



Michigan State University

Team MSUFCU

Banking with Amazon's Alexa and Apple's Siri

Project Plan

Spring 2017

MSUFCU Contacts:

Emily Fesler

Collin Lochinski

Judy Lynch

Benjamin Maxim

Andy Wardell

Michigan State University Capstone Members:

Qiuning Ren

Ethan Boyd

Kieran Hall

Steven Jorgensen

Will Rudnick

1 Table of Contents

- 2 Executive Summary..... 3
- 3 Functional Specifications 4
 - 3.1 Overview 4
- 4 Design Specifications 5
 - 4.1 Overview 5
 - 4.2 Apple Watch Interface 6
 - 4.3 Android Wear Interface..... 7
 - 4.4 Alexa Interface 8
 - 4.5 Web Portal Interface 9
- 5 Technical Specifications 12
 - 5.1 System Architecture..... 12
 - 5.2 API Communication..... 13
 - 5.3 Security..... 13
 - 5.4 Voice Interface 13
 - 5.4.1 Amazon Alexa..... 13
 - 5.4.2 Apple Watch..... 13
 - 5.4.3 Android Wear..... 14
 - 5.5 Technologies..... 14
- 6 Risk Analysis 15
- 7 Schedule 17

2 Executive Summary

Michigan State University Federal Credit Union (MSUFCU) is a federal credit union founded at Michigan State University in 1937. MSUFCU offers financial services to members of Michigan State University and Oakland University communities, including checking and savings accounts, loans, and insurance. With nearly 230,000 members and over \$3.3 billion in assets, MSUFCU is the largest university-based credit union in the world.

MSUFCU is invested in its members, providing many convenient ways to manage accounts. ComputerLine online banking and the MSUFCU mobile app allow members to have complete access to their funds at any time. To maintain their technological edge, MSUFCU plans to expand their digital banking offerings to wearable and voice-controlled smart device technologies. This new digital banking service will be available on Amazon Alexa, Apple Watch, and Android Wear.

The Amazon Alexa skill, Apple WatchOS application, and Android Wear application will offer a variety of services to members on the go. Members will be able to request information about MSUFCU such as the hours, routing number, and the location of the nearest ATM or Branch. They will also be able to access and manage their account information, account balance, and recent transactions, as well as transfers and payments. All of the above functionality will be supported by voice-control, and account information will be secured by a security code entered by the user at the time of enrollment. MSUFCU will be able to manage and configure the information and services offered by these technologies through an administrative web portal. Additionally, MSUFCU can internally view analytics detailing which questions are being asked more or less frequently. These new digital banking offerings will grant MSUFCU members even greater flexibility in accessing and managing their banking information, no matter where they are.

3 Functional Specifications

3.1 Overview

The primary goal of this project is to prove the viability of voice-controlled smart device technologies for banking. Amazon Alexa, Apple Watch, and Android Wear are the three focal points for implementation, with the objective of expanding digital banking offerings for MSUFCU members. Currently, their banking services can be accessed through both computers and mobile devices. Smartwatches and virtual assistants are the next targets for expansion, leveraging their voice control capabilities and portability to enhance digital banking.

The Amazon Alexa skill, Apple WatchOS application, and Android Wear application will offer the same core set of services, although the Amazon Alexa will offer only a voice interface and not a touch interface, as the Alexa has no screen of its own. First and foremost, members will be able to login to access their accounts through a security code setup on the MSUFCU mobile app. Once authenticated, they can check their account balances, make transfers and payments, view recent transactions, and more. For example, a user may ask Alexa “What is my checking account balance,” to which Alexa may then respond, “You have \$123.45 in your checking account.”

Users can also ask for information about the credit union without being logged in, such as the hours, routing number, or the current loan rates. For these cases, a user may ask “Siri, what are the branch hours?” to which Siri will respond with the branch hours. The exact information offered through this service will be managed and customized by MSUFCU administrators using the web portal. The portal will have an analytics page which keeps track of traffic to different endpoints accessed by the devices. This will allow MSUFCU to tailor the information available based on the most frequently asked questions by members, as well as ensuring that the information can be easily updated.

Between the three voice-controlled smart device technologies and the web portal, there will be a middleware layer. The purpose of this intermediary application is to connect credit union data from MSUFCU databases to the three types of devices. The middleware layer will receive requests from devices and retrieve the required information from the database, before serving up the information in the correct format for the requesting device.

