**CSCI 4738, CSCI 4739 Senior Design I and II**

**Practice \_\_\_\_\_? (Team 6)**

**Ahmed Aldallee  
James Nguyen  
Tanveer Singh**

**Chase Williamson**

**Software Requirements Specification Document**

**v0.1 (9/15/2022)**

**v0.2 (9/25/2022)**

[**1. Introduction**](#_aaxjyps9hc2n) **5**

[1.1 Purpose](#_9pdv0lw3hfic) 5

[1.2 Scope](#_gzu16fgmqk3v) 5

[1.3 Definitions, Acronyms, and Abbreviations.](#_gtvn2odiy8nf) 5

[1.4 References](#_yebu41x8ckmd) 6

[1.5 Overview](#_ebpwpif0ccsc) 8

[**2. The Overall Description**](#_ly8wj9afaap1) **9**

[2.1 Product Perspective](#_82cyob3l1rj1) 9

[2.1.1 System Interfaces](#_kqa3fywdu8n3) 9

[2.1.2 Interfaces](#_auhziqdow9ic) 9

[2.1.3 Hardware Interfaces](#_lejuhxo577zb) 10

[2.1.4 Software Interfaces](#_nm8b4piepsil) 10

[2.1.5 Communications Interfaces](#_jblrm1xd2jeb) 10

[2.1.6 Memory Constraints](#_7629fhjval33) 10

[2.1.7 Operations](#_6f7hanpnng01) 10

[2.1.8 Site Adaptation Requirements](#_v9mkbzojqalm) 10

[2.2 Product Functions](#_mfbrvtgrptkp) 11

[2.2.1 Coaches (Administrators)](#_3uxs0yf2m3ev) 11

[2.2.2 Parents/players](#_9bmaw0tg7h09) 11

[2.3 User Characteristics](#_gvopnuogrupc) 11

[2.4 Constraints](#_ansp46bh8raq) 11

[2.4.1 Communication Constraints](#_a223daa4fwmi) 11

[2.4.2. Operational Constraints:](#_x7yni43dp5tu) 12

[2.4.3. User Interface Constraints:](#_yvn7tcxeze8q) 12

[2.4.4 Software Constraints:](#_vywbntp5xy9r) 12

[2.5 Assumptions and Dependencies](#_kn2c5kd46xt) 12

[2.6 Apportioning of Requirements](#_xxotsr1859k2)

[**3. Specific Requirements**](#_yydvjspe9d13) **13**

[3.1 External Interfaces](#_6diue9wod54p) 13

[3.2 Functions](#_vi5jnk1ih9lb) 15

[3.2.1 Register New User](#_rbenbnbph8qd) 15

[3.2.2 User Login](#_lmdurksqi6qf) 15

[3.2.3 Password Reset](#_8bgh32tpmoql) 16

[3.2.4 Create Practice Plan](#_f5k5f3m6blf1) 16

[3.2.5 Create a Team](#_6jpg84ywhl6m) 17

[3.2.6 Delete a Team](#_v5puomvk3h9o) 18

[3.2.7 View saved practice plans](#_u0zfs75btgjv) 19

[3.2.8 View previous seasons](#_ezc1ffiyeu17) 20

[3.2.9 View drills](#_ov5d8fo7oqg7) 20

[3.2.10 Edit a team](#_7bc3ylojua14) 21

[3.3 Performance Requirements (SUBJECT TO CHANGE)](#_fqycxr3t6hdk) 23

[3.4 Logical Database Requirements (WIP)](#_pri2c934jufv) 24

[3.5 Design Constraints](#_38tnw9p9qzqd) 27

[3.5.1 Standards Compliance](#_3ig96i591pt5) 27

[3.6 Software System Attributes (Non-functional requirements)](#_zfj9lp1kjct4) 28

[3.6.1 Reliability](#_ctcsu1qxfvbk) 28

[3.6.2 Availability](#_v1svr9dtka0a) 28

[3.6.3 Security](#_9wv7s66ke066) 29

[3.6.4 Maintainability](#_9iub1pszoyl8) 30

[3.6.5 Portability](#_909gt1kn0127) 31

[3.8 Additional Comments](#_dcogod5x1ifd) 31

[**4. Change Management Process**](#_6thv13ozct2n) **31**

[**5. Document Approvals**](#_spt87vmtzzbc) **31**

[**6. Supporting Information**](#_z4z5g4mbs5rc) **31**

# 

# 1. Introduction

## 1.1 Purpose

This software is built for youth baseball coaches to make up practice plans in advance of their team’s practice. Our software will be helping coaches schedule the practice as well as help players and parents visualize the plans.

