was unable to obtain access, or disrupt the application or network from functioning fully. Advanced testing would be required for the NextGen Playbook project to go live; however, it has proven to be a secure system.

## **1.10 Future Plans**

We have proven that our application works and have created a example of the network we would like to be used with our product. We have applied for a patent on the concept and system design, and we will look to see if there is a market value available for our product. We were successful in our proof of concept, and we look forward to seeing future improvements come to life. The improvements may be done by ourselves, or a company that purchases the product. NGP needs improvement in the user GUI and potentially a restructure depending on the coach's preferences; this would allow it to appear like the current calling cards that coaches or coordinators use. In addition to this, gaining access to the NFL's data to implement and show the play predictability potential use and effectiveness. Lastly, having a wrist device created to be used in the NFL will be required. With these three goals accomplished, this product would be ready to help enhance and improve the game of football as we see it today.

## **Conclusion**

Roger Goodell believes the pace in which the game is being played is a direct cause of the rating issues the NFL has experienced. Per Goodell, the decrease in TV ratings is also related to the number of commercial breaks and the length of the average game. NextGen Playbook makes both issues easier to address. Once completely established, NextGen Playbook will allow for game-speed increases, which will have the ball in play more often to allow for a shorter game as the clock is not stopping as frequently. It is a recipe for success (shorter game time + more plays = more action), which will result in TV rating stabilization and potential viewer growth. Shortening the game will also allow for less commercials and possibly allow for more games to be seen by viewers by adding an additional game time on Sundays.

NextGen Playbook could solve the issue of lower TV ratings; even if it does not, it will improve the play-calling ability and change the pace of the game to allow for more action and better communication, and eliminate current issues that football deals with play calling today.

We started this project because we wanted to establish a solution to enhance the way play calling is performed in college and NFL football. NextGen Playbook is the program we designed to do this, and by sending a picture of the play via an asynchronous push to the end user, it accomplishes this goal. With players wearing the wrist device that receives the pictures sent out, we have eliminated communication issues for play calling, play recognition, and timeout calling issues. Along with resolving these issues of communication, we can increase the game

speed to enhance the play rate on the field and decrease the length of time it takes to play a game. By improving TV ratings and allowing fans to watch more games. NextGen Playbook was created but not without any struggles; our largest difficulty was with our ASUS Zen watch used as the wrist device. In the end, NGP worked, but it took a lot of additional programming due to our droid wear version being on 1.5 and 2.0 not being released until May of 2017. It was a good discovery that we made early so we could find a way around the issue. We look forward to seeing how version 2.0 will enhance our design, it should allow Wi-Fi connections to be enhanced. Outside of this issue we ran into things taking much longer than originally expected, and being far more difficult. Skylar and I did learn a valuable lesson of backing up everything after losing our database over the winter break, we recovered quickly but it ultimately cost us a good amount of time. Unfortunately, we were unable to gain access to team play calling data, which then led us to not implementing the play predictability portion of the original idea. We certainly learned valuable lessons, but we managed to build an original idea, dreamed up while watching a football practice and brought it to life. NGP proves that a system of this design would work, and hopefully one day be utilized by football teams across America.

NextGen Playbook will help ensure football remains the most popular sport in the U.S. and secure its overall dominance of TV ratings while advancing the technology used in the sport. This is all accomplished while not changing the way the game is viewed by fans, other than the increase speed of the game.

## References and Citations

- Belson, Ken. "As Ratings Plummet, N.F.L. Considers Reducing Ads and Length of Games." The New York Times. November 10, 2016. <http://www.nytimes.com/2016/11/11/sports/football/nfl-tv-ratings-shorter-games-fewer-commercials.html>. (Accessed November 20, 2016).
- Smith, Michael David. "Coaches say headset problems happen often outside New England." NBC Sports. September 12, 2015. <http://profootballtalk.nbcsports.com/2015/09/12/coaches-say-headset-problems-happen-often-outside-new-england>. (Accessed November 16, 2016).
- Wassink, Zac. "Ranking The Most Popular Sports in America." TheRichest. February 4, 2016. <http://www.therichest.com/sports/ranking-the-15-most-popular-sports-in-america>. (Accessed November 19, 2016)

