# Creating A Minimum Viable Product for Worcester Red Sox's Enterprise Mobile Application

A Worcester Red Sox and WPI Collaboration


**Project Team**

David Rosenstein | drosenstein@wpi.edu
Miguel Duran | mfduran@wpi.edu
Mohamed Barry | mbarry2@wpi.edu

**Project Advisors**

Professor Rob Sarnie | WPI Business School
Professor Sara Saberi | WPI Business School
Professor Wilson Wong | Department of Computer Science
Professor Joshua Cuneo | Department of Computer Science

# Abstract

This project aimed to develop a minimum viable product for the mobile app that was started by last year's MQP team for the Worcester Red Sox. The app they developed laid the framework that this year's team added functionality which allows fans to purchase and access tickets, as well as review and retrieve rewards through the app. This work aligns with the WooSox's goal to enhance the fan experience and encourage more ticket sales. Additionally, this project incorporated an economic analysis to introduce dynamic pricing strategies aimed at optimizing revenue streams for the WooSox. By leveraging regression models and data-driven approaches, the team assessed factors influencing last-minute ticket sales, paving the way for innovative pricing strategies in the app.

# Executive Summary

The Worcester Red Sox (WooSox), a minor league baseball team, has expressed interest in a mobile App for multiple years now. To explore and implement this idea, an MQP Group last year started to build an App for the WooSox; however, they were unable to complete a fully functioning App. This report explains how this year's MQP group took the framework built by last year's group and expanded it into a fully functional mobile App.

Before the project began, some key aspects of the project needed to be decided. First, the process used for software development was Agile Scrum. This is a commonly used development method and was crucial for developing the App effectively. Next, were the environments all the processes would be completed in. Software like Atlassan's JIRA and Microsoft's Visual Studio Code were used. The App itself was built in React Native, used for both its own merits and so that building on the last group would be easier. From there, the requirements for the App were laid out. The most impactful of these requirements was the lack of a dedicated server, but many other requirements informed the design choices for the App. With this in mind, a full process control flow was created to show how users would use the App. Also, while much of the visual design of the App was developed by last year's team, it was adjusted to reflect new design goals. Once all of this was laid out, the App was ready to be developed.

Once development started, the team focused on following goals of each week, using the Agile Scrum methodology. This created a log of what was done each week and how the team felt about its processes and how it should improve. This increase in efficiency can be seen in the team's product burndown chart. With this in mind, a whole App is completed. Each page of the App is covered, going over each of the design decisions and an explanation on how the page functions. These design decisions were made to ensure the best user experience while also encouraging users to want to come to games and spend money at the ballpark.

Another important aspect for this project was the findings on how to improve the business at Polar Park. The first breakthrough for this was a multivariable linear analysis of last minute ticket sales. Using data from 2023, a formula was created that can give an indication of what the last minute ticket sales will look like, based on the weather and day of the week. The second breakthrough was an analysis of the profitability of using dynamic pricing for ticket sales. It was found that the use of dynamic pricing could lead to more profitability for the WooSox and it should be considered, especially on busier days. These findings could be helpful for the WooSox business, and the team hopes that the WooSox will consider them for the future.

Lastly, the team came out of the project with a lot. First is all the experience gained from the MQP. These skills include App development, Agile Scrum, Multi-Variable Linear Regression, and much more. Secondly was where the team believes this project should move in the future. While the team put in a lot of work to create a complete App, it can be pushed so much further and there are many suggestions listed to help with that.

# Acknowledgements

We extend our heartfelt gratitude to a group of exceptional individuals whose guidance and support have been instrumental in the success of our project.

Firstly, we wish to express our sincere thanks to our dedicated advisors from the WPI Business School, Professor Rob Sarnie and Professor Sara Saberi, for their invaluable insights and unwavering support throughout this journey. Similarly, our gratitude extends to Professor Wilson Wong and Professor Joshua Cuneo from the Department of Computer Science, whose expertise and guidance have significantly contributed to our project's success.

We are also immensely grateful to the Worcester Red Sox (WooSox) organization for their collaboration and support. Special thanks to Jason Emmet and Jiwon Choi from the WooSox, whose cooperation and assistance have been pivotal in the realization of our project goals.

Each of these individuals has played a crucial role in our project's development and success, and we are deeply appreciative of their contributions. Their expertise, encouragement, and feedback have not only enhanced our project but also enriched our learning experience.

We look forward to maintaining these valuable relationships and hope our collaborations continue to flourish in the future.

# Table of Contents

# List of Tables

# List of Figures

# 1.0 Introduction

In an era increasingly defined by the digital transformation of user experiences, the sports industry stands at a critical juncture, with digital engagement emerging as a key driver in redefining the dynamics between sports franchises and their fan bases. Recognizing this pivotal shift, the Worcester Red Sox (WooSox), a beacon of baseball tradition and community engagement, embarked on an ambitious and pioneering journey to significantly enhance the fan experience through digital means. This endeavor culminated in the development of the Polar Park App, a project conceived with the visionary goal of crafting an immersive, interactive, and intuitive mobile platform. This application was designed to encapsulate not just the essence of the WooSox and the vibrancy of Polar Park but also to serve as a bridge connecting fans with the game they love in unprecedented ways.

At the heart of this initiative was the ambition to transcend traditional boundaries of fan engagement by leveraging the latest advancements in mobile technology. The Polar Park App was envisioned as a comprehensive digital companion for fans, offering a suite of features meticulously tailored to enrich their engagement with the team and enhance their visit to Polar Park. From seamless ticketing and personalized rewards programs to real-time game updates and interactive park navigation, the app aimed to integrate every aspect of the fan experience into a single, cohesive platform.

This paper delves into the intricate development process of the Polar Park App, offering a detailed narrative that chronicles the challenges faced, solutions implemented, and the innovative technological strategies employed to bring this ambitious project to fruition. It underscores the collaborative efforts of a multidisciplinary team that navigated the complexities of app development, from conceptualization to deployment, with a steadfast commitment to excellence and user satisfaction.

Moreover, this exploration highlights the strategic importance of digital engagement in the contemporary sports industry, where the digitalization of fan experiences can significantly influence a team's relationship with its supporters. By integrating advanced technological solutions, the Polar Park App addresses the multifaceted needs of fans and the organization alike, setting a new standard for how sports teams can interact with their audience in a digital age. Through this comprehensive account, the paper aims to provide valuable insights into the transformative potential of mobile applications in enhancing sports fan experiences, offering a roadmap for future innovations in the field.

In light of the rapidly evolving landscape of sports entertainment, particularly within the minor league baseball sector, there arises an imperative for the WooSox to embrace innovative strategies to maximize revenue and enhance fan engagement. Dynamic pricing, though a common practice in professional sports, remains relatively unexplored in Minor League Baseball (MiLB). Recognizing this untapped potential, the introduction of dynamic pricing into the

organization's ticketing strategy represents a significant first step towards staying ahead in the competitive market. By implementing dynamic pricing, starting with a specific seating section, only for last-minute ticket sales, the WooSox can initiate a process of testing customer reactions to dynamic pricing while simultaneously boosting revenue from general ticket admissions. This strategic move not only addresses immediate revenue optimization needs but also sets the stage for future implementations of dynamic pricing across various sections and ticket types within the app. Through this proactive approach, WooSox can leverage economic analysis to adapt to changing market dynamics and continually enhance the fan experience while ensuring long-term sustainability and success in the sports entertainment industry.

## 1.1 Project Genesis

The relocation of the WooSox to Worcester's Polar Park presented an opportune moment to rethink and revitalize fan engagement strategies. The project was conceived with the dual objective of consolidating ticketing and rewards programs into a unified mobile platform and offering an array of features designed to enrich the game-day experience. This endeavor was driven by a vision to seamlessly integrate operational efficiencies with fan satisfaction, setting a new benchmark for sports entertainment applications.

## 1.2 Methodological Foundation

Adhering to an Agile Scrum framework, the project was characterized by its iterative approach, embracing flexibility, stakeholder feedback, and continuous improvement. This methodology facilitated the integration of complex features, including a dynamic ticketing system, a comprehensive rewards program, and an interactive map of Polar Park. The development process was further enriched by the exploration of dynamic pricing strategies, employing regression models to understand and predict last-minute ticket sales, thus optimizing revenue while enhancing fan accessibility.

## 1.3 Technical Challenges and Strategic Solutions

Throughout the project, the team encountered numerous challenges, ranging from the integration of external APIs for ticketing to ensuring the app's functionality without a dedicated server. Innovative solutions were employed to navigate these obstacles, such as leveraging the react-native-webview for ticketing integration and creating scalable vector graphics for an intricate ballpark map. The introduction of dynamic pricing represented a foray into strategically managing ticket sales based on real-time data and market dynamics, showcasing the team's commitment to leveraging technology for business intelligence.

## 1.4 Future Enhancements

The inception and subsequent launch of the Polar Park App represents not merely the culmination of a groundbreaking project but the dawn of a continuous voyage towards digital

innovation and enhancement within the sports entertainment domain. This pioneering endeavor has effectively laid the groundwork for a future ripe with possibilities, illuminating a path laden with opportunities for technological advancements and enriched fan engagement strategies. At its core, the project has not only highlighted the app's initial successes but has also opened the door to a myriad of future enhancements poised to elevate the fan experience to unprecedented heights.

Central to the envisioned future enhancements is the strategic establishment of a dedicated server, a move anticipated to revolutionize the app's operational framework. This advancement promises to significantly streamline ticketing processes, enabling a smoother, more efficient user journey from ticket purchase to game day entrance. More than just operational efficiency, the introduction of a dedicated server is expected to facilitate a more personalized interaction between the WooSox and its fans, thereby fostering a deeper, more meaningful connection through customized content, tailored recommendations, and direct communication channels.

Equally paramount is the proposed deeper integration with Minor League Baseball (MiLB) data, a step that seeks to vastly enrich the content delivery mechanism within the app. By tapping into a broader spectrum of MiLB data, the app aims to offer fans a holistic view of the game, including real-time scores, player statistics, team standings, and even historical data, thus providing a comprehensive digital compendium of the WooSox universe. This enriched content delivery is not just about keeping the fans updated; it's about immersing them in the narrative of the game, allowing them to engage with the team and its journey on a more profound level.

Moreover, the project has underscored the critical importance of comprehensive user testing in the iterative refinement of the app's usability. Recognizing that the true measure of an app's success lies in its ability to meet the users' needs, future phases of the project will place a heightened emphasis on engaging with the user base through systematic testing protocols. This process will involve gathering feedback, identifying pain points, and continuously iterating on the design and functionality to ensure that the app not only meets but exceeds user expectations.

Looking forward, the trajectory for future work is set to broaden the app's capabilities even further, with an eye towards solidifying its standing as a beacon of innovation in the sports entertainment landscape. This includes exploring cutting-edge technologies, integrating augmented reality features for an immersive stadium experience, enhancing social features to build a vibrant fan community within the app, and leveraging machine learning algorithms to predict and personalize fan preferences.

In essence, the Polar Park App stands at the threshold of a new era in digital sports entertainment, one where the boundaries of fan engagement are continually expanded through technological innovation. The roadmap for the app's evolution is clear—each step forward is a step towards redefining the fan experience, making every game not just an event to attend, but a journey to be a part of. As the app grows and evolves, it promises to remain at the forefront of

the digital transformation sweeping through the sports world, offering fans a front-row seat to the future of sports entertainment.

## 1.5 Navigating the Path Forward

The introduction of the Polar Park App into the sports technology landscape represents a significant stride towards redefining fan engagement for the Worcester Red Sox. By blending technological innovation with a deep understanding of fan needs, the project not only achieved its initial objectives but also laid the groundwork for future advancements. As the app evolves, it stands to offer an increasingly integrated, engaging, and enjoyable experience for WooSox fans, further cementing the bond between the team and its community.

# 2.0 Background Research

To start the MQP, it was important to learn about the organization we are working with and the past work that was done on this project.

## 2.1 The Worcester Red Sox

The Worcester Red Sox, also known as the WooSox are a Minor League Baseball (MiLB) team located in Worcester Massachusetts. They are a AAA baseball team, meaning that they are the highest level of the minors a player can be promoted to before being promoted to a MLB team. They are a minor league team for the nearby Boston Red Sox.

While the team is located in Worcester, the team's history is not rooted there. The organization started as the Pawtucket Red Sox, located in Pawtucket, RI. They existed there from 1970; however, in 2015 the organization wanted a new stadium for the beloved minor league team (Papineau). After a failure to negotiate with both the Pawtucket and Providence local governments, the team decided to move to Worcester, and to build their new stadium in Kelly Square district of Worcester (Anderson). By 2021, the brand new Polar Park stadium was built, and the WooSox played their inaugural season in Worcester. So far, this move has been a massive success. The WooSox see the sixth highest attendance of any MiLB team (out of 120 total teams), and the stadium has been one of many parts to help revitalize the Kelly Square district of Worcester ("Minor League Attendance").

The WooSox are continuing to look to expand this attendance, and they do this through many different methods. The first method is having a lot of special events at the ballpark. The regular events include Friday Fireworks, Tequila Tuesdays, and Pet days. In addition, there are also some once-a-season events, like WooU night and multiple nights highlighting many of the minority communities in Worcester ("WooSox Theme Nights"). These events help build a community around the WooSox and help integrate them with Worcester. The next method that the organization uses to help with attendance is their WooSox Rewards program. This is a system where fans can earn points for going to games and spending money at the ballpark, then redeem these points for prizes. This is a great program for regular ballpark attendees, encouraging them to come to more games and spend more money. A third method of boosting attendance is through the sale of ticket packages. Deals like season tickets, half season tickets, 10 ticket packs, and more can commit some fans to coming to more games which all help boost attendance ("Ticket Memberships"). There are more methods that the organization uses to encourage fans to come out to games, but these are some of the most significant.

## 2.2 Services that the WooSox Work With

As explained, there is a lot going on that the WooSox have to manage. To do this, they get help from a lot of different services. The two of the most important of these services are Tickets.com and FanMaker.

### 2.2.1 Tickets.com

Tickets.com (TDC) is an online platform built with the explicit purpose of buying and selling tickets. They are fully owned by the MLB, and are the exclusive seller of most MiLB tickets, including those of the WooSox ("Tickets.com Inc - Company Profile and News"). While this arrangement may seem somewhat limiting, TDC has all the features that the WooSox are looking for when selling tickets, so this hasn't been a problem. These features include a fully fledged account system for users to buy, view, and sell tickets, the ability to sell multi packs of tickets, the ability to sell half-season and full-season tickets, and much more. This has led to a good relationship between the WooSox and TDC, as well as a streamlined experience for customers (Budnick and Baron).

### 2.2.2 FanMaker

FanMaker is an online platform built with the purpose of creating rewards programs. They have been used by many different organizations to run rewards programs, with a focus on sports teams. They are the company that runs the WooSox Rewards program. Their main features include earning points, prize redemption, and more. In addition to this backbone, they provide two key things for the Rewards program: a website and a mobile app ("Fanmaker Loyalty Programs"). These are only related to the rewards program, and do not connect to TDC or any other service that the WooSox run. However, they are well-built and fully functional for customers to use.

## 2.3 The Need for a (New) App

The WooSox have identified that having an app to enhance the fan experience would be helpful for them. The MQP group were able to identify these key aspects that an app would be useful for the WooSox: ease of ticketing for attendees, the integration of the WooSox Rewards program and other systems under one app, the ability to upsell attendees with concessions and merchandise, another key source of information about the ballpark, and another way of sending reminders and notifications to attendees to push for more tickets (Gustec).

The WooSox already have an app on Apple's App Store, so why would they need a new one? To put it simply - this app doesn't work. It doesn't exist on the Google Play Store, and the version that is on the Apple App Store doesn't work - it is impossible to register and therefore impossible to do anything. This has led to the one star out of five it has gotten in its reviews, and as far as the WooSox team knows, no one is actually using this app ("The Official App of Polar Park"). This app was deemed to not be salvageable, because it was developed by contractors that no longer work for the WooSox, and the codebase for it was not found by last year's team.

There is one app related to the WooSox that sees some use: the WooSox Rewards App. As mentioned above, this app is perfectly functional, and provides all the interactions with the WooSox Rewards program that most fans are looking for. However, this app has no connections

to any other services that are associated with the WooSox. Two of the most important services are Tickets.com and the food vendors. Integration between the existing WooSox Rewards app and these services is impossible because this app is developed by the team at FanMaker, so their development efforts are focused only on the rewards aspects and nothing else.

Since neither of these existing apps can be used, last year's group decided to develop a new app from scratch.

## 2.4 The Previous Group's Work

As implied above, this year's MQP group was not the first group to work on this project. The project was started by another MQP in 2022, done by Qui Nguyen, Bridget Redgate, and Kara O'Neil. Their group laid out a lot of the foundation for the app, which includes choosing to use React Native to build the app for both iOS and Android, choosing a color palette to set a tone for the app, running an initial analysis of how much the app could help increase revenue for the WooSox, and much more. However, one key thing they were unable to accomplish was any connections with the services that the WooSox use. This inability to connect to any of these services left the app in a rough state, where the visuals of the app were well put together and complete, but none of it was functional (Nguyen, Redgate, and O'Neil).

# 3.0 Methodology

## 3.1 Agile Scrum

Agile Scrum is a widely adopted project management and development framework, typically used for developing code. It emphasized flexibility, collaboration, and iterative progress, with an importance on having something to show each cycle. Agile scrum can be divided into 4 sections to come together to a cohesive methodology for developing software: Principles, Sprints, Stories, and Roles.

### 3.1.1 Agile Principles

The core of Agile is built upon a set of principles that are important for a group of people (or squad) to work effectively in an Agile environment. First is the ability to emplace change. Previous development cycles used a system called waterfall, where development teams would lay out the requirements of their software before the creation process, then create the code based on those requirements (Hoory and Bottorff). This has fallen out of favor, because many people who are asking for software to be built (customers) do not know exactly what they want until they see a product that is in progress. To account for this, Agile has built itself around being able to change its requirements throughout the development process (“Agile vs. Waterfall”). Instead of developing an entire software at once, we can develop our software in cycles (sprints). This will allow us to have something to show throughout the process so that customers can refine their needs, and the requirements better match what they are looking for. The next principle to cover is transparency. A key aspect of scrum is that it's important that everyone’s opinion is heard so that the best product can be developed. This means that everyone from customers to developers need to be able to express their opinion. This will allow all of their best ideas to come together and create a shared vision of what the final product will look like. With these two principles, teams will be ready to implement the specifics of Agile Scrum.

### 3.1.2 Sprints

Given the cyclical nature of Agile Scrum, it is important to see what each cycle, or sprint, consists of. To start, the length of each sprint must be determined. Sprints are usually between 1-4 weeks long, and for our MQP we had 1 week long sprints. Within each sprint, there is a planning session, a daily standup, a review, and a retrospective. Before development in a sprint starts, there is a planning session. In a sprint planning session, the team will decide what will get done during that sprint, how long each action item (user story) will take, and who will do it. After the planning session, the sprint will begin. Everyday in a sprint, there will be a daily standup. This is where a team member will talk about their user stories, their current progress in completing them, and any help they may need. This will ensure that each developer can complete what needs to get done for that sprint, and if they need help so that some of their work can be redistributed or put in the backlog. Once the sprint is complete, a sprint review will be held. This is where the product is shown to any stakeholders in its current state to make sure that it is still

meeting expectations. This will allow them to get any feedback that they may have, and ensure that the final product will meet their requirements. Lastly, at the end of a sprint there is a sprint retrospective. This is how a team looks back upon the last sprint to make sure that the team is working well. If anything in the process could be improved, this is the best chance to do so ("Scrum Sprints"). All of these events make up a Sprint, and give developers the ability to make the best products possible.

### 3.1.3 Stories

If sprints are the 'how' the development happens, stories represent the 'what' is being developed. These help organize an Agile squad so that developers know what needs to be done, and when it needs to be done. Each user story is a description of something that needs to be done by a developer. They all take the form "As X, I want Y, so that Z." An example relevant to the WooSox would be "As a WooSox Rewards Member, I want to see how many Rewards Points I have, so that I know when I can redeem them." A developer would read that story and turn it into something functional in the final product. User Stories can additionally have a statement that says "We will know this is done when…" to have a defined completion point ("User Stories"). An Agile Scrum team would create these stories based on what they define as their initial requirements upon starting the project, and they would create more stories as they see fit based on the feedback they get from shareholders and others. From here, user stories are placed into one of two places: the product backlog or the current sprint. Stories in the current sprint are to be completed by the end of the sprint and should have a developer assigned to it. Stories in the backlog are to be completed at a future point and should not have anyone assigned to it. During the sprint planning sessions before Sprints, stories are moved from the backlog into the upcoming sprints to define what developers will be doing in the next sprint ("What is a Product Backlog?"). Overall, stories help define what each developer is doing and keep a squad organized.

### 3.1.4 Roles

Scrum is certainly a lot of work to organize successfully. To facilitate this, each squad will have dedicated roles on who completes certain tasks, so that they can run effectively. The first key role is the Product Owner. The product owner is the role that represents the customer in the squad. They do this by creating and updating user stories in the product backlog so that the requirements of the product match the customer's expectations. They also play a key role during Sprint Planning, as they are the one who prioritizes what a squad needs to do so that they complete what is most necessary first ("Agile Scrum Roles"). Next is the Scrum Master. The scrum master is the person who ensures that all of the processes of Agile Scrum are running effectively. They do this by being the leader of each of the aspects of a sprint. In our MQP, they facilitate the sprint planning, daily standups, sprint reviews, and sprint retrospectives, although different implementations of scum may have them facilitate more or less. While they are the leaders of these meetings, this is mostly to ensure that everyone has a voice and to have someone to go through what each meeting has planned. The last role that will be used for this MQP is

developer. Developers are the ones who are actually writing the code needed to complete the user stories. On smaller teams, roles are usually split, so the Product Owner and Scrum Master are also developers, while on larger teams these are separate roles due to the larger amount of work needed to complete those roles. By defining these roles, it allows an Agile Scrum team to work effectively and efficiently.

## 3.2 Economic Analysis to Increase WooSox Revenue

Before delving into the intricacies of dynamic pricing, it is imperative to understand the broader context of revenue optimization in the sports industry. As sports franchises navigate an increasingly competitive landscape, maximizing revenue streams while enhancing fan experiences has become a primary objective. This section explores the strategic importance of dynamic pricing within this context, highlighting its role in adapting to market dynamics, optimizing ticket sales, and fostering deeper fan engagement. Through a comprehensive analysis of dynamic pricing strategies, the team aimed to provide valuable insights into how organizations like the WooSox can leverage these approaches to achieve their revenue goals while delivering unparalleled value to their fan base.

### 3.2.1 Dynamic Pricing

As the team delved into the intricacies of dynamic pricing, a sophisticated and adaptive strategy emerged as a pivotal component in optimizing revenue streams and enriching the overall fan experience. Dynamic pricing involves the real-time adjustment of ticket prices based on a multitude of factors, including demand, historical sales data, team performance, time until the event, and other relevant market dynamics.

### 3.2.2 Dynamic Pricing in Sports

In the context of sports, where demand can fluctuate based on factors like team performance, rivalries, and the significance of the game, dynamic pricing enables organizations to remain agile and competitive in the ticketing market. The implementation of dynamic pricing strategies in sports is not only about revenue optimization but also about creating an inclusive and engaging fan experience, aligning with the evolving preferences and expectations of today's sports enthusiasts.

### 3.2.3 Dynamic Pricing for the WooSox

For the WooSox, dynamic pricing becomes particularly intricate, considering the unique characteristics of minor league baseball, the team's fan base dynamics, and the distinctive features of Polar Park. The implementation and fine-tuning of dynamic pricing strategies for the WooSox involve a strategic blend of data analysis, fan engagement considerations, and responsiveness to the dynamic nature of the sports industry.

### 3.2.4 Dynamic Pricing Implementation

Dynamic pricing can elicit a range of responses from fans, both positive and negative, making it a crucial consideration for sports teams. While it is a common practice in the realm of professional sports, it is relatively new in Minor League Baseball. Taking these factors into account, the team decided to implement a modest adjustment, introducing a $1.00 increase in last-minute ticket sales. This adjustment is applied during a specific timeframe, ranging from one to three hours before a game and extending to one to two hours into the game.

### 3.2.5 Regression Analysis

In the pursuit of economic analysis to maximize WooSox revenue, regression analysis was conducted through Microsoft Excel. The team identified several variables believed to impact last-minute ticket sales, including rain precipitation, forecasted temperature, weeksplit (day of the week or weekend), and the season. Each variable's individual impact was assessed before incorporating them into a comprehensive model. By leveraging regression analysis, insights into the intricate interplay of these factors were gained, paving the way for the implementation of dynamic pricing strategies. This approach allows the WooSox to adjust ticket prices in real-time based on demand and market dynamics, ultimately maximizing revenue while enhancing fan accessibility and satisfaction.

# 4.0 Software Development Environment

## 4.1 Project Management Software

The main software used for managing the project was Atlassian's JIRA. JIRA is an online application which is meant to help teams that are using Agile Scrum in their workflow, as what was done for this MQP. This tool allowed the team to keep a detailed log of all the stories so that it is known which stories are complete, which are in progress, and which have not been started. In addition, it allowed us to divide all our stories into Epics, which was crucial to keep the needs of the team organized so that stories could be prioritized. JIRA was chosen over common alternatives due to its robust features tailored specifically for Agile Scrum methodologies, its user-friendly interface, and its seamless integration with other essential project management tools such as Confluence and Bitbucket. This made JIRA the ideal choice to streamline our project management and ensure efficient collaboration within the team, ultimately contributing to the successful completion of our MQP.

## 4.2 Programming Environment

An integrated development environment (IDE) is a software application that helps developers create software more efficiently. IDEs provide many capabilities such as software editing, building, testing, and packaging ("What is an IDE"). In the past, the text editor was used to write code, but IDE's provide functionality beyond text editing. Developers utilize IDE's for a plethora of reasons, including code editing automation, syntax highlighting, intelligent code completion, refactoring support, local build automation, compilation, testing, and debugging ("What is an IDE").

Visual Studio Code was the IDE utilized by the developers working on the Polar Park App. Visual Studio Code offers precise support and specialized utilities that significantly streamlined the development process, enhancing overall efficiency. Some advantages of Visual Code include excellent support for multiple programming languages, high customizability, and being fast and efficient with a small footprint (Muhammad Mir). Support for multiple programming languages for apps such as the Polar Park App, because multiple languages are used to create a mobile application of this scale. Efficiency is essential for developers as it streamlines development and provides a smoother programming experience.

### 4.2.1 Version Control

Creating this application entailed multiple people working on the same codebase at once due to its complexity. The tool utilized to best accomplish this was Git version control. Git version control allows developers to not only collaborate, but also allows developers to track different versions of codes, and its modifications. Git enables teams to have multiple copies of the codebase independent of each other ("What is Git version Control?").

Version control, often termed as source control, is an indispensable software tool crucial for tracking changes, streamlining code integration, and effectively managing diverse project elements like design components, data, and images (“What is Git version Control?”). Its absence could substantially complicate collaboration among teams working on a shared code base. Within this structure, team members function on distinct segments of the codebase, commonly known as branches, enabling more efficient and organized collaborative development.

In these branches, developers can commit, stage, and add code, providing visibility for others working on the codebase to view and make alterations. Git stands out, utilized by 87.2% of developers over other source control systems like CVS, Mercurial, and Perforce due to its adaptability, speed, and the stability required in fast-paced markets (“What is Git version Control?”). With version control, developers experience seamless communication and expedite their software delivery process.

### 4.2.2 App Framework

This application was built using React Native CLI 0.70.3, utilizing the Hermes JS engine. The decision to use the CLI rather than the Expo Go framework was mainly that the React Native CLI allowed access to native device features, which we thought was more essential for better user testing. The React Native CLI also provides more control over project setup and configuration (“Expo vs React”). Based on this decision, the team used Android Studio Giraffe 2022.3.1 along with Xcode 15.0.1, each respectively used for launching Android and IOS local environments. In order to launch each simulation in their environments, Android required OpenJDK 11 and IOS required CocoaPods 1.13.0 and Ruby version 2.6.10. The IDE used was Visual Studio Code 2022 version 17.7.

For a better programming experience, the primary programming language used for this application was Typescript 5.2.2. Typescript is a high-level programming language deployed by Microsoft in 2012 (“The TypeScript Language”). Typescript is a superset of Javascript that adds additional syntax and supports tighter integration with the editor (“The TypeScript Language”). Typescript was chosen as the primary language over Javascript for a multitude of reasons. Mainly, Typescript can be strongly typed, while Javascript is only dynamically typed, Typescript allows for better abstraction through its implementation of interfaces and is more readable and maintainable than Javascript (Cameron McKenzie). At run-time, typescript converts to Javascript, and is then interpreted by the Hermes engine.

# 5.0 Software Requirements

## 5.1 Determining Software Requirements

Many of the software requirements for the Worcester Red Sox's Polar Park application were provided by members of the Worcester Red Sox organization. Specifically, the rewards section of the application, along with the idea for idea mapping services within the application. Some aspects of the app are nuance requirements the team found during the development process. The most notable of these is the interactive map of the ballpark and parking garage, allowing fans to better navigate the park. An original requirement the team had was incorporating ticketing into the application, allowing users to buy tickets, and see where they are seated on the map; however, this requirement was later updated to reflect a change in the App's approach for ticketing (see section 5.2.1)

## 5.2 Functional and Non-functional Requirements

The requirements for developing this app, mainly centered around interactive mapping, and the rewards program. The functional requirements are as follows:

1. The interactive map should show users where concessions, seating, and parking is.
2. While the user is using the map, they should see exactly where their seats are.
3. When a user signs in, their points should be displayed on the dashboard.
4. The rewards section of the application should display all rewards available, along with those that are features.
5. Users should be able to redeem points

The non- functional requirements mainly surround user experience. The nonfunctional requirements are as follows:

1. When a user is signed in, their points should update live.
2. The interactive map should load quickly.
3. When using the map, user's seats should be highlighted distinctly.
4. Parking garages, and concession stands, should distinctly be displayed on the map, highlighting different colors so users could easily find these locations.
5. Rewards on the app should be updated consistently.

### 5.2.1 App Server

Quickly upon development, it became clear to the team that a dedicated server for this app would be extremely helpful for creating a great system. Unfortunately, the WooSox decided early on that this server would not be possible for them to host. This is because there is not a large amount of IT experience among the WooSox team, and server maintenance would be impossible. This caused some significant hurdles for the team, and required the modification of some key functional and design requirements that were made earlier in the process.

