Sports Facility

Project Report

V1.1

10-7-2017

Anthony Saleh

Colin Neil

Edward Marks

Graydon Peck

Matthew Degraeve

Peter Kowalski

Table of Contents:

Project vision

Backgrounds

Socio-economical Impact, Business Objectives, and

Gap Analysis

Security and ethical concerns

Glossary of Key Terms

Project Execution and Planning

Team Information

Tools and Technology

Project Plan

Best standards and Practices

System Requirement Analysis

Function Requirements

Non-functional Requirements

On-Screen Appearance of landing and other pages requirements.

Wireframe designs

Functional Requirements Specification

Stakeholders

Actors and Goals

User stories, scenarios and Use Cases

System Sequence / Activity Diagrams

User Interface Specifications

Preliminary Design

User Effort Estimation

Static Design

Class Model

System Operation Contracts

Mathematical Model

Entity Relation

Dynamic Design

Sequence Diagrams.

Interface Specification

State Diagrams

System Architecture and System Design

Subsystems / Component / Design Pattern Identification

Mapping Subsystems to Hardware (Deployment Diagram)

Persistent Data Storage

Network Protocol

Global Control Flow

Hardware Requirement

Algorithms and Data Structures

Algorithms

Data Structures

User Interface Design and Implementation

User Interface Design

User Interface Implementation

Testing

Unit Test Architecture and Strategy/Framework

Unit test definition, test data selection

System Test Specification

Test Reports per Spring

Project Management

Project Plan

Risk management

References

Project Vision

Background

As a customer at an ice rink it seems like a very simple business where people go on a Friday night to skate around for a couple of hours and maybe get a snack while you are there. When analyzed more deeply there is many things going on from day to day. Ice facilities make their money in a number of ways such as concessions, ice rink rentals, and a pro shop where hockey players can buy new gear and get their current gear serviced. SportsPlex is system that will manage all aspects of such a business from a customer logging in to find a time to reserve an ice rink for a game with friends, to a manager assigning their employees a timesheet and workstation.

Business Objectives

* Reduce business cost by allowing system to handle low level tasks
* Increase customer satisfaction by making business offerings more accessible
* Increase employee effectiveness with system’s tools
* Save management’s time with system’s management tools

Security and Ethical Concerns

Since SportsPlex has a number of different components there is a lot of points where we need to store important data and keep is secure. Since we have a Pro Shop marketplace we will be storing a user’s payment information (Credit Card, Address). This data is of the utmost importance to keep private. Any breach or loss of this data would lose the company a large amount of money from loss in customer base and investigations. Allowing a system control business aspects such as the ice rink rental means that the system must always store data accurately. If the data is incorrectly stored such as saving a customer’s reservation as a different time as selected can cause issues such as having an ice empty ice empty because it was unable to be reserved or angry customers over the reservation issue. Both of those examples lose the company money and make such a system much riskier.

Glossary of Terms

* Attribute - Metadata that describes the elements associated with a particular entity.
* CSS - Cascading Style Sheets. The code that will be implemented on top of the HTML code that will style and further format website appearance.
* Field - Refers to the column name within the mySQL table that labels the data being stored.
* HTML - Hypertext Markup Language. The code that will be used for the backbone website structure.
* Metadata - Data/Labels that describes other data.
* PHP - PHP: Hypertext Processing. The background code that will be used to add functionality.
* Primary Key - A key that is used to uniquely identifies a value within the mySQL database.
* Record - A row or entry into the mySQL table that contains a user’s login credentials and stored information.
* SQL - Structured Query Language. A database software that is used to add, retrieve or modify stored data.
* Wireframe - The mock up image designs that provide a visual on how each website page will appear.

System Requirement Analysis

Functional and Non-Functional Requirements

Requirements:

Req1: New users can register

Req2: A Manager can create an employee account

Req3: A Manager can assign employees to stations

Req4: An Employee can manage service queue

Req5: A customer can view ice rink reservation times and dates

Req6: A customer can reserve a spot in the facility

Req7: An employee can view their schedule

Req8: Returning users obtain secure logins

Req9: Customers are shown upon logging in what interests them (recently view items or purchases).

Req10: Returning users can see recently viewed rink times (must sign in).

Req11: Admins monitor new user entries.