## Appendix A: Code Snippets

### Play WS Class

```
[Table("Play")]
public class Play
{
    [Key]
    public int Id { get; set; }
    public String Name { get; set; }
    public int SubFormationID { get; set; }
    public String Type { get; set; }
    public Byte[] Image { get; set; }

    [ForeignKey("SubFormationID")]
    public virtual SubFormation SubFormation { get; set; }

    [NotMapped]
    public String ImageBase64 { get; set; }
}
```

### GetPlaysJSON WS Method

```
[WebMethod]
[ScriptMethod(ResponseFormat = ResponseFormat.Json)]
public Play[] GetPlaysJSON(int subFormId)
{
    ArrayList plays = new ArrayList();
    Play[] playArray;

    using (DFPDBContext context = new DFPDBContext())
    {
        IQueryable<Play> matches;
        matches = context.Plays.Where(row => row.SubFormationID.Equals(subFormId));
        foreach(Play match in matches)
        {
            Play play = new Play();

            play.Id = match.Id;
            play.Name = match.Name;
            play.SubFormationID = match.SubFormationID;
            play.ImageBase64 = Convert.ToBase64String(match.Image);
            plays.Add(play);
        }
    }
    playArray = new Play[plays.Count];

    for (int i = 0; i < plays.Count; i++)
    {
        playArray[i] = (Play)plays[i];
    }
    return playArray;
}
```

## PlayDAO Android Class

```
public class PlayDAO {  
    private NetworkDAO networkDAO;  
  
    public PlayDTO fetchPlay() throws Exception {  
        PlayDTO playDTO = new PlayDTO();  
        networkDAO = new NetworkDAO();  
  
        String response = "";  
        response = networkDAO.request("http://localhost:56560/DigitalPlaybookWS.asmx/GetPlayJson?playId=1");  
  
        JSONObject play = new JSONObject(response);  
  
        int id = play.getInt("Id");  
        String name = play.getString("Name");  
        String imageBase64 = play.getString("ImageBase64");  
        byte[] image = Base64.decode(imageBase64, Base64.DEFAULT);  
  
        playDTO.setId(id);  
        playDTO.setName(name);  
        playDTO.setImage(image);  
  
        return playDTO;  
    }  
}
```

## Appendix B: Poster

Technical Advisor – Russell McMahon

Special Thanks to USAFunds for Donating Two Microsoft Surfaces

University of CINCINNATI

### Technical Elements

- ASUS ZenWatch
- Android Studio
- SQL Server
- Visual Studio
- Microsoft Surface
- Windows Server 2016
- NETGEAR

### NextGen Playbook

Ryan Cash & Skylar Mills  
College of Education, Criminal Justice, Human Services  
School of Information Technology

### Abstract

NextGen Playbook (NGP) is a secure application that revolutionizes the way coordinators are able to call and deliver plays. NGP increases the ability to communicate plays between the sideline and those on the field. NGP improves game speed by eliminating the need to huddle, and it is designed to eliminate the communication issues with radio and current helmet devices.

### Problem

- Communication malfunctions while calling plays, either due to radio issues, crowd noise, or misunderstood calls.
- Communication issues while gaining the attention of the referee.
- Falling TV ratings, due to game entertainment level, game length, and competition.

### Solution

- Communication is enhanced by NGP by allowing every offensive player to know the play call immediately and to see it in picture form. Crowd noise and radio issues would no longer be an issue, but radio would stay in place as is to provide communication redundancy.
- Referee's would also be wearing an armband device.
- TV ratings would stabilize due to the ability to speed up the game