First, and most importantly, this means there is a heavy restriction on the ticketing aspect of the app. The app is unable to call the tickets.com API directly due to security concerns (see section 9.2), and without the ability to call the tickets.com API, the app is unable to fetch any data from it. The best solution for this is to use the tickets.com website, which is outlined in section 9.1.2.

Second, it means that users would have to sign into separate accounts for WooSox Rewards and tickets.com. This may be a major point of confusion among consumers, as despite attempts of communicating from WooSox, people are already getting confused between these two services. The combination of them within one app will likely only lead to more confusion, as fans try to sign into their WooSox Rewards account with their tickets.com credentials and vice versa.

Thirdly, this app lacks the capability to introduce innovative features to enhance the rewards program. During the initial brainstorming sessions for the app, the team had envisioned a system where points would be automatically granted upon ticket purchases, among other similar actions. Implementing such actions would have seamlessly integrated the rewards experience with the overall app functionality, but unfortunately, this behavior necessitates the use of a dedicated server for security considerations. Consequently, without access to this dedicated server, our app is constrained to replicating the same functionality as the website, limiting its potential for innovation in the rewards program.

Lastly, it limits the available additions that future groups are able to do. Most modern apps have their own servers. It's seen as a crucial part of many apps and its part of what makes them so convenient in the modern world. This functionality not existing for the Polar Park App will be inherently limiting and holds back much of the potential this app has for fan experience.

## 5.3 Initial User Stories

To structure the initial user stories for this application, the team divided the stories by flow. The flow refers to the flow of actions performed by the user within different sections of the application.

| **Interactive Map Flow** | |
|---|---|
| User Stories | Points |
| As a user I want to be able to view all of the different seating sections in Polar Park. | 5 |
| As a user I want to be able to distinguish between each seating section easily by viewing the map. | 5 |

| | |
|---|---|
| As a user I want to be able to interactive with a map of Polar Park and navigate between the different seating levels in Polar Park | 13 |
| As a user I want to be able to zoom in and out of different sections of the map | 13 |

Table 1: Initial Interactive Map Flow User Stories

| **Ticketing Flow** | |
|---|---|
| User Stories | Points |
| As a user, I want to be able to pull up my ticket on the application so I can quickly access it | 1 |
| As a user, I want to be able to purchase tickets in advance to a game so I don't have to do it at the game | 3 |
| As a user, I want to see the team's schedule for the season so I can make plans accordingly | 3 |
| As a user I want to be able to see all of the tickets I have purchased in one succinct location | 3 |
| As a user I want to be able to purchase season tickets | 5 |
| As a user I want to be able to purchase tickets for a multitude of people in cases where I plan on having an outing or holding an event | 5 |

Table 2 Initial Ticketing Flow User Stories

| **Concessions Flow** | |
|---|---|
| User Stories | Points |
| As a user I want to be able to view all of the concession stands available in Polar Park | 1 |
| As a user I want to be able to view all of the food items that each concession stand sells | 1 |
| As a user I want to be able to filter concession stands by food item, rather than going through each stand one by one to find a food item | 5 |

Table 3 Initial Concessions Flow User Stories

| **Rewards Flow** | |
|---|---|
| User Stories | Points |
| As a user, I want to be able to register for a WooSox Rewards account | 3 |
| As a user I want to be able to login in to my WooSox Rewards account using my email and password | 1 |
| As a user I want to be able to reset my password on the login page in case I ever forget my password | 3 |
| As a user I want to have a dashboard that displays the current points number of rewards points I have | 1 |
| As a user I want to be able to view my progress towards the next rewards available to me on the dashboard | 5 |
| As a user I want to be able to see all of the featured offers and rewards that Polar Park is giving to its rewards members | 3 |
| As a user I want to be able to scan QR codes on receipts and have those points be directly reflected in the application | 5 |
| As a user I want to be able to see a leader board that displays which user's hold the most points in the rewards program | 3 |
| As a user I want access to a FAQ Screen, that would allow me to see commonly asked questions | 1 |
| As a user I want to be able to update my account information and be able to change my profile photo | 3 |
| As a user I want to be able to see a list of all rewards available | 1 |
| As a user I want to be able to sign out of my WooSox Rewards account | 1 |

Table 4 Initial Rewards Flow User Stories

| **Services Flow** | |
|---|---|
| User Stories | Points |
| As a user I want to be able to contact park security services at any time within the application rather than going and finding someone in the park | 1 |
| As a user I want to be able to know what to do in case of inclement weather situations | 1 |
| As a user I want to be able to see where the parking garages are near me | 1 |

| | |
|---|---|
| in the park | |
| As I user I want to be able to see how many seats are available to me in each of these parking garages | 1 |

Table 5 Initial Services Flow User Stories

## 5.4 Initial Use Cases for Application

The beginning of the project was highlighted by the importance to a functioning ticketing system within the application. Before all, it was important that the app needed to be something that users would want to use. The main system within the app that was seen as important to facilitate this would be a fully functioning ticketing system. Many fans going to a WooSox game would appreciate the ability to just show their tickets on their phones without needed to print them out or show some sort of PDF or other document that they downloaded, and a fully integrated ticketing system within the application would be the best way to facilitate this. The other important use case that was deemed important was the integration of the WooSox rewards program. The marketing team at the WooSox were interested in expanding this program to more fans. By combining the ticket purchasing and viewing experience with the rewards program all in one location, they hoped that there would be an increase in awareness of the rewards program. This increased awareness and utilization, they hoped, would be accompanied with increased ticket sales and increased revenue. The other initial use cases for the app included a map of the ballpark, a page for a fan to order food to their seat, connections to the WooSox social media pages, easy accessibility to parking, and a system for their parking shuttle service. The application also includes an in park services section that provides information to users about what users should do in case of inclement weather, direct contact to park security and information about the park. However, the importance of ticketing and rewards made those two sections of the app the priority for the team.

## 5.5 Interface Mockups

To ensure a cohesive and user-friendly application structure, our team utilized Figma, a versatile platform for creating mockups and designing application screens. Leveraging both the existing framework from last year's teams and developing an entirely new set of screens specifically tailored for the rewards program, we crafted comprehensive wireframes for each page. For reference, these wireframes are available in Appendix C. Figma empowered the team to not only envision but also design and refine the user interface, ensuring a seamless and visually appealing experience for our application users. An example mockup can be seen below.

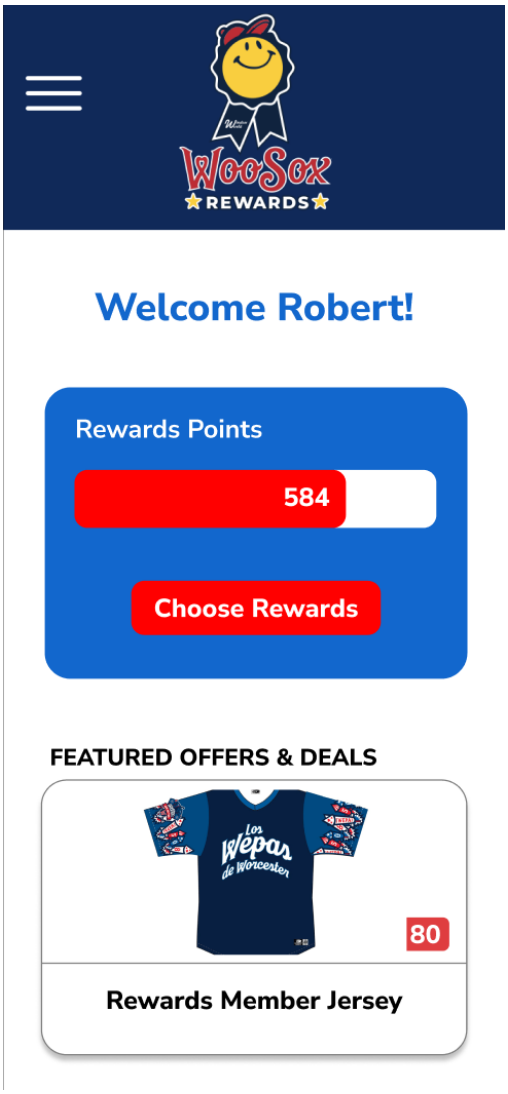



Figure 1 Rewards Dashboard Mockup

# 6.0 Process Analysis

To ensure a seamless user experience in the Polar Park app, the team meticulously crafted a comprehensive process flow chart. This detailed chart not only captures the existing functionalities of the app but also outlines an optimal sequence for integrating future enhancements.

The process flow chart takes into account forthcoming improvements, exemplified by the integration of the Tickets.com website within the app itself in the current Tickets section. While adhering to the fundamental structure outlined in last year's MQP Report, subtle adjustments were made to accommodate prospective interface enhancements within the app.

Figure 2 visually represents a portion of the features on the app, some of which are not implemented yet. For instance, the tickets portion is not currently implemented as the section sends the user to the website version of Tickets.com since the team was not given access to integrate the organization's features into the app. The figure also highlights the concessions section, which is in its current state. The team anticipates next year's MQP group will work with the food and beverages' organizations to properly integrate concessions online orders into the app. Figure 2 also portrays some sections the team considered for the app, which are not fully integrated. These sections being services for the overall ballpark experience, as well as news and events regarding the WooSox organization, as a whole.

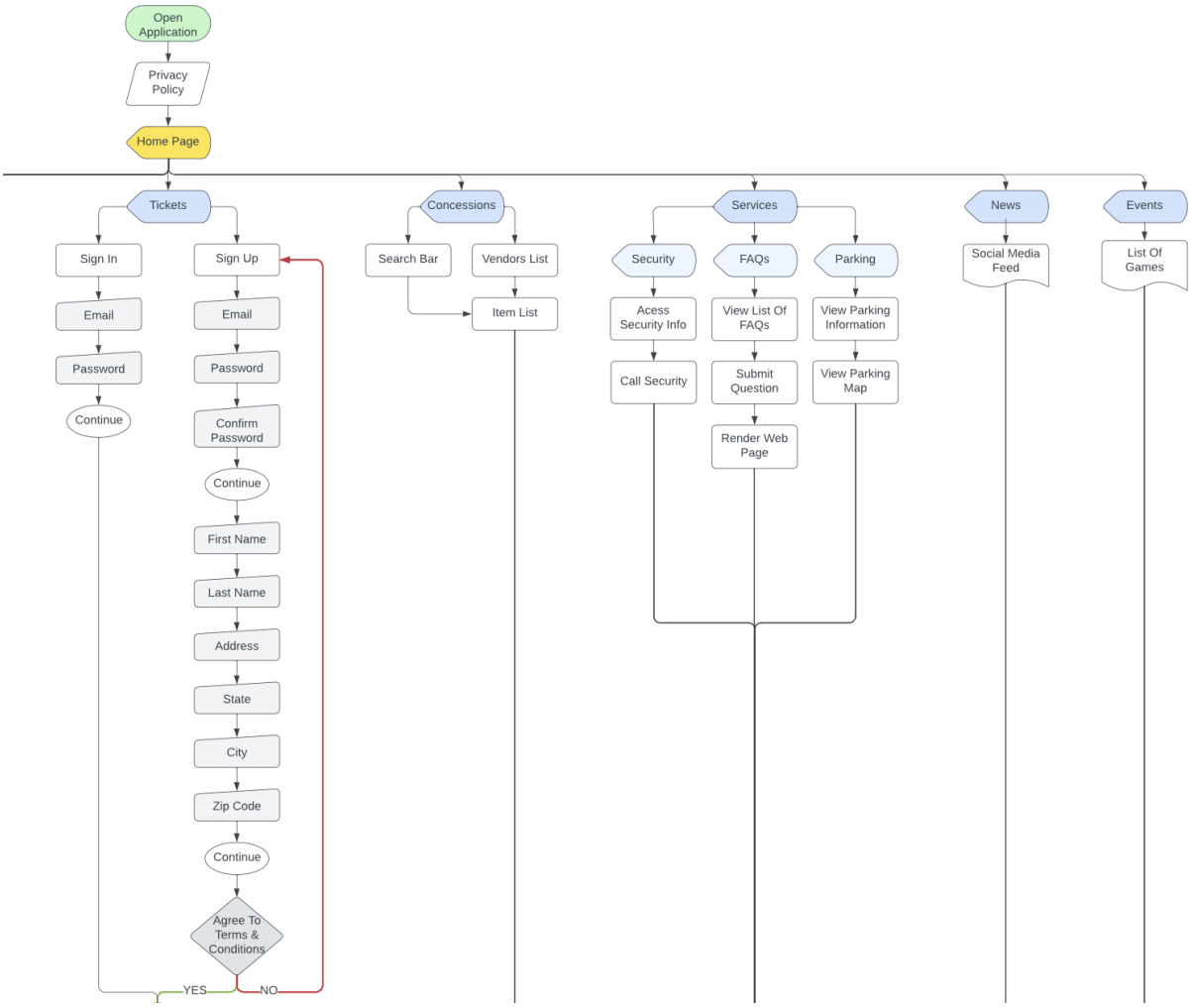



Figure 2 Process Flow Chart - Future Additions

The initial steps of the rewards program are depicted in Figure 3. These steps encompass every process related to user creation within the app itself. The ballpark map is also in a functional stage; the team managed to develop the ballpark map fully based on the organization's website. Further details regarding the ballpark map can be found in section 9.1.3. The app settings, as shown in the figure, are in a preliminary state, with some features being nonexistent. For instance, the language option has not been implemented since the feature is not yet available. The team contemplated adding multiple language availability for the organization to consider in future app reviews.

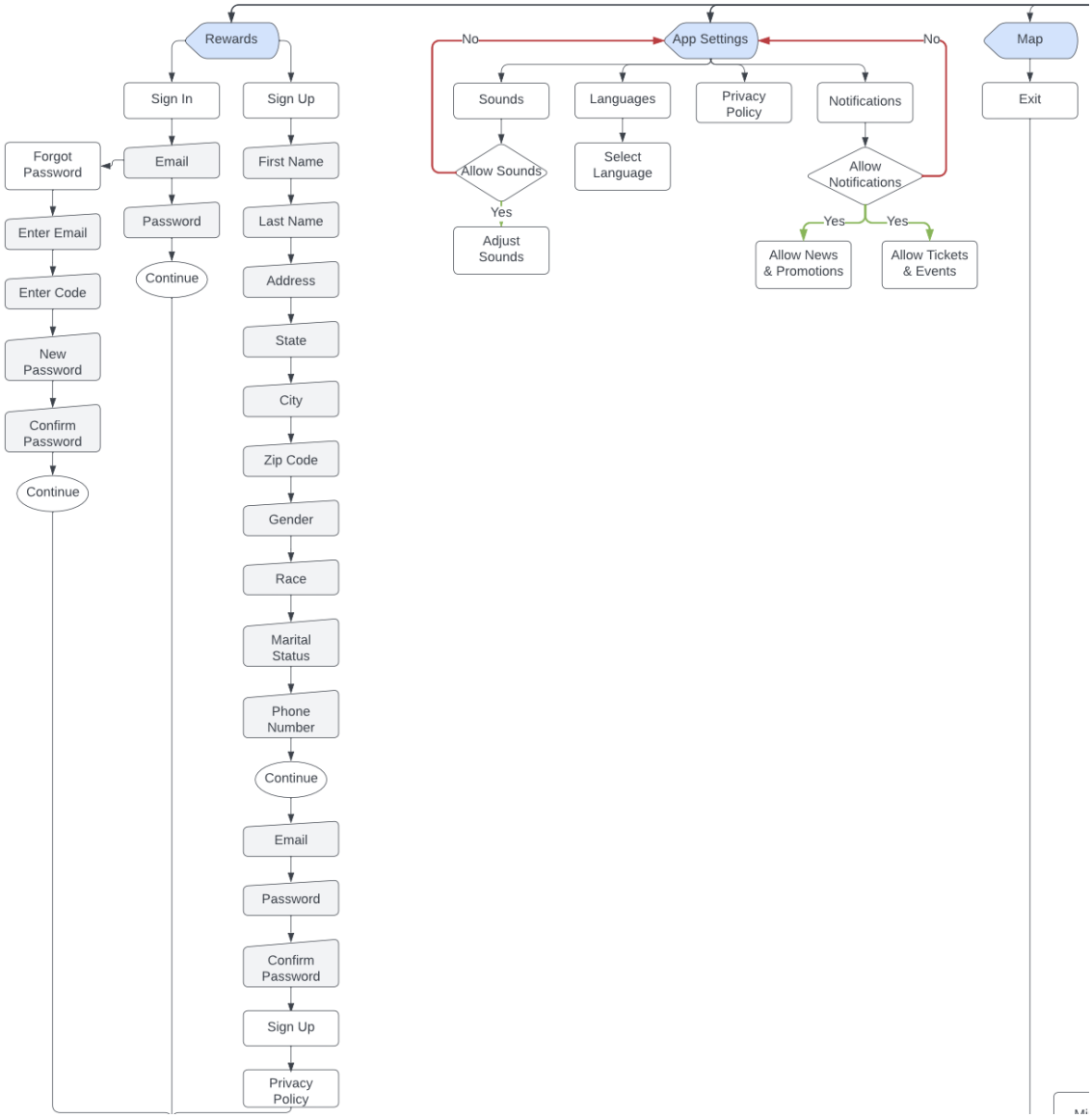



Figure 3 Process Flow Chart - Current Features

Taking cues from last year's MQP Group, the team refined the flowchart to include various updates. Particularly, the Rewards program received a thorough refinement to construct a robust framework. Every process of the program is depicted in Figure 4, where it's notable that the section has a proper flow from start to end. Additionally, the team included the 50/50 raffle and crowd cameos for future implementations that the organization can explore.

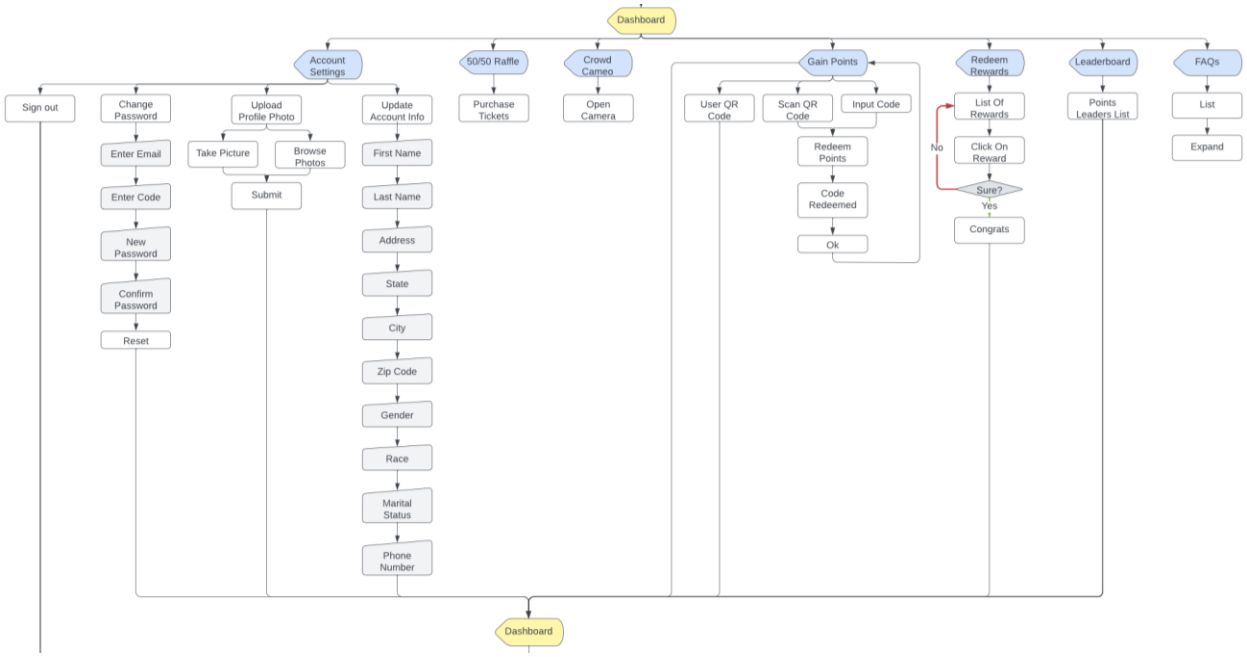



Figure 4 Process Flow Chart - Rewards Program

The final segment of the process flow chart is depicted in Figure 5, showcasing a preliminary version of what the ticket section will resemble once it is seamlessly integrated into the app. The team collaborated closely with the WooSox organization to meticulously establish each feature for the tickets section. For example, this collaboration facilitated the implementation of three options for purchasing game tickets. While the account settings underwent thorough scrutiny, the team envisions the possibility of linking the account with the rewards program account. If feasible, this would afford the organization the opportunity to consolidate the account settings as another primary section within the app, rather than dispersing them into subsections within the rewards program and Tickets.com.

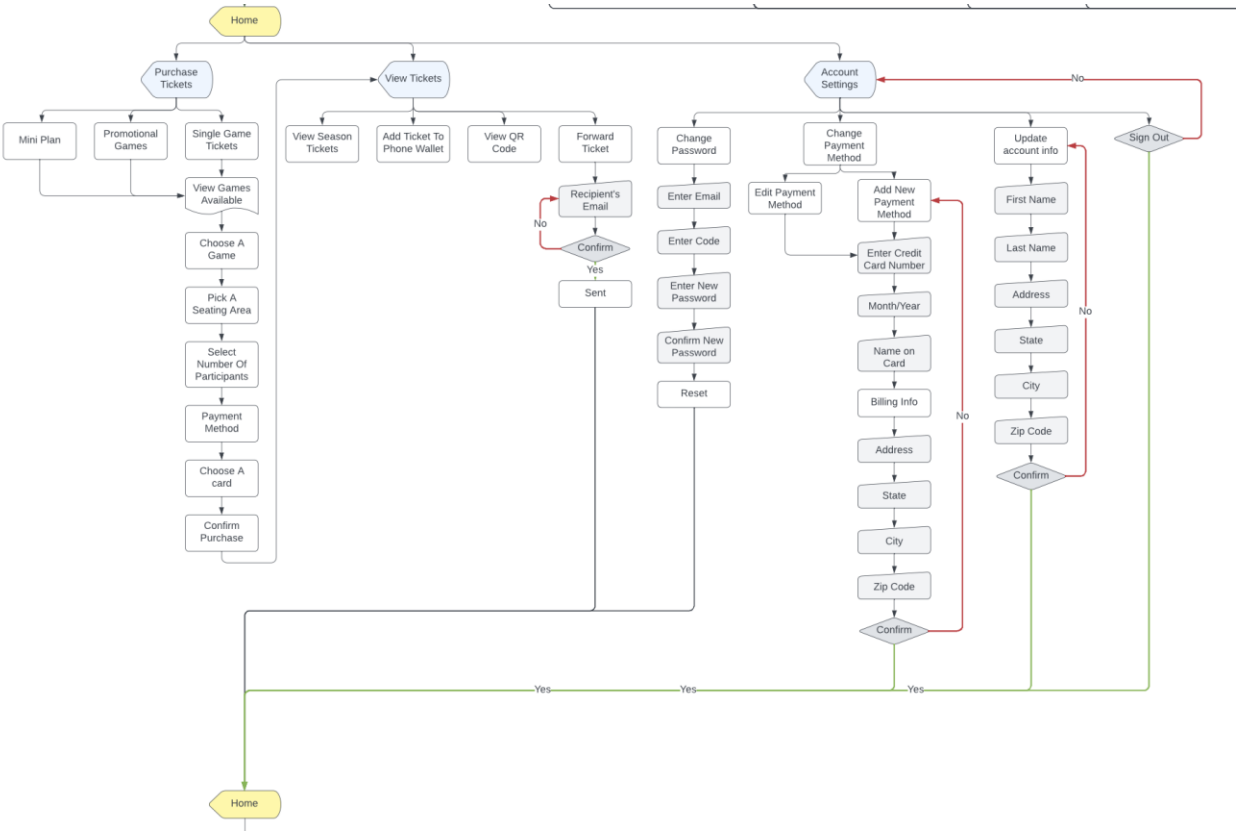



Figure 5 Process Flow Chart - Future Tickets Additions

# 7.0 Design

Much of the design used for the app was taken from the previous team's work. Since the previous team had a lot of time to focus on UI design, their work was well thought through and was crucial for building a framework for the functionality to be built upon. The current team tweaked these designs and added new ones to create the app.

## 7.1 App Layout

The main goal for deciding the layout of the app was for the app to be easily navigable. To accomplish this, the team decided to choose design elements from existing apps to implement into the Polar Park App, so that users familiar with smartphone usage would have an easy time handling the app.

The main element of the app Layout that was used was the bottom navigation bar. A bar on the bottom of the screen like this is commonly seen in hundreds of apps and is shown to be easily navigable. This was designed and initially implemented by the previous team, and much of the functionality comes from them. However, this initial design needed some changes to reflect different needs from the WooSox from last year to this year. Firstly, last year the WooSox Market was a newer aspect to the ballpark, and last year's team felt that it was important to highlight this as one of the sections of the app. However, the WooSox seemed to want to shift focus to promoting the WooSox Rewards program, so the WooSox Market aspect simply became part of the concessions page, and the page that had the WooSox Market was replaced with the WooSox Rewards Section seen in the app now.

The other main part of the layout that was used is the header bar. This bar was initially added by the previous team; however, it was also changed in some ways. First, its color scheme was changed from white to the WooSox Dark Blue to reflect the new designs for most of the pages. Next, the icon used for going back from subpages was tweaked so that it is easier to click. And finally, it was removed from select pages for a custom header. The functionality of this header and the navigation bar is further explained in section 9.1.

## 7.2 Color Choices

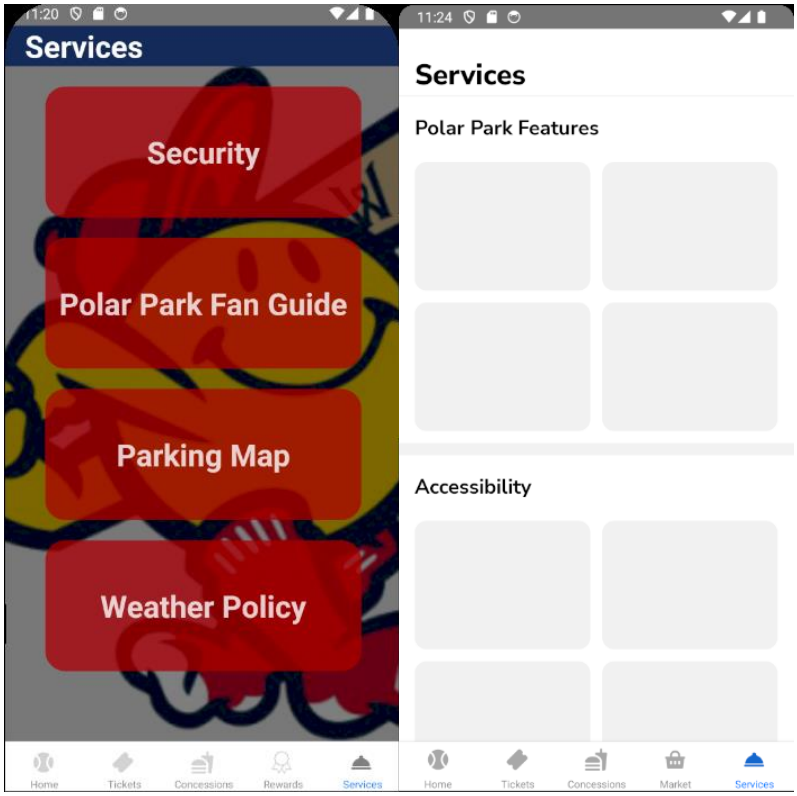



Figure 6 (left) The new design of the services page.

Figure 7 (right) The original design of the services page.

When designing anything artistic, a color palette is often used. It is a simple set of colors to stick to so that the design stays consistent throughout the whole piece. This concept is also applicable to UI design. A simple design should only use a couple of colors to make it easy to read and understand ("Building Your Color Palette"). This was understood by last year's group, and they built a color palette to be used throughout the app. However, these colors were sparingly used, completely unused in the Ordering (now Concessions) page, Services Page, and for most of the homepage. This led to a large amount of whitespace within the app and the app having a somewhat empty look to it. To fix this, many design elements that used a white background and colored text on it were replaced with a colored or picture background with high contrast text. The simplest example of this is seen on the Services Page. In the original design, the services page featured a white background with squares that could be filled in with images for the user to click on (Figure 7). However, this was seen as empty even as these squares were populated in development. So, a new design for the Services page was made to take advantage of the color and make the app pop out more (Figure 6). The MQP team wanted to stick with the color palette, as it was well-selected by last year's team and reflects the WooSox perfectly; however, this year's group chose to embrace those colors to a much greater degree than last year's group.

# 8.0 Software Development

## 8.1 Sprint Development

These are the 8 sprints that took place between the start of the term in October 2023 and the end of the term in December 2023. They include a list of the different user stories that were completed and some overall thoughts about the sprint as well as the retrospective that was held after it.

### 8.1.1 Sprint 0

In our inaugural sprint, the transition to a larger workload transformed our team dynamics. Commencing our sprints early proved pivotal, ensuring successful task completion for the week. Adequate communication throughout the sprint kept us informed and aligned on our progress. Balancing individual and group work was beneficial, fostering focused efforts.

However, during our retrospective we found opportunities for improvement. We recognized the need for more collaborative time to navigate challenges efficiently, supplementing our individual efforts. Enhancing communication with our sponsor, Jiwon Choi, a data analyst for the Worcester Red Sox, became apparent, as clarity on app requirements was lacking. Additionally, we identified a conservative approach to workload, realizing the potential for taking on more tasks without hesitation.

| **Sprint 0: Initial Work** | | |
|---|---|---|
| User Stories | Points | Status |
| As a user, I want a basic parking map to see where parking is available around the park | 3 | Completed |
| As a user, I want a seating map to see where my seats are located in the park | 1 | Completed |
| **Total Points Completed:** | | 4 |

Table 6 Sprint 0 User Stories

### 8.1.2 Sprint 1

In our sprint retrospective, we pinpointed crucial areas for improvement. We recognized the necessity of increasing meeting frequency to bolster team communication and cohesion. Emphasizing the need for clearer task breakdowns, we aimed to enhance workflow efficiency by refining user story comprehension. Keeping our project management tool, JIRA, consistently updated emerged as a shared commitment to accurately track task progress. Moreover, we

pledged to allocate more time to incremental progress on the MQP paper, ensuring its steady advancement. These actionable steps are pivotal for boosting our efficiency and collaboration in future sprints.

