

Table of Contents

1.0 Introduction	2
2.0 User Roles	3
3.0 Glossary	5
4.0 Functional Requirements	7
4.1 System Response Table	7
4.2 Cross Functional Process Map	9
4.3 Use Cases	10
4.4 Logical Data Model	36
4.5 Data Dictionary	37
4.6 Logical Process Model.....	41
4.6.1 Data Flow Diagrams	41
4.6.2 Data Description	52
4.6.3 Process Specification	53
4.7 User Experience Diagrams.....	59
4.8 Wireframes.....	68
5.0 Non-Functional Requirements.....	85
5.1 Usability.....	85
5.3 Accessibility.....	85
5.3 Availability	86
5.4 Documentation and Training	86
5.5 Performance	87
5.6 Capacity	87
5.7 Security	88
5.8 Longevity	89
6.0 Collaborating Systems and Additional Software	89
7.0 Schedule	90

1.0 Introduction

This document is designed to thoroughly describe all aspects of the creation of a workout management system for the Taylor University Athletic Department. The purpose of this document is to include details on functional and non-functional user and system requirements, descriptions of user roles, and to provide a rough schedule of the development process.

The proposed system will address the current system problems of the Athletic Department. The current system is mostly paper-based and suffers from poor organization, data loss, and general inefficiency. Coaches are not able to easily monitor the progress of their athletes, especially when workout data sheets are lost. The new system will remove the need for paper documents and will provide better efficiency for both athletes recording their information and the coaches' ability to create and monitor workouts.

2.0 User Roles

There are numerous possible users that might interact with this system. Below we will highlight who these users are and how they will be interacting with the system.

View Only/Any User: A general user with the ability to perform a number of general functions. The View Only can view and analyze data, as well editing their own personal information. All other users, with the exception of Athletes, have these abilities.

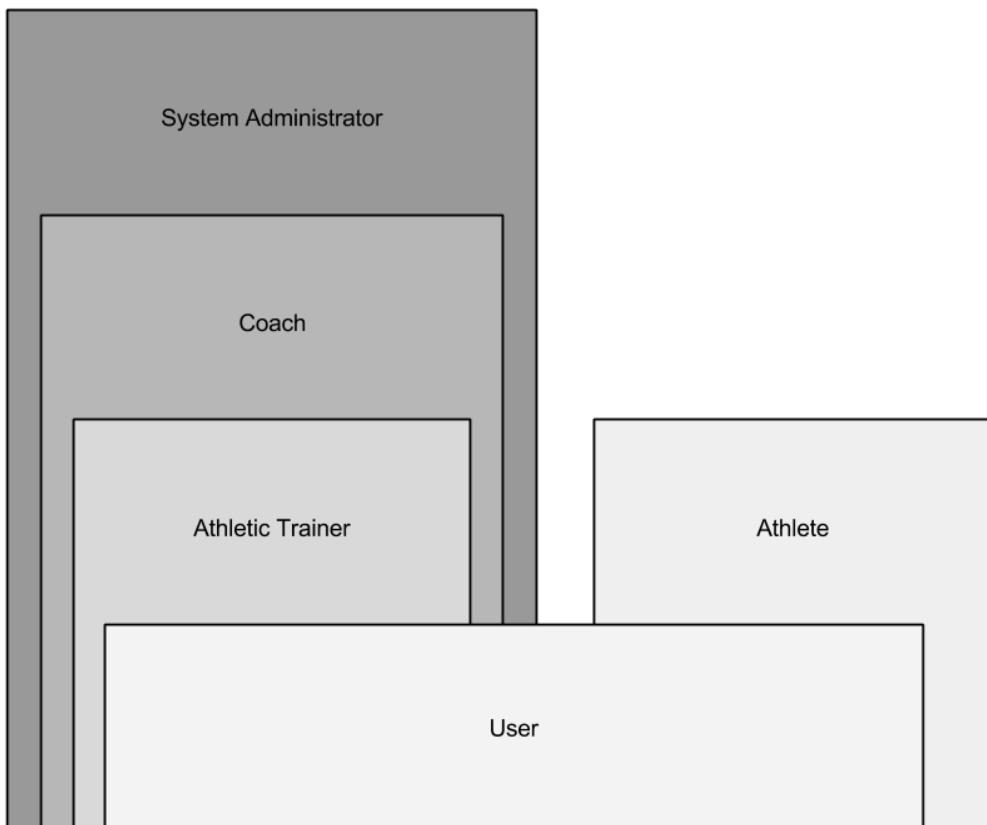
Athlete: The Athlete role will be any student athlete who is a member of a sport using this system to track their workouts. Athletes will be the only users who can add their own personal data to the system. In the case of viewing and analyzing data, Athletes are restricted to their own data.

Athletic Trainer: The Athletic Trainer will be given permission to deal with injured players. This will involve creation of special workouts, exercises and injury details. They will also be allowed to view roster information, previous workout data, nutritional data, and sleep data (for their particular sports/teams).

Coach: The Coach user represents the coach of a team whose athletes are interacting with the system. The Coach's main interaction with the system will involve creating and assigning workouts to his/her sport, as well as viewing and analyzing their athlete's submitted workout data. The Coach also has the abilities of the Athletic Trainer.

System Admin: The System Admin has control over permissions and account creation.

The System Admin will have unique access to all parts of the system to protect against users unknowingly deleting or creating false data.



3.0 Glossary

Attribute : A value describing an entity

Boolean : A data type with the a value of “true” or “false”

Cycles : A period of time in which workouts of the same nature are done

Database : A store of data

Entity : An object within the database, made up of attributes

Entity Relationships : A description of how entities interact with each other

Exercise : A physical activity to encourage growth in endurance and muscular ability in the body

Foreign Key : An attribute within an entity that refers to the primary key of a related entity

Functional Requirements : The expectations of the customer with regards to how the product should look and operate

Inactive : Is no longer accessible for use by a user, but is still stored in the database

Instance : A member of an entity

Interface : The way in which a user interacts with the system

Non-Functional Requirements : Requirements that further specify how the system should behave, not covered by the functional requirements

Nutrition Information : data filled out by the athlete specifying whether or not the athlete ate specific meals or snacks on certain days

Permissions : permissions restrict user interaction of the system in order to assure that users are only able to manage functions specific to their role

Personal Information : information about a specific user that includes name, email, and position.

Primary Key : An attribute that uniquely refers to an entity

Report : A compilation of data presented in a manner to show progression or analysis

Roster : a listing of athletes specific to a certain team or sport

Sleep Information : data filled out by the athlete specifying information about an athlete's sleep.

Sport : A competitive athletic activity comprised of a team or teams (ex. football, soccer, women's basketball, etc.)

Spreadsheet : A document made up of cells for the organization of data

Team : A group of Athletes with similar responsibilities (ex: offense, defensive line, forwards, etc.)

Wireframe : A blueprint of the system interface

Workout : a list of exercises created by the Coach to be done by the athletes.

Workout Data : Specific information with regards to how a workout and/or exercise is completed by an Athlete

4.0 Functional Requirements

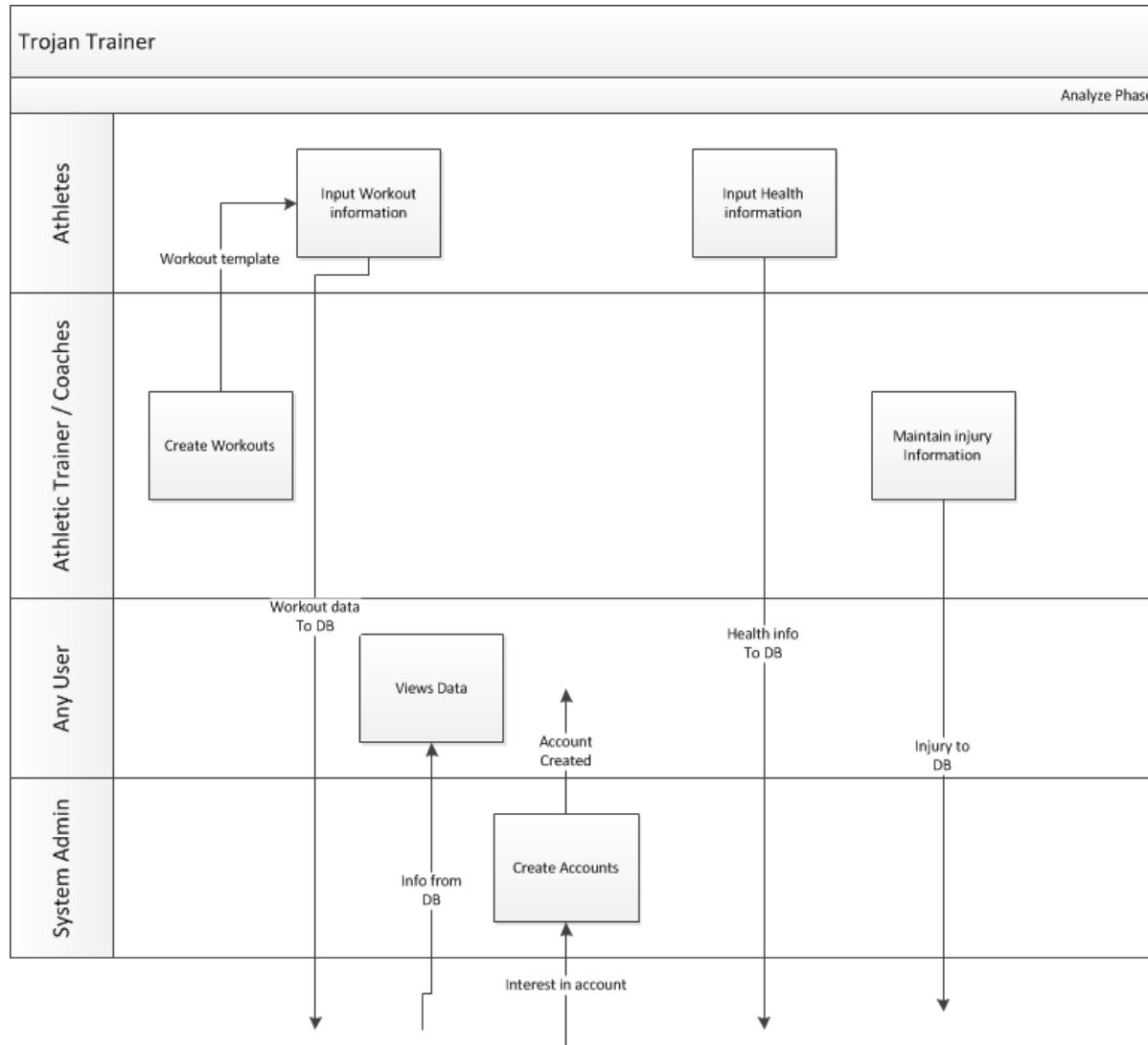
4.1 System Response Table

Evt.	Source	Trigger	Event Response	Major Outputs	External Dest.
<i>ID</i>	<i>Cause of event</i>	<i>Initiates action</i>	<i>From system after handling event</i>	<i>Created by handling event</i>	<i>For output</i>
U1	Any User	Desire to log in to the system	Session started	User is logged into their account	System
U2	Any User	Desire to log out of the system	Session terminated	User is logged out of the system	System
U3	Any User	Desire to view own personal information	Personal information is retrieved	Display of personal information	Any User
U4	Any User	Desire to view previous workout data	Previous workout data is retrieved	Display of previous workout data	Any User
U5	Any User	Desire to view current workout	Current workout is retrieved	Display of current workout	Any User
U6	Any User	Desire to view nutrition information	Nutrition information is retrieved	Display of nutrition information	Any User
U7	Any User	Desire to view sleep information	Sleep information is retrieved	Display of sleep information	Any User
U8	Any User	Desire to view weight information	Weight information retrieved	Display of weight information	Any User
U9	Any User	Desire to update own personal information	Personal information is updated	Notification of change	System Admin
				Updated personal information	Database
U10	Any User	Desire to view cycles	Cycles are retrieved	Display of cycles	Any User
U11	Any User	Desire to view report graphs	Report graphs are compiled	Display of report graphs	Any User
U12	Any User	Desire to view attendance	Attendance information retrieved	Display of attendance information	Any User
A1	Athlete	Desire to input workout data	Workout data recorded	Workout data submitted	Database
A2	Athlete	Desire to input nutrition information	Nutrition information recorded	Nutrition information submitted	Database

Evt.	Source	Trigger	Event Response	Major Outputs	Ext. Dest.
A3	Athlete	Desire to input weight information	Weight information recorded	Weight information submitted	Database
A4	Athlete	Desire to input sleep information	Sleep information recorded	Sleep information submitted	Database
AT1	Athletic Trainer	Desire to report an injury	Injury record created	Injury	Database
AT2	Athletic Trainer	Desire to create an exercise	Exercise record created	Exercise	Database
AT3	Athletic Trainer	Desire to create a workout	Workout record created	Workout	Database
				Notification of creation	Coach
AT4	Athletic Trainer	Desire to view injured athletes	Injuries and athletes are retrieved	Display of injuries and athletes	Athletic Trainer
AT5	Athletic Trainer	Desire to update injuries of athletes	Athlete's injuries are updated	Notification of update	Coach
				Updated athlete's injuries	Database
AT6	Athletic Trainer	Desire to update workouts	Workout is updated	Updated workout	Database
				Notification of update	Coach
AT7	Athletic Trainer	Desire to update exercises	Exercise is updated	Updated exercise	Database
AT8	Athletic Trainer	Desire to assign a workout	Workout is assigned	Notification of assignment	Coach
				Updated current workout	Database
AT9	Athletic Trainer	Desire to remove injury from player	Injury is removed	Deactivation	Database
AT10	Athletic Trainer	Desire to copy a workout	Workout is copied	Notification of creation	Coach
AT11	Athletic Trainer	Desire to make a workout inactive	Workout becomes inactive	Inactivation	Database
AT12	Athletic Trainer	Desire to view sport & team roster	Rosters for sports and teams are retrieved	Display of rosters for sports and teams	Athletic Trainer
CO1	Coach	Desire to create a team	Team record created	Team	Database
CO2	Coach	Desire to update sport or team roster	Sport or team roster is updated	Updated sport or team roster	Database
SA1	System Admin	Desire to create an account	Account is created	User account	Database
				Email notification	User
SA2	System Admin	Desire to create a sport	Sport record created	Sport	Database
SA3	System Admin	Desire to view permissions	Permissions are retrieved	Display of permissions	System Admin

Evt.	Source	Trigger	Event Response	Major Outputs	Ext. Dest.
SA4	System Admin	Desire to view list of all sports	Sports are retrieved	Display of sports	System Admin
SA5	System Admin	Desire to set permissions	Permissions are updated	Updated permissions	Database
SA6	System Admin	Desire to make an account inactive	Account becomes inactive	Deactivation	Database
SA7	System Admin	Desire to make a sport inactive	Sport becomes inactive	Deactivation	Database

4.2 Cross Functional Process Map



4.3 Use Cases

Name	Log In	ID	U1
Primary Actor	Any User		
Other Actors	N/A		
Description	User logs in to system.		
Preconditions	Must have an account		
Trigger	Desire to log in to system		
	Actor Actions	System Response	
Typical Event Flow	User clicks "Log In"		
		System displays boxes for username and password	
	User fills in the username and password		
	User clicks "Submit"		
		System checks for valid username and password combination	
		System takes user to the appropriate home page	
Alternate Event Flow	User clicks "Log In"		
		System displays boxes for username and password	
	User fills in the username and password incorrectly		
	User clicks "Submit"		
		System checks for valid username and password combination	
		System finds combination invalid and takes user back to the Log In page	
Conclusion	User is logged in		
Postconditions	User can now interact with the system		

Name	Log Out	ID	U2		
Primary Actor	Any User				
Other Actors	N/A				
Description	User logs out of the system to and ceases interaction				
Preconditions	User must be logged in				
Trigger	Desire to log out of the system				
	Actor Action	System Response			
Typical Event Flow	User clicks "Log Out"				
		System displays confirmation pop-up			
	User clicks "Log Out"				
		System logs user out of account			
Alternate Event Flows	User clicks "Log Out"				
		System displays confirmation pop-up			
	User clicks "Cancel"				
		System aborts log out procedure			
Conclusion	User is logged out of system				
Postconditions	User can no longer interact with the system				

Name	Viewing Personal Information	ID	U3
Primary Actor	Any User		
Other Actors	N/A		
Description	Any user views own personal information		
Preconditions	Must be logged in		
Trigger	Desire to view own personal information		
	Actor Action	System Response	
Typical Event Flow	User clicks own name		
		System displays personal information	
Alternate Event Flows			
Conclusion	User views own personal information.		
Postconditions	N/A		