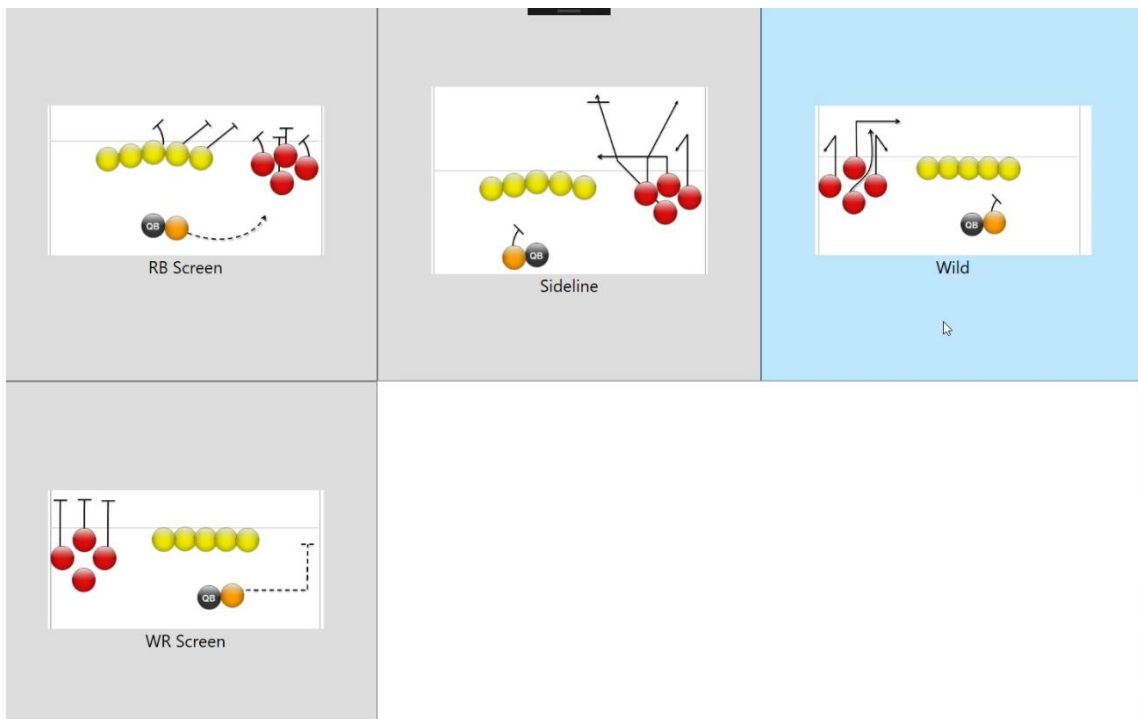
### Design

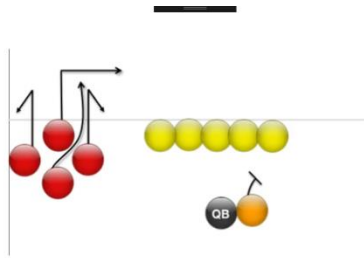
The design section includes a flowchart showing the interaction between the NGP Surface, Business Logic Data, and NGP Android/iOS. The flowchart details steps like User Login, Register New Structure, Choose Formation, Choose Substitution, and Choose Play on the surface, which then interact with an SQL Server Database and various data services (Get Formation, Substitution/Play, Profile CurrentPlay, Get CurrentPlay, Display CurrentPlay) to provide information to the NGP Android/iOS device.

The network diagram shows an 'Isolated Network' for the field. It features a Home Team Switch and an Away Team Switch, both connected to a central Server. The network includes 12 Field Routers (802.11n) and 12 Field Routers (802.11ac). The field is divided into 'Play Calling Microsoft Surface' and 'Players' for both teams. A 'Coordination Coach in Booth' is also connected to the network.

## Appendix C: Screen Shots

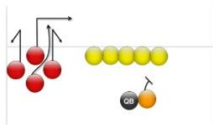
- | Formations  | SubFormations |
|-------------|---------------|
| I Formation | Dual TE       |
| Split Back  | Bunch         |
| Shotgun     | Empty         |
| Goalline    | Spread        |
| WildCat     | Tight Slot    |
|             | Trips         |





Wild  
Send

Previous Play



- Formations
- I Formation
  - Split Back
  - Shotgun
  - Goalline
  - WildCat

## Appendix D: System Architecture Layout and Work Flow Model

