Business Requirements Document

Team Gen ChimpanZ's

October 5, 2022

Team Leader

Mark Fastner

Team Members

Liam Joseph Abalos

Anh Huynh

Justin Le

Aster Lee

Brendan Paing

Github: https://github.com/markfastner/Nuclei

Table of Contents

Summary

This business requirements document will outline all the business rules needed to develop Nuclei. Nuclei is a sports training management software that will provide schedule management tools, user communication tools, and document management tools for sports managers, trainers, and trainees. The goal of this project is to provide an intuitive all-in-one solution for our target group, allowing less time spent on management and more on training quality. Some challenges this project will face is first building and deploying the application, and then ensuring our application delivers on its original goal in future feature updates based on user feedback.

Business Objectives

The business objectives that Nuclei will deliver to Team Gen ChimpanZ, superiors and associates consist of ownership and collaboration experience on a real use system that can be used to reshape the sports training world. Gen ChimpanZ has to gain experience using a large variety of tools, scrum protocol, and team collaboration. Having ownership of Nuclei will give Gen ChimpanZ power in the sports training world and can eventually be used to make profits in the form of advertisements or sponsorship.

Background

Currently in the world of sports training there are many different applications being used to link trainers to trainees, help trainers schedule training sessions with trainees, and help managers manage their business. Currently Managers need to spend a lot of money on systems that manage all their needs. Trainers have a hard time finding quality trainees and struggle with managing all the tools they need to conduct their operations. Trainees have a hard time finding quality trainers and aren't given the best insight into their training progress. There is a serious business need for a free to use powerful system that handles all user group’s sports training needs while providing a high level of reliability through a dual feedback system.

Limitations

In order to construct a realistic business perspective it is important to acknowledge the limitations of Nuclei. Nuclei is restricted in multiple ways which means that there are pain points that need to be considered. Nuclei is restricted financially. Team Gen ChimapanZ has a limited amount of funds towards the development of Nuclei. The cloud service we will need to be using has many free features however as Nuclei plans to become more scalable it will require funding. Nuclei is also restricted by the limitation of the tools that we plan on using. The tools listed in the HLD document are all powerful in their own right however all of them have limitations on what

they can do. In the case of scaling and having Nuclei grow dynamically these tools limit what we can do and additional tools may be needed in order to meet all business needs. Nuclei is also limited by the project scope defined in our project proposal. The scope limits what NUclei is meant to do and must be followed. Finally Nuclei is limited by the business Niche we are attempting to fill, as Nuclei is limited to a sports training session.

Business Requirements

App-Wide Impact

Authentication

Login

Being logged in will be considered a precondition for all standalone feature requirements except for the “Send Notification” function under Notification System.

Functional

<table>
<tr><td colspan="2">Preconditions:
● The user has an existing account registered with the application.</td></tr>
<tr><td colspan="2">Requirements:
● A user enters their account credentials to access the application.
● Can only access the login function if there is content in the user input.</td></tr>
<tr><td>Success Outcomes:
1. User enters credentials correctly and is given access to their account and taken to the dashboard
2. User enters credentials incorrectly and is denied access

All 2 of these are required to be considered a success.</td><td>Failure Outcomes:
1. User enters credentials correctly but is denied access to account
2. User enters credentials correctly but isn’t taken to the dashboard
3. User enters credentials incorrectly but is given access to account</td></tr>
<tr><td colspan="2">Post-Condition:
The user is now logged into their account and can access the application.</td></tr>
</table>

Non-Functional

1. Authentication and application access processes under 5 seconds after the user submits the login request.
2. The user receives an interface response if they attempt to authenticate with invalid

credentials.
3. The user will be timed out for a period of time if they submit too many invalid logins.
4. The user receives a prompt that login was successful if there is a delay between successful authentication and application access.

Logout

Functional

Preconditions:
- The user must be logged in.

Requirements:
- The user should be able to access a logout function while authenticated to exit the application.

Success Outcomes:	**Failure Outcomes:**
1. User is logged out and taken to the login screen. 2. User must login again to access the application.	1. User does not get logged out 2. User is not taken back to the login screen 3. User can still access application features after logging out

Post-Condition
The user is now logged out of their account.

Non-Functional

1. Users are prompted with confirmation to see if they really want to log out.
2. Application announces to the user that they have logged out.

User Management

Create Account

Functional

Preconditions
- User does not have an existing account associated with the same email.