Name	Viewing Data	ID	U4
Primary Actor	Any User		
Other Actors	N/A		
Description	User view previous workout information		
Preconditions	Must have previous workout data to view		
Trigger	Desire to view previous workout data		
	Actor Action	System Response	
Typical Event Flow	User clicks "View Data"		
		System takes user to the View Data page	
	User fills in the search form		
	User clicks "Search"		
		System checks that request is valid	
		System displays Athlete's workout information	
Alternate Event Flows	Athlete goes to workout page		
	Athlete clicks desired week		
		System displays previous workout data	
Alternate Event Flows	User clicks "View Data"		
		System takes user to the View Data page	
	User fills in the search form		
	User clicks "Search"		
		System checks that request is valid	
		System finds request invalid and sends User back to search page	
Conclusion	User has seen previous workout information		
Postconditions	N/A		

Name	View current assigned workout	ID	U5
Primary Actor	Any User		
Other Actors	N/A		
Description	User views the workout that is currently assigned		
Preconditions	Workout must be assigned, must be at Home page		
Trigger	Desire to view current workout		
	Actor Action	System Response	
Typical Event Flow	User clicks "Workout"		
		System displays Current Workout page	
Alternate Event Flows	User clicks desired cycle		
		System displays workouts from that cycle	
Conclusion	Athlete views current workout on home page		
Postconditions	N/A		

Name	Viewing Nutrition Information	ID	U6
Primary Actor	Any User		
Other Actors	N/A		
Description	User views nutrition information.		
Preconditions	N/A		
Trigger	Desire to view nutrition information		
Typical Event Flow	Actor Action	System Response	
Typical Event Flow	User clicks "View Data" page		
		System displays View Data page	
	User clicks "Health"		
		System takes user to a search page	
	User fills in search form		
	User clicks "Search"		
		System checks that request is valid	
		System displays search results health information	
Alternate Event Flows	Athlete clicks "Sleep/Nutrition"		
		System displays Sleep/Nutrition data	
Alternate Event Flows	User clicks "View Data"		
Alternate Event Flows		System takes user to a search page	
	User fills in search form		
	User clicks "Search"		
		System checks that request is valid	
		System finds request invalid and takes User back to search page	
Conclusion	User viewed nutrition information		
Postconditions	System is unchanged		

Name	Viewing Sleep information	ID	U7
Primary Actor	Any User		
Other Actors	N/A		
Description	User views sleep information		
Preconditions	N/A		
Trigger	Desire to view sleep information		
	Actor Action	System Response	
Typical Event Flow	User clicks "View Data"		
		System takes user to the View Data page	
	User clicks "Health"		
		System takes user to a search page	
	User types Athlete's name into the search box		
	User clicks "Search"		
		System checks that request is valid	
		System displays Athlete's health information	
Alternate Event Flows	Athlete clicks "Sleep/Nutrition"		
		System displays Sleep/Nutrition data	
Alternate Event Flows	User clicks "View Data"		
		System takes user to the View Data page	
	User clicks "Health"		
		System takes user to a search page	
	User fills in search form		
	User clicks "Search"		
		System checks that request is valid	
		System finds request invalid and sends User back to search page	
Conclusion	User viewed sleep information		
Postconditions	System is unchanged		

Name	Viewing Weight Information	ID	U8
Primary Actor	Any User		
Other Actors	N/A		
Description	User views weight information		
Preconditions	N/A		
Trigger	Desire to view weight information		
Actor Action	System Response		
Typical Event Flow	User clicks "View Data"		
		System takes user to the View Data page	
	User clicks "Health"		
		System takes user to a search page	
	User types Athlete's name into the search box		
	User clicks "Search"		
		System checks that request is valid	
		System displays Athlete's health information	
Alternate Event Flows	Athlete clicks "Sleep/Nutrition"		
		System displays Sleep/Nutrition data	
Alternate Event Flows	User clicks "View Data"		
		System takes user to the View Data page	
	User clicks "Health"		
		System takes user to a search page	
	User types Athlete's name into the search box		
	User clicks "Search"		
		System finds request invalid and sends User back to search page	
Conclusion	User viewed the weight information		
Postconditions	System is unchanged.		

Name	Updating Personal Information	ID	U9
Primary Actor	Any User		
Other Actors	N/A		
Description	User updates own personal information		
Preconditions	Must be viewing personal information		
Trigger	Desire to update personal information		
	Actor Action	System Response	
Typical	User fills in form		
	User clicks "Save Changes"		
		System check for valid inputs	
		System updates the database	
		System sends notification of update to System Admin	
Alternate Event Flows	User fills in form		
	User clicks "Save Changes"		
		System checks for valid inputs	
		System finds inputs invalid and displays an error message	
Conclusion	User has changed own personal information		
Postconditions	Database has been updated		

Name	Viewing Cycle	ID	U10
Primary Actor	Any User		
Other Actors	N/A		
Description	User views workout cycles		
Preconditions	N/A		
Trigger	Desire to view cycles		
	Actor Action	System Response	
Typical Event Flow	User clicks "Workout"		
		System displays Current Workout page	
	User clicks desired cycle		
		System displays workouts from that cycle	
Alternate Event Flows	Athlete views current workout on home page		
Conclusion	Cycles are viewed		
Postconditions	N/A		

Name	Viewing Report Graphs	ID	U11
Primary Actor	Any User		
Other Actors	N/A		
Description	User views graphs from reports		
Preconditions	N/A		
Trigger	Desire to view report graphs		
Typical Event Flow	Actor Action	System Response	
Typical Event Flow	User clicks "View Data"		
		System displays View Data page	
	User clicks "Graphs"		
		System takes user to a search page	
	User fills out search request		
	User clicks "Search"		
		System checks that request is valid	
		System displays report's graphs	
Alternate Event Flows	Athlete goes to workout page		
	Athlete clicks "Look at Graphs"		
		System displays search form	
	Athlete fills in search form and clicks "Graph"		
		System displays graph	
Alternate Event Flows	User clicks "View Data"		
		System displays View Data page	
	User clicks "Graphs"		
		System takes user to a search page	
	User types Athlete's name into the search box		
	User clicks "Search"		
		System checks that request is valid	
		System finds request invalid and sends User back to search page	
Conclusion	User viewed graphs		
Postconditions	System is unchanged		

Name	Viewing Attendance	ID	U12
Primary Actor	Any User		
Other Actors	N/A		
Description	User views attendance record		
Preconditions	N/A		
Trigger	Desire to view attendance		
	Actor Action	System Response	
Typical Event Flow	User clicks "View Data"		
		System displays View Data page	
	User clicks "Attendance"		
		System takes user to a search page	
	User fills out search request		
	User clicks "Search"		
		System checks that request is valid	
		System displays attendance record	
Alternate Event Flows	Athlete goes to workout page		
	Athlete clicks desired weeks		
		System displays workouts	
Alternate Event Flows	User clicks "View Data"		
		System displays View Data page	
	User clicks "Attendance"		
		System takes user to a search page	
	User fills out search request		
	User clicks "Search"		
		System finds request invalid and sends User back to search page	
Conclusion	User viewed attendance information		
Postconditions	System is unchanged		

Name	Athlete Inputs Workout Data	ID	A1
Primary Actor	Athlete		
Other Actors	N/A		
Description	Athlete submits the data for the current workout		
Preconditions	Must be on Athlete's Workout page		
Trigger	Desire to input workout data		
	Actor Action	System Response	
Typical Event Flow	Athlete fills in form to submit workout data		
		System sends data to the database	
Alternate Event Flows			
Conclusion	Workout data is recorded		
Postconditions	Data is in the database and can be viewed by the athlete and the coach		

Name	Athlete Inputs Nutrition Information	ID	A2
Primary Actor	Athlete		
Other Actors	N/A		
Description	Athlete submits own nutrition information		
Preconditions	Must be on Athlete's Workout page		
Trigger	Desire to input nutrition information		
	Actor Action	System Response	
Typical Event Flow	Athlete clicks "Sleep/Nutrition"		
		System displays form for nutrition information	
Athlete fills in form			
		System sends nutrition information to the database	
Alternate Event Flows			
Conclusion	Athlete has submitted nutrition information		
Postconditions	Database has been updated		

Name	Athlete Inputs Weight Information	ID	A3
Primary Actor	Athlete		
Other Actors	N/A		
Description	Athlete submits own weight information		
Preconditions	Must be on Athlete's Workout page		
Trigger	Desire to input sleep information		
Typical Event Flow	Actor Action	System Response	
	Athlete clicks "Sleep/Nutrition"		
		System displays form for weight information	
	Athlete fills in form		
		System sends weight information to the database	
Alternate Event Flows			
Conclusion	Athlete has submitted weight information		
Postconditions	Database has been updated		

Name	Athlete Inputs Sleep Information	ID	A4
Primary Actor	Athlete		
Other Actors	N/A		
Description	Athlete submits own sleep information		
Preconditions	Must be on Athlete's Workout page		
Trigger	Desire to input sleep information		
Typical Event Flow	Actor Action	System Response	
	Athlete clicks "Sleep/Nutrition"		
		System displays form for sleep information	
	Athlete fills in form		
		System sends sleep information to the database	
Alternate Event Flows			
Conclusion	Athlete has submitted sleep information		
Postconditions	Database has been updated		

Name	Report Injury	ID	AT1
Primary Actor	Athletic Trainer		
Other Actors	Coach, System Admin		
Description	An injury is added to the database		
Preconditions	N/A		
Trigger	Desire to report an injury		
Typical Event Flow	Actor Action	System Response	
	Athletic Trainer clicks "Injury"		
		System displays form for injury report	
	Athletic Trainer fills in form and submits it		
		System checks for form completion	
		System sends submission to database	
Alternate Event Flows	Athletic Trainer clicks "Injury"		
		System displays form for injury creation	
	Athletic Trainer submits form without completing it		
		System checks for form completion and finds error	
		System displays an error message	
Conclusion	Athletic Trainer has reported an injury		
Postconditions	Database is updated		

Name	Create Exercise	ID	AT2
Primary Actor	Athletic Trainer		
Other Actors	Coach, System Admin		
Description	Athletic Trainer creates an exercise and adds it to the database		
Preconditions	Must be on workout page		
Trigger	Desire to create an exercise		
Typical Event Flow	Actor Action	System Response	
	Athletic Trainer creates an exercise by interacting with the spreadsheet		
	Athletic Trainer clicks the save button		
		System adds exercise to the database	
Alternate Event Flows			
Conclusion	Athletic Trainer has created an exercise		
Postconditions	Database is updated		

Name	Create Workout	ID	AT3		
Primary Actor	Athletic Trainer				
Other Actors	Coach, System Admin				
Description	Athletic Trainer creates a new workout				
Preconditions	Must be on Workout page				
Trigger	Desire to create a new workout				
Typical Event Flow	Actor Action	System Response			
	Athletic Trainer clicks "Add"				
		System displays workout creation form.			
	Athletic Trainer fills out form and clicks "Save"				
		System displays blank workout template.			
	Athletic Trainer creates workout by interacting with the spreadsheet				
	Athletic Trainer clicks the save button				
Alternate Event Flows		System adds workout to the database			
		System sends notification to Athletic Trainer and/or Coach			
Conclusion	Workout is created				
Postconditions	Workout is added to the database				

Name	Viewing Injuries	ID	AT4
Primary Actor	Athletic Trainer		
Other Actors	Coach, System Admin		
Description	Athletic Trainer views a list of injured athletes		
Preconditions	N/A		
Trigger	Desire to view injured athletes		
	Actor Action	System Response	
Typical Event Flow	Athletic Trainer clicks "Injuries"		
		System displays injury page	
Alternate Event Flows			
Conclusion	List of injured athletes has been viewed by Athletic Trainer		
Postconditions	N/A		
Name	Update Injuries	ID	AT5
Primary Actor	Athletic Trainer		
Other Actors	Coach, System Admin		
Description	Injuries are updated		
Preconditions	Must be on Injuries page		
Trigger	Desire to update injuries of Athletes		
	Actor Action	System Response	
Typical Event Flow	Athletic Trainer updates injuries by interacting with the table		
	Athletic Trainer clicks the save button		
		System updates injuries in the database	
		System sends notification to Coach	
Alternate Event Flows			
Conclusion	Athletic injuries are updated		
Postconditions	Database is updated		

Name	Update Workouts	ID	AT6		
Primary Actor	Athletic Trainer				
Other Actors	Coach, System Admin				
Description	Workouts are updated				
Preconditions	Must be on Workouts page				
Trigger	Desire to update workouts				
	Actor Action	System Response			
Typical Event Flow	Athletic Trainer updates workouts by interacting with the spreadsheet				
	Athletic Trainer clicks the save button				
		System updates workout in the database			
		System sends notification to Coach			
Alternate Event Flows					
Conclusion	Workouts are updated				
Postconditions	Database is updated				

Name	Update Exercises	ID	AT7		
Primary Actor	Athletic Trainer				
Other Actors	Coach, System Admin				
Description	Exercises are updated				
Preconditions	Must be on Workout page				
Trigger	Desire to update exercises				
	Actor Action	System Response			
Typical Event Flow	Athletic Trainer updates exercises by interacting with the spreadsheet				
	Athletic Trainer clicks the save button				
		System updates the workout in the database			
Alternate Event Flows					
Conclusion	Exercises are updated				
Postconditions	Database is updated				

Name	Assign Workout	ID	AT8
Primary Actor	Athletic Trainer		
Other Actors	Coach, System Admin		
Description	Athletic Trainer assigns a workout for the Athletes to complete		
Preconditions	Must be on Workout page		
Trigger	Desire to assign a workout		
Typical Event Flow	Actor Action	System Response	
	Athletic Trainer clicks "Assign Workout"		
		System displays form for assignment	
	Athletic Trainer fills in the form		
	Athletic Trainer clicks "Assign"		
		System updates the database	
		System sends notification to Coach	
Alternate Event Flows			
Conclusion	Workout is assigned		
Postconditions	Database is updated		

Name	Remove Injury	ID	AT9
Primary Actor	Athletic Trainer		
Other Actors	Coach, System Admin		
Description	An Athlete's injury is deactivated		
Preconditions	Must be on Injuries page		
Trigger	Desire to make an injury inactive		
Typical Event Flow	Actor Action	System Response	
	Athletic Trainer clicks "Field"		
	Athletic Trainer clicks "X" next to Athlete's name		
		System removes injury from athlete	
		System sends notification to Coach	
Alternate Event Flows			
Conclusion	Injury is removed from Athlete		
Postconditions	Database is updated		

Name	Copy Workout	ID	AT10		
Primary Actor	Athletic Trainer				
Other Actors	Coach, System Admin				
Description	Athletic Trainer copies a workout to another day				
Preconditions	Must be on Workout home page				
Trigger	Desire to copy a workout				
	Actor Action	System Response			
Typical Event Flow	Athletic Trainer clicks and holds day tab				
	Athletic Trainer drags original day over new day				
		System adds workouts from original day to new day			
	Athletic Trainer clicks the save button				
		System saves workout for new day to the database			
Alternate Event Flows	Athletic Trainer clicks and holds day tab				
	Athletic Trainer drags original day over new cycle tab				
		System switches spreadsheet to new cycle			
	Athletic Trainer continues to drag original day over new day				
		System adds workouts from original day to new day			
	Athletic Trainer clicks the save button				
		System saves workout for new day to the database			
Conclusion	Workout is copied				
Postconditions	Database has been updated				

Name	Make Workout Inactive	ID	AT11
Primary Actor	Athletic Trainer		
Other Actors	Coach, System Admin		
Description	A workout is made inactive		
Preconditions	Must be on Workout page		
Trigger	Desire to make a workout inactive		
	Actor Action	System Response	
Typical Event Flow	Athletic Trainer clicks "X" in workout tab		
		System displays confirmation pop-up	
	Athletic Trainer clicks "Yes"		
		System updates the workout in the database	
Alternate Event Flows	Athletic Trainer clicks "X" in workout tab		
		System displays confirmation pop-up	
	Athletic Trainer clicks "Yes"		
		System updates the workout in the database	
Conclusion	Workout is no longer active		
Postconditions	Database has been updated		

Name	Viewing Rosters	ID	AT12
Primary Actor	Any User		
Other Actors	N/A		
Description	User views rosters for teams and sports		
Preconditions	Must be on home page		
Trigger	Desire to views roster for a team or a sport		
	Actor Action	System Response	
Typical Event Flow	User clicks "Rosters"		
		System displays Roster page	
Alternate Event Flows			
Conclusion	User viewed roster for a team or sport		
Postconditions	N/A		

Name	Create Team	ID	CO1
Primary Actor	Coach		
Other Actors	System Admin		
Description	A team is created within a sport		
Preconditions	N/A		
Trigger	Desire to create a team		
Typical Event Flow	Actor Action	System Response	
	Coach clicks "Roster"		
		System displays spreadsheet for team roster	
	Coach clicks team field		
	Coach enters desired team name		
		System checks that key cells have been filled out	
		System adds team to the database	
Alternate Event Flows	Coach clicks "Roster"		
		System displays spreadsheet for team creation	
	Coach clicks team field		
	Coach enters desired team name		
		System checks that key cells have been filled out	
		System finds missing values and displays an error message	
Conclusion	Team is created		
Postconditions	Database is updated		