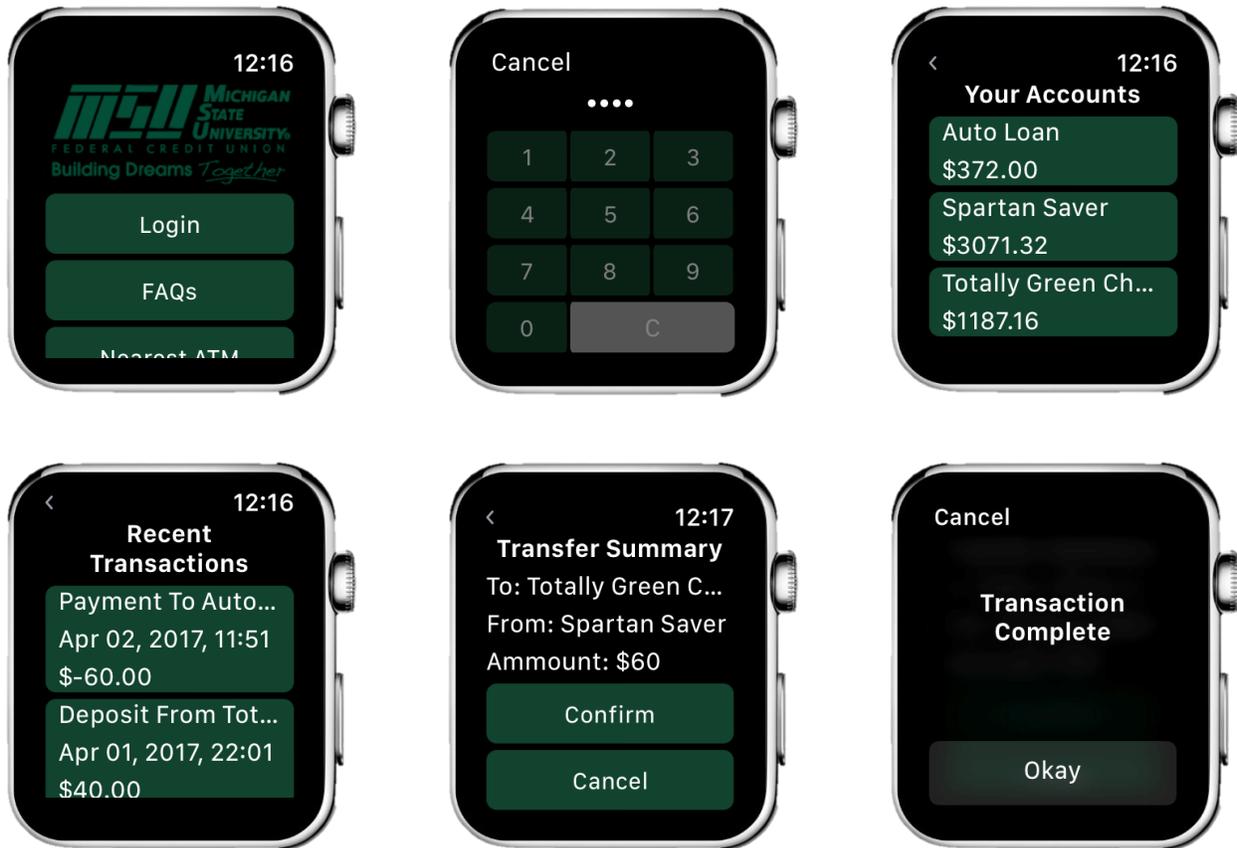
4 Design Specifications

4.1 Overview

Our project spans across many different kinds of platforms and thus results in a diverse set of interfaces. Despite the technical differences, users will have similar experiences across all platforms. For the Apple Watch, users will interface with a WatchOS app as well as Siri for voice commands. For the Android Wear, users will interface with an Android app built for Android Wear and Google Now for voice commands. For Amazon Alexa, users will interface by voice commands and the Amazon Alexa app.

4.2 Apple Watch Interface

The interface for the Apple Watch is a combination of touch controls on the physical watch as well as voice commands with Siri. The WatchOS app is a page-based application allowing the user to complete many of the same tasks they might find on their MSUFCU iPhone or Android app.



From the initial startup, a user can access a set of features like FAQs and finding the nearest MSUFCU ATM or branch. They can also login to their MSUFCU account with a user-specified security code. Once logged in, a user has access to additional features such as: transfers, payments, and viewing account activity. When a user decides to finish their banking activity, they can select logout to end their session. The session will also automatically end after 15 minutes of inactivity.