This sprint marked the inception of designing the Rewards Program flow, showcased in Appendix C. Drawing inspiration from Dunkin Donuts Rewards, our goal was to model the WooSox rewards section of our app similarly. Additionally, we fully integrated the parking garage map as per Jiwon's request in one of our meetings.

| **Sprint 1: Setting up WooSox Rewards** | | |
|---|---|---|
| User Stories | Points | Status |
| As a user, I want to log into my rewards account, so that I can keep track of my rewards | 5 | Completed |
| As a user, I want to be able to create a rewards account so that I can earn points | 8 | Completed |
| As a user I want to have an easy to use dashboard that displays all of the points I have | 5 | Incomplete |
| **Total Completed:** | | 13 |

Table 7 Sprint 1 User Stories

### 8.1.3 Sprint 2

Until now, our primary blocker was gaining access to the Tickets.com API. Consequently, we pivoted our focus towards enhancing the WooSox rewards functionality and temporarily shifted the ticketing aspect of the application aside, slightly adjusting our project scope. Simultaneously, we made a deliberate decision to amplify our efforts on developing the interactive map, recognizing its high utility for users.

A significant breakthrough came with our acquisition of the FanMaker SDK access. This enabled us to initiate the development of crucial API endpoints essential for the various facets of our app's rewards system.

During the retrospective, we realized that this approach should be utilized more throughout the project. We need to do things that can be done now ASAP because given the short timeline of the project, we do not have the luxury of waiting. In addition, the continued importance of starting work early and bringing up problems as they happen were key points to focus on to improve productivity.

| Sprint 2: Improving Rewards Flow | | |
|---|---|---|
| User Stories | Points | Status |
| As a user I want to see the WooSox Rewards privacy policy within the application | 3 | Complete |
| As a user I want to be able to get redeem a code by typing it in or scanning a receipt. | 5 | Incomplete |
| As a user, I want to have a list of all the foods I can get so that I know what I can buy at the ballpark. | 3 | Complete |
| As a user, I want to see the leaderboard of all the user's, so that I can try to earn the most points | 8 | Complete |
| As a user, I would like to see the FAQ so that I know how the rewards program works. | 3 | Complete |
| As a user, I want to see the list of prizes I can spend my rewards points on. | 8 | Incomplete |
| **Total Points Completed:** | | 17 |

Table 8 Sprint 2 User Stories

### 8.1.4 Sprint 3

During this sprint, David's absence due to illness impacted the overall application progress, affecting his usual workload contribution. On a positive note, Miguel completed his travels for the term, enabling him to channel full attention towards the project without external distractions. Similarly, Mohamed concluded his football commitments, freeing up more time for project dedication, potentially boosting development productivity and accelerating progress.

However, we encountered several blockers this week. Mohamed faced storage issues on his machine when launching xCode, while David encountered difficulties with his Android Simulator restricting certain API requests inexplicably. Additionally, we encountered challenges in locating the API calls for redeeming points and posting information to the server. Fortunately, we discovered an alternative set of API calls that facilitated these functionalities.

It was hard to do the retrospective for this sprint. Many of the issues that came from it felt out of our control, like illness, issues that had zero documentation, and computer storage problems. To improve we needed to look at improving our response to these problems so that we can work around them. This is done somewhat on a case by case basis, but for David’s illness, having work-together sessions on Zoom instead of not having them at all is one possible work around.

| Sprint 3: Enriching Application Home Page | | |
|---|---|---|
| User Stories | Points | Status |
| As a user, I want to see the list of prizes I can spend my rewards points on. | 1 | Complete |
| As a user I want to have an easy to use dashboard that displays all of the points I have | 3 | Complete |
| As a user I want to be able to get redeem a code by typing it in or scanning a receipt. | 3 | Complete |
| As a user I want to be able to access the interactive map from the home page | 3 | Complete |
| As a user I want to be able to see all of the upcoming events from the home page | 5 | Complete |
| **Total Points Completed:** | | 15 |

Table 9 Sprint 3 User Stories

### 8.1.5 Sprint 4

Previously, accessing the Tickets.com API posed a significant obstacle, crucial for integrating a ticketing system into our primary application. Fortunately, through the collaborative efforts of our professors, we secured access to this API. This breakthrough allowed us to revisit and strategize the integration of ticketing, aiming for optimal functionality and aesthetics within the main application. Leveraging UI components from last year's team streamlined our focus solely on backend development for this section.

Furthermore, the rewards functionality within the application has been successfully finalized. Presently, we have two separate apps: the rewards app and the Polar Park application. With the completion of the rewards app, our current focus revolves around seamlessly integrating it into the main Polar Park app.

The retrospective for this sprint was to reinforce a lot of the Agile Scrum concepts that have been forgotten about as the project has gone along. For example, Sprint was not happening daily like it is supposed to, our JIRA was not being updated as tasks were being completed, and the Agile roles weren’t being followed as closely as they should have. The advisors told the group that they needed to improve upon these things to ensure the success of this project for the rest of the term, so the team talked about improving these things.

| Sprint 4: Finalizing Rewards and Introducing Ticketing | | |
|---|---|---|
| User Stories | Points | Status |
| As a user I want to be able to easily view my points on the dashboard | 1 | Complete |
| As a user I want to be able to see the featured offers and rewards on the dashboard | 3 | Complete |
| As a user I want to be able to access the navigation bar from the dashboard | 3 | Complete |
| As a user I want to have my phone's location services pinpoint where I am in the park | 13 | Incomplete |
| As a user I want to be able to see seating and concession information on the app | 5 | Complete |
| As a user I want to be able to see all of the concession stands available to me in the park | 5 | Complete |
| As a user I want to be able to search for specific food items, and filter the concession stands by food item. | 13 | Incomplete |
| **Total Points Completed:** | | 17 |

Table 10 Sprint 4 User Stories

### 8.1.6 Sprint 5

During this sprint, we initiated UI enhancements for the application, notably introducing a dedicated ticketing section on the homepage instead of integrating ticket functions directly. Although the rewards UI is now integrated into the main Polar Park app, backend API connections for the rewards section remain incomplete. Additionally, we identified minor gaps in the rewards flow, such as the absence of user account settings within the application.

Improvements were made to the forgot password flow, ensuring users can sign in even if they forget their password. Furthermore, Jiwon provided Miguel with last-minute ticket sales data, a crucial dataset for our planned linear regression model. This model aims to delineate how various factors influence the park's profitability.

This week's retrospective had a major emphasis on working closer with Jiwon and the advisors. Jiwon's input has been a major push forward in aligning what the WooSox wanted the app to look like with the team's vision. While these were close, there were some key details that needed to be fixed so that the interests were aligned. Our advisors were also crucial in helping the team realize some of the problems with the current app, particularly with UI. These ideas needed to be incorporated into the app so that the best product was made.

| **Sprint 5: Fully Integrating Ticketing and Rewards** | | |
|---|---|---|
| User Stories | Points | Status |
| As a developer I want to complete all rewards pages and rewards flow. | 8 | Complete |
| As a user I want to be able to buy tickets in the app | 5 | Complete |
| As a user I want to be able to view the tickets I have purchased in the app | 3 | Complete |
| As a user I want to be able to see my seating information on each ticket in the app | 3 | |
| As a user I want to be able to search for specific food items, and filter the concession stands by food item. | 13 | Complete |
| **Total Points Completed:** | | 32 |

Table 11 Sprint 5 User Stories

### 8.1.7 Sprint 6

This sprint centered on finalizing the UI structure and ensuring its consistency, aiming for a cohesive and user-friendly application. While most components are refined, the ticketing and interactive map functionalities remain works in progress.

Regarding ticketing, our attempts to retrieve specific user information from the TDC via the tickets.com API were unsuccessful. As a workaround, we're considering loading the tickets.com webpage directly into the application for users to access their ticketing accounts, although this may lead to future issues requiring dual logins. To visualize the application's flow comprehensively, we developed a detailed process flow chart encompassing all aspects.

Upon discussions with Jiwon, we revamped the services section to offer a broader range of services to users. Expanding beyond parking map services, this page now includes additional features such as security services, the Polar Park fan guide, and the inclement weather policy.

This week's retrospective was about figuring out what we needed to focus on to make sure everything was done for next week. One of the first items of importance was working together. We all agreed that we were our most productive working together, so since we need to crunch to get the final things done, maximizing this time working together would be advantageous. Next was trying to fit in an extra meeting with Jiwon to make sure that every detail of the app was to his preference. Lastly, it was making sure that we were all fit to finish everything that was needed and finishing strong.

| **Sprint 6: Finalizing UI and Backend Structure** | | |
|---|---|---|
| User Stories | Points | Status |
| As a user I want to be able to tap on parts of the map, and have information displayed about the section I click on | 8 | Complete |
| As a user, I want to be able to scan all my QR code at different kiosk, and have my points update depending on what I buy in the park. | 5 | Complete |
| As a user I want to be able to edit my account information | 5 | Complete |
| As a user I want to be able to find what to do in cases of inclement weather within the application | 1 | Complete |
| As a user I want to find the Polar Park Fan Guide in the application | 1 | Complete |
| As a user, I want to be able to call park services at any time | 3 | Complete |
| As a user I want to use the rewards program within the | 13 | Complete |

| | | |
|---|---|---|
| main polar park application | | |
| **Total Points Completed:** | | 36 |

Table 12 Sprint 6 User Stories

### 8.1.8 Sprint 7

In our final sprint, our focus was on refining our app to achieve a fully functional minimum viable product with top-notch usability. We addressed every detail, from minor UI adjustments to revamping entire sections like the concessions page.

Simultaneously, we prioritized crafting a comprehensive, well-designed presentation that effectively showcased our app's development journey and meticulous work.

Realizing our heavy emphasis on app development had halted progress on our accompanying paper, we devised a structured plan. This plan ensures the paper's completion in time for the eCDR review, allowing us to balance both aspects of our project effectively.

Since this was the last sprint, our retrospective was kept brief, but we really shouldn’t have saved this much of the paper for after the end of the project.

| Sprint 7: Finalizing MQP | | |
|---|---|---|
| User Stories | Points | Status |
| As a user I want an overall smooth experience in the Polar Park App | 8 | Complete |
| **Total Points Completed:** | | 8 |

Table 13 Sprint 7 User Stories

## 8.2 Product Burndown Chart

Throughout the development timeline, our team set out to tackle 159 user story points across 7 sprints, aiming for a steady completion rate of 27 points per sprint. This target, however, proved ambitious as unforeseen challenges emerged, disrupting our workflow. Despite these setbacks, the product burndown chart demonstrates a consistent descent, signifying relentless progress. Ultimately, the team adeptly navigated through the obstacles, completing a commendable total of 120 user story points by flexibly adjusting project scopes to meet the real-time needs of the development.

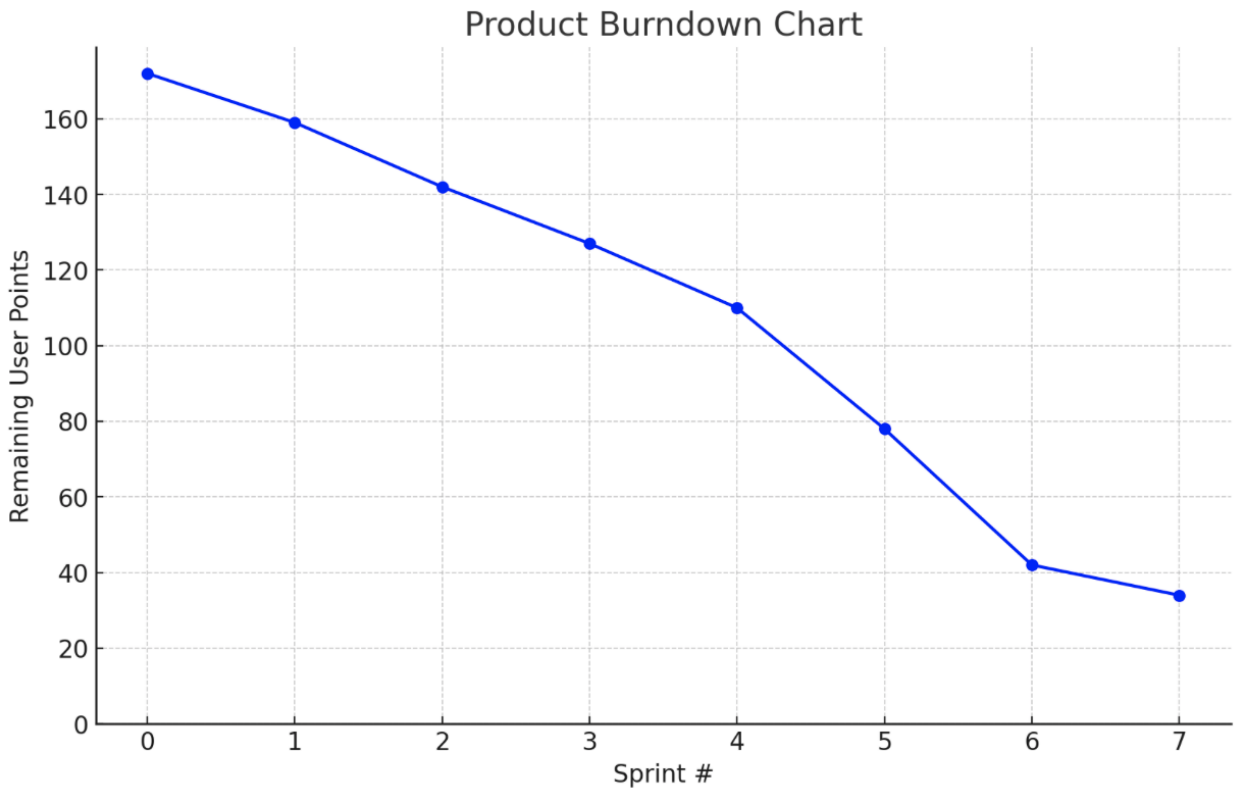



Figure 8 Product Burndown Chart

# 9.0 Implementation

With everything designed, the team had to go through implementing everything into the app. Each page was designed using different features that catered to its needs, using different libraries to help it succeed.

## 9.1 App Pages

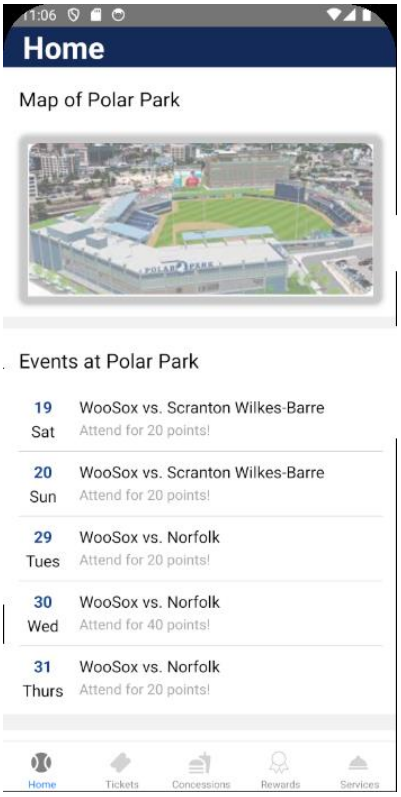



Figure 9 The home page of the Polar Park App, simulated on 8/19/2023.

When the user opens the app for the first time, they will see the homepage of the app. This home page contains two sections, a large button that will bring the user to the ballpark map, and a list of upcoming events at polar park. This list of upcoming events is taken from Fanmaker, and shows the name of the event, the date of the event, and how many points one can earn by buying tickets and attending the event.

Most main pages share two aspects to them: a header and the bottom navigation bar. The header uses the dark blue color from the WooSox's logo and is simply there to tell the user what page they are on and handle some minor issues with the top area of some phones (like the "island" or the "notch" seen on many phones). On subpages, the header has a back button that will bring the user back to the main page that they are on. The WooSox Rewards section of the app uses its own custom header, which makes navigating those pages easier. The bottom navbar is shown on

all the main pages, and allows the user to navigate between the different main pages of the app. This navbar is not shown on most subpages.

### 9.1.1 Rewards Pages

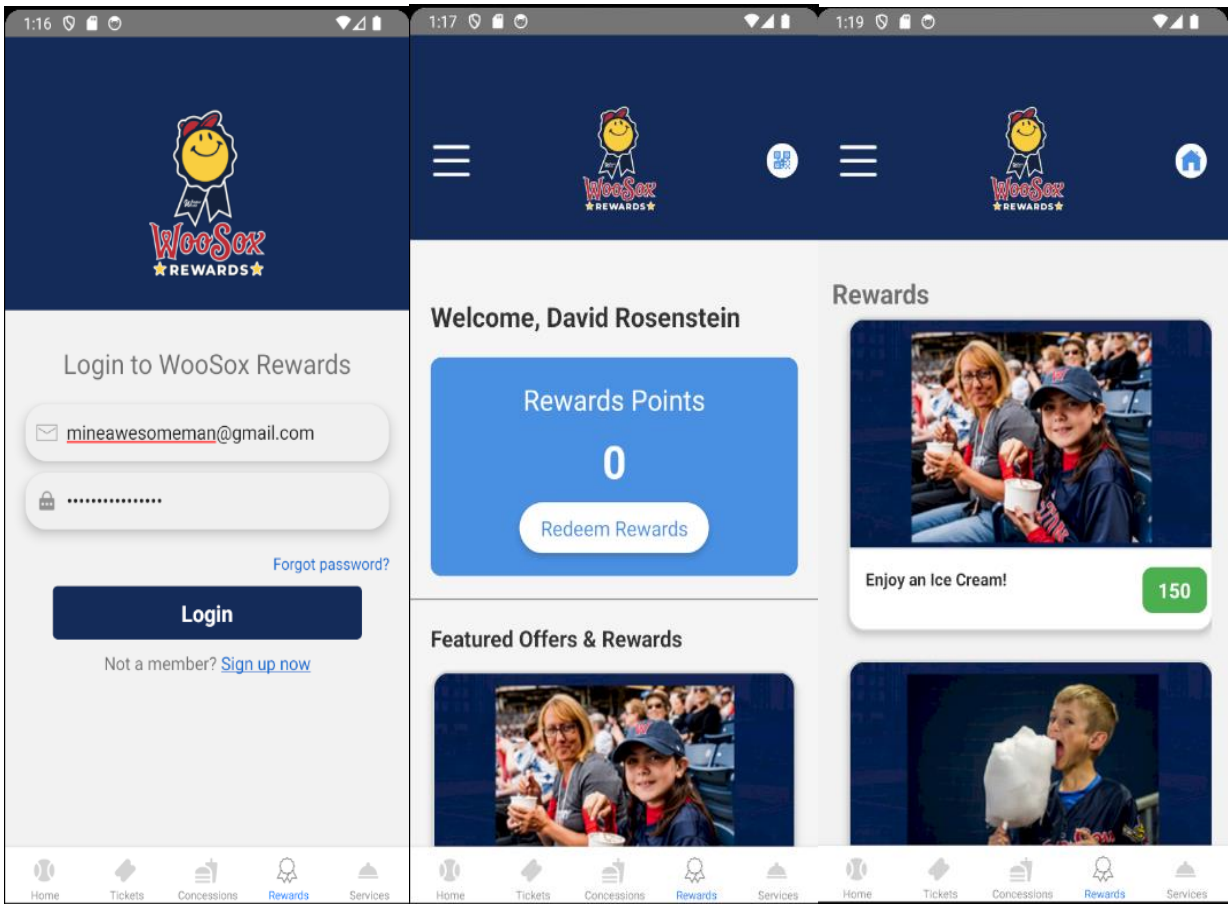



Figure 10 (left) The login page of the rewards section.

Figure 11 (middle) The home page of the rewards section.

Figure 12 (right) The redeem rewards page of the rewards section.

One of the two main sections of the app outlined earlier was the connection to WooSox Rewards. The Rewards Pages are the most intensively designed pages of the app, as the complete API was able to be used in a way to make these pages in the best way for users. This will allow users to complete almost any action related to their WooSox Rewards account, including earning points, redeeming prizes, viewing their account information and more.

On the first time they open the app, users will see a login page prompting them to enter their email and password (Figure 10). This is the email and password connected to their existing WooSox Rewards account. If they do not already have an account, or they forgot their password, they can click on the links provided (more details below). Upon entering a valid email and password, they will be logged in and sent to the dashboard page (Figure 11). Once a user is logged in, the app will remember this, and in the future when users click the rewards tab, they

will be sent directly to the dashboard page instead of to the login page. If the user enters an invalid email and password combination, they will be alerted that their email + password combination is not valid and will not proceed to the dashboard, or be logged in.

Once the user is logged in, they will see their dashboard page (Figure 11). This dashboard is the center of all the activities related to the WooSox Rewards. First is the header, which is shown on all pages. If the user clicks the three horizontal lines, a menu will appear on the side of the screen that they could use to navigate between different pages (Figure 13, more details below). Also on the header is a white button with a tiny QR code on it. Upon clicking this, a popup will appear that shows a QR code. This QR code is associated with their WooSox rewards account - each account will generate a different QR code. This feature was asked to be added to the app by the WooSox team so that when users purchase something at merchandise or concessions stands at the ballpark, they can show their QR code to be scanned by the WooSox employee to make earning points easier. The QR code represents their WooSox Reward ID, which is replicated in the standalone WooSox Rewards App and is a unique identifier for each WooSox Rewards account. After the header is the main content of the page. The main object is a display showing the user's number of points. This was made front and center to the page so that the user can see their points go up over time as they save up for different prizes. This section took inspiration from other popular apps with rewards systems (see Appendix B for more information). This area also has a button that says Redeem Rewards, which sends the user to a page that lists all the rewards options (Figure 12). In addition, a small selection of 5 rewards is shown on the home page. This selection should contain some rewards that the user can currently purchase and some rewards that they are close to but need some more points to purchase. On either page, if the use selects a reward, a dialog box will appear confirming that they want to redeem the prize selected, if they confirm, another dialog box will appear, either saying “Prize Redemption Successful” with some information on what the user should do to get the prize or saying “Prize Redemption Failed” with some information saying why the redemption failed.

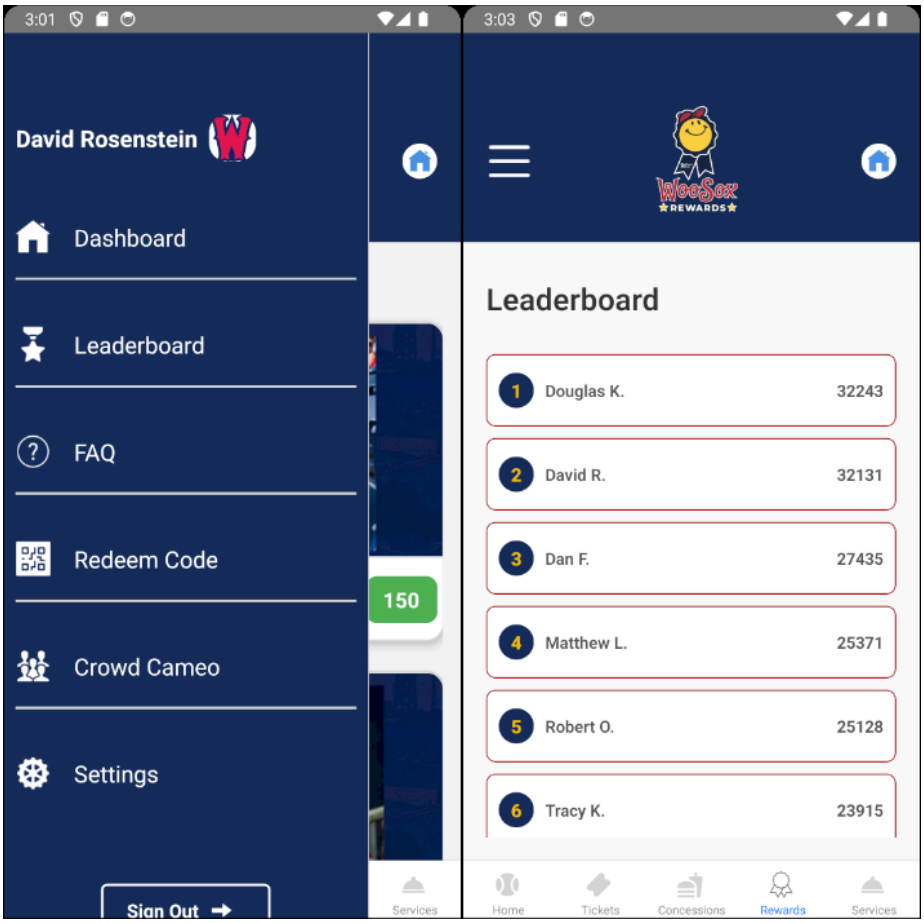



Figure 13 (left) The 'Hamburger Menu' with all the options of the pages to navigate to.

Figure 14 (right) The leaderboard page, showing the users with the highest scores.

On all pages after login, the header will contain a symbol with three horizontal lines. When clicked on, it will open a menu appearing from the left side. This menu is known as a hamburger menu (Figure 13). It contains options to go to the Dashboard, Leaderboard, FAQ, Redeem Code, Crowd Cameo, and Settings pages. It also shows the user's name and profile picture at the top, to make it clear who they are signed in as. Lastly, there is an option to sign out from their account on the bottom. When the user signs out, the app will mark them as no longer logged in and send them to the login page. This menu is designed so that no matter which of these pages the user is on, it is easy for them to navigate to another page, ensuring that users will not 'get lost' within the app.

The first option of the hamburger menu is the leaderboard. This is a feature seen on the WooSox Rewards website that was also put within the app. When opened from the hamburger menu, it will show a list of the top 10 users within the WooSox rewards system by total number of points earned (Figure 14). The goal of this page is for it to be another incentive for users to earn as many points as they can.

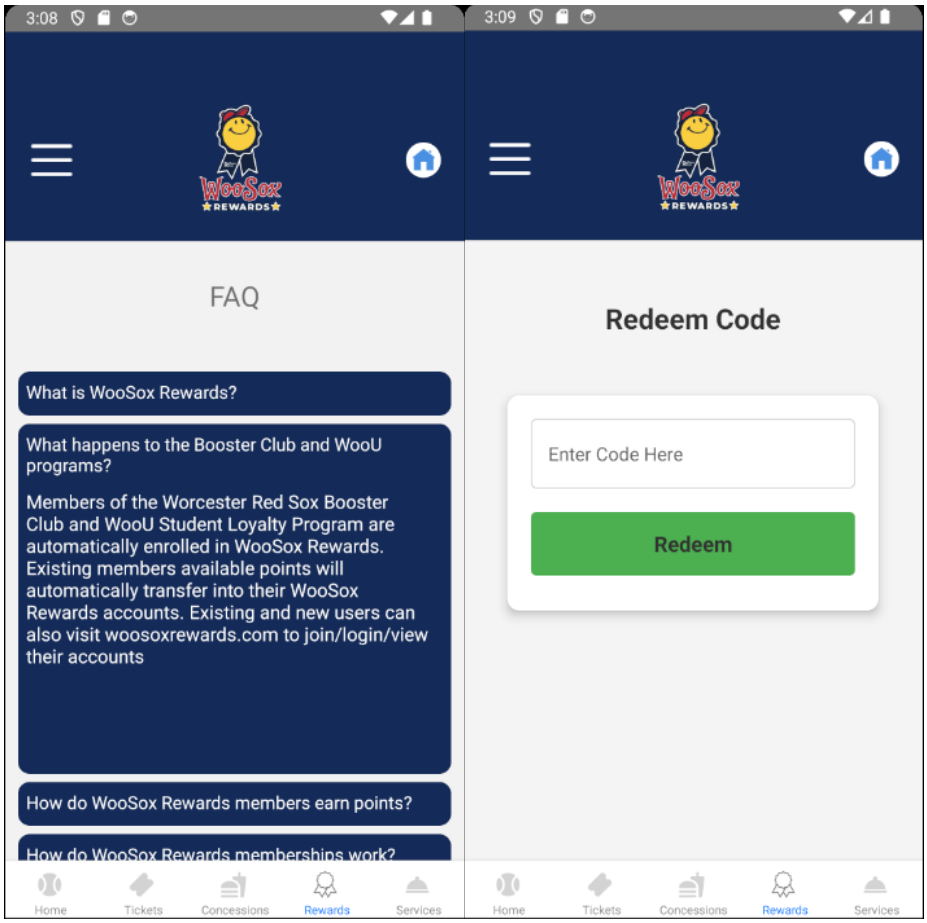



Figure 15 (left) The FAQ page, shown with the second question expanded.

Figure 16 (right) The redeem code page, where users can enter promo codes.

The second option of the hamburger menu is the FAQ. This page answers a lot of commonly asked questions about the WooSox Rewards program, so users can easily figure out what it is and how it works without needing to ask staff or others (Figure 15) The questions and answers are the same as from the WooSox Rewards website; however the format they are shown was changed to make it easier to use on a mobile device. The page is formatted so that when it is first visited, only the questions can be seen. Then upon clicking on a question, it will expand to also show the answer to the question. This format is easy to use and allows for more questions to be added later

The third option of the hamburger menu is the Redeem Code page. This page has an option for the user to type in a code that they received to add points to their account (Figure 16). The use case for this page is that when users purchase things from the WooSox store, one of these codes gets printed on their receipt so that they can get points for it. While this may get phased out for the QR scanning feature, it was put in the app so that there is still a way to earn points through either means. Upon opening the page, the user will be prompted with a text field and a submit button. They can type their code in the text field, then upon clicking submit, that code will be sent. If it is a valid code, a message will appear saying that the code redemption was successful.

If it is an invalid code, a message will appear saying that the code was invalid. This page is a simple yet effective way for users to earn points through existing promotional codes.

The fourth option of the hamburger menu is the Crowd Cameo page. This did not seem to be available within the WooSox rewards API, and therefore it did not make it into the app. The page simply shows “Coming Soon” with the hope that a future team could include this within the app.