Name	Update Rosters	ID	CO2
Primary Actor	Coach		
Other Actors	System Admin		
Description	A team or sport roster is updated		
Preconditions	Must be on Roster page		
Trigger	Desire to update sport or team roster		
Typical Event Flow	Actor Action	System Response	
	Coach updates rosters by interacting with the table		
	Coach clicks the save button		
		System updates the database	
Alternate Event Flows			
Conclusion	Roster is updated		
Postconditions	Database is updated		

Name	Account Creation	ID	SA1
Primary Actor	System Admin		
Other Actors	N/A		
Description	System Admin creates an account for a user		
Preconditions	Must be on Accounts page		
Trigger	Desire to create an account		
	Actor Action	System Response	
Typical Event Flow	System Admin clicks "Create Account"		
		System displays form for account creation	
	System Admin fills in form		
	System Admin clicks "Save"		
		System checks that form is complete	
		System adds account to the database	
Alternate Event Flows	System Admin clicks "Create Account"		
		System displays form for account creation	
	System Admin fills in form incompletely		
	System Admin clicks "Save"		
		System checks for form completion	
		System finds form is incomplete and displays an error message	
Conclusion	Account is created		
Postconditions	Database is updated		

Name	Sport Creation	ID	SA2		
Primary Actor	System Admin				
Other Actors	N/A				
Description	System Admin creates a sport				
Preconditions	Must be of Sports page				
Trigger	Desire to create a sport				
	Actor Action	System Response			
Typical Event Flow	System Admin clicks "Add/Remove"				
		System displays form for sport creation			
	System Admin fills in form				
	System Admin clicks "Save"				
		System checks for form completion			
Alternate Event Flows		System adds sport to the database			
	System Admin clicks "Add/Remove"				
		System displays form for sport creation			
	System Admin fills in form				
	System Admin clicks "Save"				
Conclusion		System checks for form completion			
		System finds form incomplete and displays an error message			
Postconditions	A sport is created				
	Database is updated				

Name	Viewing Permissions	ID	SA3
Primary Actor	System Admin		
Other Actors	N/A		
Description	System Admin views permissions		
Preconditions			
Trigger	Desire to view permissions		
	Actor Action	System Response	
Typical Event Flow	System Admin clicks "Accounts"		
		System displays account information page	
	System Admin clicks field		
	System Admin clicks an arrow next an accounts information		
		System displays permission edit page	
Alternate Event Flows			
Conclusion	Permissions are viewed		
Postconditions	N/A		

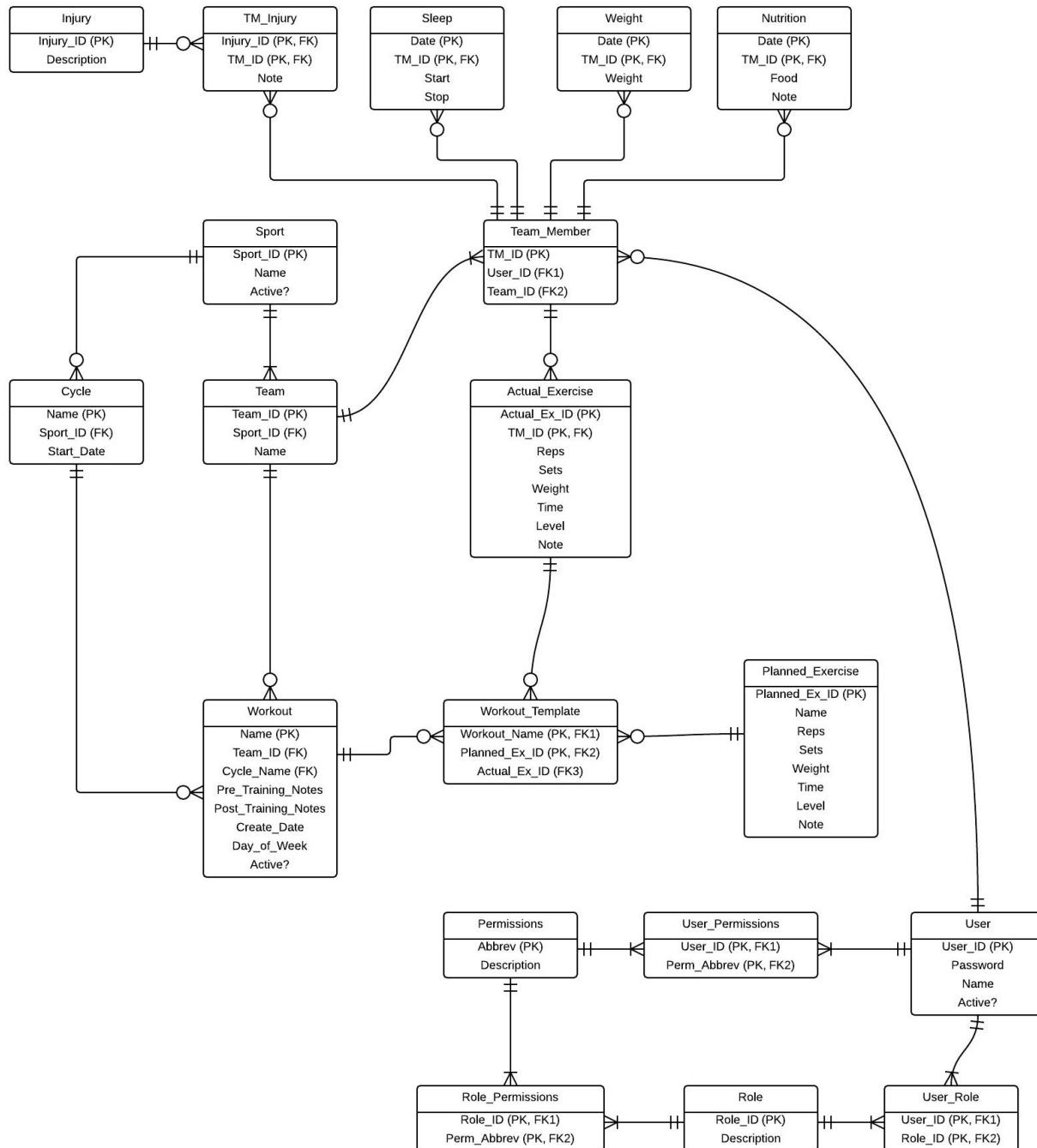
Name	Viewing Sports List	ID	SA4
Primary Actor	System Admin		
Other Actors	N/A		
Description	System Admin views a list of sports		
Preconditions			
Trigger	Desire to view a list of sports		
	Actor Action	System Response	
Typical Event Flow	System Admin clicks "Sports"		
		System displays sports page with list of sports	
Alternate Event Flows			
Conclusion	List of sports is viewed		
Postconditions	N/A		

Name	Set Permissions	ID	SA5
Primary Actor	System Admin		
Other Actors	N/A		
Description	System Admin sets permissions for a role		
Preconditions	Must be on Permissions page		
Trigger	Desire to set permissions		
Typical Event Flow	Actor Action	System Response	
	System Admin clicks "Accounts"		
			System displays account information page
	System Admin clicks field		
	System Admin clicks an arrow next to an accounts information		
			System displays permission edit page
	System Admin updates permissions		
	System Admin clicks the save button		
			System updates the database
			System displays account information page
Alternate Event Flows			
Conclusion	Permissions are set		
Postconditions	Database is updated		

Name	Making Account Inactive	ID	SA6		
Primary Actor	System Admin				
Other Actors	N/A				
Description	System Admin makes a user's account inactive				
Preconditions	Must be on Accounts page				
Trigger	Desire to make an account inactive				
	Actor Action	System Response			
Typical Event Flow	System Admin clicks "X" next to account				
		System displays confirmation pop-up			
	System Admin selects "Yes"				
		System makes accounts inactive in the database			
Alternate Event Flows	System Admin clicks "X" next to account				
		System displays confirmation pop-up			
	System Admin selects "No"				
		System aborts event			
Conclusion	Account is inactive				
Postconditions	Database is updated				

Name	Making Sport Inactive	ID	SA7
Primary Actor	System Admin		
Other Actors	N/A		
Description	System Admin makes a sport inactive		
Preconditions	Must be on Sports page		
Trigger	Desire to make an account inactive		
	Actor Action	System Response	
Typical Event Flow	System Admin "Add/Remove"		
		System displays sports edit page	
	System Admin clicks "X" next to sport		
		System displays updated sports edit page	
	System Admin clicks "Confirm"		System updates the database
Alternate Event Flows	System Admin "Add/Remove"		
		System displays sports edit page	
	System Admin clicks "X" next to sport		
		System displays updated sports edit page	
	System Admin clicks "Cancel"		System aborts event
Conclusion	Sport is inactive		
Postconditions	Database is updated		

4.4 Logical Data Model



4.5 Data Dictionary

This data dictionary contains definitions and clarifications for entities, attributes, and relationships found in the Logical Data Model. All non-obvious information is listed below, while assumed obvious information has been left out of this section.

- **Actual_Exercise:** This entity contains the workout data submitted by an athlete. In other words, this is where we store what the athlete “actually” did.
 - Attributes
 - Time: Time refers to an amount of time the exercise needs to be performed for. This attribute will usually not apply for weight lifting, but rather for cardio and fitness oriented exercises. Examples include time on an elliptical, or time on a stationary bike.
 - Level: Level refers to an intensity level in a cardio or fitness oriented exercise. In other words, the level is the degree of difficulty the machine is set to. Examples include the intensity level of an elliptical or a stationary bike.
 - Note: This attribute is available for information that might be necessary, but not consistent enough to give a more specific name. For example, an athlete might need to make a note that a specific exercise was painful, although he/she completed it. Or, an athlete might need to explain he/she was unclear on how to perform the exercise, although still completed it.
 - Relationships
 - One-to-many with **Team Member**
 - Many team members will record data for a specific exercise, while data for a specific exercise will only apply to one team member.
 - One-to-many with **Workout_Template**
 - Actual exercise data can apply to many workouts, while workouts will have many actual exercise data. The **Workout_Template** entity resolves this many-to-many relationship.
- **Cycle:** A cycle is an important business principle that needs to be represented in the database. The cycle entity makes it possible for workouts to be sorted and stored by cycle.
 - Relationships
 - One-to-many with **Sport**, and **Workout**
 - A sport will have as few as 0 cycles, but as many as “many”. In the opposite direction, a cycle will apply to one and only one sport. In the same manner, a cycle will have as few as 0 workouts, but as many as “many”. In the opposite direction, a workout will be a part of only one

cycle. The idea behind organizing by cycle is that specific workouts will be grouped together into cycles, therefore many need to be a part of a given cycle.

- **Permissions:** Refers to level of access users have within the system. Different permissions will be given to each user, on a specific level or based on their role. By assigning permissions separate from role, variation and customization is possible.
- **Planned_Exercise:** This entity contains all the data for a specific exercise. It is what the coach desires to be done in that particular exercise.
 - Attributes
 - Active?: This attribute is a boolean value that refers to whether or not the exercise is still being used by the coach. When set to false, the exercise will not appear to the coach for choosing.
 - Relationships
 - One-to-many with **Workout_Template**
 - Planned exercises can be used in many workouts, while workouts will use many planned exercises. The **Workout_Template** entity resolves this many-to-many relationship, and therefore has a one-to-many relationship with **Planned_Exercise**.
- **Role:** The role entity specifies what role a user has in the sport. Examples are athlete, coach, or athletic trainer.
- **Team:** The team entity refers to subgroups of a sport. In many sports, it is common to be divided into smaller groups. For example, in football, basic common teams are offense, defense, and special teams. The Team entity allows for making groups within the sport.
 - Relationships
 - One-to-many with **Sport**
 - A sport will have as few as one team, but as many as “many”. In the opposite direction, a team will apply to one and only one sport. In cases where sports do not divide into smaller teams, an initial team will exist containing every member.
- **Team_Member:** This is a crucial entity that connects a user to the rest of the data the system stores. In our model, a team member could be an athlete, a coach, or an athletic trainer. The user’s access to specific information will be guided by the permissions they have been assigned.
 - Attributes
 - TM_ID: This is the unique ID for the team member.
 - Relationships
 - One-to-many with **TM_Injury, Sleep, Weight, Nutrition**

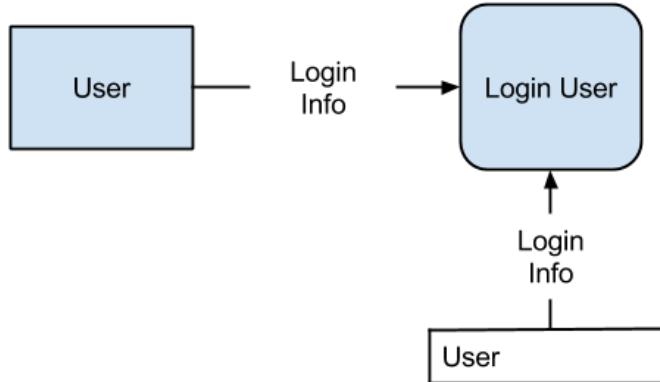
- Each team member can have as few as 0, but up to “many” injuries, sleep information entries, weight information entries, or nutrition information entries. In the opposite direction, each of those entries can only apply to one specific team member.
- One-to-many with **Actual_Exercise**
 - Each team member can complete as few as 0, but up to “many” exercises. This exercise information is stored in the actual_exercise entity. In the opposite direction, each actual exercise applies to only one team member.
- One-to-many with **User**
 - This relationship allows a user to be a member of multiple teams. Examples might include an athlete playing for multiple teams, a coach coaching multiple teams, or an athletic trainers working for multiple teams. So, a user can be a member of multiple teams, while a team member only applies to one specific user.
- **TM_Injury:** This is an associative entity between Team_Member and Injury. This entity allows a team member to be paired with a specific injury, or multiple injuries, if they are injured.
 - Relationships
 - One-to-many with **Injury**, and **Team_Member**
 - Due to this entity being an associative entity, it will take one and only one team member and injury. In the opposite direction, a team member or an injury can be used as few as 0 times, but up to “many” times in TM_Injury entities.
- **Workout:** The workout entity is the basic information about a given workout.
 - Attributes
 - Create_Date: The create date stores exactly when this workout was originally created.
 - Day_of_Week: This attribute stores what day of the week the workout will be completed on. This attribute was created for use in assigning workouts during cycles, as they are usually assigned on a consistent weekly basis.
 - Relationships
 - One-to-many with **Cycle**, and **Team**
 - Each workout will belong to a specific cycle and a specific team. However, a cycle and a team can have as few as 0 workouts, but as many as “many” workouts.
 - One-to-many with **Workout_Template**

- A workout will have many exercises, and therefore many workout templates. In the opposite direction, a workout template will only correspond to one specific workout.
- **Workout_Template:** This entity refers to planned exercise information, the corresponding actual exercise information, as well as the workout to which they belong.
 - Relationships
 - One-to-many with **Workout**, **Planned_Exercise**, and **Actual_Exercise**
 - This is really a ternary relationship that combines these three entities, as they all work together. This entity and its relationships solve many-to-many relationships between the three entities. A workout will have many actual and planned exercises, while each actual and planned exercise can correspond to many workouts.