Requirements
- User selects option to create new account
- User will input credentials for account
    - First Name
    - Last Name
    - Birthdate
    - Email

<table>
<tr><td colspan="2">

Username
Password
Password confirmation

User accepts EULA.

</td></tr>
<tr><td>Success Outcomes
1. User inputs valid credentials and account is created correctly
2. User inputs invalid credentials and required to reinput until correct
3. Account is added to database

All 3 of these outcomes are required to be considered a success.</td><td>Failure Outcomes
1. User inputs valid credentials but account is not made.
2. User inputs valid credentials but account does not reflect correct information.
3. User inputs invalid credentials but account is made.
4. User successfully created an account but is not added to the database.</td></tr>
<tr><td colspan="2">Post-Condition
● Account is created and added to the database</td></tr>
</table>

Non-Functional

1. First and Last names are at most 32 text characters.
2. Username is unique, at most 32 text characters, and does not contain profanity.
3. Birthday must be a valid date and must show that the user is over the age of 13.
4. Passwords must be at least 8 characters, must contain at least 1 capital and 1 lowercase character, at least 1 number, and at least 1 special character, and this requirement will be known to the user.

Change Password

Functional

<table>
<tr><td colspan="2">Preconditions
● User is in account settings</td></tr>
<tr><td colspan="2">Requirements
● User selects ‘change password’
● User inputs old/current password to confirm they want to change password
● User will input a new password
● User will confirm new password</td></tr>
<tr><td>Success Outcomes
1. User correctly inputs requirements and password is changed
2. Password is updated in database</td><td>Failure Outcomes
1. Password is changed with incorrect requirements.
2. User correctly inputs requirements but the password does not change.</td></tr>
</table>

	3. User correctly inputs requirements but password is not updated in database 4. Wrong password is updated in database. 5. Wrong user's password is updated in the database.
Post-Condition ● Password is changed and updated in the database.	

Non-Functional Requirements

1. New passwords must be at least 8 characters, must contain at least 1 capital and 1 lowercase character, at least 1 number, and at least 1 special character.

Forgot Password

Functional Requirements

Preconditions: ● User is in log in screen	
Requirements: 1. User clicks on 'forgot password' 2. User is prompted to enter email and then verify email	
Success Outcomes 1. When user enters and verifies their email he is sent to a page to input a new password 2. If user enters email but does not confirm user stays on confirmation page Only #1 is required to be considered a success.	**Failure Outcomes** 1. User is able to change password without verifying email 2. Users password is not changed after entering, verifying, and entering a new password 3. Wrong password is updated in database. 4. Wrong user's password is updated in the database. 5. User does not receive verification email
Post-Condition: The database is now updated with the new password.User now has a new password and the database is updated with the new pass	

Non-Functional Requirements

1. The user can only change password 3 times a week.
2. The user can not have the same password as the current one.

Delete Account

Functional Requirements

Preconditions:
- User is in account settings.

Requirements:
- User selects 'Delete Account'
- User will enter account credentials
    - Email
    - Password
- User will get confirmation prompt (Yes/No) to confirm account deletion
- If 'Yes' user is removed from any groups they were apart of

Success Outcomes:	**Failure Outcomes:**
1. If user enters valid credentials and confirms to delete account the user is removed from all groups and from our database 2. If user enters valid credentials and denies then nothing happens and user is taken back to account settings 3. If user enters invalid credentials he is not able to delete account Only #1 is required to be considered a success.	1. User inputs valid credentials and selects yes, but account is not deleted from our database 2. User inputs valid credentials and selects yes, but user is not removed from all groups 3. User inputs valid credentials and selects yes but remains in account 4. User inputs valid credentials and selects no and anything happens 5. User inputs invalid credentials and is able to delete account

Post-Condition:
Account is deleted; Removed from all groups and removed from the database.

Non-Functional

1. If user inputs the wrong password, the user's email will be notified
2. If user inputs the wrong password for an email 5 times, the user is kicked out of the account
3. User credentials can no longer be used to sign in after account has been deleted

Logging

Logging will be considered a non-functional requirement for all standalone features.

Functional

Preconditions
- A user exists.
- A user performs an action on the application that invokes a business rule.

Requirements
- A persistent data storage where logs are written.
- The feature interacted with must be fully functional.
- The log should contain a timestamp of when it is created.