## **1.2 Scope**

1. Practice Plan Creator
2. What will the product do?
   1. Ability to create a list of drills/scenarios/practice plans and keep track of them.
   2. Ability to add players names and be able to add their names in practice plans.
   3. Will have a drag and drop UI for creating scenarios/drills with names of kids and positions and a baseball diamond.
   4. The software will be used by coaches.
   5. Ability to set time and date on a practice plan.
   6. Emails parents the practice plan for an upcoming practice.
   7. Ability to print practice plans.
   8. Enable coaches to have assistants with authority to access, edit, and print certain parts of the practice plans.
   9. View previous seasons with breakdowns of the focuses each season and timelines for when certain drills were done.
3. What will the product *not* do?
   1. Will *not* be used for tracking data for the season (team management).
4. The software will reduce the time it takes to create a practice plan by hand. The software aims to cut the time for creating a practice plan by at least half.

## 1.3 Definitions, Acronyms, and Abbreviations.

*Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendices in the SRS or by reference to documents. This information may be provided by reference to an Appendix.*

**1- GUI: (**Graphical User Interface) is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators such as primary notation.

**2- DB:** (Database) is an organized collection of data. More specifically, a database is an electronic system that allows data to be easily accessed, manipulated and updated.

**6- UX:** User experience design is the process of defining the experience a user would go through when interacting with a digital product or website.

**7- API:** An application programming interface is a way for two or more computer programs to communicate with each other.

**8- CRUD:** In computer programming, create, read, update, and delete are the four basic operations of persistent storage.