A user can also access some of their banking information from voice commands using Apple's Siri. After a user initializes Siri, they can ask questions or make requests to the MSUFCU app by simply speaking. A user can tell Siri, "Send with MSUFCU, what are the branch hours?" and the user will get a text response on their screen outlining the hours of operation for their MSUFCU branch. To activate Siri, the user can either hold down the watch crown on the side of the watch, or say "Hey Siri" out-loud.

4.3 Android Wear Interface

The interface for the Android Wear is also a combination of touch controls on the physical watch as well as voice commands with Google Now. The Android app will be a page-based application allowing the user to complete most of the same tasks they might find on their MSUFCU mobile app as shown below.



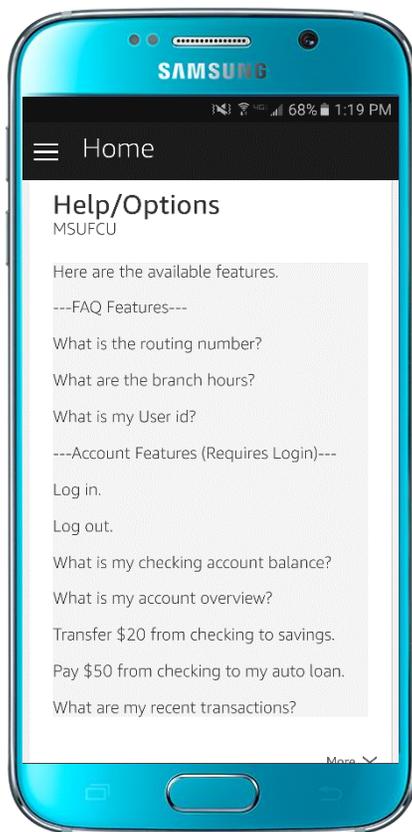
When the user opens the app normally, they are presented with a screen similar to the Apple Watch. Here the user can access the same basic features, or login to access the full capabilities of the app.

The user can also access the app through Google Now, by saying "Okay Google, open MSUFCU." The app will then open and immediately begin listening for the user's next command, such as "what is my account balance?" All functionalities of the app can be accessed through voice commands, from logging in to transferring money between accounts. Even if the app is started without the Google Now launcher, the user can still make the app listen for spoken commands by selecting the 'microphone' button on the home screen.

4.4 Alexa Interface

Unlike the other two platforms, the interface for Alexa is entirely voice operated. To start the applications, users must first download the MSUFCU Alexa skill from the Alexa app on their smartphone. Once the skill is downloaded, users can begin the application by saying: "Alexa, open MSUFCU." Then the Alexa will initialize the main activity of the application and begin listening for the user's next command.