Success Outcomes	**Failure Outcomes**
1. The log contains time, username, and the action the user has taken on using the feature. 2. The log is saved to the database. All 2 of these outcomes are required to be considered a success.	1. The user's action has not been logged. 2. Parts of the log that should be filled in are empty. 3. Logs written are false information.

Post-Condition
The log is saved in the database for future use.

Non-Functional

1. The log will be listed in reverse chronological order.

Analytics

Functional

Preconditions ● A collection of data taken using the logging function.
Requirements ● Track user traffic in the system. ● Create user-base demographic with user profile data. ● Track what items on the interface are used most often.
Success Outcomes 1. An accurate trend is determined. | **Failure Outcomes** 1. An incorrect trend is determined.
Post-Condition ● Trends and analytics are studied to learn the patterns and behaviors of clients. ● Following trends steers development in a direction towards making changes to the application to benefit as many users as possible.

Non-Functional

1. Trends are displayed visually as graphs: frequency, heat, line, etc.

Standalone Features

Dashboard Interface

Create Profile

Functional

Preconditions User is logged in**.**
Requirements: ● A user can create a profile which details the type of profile, name, birthdate, gender, location, and additional information about themselves. ● The application will use user account data to fill in name and birthdate. ● An account accessing the application for the first time will automatically be taken to create a new profile. ● There are 3 profile types: Manager, Trainer, and Trainee. ● A single user can have a maximum of 3 profiles. ● A single user can not have two of the same profile type.
Success Outcomes 1. Profile is created with the correct | **Failure Outcomes** 1. Profile is not created.

information. 2. New profile is stored in the database. All 2 of these outcomes are required to be considered a success.	2. Profile is created but has different information than the user input. 3. Profile is not stored in the database.
Post-Condition The user now has a new profile associated with their account.	

Non-Functional

1. The application will generate a success message when the user creates a new profile.
2. The application will show an interface graphic to denote different profile types.

Delete Profile

Functional

Preconditions User has an existing profile.	
Requirements ● A user can delete a profile which will remove the profile type from their account. ● Any class scheduling related to that profile will be removed from the database and from the user's account. ● A user can not delete a profile if they only have one profile associated with their account.	
Success Outcomes 1. The profile data is deleted from the database. 2. The user can no longer access the profile. 3. Any scheduling data related to that profile is deleted. All 3 of these outcomes are required to be considered a success.	**Failure Outcomes** 1. The profile is not deleted from the database. 2. The profile is not deleted from the user interface. 3. Any scheduling data related to that profile is not deleted. 4. The user can delete a profile when they only have one profile associated with their account.
Post-Condition The profile is now deleted from the application.	

Non-Functional

1. The application generates a confirmation prompt asking if the user wants to delete the profile.

2. If the profile is a trainer type and has any classes created or training scheduled, send an additional warning to the user.
3. If the profile is a trainee type and is in a class or has training scheduled, send an additional warning to the user.

Edit Profile

Functional

Preconditions:
- User has an existing profile.
- User is currently viewing their list of profiles.

Requirement
- The user can edit the information of gender, location, and additional information about themselves.
- Can not edit name or birthdate within the profile settings.
- The user saves the information.

Success Outcomes	**Failure Outcomes**
1. The gender data is updated in the database. 2. The location data is updated in the database. 3. The additional information data is updated in the database. All 3 of these outcomes are required to be considered a success.	1. The information of gender, location, or additional information about themselves are not changed on the profile. 2. The information of gender, location, or additional information about themselves are not updated in the database 3. The information of gender, location, or additional information about themselves is updated on the wrong user.

Post-Condition
Users can see the profile with updated information.

Non-Functional

1. The application generates a success message if any profile information is modified.

Switch Profile

Functional

Preconditions

- A user has 2 or more existing profiles.

Requirements

- The user goes to their list of profiles.
- The user selects the profile that they want to change to.
- Can not switch to a profile they are currently using.
- Switching a profile will change what items the dashboard interface will display.
- All profile types can see:
    - Personal Calendar
    - Account Settings
    - View own profile
    - Switch Profile function
    - Search Users function
    - Recent Conversations function
- User is a trainee and can see:
    - List of classes they are in
- User is a trainer and can see:
    - List of classes they have created
- User is a manager and can see:
    - Trainer list
    - Training Camp list

Success Outcomes

1. The user is not using the profile they just switched from.
2. The user accesses Nuclei using the specified profile.
3. The user can see

All 3 of these outcomes are required to be considered a success.