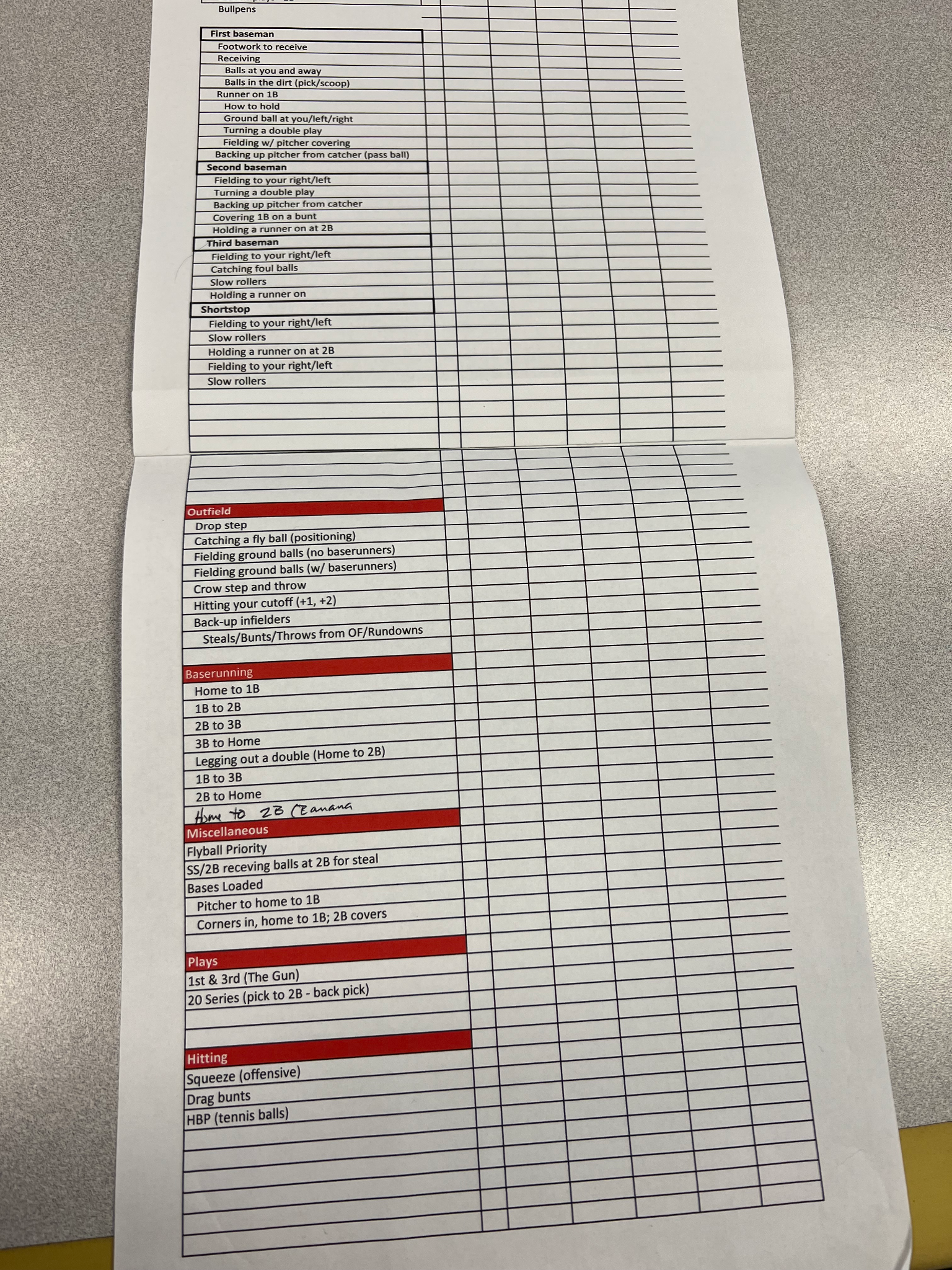
## 1.4 References

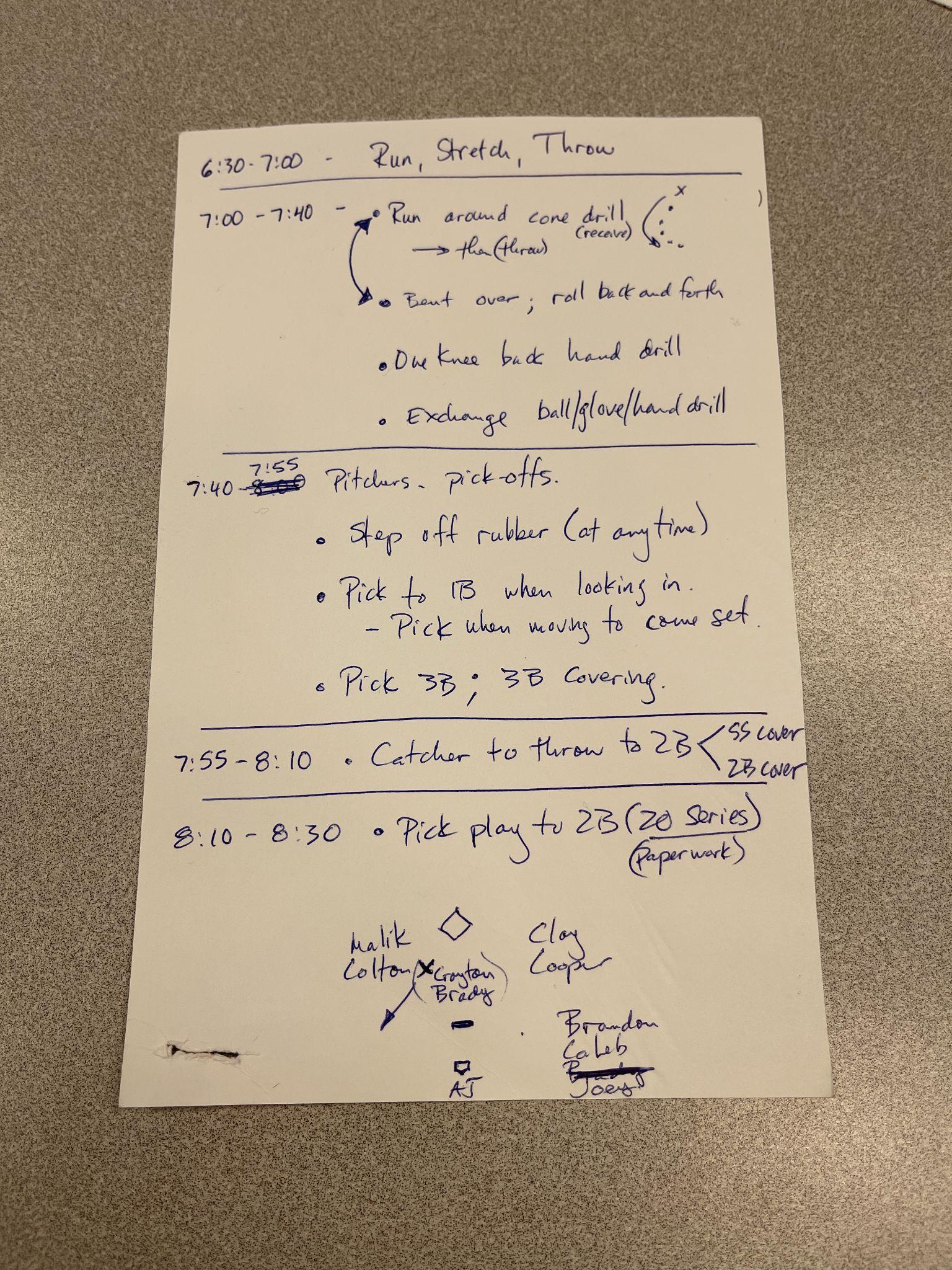
*In this subsection:*

*(1) Provide a complete list of all documents referenced elsewhere in the SRS (2) Identify each document by title, report number (if applicable), date, and publishing organization*