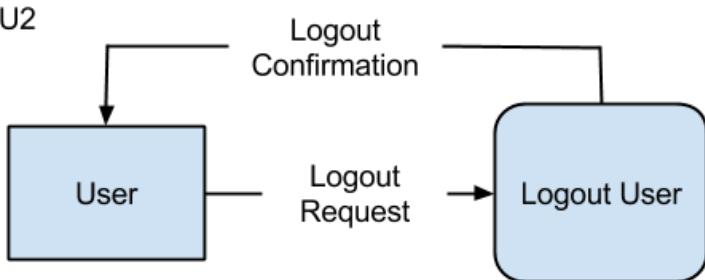
4.6 Logical Process Model

4.6.1 Data Flow Diagrams

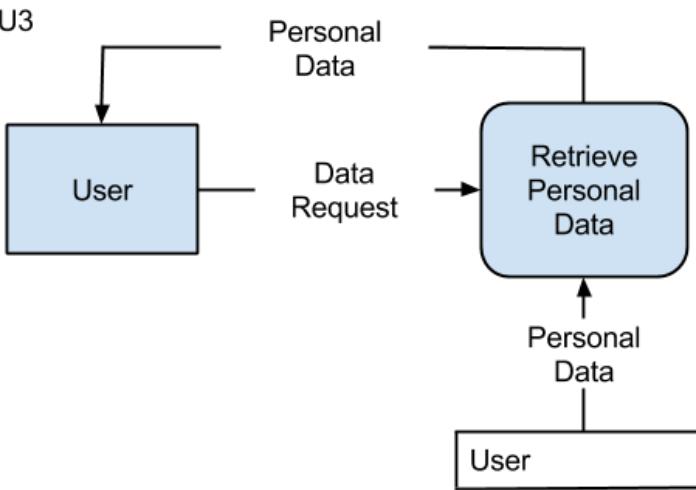
U1



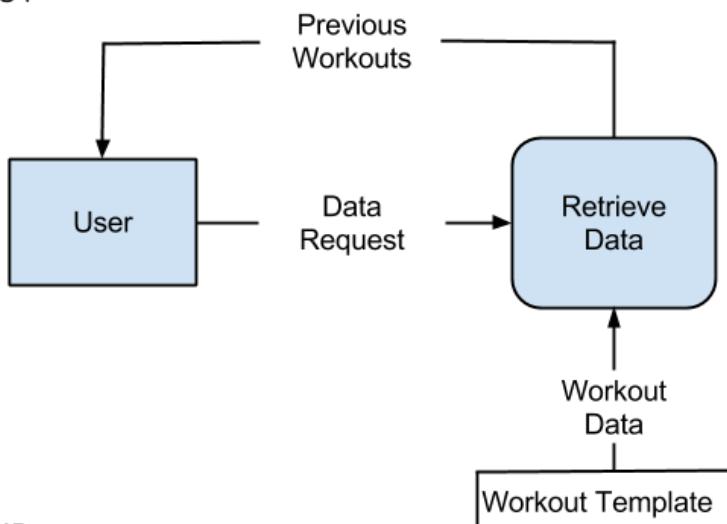
U2



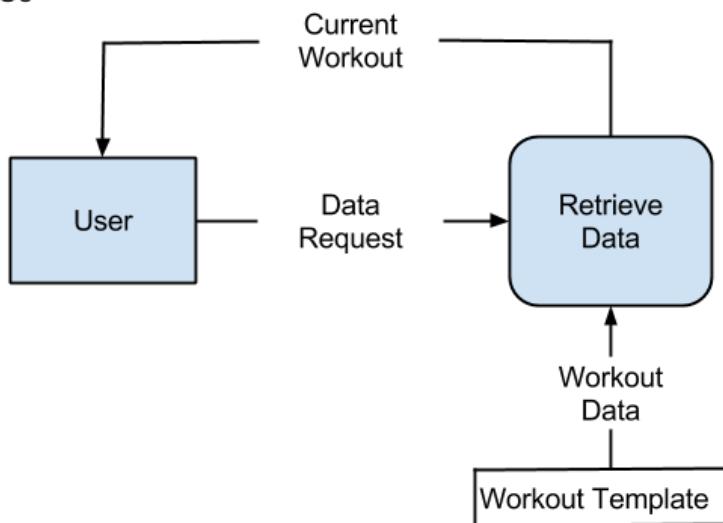
U3



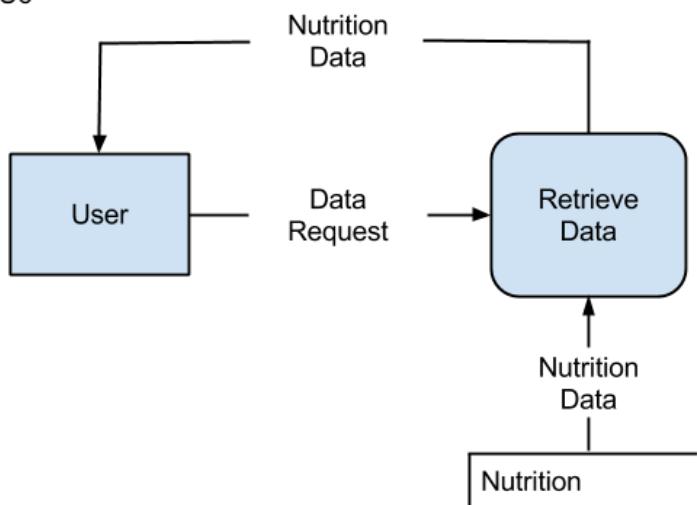
U4



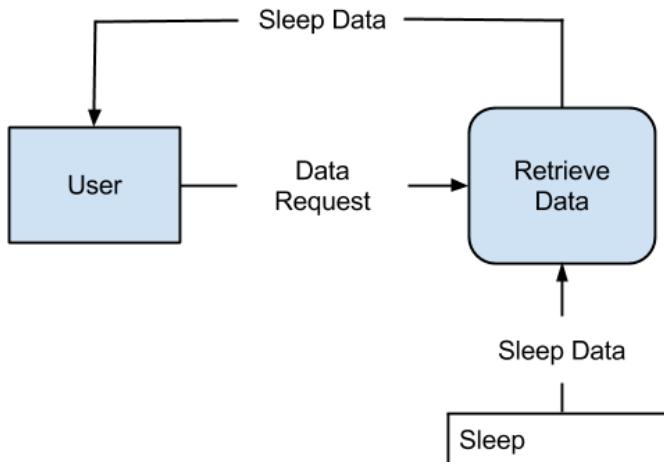
U5



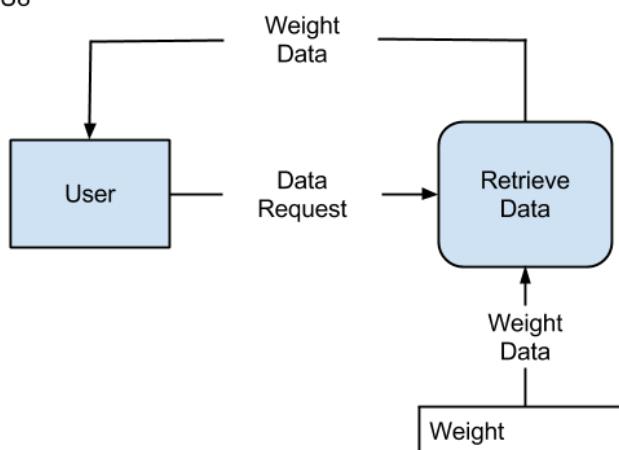
U6



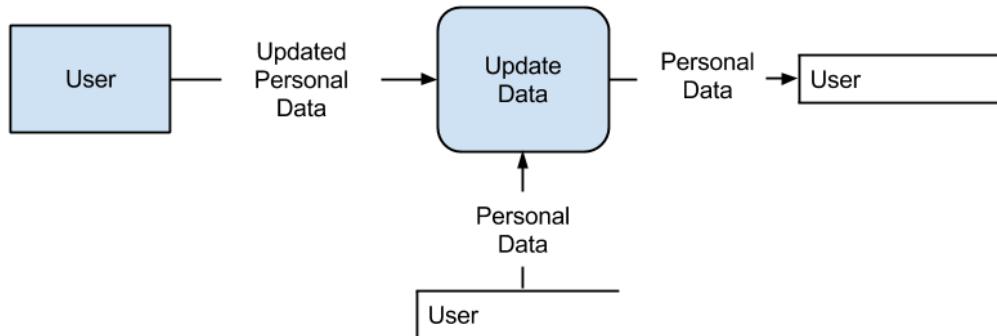
U7



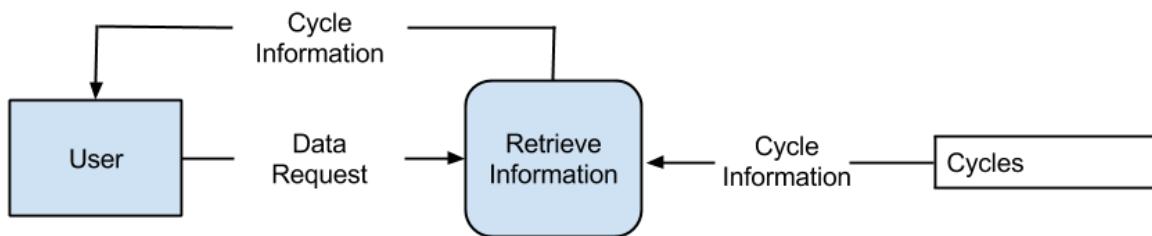
U8



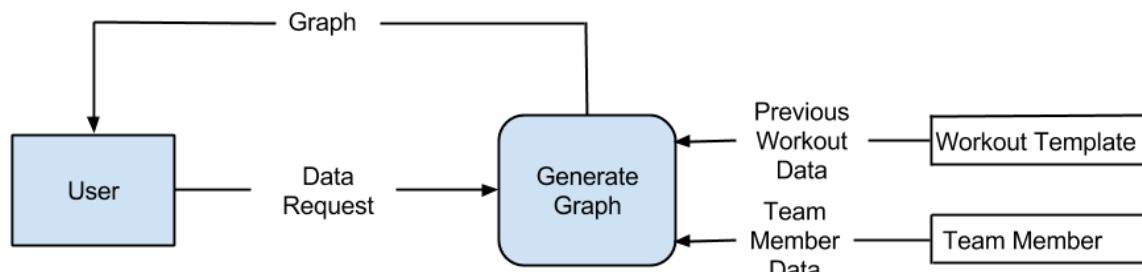
U9



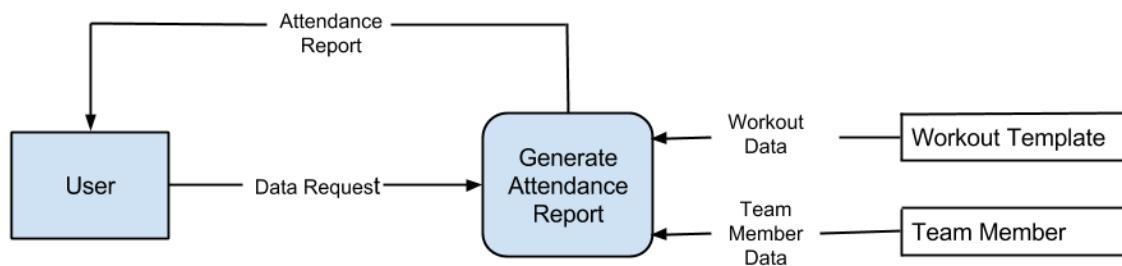
U10



U11



U12



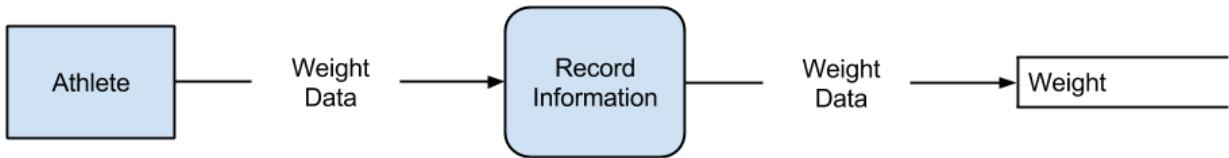
A1



A2



A3



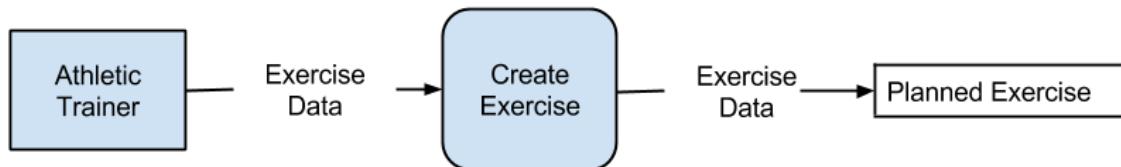
A4



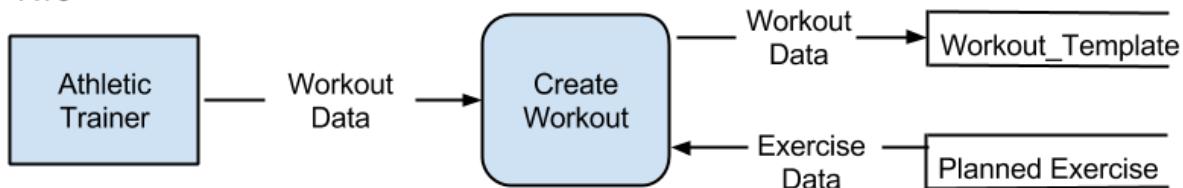
AT1



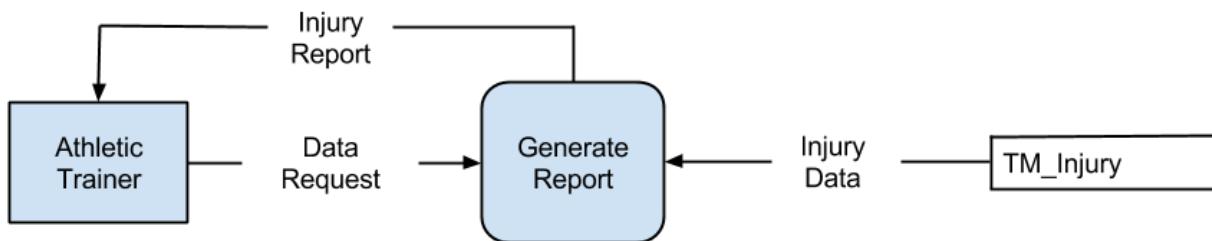
AT2



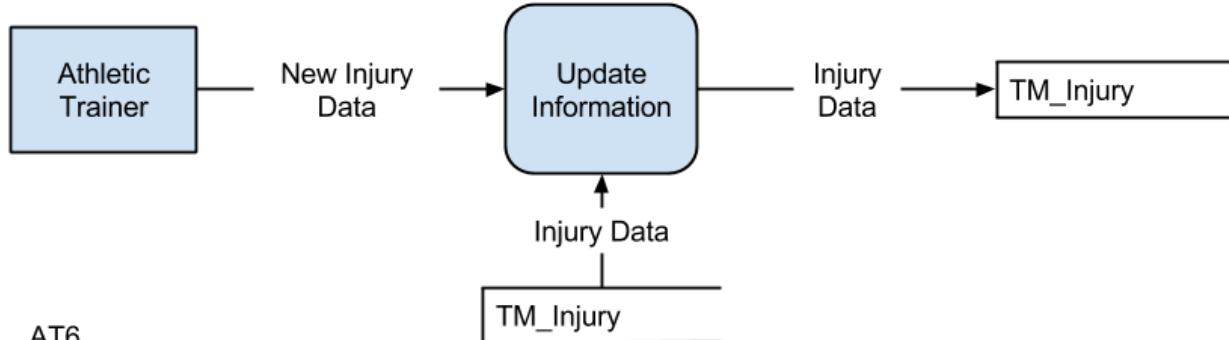
AT3