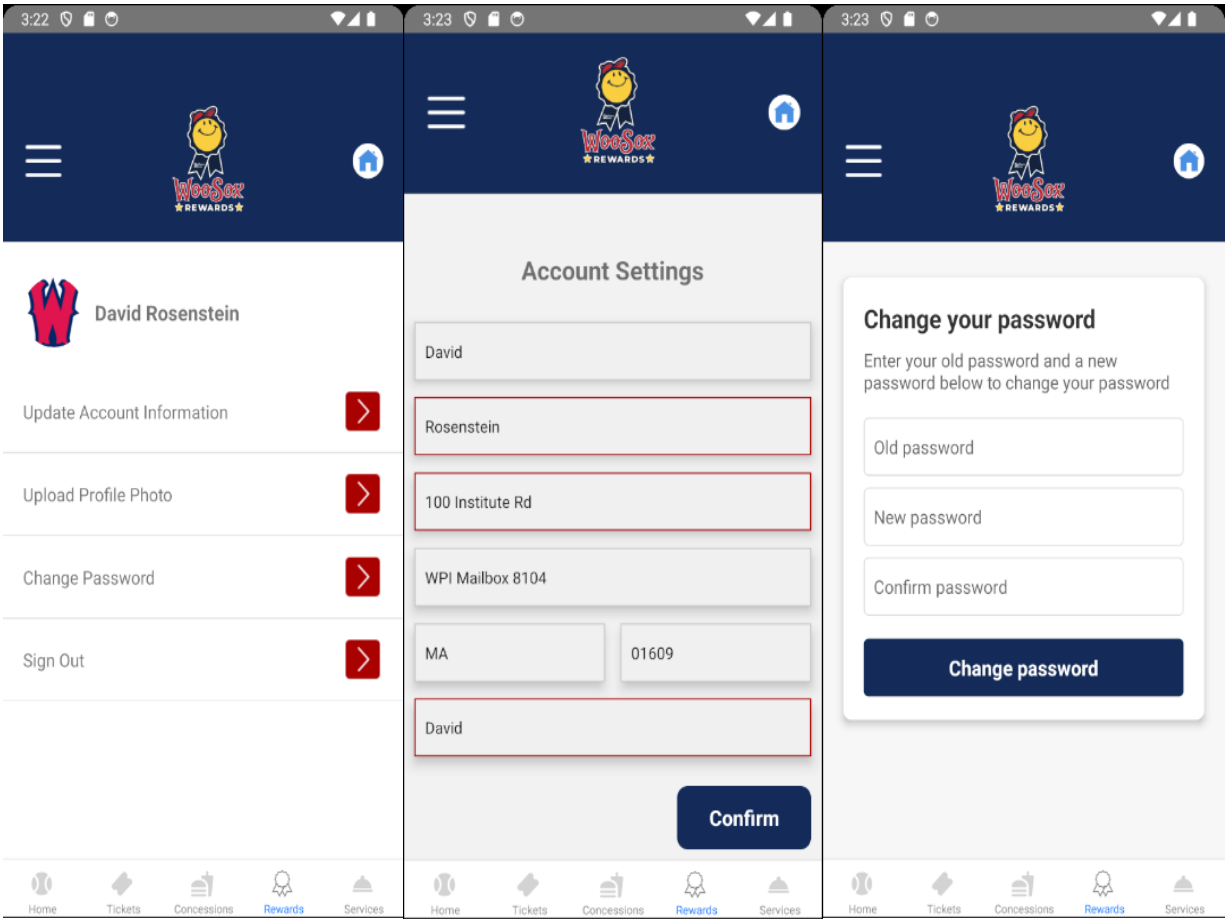



Figure 17 (left) The settings menu.

Figure 18 (center) The update account information page from the settings menu.

Figure 19 (right) The change password page from the settings menu.

The fifth and final option of the hamburger menu is the Settings page. This opens a page with a list of subpages for the user to open (Figure 17). The first subpage is account settings (Figure 18). This allows a user to change the name and address associated with their account, to account for users changing name or moving. The second subpage is to change the profile picture associated with the user’s account, which currently just appears as ‘coming soon’. The third subpage allows the user to change their password (Figure 19). To do this, they have to type their old password, their new password, and their new password again. This ensures that only the authorized user can change their password, and that they are changing it to what they want. The fourth and final subpage is another location for the user to sign out, and it performs the exact

same routine as the signout button on the bottom of the hamburger menu. These options are just for specific circumstances and allow users to have full control of their WooSox Rewards account.

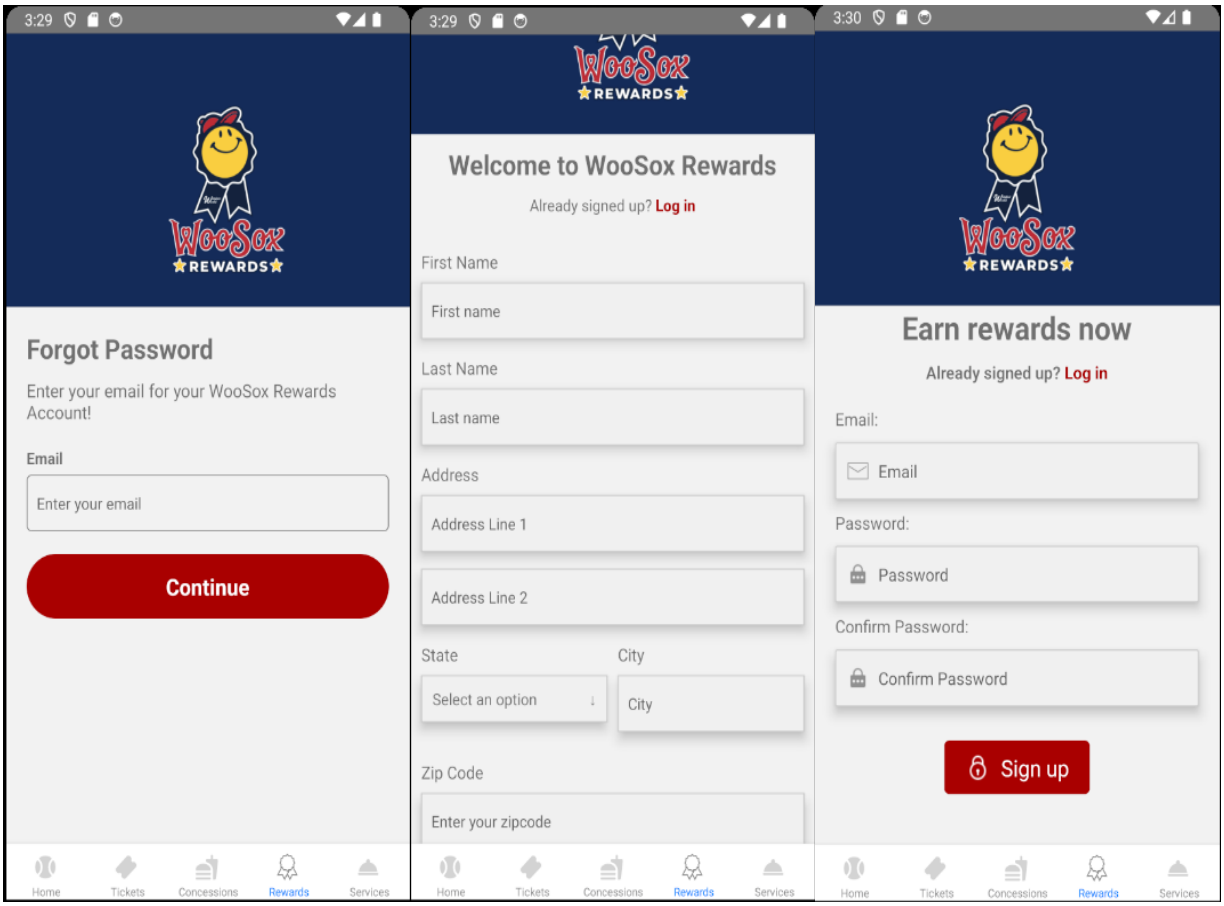



Figure 20 (left) The 'Forgot Password' page where the user can enter their email.

Figure 21 (center) The first sign up page scrolled down, where the user inters their info.

Figure 22 (right) The second sign up page where the user enters their final details.

From the login page, if the user does not know their password, they can select the “Forgot my password” option. This will bring them to a page where they can type in their email (Figure 20). Once their email is submitted, a popup will appear instructing them to check their email for more instructions. This page recreates the forgotten password experience from the WooSox website on mobile. This means that they will receive an email with a link, and upon opening that link they will have the option to change their password to something new. Once they do, they can sign in normally with that new password on the app.

Also from the login page, if the user does not have an account, they can choose to create a new account. This will bring them to a page for them to fill in all their account information, including name, address, phone number, race, gender, and marital status, with some of this information being optional (Figure 21). Once this information is filled out, the user can select continue,

where the user is then prompted to enter their email and the password they want to use for their account (Figure 22). Once that information is entered, the user's account will be created, and they will be congratulated for making an account. From there they will be sent to the dashboard where they will be able to do all account activities.

This complex structure for all the aspects for rewards was made possible by giving the rewards sections its own navigation separate from the rest of the application, this allows a user to do something within the rewards section, then go do something in a different section of the app, then come back to what they were doing before. This also allows the rewards section to be its own robust section so that if things are changed in different parts of the app, the rewards section will not be impacted. This is helpful for app resiliency and maintainability, making problems easier to isolate and fix in the future.

### 9.1.2 Ticketing Page

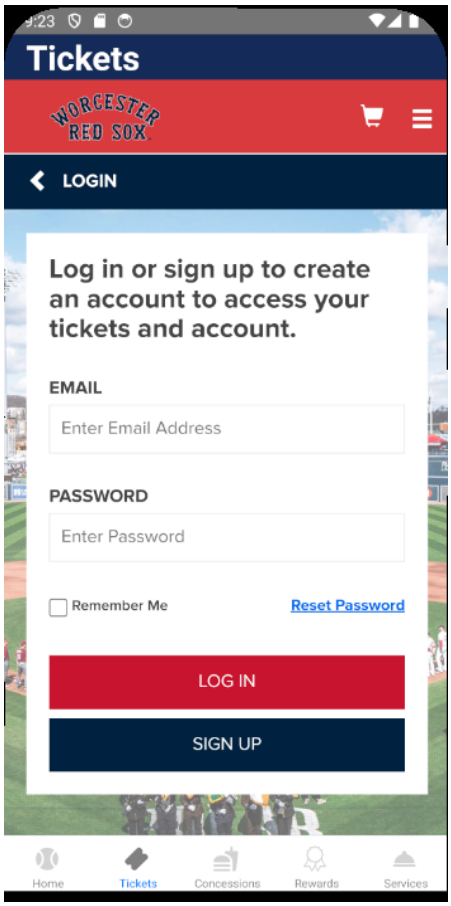



Figure 23 The ticketing page.

The ticketing page was the other key aspect that was deemed important for the app to succeed. However, due to the inability to utilize the tickets.com API in an effective way, a simpler solution was chosen to display this information to users. The WooSox already have their own endpoint on the tickets.com website for selling their tickets, and this website is optimized for

both mobile and desktop applications. This website is publicly available, which means that it can be displayed within any context, including within the Polar Park App. To accomplish this, the library react native webview was used. This library mimics an old inbuilt function that used to exist within React Native that displays web pages using the devices native browser engine - Safari for iOS and Google Chrome for Android (react-native-webview). This means that the tickets.com website can be viewed in its entirety, which is crucial for integration within the app. Every feature that is available for tickets.com is also available to all users in the app, and if the tickets.com website is changed in any way, the app will update with those changes. This ability is great for app maintainability and allows much of the work to be offloaded to the tickets.com team.

However, this solution also has some major downsides. First, ticketing information cannot be shared to other aspects of the app and are confined to only this page in the way that tickets.com wants to display it. It also means that no direct connection between the ticketing and rewards can exist, which makes getting points from going to games more difficult for users. Lastly, it greatly increases load times, which makes the app less usable. Despite all these downsides, this solution was the most feasible for creating an app with all of the features desired, so it was chosen and allows users to perform all their ticketing needs within the Polar Park App.

### 9.1.3 Ballpark Map Page

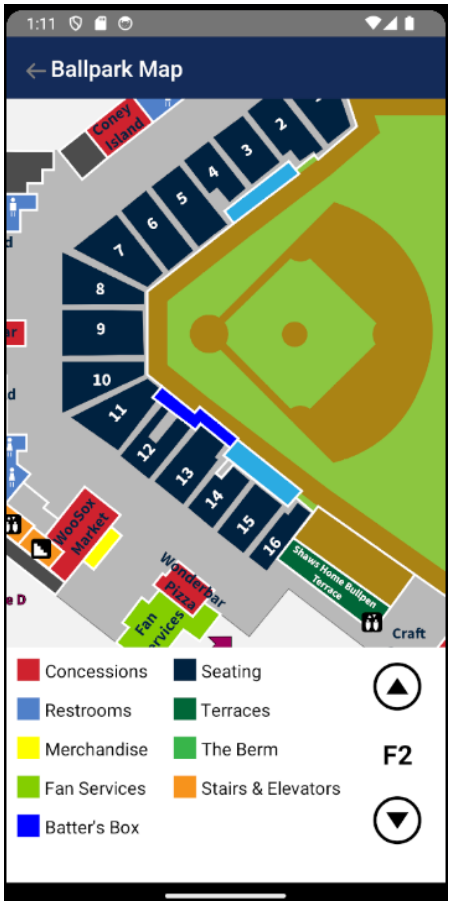



Figure 24 The ballpark map page.

The Ballpark Map page had the most unique design from a coding perspective. The main idea of the page was for it to replicate the user experience of popular mobile mapping apps like Google Maps and Apple Maps. The most important interactions that were considered here were "pinching" to zoom into and out of the map and "panning" to move the map around. Since these are the most common ways of interacting with maps on other applications, we believed that it would allow users to easily interact with the map. The first step in facilitating this was to create a new version of the ballpark map. The existing versions had two main problems: they were in a traditional picture format, meaning when zoomed in they would become blurry, and that they did not include all of the information of the ballpark, particularly with the area around the private suites. The first problem was solved by creating the map in a format called a Scalable Vector Graphic, or SVG. An SVG is vector based, which means that all of the things to be displayed are represented by a large number of lines, curves, and shapes. The device can then use complex math to calculate what each object within the graphic should look like at each location, zoom, and rotation ("PNG vs SVG: What are the differences?"). The map SVGs were created using Adobe Illustrator, which is designed with creating vector graphics in mind. To solve the issue of showing the whole stadium, three different maps of the ballpark were created: one for each floor.

This was the most reasonable way to show everything in the ballpark, as the main concourse between sections 7-12 have important parts of the ballpark both above and below it. (See Appendix A for a comparison between the existing ballpark maps and the ones created for the app.) Once the maps were created, next was the process of integrating them into the application. To perform the actions defined before, a library known as “React Native Svg Pan Zoom” was used. This library advertises “Google Maps-style pinch and drag gestures” which was perfect for what was needed for the page (garblovians). The last aspect of this page was the key of the map, to make it clear what each of the elements of the map represented.

### 9.1.4 Concessions Page

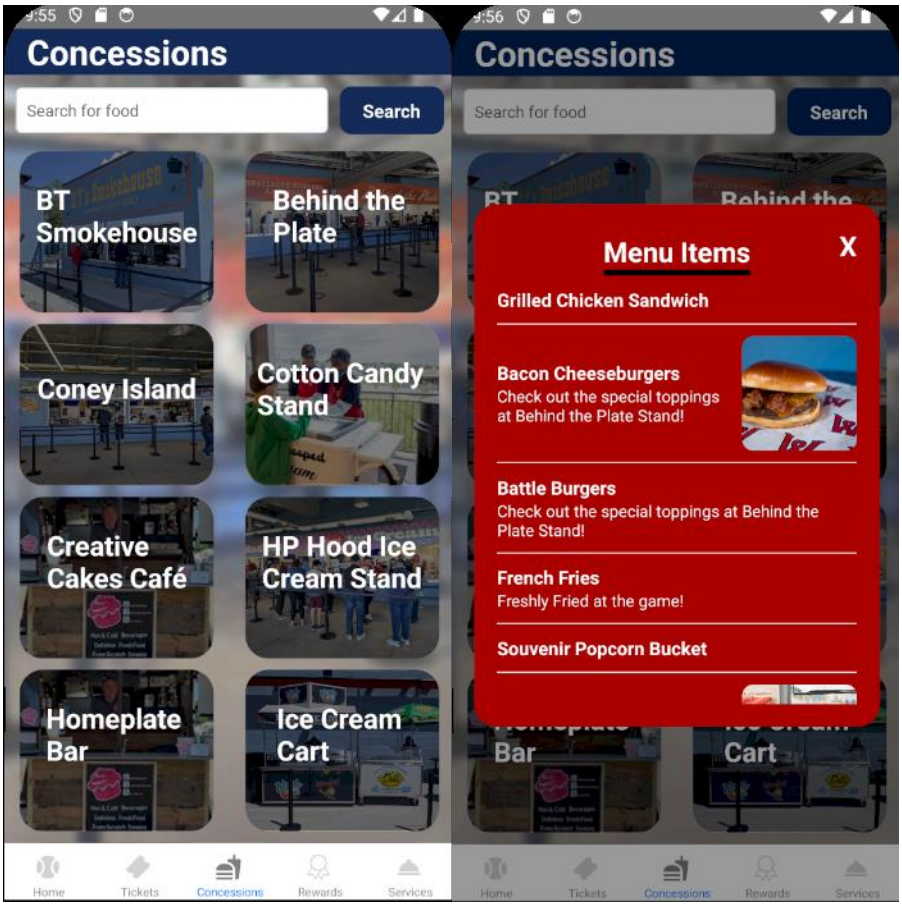



Figure 25 (left) The concessions page.

Figure 26 (right) The popup shown when a concessions location is chosen.

The original app from last year's group contained a page called “Ordering”. The purpose of this page was to connect to Polar Park’s internal vendors so that fans could order food directly to their seats. However, Polar Park is currently in the process of switching which vendor they use, and this would mean that building an ordering system into the app would be impossible. Instead, the page was turned into the concessions page. When the page is first opened up, the user sees a search bar and a set of boxes, with each box representing a different concession stand at Polar Park. If the search option is used, the user can type the name of an item that they are interested in

and click search, and the app will only display the concession stands that serve that item. Upon clicking on one of these boxes, a popup opens with information about all of the unique food options that the concession sells, with the ability to add descriptions and images for each item. All of this information is controlled by a Google Sheets document that is owned by the Polar Park staff. The app requests the information from this Google Sheets document and converts it into the concessions page. This design allows Polar Park staff to edit their menus at any time, and have that menu be updated in the app without needing to update the app itself. This is great for maintainability, especially for the WooSox Market - a concession location that has a shifting menu each week.

### 9.1.5 Services Pages

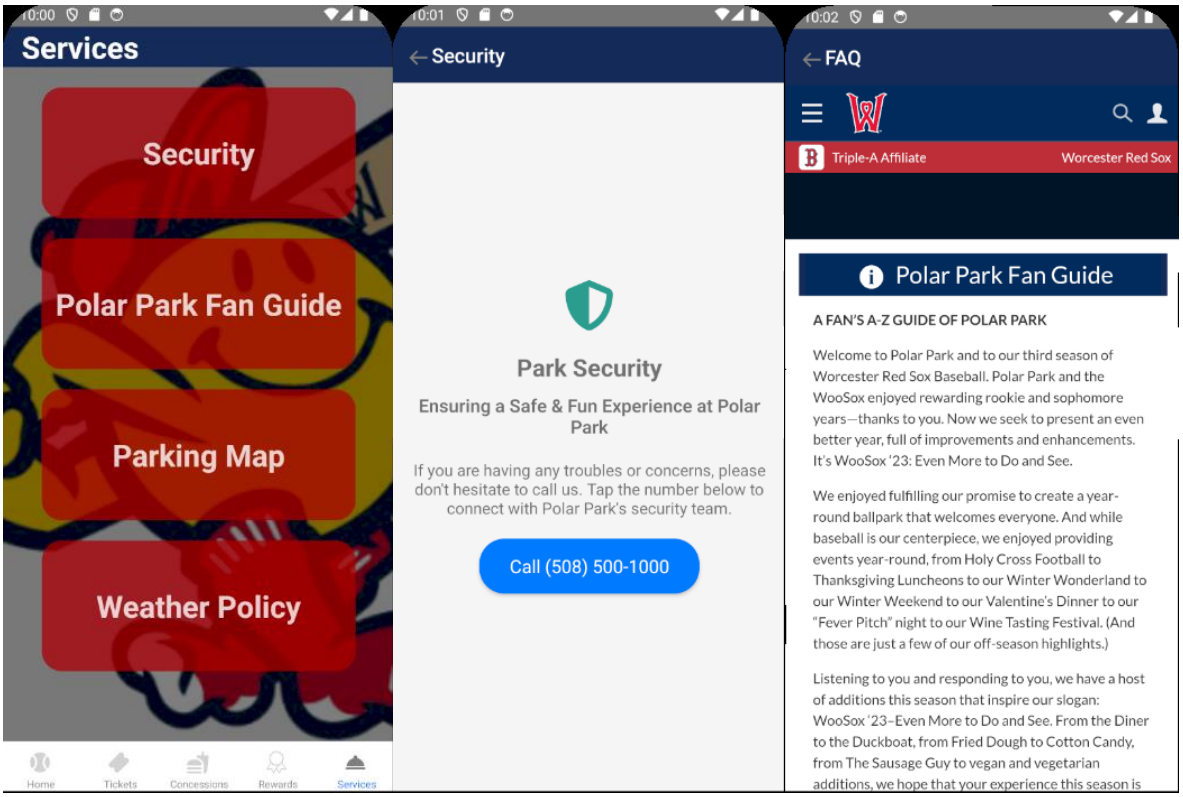



Figure 27 (left) The services page.

Figure 28 (center) The security subpage, featuring the option to call Polar Park Security.

Figure 29 (right) The Polar Park Fan Guide, loaded from the WooSox website.

The final major page of the app is the Services Page. The goal of this section of the app was to provide a page that contains a set of subpages of different things that fans of Polar Park may want. The main page has a simple design: four large red boxes, each of which are clickable so that the user can navigate to each of the subpages (Figure 27). This design was kept simple, as the main services page is just supposed to be a 'hub' of sorts that leads to other pages. The original services page had a lot of whitespaces, as last year's group left quite a few spots open

for services when only a couple were needed, so to fix this the page was changed to have the WooSox's mascot Smiley in the background, with larger boxes representing the subpages.

The first subpage in the services section is park security. This page was left in a simple state intentionally, as if someone truly needed to call security, they should be able to do so quickly (Figure 28). The page contains a short description of park security and why they may need to be called and a large button to call them. When this button is clicked, the user will be redirected to the inbuilt Phone App with the number for Polar Park security already filled in. The user will just need to click the 'call' button on their phone to initiate the call. This process was done using React Native's inbuilt Linking feature, which allows simple linking to external links, emails, phone numbers, and more. This simplicity makes it easier to handle emergencies, making everyone feel safe at the ballpark.

The second subpage is the services section for the Polar Park Fan Guide (Figure 29). This page works similarly to the ticketing page, where it fetches the Fan Guide directly from a website that already exists using the react native webview library. In this case it loads the fan guide from the WooSox Official Website. Because of this, it has a lot of the same benefits and downsides to the ticketing page where it automatically updates when the webpages update, but it is much slower than rendering natively. Since this page does not frequently get updated, it should be considered by future groups to be rendered natively, as it will improve load times and also remove some clutter that exists on the current webpage.

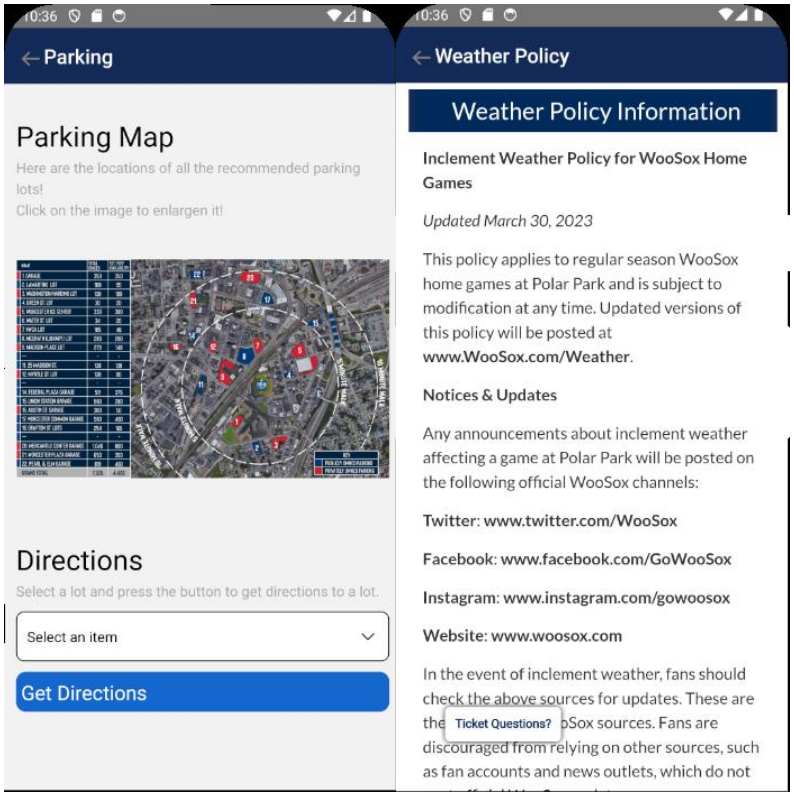



Figure 30 The parking map, with the ability to get directions to certain lots.

Figure 31 (right) The weather policy subpage, loaded from the WooSox Website.

The third subpage for the services section is the parking page. This page has two major elements to it: first is the parking map. This is a map of all the nearby lots that are usually open on game nights, separated between publicly owned and privately owned lots. The user can click on this image to enlarge it, where they can zoom and pan the image around. This works similar to many common social media apps like X (formerly Twitter) and Reddit, so replicating this behavior was seen to be beneficial for users. It uses the React Native Image Viewing library to accomplish this. While that is a library designed for seeing a whole gallery of images, it worked fantastically for individual images as well (jobtoday). The second section of this subpage is for directions. On the parking page of the official WooSox website, there are six parking lots highlighted as recommended. These six lots are also highlighted on the parking page by having this directions option on the bottom. Users can select a lot from the dropdown, and then click the 'Get Directions' Button below it. When this happens a popup will appear, prompting the user with all of the map apps they have installed on their phone. They can select whichever one they prefer, and that app will open automatically with the location already set so they will be able to get directions there. These two features make getting to the ballpark and parking easy for all.

The fourth and final subpage is for the weather policy. This is another page from the WooSox website simply loaded into the app, like the Fan Guide and Ticketing pages. Unlike those pages however, this page should probably stay in this state. This is because the weather policy is something that may need to be updated frequently and flexibility is much more important than the load time issues, which is unlike the Fan Guide Page. However, the WooSox Website team should consider removing some advertisements and other popups to this site, as it worsens the user experience and seems very tacky coming from a sports team that customers are already paying money to go see. The most important part is that the weather policy can still be seen and customers can find out what they need to do if inclement weather changes their plans.

## 9.2 External Connections

For the app to work as intended, it needs to connect to servers hosted by other organizations. There are three main places that the app connects to for this purpose: the Fanmaker API, the tickets.com website, and the official WooSox website.

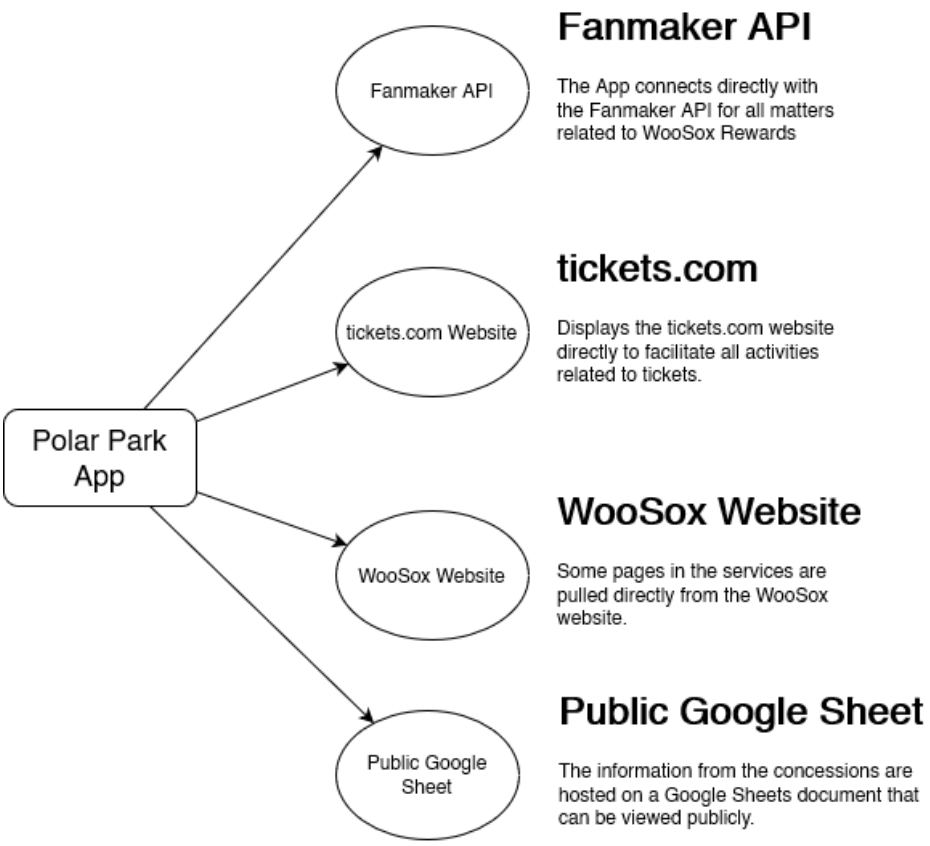



Figure 32 A diagram that shows the connections performed by the Polar Park App.

The first connection is the Fanmaker API. Fanmaker is the third-party provider that runs the WooSox Rewards program. Fanmaker gives all their customers access to an API so that they can develop their own applications that use Fanmaker's services. There are two versions of this API. API 1.0 is designed with administrators and central servers in mind. This allows users to directly add points, custom pages, and more, which should only be accessed by whomever is running the Fanmaker site. This is not applicable to the app. API v2 is designed so that users of the rewards program can connect to it directly, which is perfect for our use case. It is what the WooSox Rewards website uses for all of its connections, which is great for our app to replicate. The app connects to the Fanmaker API to allow users to sign into their rewards account, create a new rewards account, view their point balance, view the different prizes they can spend their points on, view the leaderboard of points havers, redeem more points, and more. To make this connection, the inbuilt function 'fetch' was used. Fetch is capable of making basic HTTP requests, most importantly allowing anything in the header of the HTTP request and allowing all types of HTTP requests. This is the same functionality seen natively within websites that use JavaScript and is capable of doing everything that is needed.