*(3) Specify the sources from which the references can be obtained.*

*This information can be provided by reference to an appendix or to another document. If your application uses specific protocols or RFC’s, then reference them here so designers know where to find them.*

  
**Excel Sheet of all baseball activities/practices and their respective positions. Given to us by Tom himself during the first Client Meeting. 12 September 2022.**

****

**Example practice plan. Given to us by Tom himself during the first Client Meeting. 12 September 2022.**

## **1.5 Overview**

1. Users
   1. Sections **1.1-1.5, 2.2**
2. Developers
   1. Sections **2.1.1-2.1.8,** all of section **3**

# 2. The Overall Description

The software will allow for a registered and authorized user to import their team’s information including player and drill names as well as parent emails. It will also allow for the creation of practice plans utilizing this data. Practice plans are created with a mix of timestamps, graphical drawings, and descriptions of the drill/scenario. Drills used on a certain date will be tracked for reference. The software will be used primarily by coaches who create the practice plans and send them out through email to the parents.

## 2.1 Product Perspective

PracticePlannerLive is a practice planner software that quickly creates, organizes, and manages your practice plans all in one place. Some of the features with this software are Planner Module, Media Uploads, Calendar feature, email communication hub, attendance tracker, and lots more. This is the main product out there that’s most closely related with this project at hand.

However, PracticePlannerLive does not seem to have a GUI for creating templated practice plans with graphics supporting drag-and-drop functionalities.

### 2.1.1 System Interfaces

1. Login/Register
   1. Users create an account and must provide credentials.
2. Import drills/players
   1. Users have an external file which must be uploaded to the software to populate it.
3. Print practice plan
   1. Users can print the created practice plan to an external device such as a printer or to their own files by downloading.
4. Create practice plan
   1. Users can use their mouse and/or keyboard to interact with the GUI to create a practice plan.
5. View the history of when certain drills were done

### 2.1.2 Interfaces

1. GUI for:
   1. Logging in, importing/exporting, and creating plans
      1. Uses mouse and keyboard (no touch)

Interacting with the website will be very similar to how a user would interact with most other websites, by clicking to go through menus and using the keyboard when necessary to give input to the website such as login or credit card information. In our case there is an element of “drawing” when creating the practice plans, but this is still controlled with a mouse and keyboard.

### 2.1.3 Hardware Interfaces

The system has no hardware interface requirements.

### 2.1.4 Software Interfaces

Our client did not specify any specific software to use.

### 2.1.5 Communications Interfaces

Emails will be the main source of communication and will act as our notification system for players/parents.

### 2.1.6 Memory Constraints

This software has no memory constraints.

### 2.1.7 Operations

1. Normal operations (user)
   1. Logging in
   2. Importing players/drills/emails
   3. Edit players/drills (CRUD)
   4. Exporting practice plans
   5. Creating practice plans
   6. Sending emails to parents
   7. Viewing the history of when drills were completed
   8. View the previous season
2. Special operations (user)
   1. Resetting a password
   2. Creating/Updating/Deleting players/drills/names after they have already been imported
   3. Creating a separate, different team
3. Special Operation (system)
   1. Backup/recovery of data

### 2.1.8 Site Adaptation Requirements

This software has no site adaptation requirements.

## 2.2 Product Functions

### 2.2.1 Coaches (Administrators)

1. Register and login with their own credentials.
2. Import players and drills.
3. Add, delete, and modify players data in a team.
4. Add, delete, and modify practice activities list.
5. Design a practice plan, add shapes and symbols, and reference players' names into certain elements (diamonds, rectangles) within the created plan.
6. Print a PDF or physical version of practice plans.
7. Able to save and restore previous practice plans, edit and delete them.
8. Can send practice plans to players' families through email.
9. View specific drills and when they were last used in a practice plan.
10. View the previous season with a breakdown of focus.
11. Invite assistant coaches to create their own account.
    1. Coaches control permissions for the assistant coaches.