Failure Outcomes

1. The user stays on the previous profile.
2. The user changes to a different profile than the one specified.
3. The user using a profile type can see dashboard items they are not supposed to see.

Post-Condition

The user is now using a different profile.

Non-Functional

1. The application will generate a success message if the user successfully switches profiles.

Personal Calendar

Functional

Preconditions:

● User is logged in. ● User has an existing profile. ● User is viewing the dashboard interface.
Requirements ● Dashboard should allow user to access their personal calendar
Success Outcomes 1. Personal Calendar opens. 2. The user is viewing the personal calendar associated with their profile. All 2 of these outcomes are required to be considered a success.
Failure Outcomes 1. Calendar does not open. 2. Calendar opens but the user can not interact with it. 3. User is viewing the calendar of a different user.
Post-Condition The user is now viewing their personal calendar.

Non-Functional

1. The personal calendar will use different colors to distinguish training sessions if the user is in more than one class.

Search For User

Functional

Preconditions The user is on the dashboard interface.
Requirements ● The user can search for other users on the application, to which they can then access the other user's profile. ● Can search using the user's email address. ● Can search using the user's name. ● Can search using the user's username.
Success Outcomes 1. The user can access the specified user's profile from the search function. 2.
Failure Outcomes 1. The specified user exists in the database but the search function does not show it. 2. The
Post-Condition The user can now search for other users within the application.

Non-Functional

1. When searching for a user, additionally show the user's created profile types using interface graphics.

View list of trainers

Functional

Preconditions
- User is a manager.
- User is on the dashboard interface.

Requirements
All trainers under the manager are shown.
Managers can click on any of their trainers and go to their profile.
While on one of their trainers profiles Managers have a high level of access to their trainers profiles by being able to manage the trainers classes.

Success Outcomes:	**Failure Outcomes**
1. All trainers are listed when manager opens list of trainers 2. Managers can go on all of their trainers' profiles and have full access over their classes.	1. Not all of the trainers under the manager are shown 2. Manager is unable to access one of their trainers profiles 3. Manager is unable to manage one of their trainers classes

Post-Condition
Manager now sees all his trainers and is able to manage them and delegate work.

Non-Functional

Trainers are ordered by Last Name
All trainers should be listed within 3 seconds of manager opening the list

Add Trainer to Trainer List

Functional

Preconditions
- User is a manager.
- User is viewing a list of trainers.
- User clicks on add new trainer function.

Requirements:
- Manager searches for a trainer by name or username.
- Manager selects the trainer from the search and a request is sent to the trainer.
- Trainer will receive a notification and can choose to accept or deny the managers request
- If the trainer accepts they will be listed as a trainer belonging to the manager

Success Outcomes	**Failure Outcomes**
1. Trainer is able search find and select the trainer they want and after the trainer accepts the request the trainer is added under the managers trainer list	1. Trainer does not receive notification 2. Trainer does not show up from managers search 3. Trainer accepts notification but does not get added under the managers trainer list

Post-Condition: The manager has a new trainer under them and it is shown in the trainer list

Non-Functional

It must take less than 3 seconds after a trainer accepts a managers request for our system to establish the trainer as under the manager

Training Camp List

Functional

Preconditions: User is a manager and on dashboard

Requirements: When accessed manager is displayed a list of training camps which they can manage and interact with

Success Outcomes:	**Failure Outcomes**
1. Manager is able to view and manage all camps under the camp list	1. One or more of the training camps do not show up

Post-Condition:
The manager user now has access to their training camps.

Non-Functional

1. Training camps are ordered by date they are added

Create Training Camp

Functional

Preconditions Manager user is viewing their list of training camps	
Requirements A New training camp is added to the training camp list and the manager has complete access to it. On creation manager is taken to training camp settings to add classes to the training camp	
Success Outcomes 1. New training camp is created 2. Manager is taken to training camp settings	**Failure Outcomes** 1. Training camp does not get created 2. Manager isn't able to add classes to training camp
Post-Condition: Manager has created a new training camp and has filled it with desired classes	

Non-Functional

Training camp is created in our database and presented to user i under 5 seconds
Generate a success message when