The next connection that the app uses is to tickets.com. Tickets.com is the website the WooSox organization uses to allow fans to pre-purchase tickets online before the game. The original plan for integrating this into the app was to use their API, similar to Fanmaker, so that it can run

natively. This did not work out for three major reasons. First, the team did not get access to the tickets.com API until November 20. 2023, which gave the team 3 working weeks to learn how to use this API and implement it into their system, which is a tight timeline for a new system. Second, their API is set up in a similar way as the Fanmaker 1.0 API, meaning that it would be completely unsafe to use directly in the app. To correctly utilize this API, a server for the app to connect to would be needed (see section 5.2.1 for more details). Lastly, the API has no way of directly interacting with its Account system. The team was unable to find an endpoint within the API to authenticate accounts, making confirming users impossible. All of these problems led to the decision for the team to forgo using the API and instead to use the webview described in section 9.1.2.

The third connection used was directly with the WooSox website. This is used to get the information for the Fan Guide and the Weather Policy subpages of the services section. This website is hosted by MiLB and can be updated by the team directly. Like with the tickets.com connection, this is done using the webview as described in sections 9.1.5.

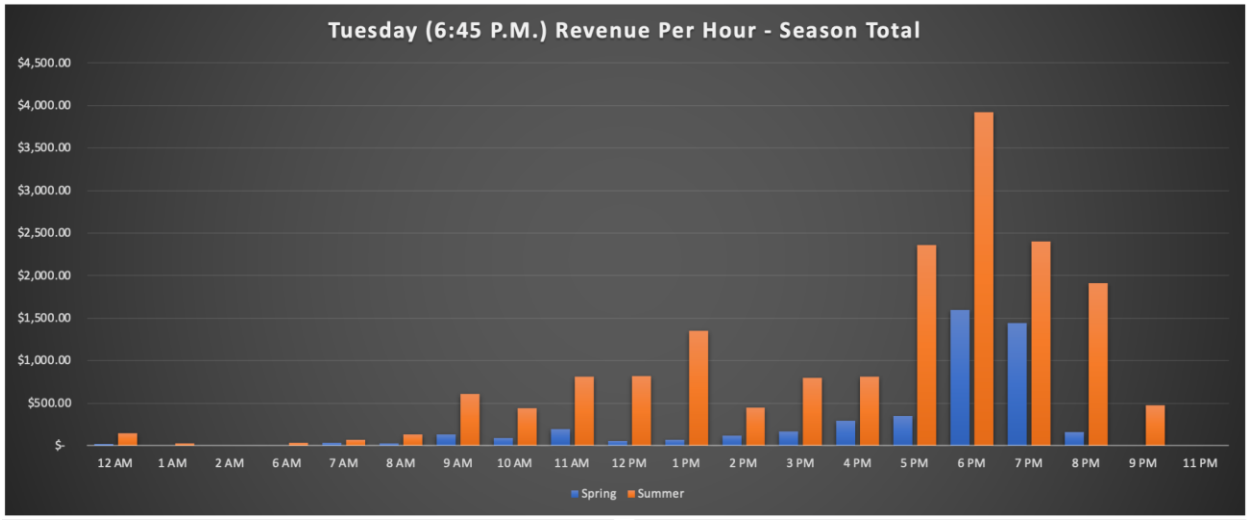
The final connection is with a public Google Sheets document. A Google Sheets document is a spreadsheet, similar to those created in Microsoft Excel. The reason it is hosted on Google Sheets is that with the right link, the app can request a CSV version of this spreadsheet at any time. This CSV version can be easily read and parsed by the app, to turn it into something easily displayable. To get the data, the react native csv library was used. This library is able to pull the data from a link and parse it into the Object format that is standard of TypeScript. From there it is easy to use this data to display the concessions page. This solution was the simplest solution to have an updating menu without needing a dedicated server.

# 10.0 Dynamic Pricing & Income Variability in Last-Minute Ticket Sales

Introducing dynamic pricing & income variability offers a detailed analysis of last-minute ticket sales trends for the Standing Room Only (SRO-Loop) section, shedding light on day-of-game and season-long patterns across various game offerings. Figure 33 through Figure 39 present a comprehensive overview of these sales dynamics, accompanied by detailed charts and tables outlining sales data for both the 2023 season and estimated projections for 2024. These projections incorporate a $1.00 increase in ticket prices and anticipate a hypothetical 5% decrease in sales volume, aligning with discussions and assumptions made in collaboration with the WooSox organization. Through consistent methodologies applied across all figures, this section aims to provide valuable insights into the potential impact of dynamic pricing strategies on revenue generation, offering a glimpse into the financial implications of price adjustments and their influence on overall profit margins.

The recommended price adjustment, factoring in the 5% sales decrease hypothesis, anticipates an estimated profit increase of $6,500.00 (total profit of Figure 33 through Figure 39) in last-minute ticket sales for the entire season in 2024. Post the 2024 season, the WooSox organization will evaluate the accuracy of this hypothesis. This evaluation will guide them in formulating a precise calculation for adjusting pricing on these sales for the 2025 season.

Figure 33 through Figure 39 utilize consistent methodologies, differing only in their income projections derived exclusively from total season last-minute ticket sales, categorized by the day of the week. Within each figure, the table labeled "2023 Total Last Minute Ticket Sales ($9.00)" presents a comprehensive overview of the total sales volume for the entire season at a fixed price of $9.00 per ticket on the top selling hours. Conversely, the table titled "2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00)" offers estimations of last-minute ticket sales under the assumption of a 5% decrease in sales, specifically accounting for the potential implementation of dynamic pricing strategies, which would raise the ticket price to $10.00 during specified hours indicated in the tables. The "Profit" cell positioned at the bottom right corner of each figure presents the cumulative estimate of profit for the specific day of the week over the entire season, contingent upon the accurate implementation of the dynamic pricing strategy and the corresponding decrease in sales.

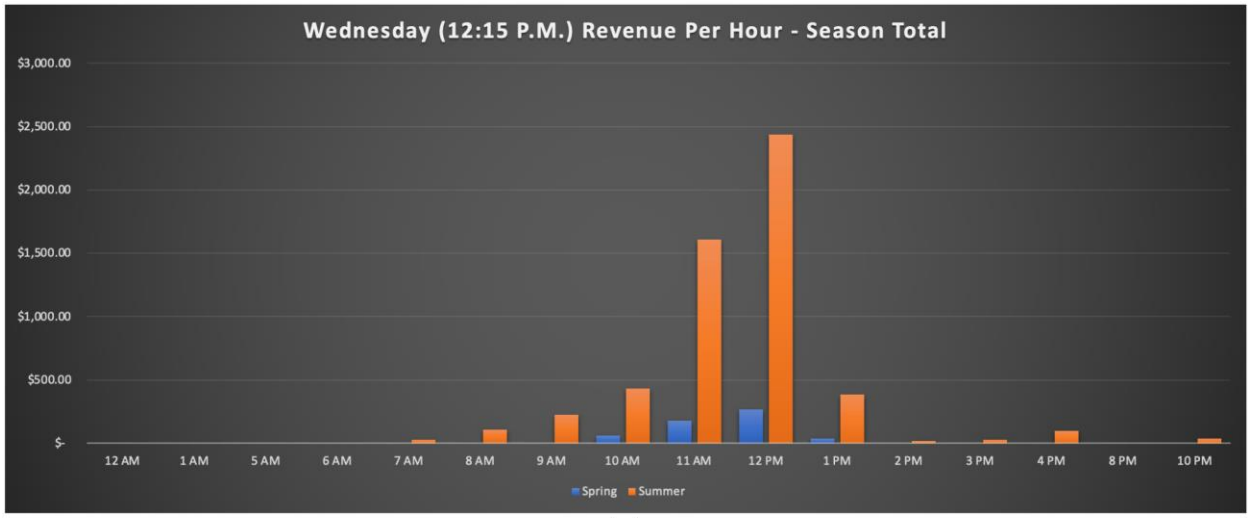



| 2023 Total Last Minute Ticket Sales ($9.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00 PM | $ 351.00 | $ 2,358.00 | $ 2,709.00 |
| 6:00 PM | $ 1,593.00 | $ 3,924.00 | $ 5,517.00 |
| 7:00 PM | $ 1,440.00 | $ 2,403.00 | $ 3,843.00 |
| 8:00 PM | $ 162.00 | $ 1,908.00 | $ 2,070.00 |
| Total | $ 3,546.00 | $ 10,593.00 | $ 14,139.00 |

| 2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00 PM | $ 370.50 | $ 2,489.00 | $ 2,859.50 |
| 6:00 PM | $ 1,681.50 | $ 4,142.00 | $ 5,823.50 |
| 7:00 PM | $ 1,520.00 | $ 2,536.50 | $ 4,056.50 |
| 8:00 PM | $ 171.00 | $ 2,014.00 | $ 2,185.00 |
| Total | $ 3,743.00 | $ 11,181.50 | $ 14,924.50 |

| Profit | $ 785.50 |
|---|---|

Figure 33 Tuesday Revenue per Hour & Estimates Sales Profit with Dynamic Pricing.

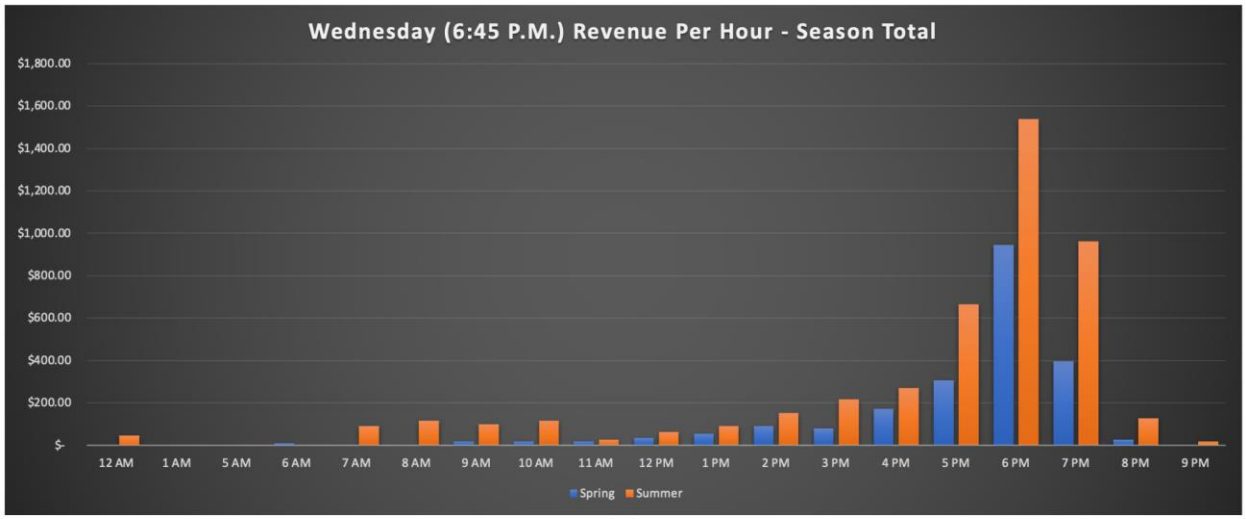



| 2023 Total Last Minute Ticket Sales ($9.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 10:00 AM | $ 63.00 | $ 432.00 | $ 495.00 |
| 11:00 AM | $ 180.00 | $ 1,611.00 | $ 1,791.00 |
| 12:00 PM | $ 270.00 | $ 2,439.00 | $ 2,709.00 |
| 1:00 PM | $ 36.00 | $ 387.00 | $ 423.00 |
| Total | $ 549.00 | $ 4,869.00 | $ 5,418.00 |

| 2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 10:00 AM | $ 66.50 | $ 456.00 | $ 522.50 |
| 11:00 AM | $ 190.00 | $ 1,700.50 | $ 1,890.50 |
| 12:00 PM | $ 285.00 | $ 2,574.50 | $ 2,859.50 |
| 1:00 PM | $ 38.00 | $ 408.50 | $ 446.50 |
| Total | $ 579.50 | $ 5,139.50 | $ 5,719.00 |

| Profit | $ 301.00 |
|---|---|

Figure 34 Wednesday Midday Revenue per Hour & Estimates Sales Profit with Dynamic Pricing.

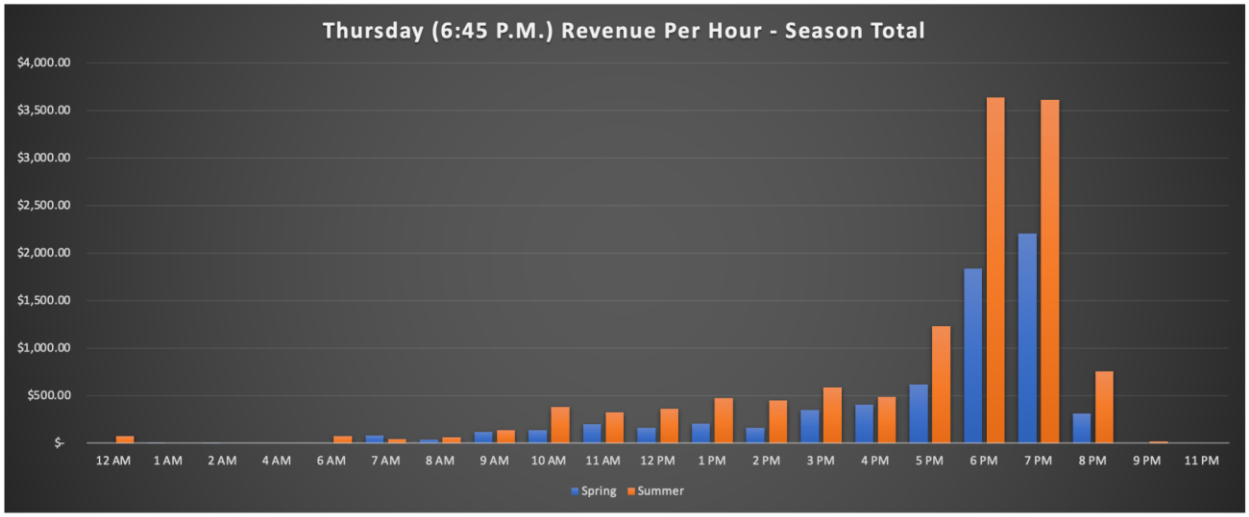



| 2023 Total Last Minute Ticket Sales ($9.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00 PM | $ 306.00 | $ 666.00 | $ 972.00 |
| 6:00 PM | $ 945.00 | $ 1,539.00 | $ 2,484.00 |
| 7:00 PM | $ 396.00 | $ 963.00 | $ 1,359.00 |
| 8:00 PM | $ 27.00 | $ 126.00 | $ 153.00 |
| Total | $ 1,674.00 | $ 3,294.00 | $ 4,968.00 |

| 2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00 PM | $ 323.00 | $ 703.00 | $ 1,026.00 |
| 6:00 PM | $ 997.50 | $ 1,624.50 | $ 2,622.00 |
| 7:00 PM | $ 418.00 | $ 1,016.50 | $ 1,434.50 |
| 8:00 PM | $ 28.50 | $ 133.00 | $ 161.50 |
| Total | $ 1,767.00 | $ 3,477.00 | $ 5,244.00 |

| Profit | $ 276.00 |
|---|---|

Figure 35 Wednesday Evening Revenue per Hour & Estimates Sales Profit with Dynamic Pricing.

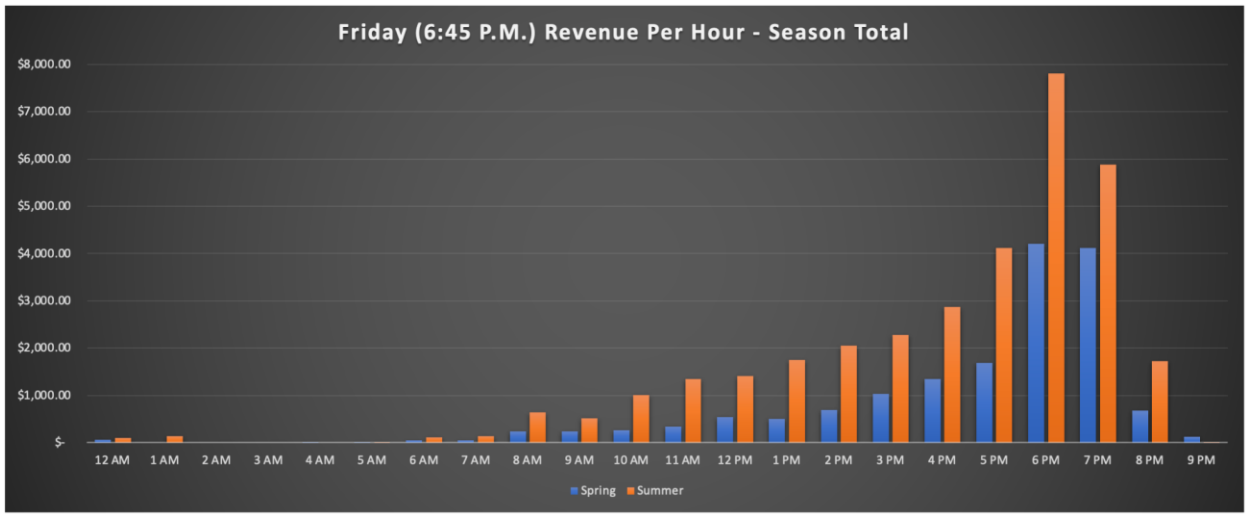



| 2023 Total Last Minute Ticket Sales ($9.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00 PM | $ 621.00 | $ 1,233.00 | $ 1,854.00 |
| 6:00 PM | $ 1,836.00 | $ 3,636.00 | $ 5,472.00 |
| 7:00 PM | $ 2,205.00 | $ 3,609.00 | $ 5,814.00 |
| 8:00 PM | $ 315.00 | $ 756.00 | $ 1,071.00 |
| Total | $ 4,977.00 | $ 9,234.00 | $ 14,211.00 |

| 2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00 PM | $ 655.50 | $ 1,301.50 | $ 1,957.00 |
| 6:00 PM | $ 1,938.00 | $ 3,838.00 | $ 5,776.00 |
| 7:00 PM | $ 2,327.50 | $ 3,809.50 | $ 6,137.00 |
| 8:00 PM | $ 332.50 | $ 798.00 | $ 1,130.50 |
| Total | $ 5,253.50 | $ 9,747.00 | $ 15,000.50 |

| Profit | $ 789.50 |
|---|---|

Figure 36 Thursday Revenue per Hour & Estimates Sales Profit with Dynamic Pricing.

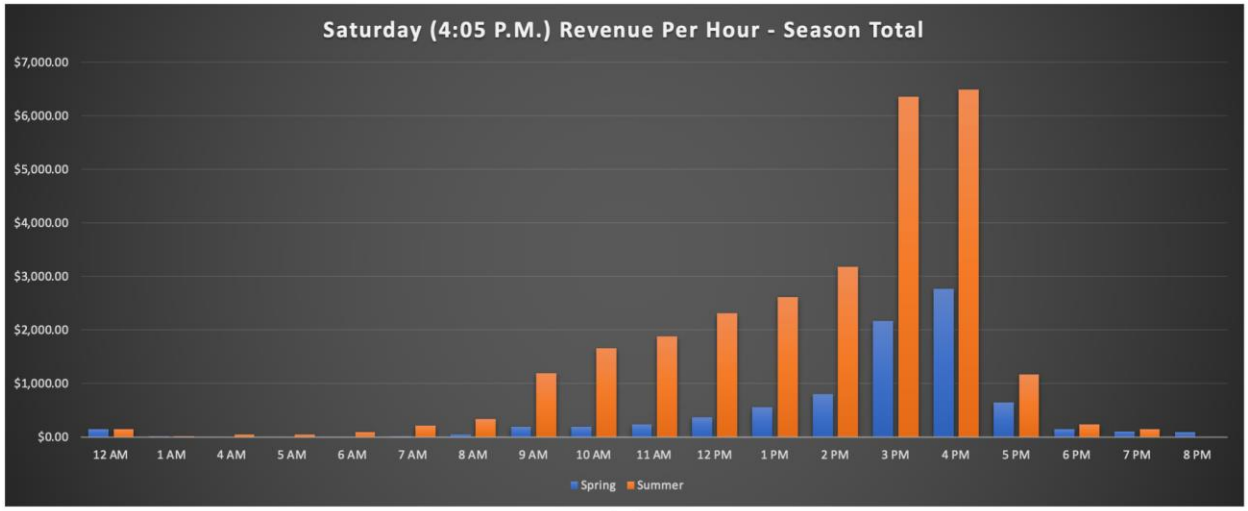



| 2023 Total Last Minute Ticket Sales ($9.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00:00 PM | $ 1,692.00 | $ 4,122.00 | $ 5,814.00 |
| 6:00:00 PM | $ 4,212.00 | $ 7,803.00 | $ 12,015.00 |
| 7:00:00 PM | $ 4,122.00 | $ 5,877.00 | $ 9,999.00 |
| 8:00:00 PM | $ 684.00 | $ 1,719.00 | $ 2,403.00 |
| Total | $ 10,710.00 | $ 19,521.00 | $ 30,231.00 |

| 2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 5:00 PM | $ 1,786.00 | $ 4,351.00 | $ 6,137.00 |
| 6:00 PM | $ 4,446.00 | $ 8,236.50 | $ 12,682.50 |
| 7:00 PM | $ 4,351.00 | $ 6,203.50 | $ 10,554.50 |
| 8:00 PM | $ 722.00 | $ 1,814.50 | $ 2,536.50 |
| Total | $ 11,305.00 | $ 20,605.50 | $ 31,910.50 |

| Profit | $ 1,679.50 |
|---|---|

Figure 37 Friday Revenue per Hour & Estimates Sales Profit with Dynamic Pricing.

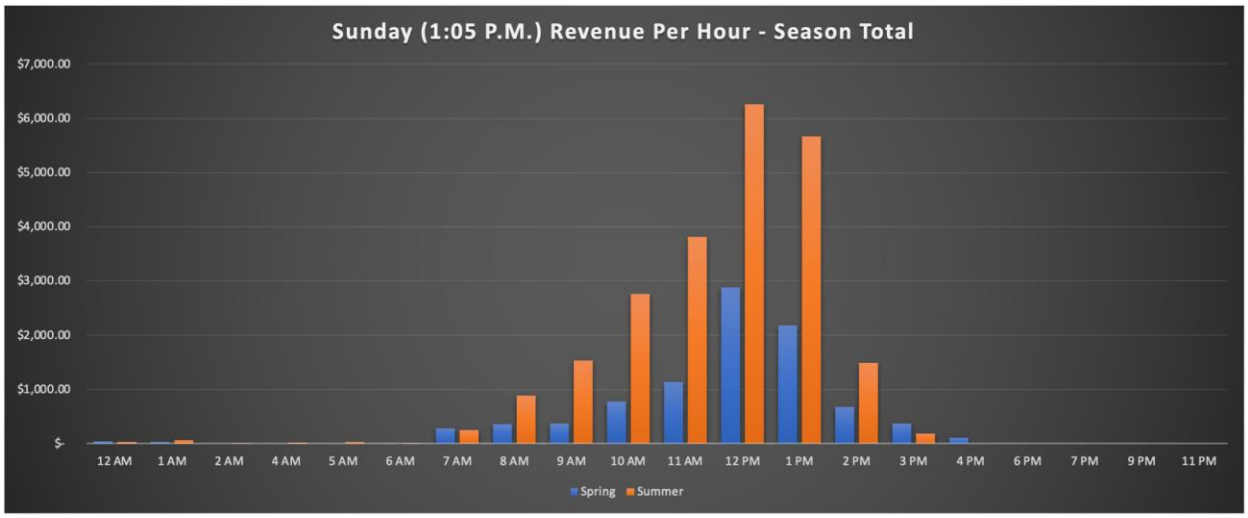



| 2023 Total Last Minute Ticket Sales ($9.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 2:00 PM | $ 810.00 | $ 3,186.00 | $ 3,996.00 |
| 3:00 PM | $ 2,169.00 | $ 6,354.00 | $ 8,523.00 |
| 4:00 PM | $ 2,772.00 | $ 6,498.00 | $ 9,270.00 |
| 5:00 PM | $ 648.00 | $ 1,170.00 | $ 1,818.00 |
| Total | $ 6,399.00 | $ 17,208.00 | $ 23,607.00 |

| 2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 2:00 PM | $ 855.00 | $ 3,363.00 | $ 4,218.00 |
| 3:00 PM | $ 2,289.50 | $ 6,707.00 | $ 8,996.50 |
| 4:00 PM | $ 2,926.00 | $ 6,859.00 | $ 9,785.00 |
| 5:00 PM | $ 684.00 | $ 1,235.00 | $ 1,919.00 |
| Total | $ 6,754.50 | $ 18,164.00 | $ 24,918.50 |

| Profit | $ 1,311.50 |
|---|---|

Figure 38 Saturday Revenue per Hour & Estimates Sales Profit with Dynamic Pricing.




| 2023 Total Last Minute Ticket Sales ($9.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 11:00 AM | $ 1,134.00 | $ 3,816.00 | $ 4,950.00 |
| 12:00 PM | $ 2,880.00 | $ 6,255.00 | $ 9,135.00 |
| 1:00 PM | $ 2,178.00 | $ 5,670.00 | $ 7,848.00 |
| 2:00 PM | $ 675.00 | $ 1,485.00 | $ 2,160.00 |
| Total | $ 6,867.00 | $ 17,226.00 | $ 24,093.00 |

| 2024 Estimated Total Last Minute Ticket Sales (5% Decrease, $10.00) | | | |
|---|---|---|---|
| Sales Hour | Spring | Summer | Grand Total |
| 11:00 AM | $ 1,197.00 | $ 4,028.00 | $ 5,225.00 |
| 12:00 PM | $ 3,040.00 | $ 6,602.50 | $ 9,642.50 |
| 1:00 PM | $ 2,299.00 | $ 5,985.00 | $ 8,284.00 |
| 2:00 PM | $ 712.50 | $ 1,567.50 | $ 2,280.00 |
| Total | $ 7,248.50 | $ 18,183.00 | $ 25,431.50 |

| Profit | $ 1,338.50 |
|---|---|

Figure 39 Sunday Revenue per Hour & Estimates Sales Profit with Dynamic Pricing.

## 10.1 Client Hypothesis

In the exploration of dynamic pricing strategies for the WooSox, the team engaged in formulating hypotheses to gauge potential customer responses. Through extensive evaluation and collaborative discussions with sponsors, three distinct customer types emerged in relation to a price increase in last-minute ticket sales.

The first customer type is characterized by the last-minute ticket sales buyer who expresses dissatisfaction with the ticket price increase. While it is acknowledged that some customers may reach a point of discontent where they cease purchasing such tickets, the current hypothesis suggests that this type of customer might shift their behavior to purchase tickets at an earlier date. This assumption is based on the idea that, despite initial dissatisfaction, these customers may choose to secure their tickets in advance.

The second customer type comprises those who either do not notice the ticket sales increase or are unaffected by it. Consequently, this group of customers is expected to maintain a similar purchasing timeline, showing little to no deviation in their behavior despite the pricing adjustment.

The third and final customer type represents new customers who typically wait until the last minute to purchase tickets. However, due to unforeseen circumstances or a lack of commitment, these customers may decide not to attend the game. The introduction of a sales increase prompts a shift in their behavior, leading them to opt for purchasing tickets in advance. This decision is

seen as a means for them to commit to attending the game despite their initial inclination to wait until the last minute.

It is crucial to underscore that these assumptions are theoretical hypotheses crafted for the WooSox organization to conduct a comprehensive analysis of potential shifts in last-minute ticket sales from a customer perspective following dynamic pricing adjustments. As the team moves forward, these hypotheses will serve as a valuable foundation for understanding and interpreting customer behaviors in response to the implemented pricing strategies.

## 10.2 Income Variability

Regression models play a crucial role in determining income variability for dynamic pricing offerings by identifying pertinent variables such as seasonality, weather conditions and customer demographics. This quantitative understanding is achieved through regression analysis, enabling businesses to grasp the relationships between these variables and income variability. This approach supports data-driven decision-making when implementing dynamic pricing strategies and facilitates demand forecasting by analyzing historical data and predicting future trends. It empowers businesses to make timely adjustments to pricing in response to fluctuations in customer behavior.

In the context of analyzing last-minute ticket sales for the WooSox organization, the use of regression models became particularly relevant for accurately determining which variables influence demand. The analysis focused on data collected for all tickets sold on the day of the game, with a strategic emphasis on the 3-4 hours leading up to the game, sometimes extending into the first hour of the game (depending on the day of the week). This time frame was chosen due to its correlation with the highest volume of ticket sales. Following extensive analysis, the team developed regression models A & B, refining their understanding of the factors influencing last-minute ticket sales to optimize pricing strategies accordingly.

### 10.2.1 Regression Model A

Regression Model A serves the purpose of calculating the estimated total income three hours before the start of the game, considering specific factors. These factors include weather conditions, measured by the amount of inches of rain fallen throughout the day; temperature in Fahrenheit degrees; and the week split, designated as 1 for Friday through Sunday or 0 for any other day of the week.

To ensure the accuracy of the variables considered, the team meticulously analyzed the highlighted numbers in Figure 40. The *R square value*, indicating that the formula accounts for approximately 43% of the instances, suggests the existence of other variables not taken into account that could explain the remaining instances. Put succinctly, the R-square gauges the collective predictive capability of the independent variables within the regression model in explaining the variance observed in the dependent variable. The *F-value*, a measure of the

formula's statistical significance, surpasses the threshold of 4, implying its significance. The *significance F* provides insights into the reliability of the results, with a conclusion that values less than 0.05 are deemed significant. Similarly, the *P-value*, with a threshold of 0.05, signifies that the variable enhances the formula's results if its value is less than this threshold. In Figure 40, the intercept has a higher *P-value*, which, in certain cases, doesn't diminish the formula's value as long as it remains within an acceptable range and is not significantly higher.