Functional Requirements

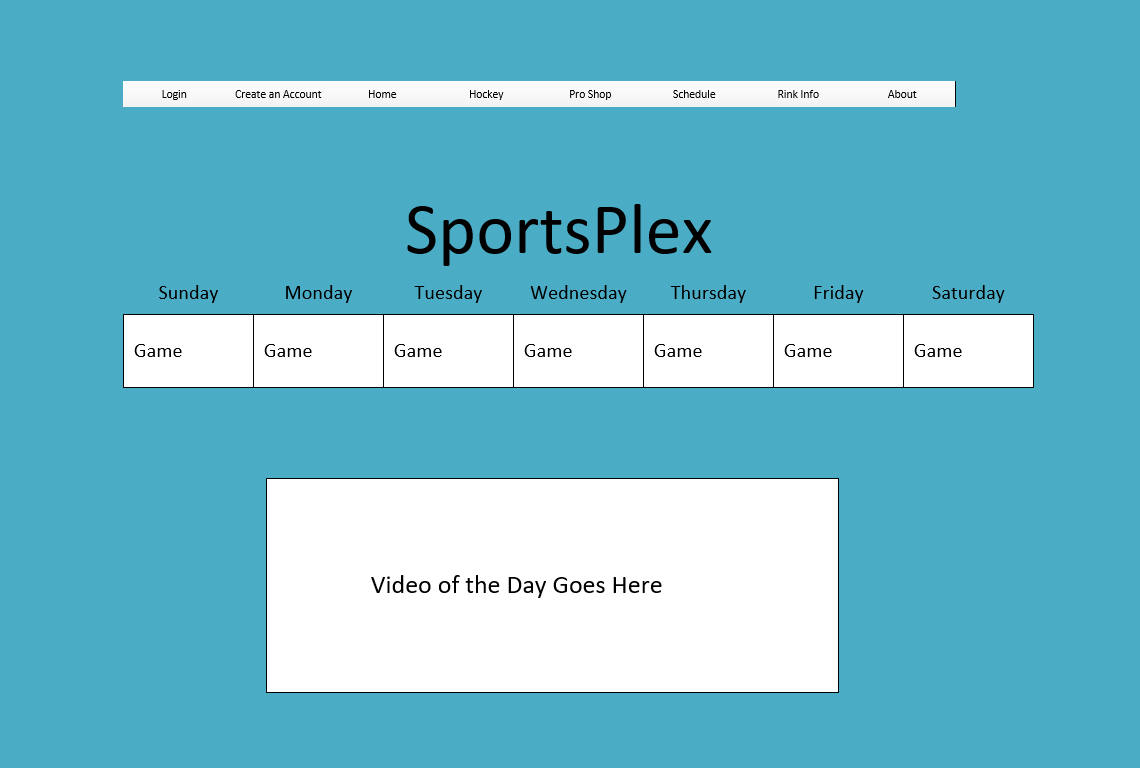
1. Web Registration
   1. Description: This shall allow the customers and employees to create a personal account with a registration page.
      1. The customers and employees will provide a web page that directs the customers and employees to a registration page.
      2. The system provides text forms allowing the customers and employees to enter and submit their account information. (username and password)
      3. The system will compare customers and employees credentials with the ones that are already created in the secure database.
         1. The system checks for valid credentials
      4. The system will alert if the customer or employee already exist.
2. Login
   1. Description: After the customer or employee are registered, they are able to login to their account
      1. The system shall compare past logins with customers and employee
         1. The system will make sure they have the correct credentials when logged in.
         2. The website has three authorization levels, Customer, Employee, Admin
         3. Upon unauthorized logins detected, the system will notify you through email that there is a potential security risk to both users.
         4. In case someone forgets their password, there will be an option to create a new one via email.

Non-Functional Requirements

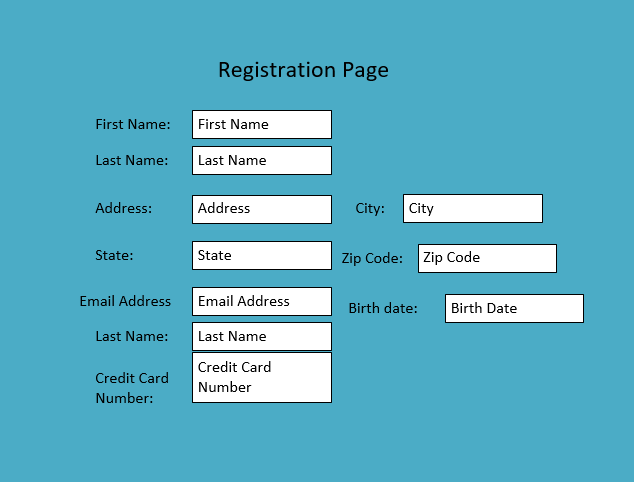
1. Web Registration
   1. Description: This will let customers and employees create a personal account from a registration page.
      1. The system uses use a secure MySQL database to store customer and employee account registration data.
      2. The system will save the customer and employee registration data in the secure database with no lost data.
      3. The system shall hash user passwords so they are stored secretively
2. Login Page
   1. Description: The customers, once registered, will be able to log in to the site to see rink availably
      1. The system will allow previously-registered customers and employees to log in to the site with an already registered login.
   2. The employees, once registered, will be able to log in to the site to see rink availably and be able to manage multiple services.
      1. The system will allow previously-registered customers and employees to log in to the site with an already registered login.
3. Rink Reservation
   1. Description: the website will provide a calendar where a rink times can be reserved.
   2. The website will provide a search bar in order to find a later date.
   3. The website shall be available for 99.98% of the time for customers, employees and admins.