Training Camp Settings

Functional

Preconditions A user manager is viewing a specific training camp	
Requirements ● User can choose to add a class to the training camp ○ User is presented with all the classes from all his trainers ○ User can select a class to add it to the training camp ● User can choose to delete a class from the training camp	
Success Outcomes: 1. User is able to add classes to camp 2. User is able to deleted class from	**Failure Outcomes** 1. User can add classes not created by a trainer in their trainer list.

camp Any of these need to happen in order for the requirement to be considered a success.	2. User selects class to add but it doesn't get added to camp 3. User selects class to delete but it doesn't get deleted 4. User is unable to view the list of classes from all his trainers
Post-Condition Manager has changed his settings for his training camp to satisfy his needs	

Non-Functional

Adding a class to a training camp should happen in less than 8 seconds from user selection

Class System

View List of Classes

Functional

Preconditions ● User is a trainer or trainee. ● User is on the dashboard dashboard.	
Requirements ● The correct class list associated with the user should be shown.	
Success Outcomes 1. The class list associated with the user should be shown.	**Failure Outcomes** 1. The user can not see their list of classes. 2. The user can see their list of classes but cannot interact with it. 3. The application returns a list of classes associated with a different user.
Post-Condition The user is now accessing the class system view model and can view all the classes they are a part of.	

Non-Functional

1. When no classes are in the list, the application should inform the user that they are not associated with any classes. 2. Add different ways to organize the list of classes.

View Class

Functional

Preconditions ● User is currently viewing their list of classes. ● User selects a class to view.
Requirements ● A user can only view a class if they are part of the class list.
Success Outcomes: 1. The class shown by the system is the same as the one the user selected.   **Failure Outcomes** 1. No class view is shown 2. A class view different from the one selected is shown
Post-Condition The user is now viewing a specific class.

Non-Functional

1. The user interface switches to the class view from the class list view without a jarring transition.

Create Class

Functional

Preconditions ● User is a trainer. ● User is currently viewing their list of classes.
Requirements Click on the new class functionality within the class system view model. Click confirm to create a new class.
Success Outcomes 1. A new empty class is created 2. The user is taken to the empty class view model. All 2 of these outcomes are required to be considered a success.   **Failure Outcomes** 1. Class is not created. 2. The user is not taken to the new class 3. Class contains other users 4. Class is accessible through different profile under same account

Post-Condition The user is now viewing and has access to the new empty class they created.

Non-Functional

1. Class creation and user access defined should be created in our database in less than 6 seconds.

Manage Class Settings

Functional

Preconditions ● The user is a trainer or manager ● Trainer is viewing a specific class they own. ● Manager is viewing a specific class of a trainer under them.	
Requirements ● Managers are able to manage classes of trainers under them. ● The manager and trainer is given the option to set the class size limit and invite new trainees to the classroom through a search bar. ● The class size limit can not be less than the amount of trainees currently in the class. ● The class size limit can not be 0.	
Success Outcomes 1. All the trainees that are added to the class have access to the classroom 2. There are no more trainees in the class then specified by the limit All 2 of these outcomes are required to be considered a success.	**Failure Outcomes** 1. 1 or more trainees that have been added to the class does not have access to the class 2. Class settings aren't saved according to what the user imputed. 3. More trainees than specified by the class size limit are able to join the class.
Post-Condition Manager or Trainer has modified the class settings for a specific class.	

Non-Functional

1. The application shows a success message if changes to the class settings were made.

Invite to Classroom

Functional

Preconditions ● Manager or Trainers is accessing the manage class settings function	
Requirements ● Manager or Trainer enters the name or username of a trainee into the search bar ● Click on the trainees who come up to add them to the class ● Can not invite a trainee already in the class.	
Success Outcomes 1. After inputting a name or username of a legit trainee and selecting them, that trainee is added to the class.	**Failure Outcomes** 1. A trainee who should have come up from search does not show. 2. A trainee is selected but is not added to class.
Post-Condition The designated trainee is added to the class.	

Non-Functional

1. A user must show up from the user search within 5 seconds if they exist.

Post Message to Class

Functional

Preconditions ● User is a trainer or trainee ● The user is viewing a specific class.	
Requirements ● Message will consist of user text input ● Message is restricted to 500 characters ● A message containing no characters is not allowed to be posted.	
Success Outcomes 1. A valid message based on the requirements can be posted to the class. 2. Message is viewable by all users who are in the class after posting.	**Failure Outcomes** 1. The user input box contains more than 500 characters. 2. A valid message is not posted to the class. 3. A message containing no text characters can be posted

All 2 of these outcomes are required to be considered a success.	4. A user who has access to the class can not view the post.
Post-Condition The message is now posted to the class for all authorized users to view.	