**SUMMARY OUTPUT**

| *Regression Statistics* | |
|---|---|
| Multiple R | 0.655713126 |
| R Square | 0.429959703 |
| Adjusted R Square | 0.405873493 |
| Standard Error | 1109.456159 |
| Observations | 75 |

ANOVA

| | *df* | *SS* | *MS* | *F* | *Significance F* |
|---|---|---|---|---|---|
| Regression | 3 | 65917516.43 | 21972505.48 | 17.85086604 | 9.79855E-09 |
| Residual | 71 | 87393400.72 | 1230892.968 | | |
| Total | 74 | 153310917.1 | | | |

| | *Coefficients* | *Standard Error* | *t Stat* | *P-value* | *Lower 95%* | *Upper 95%* | *Lower 95.0%* | *Upper 95.0%* |
|---|---|---|---|---|---|---|---|---|
| Intercept | -1277.180737 | 870.7722142 | -1.466721969 | 0.146867218 | -3013.451218 | 459.0897448 | -3013.451218 | 459.0897448 |
| Weather Condition | -876.7073603 | 254.7154824 | -3.441908407 | 0.000971929 | -1384.595607 | -368.8191134 | -1384.595607 | -368.8191134 |
| Temperature | 58.84615559 | 12.94916613 | 4.5443973 | 2.2058E-05 | 33.02625167 | 84.66605951 | 33.02625167 | 84.66605951 |
| Week Split | 1386.444453 | 258.3102279 | 5.367361811 | 9.55429E-07 | 871.3884868 | 1901.500418 | 871.3884868 | 1901.500418 |

Figure 40 Regression Model A

Having validated the significance of the variables incorporated into the regression model, the team proceeded to estimate a formula comparable to the one depicted in Figure 41. Drawing on data from the 2023 season, the regression model inputs were determined as **A** = -1277.18, **B** = -876.71, **C** = 58.85, and **D** = 1386.44. **A**, representing the intercept, denotes the baseline change in sales. **B** signifies the slope corresponding to anticipated precipitation, while **C** adjusts for temperature fluctuations. **D** serves to differentiate between weekday and weekend games, encompassing Fridays as part of the weekend.

In more basic terms, **A** is negative because for every day there is a loss of -$1277.18 if every other coefficient was multiplied by 0, which could never happen. Therefore, it shows an intercept in the axis were based on the formula, the income would be $0.00. **B** shows there would be a decrease of $876.71 for every inch of precipitation, which isn't much considering in 93.33% of the days of the 2023 season there was less than 1.00 inch of precipitation. **C** is multiplied by the temperature in degrees Fahrenheit, with the lowest ever recorded throughout the season being 40°F, which assumes a minimum increase of 2354.00 to the formula. It can also signify an increase in revenue by $58.85 with every one-degree rise in temperature. Lastly, the **D** coefficient adds an increase of $1386.44 if the game is on a weekend.

Put simply, **A** is negative, indicating a loss of -$1277.18 per day if all other coefficients were zero, which is practically unattainable. Thus, it signifies the baseline sales level when all other factors are absent. Despite its negative value, it represents the hypothetical income when sales are non-existent. **B** reveals a moderate decrease of $876.71 for each inch of precipitation, considering that nearly 93.33% of days in the 2023 season experienced less than an inch of precipitation. **C**, contingent on temperature in Fahrenheit, yields a minimum sales increase of $2354.00, derived from the lowest recorded temperature of 40°F during the season. Finally, **D** contributes a $1386.44 increase in sales for weekend games.

At this point, this data offers a preliminary estimate of the expected income for last-minute ticket sales in the upcoming 2024 season. However, for the formula to maintain its value, it requires additional data inputs from the 2024 season and subsequent seasons. Without ongoing data inputs, the formula would predominantly reflect the characteristics of the 2023 season, as illustrated in Figure 42, where the formula is employed to estimate ticket sales in comparison to the actual sales of such season.

Moreover, in line with the findings from the regression analysis, Figure 42 provides a proper portrayal of how the formula covers 43% of the instances. Such a percentage indicates that other variables not accounted for in the model may influence the remaining instances. Section 10.2.3 outlines additional variables that should be considered for enhancing the accuracy and scope of this formula, emphasizing the importance of ongoing refinement and adaptation based on evolving data inputs from successive seasons.


Figure 41 Income Formula for Total Last-Minute Ticket Sales.

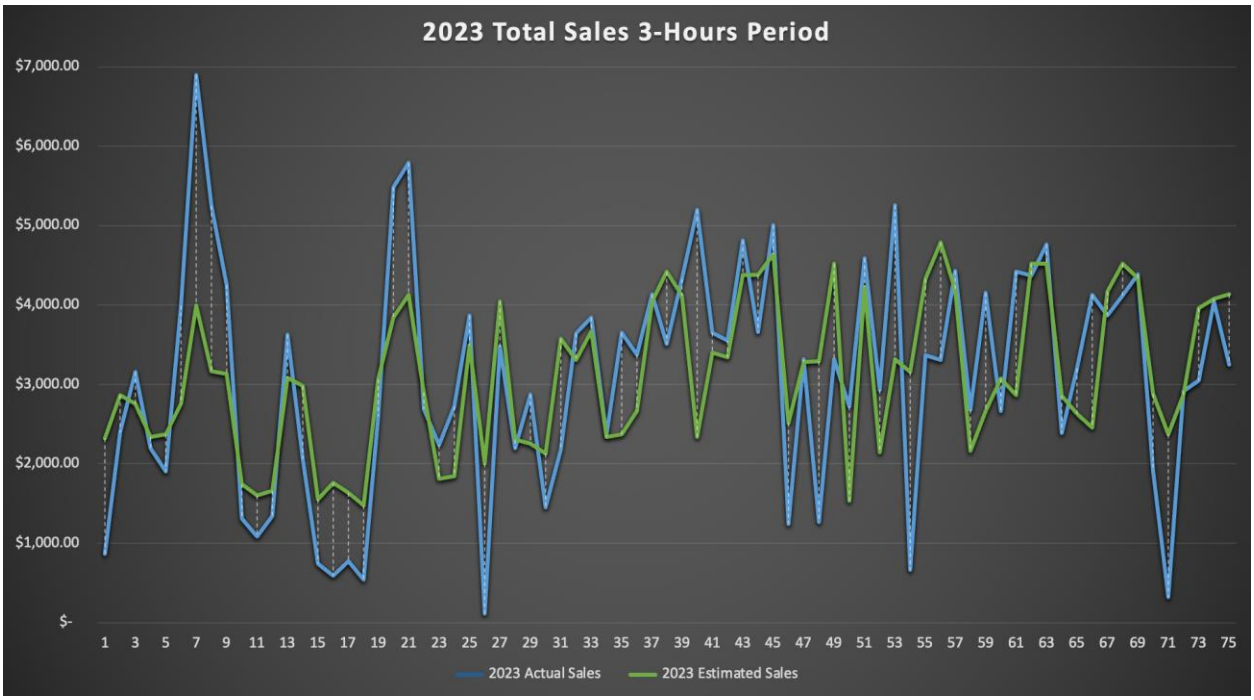



Figure 42 2023 Total sales vs Total estimate sales using income formula.

### 10.2.2 Regression Model B

Similar to regression Model A, regression Model B operates to determine income for last-minute ticket sales. However, it distinguishes itself by focusing exclusively on the Standing Room Only (SRO) section, renowned for having the highest number of sales and revenue across the board. The values of the highlighted factors, crucial for assessing the significance of the regression model, are presented in Figure 43. Additional insights into why these values hold relevance can be explored in section 10.2.1. It's essential to acknowledge that the *R square value* from the analysis indicates that the formula is accurate for approximately 40% of the instances, emphasizing the need for further consideration of potential variables for the formula.

**SUMMARY OUTPUT**

| *Regression Statistics* | |
|---|---|
| Multiple R | 0.629599719 |
| R Square | 0.396395806 |
| Adjusted R Square | 0.370891404 |
| Standard Error | 831.6579265 |
| Observations | 75 |

ANOVA

| | *df* | *SS* | *MS* | *F* | *Significance F* |
|---|---|---|---|---|---|
| Regression | 3 | 32249620.91 | 10749873.64 | 15.54225024 | 7.18917E-08 |
| Residual | 71 | 49107498.37 | 691654.9067 | | |
| Total | 74 | 81357119.28 | | | |

| | *Coefficients* | *Standard Error* | *t Stat* | *P-value* | *Lower 95%* | *Upper 95%* | *Lower 95.0%* | *Upper 95.0%* |
|---|---|---|---|---|---|---|---|---|
| Intercept | -1218.774126 | 652.7383786 | -1.867170931 | 0.066006154 | -2520.297487 | 82.7492349 | -2520.297487 | 82.7492349 |
| Weather Condition | -528.6669302 | 190.9369273 | -2.768804011 | 0.007172261 | -909.3843501 | -147.9495102 | -909.3843501 | -147.9495102 |
| Temperature | 35.70877761 | 9.706806861 | 3.67873577 | 0.000452658 | 16.35395445 | 55.06360077 | 16.35395445 | 55.06360077 |
| Week Split | 1089.395772 | 193.6315796 | 5.626126557 | 3.4176E-07 | 703.3053684 | 1475.486176 | 703.3053684 | 1475.486176 |

Figure 43 Regression Model B

Moreover, Figure 44 serves as an illustration of the inclusion of an additional variable into the formula, which, based on the significance of the *P-Value*, did not contribute any discernible value. The variable under consideration was the *season*, categorized into spring (equivalent to 0 in the formula) and summer (equivalent to 1 in the formula). This same variable was tested for Regression Model A, yielding similar results of no additional value. This is evident in the marginal increase in the *R square value* from Figure 43 to Figure 44, indicating that the formula still only accounts for 40% of the instances. The limited impact of the season variable emphasizes the importance of carefully evaluating the relevance and significance of each variable to enhance the overall accuracy of the regression model.

**SUMMARY OUTPUT**

| *Regression Statistics* | |
|---|---|
| Multiple R | 0.631218629 |
| R Square | 0.398436957 |
| Adjusted R Square | 0.364061926 |
| Standard Error | 836.1598978 |
| Observations | 75 |

ANOVA

| | *df* | *SS* | *MS* | *F* | *Significance F* |
|---|---|---|---|---|---|
| Regression | 4 | 32415683.05 | 8103920.764 | 11.59088284 | 2.81413E-07 |
| Residual | 70 | 48941436.23 | 699163.3746 | | |
| Total | 74 | 81357119.28 | | | |

| | *Coefficients* | *Standard Error* | *t Stat* | *P-value* | *Lower 95%* | *Upper 95%* | *Lower 95.0%* | *Upper 95.0%* |
|---|---|---|---|---|---|---|---|---|
| Intercept | -1453.818836 | 814.4275926 | -1.785080527 | 0.078580545 | -3078.143451 | 170.5057795 | -3078.143451 | 170.5057795 |
| Season | -142.8393093 | 293.0905066 | -0.487355633 | 0.627529125 | -727.3898929 | 441.7112742 | -727.3898929 | 441.7112742 |
| W Condition | -527.7031239 | 191.9807026 | -2.748730037 | 0.007604663 | -910.5965619 | -144.8096858 | -910.5965619 | -144.8096858 |
| Temperature | 40.7256998 | 14.18502418 | 2.871034923 | 0.005408256 | 12.43456115 | 69.01683846 | 12.43456115 | 69.01683846 |
| WeekSplit | 1089.029032 | 194.6812099 | 5.593909308 | 4.01456E-07 | 700.7496022 | 1477.308462 | 700.7496022 | 1477.308462 |

Figure 44 Regression Model B with Extra Variables

Following the identification of the appropriate variables for the formula, the team crafted a formula based on the one depicted in Figure 45. Utilizing data from the 2023 season, the regression model inputs were established as **A** = -1218.77, **B** = -528.67, **C** = 35.71, and **D** = 1089.40. In this model, **A** represents the intercept, signifying the baseline change in sales. **B** denotes the slope corresponding to anticipated precipitation, while **C** adjusts for temperature fluctuations. D serves to differentiate between weekday and weekend games, including Fridays as part of the weekend.

To simplify, **A** is negative, indicating a loss of -$1218.77 per day when all other coefficients are zero—a scenario practically unattainable. Consequently, it signifies the baseline sales level when all other factors are absent. Despite its negative value, it represents the hypothetical income when sales are non-existent. **B** reveals a modest decrease of $528.67 for each inch of precipitation, considering that nearly 93.33% of days experienced less than an inch of precipitation during the season. **C**, contingent on temperature in Fahrenheit, yields a minimum sales increase of $1428.40, derived from the lowest recorded temperature of 40°F during the season. It can also indicate a revenue increase of $35.71 for each one-degree increase in temperature. Finally, **D** contributes a $1089.40 increase in sales for weekend games.

This formula provides an initial estimate of the expected income for Standing Room Only (SRO) last-minute ticket sales in the upcoming 2024 season. However, similar to Regression Model A, to preserve its value, it necessitates ongoing data inputs from the 2024 season and subsequent seasons. Subsequently, the formula is presented in Figure 46 using data from the 2023 season, illustrating how the formula accounts for roughly 40% of the cases throughout that season.

While Regression Model B bears similarities to Regression Model A, it offers valuable insights into the specific data that could be utilized if the WooSox organization decides to create an income formula for each seating section in the ballpark. Each section may be influenced by distinct parameters concerning last-minute ticket sales, emphasizing the potential benefits of tailoring formulas to different sections to enhance accuracy and effectiveness in predicting income variability.


Figure 45 Income Formula for SRO Last-Minute Ticket Sales.

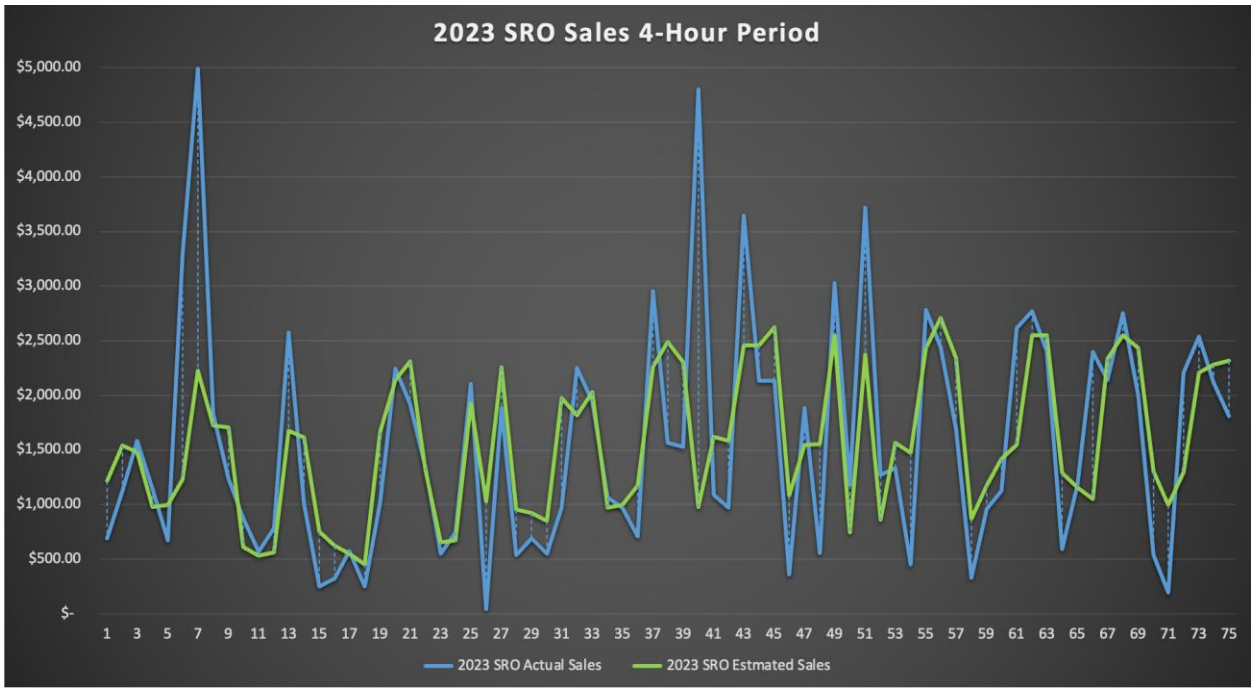



Figure 46 2023 SRO Sales vs SRO Estimate Sales using Income Formula.

### 10.2.3 Future Work

After the culmination of the initial regression models, there is room for further improvement. For starters, creating a regression model for each section in the ballpark, could provide a set of income formulas for the top selling seats, which will account for a higher percentage of instances, considering each section sells with regards to different parameters.

On another note, there are a couple of variables the team believes would make an impact; however most of these points require data gathering over each season. The first one, is matchup popularity. This is something the WooSox organization can historically analyze to determine which matchups have a higher volume of selling tickets. The team was unable to consider this variable considering there are many factors gauged within a matchup's popularity.

In addition, the team believes the opposing team rank on the day of the game might have an impact on last minute ticket sales. However, this data is not gathered publicly, so this is something the WooSox org. will have to gather themselves. In doing so, it would also be valuable to add the WooSox's standing record or rank on the day of the game.

The team also finds value in analyzing the variability of the start times of the game and the dates. Most games differ in start time with regards to their day of the week, however the team believes there could be an improvement if they take into account season of the year. Although the team acknowledges the value in maintaining a standard hour each date for fan containment, it might increase their profits if the organization tests other times on certain dates. Regardless of doing so, it is anticipated to be of value if added into the regression analysis.

A variable believed to be of extreme importance is extra events. The team has noticed peaks on certain dates, with no explanation, which is believed to be caused by the extra events the organization hosts. However, considering these events vary in audience target, the team believes the WooSox organization will be more adequate to categorize the variable properly. In doing so, there will also be valuable data as to which events have lower demand. Besides that, there will be enough data to contrast the ticket revenue, caused by the event, versus the costs of the event.

When it comes to building a regression analysis for each seating section, there is a category the team believes will account for a high percentage of instances, which is the seating availability. Each seating section has a certain amount of seating availability, and there are many ways in which this data can be analyzed for last-minute ticket sales. It can be contrasted with the entire seating capacity, to analyze how many seats are left at a specific time, or more specifically, categorizing by the amount of seats together that are left (e.g. five spots left with four seats together, and three with seven seats together). This data will shine light on which customers change to the SRO due to ticket unavailability.

Another variable of extreme importance is an MLB Star Appearance. In the 2021 season, MLB Star Chris Sale was coming out of an injury and had the opportunity to play a game at Polar Park

on his way back to the Major League. According to a member of the WooSox organization, that day had “the highest number of tickets sold in WooSox history!” This data would clearly impact the income variability and is one set that will have to be manually gathered by the WooSox organization. In addition, the team notes this data point should be analyzed for the opposing team as well, considering this does have an impact on opposing fans who live near the stadium or are in town around time of the game.

Lastly, one point of data the organization could delve into is customer demographics. To do so, the corporation would have to create a survey for fans to fill out, possibly offering them rewards for doing so. Regardless of the rewards, this survey could shine a light on customer decisions based on age, income, amount of people they attend with, interest in the extra events… This final thought was left to the WooSox organization considering they can use the proper amount of time to analyze which data points are of interest, and for how long the survey could be active.

# 11.0 Assessment

## 11.1 General Assessment

The MQP team was successful in their main goal: to develop a mobile app for WooSox. This app is able to perform the two main tasks that were asked for: allow the user to interact with the tickets.com ticketing system and allow the user to interact with the WooSox Rewards program in one place. In addition the team was able to add other great features for ballpark fans like an updated ballpark map, a consistently updated menu of the concessions, and many other minor services to make the ballpark experience better. There were some aspects that were unable to be complete. The main recurring blocker for this was the inability to have a dedicated server for the app. This hindered how much the team was able to make the app do. However, the most important aspect that was looked at was that it is the app that was made worth using, and the answer for the team was clearly yes.

## 11.2 Business Learnings

### 11.2.1 Leadership

Before initiating the project, the team meticulously defined roles and responsibilities, aligning them with the core principles of Agile methodology. In our setup, David shouldered the crucial position of Product Owner, serving as the primary link between sponsors and the team, while also actively contributing as a full-stack Developer. Miguel undertook the roles of Scrum Master and Developer, responsible for coordinating daily scrum meetings, guiding sprint planning, and facilitating reviews and retrospectives. Simultaneously, he played a vital role as a front-end developer, providing feedback and analysis from a user’s perspective.

Meanwhile, Mohamed functioned as a full-stack Developer, swiftly addressing technical challenges that arose throughout the project's duration while focusing on the interface development. The assignment of roles was a purposeful decision, influenced by the diverse backgrounds and experiences of team members. Both David and Mohamed brought technical expertise rooted in computer science backgrounds, complemented by Miguel’s valuable business background. This strategic distribution of responsibilities established a robust foundation for the team to function cohesively.

Each team member embraced leadership responsibilities within their designated domain, fostering a streamlined development process. The team's adept implementation of Agile methods became evident as they operated seamlessly, ensuring the project's success.

### 11.2.2 Work Environment and Culture

Throughout the project, the team fostered a dynamic work environment and a culture of collaboration. Weekly meetings with dedicated advisors served as a cornerstone, providing

valuable guidance and facilitating open discussions on project progress. This consistent engagement ensured that the team remained aligned with project goals and addressed any challenges promptly.

The team's collaboration extended beyond internal interactions, with frequent meetings occurring once or twice a week at the facilities of the esteemed project sponsors, the WooSox. These sessions played a pivotal role in resolving blockers swiftly and gaining insights into the sponsors' expectations. The in-person nature of these meetings allowed for deeper discussions, fostering a constructive environment for idea exchange and problem-solving.

Despite the challenges posed by external vendors and communication complexities, the collaboration between the team and sponsors was a driving force in overcoming obstacles. The sponsors' open-minded approach and active involvement in the development process contributed to a culture of mutual understanding, enabling the team to tailor the app framework to enhance the overall fan experience.

This collaborative work environment, enriched by weekly engagements with advisors and frequent in-person meetings with sponsors, not only facilitated efficient project management but also nurtured a culture of adaptability and constructive feedback. The team's commitment to these collaborative practices significantly influenced the success of the Polar Park app project.

## 11.3 Industrial Engineering Learnings

Throughout the project, significant learnings were derived from the application of Industrial Engineering methodologies, particularly focusing on Process Flow Chart and Regression Models. The Process Flow Chart, detailed in Section 6, provided valuable insights into the intricacies of app development, documenting existing functionalities while strategically planning for future enhancements. By refining and iterating on the original chart, the team gained a deeper understanding of user journey optimization and interface improvements, enhancing overall user experience.

Similarly, delving into Regression Models yielded invaluable insights into predicting various factors influencing business outcomes. Through analysis, the team acquired key statistical metrics such as the R value, F value, and P value, shedding light on the significance of relationships between variables within the sports business context. These models not only provided a quantitative basis for decision-making but also fostered a deeper understanding of the dynamic nature of sales and revenue optimization strategies.

Additionally, the implementation of dynamic pricing strategies offered valuable lessons in revenue management and fan engagement. By adjusting ticket prices based on factors such as demand and historical sales data, the team gained insights into customer behavior and market dynamics. These experiences highlighted the importance of agility and adaptability in pricing

strategies, as well as the need for careful consideration of customer responses and market reactions.

Looking ahead, the learnings from these methodologies lay the groundwork for future innovations and enhancements in sales strategies. By leveraging technology and data-driven approaches, the team acquired insights to enhance revenue streams and enrich customer experiences in the constantly evolving sports entertainment landscape, with potential applications extending beyond the realm of sports to other industries.

## 11.4 Technical Learnings

There was a lot of technical experience that was gained throughout the creation of this app. To start, while both developers on the team have had experience with React, only one of them had heard of React Native and neither have used it in a project to this scope. Once React Native was learned, the two developers split up in their expertise. Mohamed developed his expertise in UI design. He was able to utilize the design elements chosen from the last team and utilize React Native and its libraries to its fullest potential to create a good-looking app. In this process, he learned and utilized a lot about UI design. David developed his expertise in API connections. He spent most of his developing time creating functional connections between the app and its API. In the process he learned a lot about internet infrastructure and object caching. Both developers then combined their acquired skills to create the app seen here, which developed more skills in Git development, team programming, and product integration. These skills will be invaluable for their futures and are extremely applicable for future employment and education opportunities.

# 12.0 Future Work

While the Polar Park App is in a complete and functional state, there is still much more that could be done for this app to reach its full potential. Here the team outlined some things future teams developing this app should work on to improve it.

## 12.1 Dedicated Server

In section 5.2.1, it was outlined how the lack of a dedicated server was a severe hindrance to what the team wanted to do with the app. The team believes that if the app wants to take the next step forward, it needs its own dedicated server.

### 12.1.1 Dedicated Server Implementation

Implementing a dedicated server would be a large process for the WooSox team. Before they could even get started they would need two things: long term staff hired to build/maintain the server and the actual server itself. The WooSox relies on contractors to do some of the business for the club; however, the IT staff should be hired in house. This would allow this IT staff to become an expert in the app and server, and it could pave the path for the WooSox to improve its online presence with something like a new website that could match the features of the app. Next is the server itself, which should probably be hosted on the cloud, using a service like Google Cloud, Amazon Web Services, or Microsoft Azure. Since the app would be a relatively small server, it is not worth it for the WooSox to rent out its own server in a data center, and the usage of Cloud Computing would be far more economical for the club.

Once the server and staff have been decided, next would be choosing what platform to use for hosting the server. This option doesn't matter a large amount, but using Node.js with a common library like express would make a lot of sense for this. Since Node applications can be written in TypeScript it means that TypeScript could be used for both the server and client side code, making maintainability easier. However, if the staff has better experience with other libraries or languages, like Flask in Python for example, this could be preferable.

From there, the WooSox would want to develop their own account system. This would require a database and a secure tokening system; however, this is important for being able to connect devices to people, and ensuring that different content can be shown to different users.

Lastly, and most importantly, endpoints will need to be made. These endpoints will be the links that the app will connect to for the data it needs to display, similar to how the Fanmaker v2 API is set up, but for all functions of the app. These endpoints would be able to be updated at any time so that they can change as the app changes.

### 12.1.2 Dedicated Server Benefits

Building and maintaining this server would require a significant amount of work from the WooSox; however, there are some clear benefits for them to do this.

First, it would allow the WooSox to build their own account system. This means that they can collect user data for themselves. This could lead to them being able to make better business decisions based on this data. For example, the WooSox would be able to see which type of tickets are redeeming rewards. If, hypothetically, fans buying standing tickets are the ones redeeming rewards the most while fans getting higher priced tickets are ignoring the rewards entirely, that might have a major impact on the value proposition of the WooSox rewards system as a whole. The truth is that it isn't known what types of customers are taking advantage of the rewards system, and this combined account could help resolve that.

Next, it would allow the WooSox to remove the two account systems currently in place. As mentioned earlier, there are two accounts that users have to manage: their tickets.com account and their WooSox Rewards account. This is unnecessarily complex for users, and a one account solution would be much preferred. The creation of the WooSox account system would solve this, as it would allow signing in once to access everything related to the WooSox, including ticketing, rewards, and potentially in seat ordering in the future.

Furthermore, it would also allow the WooSox to stop using Fanmaker for the WooSox Rewards program. Much of what the WooSox are paying Fanmaker for are their servers, and if they are going to switch to hosting their own servers, there isn't much of a purpose to paying Fanmaker for hosting this rewards program when they can just do it themselves.

In addition, this would also allow tickets.com services to be fully implemented into the app. This is fully outlined in section 12.2.

Finally, it gives future teams working on the Polar Park App the ability to make what they truly want. The current app is significantly held back by the limitations of not having a dedicated server. There are many actions that are unsafe to do in an app because it will lead to bad actors being able to steal from the WooSox. Things like API keys, master passwords, and other sensitive data cannot be stored in the app in a safe manner, and these things need to be handled by a dedicated server. And the more things that a future team will want to add to the app, the more impossible this becomes without a dedicated server. While the lack of a dedicated server may save money for the WooSox in the short term, it is leading them to miss out on many opportunities for this app, which may cost them in the long term.

### 12.1.3 Dedicated Server Costs

To fully determine the value of a dedicated server, a minor cost analysis was done to see how much this would cost the WooSox.

The first calculation is for the Server Cost. Google Cloud was used as an example of a cloud computing service, prices are similar between the major cloud computing providers. First a server is needed. Since the app is likely to have a relatively high amount of simultaneous usage, a high CPU count is preferred, so the team chose the CS-highcpu-44 server, which would cost approximately $1384/month ("Pricing | Computer Engine"). The CS-highcpu-44 server is one of many options that Google Cloud offers, and it is chosen for its higher CPU speed (which helps internet servers run) and its compatibility with the higher bandwidth option described below. In addition, during gametime, a higher amount of bandwidth should be used to ensure that connections could go through. This would only need to happen during gametime, as regular internet speeds should be acceptable for when less are using the app. With an average of 12.5 home games per month, and 5 hours of extra bandwidth needed at $0.18/hour, this would add an extra $12/month of server costs in the 6 months where the WooSox are playing. A benefit of this server is that its usage will be very dependent on when games are happening at Polar Park. This means that during its off time, this extra computing power could be used to do something else. For example, the WooSox could consider training a Machine Learning Model to better predict ticketing sales. If this server space is already being paid for, it may as well be used!

The next calculation is for hiring an IT crew to keep the app and server maintained, and for updates to be pushed out. From the experience of the team, a team of 3 would be more than enough to both maintain the server and create new content for the app and other platforms as needed. IT jobs in the Worcester area are paying around $25/hr, which comes out to $4400/month per person (Indeed). Salary only makes up about 68% of the cost of an employee, with benefits and tax costing the employer significantly as well (Cynthia Meyer), so the true cost of the employee would be around $6500/month. While this price is steep, this IT team should be far more capable than just hosting an app. They would be able to recreate the WooSox website to have all the features of the app. They would be able to collect a larger amount of user data which could be analyzed to sell more tickets. This team shouldn't be used just for the app and its server.

The final calculation to consider would be the removal of paying Fanmaker. As outlined in 12.1.2, the WooSox should run their own rewards program if they decide to partake in this venture instead of using Fanmaker for their rewards system. This will save them some money, and make the cost of running and maintaining the servers easier to swallow. Unfortunately, fanmaker does not make their costs public. Similar services run at $2.40/profile, and with over 17,000 accounts registered on WooSox Rewards, this would be $40,800/month saved (Traject Social Pricing). It is unlikely that Fanmaker is costing the WooSox that much, as costs per user tend to go down the higher the number of users that exist. The team estimates that Fanmaker is likely costing around $5000/month for the WooSox, so this will be the number that is used.