Wireframe Designs

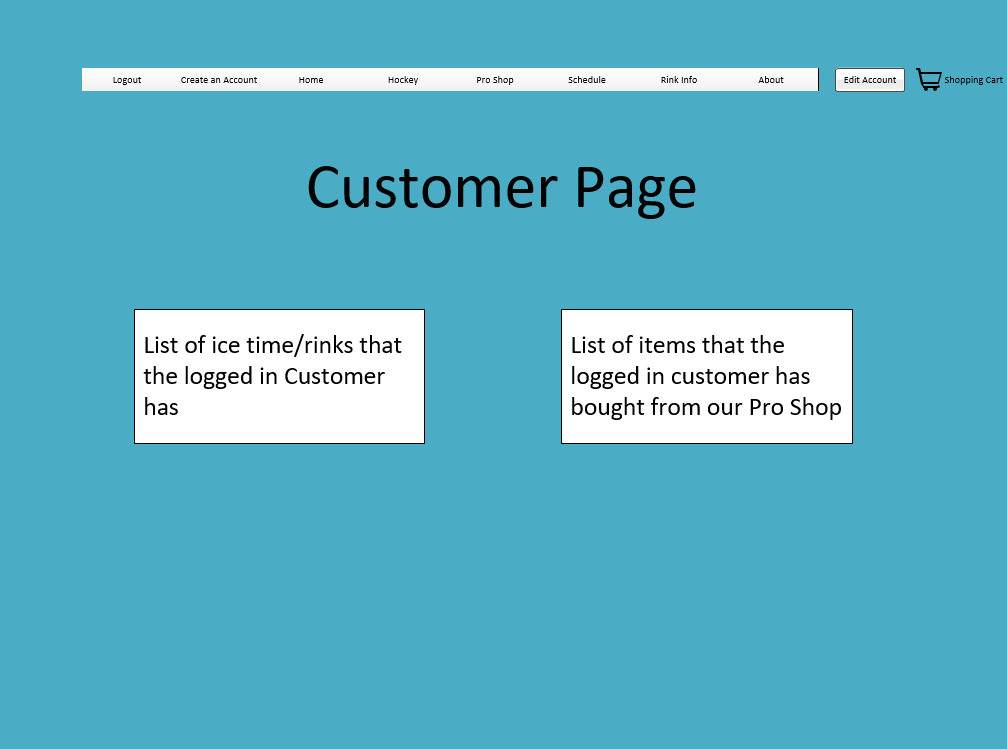
1. Home Page



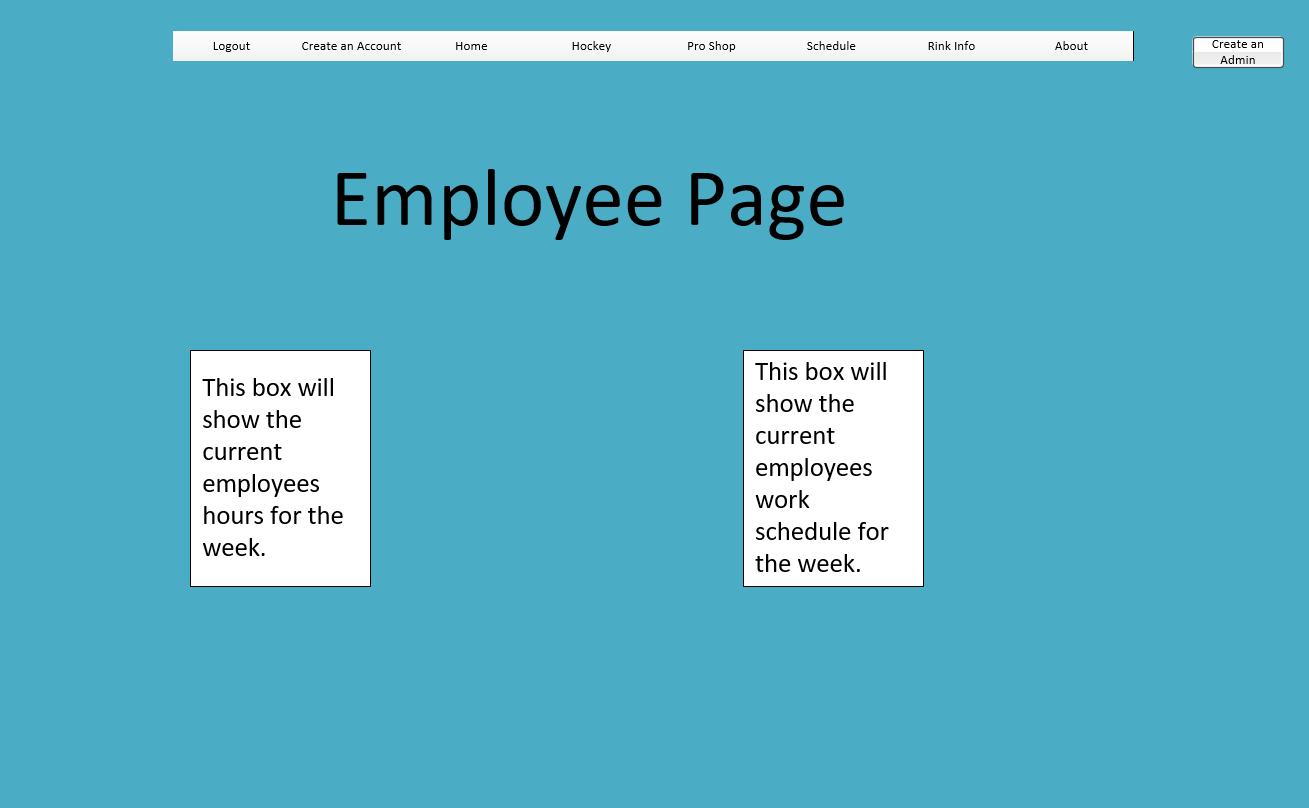
1. Registration Page



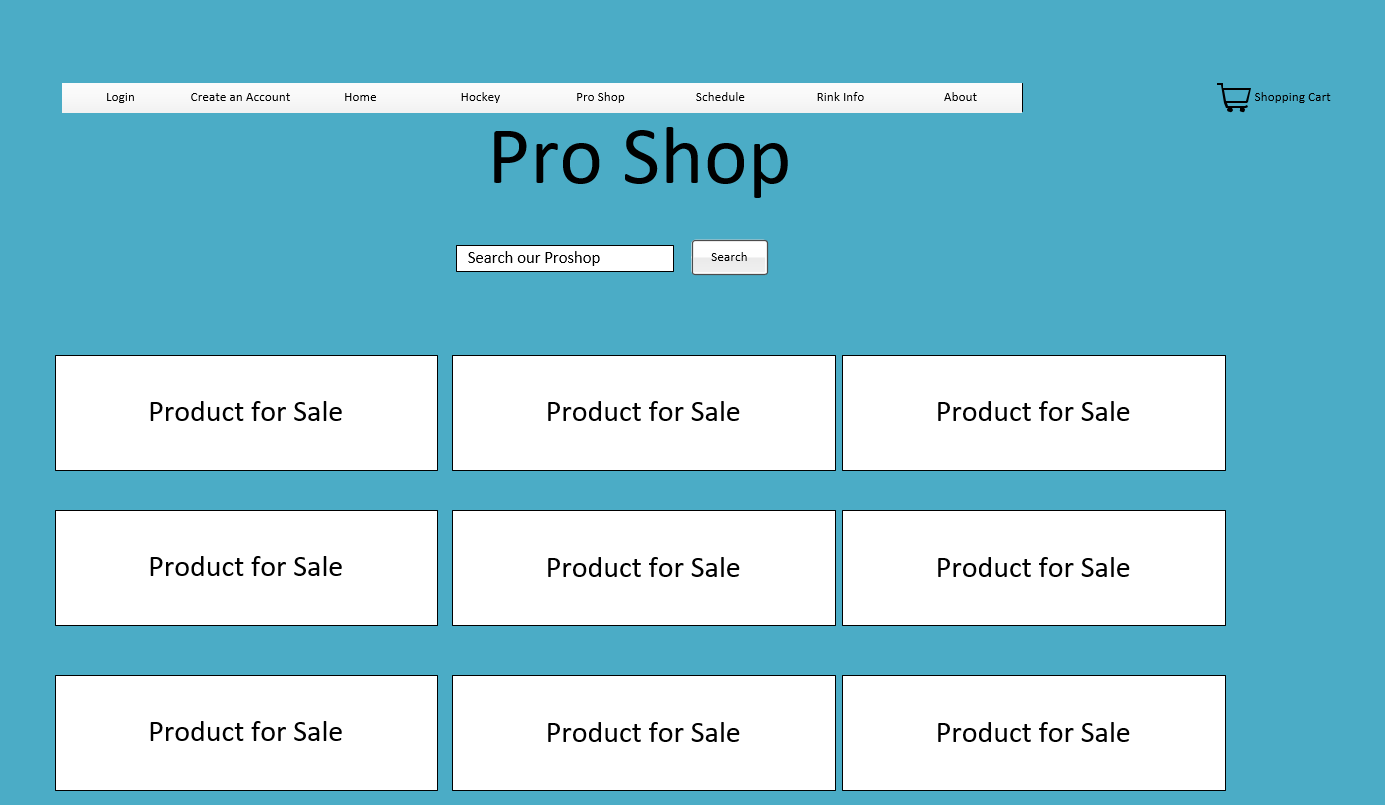
1. Customer Page



1. Employee Page



1. Pro Shop



1. Reservation Page



Functional Requirements Specification

* **Use Case UC-1:** Generate Account Information
  + **Related Requirements:** Req1,Req2
  + **Summary:**
    - This use case is designed for users who don’t already have an account. When a user first visits our website he/she will click on “Get Started”. At this point, a user will enter valid information in the steps for creating an account. Once all the required fields contain data, the user will click “Create”. As soon as the data is retrieved by the database, the user is signed into their new account.
  + **Initiating Actor(s):**
    - Customer
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
  + **Actor’s Goal:**
    - Register for SportPlex website and capability to review account information.
  + **Preconditions:**
    - A customer doesn’t have an account affiliated with them.
  + **Postconditions:**
    - Customers gain access to SportPlex’s home page.
    - Added features are now available to view.
  + **Usernames:**
    - Must at least have 1 Number.
    - Must at least have 1 Capital Letter.
    - Must be greater than 5 characters.
  + **Password:**
    - Must at least contain 1 number.
    - Must at least contain 1 Capital Letter.
    - Must be greater than 8 characters.
  + **Email Address:**
    - Must include an @ symbol.
  + **Flow of events for Main Success Scenario:**
    - Customer Presses the “Registration Button” (Trigger Action)
    - System prompts the user for a username.
    - Customer Provides a valid username.
    - System prompts the user for a password.
    - Customer Provides a valid password.
    - System prompts the user for an optional email.
    - Customer supplies a valid email address.
    - System prompts the user for Date of Birth (DOB).
    - Customer Provides correct DOB.
    - System prompts the user with these required fields: Address, Zip, Phone Number, City, State, and SSN.
    - Customer Provides correct information.
    - System checks to see if all required fields are filled in. (\* is a required field).
    - System checks for unique usernames and passwords to see if they already have been used.
    - Saves users credentials to the database if valid.
    - Customers press the “Submit” Button to confirm.
    - System notifies user via email that their account was successfully created.
  + **Alternate flows:**
    - System check discovers fixable deficiency: The user discovers a problem with connecting to Wi-Fi, but can fix it by powering the modem/ router down.
    - Button malfunction: The register button doesn’t operate as it’s supposed to, admins can fix it in a timely manner. (allowing users to continue registration).
    - Failed to register: The credentials entered failed. The user must make sure all the required fields are filled out. ( \* indicates a required field).
  + **Exceptions:**
    - No Internet connection.
    - Error with connecting to database.
    - Account already exists.
* **Use Case UC – 2: Login into Account**
  + **Related Requirements:** UC1, Req3, Req4, Req5, Req6, Req7, Req8, Req9, Req10
  + **Summary:**
    - This use case is designed for users who already made an account. In this case, A user will click on “Sign in” to access their account. When users access their account, they will gain different permissions than someone who hasn’t made an account. The user will gain access to buying items and setting rink date/times and possibly more.
  + **Initiating Actors(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
  + **Actor’s Goals:**
    - Login to SportPlex and begin using the website.
  + **Preconditions:**
    - All users are already registered for the website.
    - All users are not already signed into the website.
  + **Postconditions:**
    - All users are sent to SportPlex’s home page.