Non-Functional

1. The message includes a timestamp of when it was sent for the users to see.

Leave Classroom

Functional Requirements

Preconditions ● User is a trainee. ● User is in a class. ● User is currently viewing a specific class.	
Requirements ● The trainee should no longer be able to view the class in their class list after leaving. ● The database should be updated with the new class list.	
Success Outcomes 1. The correct class data is updated in the database. 2. The class disappears from the class view model for the user who left. 3. Any user still in the class should not be able to see the user who left in the class list. All 3 of these outcomes are required to be considered a success.	**Failure Outcomes** 1. The database is not updated with the modified class list. 2. A different class in the database than the one selected is updated. 3. The correct class is updated but the view model does not update.
Post-Condition The trainee has left the class and the class is updated to reflect the new list of users.	

Non-Functional

1. The user interface generates a confirmation prompt to confirm class leave or to return to the class list view model with no changes.
2. The database request, database response, and user interface should finish in under 3 seconds.

Delete Classroom

Functional

Preconditions
- User is a trainer.
- User is the one who created the class.
- User is currently viewing the class view model.

Requirements
- Class data should be deleted from the database.
- Update class view model to no longer show class for all users.

Success Outcomes	**Failure Outcomes**
1. The correct class is deleted from the database. 2. The class should no longer be available to view for all associated users All 2 of these outcomes are required to be considered a success.	1. The class is not deleted from the database. 2. A different class is deleted from the database. 3. The class still appears on the user interface after deletion.

Post-Condition
The class was successfully deleted from the application.

Non-Functional

1. The user interface generates a confirmation prompt to confirm class deletion or to return to the class system view model with no changes.
2. The system will delete a class automatically after 3 weeks of inactivity (no posts or schedule training).
3. The database request, database response, and user interface should finish in under 3 seconds.

Schedule Training Session

Functional

Preconditions
- User is a trainer type.
- The user is viewing a class with at least one trainee in it.
- Personal Calendar function in Dashboard Interface.

Requirements
- A trainer can schedule a date and time for a training session for all trainees in class.
- Date and time is required to schedule.
- Personal calendars of each user in the class should be updated to show the training session.

Success Outcomes	**Failure Outcomes**
1. The scheduled training session is created with a specified date and time. 2. All users invited to the class including the trainer can see the new training session in their personal calendars.	1. The scheduled training session is not created. 2. The scheduled training session is created without a date and time. 3. The scheduled training session is created but not viewable in the personal calendar by users in the class.

Post-Condition:
All users in a particular classroom can now see a scheduled time for training.

Non-Functional

1. The trainer user receives a confirmation prompt to ensure the right date and time were selected.
2. The trainer user receives a success message after scheduling a training session.

Notification System

Manage Notifications

Functional

Preconditions
- User is accessing their account settings.

Requirements
- A user can specify whether to receive notifications for specific feature actions.
- Class System
  - Invite received from a trainer
  - Invite accepted from a trainee
  - Class post created
  - Class deleted
  - Training scheduled for class
- Dual Feedback System
  - Review was left on user profile
- Direct Messaging System
  - User was sent a message

● Document System ○ Granted access to a private document
Success Outcomes 1. User is able to change their notification settings 2. Future notifications reflect the new user settings. All 2 of these outcomes are required to be considered a success.
Failure Outcomes 1. The notification settings are not updated. 2. The user receives a notification from an action they specified not to send notifications.
Post-Condition Notification sending will now be based on the user specified setting.

Non-Functional

1. The application should generate an interface message to confirm that notification settings were changed.

Send Notification

Functional

Preconditions ● An action related to a user's account occurred. ○ No accessing or viewing functions from any feature is included in this.
Requirements ● Send notification as an email to user's third-party email address inbox ● Generate notification message using following format ○ Timestamp of when action occurred ○ Action
Success Outcomes 1. Notification sent to correct email inbox 2. Notification content is correct All 2 of these outcomes are required to be considered a success.
Failure Outcomes 1. Notification message not sent to user's third-party email address inbox 2. Notification message sent to different user's third-party email address inbox 3. Message content is not related to the action that occurred.
Post-Condition Notification sent to third-party email address associated with user's account

Non-Functional

1. Send a notification within 3 minutes of the action occurring if the third party email service is functional.

Dual Feedback System

View Reviews

Functional

Preconditions
- User is viewing a user profile, either their own or a different user.