With all of these costs combined, the total monthly cost for running a server for this app would be $15,900. While this is quite expensive, it comes with a lot of benefits. On top of all the benefits outlined in section 12.1.2, the app would have much more potential, with the ability to

do whatever the WooSox would want it to do. Next, the WooSox would have a true IT staff, meaning it could truly become the tech-forward ballpark club that it aspires to be. The WooSox will have tons of more user data to work with, so they could truly find out what will truly lead them to maximize their profits. And lastly, the WooSox will have the server space to compute complicated models to understand this data and do something with it. And it is worth mentioning that the largest cost here is from staffing, and while the team believes that a 3 man IT team should be considered for this app, a 2 man team would likely work, making the costs go under $10,000/month.

## 12.2 Integrating Ticketing

A dedicated server holds immense potential for the app, particularly in revolutionizing the ticketing experience—a significant advantage that currently remains untapped. The existing method of handling ticketing, primarily by redirecting users to the tickets.com webpage, presents unnecessary complexities in accessing and purchasing tickets.

The key solution lies in integrating ticketing directly into the app's framework. Retaining the user-friendly front-end developed by the previous team for ticket viewing offers a solid foundation that aligns with the intuitive ticket display format seen in apps like the MLB Ballpark App.

However, the challenge lies in establishing a robust backend system. As highlighted in section 5.2.1, the limitation in integrating ticketing without a dedicated server stems from the risk of exposing the tickets.com API key, potentially leading to misuse. The dedicated server serves as a secure repository for these API keys, ensuring data integrity and preventing unauthorized access. By managing the API keys centrally within the server, it enables secure and controlled data transmission between the app and tickets.com, enhancing both security and efficiency.

Creating a system for purchasing tickets through this method presents additional complexities. While the backend could leverage the existing structure for ticket viewing, developing a new frontend is imperative due to the intricate nature of ticket options. Each ticket type (single game, half season, bundle, full season) and its respective details would likely necessitate dedicated pages, dynamically adjusting as seats are filled and tickets are sold.

Despite the substantial workload required for this setup, the payoff in terms of app usability would be immense. Presently, the app's utility is limited as it mirrors the tickets.com website. Establishing a seamless, in-app ticketing system would offer users unparalleled convenience, transforming their experience with the app and amplifying their enjoyment of ballpark visits. This advancement would distinctly differentiate the app and drive increased user engagement and satisfaction.

## 12.3 MiLB Data

The Polar Park App was meticulously crafted with a primary focus on enhancing the ballpark experience for fans. This aligns seamlessly with the app's current use cases and fulfills the most immediate needs of its users. However, the team recognizes an untapped potential within the app—specifically, a deeper integration with the WooSox Baseball team. Expanding the app's scope to encompass a comprehensive array of WooSox-related content could substantially enrich its value proposition.

An overlooked yet compelling use case involves providing fans with robust access to WooSox-related information. For instance, a fundamental feature would allow users, whether at Polar Park or not, to readily view the live score of ongoing games. This simple yet powerful addition can significantly enhance fan engagement, catering to the passionate interest fans hold in following the team's progress.

This initiative can extend further to encompass an array of enriching content, such as schedules for upcoming games, the team's performance record, curated news articles, and a direct feed from the team's social media platforms like Facebook, Twitter, and Instagram.

However, realizing this vision poses challenges. Acquiring official game data from MiLB, the governing body, necessitates seeking permission, potentially involving data access through a dedicated server, depending on the architecture on MiLB’s side. Additionally, aggregating and presenting news articles and social media feeds may require dedicated servers due to the nature of accessing and delivering this content.

Navigating these hurdles involves not only securing permissions but also potentially setting up dedicated servers to ensure seamless integration and reliable access to this valuable data, thereby elevating the app's capacity to provide comprehensive and up-to-date WooSox-related content.

## 12.4 User Testing

The MQP team prioritized delivering a fully functional app within the allocated 7-week timeframe. However, the expedited timeline limited the thoroughness of essential processes, particularly in the realm of designing user-friendly software. Among these crucial processes, user testing stands out as pivotal. It ensures that the application is used as intended by its developers, yet this aspect often gets sidelined due to time constraints.

User testing holds immense value as it reveals how users interact with the application, which might significantly differ from developers' initial assumptions. However, conducting comprehensive user testing is a time-intensive endeavor. It involves several stages, starting with obtaining approval from an ethics board to ensure fairness and safety in the testing process. Subsequently, assembling a representative focus group, which mirrors the app's user base, is a time-consuming task.

Once formed, this group engages with the app over a specific duration, where they report any encountered bugs, highlight pain points, and provide feedback on the app's functionality and experience. The development team can then utilize this feedback to refine and enhance the application, creating an iterative and improvement-focused cycle.

For future teams, conducting a usability study on the app is strongly recommended. This study can uncover potential pain points or bugs that might exist, addressing which is crucial for enhancing the app's overall usability. Investing time in such a study could significantly elevate the app's user experience and effectiveness.

## 12.5 Application Approval

The most common critique of the app in its current state is that most people cannot currently get it on their phone. The process to get it on a phone currently is to have the source code and use Android Studio or xCode to export the development version of the app to their respective device. Obviously, the real end goal is for the app to appear on the official stores of the respective device: the App Store for iOS and the Google Play Store for Android.

The Google Play Store approval process is relatively simple. Firstly, access to the “Google Play Console” would need to be achieved which is done through a standard Google Account. Once this is achieved, the Android Build of the app (located within the android folder in the repository) can be compiled into one file and uploaded to the console. There may need to be a few things added to the app for Google Play approval, mainly a privacy policy. This privacy policy needs to be accessible somewhere in the app (likely an option on the services page) and also needs to be uploaded to the Google Play Console. Once this is all done, the final things that need to be added are just some warnings about the content of the app - it is safe for all ages to use, it does not contain advertisements, it does use authentication with an email and password, and a couple other minor things (“Prepare your app for review”). Google’s approval process is fairly quick, as its guidelines are relatively loose.

The App Store approval process is a more difficult one. The start is similar to the Android App - building the app and taking the export in the ios folder as the app file. From there the app approval process requires a lot of paperwork and minor app changes. This includes having test accounts that can be used by Apple for ensuring app safety, having a contact link for users to contact if something is wrong with the app, the ability to retract user information from any accounts, the ability to users to delete any of their accounts, potential changes to the privacy policy used, and more (“App Store Review Guidelines”). Whomever is going through App Store approval should go through the privacy guidelines in particular as there are many specific requirements that are important for the app to achieve so it can enter the App Store.

# 13.0 Conclusion

Throughout the team's collaboration with the WooSox, a functional, practical, and stylish app has been successfully created. This app addresses the WooSox's challenges of disjointed systems by providing a unified platform for fans to fulfill all their Polar Park-related activities.

Building upon the framework established by the previous year's group, the team enhanced the app's functionality to create a comprehensive solution. With the addition of over 20 new pages and integration with live API endpoints, the app now offers real-time data that aligns with the WooSox's objectives. It is poised for deployment on both iOS and Android devices, boasting user-friendly features for seamless operation.

Moreover, the team's journey has not only resulted in the development of the app but also in significant skill refinement. The team has acquired invaluable expertise in software design, UI design, backend connections, as well as proficiency in utilizing frameworks such as React and React Native. Additionally, the exploration into methodologies like Agile Scrum and regression analysis has further enriched the team members' software development toolkits.

The  Industrial Engineering methodologies, specifically Process Flow Chart and Regression Models, have been instrumental in shaping the development of the Polar Park app. The Process Flow Chart provided a structured roadmap for app development, ensuring a seamless user experience. Through iterative refinement, the team was able to enhance existing functionalities and strategically plan for future enhancements. Additionally, Regression Models empowered the group to analyze and predict factors influencing last-minute ticket sales, facilitating informed decision-making and revenue optimization strategies. These methodologies have not only contributed to the success of the app but have also expanded the members' understanding of industrial engineering principles in real-world applications.

The exploration into dynamic pricing has shed light on its pivotal role in revenue optimization and fan engagement within the sports industry. By implementing dynamic pricing strategies, the interdisciplinary team has gained insights into the real-time adjustment of ticket prices based on demand and market dynamics. This experience has underscored the importance of strategic pricing adjustments in maximizing revenue while ensuring fan accessibility. Moving forward, the learnings from dynamic pricing will inform future endeavors in enhancing income strategies and enriching the overall customer experiences.

While the app represents a significant milestone, there is still room for growth and enhancement. However, the team has demonstrated the potential of a fully-fledged Polar Park app to the WooSox, paving the way for future endeavors in enhancing fan engagement and operational efficiency. The group takes immense pride in the contributions made to the WooSox organization and remain hopeful that the app will be utilized to its fullest potential, enriching the fan experience and further strengthening the bond between the WooSox and its supporters.

# 14.0 References


*2023 Minor League Attendance*. (n.d.). The Baseball Cube. Retrieved February 10, 2024, from https://www.thebaseballcube.com/content/minor_attendance/2023/

Agency, D. (2021, February 11). *13 benefits of mobile apps for business You need to know in 2023 | DECODE*. https://decode.agency/article/benefits-mobile-apps-business/

*Agile vs. Waterfall*. (n.d.). Retrieved October 12, 2023, from https://www.productplan.com/learn/agile-vs-waterfall/

Anderson, P. (n.d.). *PawSox plan move to Worcester, team says R.I. House is to blame*. The Providence Journal. Retrieved February 10, 2024, from https://www.providencejournal.com/story/sports/2018/08/17/pawsox-plan-move-to-worcester-team-says-ri-house-is-to-blame/11009957007/

Atlassian. (n.d.-a). *Agile Scrum Roles*. Atlassian. Retrieved October 12, 2023, from https://www.atlassian.com/agile/scrum/roles

Atlassian. (n.d.-b). *Scrum Sprints: Everything You Need to Know*. Atlassian. Retrieved October 12, 2023, from https://www.atlassian.com/agile/scrum/sprints

Atlassian. (n.d.-c). *User Stories | Examples and Template*. Atlassian. Retrieved October 12, 2023, from https://www.atlassian.com/agile/project-management/user-stories

Atlassian. (n.d.-d). *What is a scrum master? [+ Responsibilities]*. Atlassian. Retrieved October 12, 2023, from https://www.atlassian.com/agile/scrum/scrum-master

Budnick, D., & Baron, J. (n.d.). *Ticket Masters—The Rise of the Concet Industry and How the Public got Scalped*. ECW Press.

*Building Your Color Palette—Refactoring UI*. (n.d.). Retrieved December 22, 2023, from https://www.refactoringui.com/previews/building-your-color-palette

*FanMaker Loyalty Programs*. (n.d.). Retrieved February 10, 2024, from https://fanmaker.com/loyalty-programs

*Flexible IT Technician Jobs in Worcester, MA | Indeed.com*. (n.d.). Retrieved December 22, 2023, from https://www.indeed.com/jobs?q=it+technician&l=Worcester%2C+MA&from=searchOnHP&vjk=d222b3b4a8945ff9

*Garblovians/react-native-svg-pan-zoom*. (2023). [JavaScript]. garblovians. https://github.com/garblovians/react-native-svg-pan-zoom (Original work published 2017)

Hoory, L., & Bottorff, C. (n.d.). *What is Waterfall Methodology? – Forbes Advisor*. Retrieved October 12, 2023, from https://www.forbes.com/advisor/business/what-is-waterfall-methodology/

*JavaScript With Syntax For Types.* (n.d.). Retrieved December 23, 2023, from https://www.typescriptlang.org/

*Jobtoday/react-native-image-viewing*. (2023). [TypeScript]. JOB TODAY. https://github.com/jobtoday/react-native-image-viewing (Original work published 2019)

Limited, S. S. P. (2023, June 27). Expo vs React Native CLI. *Medium*. https://medium.com/@softworthsolutionspvtltd/expo-vs-react-native-cli-7e47c7630039

McKenzie, C. (n.d.). *JavaScript vs. TypeScript: What's the difference? | TheServerSide*. TheServerSide.Com. Retrieved December 23, 2023, from https://www.theserverside.com/tip/JavaScript-vs-TypeScript-Whats-the-difference

Meyer, C. (n.d.). *How Much Are Your Benefits Really Worth?* Forbes. Retrieved December 22, 2023, from https://www.forbes.com/sites/financialfinesse/2018/09/24/how-much-are-your-benefits-really-worth/

Mir, M. A. (2023, April 2). What are the advantages and disadvantages of using Visual Studio Code or Atom? *Medium*. https://medium.com/@ssc.ahmed.926748/what-are-the-advantages-and-disadvantages-of-using-visual-studio-code-or-atom-d3132bf1af85

Nguyen, Q., Redgate, B., & O'Niel, K. (n.d.). *Redesigning the Worcester Red Sox's Enterprise Mobile Application*.

Papineau, L. (2020, April 15). The Pawtucket Red Sox, By the Numbers. *Rhode Island Monthly*. https://www.rimonthly.com/pawtucket-red-sox-numbers/

*PNG vs. SVG: What are the differences? | Adobe*. (n.d.). Retrieved December 19, 2023, from https://www.adobe.com/creativecloud/file-types/image/comparison/png-vs-svg.html

*Prepare your app for review—Play Console Help*. (n.d.). Retrieved December 22, 2023, from https://support.google.com/googleplay/android-developer/answer/9859455?hl=en

*Pricing | Compute Engine: Virtual Machines (VMs)*. (n.d.). Google Cloud. Retrieved December 22, 2023, from https://cloud.google.com/compute/all-pricing

*React-native-webview/react-native-webview*. (2023). [TypeScript]. react-native-webview. https://github.com/react-native-webview/react-native-webview (Original work published 2018)

*The Official App of Polar Park*. (2021, September 1). App Store. https://apps.apple.com/us/app/the-official-app-of-polar-park/id1534756856

*Ticket Memberships*. (n.d.). MiLB.Com. Retrieved February 10, 2024, from https://www.milb.com/worcester/tickets/ticket-memberships

*Tickets.com Inc—Company Profile and News*. (2024, February 10). Bloomberg.Com. https://www.bloomberg.com/profile/company/TIXX:US

*Traject Social Pricing—Pick the plan that works for you*. (2020, May 14). https://fanbooster.com/landing-pricing/

*What is a Product Backlog? | Agile Alliance*. (2015, December 16). https://www.agilealliance.org/glossary/backlog/

*What is an IDE? - Integrated Development Environment Explained - AWS*. (n.d.). Amazon Web Services, Inc. Retrieved December 23, 2023, from https://aws.amazon.com/what-is/ide/

*What is Git version control?* (n.d.). Retrieved December 23, 2023, from https://about.gitlab.com/topics/version-control/what-is-git-version-control/

*WooSox Theme Nights*. (n.d.). MiLB.Com. Retrieved February 10, 2024, from https://www.milb.com/worcester/tickets/theme-nights

# 15.0 Appendices

## Appendix A: A Comparison Between the Existing Polar Park Ballpark Maps and the New Ones Created for the App

There are multiple different maps for Polar Park already. In section 9.1.3, it is explained that a reason the map was remade was due to the need for an SVG rather than a traditional image. In addition to that, there were other reasons why a new map was created for the app, and this will show why different maps were not used.


Figure 47 The map of Polar Park used for buying tickets.

This map is used for the seating guide both on the WooSox official website and when purchasing a ticket on tickets.com. This map is great for showing the seating options at Polar Park but fails to show any other features at the park. A key requirement of the ballpark map was to show all the different concessions, and since many of those concessions are located behind the main seats and under the suites/200 level seats, that would be impossible on this map.


Figure 48 The map of Polar Park shown throughout the WooSox Website.

This map is seen on the ‘fan guide’ page of the WooSox website, under the letter C and again under the letter W. This map was the basis used for the final maps used in the app and was hugely helpful in the map creation process. The reason it was not used as the final map was due to some minor inaccuracies, some symbols being shown but never explained, a lack of a key, an inconsistent design that was not wanted in the app, and the need to show what is above and below the main concourse area.

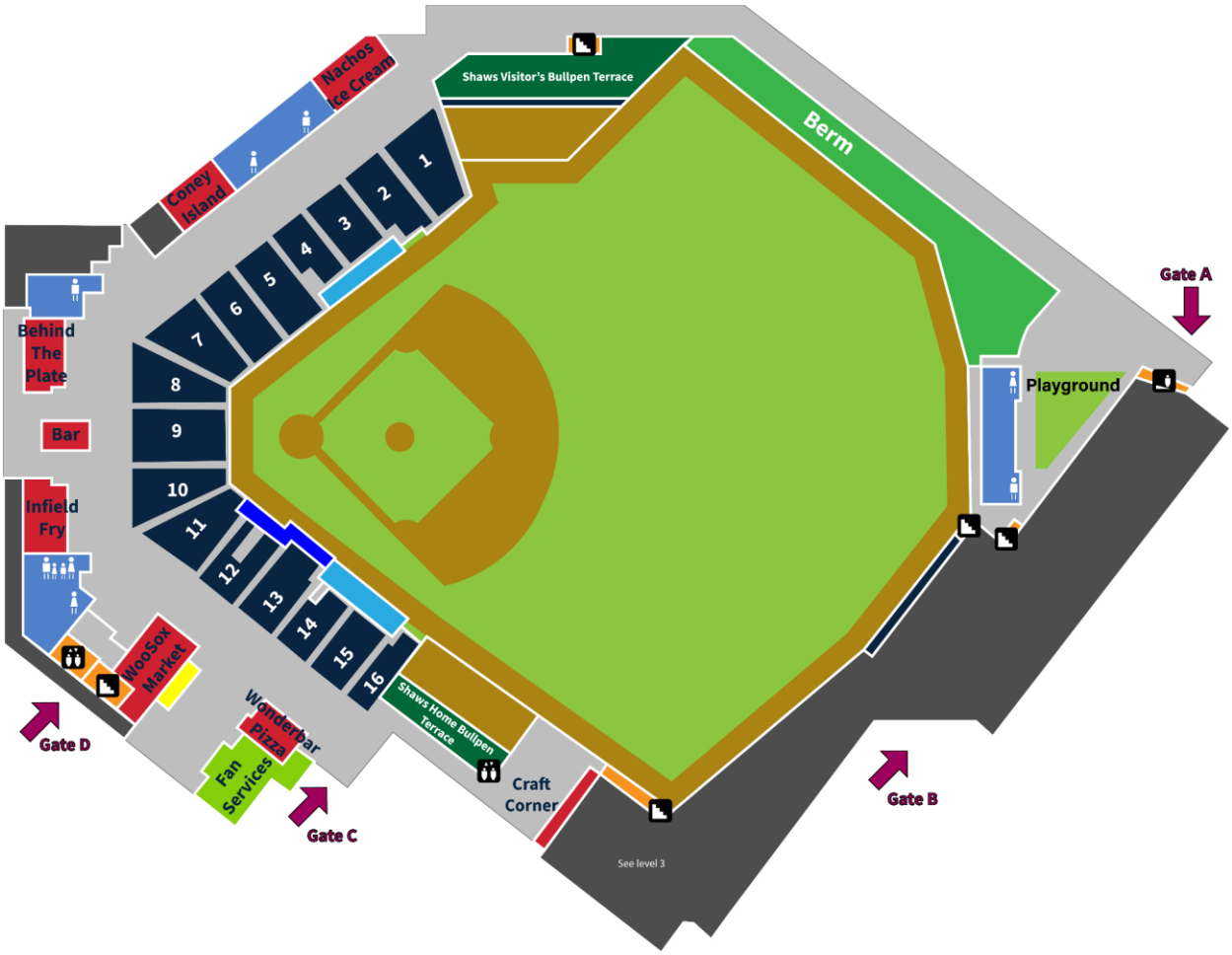



Figure 49 The final map of the main floor of Polar Park.

This was the map created for the app. Note the version in this paper is an export of the original SVG, and therefore is not scalable like the original SVG. This map simplifies a lot of the problems with Figure 48 in a few ways. First it makes clear what some things are. In the original map, the orange areas labeled as the Shaw's Home and Visitor Bullpens are unclear and could make a potential user think that those are the actual bullpens instead of group seating areas. To solve that, in this map, they are labeled as Dark Green to match with the key (Figure 50) and labeled as terraces to make them more distinct. Next, a lot of the unnecessary icons, like the stars and circles, were removed for more visual clarity. Finally, stairs and elevators were labeled more clearly, to make moving around the ballpark easier, especially for those with accessibility needs. These changes, on top of the creation of an SVG, were crucial in making the ballpark map clearer for all users.


Figure 50 The legend of the map, along with the floor controls.

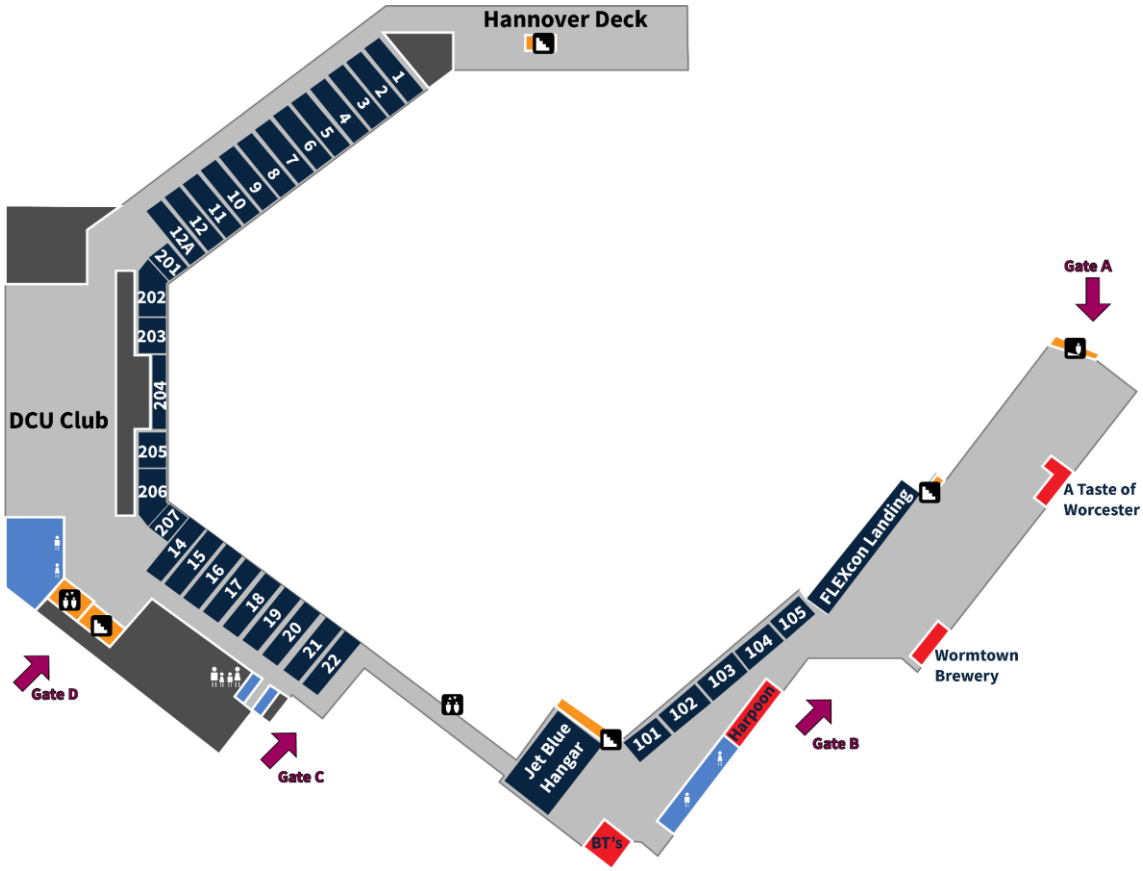



Figure 51 The map of the upper floor of Polar Park with the suites, 200 level seating, and the upper area of right field.


Figure 52 The map of the lower floor of Polar Park featuring the store and the path to the main floor.

# Appendix B: Apps used as Inspiration for the Rewards Pages

A large goal of the Polar Park App was to integrate the WooSox rewards experience with the rest of the actions that users want to do regarding the WooSox. However, it was quickly realized that there are other applications that achieve this goal, so they were looked at for ideas for the Polar Park App.

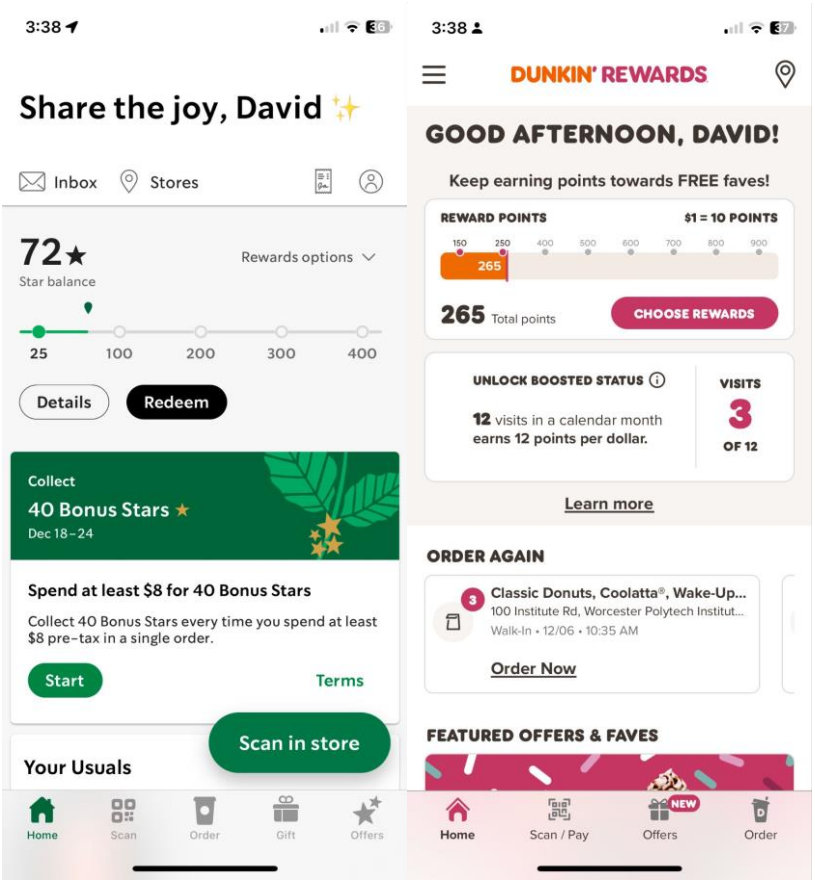



Figure 53 (left) The homepage of the official Starbucks mobile app.

Figure 54 (right) The homepage of the official Dunkin' mobile app.

The WooSox rewards program is similar in nature to the rewards programs of two notable coffee stores: Starbucks and Dunkin'. Both of their apps feature the number of points the user has, some ways to earn more points, and some ways to spend points. Since the ticketing part of the app needed to be separate from the rewards part, the only way earning points was able to be replicated was by having the QR code scan feature. The number of points sections was replicated, and in early versions of the app a progress bar similar to these apps also existed; however, it was removed in later versions because unlike these apps, there are not "tiers" of rewards in WooSox rewards similar to these systems. Lastly, there is a section to show things to spend points on, which was replicated in the polar park app in a similar way to the Dunkin' app.

# Appendix C: Rewards Flow UI Mockups

Below displays the original UI mockup choices for the application. The screenshots below from top to bottom follow the natural flow a user would follow when they enter the rewards section of the Polar Park application. Figure 55 displays a login screen for users. It is a single sign on screen. If users don’t have an account, then they have the opportunity to sign up for the app, as shown in Figure 56. Figure 58 to Figure 63, display all the screens involved with the rewards functionality of the application. It includes a comprehensive dashboard, along wit a hamburger menu that opens up a navigation bar that displays options provided to us by the WooSox organization. Thus, addition was to meet business needs. A plethora of changes were made to some of these screens to improve user experience, along with some additional screens being implemented where the team saw fit, to yet again improve experience, and end with a complete application.

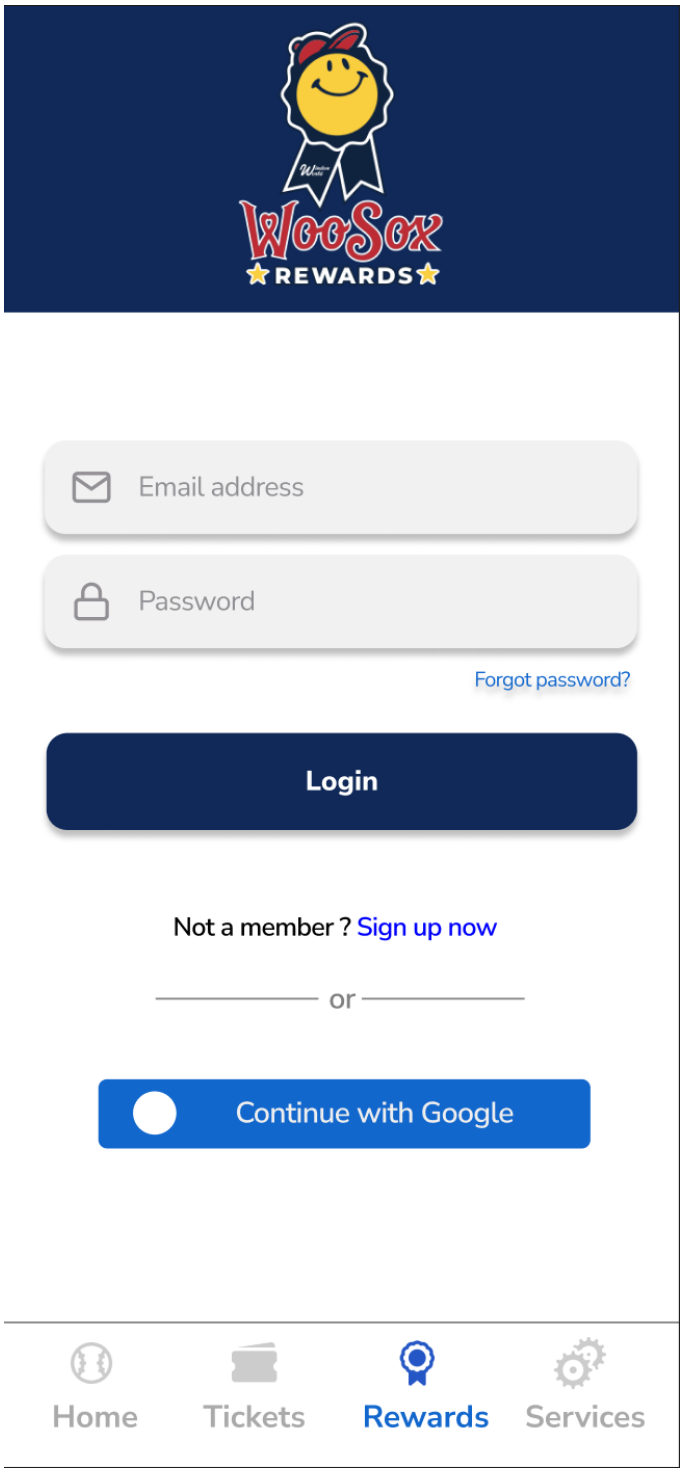



Figure 55 The mockup rewards login screen for existing users.