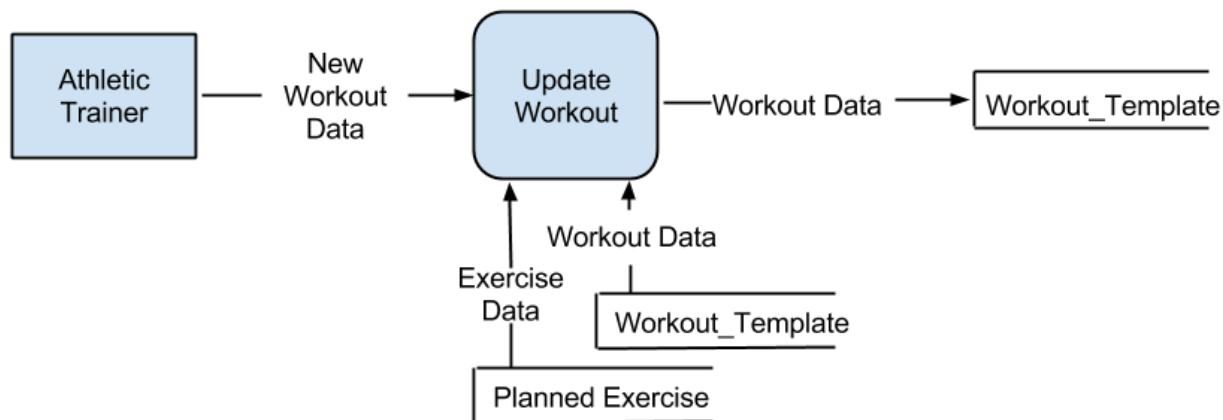
AT4



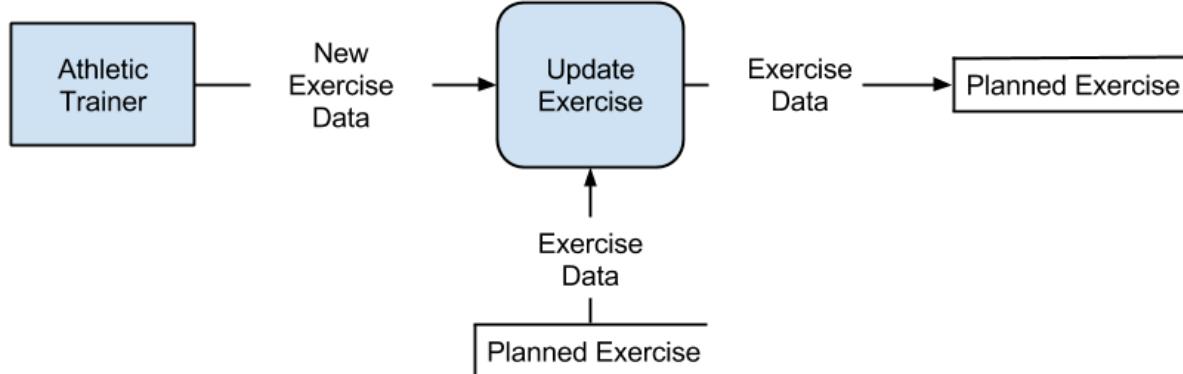
AT5



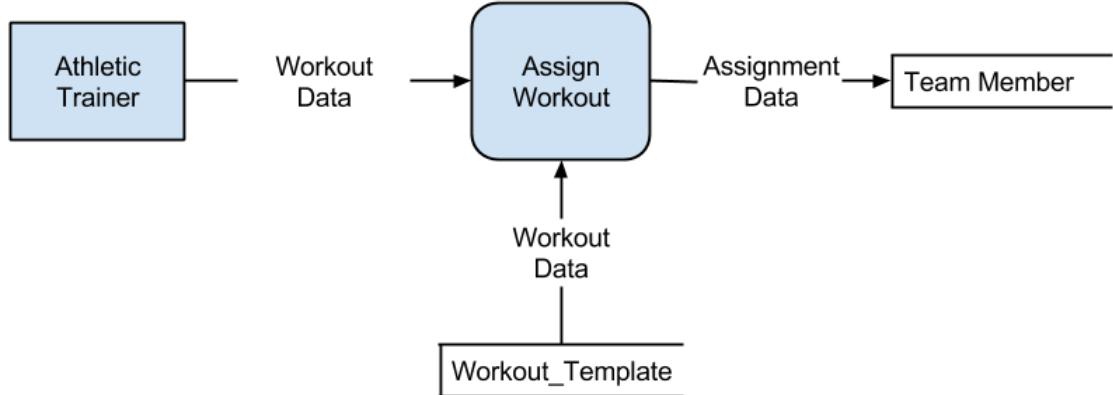
AT6



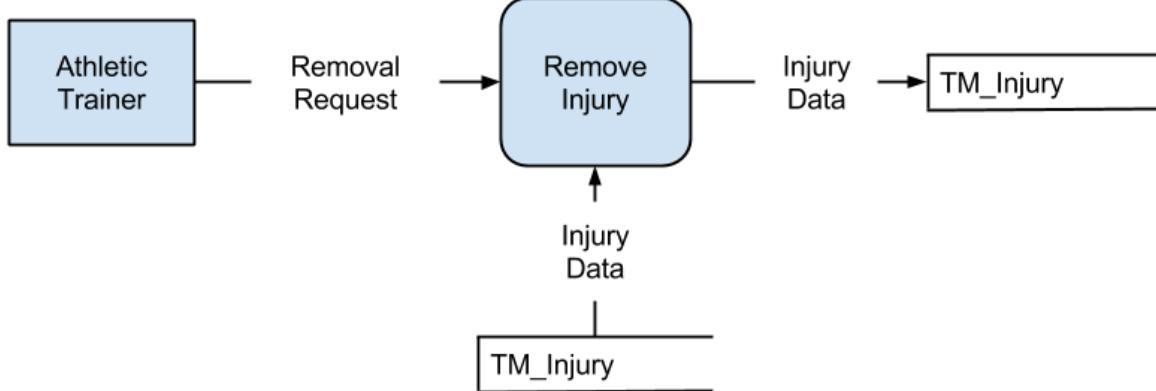
AT7



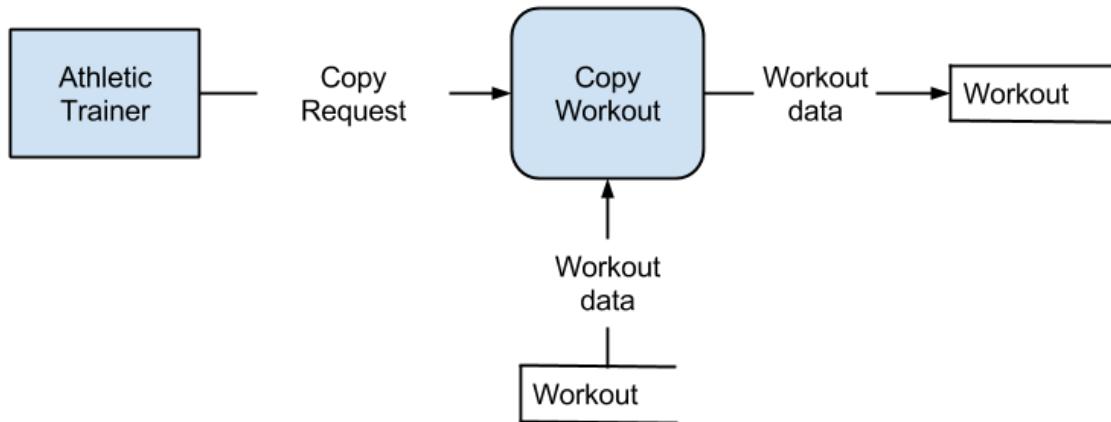
AT8



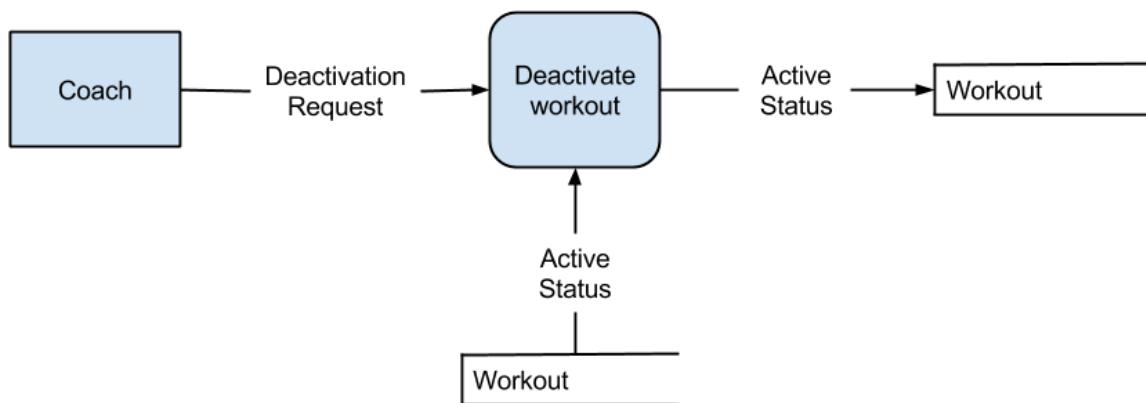
AT9



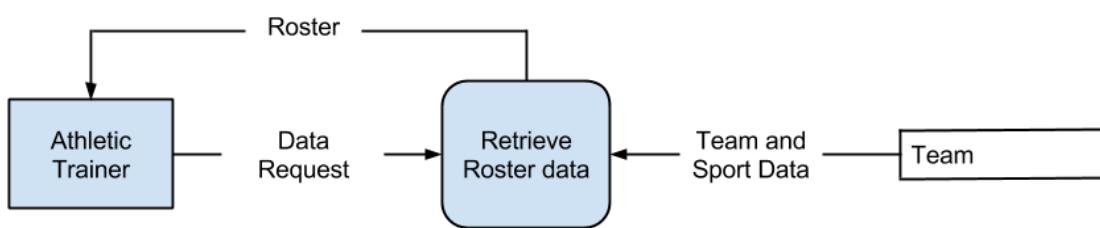
AT10



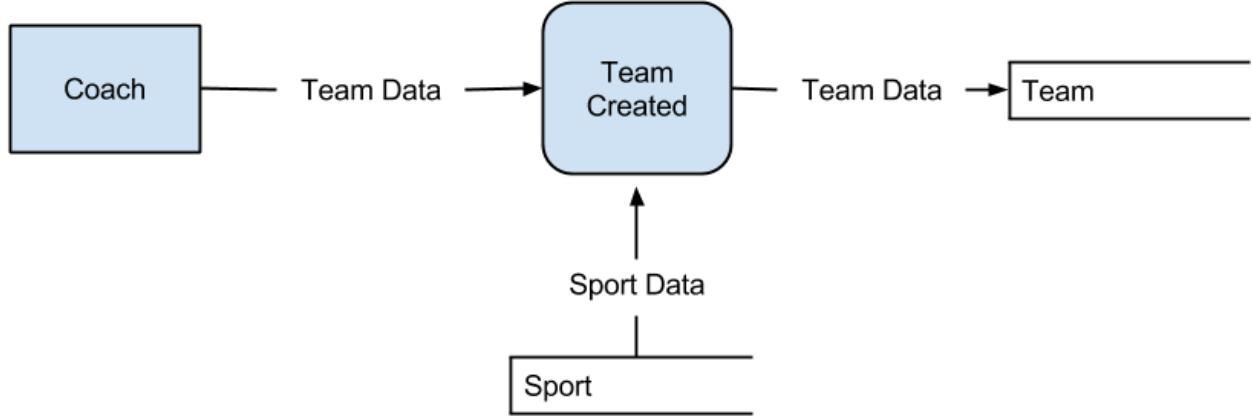
AT11

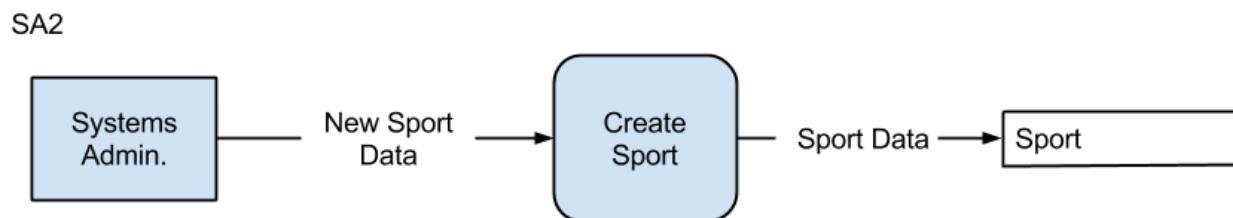
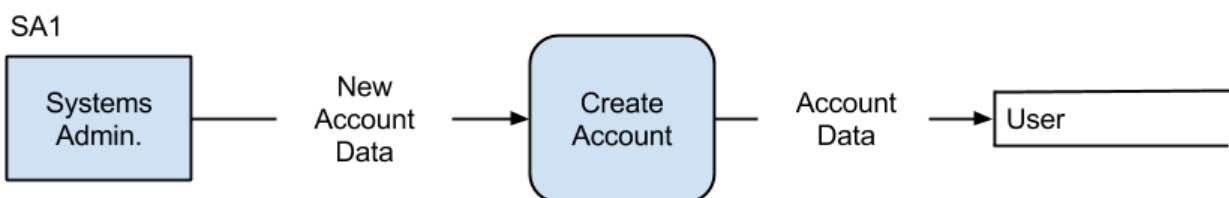
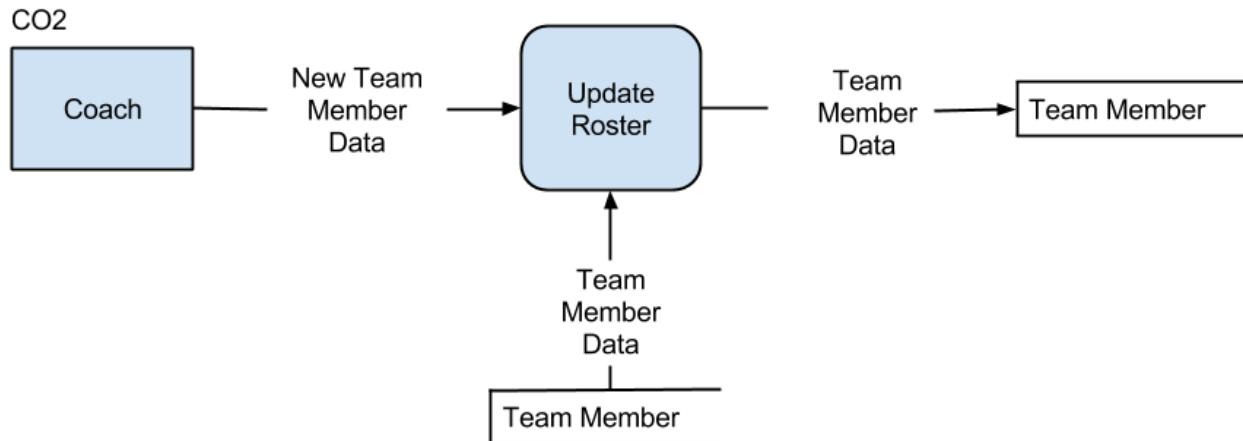


AT12

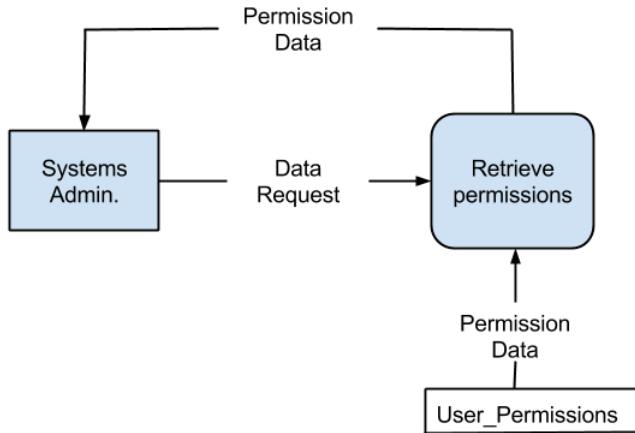


CO1

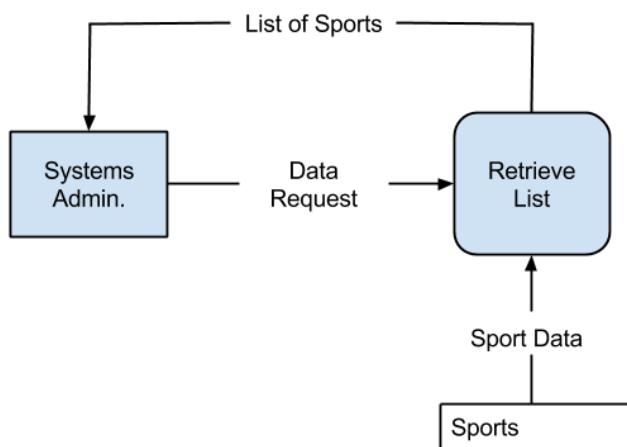




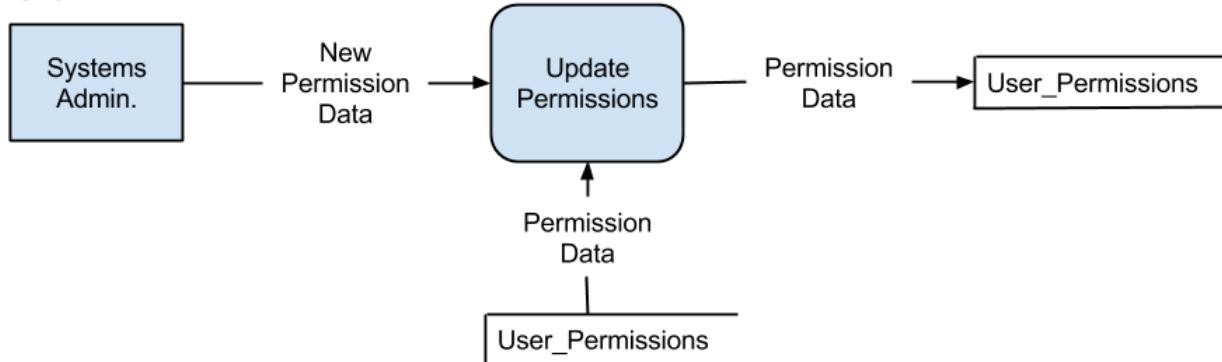
SA3



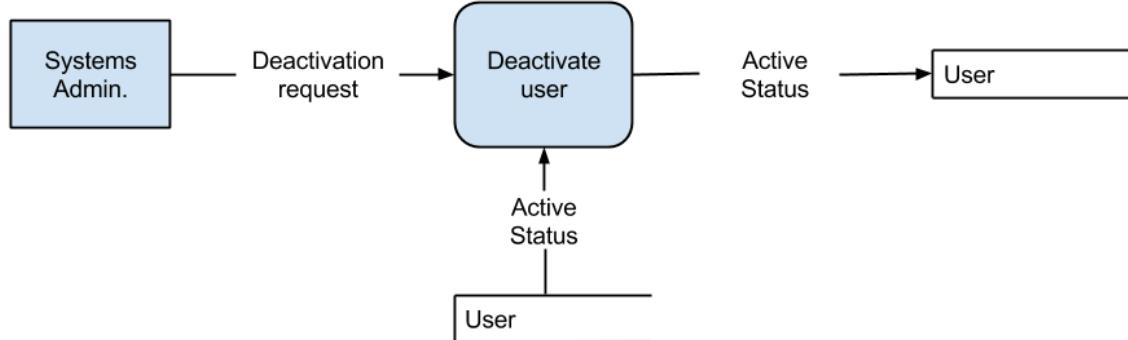
SA4



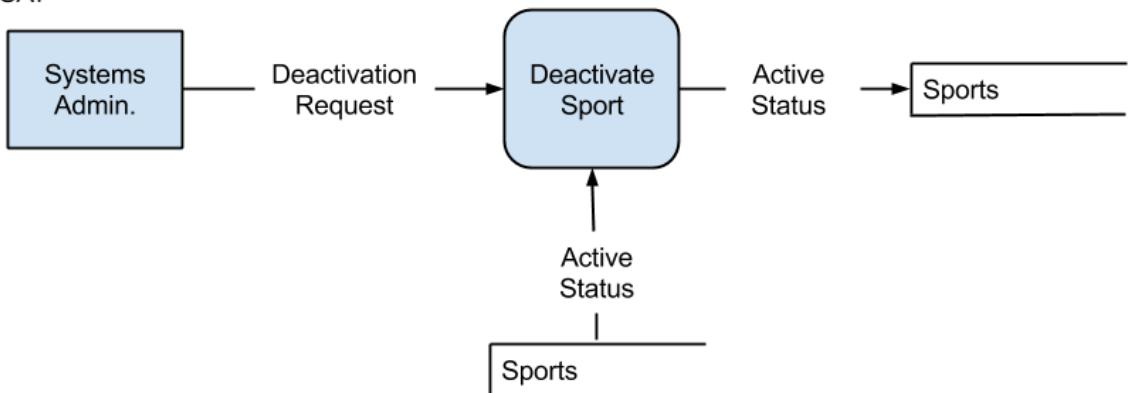
SA5



SA6



SA7



4.6.2 Data Description

Account Data = User_ID, Name, Password, Active?