### 2.2.2 Parents/players

1. Receive emails that include:
   1. Practice plans in a PDF format with video links.

## 2.3 User Characteristics

* Tom
  + Specified his preference for a website (no app) accessible through a PC/laptop.
  + Tom has specified a liking to a drag-and-drop UI for the graphical parts of the creation as well as keyboard input for descriptions of scenarios that aren’t already pre-made.
* Coaches
  + The “admins”. Can do everything listed in 2.2.1 (product functions).
* Assistant coaches
  + Coaches specify what this user can do through permissions. Can do some or all of the product functions.
* Parents/players
  + Don’t have an account. Coach supplies system with their email and they are emailed the created practice plan rather than logging in to view it.

## 2.4 Constraints

### 2.4.1 Communication Constraints

* This application must be able to communicate with a printer.
* This application must be able to send emails to parents/players.

### 2.4.2. Operational Constraints:

* This application must minimize the user’s effort to create practice plans at a minimum of half the time done on paper.

### 2.4.3. User Interface Constraints:

* A mix of drag-and-drop and text input for creating practice plans.

### 2.4.4 Software Constraints:

* This software/website must be accessible on every modern web browser.

## 2.5 Assumptions and Dependencies

* Assumption: Only a coach/assistant coach needs to access the website, and that the coach has all of the parents’ emails and player names.

## 2.6 Apportioning of RequirementsFuture versions of our software will allow flexibility for all sports practice planning, coaches to have multiple teams, and coaches to create leagues with multiple teams.

# 3. Specific Requirements

## 3.1 External Interfaces

| *Name of item* | Team import |
| --- | --- |
| *Description of purpose* | Lets the coach load all of the players’ names into the software for reusability |
| *Source of input or destination of output* | Coach (user) |
| *Valid range, accuracy and/or tolerance* | First Name, Last Name |
| *Timing* | < 2 seconds (if importing automatically) |
| *Relationships to other inputs/outputs* | N/A |
| *Data formats* | .csv, .txt, manual input |
| *End messages* | “Successfully imported {players} to {team}” |

| *Name of item* | Drill import |
| --- | --- |
| *Description of purpose* | Lets the coach load all of the drills’ names into the software for reusability |
| *Source of input or destination of output* | Coach (user) |
| *Valid range, accuracy and/or tolerance* | Any |
| *Timing* | < 2 seconds (if importing automatically) |
| *Relationships to other inputs/outputs* | N/A |
| *Data formats* | .csv, .txt, manual input |
| *End messages* | “Successfully imported {drills} for {team}” |

| *Name of item* | Print practice plan |
| --- | --- |
| *Description of purpose* | Lets the coach print the practice plan |
| *Source of input or destination of output* | Physical paper |
| *Timing* | < 1 second |
| *Units of measure* | 8.5x11in or 8.5x5.5in |
| *Relationships to other inputs/outputs* | Uses information that is imported to the software (player and drill names) |
| *Data formats* | .pdf, physical |
| *End messages* | “Successfully printed out {practice plan}” |

## 3.2 Functions

### 3.2.1 Register New User

| Use Case Name | Register New User. |
| --- | --- |
| Trigger | The user selects to create a new account. |
| Precondition | None. |
| Basic Path | 1. **User creates a new account**    1. The user clicks “create new account”.    2. The user enters all the required fields that are marked with (\*).       1. The application requires the user to enter a username(\*), password(\*), date of birth, email address(\*) and phone.          1. The application checks if there are any duplicates with the entered username or email.       2. The application checks if all the required fields marked with (\*) are filled out.       3. The user clicks on the create account button.       4. The new user information is stored in the DB.       5. The application starts the login session and displays the main interface. |
| Alternative Path | 1. If the application catches the entered username as a duplicate of an existing username/email in step 1.b.i.1, use case goes back to step 1.b.i 2. If the application checks the required fields marked with (\*) are not filled out in step 1.b.ii, use case goes back to step 1.b. |
| Postcondition | The user can start using the application features according to his level of access and now has an account on the website. |

### 3.2.2 User Login

| Use Case Name | User Login |
| --- | --- |
| Trigger | The user selects to login to the application. |
| Precondition | The user is registered through 3.2.1 |
| Basic Path | 1. The application requires the user to enter his username and password. 2. The user enters his own username and password. 3. The application runs a validation check on the user's credentials by making a call to the DB. 4. The application starts the login session and displays the main interface. |
| Alternative Path | 1. The user enters unvalid username or password, and use case goes back to step 2. 2. The user clicks on the forgot password button and the application executes *Password Reset*. |
| Postcondition | The user can start using the application features according to his level of access. |