Requirements
- Must show all reviews left for a user profile.

Success Outcomes	**Failure Outcomes**
1. The user can view the review list for the specified user.	1. The user can not see the review list. 2. The user sees the review list for a user different than the one specified.

Post-Condition
The user is now viewing reviews created by the dual feedback system.

Non-Functional

1. The data request, data request response, and user interface update to get the list of reviews should take less than 3 seconds.

Leave Review for User

Functional Requirements

Preconditions
- User is viewing a different user's profile.

Requirements
- Can only leave a review on a different user that a user has performed scheduled training with.
    - A single user can only create one review per training session.
- Only trainer profiles can leave reviews on trainee profiles.
- Only trainee profiles can leave reviews on trainer profiles.

- Numerical rating categories depending on whether the user being reviewed is a trainer or trainee.
- Ratings range from 1 to 10.
    - Trainer: coaching ability, skill, and training
    - Trainee: training summary and training satisfaction
- Both review types will have an additional feedback user input
- Required to fill in all categories except for the additional feedback input
- Can only submit review after required sections are filled in.

Success Outcomes	**Failure Outcomes**
1. The review is viewable by all users on the specified user's profile. 2. The review is saved to the database. All 2 of these outcomes are required to be considered a success.	1. The review was not created. 2. The review is not viewable on the specified user's profile. 3. A review can be submitted without filling in all required sections.

Post-Condition
A review is now viewable on a user's profile.

Non-Functional

1. The review includes the date of the training session.

Direct Messaging System

Access Through Dashboard

Functional Requirements

Preconditions
- User is on the dashboard interface

Requirements
- User clicks on the direct messaging tab to show all recent conversations
- User clicks on any conversation displayed to enable direct messaging with another user

Success Outcomes	**Failure Outcomes**
1. Conversations within three days are displayed in the direct messaging tab 2. The direct messaging view model is accessed with the correct specified user	1. Direct messaging view model is not accessed. 2. A conversation is accessed with the incorrect user. 3. The direct messaging view model does not display messages from a previous

All 2 of these outcomes are required to be considered a success.	conversation.
Post-Condition User is in the direct messaging view model with the specified user.	

Non-Functional Requirements

1. It takes less than or equal to 2 seconds for the application to access the direct message system.

Access Through Profile

Functional

Preconditions ● User is on a different user's profile.	
Requirements ● User clicks on the direct messaging tab on a user's profile to access the direct messaging view model. ● If a recent conversation exists, switch to the direct messaging view model with the specified user. ● If a recent conversation doesn't exist, create a new direct messaging view model involving the two specified users.	
Success Outcomes 1. Direct messaging view model is accessed with the specified user as a recipient.	**Failure Outcomes** 1. Direct messaging view model is not accessed. 2. A conversation is accessed with the incorrect user. 3. The direct messaging view model does not display messages from a previous conversation.
Post-Condition User is now in the direct messaging view model with the specified user	

Non-Functional

1. It takes less than 2 seconds for the application to access the direct message system.

Message Exchange

Functional

Preconditions ● User is accessing the direct messaging view model to chat with another user.
Requirements ● A user should be able to input text content and send that message to the other user. ● 200 text character limit. ● Can not send an empty text field.

Success Outcomes	**Failure Outcomes**
1. The message is sent to the correct user. 2. Both sender and receiver users can see the message. All 2 of these outcomes are required to be considered a success.	1. The user can send an empty text field as a message. 2. The message does not get sent. 3. Either the sender or receiver user can not see the message after it is sent.

Post-Condition The two users can now see the message one of the users created and sent.

Non-Functional

1. The message includes a timestamp of when it was sent for the users to see.

Document System

Access Document System

Functional

Preconditions ● The user is currently accessing a user profile, either their own profile or a different user's profile.
Requirements ● Application should provide access to document system associated with the user profile ● Only display documents the user is authorized to view.

Success Outcomes	**Failure Outcomes**
1. Document system view model appears. 2. The user can interact with the view model. 3. The user can see all documents they	1. The document system view model does not appear. 2. The document system view model appears but the user cannot interact with it.

are authorized to view. All 3 of these outcomes are required to be considered a success.	3. The user can see documents they are not authorized to view.
Post-Condition The user is now accessing the document system view model.	