  + **Flow of Events for Main Success Scenario:**
    - Users navigate to the SportPlex website.
    - Customers click on “Customer Login” Button (Trigger Action).
    - Employee’s click on “Employee Login” Button (Trigger Action).
    - System prompts the Customer for login credentials.
    - Customer enters the login credentials and presses “Login”.
    - System prompts the Employee for login credentials.
    - Employee enters the login credentials and pressed “Login”.
  + **Alternate Flows:**
    - System hesitates to log in user: User trying to access their account with entering the wrong credentials. Entering the correct credentials will fix this problem.
  + **Exceptions:**
    - No Internet Connection.
    - Error Connecting to Database.
    - Error in Login Credentials.
* **Use Case UC – 3: Navigation Bar**
  + **Related Requirements:** UC1, UC2, Req4, Req5, Req6, Req7
  + **Summary:**
    - This use case is designed for making our website easier to navigate. At this point, a user will be able to make efficient use of time in navigating to each page they wish to view. This navigation bar will contain every page users can view even including: help, question and answers, and etc.
  + **Initiating Actor(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
    - HTML
    - CSS
    - Bootstrap
    - JavaScript
  + **Actors Goals:**
    - Users will be able to easily navigate through the website.
  + **Preconditions:**
    - Successfully registered for SportPlex
    - Logged into Account.
  + **Postconditions:**
    - Ability to set up reservations
    - Access for watching training videos (all different kinds).
  + **Flow of Events for Main Success Scenario:**
    - Appears at top of screen “ability to search and browse the page” (Trigger).
    - Users are presented with a search bar which searches words on the page.
    - System provides users with tabs allowing them to navigate the website.
    - System provides drop downs underneath each tab with different criteria.
    - System then navigates based on the users click.
    - Users at any time are able to log out of their session.
  + **Alternate Flow:**
    - Human error: if the user doesn’t know what page to visit, they will have the ability of navigating to the help page for solving their problem(s).
  + **Exceptions:** 
    - No Internet Connection.
    - Error Connecting to Database.
* **Use Case UC – 4: Home Page**
  + **Related Requirements:** UC3, Req1, Req4, Req5, Req6, Req7, Req8
  + **Summary:**
    - This use case is designed for giving users ability to see weekly events and important information. Each user will be able to view the home page without signing into their account. With that in mind, the functionality of what each user can do will be very limited. Users are giving the ability to a little preview of our site before creating an account.
  + **Initialing Actor(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
    - HTML
    - CSS
    - BOOTSTRAP
    - JavaScript
  + **Actor Goals:**
    - Giving users ability to view important/needed information.
  + **Preconditions:**
    - Users must be logged in.
  + **Postconditions:**
    - Reservations made by each user will be listed.
    - Map will be available to view.
    - Calendar with available rink times/dates will be viewable.
  + **Flow of Events for Main Success Scenario:**
    - Users click on reservations to view who booked each time.
    - Users click on Map to view additional functionality other than just destinations.
    - Users press “minimize” to exit map features.
    - System Stores the Important information in database for viewing purposes.
  + **Alternate Flow:**
    - Video freezes: In the case of are home video freezing there will be a button for user’s, allowing them to put video into background until it becomes available again. (Also, they can cancel the video if they want).
    - Failed to load: user is unable to connect to our home video at this time. Pop up window tells the user “try again in a couple of hours or so”.
  + **Exceptions:**