Active Status = a boolean that determines whether or not a sport is active in the database

Copy Request = a packet of data specifying a request for specific data to be copied

Current User = User_ID

Cycle Information = Name, Sport_ID, Start_Date,

Data Request = a packet of data from the user specifying what data the system should pull from the database

Deactivation Request / Removal Request = a packet of information that deactivates data in the database

Exercise Data = refers to individual exercises that make up a workout

Graph = a graph generated from previous workout data allowing an athlete to view his or her progress

Injury Data = Injury_ID, TM_ID, Note

Injury Report = A list of injured athletes

Login Info = User_ID, Password

Logout Request = a packet of data triggered by a user clicking the “logout” button

Logout Confirmation = a message sent to the user confirming that the user has been properly logged out of the system

Nutrition Data = Date, TM_ID, Food, Note

Permission Data = User_ID, Perm_Abbrev

Personal Data = Information found in the User entity specific to the current user

Roster = a list of athletes belonging to a team or sport

Sleep Data = Date, TM_ID, Start, Stop

Sport Data = Sport_ID, Name, Active?

Team Data = Team_ID, Sport_ID, Name, Active?

Team Member Data = Team_ID, TM_ID, User_ID

Weight Data = Date, TM_ID, Weight

Workout Data= Contains information regarding which exercises are to be done / have been done by an athlete in a workout

4.6.3 Process Specification

The Process Specification of the Logical Process Model describes what actions the system takes in response to a specific event. Entries in **bold** refer to **entities** in the Logical Data Model.

U1

```
#login
get user input
if user input = false
display login error
if user input = true
display home page
```

U2

```
#logout
get user request
end user session
```

U3

```
#view personal information  
get user request  
get requested data from User  
display user data
```

U4

```
#view previous workouts  
get user request  
get requested data from Workout_Template  
display workout data
```

U5

```
#view current workout  
get user request  
get requested data from Workout_Template  
display workout data
```

U6

```
#view nutrition information  
get user request  
get requested data from Nutrition  
display nutrition data
```

U7

```
#view sleep information  
get user request  
get requested data from Sleep  
display sleep data
```

U8

```
#view weight information  
get user request  
get requested data from Weight  
display weight data
```

U9

```
#update user information  
get user input  
get current data in User  
overwrite current data in User with user's input
```

U10

```
#view cycle information
get user request
get requested data from Cycles
display cycle data
```

U11

```
#view report graphs
get user request
get requested data from Workout_Template and Team_Member
generate graph from requested data
display graphs
```

U12

```
#view attendance
get user request
get requested data from Workout_Template and Team_Member
generate attendance report from requested data
display attendance report
```

A1

```
#record workout information
get user input
record user input in Actual_Exercise
```

A2

```
#record nutrition information
get user input
record user input in Nutrition
```

A3

```
#record weight information
get user input
record user input in Weight
```

A4

```
#record sleep information
get user input
record user input in Sleep
```

AT1

#report injury
get user input
record user input in **Injury**

AT2

#create exercise
get user input
record user input in **Planned_Exercise**

AT3

#create workout
get user input
get exercise data from **Planned_Exercise**
record created workout in **Workout_Template**

AT4

#view injuries of athletes
get user request
get requested data from **TM_Injury**
display list of injuries

AT5

#update injury information
get user input
get current injury data from **TM_Injury**
overwrite current data in **TM_Injury** with updated data

AT6

#update workout information
get user input
get current workout data from **Workout_Template**
get current exercise data from **Planned_Exercise**
overwrite current data in **Workout_Template** with updated data

AT7

#update exercise
get user input
get current exercise data from **Planned_Exercise**
overwrite current data in **Planned_Exercise** with updated data

AT8

```
#assign workout
get user input
get specified workout from Workout_Template
record assignment data in Team Member
```

AT9

```
#remove injury
get user request
get selected injury from TM_Injury
deactivate selected injury
```

AT10

```
#make copy of workout
get user request
get requested data from Workout
copy retrieved data into Workout
```

AT11

```
#make workout inactive
get user request
get selected workout from Workout
deactivate selected workout
```

AT12

```
#view roster
get user request
get requested data from Team
display roster
```

CO1

```
#create a team
get user input
get specified data from Sport
create team in Team
```

CO2

```
#update sport or team roster
get user input
get data from Team_Member
overwrite current data in Team_Member with updated data
```

SA1

```
#create user
get user data from user input
if username exists
display login error
else
create user data in User
```

SA2

```
#create sport
get user input
if sport name already exists
display creation error
else
create sport data in Sport
```

SA3

```
#view permissions
get user request
get permission data from User_Permissions
display permission data
```

SA4

```
#list of all sports
get user request
get data from Sport
display data from Sport
```

SA5

```
#update permissions
get user input
get current permission data from User_Permissions
overwrite current User_Permissions data with updated data
```

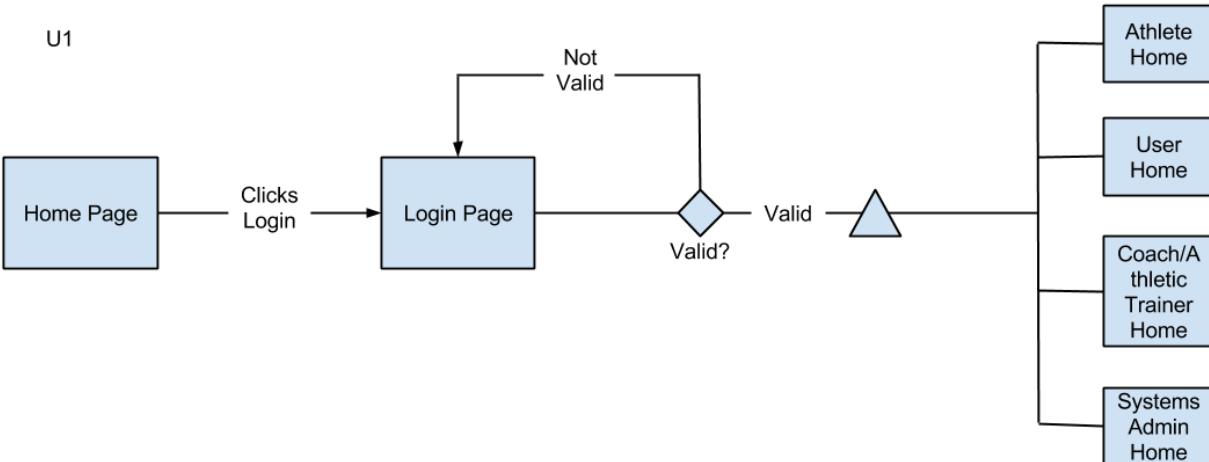
SA6

```
#deactivate user
get user request
get selected user data from User
deactivate user
```

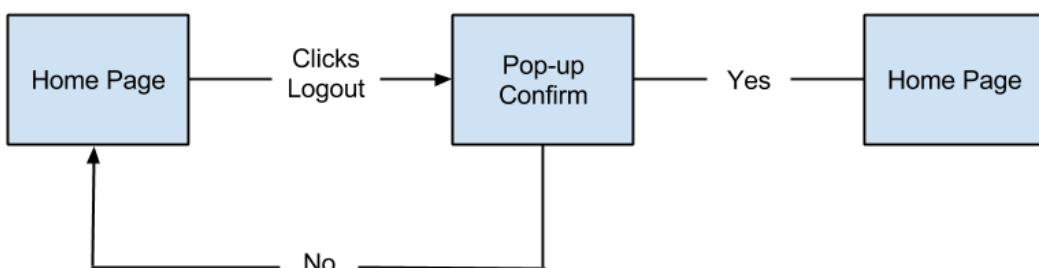
SA7

```
#deactivate sport
get user request
get selected sport data from Sport
deactivate sport
```