Non-Functional

1. Display the file size for each document
2. Display the upload date for each document

Upload Document

Functional

Preconditions ● User is currently accessing the document system view model	
Requirements ● Application should store a file in the database and update the document system to show the new file for viewing. ● Only accept PDF files ● File size limit of 8MB ● Set document visibility to private by default when a new document is uploaded.	
Success Outcomes 1. User chooses a valid file type. 2. File is stored in the database. 3. View model is updated to reflect database response. All 3 of these outcomes are required to be considered a success.	**Failure Outcomes** 1. User uploads a non-PDF file. 2. User uploads a file larger than 8MB. 3. File was not saved to the database. 4. Document view model did not update to show changes.
Post-Condition The uploaded document is now displayed in the document view model for the user to view.	

Non-Functional

1. The user interface generates a message to confirm that a document was uploaded.

View Document

Functional

Preconditions ● User is currently accessing the document system view model ● User has at least one file uploaded to the document system.	
Requirements ● Application should request the user selected file from the database and display it for the user to view after receiving a response.	
Success Outcomes 1. The correct file is returned from the database. 2. The application switches to the file view model. All 2 of these outcomes are required to be considered a success.	**Failure Outcomes** 1. No file is returned from the database. 2. A different file is returned from the database. 3. A file is returned but the user cannot view it.
Post-Condition The user is now viewing the PDF file of the document they selected.	

Non-Functional

1. The data request, response, and switch to file view should all take less than 3 seconds.

Remove Document

Functional

Preconditions ● User is currently accessing the document system view model. ● User has at least one file uploaded to the document system.	
Requirements ● Application should delete the specified file in the database and update the document system view model to no longer show the file.	
Success Outcomes 1. The correct file is deleted from the database. 2. The document system no longer shows the file in the view model. All 2 of these outcomes are required to	**Failure Outcomes** 1. The specified file is not deleted from the database. 2. A different file in the database is deleted. 3. The view model does not update its interface.

be considered a success.	

Post-Condition
The document is now removed from the database and the document view model.

Non-Functional

4. The user interface generates a confirmation prompt to confirm document deletion or to return to the document system view model with no changes.
5. The user interface generates a message to confirm that a document was deleted.

Change Document Visibility

Functional

Preconditions
- The document system is currently being accessed.
- A document is already uploaded to the user's profile.

Requirements
- A user can change the visibility setting on a document they uploaded.
- Public visibility should be viewable by all other users.
- Private visibility should only be viewable by the uploader user until they authorize more users to view.

Success Outcomes	Failure Outcomes
1. The document visibility type changes to the user specified setting. 2. The document visibility type follows the logic specified in the requirements. All 2 of these outcomes are required to be considered a success.	1. The document visibility type does not change to the user specified setting. 2. The actual document visibility is different from the visibility type specified.

Post-Condition
The document visibility type is now changed to what the user specified.

Non-Functional

1. The document has an interface graphic to designate its visibility type at a glance.

Private Document Authorization

Functional

Preconditions ● The document system is currently being accessed. ● A document is already uploaded to the user's profile. ● The document in question is set to private visibility.	
Requirements ● The user can add new users to the authorization list. ● The user can remove users from the authorization list. ● The user can save changes made to the authorization list.	
Success Outcomes 1. Document is only viewable by the uploader user and authorized users on the list.	**Failure Outcomes** 1. Document is viewable by unauthorized users 2. The list of authorized users does not update if changes were made.
Post-Condition All newly authorized users can now view the private document through the uploader's profile.	

Non-Functional

1. The application shows a success message if changes to the authorized user list were made.

Glossary

Term	Definition
End-User License Agreement (EULA)	An agreement between the user of the product and software producer that lets the user know what the product is allowed to do with their data.
View Model	The data an application displays on a view/page.

References

1. https://cutewallpaper.org/24/sports-png/view-page-24.html

Version Changelog

Version	Submission Date	Changelog
1	10/03/22	Initial Draft Version
2	10/05/22	Initial Submission ● Reformat Business Requirements ● Remove the following sections: ○ Scope ○ Features ○ Personnel Requirements ○ Delivery Schedule ○ Reporting and Quality Assurance ○ Risks ● Update Glossary section