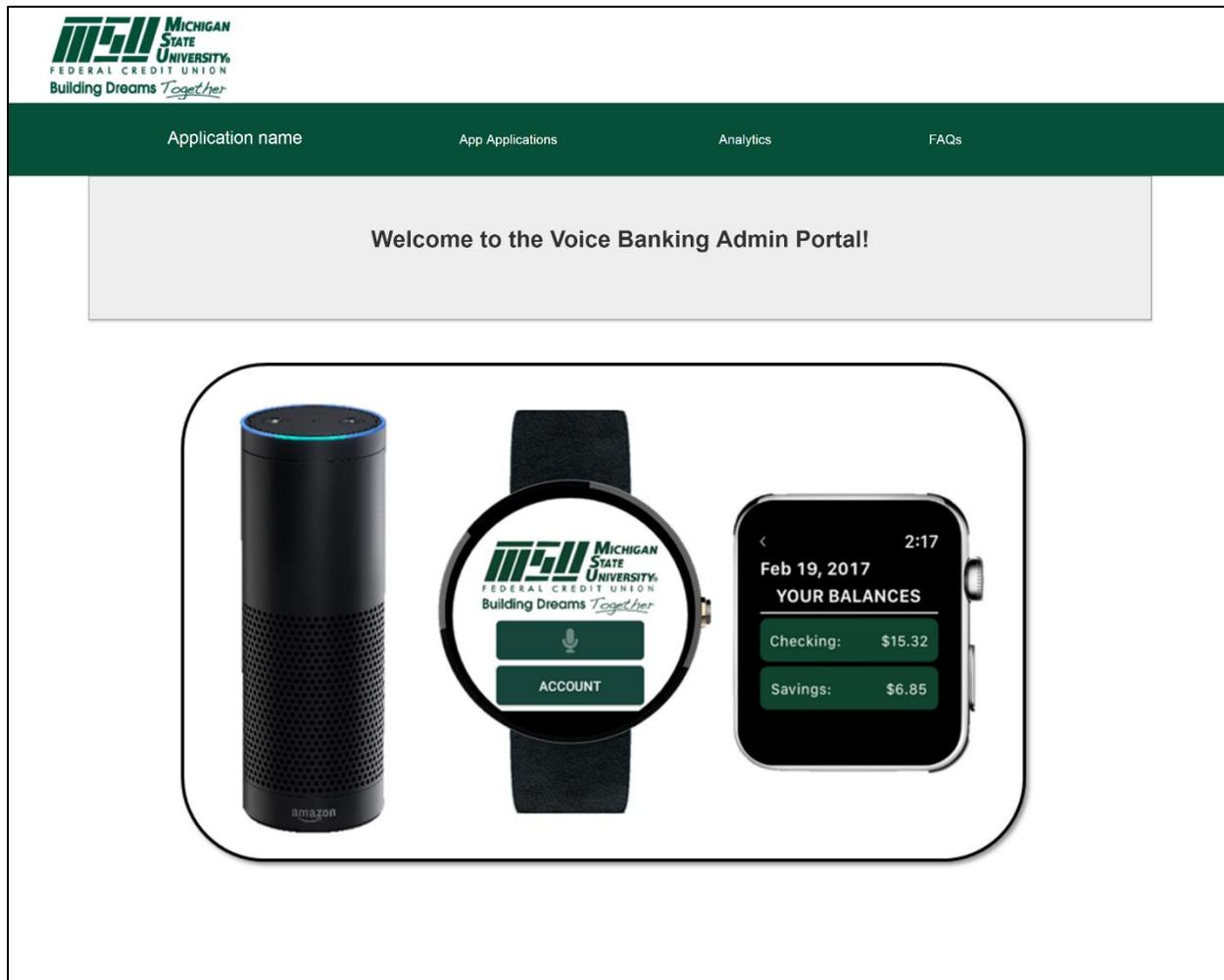
From the main activity, a user can ask simple FAQs to their Alexa or login to their account to access more secure banking information and functions. When a user is done with their banking they can say "Logout" and they will be logged out of their MSUFCU account and the Alexa will terminate the application session. All of these interactions are completed entirely via voice commands. If a user would like to see a list of all available commands within the application, they can say "Help" and the Alexa will send a formatted card to their Alexa app on their smartphone.



Alexa can also interact with the user's account directly, for things such as paying loans, transferring money, or requesting account status information. To access this more sensitive data, users will be required to state an authentication pin before receiving the requested info.

4.5 Web Portal Interface

By using the administrative web portal interface, MSUFCU admins will be able to augment the user experience of the applications without taking the apps offline for construction or pushing updates to the respective app stores. Instead, administrators can quickly change or update answers to FAQs from within the web portal and those changes will be immediately reflected within the applications themselves.