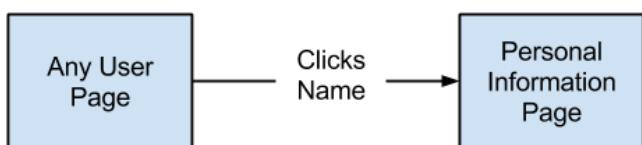
4.7 User Experience Diagrams



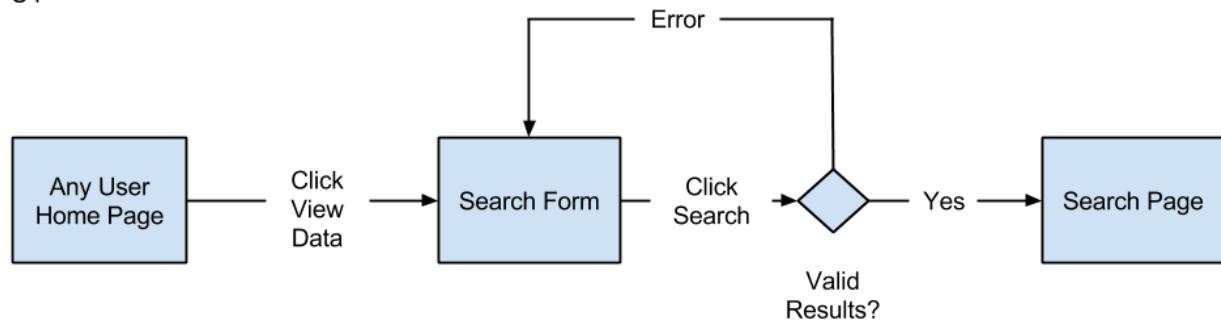
U2



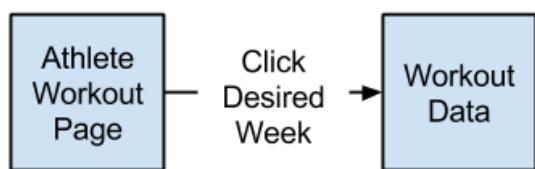
U3



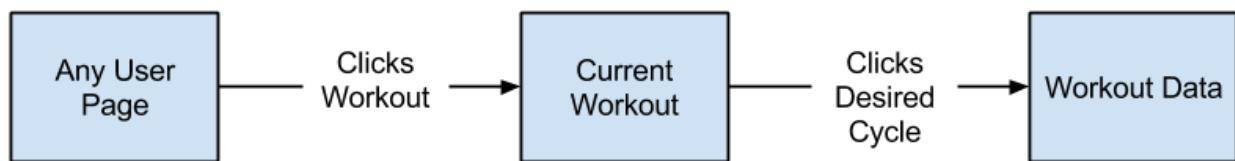
U4



ATHU4 / ATHU12



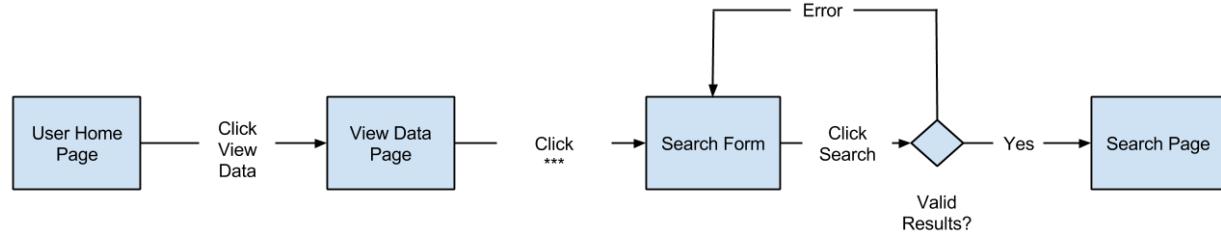
U5 / U10



ATHU5 / ATHU10

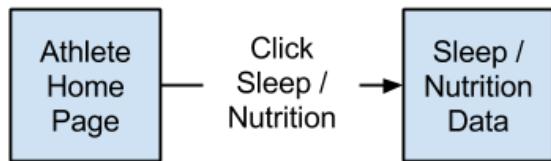


U6 / U7 / U8 / U11 / U12

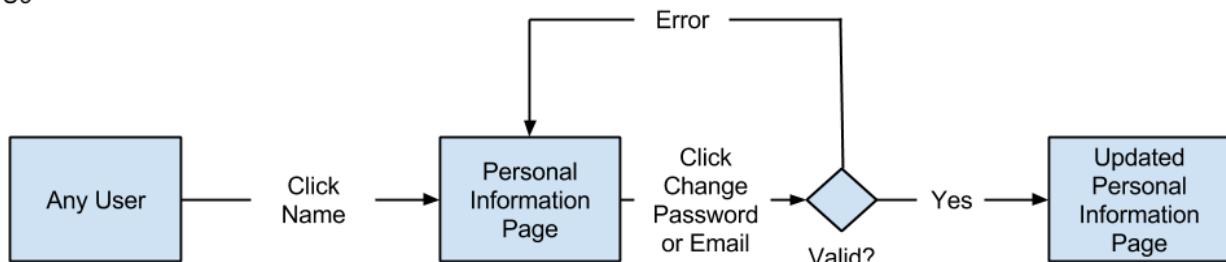


U6, U7, U8 *** = Health
U11 *** = Look at Graphs
U12 *** = Attendance

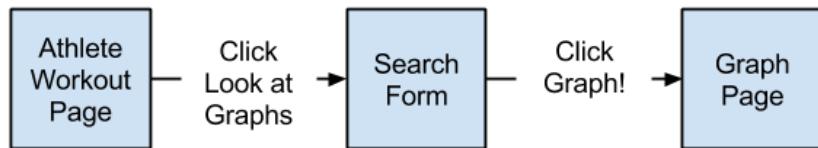
ATHU6 / ATHU7 / ATHU8



U9



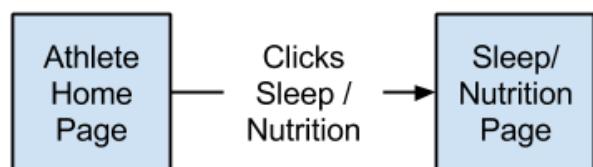
ATHU11



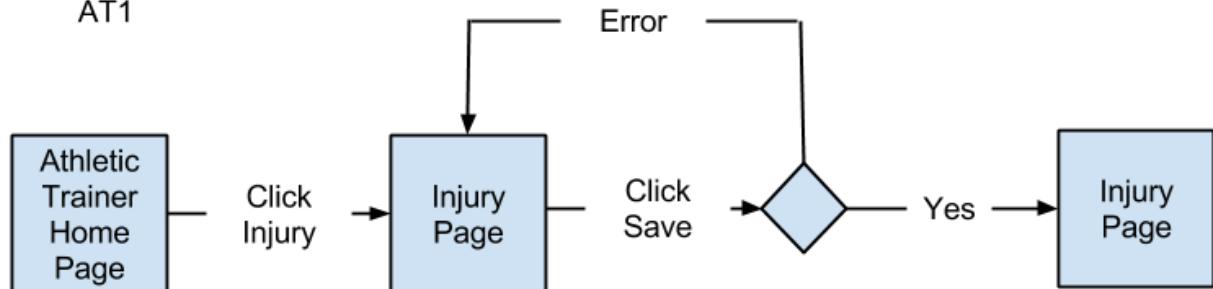
A1



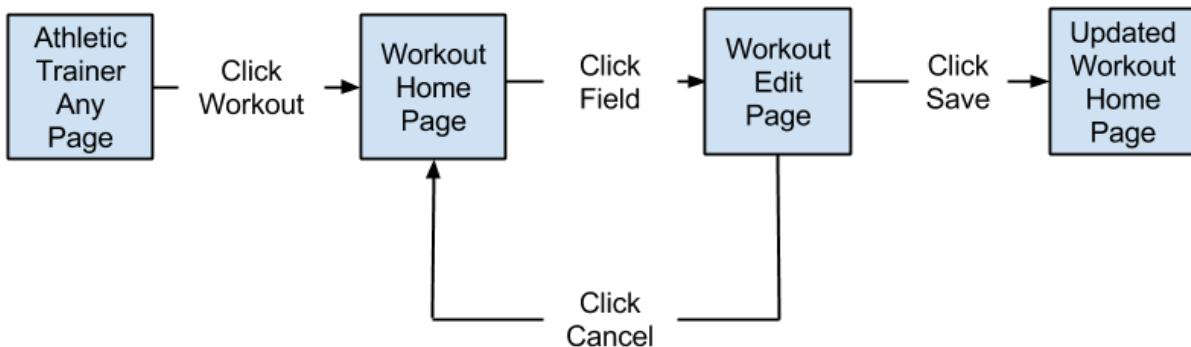
A2 / A3 / A4



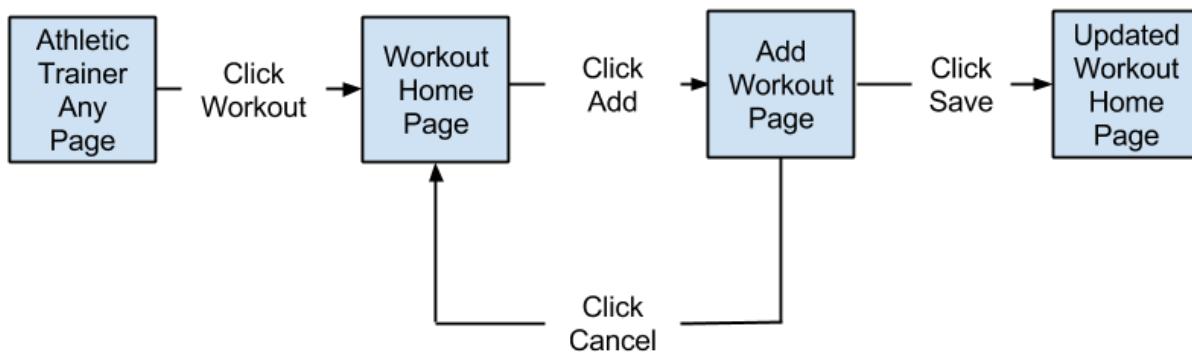
AT1



AT2



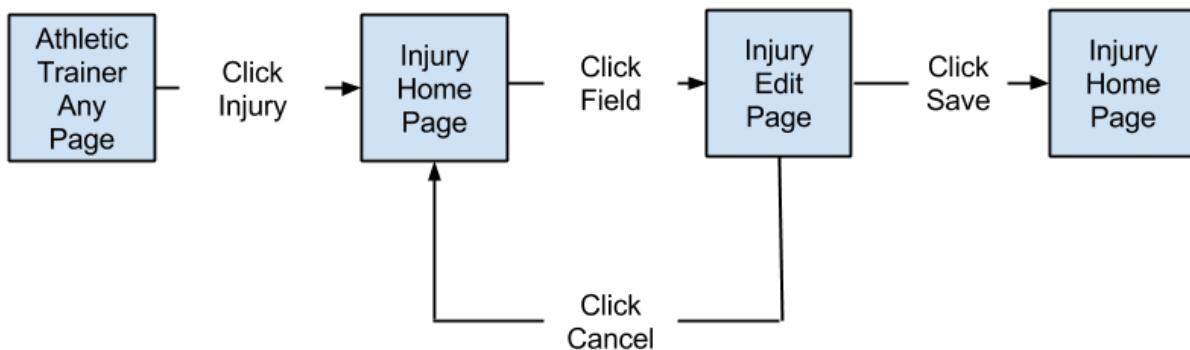
AT3



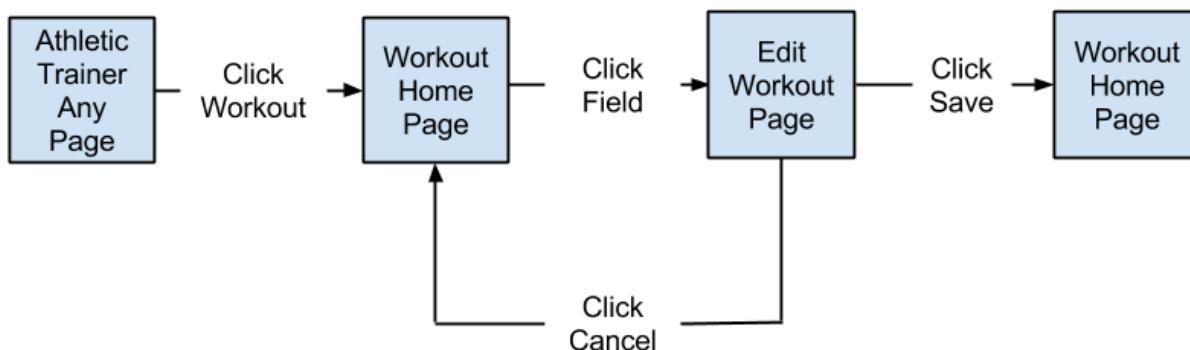
AT4