## 

### 3.2.3 Password Reset

| Use Case Name | Password Reset. |
| --- | --- |
| Trigger | The user selects to reset his password. |
| Precondition | The user is currently registered. |
| Basic Path | 1. The application requires the user to enter his email. 2. The user enters his own email. 3. The application verifies if the email is valid and exists in the DB. 4. The application prompts the user to enter a new password. |
| Alternative Path | 1. User clicks on *Back* and returns to the main login screen. |
| Postcondition | The user sets a new password for his account. |

### 3.2.4 Create Practice Plan

| Use Case Name | Create Practice Plan |
| --- | --- |
| Trigger | The coach selects to create new plan |
| Precondition | The user has logged in to the main application interface successfully. |
| Basic Path | 1. The use case starts when the user clicks on the *Create Practice Plan* button on the application’s main interface after logging in successfully. 2. The application provides an empty field in the header to set the date. Below it is the main interface to create a practice plan. 3. The application provides users with three different empty square fields in the main body. 4. First field gives an option to set a time frame for a drill. 5. Second field gives an option to write down or drag-and-drop the drill name and notes regarding that drill. 6. Third field gives the option to create a graphical drill utilizing a template of a baseball diamond where empty text boxes are on certain parts of the field corresponding to actual player position in a game. Manually input text into these boxes or drag player names from a sidebar into the box. A toolbar on top will provide means to freeform draw and insert other shapes and text (similar to Paint). 7. Practice plans consist of a collection of these drills/scenarios. Another drill can be added after one has been completed by pressing a ‘+’ button at the bottom of the finished drill. 8. The application will have buttons for *save*, *new, print,* and *send* on top of the navigation bar. 9. The *save* button will save the created practice plan as a PDF format and on the back-end as JSON which can be loaded again if a practice plan needs to be edited. 10. The *new* button will ask to save the current practice plan then restart the practice draft back to the initial state. 11. The *print* button will print a physical copy of the created practice plan (or print to pdf for local file system). 12. The *send* button will be a popup (once clicked) to the user’s default email app/website. The ‘To’ area of the email will be populated with the emails of the parents/players that the user filled in manually or by importing. A PDF of the practice plan will be inserted as an attachment to the email. The user can add notes to the email such as the subject line. Otherwise, they can just hit send since all the work is done for them. 13. The application includes a *Files* button on the right side, when clicked it displays an option to view other practice plans. |
| Alternative Path |  |
| Postcondition | The Coach has created a new practice plan successfully. |

### 3.2.5 Create a Team

| Use Case Name | Create a Team. |
| --- | --- |
| Trigger | The user clicks on *create a team* button from the main screen. |
| Precondition | The user is logged in and is on the main menu. |
| Basic Path | 1. The use case starts when the coach clicks on creating a new team from the main menu. 2. The user enters a team name. 3. The application requires the user to enter a list of players, specifying the players names. This process is done either manually or by importing spreadsheets of data. No validation needed for this step. 4. The user selects to enter players data manually or by importing a file. 5. If the user chooses to enter the players manually, the application will offer an empty field to input text, with a ‘+’ button underneath if the user would like to enter more players. 6. If the user chooses to import a file, the application will offer a button to press and select a CSV file from a device. 7. The user selects a CSV file from a device and imports it into the application.    1. CSV’s must be in the correct format or else the program will reject the bad data with an error message detailing to the user in layman’s terms what is wrong with the data. 8. The user also can enter drills manually or by importing a file in the same fashion as part 4 of this requirement. However, unique to drills is the option to specify a category for the drill for statistic and categorization purposes. 9. The user also can enter parent/player emails manually or by importing a file in the same fashion as part 4. 10. The application displays a *save* and *back* button at the bottom of the interface. 11. The user clicks on the *save* button, and the team is created and saved in the application successfully. |
| Alternative Path | 1. The user clicks on the back button on step(7), if the user clicks on the *back* button, the process of *Create a Team* gets canceled and the user is back to the main screen. |
| Postcondition | The user has successfully created a new team. |