    - No Internet Connection.
    - Error Connecting to Database.
    - Error in viewing table of reservation time/dates.
    - Error with opening the map.
* **Use Case UC – 5: Video Page**
  + **Related Requirements:** UC2, UC9, UC10, Req3
  + **Summary:**
    - This use case is designed for giving customers or employees the ability to watch instructional videos. In order to view these instructional videos a customer or employee must log into their account. Once they’re logged in they will gain access to very helpful videos. Whether it’s giving them lessons on how to succeed at their job or how to sharpen skate’s the most efficient way. There will be a search bar allowing customer or employees to search a vast variety of helpful videos.
  + **Initiating Actor(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
    - HTML
    - CSS
    - BOOTSTRAP
    - JavaScript
  + **Actor Goals:**
    - Watch an Instructional video pertaining to the specified training required.
  + **Preconditions:**
    - The users will have to be logged in already and navigate to the video page.
  + **Postconditions:**
    - Customers or Employee’s will be able to watch instructional videos.
    - Video’s will teach new Employee’s about how to manage their position in the company.
  + **Flow of Events for Main Success Scenario:**
    - Customer or Employee presses “video” on home page navigating to different Instructional videos.
    - Customers or Employees search videos pertaining to position in the company.
    - Customers or Employees select their video.
    - System presents user with a stopped video starting from the begging.
    - Customer or Employee presses the play button on the video.
    - System begins playing the video on the Customer or Employee screen.
    - System stops the video once complete.
  + **Alternate Flows:**
    - Failed component: A user is unable to view a video. The user must make sure that he/she is still signed into their account. (Gives access to movies).
    - Weak connection: The videos consistently freeze because of the weak connection.
    - System error: The system had a crucial error, experiencing a crash. This happened when the employee accessed instructional videos. An error message appeared on the screen for a database manager to assist.
  + **Exceptions:**
    - No Internet Connection.
    - Error in Connecting to Database.
    - Video no longer exists.
    - Error in video playback
* **Use Case UC – 6: Schedule page**
  + **Related Requirements:** UC2, UC9, UC10, Req5, Req6
  + **Summary:**
    - This use case provides users with options to schedule time/date for each event appealing to their interests. At this point, a customer or employee must be signed in to add an event to their schedule. This page will require users to add a credit card to their account profiles. With doing this, it will allow each user the ability to book the set time/date of their choosing. After the date/time is booked in the schedule, each user will be able to view it in a view tab. For example, if a user accidently chooses the wrong time/date; they can click the “Cancel” button. (A dialog window will appear, asking “are you sure you want to cancel this transaction?”).
  + **Initiating Actor(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
    - HTML
    - CSS
    - BOOTSTRAP
    - JavaScript
  + **Actor Goals:**
    - Ability to reserve a rink time.
    - Instructions on specific details on how to set up a rink time.
  + **Preconditions:**
    - User must be logged in/registered in SportPlex.
    - Navigate to reservation page.
  + **Postconditions:**
    - Calendar will be available with ability to schedule a rink time/date.
  + **Credit Card Information**
    - Must contain a Balance
    - Cannot be expired
  + **Flow of Events for Main Success Scenario:**
    - Customer clicks “Reservation” tab, navigating to the correct page (Trigger Action).
    - System prompts user with Instructions of adding a reservation.
    - Customer clicks “Reserve Rink” button to set up a reservation.