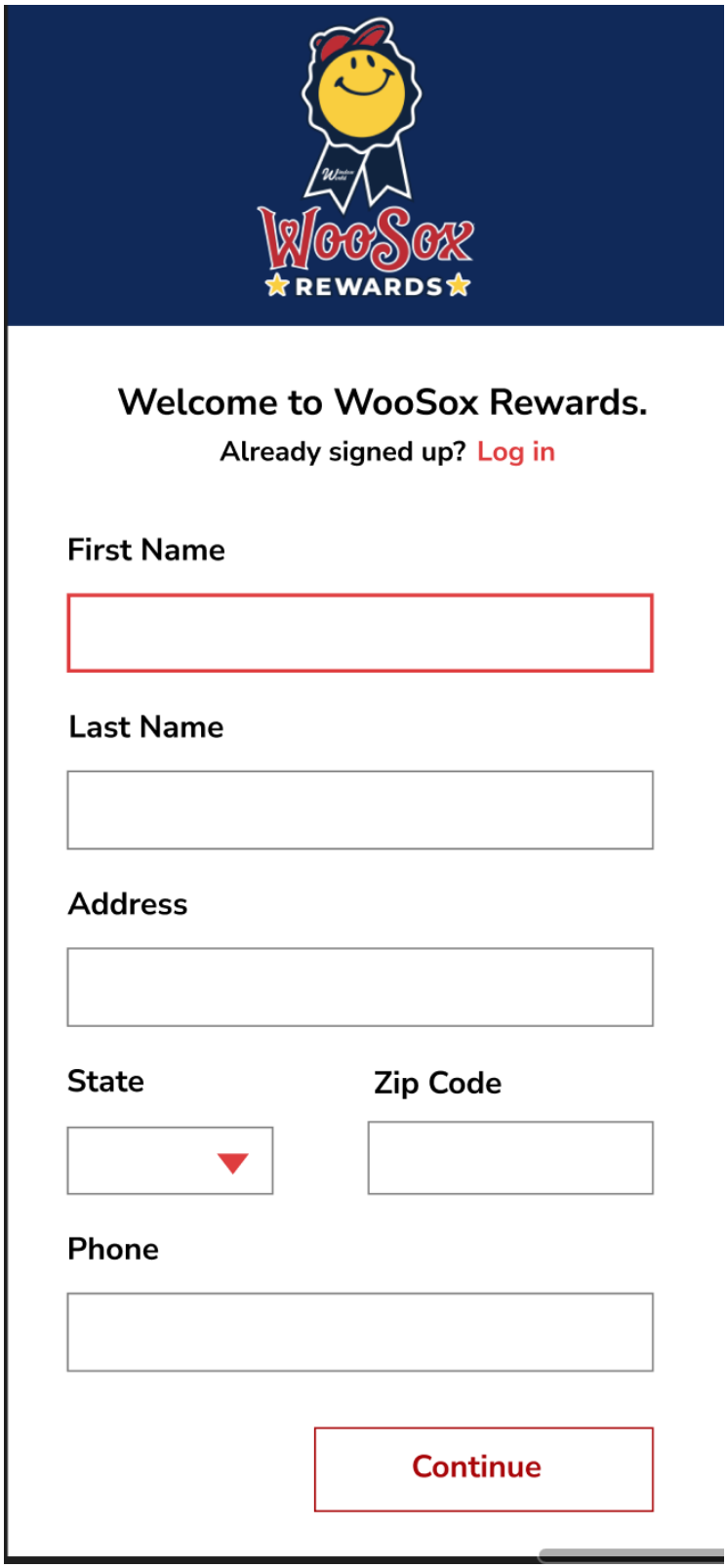



Figure 56 Mockup rewards sign up screen for new users.

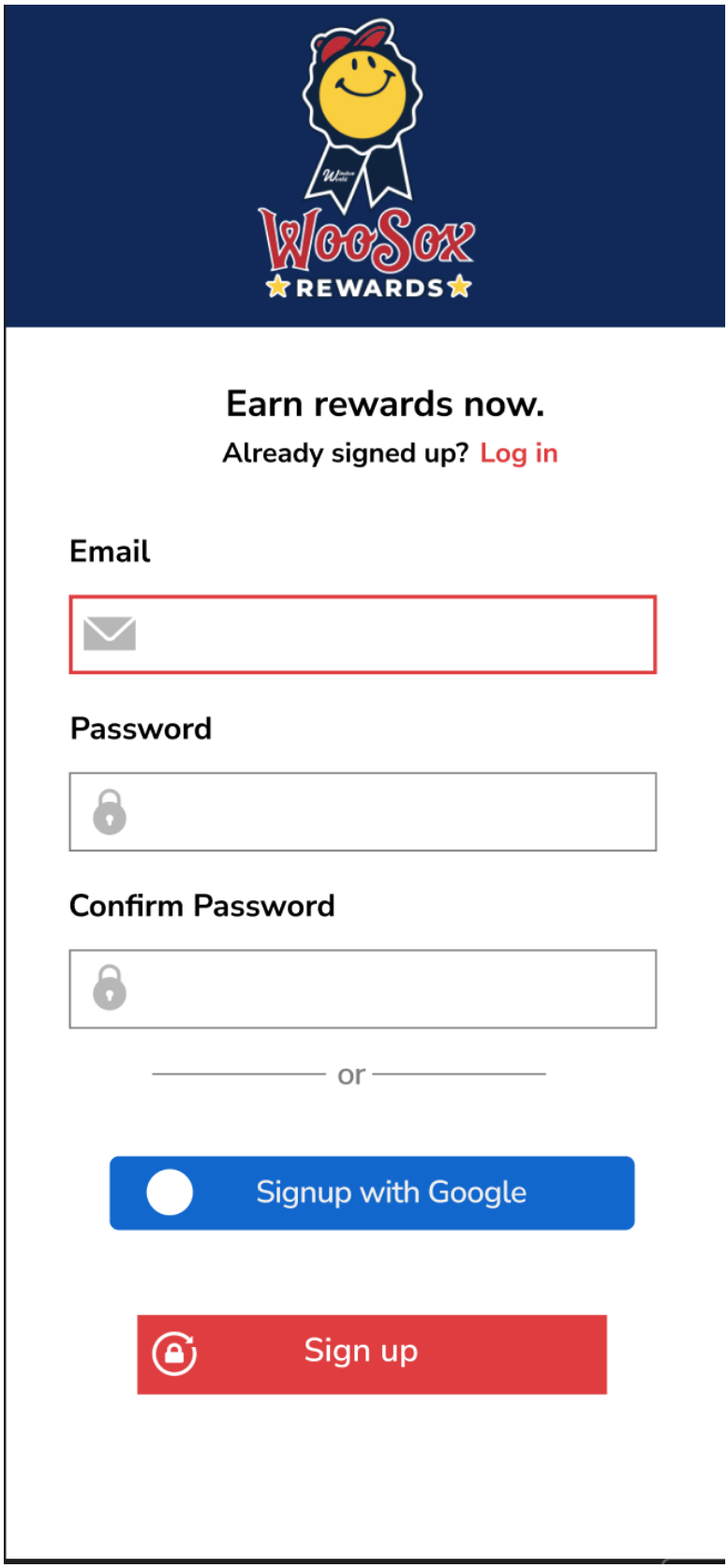



Figure 57 Mockup rewards sign up screen for new users (second screen).

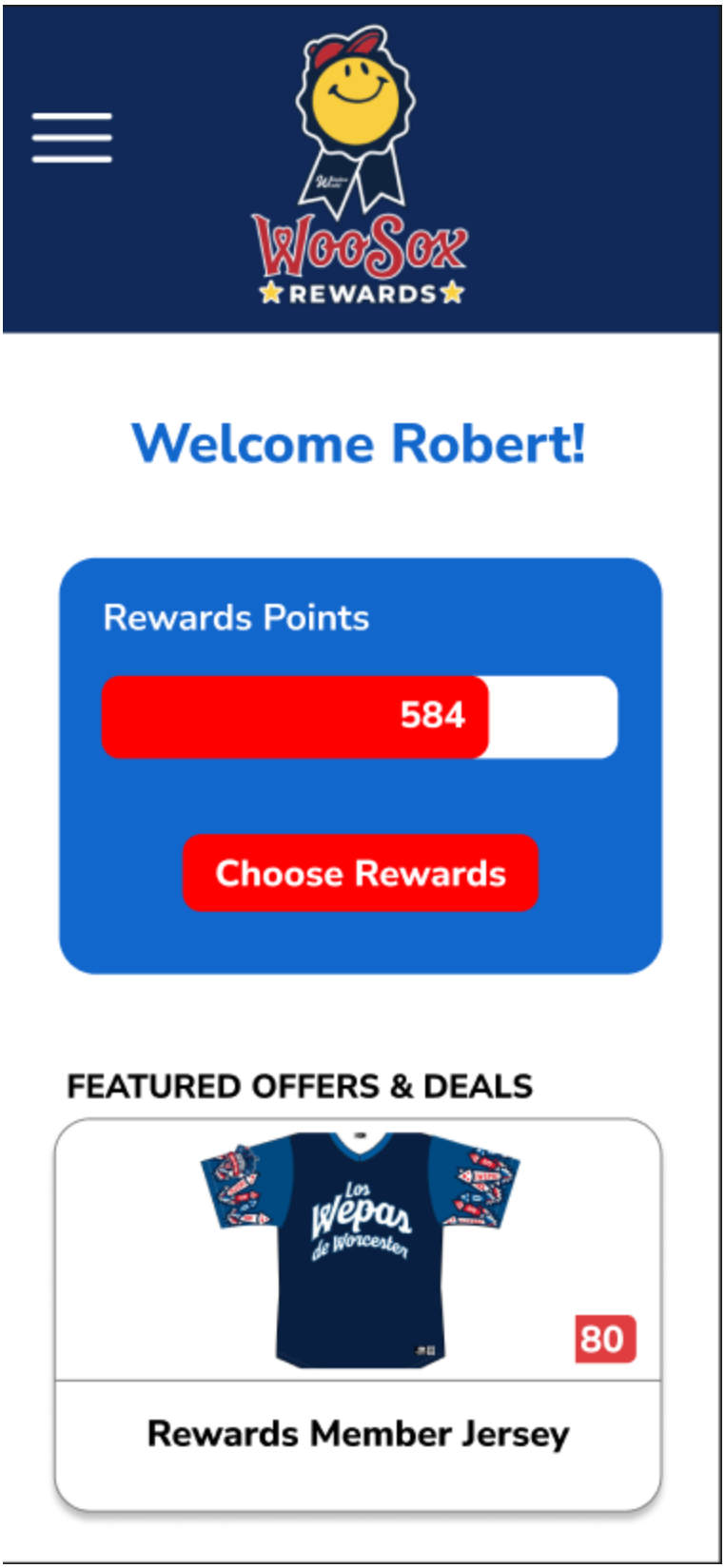



Figure 58 Home dashboard of the rewards part of the application. Displays the users' current points, and the features offers and deals, along with a hamburger menu that allows users to access different aspects of rewards.

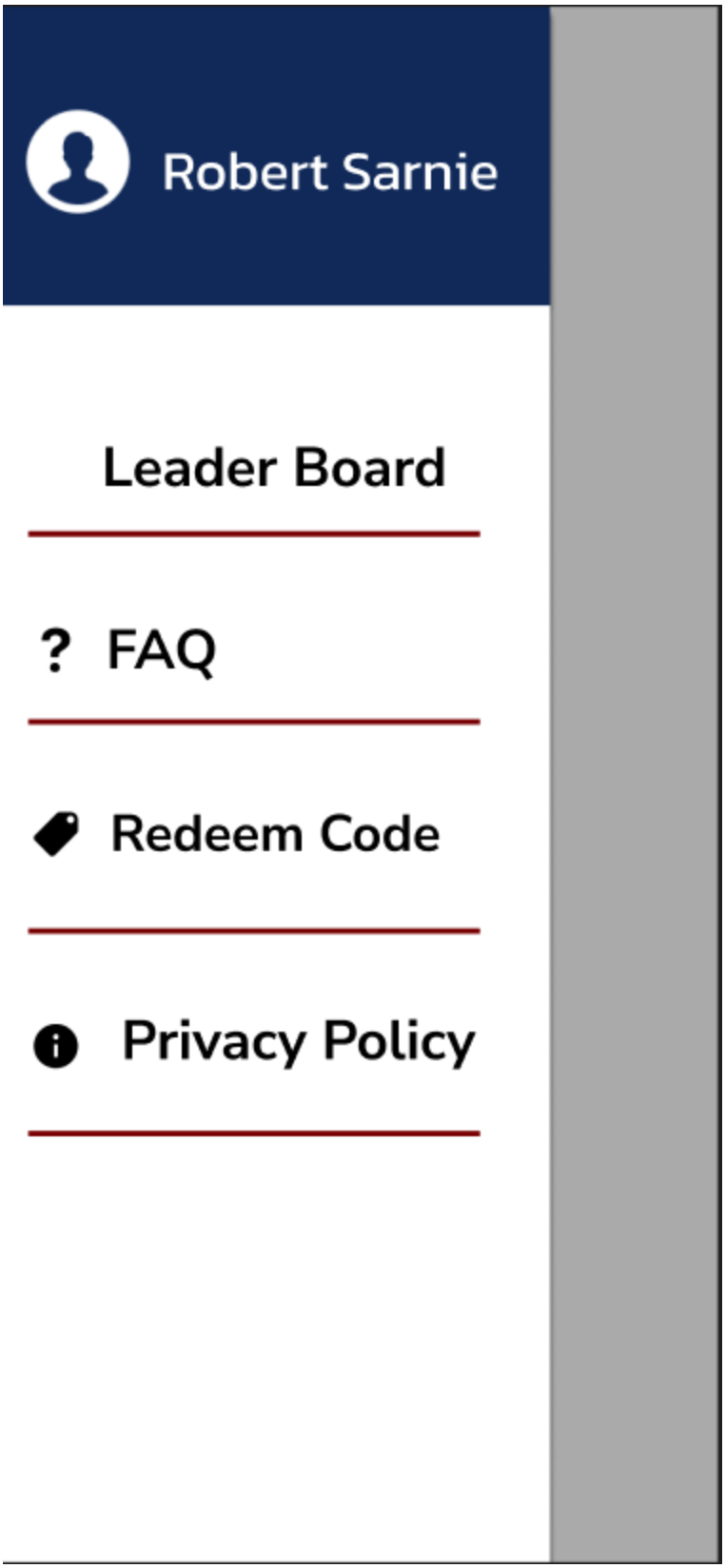



Figure 59 Navigation bar, which opens when user clicks on the hamburger menu.

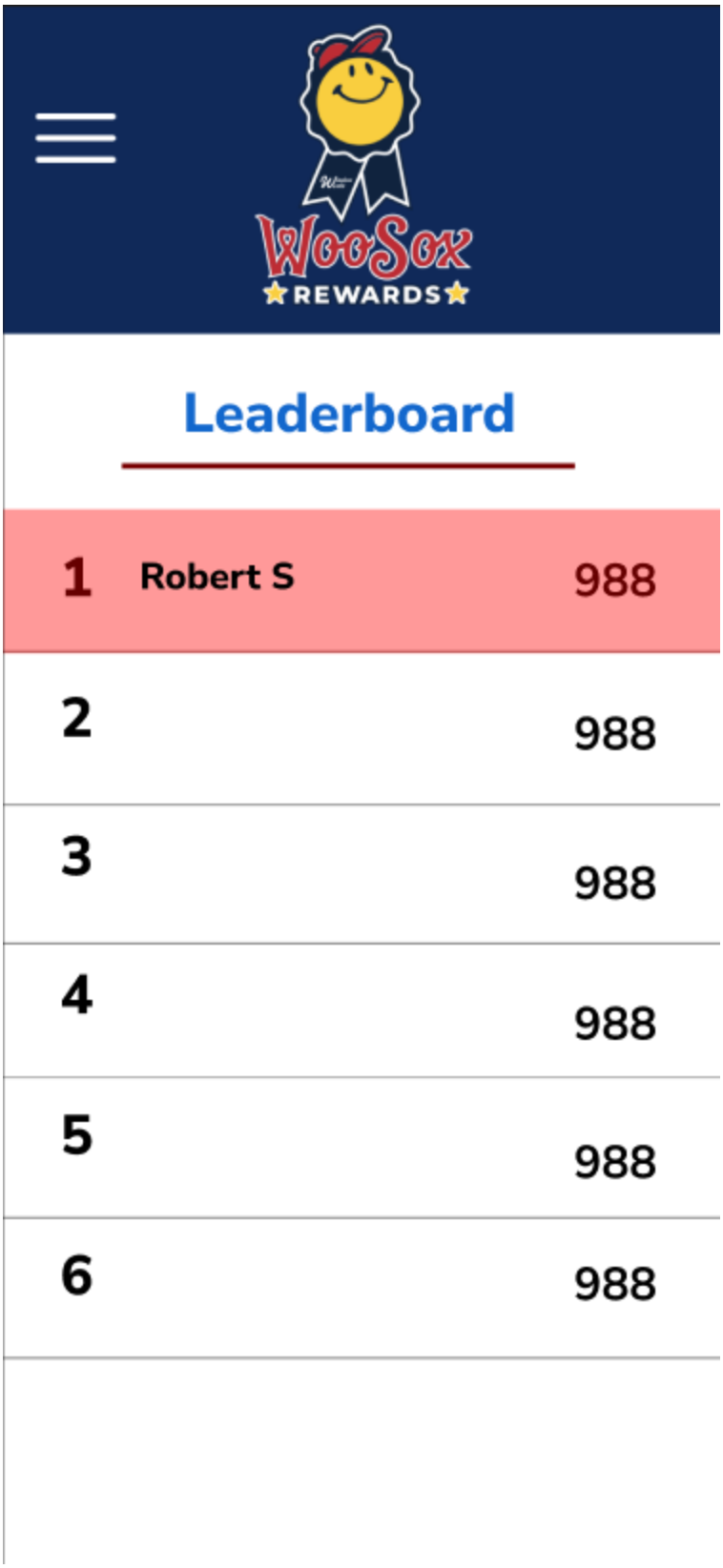



Figure 60 Leader section of application. Allows users to view where they stand in terms of points in comparison to another user.


Figure 61 FAQ section of hamburger menu. Provides answers to common questions users may have about the WooSox rewards program.

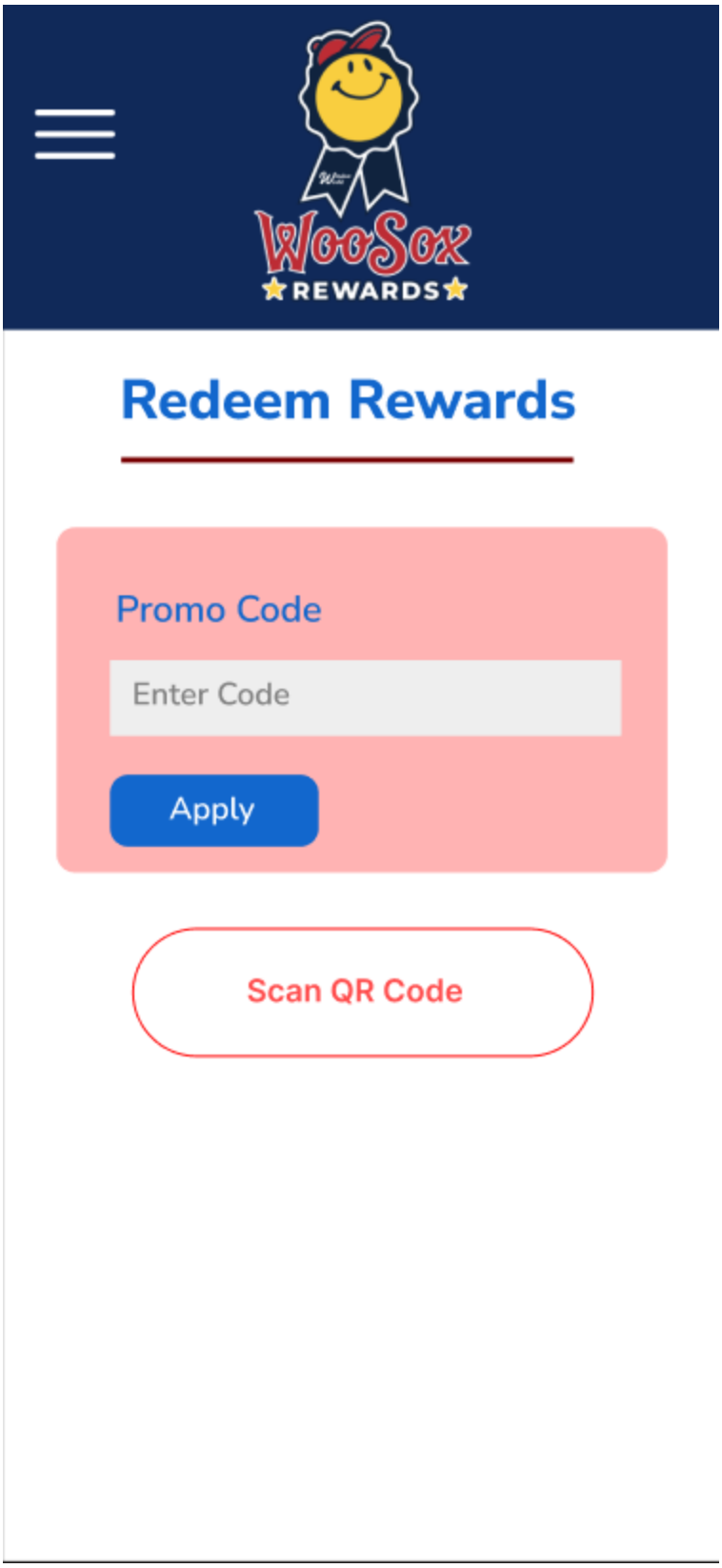



Figure 62 Code redemption screen. Allows users to redeem promo codes to gain more points. These codes are on receipts or earned in various ways.

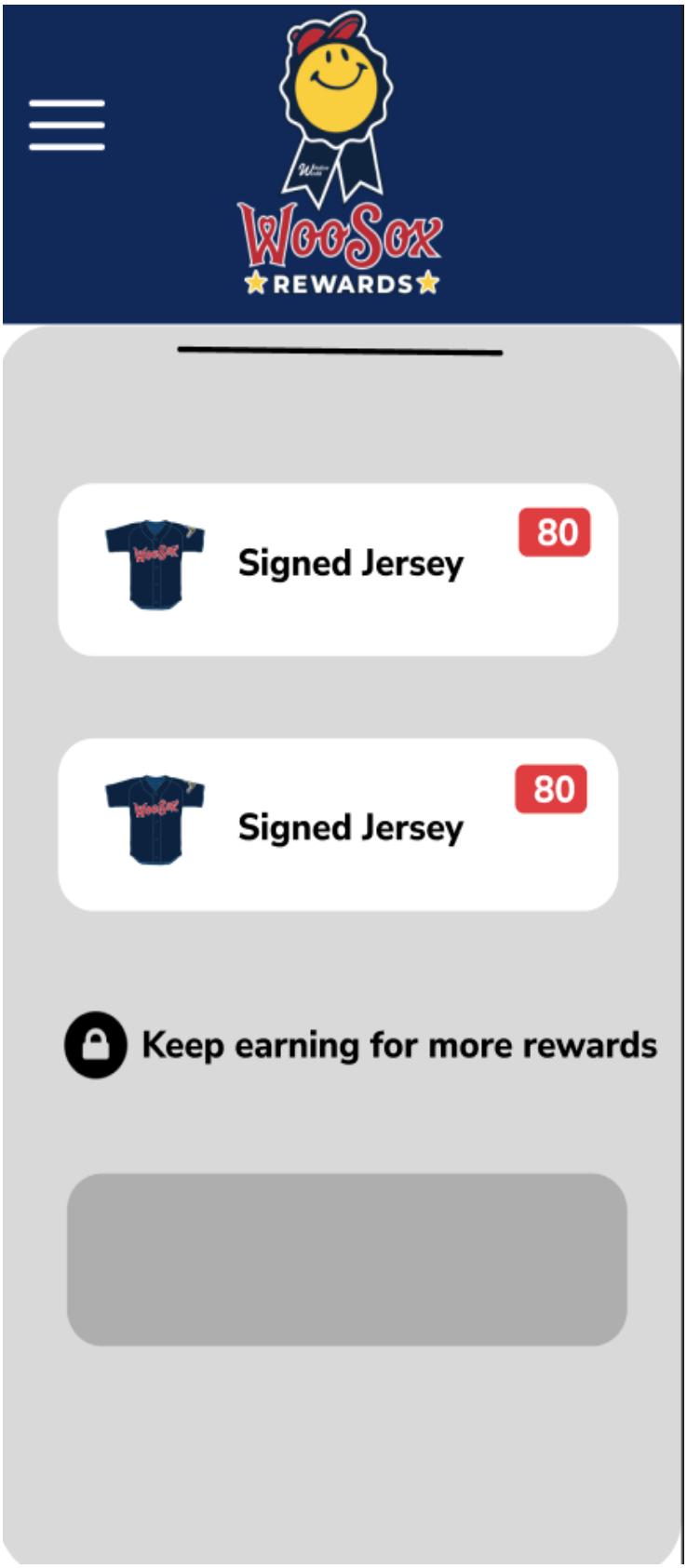



Figure 63 Screen allowing users to view their current rewards, along with being able to redeem any rewards that they have available to them.

# Appendix D: Process Flow Chart

Beneath is attached the Process Flow Chart which has been split in Section 6. This is what it looks like all together, however much of it is too small to see in this form.

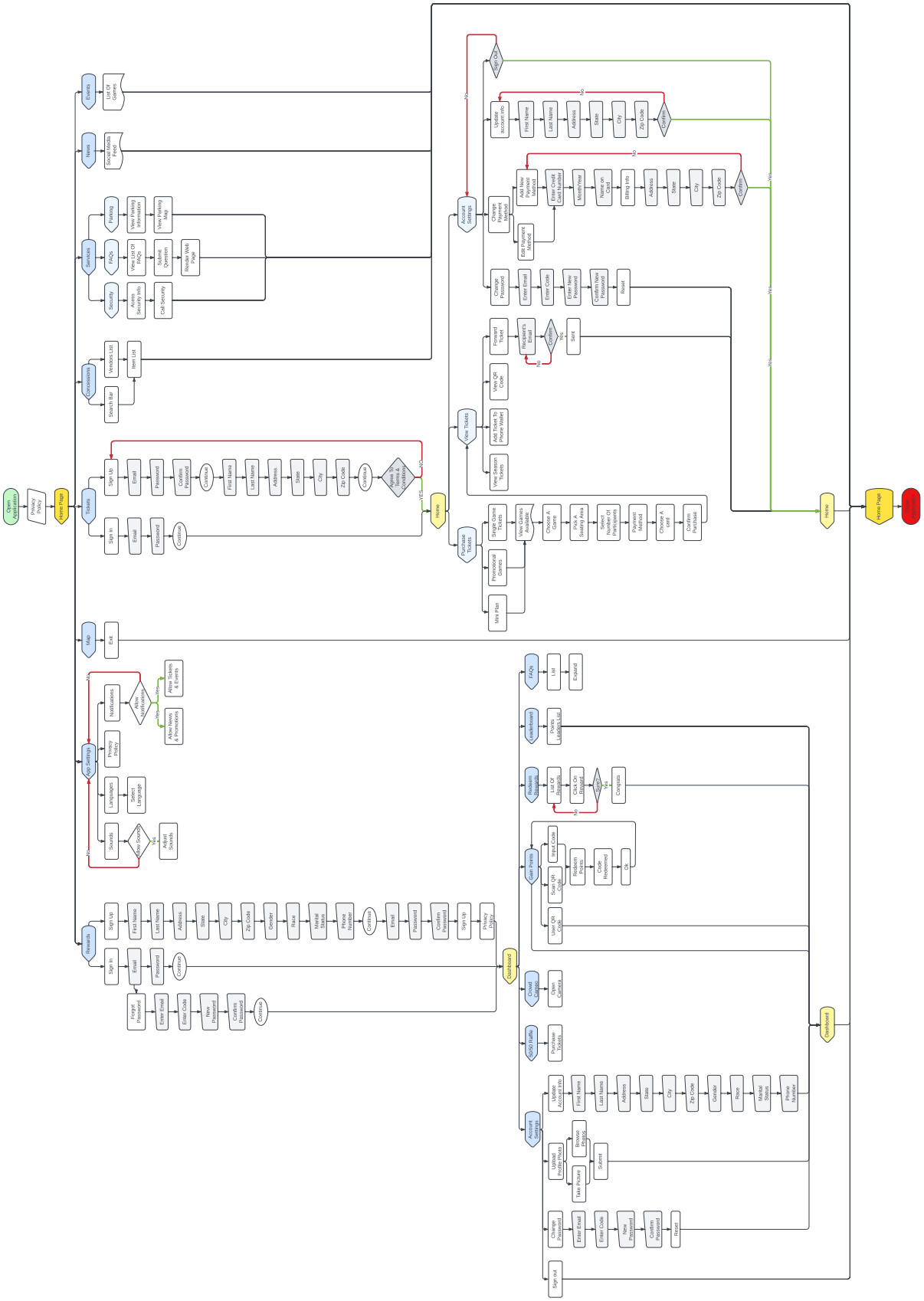


Figure 64 Process Flow Chart