## 

### 3.2.6 Delete a Team

| Use Case Name | Delete a Team. |
| --- | --- |
| Trigger | The user clicks on the list of teams button on the side of the main screen and chooses the team they want to delete. |
| Precondition | The user successfully accessed the main menu and has created at least one team they want to delete. |
| Basic Path | 1. The use case starts when the user clicks on the list of teams button on the side of the main screen. 2. The application will display a list of created teams with two buttons to edit and delete next to each team, and an arrow on top to hide the list . 3. The user clicks on the delete button and the team’s file is “deleted”    1. The software will keep a backup of the team for 30 days in case the user wants to restore the team. |
| Alternative Path | 1. When the user wants to delete a team and when they select ‘delete’ there is a warning that the entire team will be deleted. So, the user can choose not to delete at this step and get the option to cancel/go back. |
| Postcondition | The user deletes the chosen team. |

## 

### 3.2.7 View saved practice plans

| Use Case Name | View saved practice plans |
| --- | --- |
| Trigger | The user clicks view saved practice plans from the main menu |
| Precondition | User is logged in and has a team |
| Basic Path | 1. Once the user clicks to view their saved practice plans, they choose a team they want to see the practice plans for. Having previous practice plans is not a prerequisite, a user’s team may not have any practice plans created yet, but they can still view the empty page. 2. Once the team is chosen, then the user is shown a list of their saved practice plans for that team, initially sorted by most recent. There is a button on the top left giving the user the ability to sort the practice plans by date and name. 3. When a user clicks on a saved practice plan, they are presented with two options, edit and view. 4. If a user clicks edit:    1. The user is brought into the same interface as when they were creating the practice plan (see 3.2.4 in this document). 5. If a user clicks view:    1. The user is shown the PDF version of the practice plan without the ability to edit the practice plan. |
| Alternative Path |  |
| Postcondition | The user is viewing the practice plan, either in edit mode or view mode. |

### 3.2.8 View previous seasons

| Use Case Name | View previous seasons |
| --- | --- |
| Trigger |  |
| Precondition | The user is logged in |
| Basic Path | 1. The user does not need to have a current team or a previous season to view the page, it will just display ‘No previous seasons’ 2. If the user does have a previous season saved:    1. The user chooses which season to view. Seasons can be sorted by date or name.    2. If the user chooses a season:       1. The user is presented with a breakdown by (initially) month of their saved practice plans and drills. The user can choose the duration of time to view at a time. For this requirement, we will assume it is one month. Each month has a list of practice plans and drills sorted by date or name. There is also a breakdown of the season. Graphical views of the stats of the practice plans/drills are presented as well, such as the percentage of defensive vs. offensive practices/ For drills this can be broken down by percentages detailing how many drills were done for infielders vs. outfielders.       2. Season view also shows who was on the team during which practices.          1. This is not ‘attendance’, this is just what players were on the team during the specific time period chosen by the user. |
| Alternative Path |  |
| Postcondition | The user is viewing the history of the specific season. |

### 3.2.9 View drills

| Use Case Name | View saved drills |
| --- | --- |
| Trigger | The user clicks ‘view saved drills’ within the team they choose |
| Precondition | The user is logged in and has created at least one team |
| Basic Path | 1. Once a user is within a team, they can view the saved drills for the specific team. 2. The drills are initially sorted by most recent but can be sorted by date, name, category and “last used”.    1. “Last used” is a property of a drill that says when a drill was last used in a practice plan. |
| Alternative Path |  |
| Postcondition | The user is able to view the list of drills they have imported or manually created |

### 3.2.10 Edit a team

| Use Case Name | Edit a created team |
| --- | --- |
| Trigger | The user clicks on the Edit Team button |
| Precondition | The user is logged in and has at least one team |
| Basic Path | 1. Edit team brings up the same menu as when the user is initially creating the team. All functionality remains the same. See **3.2.5.** |
| Alternative Path | N/A |
| Postcondition | The user is able to edit their created team |

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 3.3 Performance Requirements (SUBJECT TO CHANGE)

* The software supports at least 1000 users logged in at once.
* The software supports at least 10 API calls per second, this includes logging in, uploading/importing, creating practice plans, and downloading the practice plans.
* The software supports uploading at least 100 drills per upload per user for a maximum of 1000 per team
* The software supports uploading at least 100 players per upload per user for a maximum of 200 per team
* The software supports creating one team at a time per user for a maximum of 10 teams per user.
* Each player has one name.
* Each drill has one name.
* Each user can create a maximum of 1000 practice plans per team.