    - Customer enters the Date/time of each reservation made.
    - Customer proceeds to entering their credit card information.
    - System Provides “View” button giving users ability to view their booked reservations.
    - System Saves users credentials.
    - System provides notification icon in the upper right corner.
    - Customer clicks on notifications for view their events coming up.
    - System provides “Cancel Reservation” button allowing Customers to cancel set reservations.
    - Customers clicks “Cancel Reservation” and his/her reservation is no longer on the schedule.
  + **Alternate Flows:**
    - Weak connection: Users are unable to successfully schedule a time/date that meets their criteria.
    - Failed component: The “add event” button is unavailable at this time and the user cannot create a scheduled time/date.
    - System error: The system lets the user know that the time/date of the event is booked. Once the user looks at the calendar they do not see the event they ended up booking.
  + **Exceptions:**
    - No Internet Connection.
    - Error Connecting to Database.
    - Reservation is already book.
* **Use Case UC – 7: Pro Shop Page**
  + **Related Requirements:** UC2, UC9, UC10, Req4, Req9, Req10, Req11
  + **Summary:**
    - This use case is designed for giving the users options for buying extra stuff as needed. When a user first visits this page, it will prompt them “please sign in to view items”. Once a user signs in, each item will be viewable for purchasing. Each user now gains the ability to add any items they want to their cart. Our database will store this information until: A user either purchases those items or deletes them from their cart. Our pro shop will function like a true store, with great customer service and even a review. (only take if you have ideas on how to grow our company).
  + **Initialing Actor(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
    - HTML
    - CSS
    - BOOTSTRAP
    - JavaScript
  + **Actors Goals:**
    - To allow Customers or Employees to purchase items (Hoodies, shirts, skates, etc..).
  + **Preconditions:**
    - Customer or Employees must be logged in/registered.
  + **Postconditions:**
    - Items will be able to be purchased.
  + **Credit Card:**
    - Must contain a balance.
    - Cannot be expired.
  + **Flow of Events for Main Success Scenario:**
    - Customers or Employees Click on Pro shop on home page, Navigates to Pro Shop.
    - Customer or Employee clicks “Add to Cart” in order to add item to a cart filled with items.
    - System Stores Customers or Employees Cart details in database.
    - Customer or Employee Navigates to cart once they want to checkout.
    - Customer or Employee clicks on “Checkout” therefore buying the items in their cart.
    - System sends Customers or Employees a verification email after they bought the items.
  + **Alternate Flows:**
    - System error: The system does not accept a user’s account information for adding a credit card.
  + **Exception:**
    - No Internet Connection
    - Error Connecting to the database.
    - Item exceeds limit on credit card.
* **Use Case UC – 8: About Page**
  + **Related Requirements:** UC1, UC9, UC10
  + **Summary:**
    - This use case is designed for giving users more information in case they are having trouble navigating our website or a generic question. Provide on this page will be: phone number, help tab (other problems people have come into, email address, and question an answer text area. Anything user may have a question on, visit this page to get helpful contact information.
  + **Initiating Actor(s):** 
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
    - HTML
    - CSS
    - BOOTSTRAP
    - JavaScript
  + **Actor Goals:**
    - Allow users to: get contact information in case of emergency, Our Mission, and How our company started.
  + **Preconditions:**
    - Customer or Employee must be logged in/registered for SportPlex.
  + **Postconditions:**