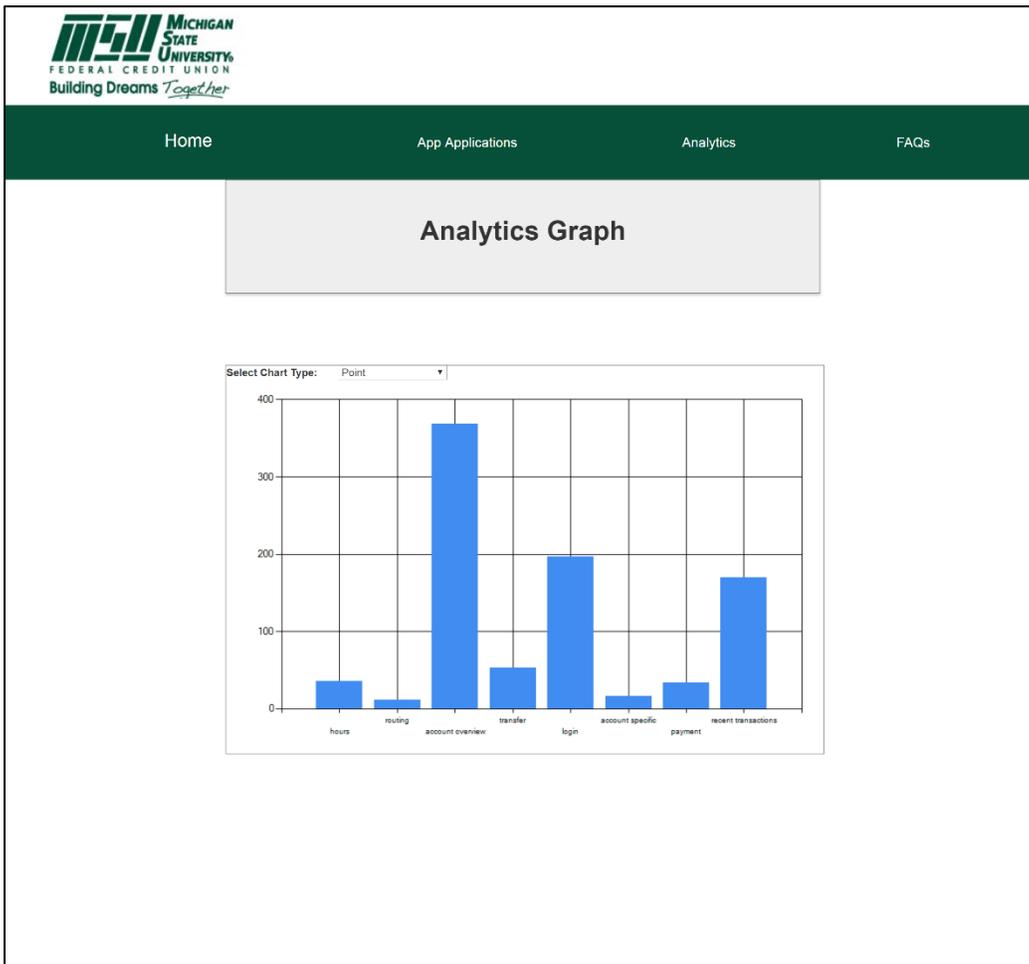
From the web portal, admins can also enable or disable entire features, (like transfers, loan payments, etc.) from being accessible within the applications. These changes are also immediately reflected within the applications across all platforms.

The screenshot displays the Michigan State University Federal Credit Union Admin Portal. At the top left is the logo with the text "MICHIGAN STATE UNIVERSITY FEDERAL CREDIT UNION Building Dreams Together". A dark green navigation bar contains the links "Home", "App Applications", "Analytics", and "FAQs". The main content area features a "Features Control Panel" table with the following data:

Features	Enabled
Transfer between accounts	<input checked="" type="checkbox"/>
Recent Transaction	<input checked="" type="checkbox"/>
Office Hours	<input checked="" type="checkbox"/>
Routing Number	<input checked="" type="checkbox"/>
Account Overview	<input checked="" type="checkbox"/>
Login Page	<input checked="" type="checkbox"/>
Account Specification	<input checked="" type="checkbox"/>
Loan Payment	<input checked="" type="checkbox"/>

A "Save" button is located at the bottom right of the panel.

Admins can also view constantly updating analytics of which questions are being asked and which features are being used by users in the form of graphs and counters.