## 

## 3.4 Logical Database Requirements (WIP)

Logical ER Diagram (Ahmed)



Details of each Entity:

**Coach Entity**

| **Attributes** | **Type** | **Description** |
| --- | --- | --- |
| Name | String |  |
| Email | String |  |
| Password | String |  |
| Phone | Integer |  |

**Practice Plan Entity**

| **Attributes** | **Type** | **Description** |
| --- | --- | --- |
| Name | String | Name of practice plan. |
| Cdate | Date | Creation date of the file. |
| Tframe | Date | Time frame of each listed drill. |
| Drill | String |  |
| DrillCategory | String |  |
| Gdrills | JSON | Graphical representation of the drill. |
| Notes | String |  |
| Activity | JSON (array) |  |

**Parents/Players**

| **Attributes** | **Type** | **Description** |
| --- | --- | --- |
| Email1 | String |  |
| Email2 | String |  |
| Name | String | Name of players. |

**Teams**

| **Attributes** | **Type** | **Description** |
| --- | --- | --- |
| Tname | String | Name of the team. |
| Players | String array | Player names. |
| Drills | String array | Drill names |

## 3.5 Design Constraints

1. It’s important to consider commercial constraints such as time and budget as valuable resources needed to be recognized in order to make the utmost quality of product.
2. Picking the correct environment in order to build the software is key, especially when taking account the practices of testing and bug fixing, thus the quality of the product would be massively improved.
3. The most important one is quality, and being able to create the software that ourselves and the client envisioned to be based on the specifications and requirements.

### 3.5.1 Standards Compliance

For the standards of compliance within our software, we will ensure:

1. For legal requirements, we will uphold the standards of plagiarism and copyright in order to keep our product as original as possible for our client and future users.
2. For the website application, we will include terms of use/terms and conditions.
3. Furthermore, the product will require ADA(Americans with Disabilities Act) in which it’s a standard for website accessibility for users with any kind of disability.

## 3.6 Software System Attributes (Non-functional requirements)

### 3.6.1 Reliability

1. There will be a guaranteed delivery as a feature for this software. The participating entities must be able to ensure that all application-level information to be sent to the party has actually been received or error reported.
2. Furthermore, duplication elimination as a feature in order to ensure that all duplicated application-level information is filtered out during the information exchange and is not received as duplicated.
3. It’s important to include a crash tolerance as a feature in which the usage of persistent storage should be supported and must not be mandated as well.
   1. In order to support this, we will use a metric called Probability of Failure on demand (POFOD), which will measure the probability that the system will fail when a service request has been made. This will be useful for which the service is intermittent and less infrequerent. (Note: There is also Rate of occurrence Failures, which is more useful for a system that processes large numbers of similar requests constantly in a short amount of time. Thus haven’t quite decided yet)

### 3.6.2 Availability

1. All of the users of the web application will be able to be accessed anywhere in the USA at around the rate of 99%.
2. Deploying a new module won’t affect the various pages such as the drill page or creating a new practice plan page availability and shouldn’t take longer than 30 minutes. Thus, a notification will be sent of the time showing when the software will run again.

### 3.6.3 Security

To protect the software from malicious access, use, modification, destruction, or disclosure, then the requirements will be:

1. Access permissions for this specific system information only be changed by the set system’s data administrator.
2. The system shall have authentication measures at all entry points, front panel, or inbound network connection.
3. The system shall support multiple authentication approaches at the same time.
4. The system will enforce 2FA in order to improve the security access and defend from unwanted attackers.

### 3.6.4 Maintainability

To ensure the utmost ease of maintenance of the software, we will allow the stakeholder to modify the database with simple buttons with labels such as add, delete, or modify. These three actions will be available for both teams/players and practice activities. This will ensure maintenance without the developers assistance as the amount of use time (How long is a season [year]) will be set by the user, and the variables used in making a practice plan (players, activities, times) are changed by the user, thus, as long as the database that the website is running on is stable, the software provides the stakeholder easy maintenance.

### 3.6.5 Portability

Due to our software being a website, we must ensure portability onto all web browsers. In order to provide cross browser compatibility, we must constantly and incrementally perform cross browser testing. This means that every time a piece of code is written, no matter how small, we must emulate the changes on all browsers in order to ensure that the changes made are consistent on all platforms.

Some examples of portability issues with Firefox:  
FileWriter API, Web SQL Database, XHTML/SMIL animations, EOT fonts  
  
Portability bugs like this will be found and resolved through cross browser testing.

## 3.8 Additional Comments

# 4. Change Management Process

1. For changes initiated on the end of the developers, the developers will come to an agreement before notifying the client of the change. A discussion with the client will be done before modifying any requirements.
2. For changes initiated from the client’s end, we will discuss the changes they would like to make before implementing them.

# 5. Document Approvals

*Identify the approvers of the SRS document. Approver name, signature, and date should be used.*

# 6. Supporting Information