    - Customers or Employees will gain access to useful information in case of an emergency.
    - Customers or Employees will learn more about what our company stands for and other useful information.
  + **Flow of Events for Main Success Scenario:**
    - Customer or Employee clicks on “About” tab to navigate to the about page.
    - System stores contact information in database so a Customer or Employee can gain a pre – defined response when we’ll get back with you.
    - Customer or Employee enters a question into the text area to receive a response within 24 hours.
    - System stores this questions in case another customer or employee has the answers.
  + **Alternate Flows:**
    - System error: The system doesn’t make the question and answer session available for users to view. (users will have to contact assistance for this scenario).
  + **Exception:**
    - No Internet Connection.
    - Error Connecting to Database.
    - Text area is greyed out (no longer takes in information).
* **Use Case UC – 9: Authenticate user**
  + **Related Requirements:** UC1, UC2, UC6, UC7, UC10, Req4, Req6, Req9, Req10, Req11
  + **Summary:**
    - This use case is designed to make sure each user signing in is authenticated. When the user signs into their account, it will do a verification check to see if this user is authenticated for a secure count. The credentials of the set user account will be compared against those whom are already stored in the database.
  + **Initiating Actor(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
  + **Actors Goals:**
    - Allowing ability for checking credentials of logged in users against those within the secured database.
  + **Preconditions:**
    - Customer or Employee must enter login credentials and click login.
  + **Postconditions:**
    - The user’s credentials will be compared against those stored within the secure database and the user whose login should have the same credentials.
  + **Flow of Events for Main Success Scenario:**
    - Customer or Employee enters the login credentials and presses “Login” (Trigger Action).
    - PHP System Checker will make sure the login and password both match what’s already in the database.
    - The PHP System Checker will let the Customer or Employee know if they are authenticated or not.
  + **Alternate Flows:**
    - System error: The system does not recognize the user logging in as an authenticated user, therefore the user is unable to log into our website.
  + **Exceptions:**
    - No Internet Connection.
    - Error Connecting to Database.
    - User is not Authenticated.
    - PHP System Checker is down.
* **Use Case UC – 10: Logout**
  + **Related Requirements:** UC1, UC1
  + **Summary:**
    - This use case is designed for allowing a user to sign out of their account on the website. As the user clicks on the “sign out glyphicon” it will ask the user “if they wish to sign out”. Now the user is successfully signed out of their account.
  + **Initiating Actor(s):**
    - Customer
    - Employee
    - Administrator
  + **Supporting Software(s):**
    - phpMyAdmin
    - PHP
  + **Actor Goals:**
    - Allowing the current Customer or Employee the ability to logout from the website.
  + **Preconditions:**
    - Customer or Employee must already be logged into the system.
  + **Postconditions:**
    - Customer or Employee is logged out of their respective accounts, and is returned to the login page.
  + **Flow of Events for Main Success Scenario:**
    - Customer/Employee clicks “Logout” button.
    - System prompts Customer/Employee “Do you wish to log out?”
    - Customer/Employee clicks “Yes”.
    - System logs Customer/Employee out
    - System Sends Customer/Employee to Registration page.
  + **Alternate Flows:**
    - System error**:** System does not show a user is signed in, therefore the sign out button doesn’t function.
    - Failed component: The user becomes locked into the system and it therefore unable to log out of their account.
  + **Exceptions:**
    - Error in closing the connection.
* **Use Case UC-11:** A Manager can create an employee account