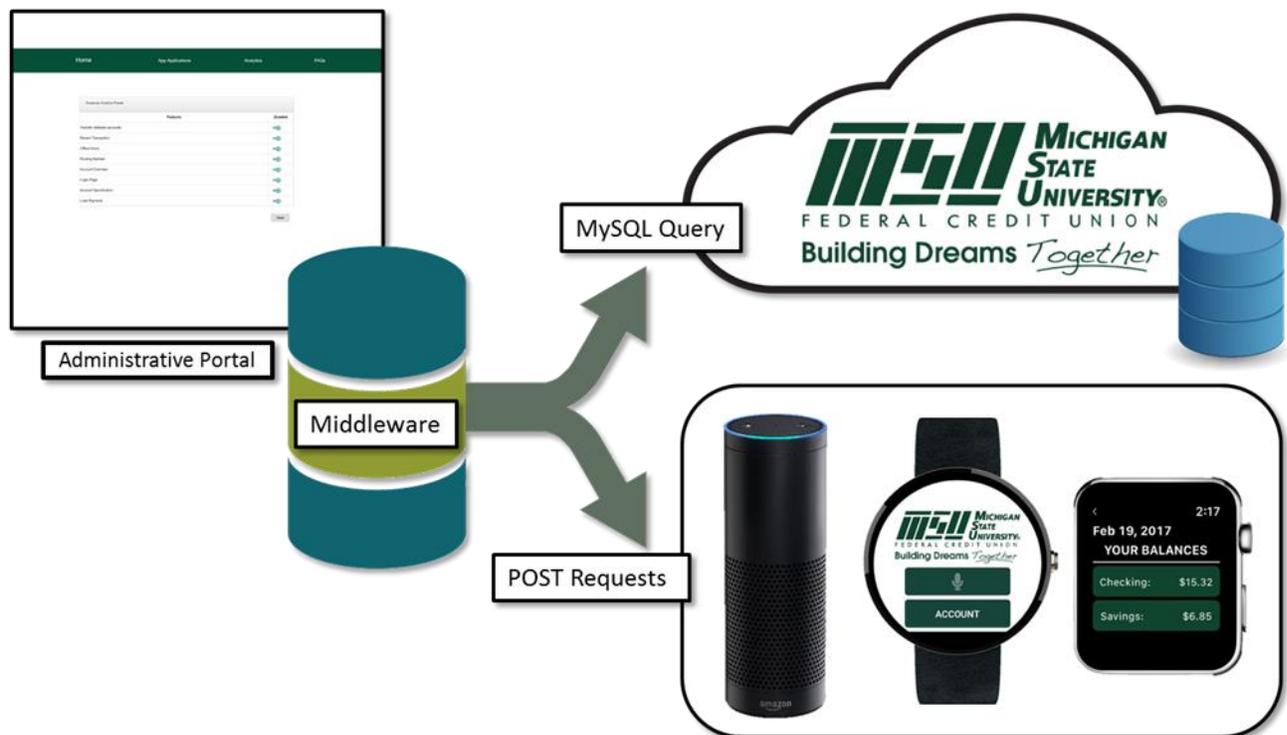


5 Technical Specifications

This section describes the technical aspects and explains each technology utilized to fulfill the functional specifications.

5.1 System Architecture

All three of the supported devices (Amazon Alexa, IOS Apple Watch, and Android Wear Smartwatch) interface with a middleware API. The middleware interprets POST requests from the devices and sends a MySQL query to the database before returning the results to the device as a string. The devices then parse and format the information correctly. The administrative portal also interfaces with the database to provide analytics and toggle features.



5.2 API Communication

Developmental Tools:

Apple Watch: runs on WatchOS and codes on Swift

Android Wear: runs on Android and codes on Java

Amazon Alexa: runs on Fire OS and codes on Python

Administrative Portal: runs and codes on JavaScript, HTML and CSS

Middleware: code is written in PHP and hosted on an Apache server

Communication:

IOS, Alexa, and Android communicate with Middleware class in the web portal using POST Requests. The Middleware communicates with Database using Database Queries.

5.3 Security

When a device successfully logs into an account it is given a unique forty-character hash code specific to that user. All future requests to the API regarding sensitive account information requires that hash code to be passed with the endpoint request instead of the username and password. The hash code expires after fifteen minutes and the user is required to log in again.

5.4 Voice Interface

5.4.1 Amazon Alexa

Amazon Alexa uses the Alexa Skills Kit to parse natural language messages given by the user. The message is turned into a specific intent which is then passed to the Alexa lambda function through a JSON package. The lambda function parses the JSON package and calls functions based on what intent was sent. From there Alexa can call middleware located on a server using POST request. The POST request returns the needed information from the database to the lambda function. The lambda function then turns the requested information into a JSON package, which is then sent back to the Alexa. The Alexa parses the JSON package and reads the response back to the user.

5.4.2 Apple Watch

Within the Apple Watch, Apple's Siri handles natural language processing and generates intent objects. These intent objects have parameters that the app will parse and determine what the user is trying to accomplish. The app will then use this information to call functions that perform the user's request. From within the app, it will send POST Requests to the API. The API handles these requests and returns a string response to the Apple Watch which is then displayed for the user to read.

5.4.3 Android Wear

Within the Android Wear app, Google Now handles natural language processing and returns a string corresponding to the user's command. This string is parsed to determine what the user is trying to accomplish. The app will then use this information in a request to the API, using POST Requests. The API handles these requests and makes calls to the database. The API then returns a string which is packed to JSON by Google Play Services which the app then interprets and presents the request result to the user.