  + **Related Requirements:** REQ1, REQ2
  + **Initiating Actor(s):**
    - Manager
    - Employee
  + Supporting Software(s):
    - MySQL
    - PHP
  + **Actor’s Goal:**
    - To register for an Employee the person doing that has to have admin rights as a Manager
  + **Preconditions:**
    - The employee is not already registered
    - The manager has admin rights
  + **Postconditions:**
    - The employee is sent to the employee page once verified
  + **Flow of Events for Main Success Scenario:**
* Manager press button to add employee **(Trigger Action)**
* System prompts the manager for a username of employee
* Actor Supplies a valid username
* System prompts the user for a password
* Actor supplies a valid password
* System checks for unique username and password to see if they are already taken/valid
* Saves user credentials to the database if they are valid
* Manager presses the “Submit” Button to Confirm
* System sends a confirmation message that account was created
* **Exceptions**
  + No Internet Connection
  + Error Connecting to the database
  + Account Already Exist
* **Use Case-12 A manager can assign employees to work stations**
  + **Related Requirements:** REQ1, REQ2, REQ3
  + **Initiating Actor(s):**
    - Manager
    - Employee
  + **Supporting Software(s):** 
    - MySQL
    - PHP
  + **Actor’s Goal:**
    - Add the employee to a given work station the time they are working
  + **Preconditions:**
    - The employee must have an account
  + **Postconditions:**
    - The employee is assigned to that workstation.
  + **Flow of Events for Main Success Scenario:**
* Manager clicks on button to assign an employee to a workstation **(Trigger Action)**
* Manager selects an empty work station
* System checks to make sure that the workstation is empty
* Manager adds an employee to that work station that is not already assigned to one
* System checks to make sure that employee is not assigned to a work station
* System will add that employee to the workstation
* **Exceptions**
  + No Internet Connection
  + Error Connecting to the database
  + Work station already assigned
  + Employee is assigned to a different workstation
* **Use Case-13 Employees and Managers can add or remove customers to service queue**
  + **Related Requirements:** REQ1, REQ2, REQ3, REQ4
  + **Initiating Actor(s):**
    - Manager
    - Employee
  + **Supporting Software(s):** 
    - MySQL
    - PHP
  + **Actor’s Goal:**
    - Employees and managers can add or remove things on the service queue
  + **Preconditions:**
    - The employee/manager must have an account
  + **Postconditions:**
    - Customers will be added or removed off of the service queue
  + **Flow of Events for Main Success Scenario:**
* Employee or Manager will add customer into service queue **(Trigger Action)**
* The system will make sure all of the customer information is filled out correctly
* The system will make sure that the customer is not already added
* Once the customer is done the employee or manager will remove the customer from the queue.
* **Exceptions**
  + No Internet Connection
  + Error Connecting to the database
  + Employee or Manager not putting correct information about the customer in the database
* **Use Case-14 An employee can check their hours for the week**
  + **Related Requirements:** REQ1, REQ2, REQ3, REQ4, REQ7
  + **Initiating Actor(s):**
    - Employee
  + **Supporting Software(s):** 
    - MySQL
    - PHP
  + **Actor’s Goal:**
    - The employee will check their hours for the week
  + **Preconditions:**
    - The employee must have an account
  + **Postconditions:**
    - The employee can check the amount of hours they are working that week
  + **Flow of Events for Main Success Scenario:**
* The employee will click on a button to check hours **(Trigger Action)**
* The table will be displayed that will show the employee how many hours they are working that week
* **Exceptions**
  + No Internet Connection
  + Error Connecting to the database
  + Employee did not work that week