5.5 Technologies

- PHP
- JavaScript
- Java
- Python
- Swift
- MySQL
- Amazon Echo
- Alexa Skills Kit
- Apple WatchOS
- Apple Watch
- Android Wear Watch

6 Risk Analysis

There are several risks that will need to be addressed during the course of this project. The risks and mitigation strategies are discussed below.

Utilizing Siri's Voice Recognition Capabilities

Difficulty: Moderate

Description: Development for Apple's Siri platform has only recently been made open to the public, and the exact capabilities are still unknown.

Mitigation: Work on prototypes to test different tasks that we want the application to be able to do.

Utilizing Google Now's Voice Recognition Capabilities

Difficulty: Moderate

Description: Like Siri, development for Google's Google Now platform has only recently been made open so documentation is minimal.

Mitigation: Work on prototypes to test different tasks that we want the application to be able to do.

Authenticating Voice for Siri, Alexa, and Google Now

Difficulty: Moderate

Description: Accessing sensitive account data requires authentication over voice. None of the team has any experience with voice authentication.

Mitigation: Research possible methods of voice authentication and create a few prototypes to test each method.

Creating Cards for Alexa app

Difficulty: Moderate

Description: Development for Alexa comes with the possibility of pushing information directly to the Alexa app for users to view. However, no one on the team has any experience doing this.

Mitigation: Creating a test application that users can speak to, then find the information they are seeking also available as a card on the Alexa app.

Creating a central API for Watches and Voice

Difficulty: Hard

Description: We need a centralized database and API that all of the different devices will be able to access. No one on the team has made an API before.

Mitigation: Work with client to understand their database schema, and research technologies that can be used for the API. Create a prototype that can retrieve a piece of information from the database and send it to each of the three device types.

Modular design of Watch apps

Difficulty: Hard

Description: The watch apps must be modularly designed so that the administrative web portal can add or remove content such as FAQs. Ensuring this type of modularity in both UI and voice commands may be difficult or impossible with the current capabilities and limitations of watch software.

Mitigation: Research methods of achieving the modularity goal with the current watch technology. Test a implementation of the design on a skeleton app.

7 Schedule

Week 1 (1/22 - 1/28):

- Finish Project Plan and create presentation
- Create detailed mockups
- Look into authentication methods
- Deploy test applications
- Setup rack-mounted server

Week 2 (1/29 - 2/04):

- Project Plan Presentation due 1/31
- Finalize mockups
- Add FAQs to apps
- Add tables based on given schema
- Create Basic UI for Web Portal
- Test voice interfacing on all platforms

Week 3 (2/05 - 2/11):

- Create Basic UI for Watches
- Decide on Phrases for Voice Recognition
- Finalize authentication methods
- Start Siri interface
- Begin adding simple API endpoints
- Work on Alpha presentation and prototypes

Week 4 (2/12 - 2/18):

- Create working prototypes with basic interaction with API
- Working prototype of web portal for presentation
- Prepare for Alpha Presentation

Week 5 (2/19 - 2/25):

- Alpha Presentation due 2/21
- Expand voice features
- Add user authentication in API
- Enhance API integration for all platforms

Week 6 (2/26 - 3/04):

- Finish API Integration for essential features
- Nearest ATM prototype
- Add ability to edit questions within database from web portal
- Finish essential capabilities for all platforms

Week 7 (3/05 - 3/11): Spring Break**Week 8 (3/12 - 3/18):**

- Implement transfers for all platforms
- Build analytics page prototype
- Complete secure login

Week 9 (3/19 - 3/25):

- Finish transfers
- Add help feature to Alexa
- Integrate ability to toggle features across devices
- Finish text-to-speech for Android watch

Week 10 (3/26 - 4/01):

- Finish payments
- Finish recent transactions
- Toggling features within API and web portal
- Testing
- Bug testing
- Preparing for Beta Presentation

Week 11 (4/02 - 4/08):

- Beta Presentation due 4/04
- Start project video
- Start project documentation
- Update Project Plan

Week 12 (4/09 - 4/15):

- Continue project video
- Continue project documentation
- Bug fixing and testing

Week 13 (4/16 - 4/22):

- Continued debugging and final modifications

Week 14 (4/23 - 4/27):

- Project Video due 4/24
- All assets due 4/26
- Design Day Setup on 4/27

Design Day 4/28 6:30am - 2:30pm