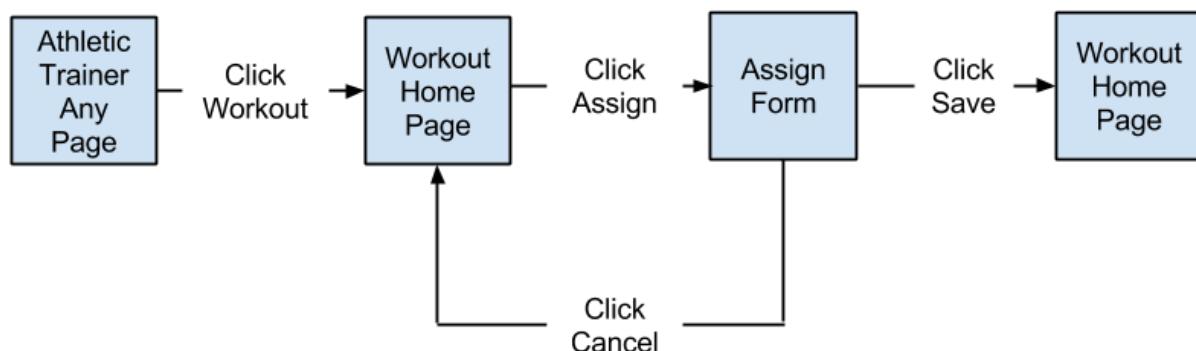
AT5



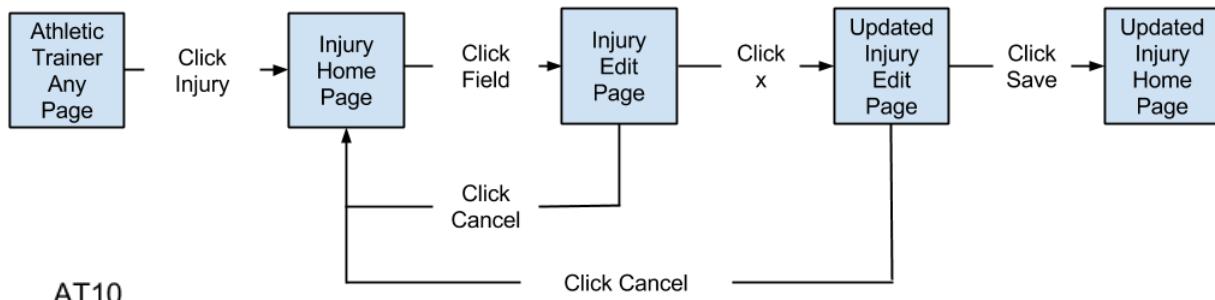
AT6 / AT7



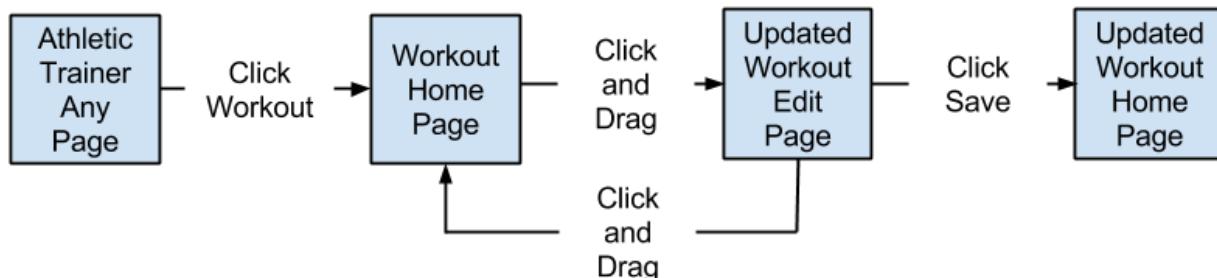
AT8



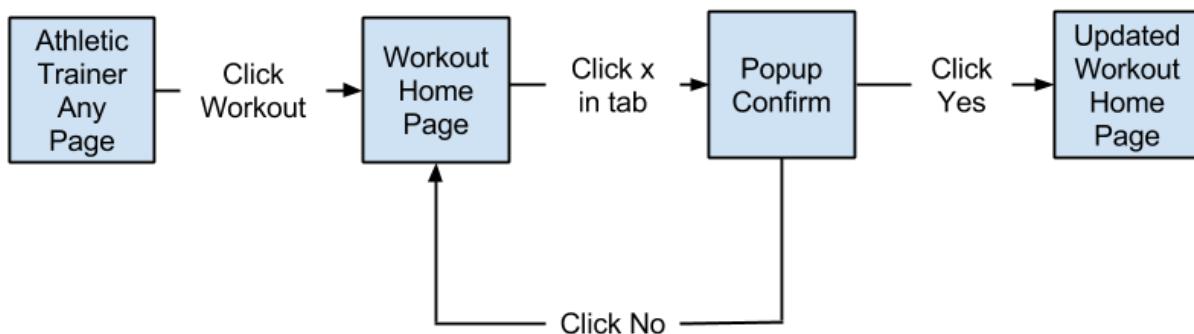
AT9



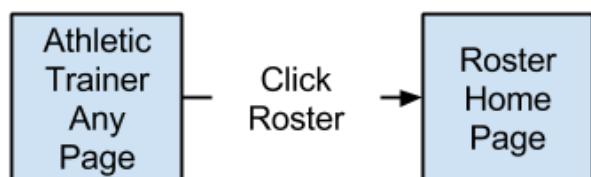
AT10



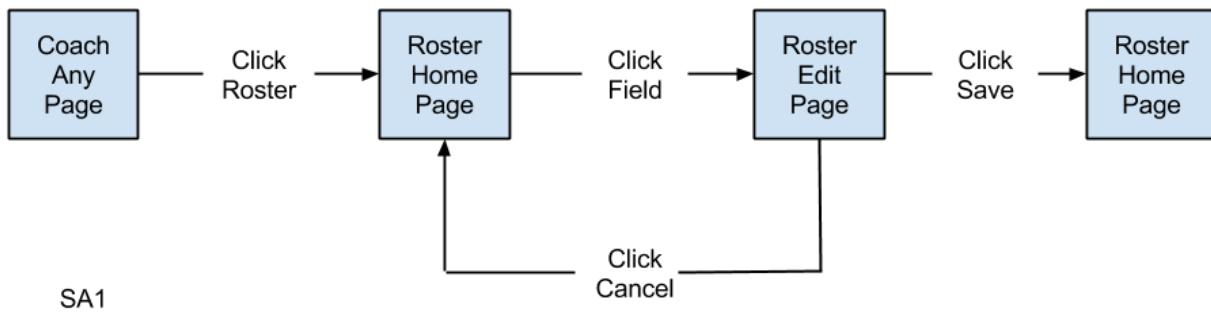
AT11



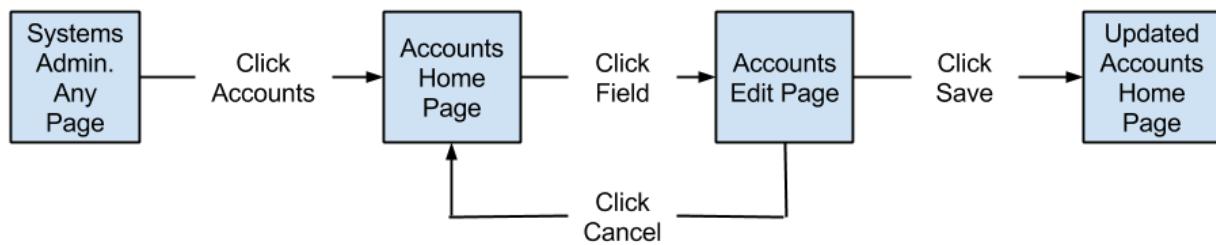
AT12



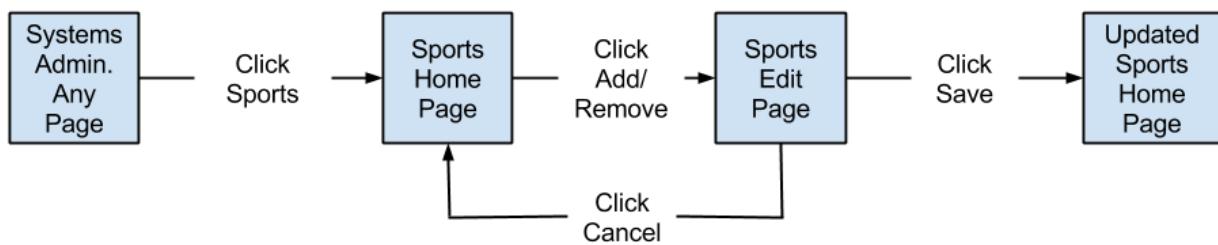
C01 / C02



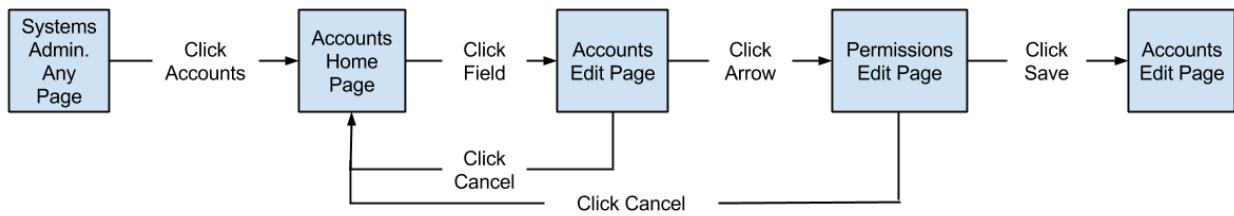
SA1



SA2



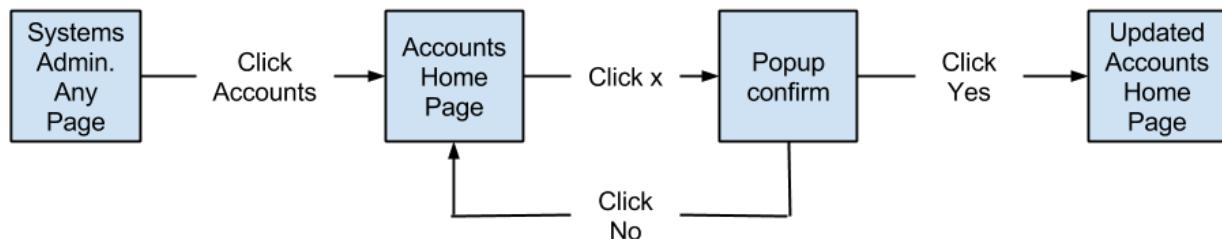
SA3 / SA5



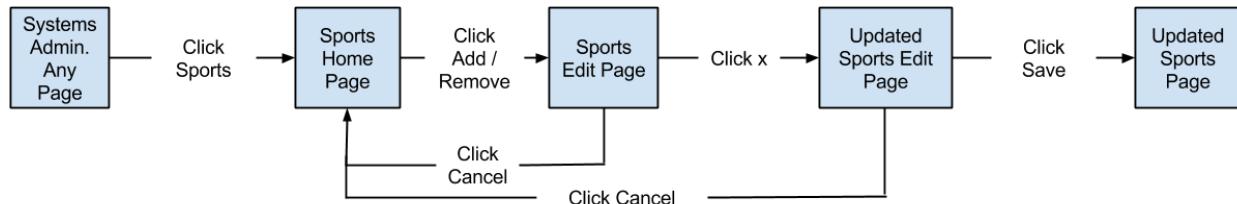
SA4



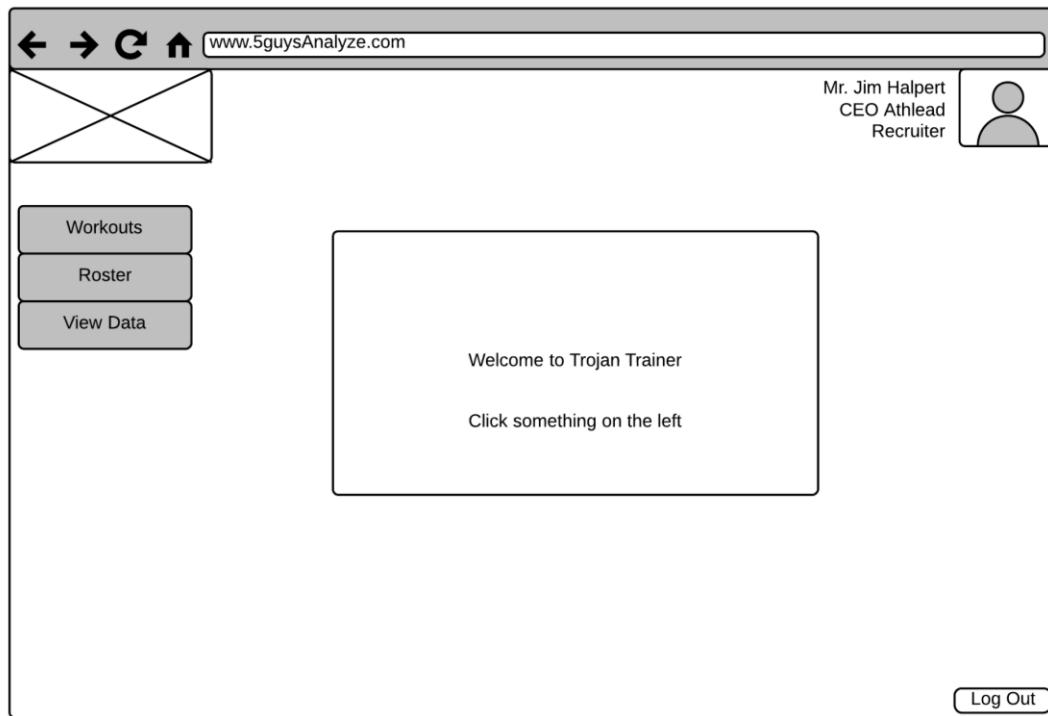
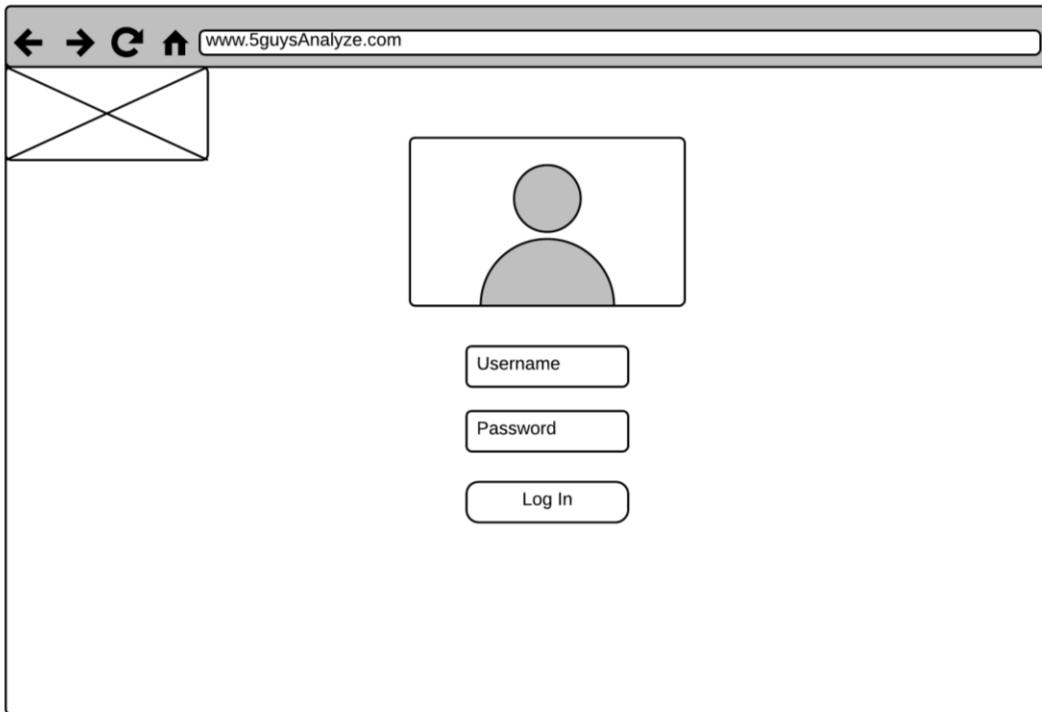
SA6

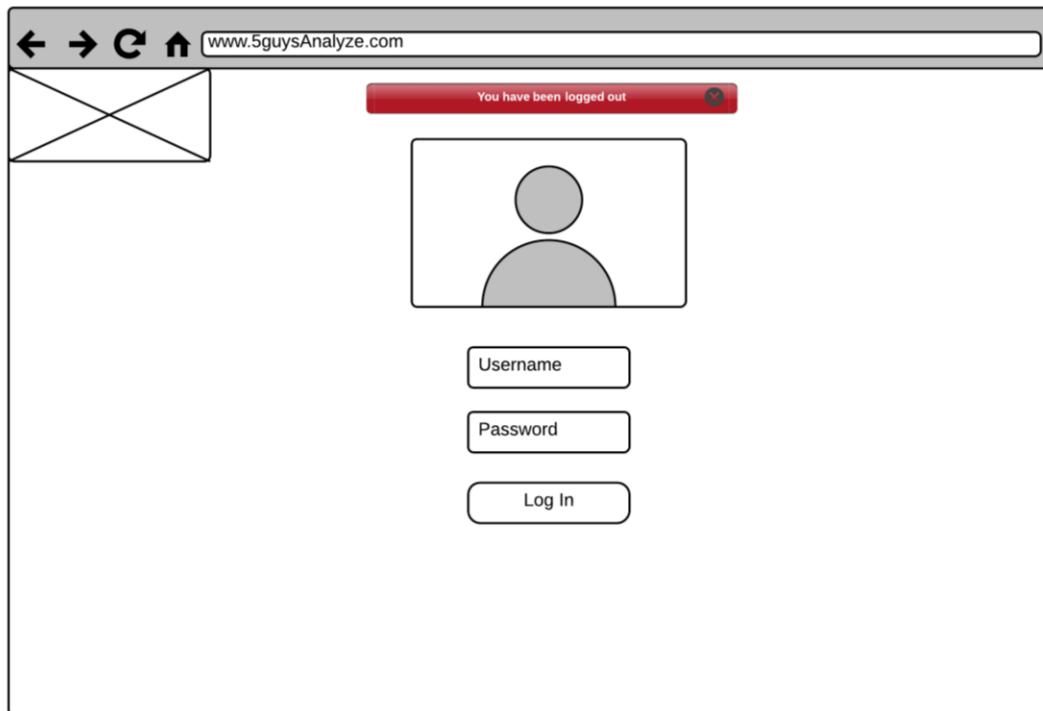


SA7



4.8 Wireframes

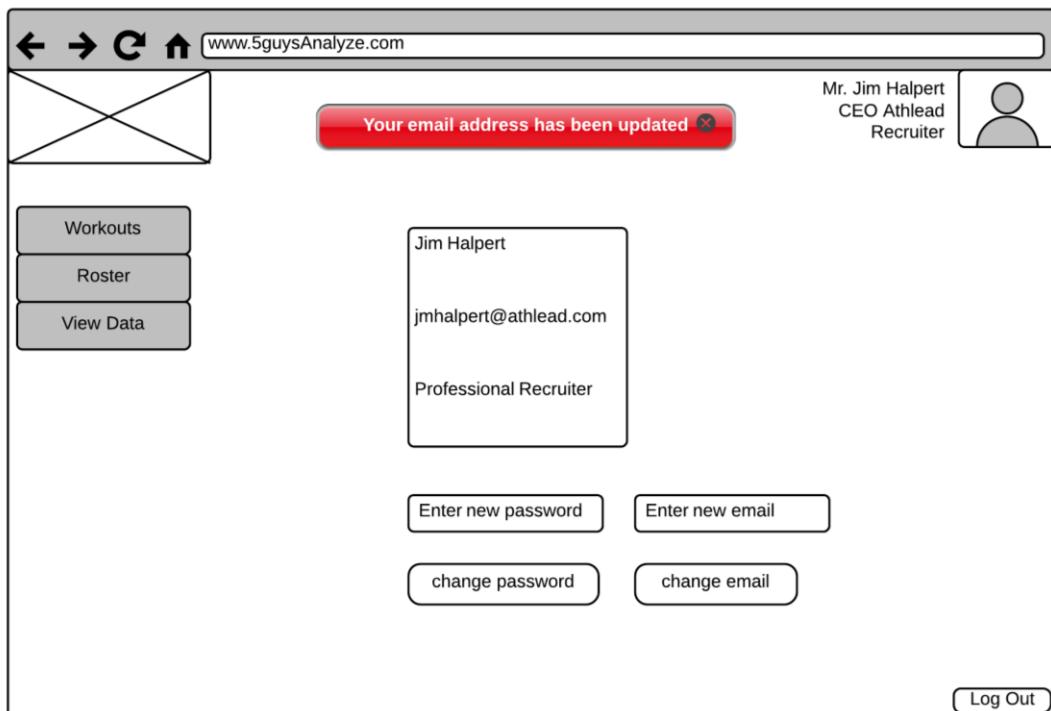
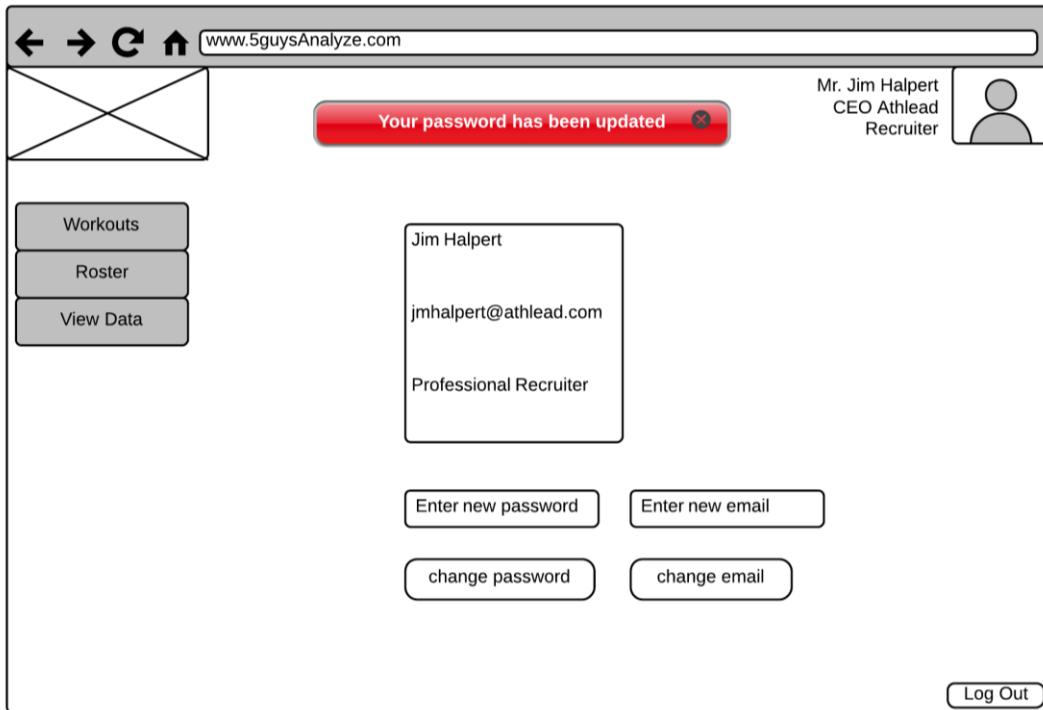


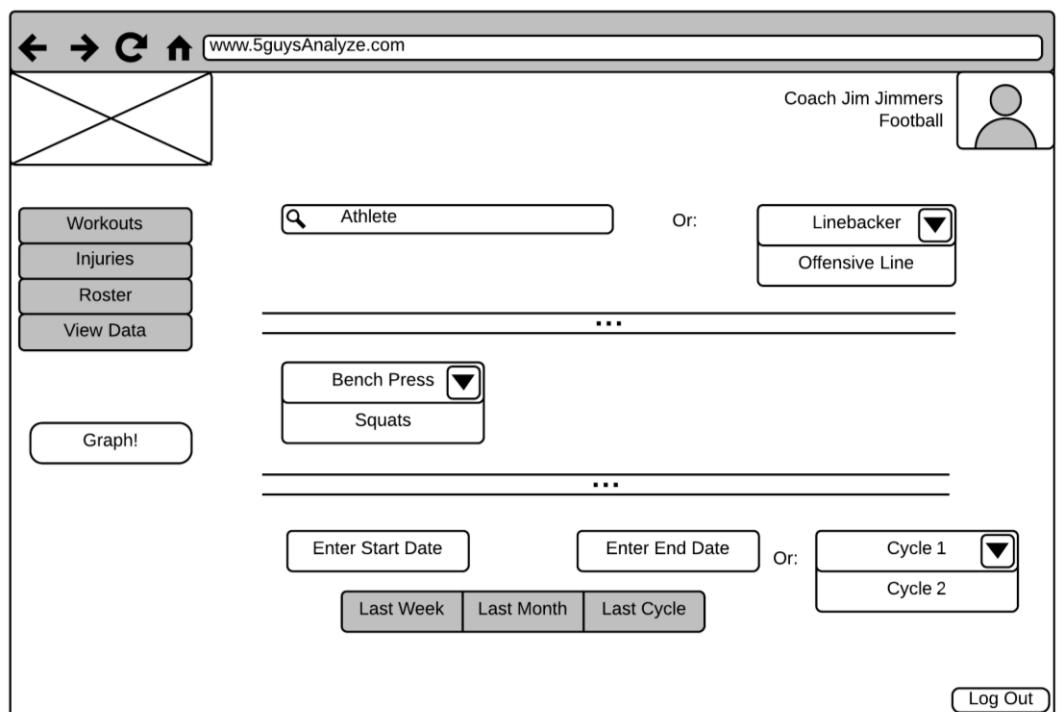
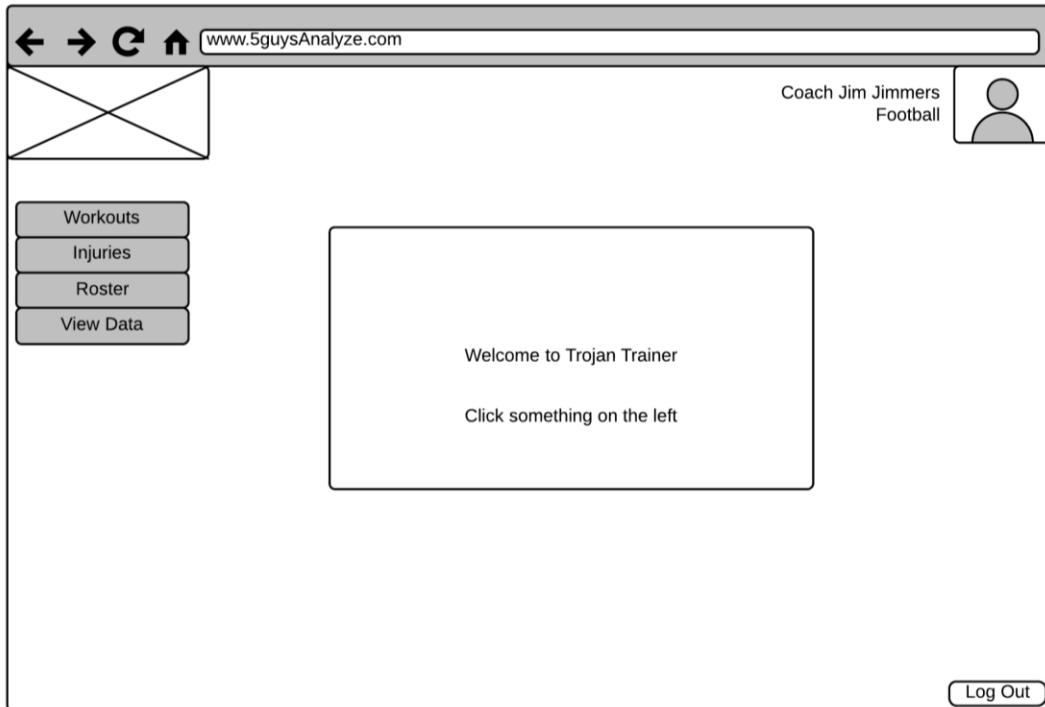


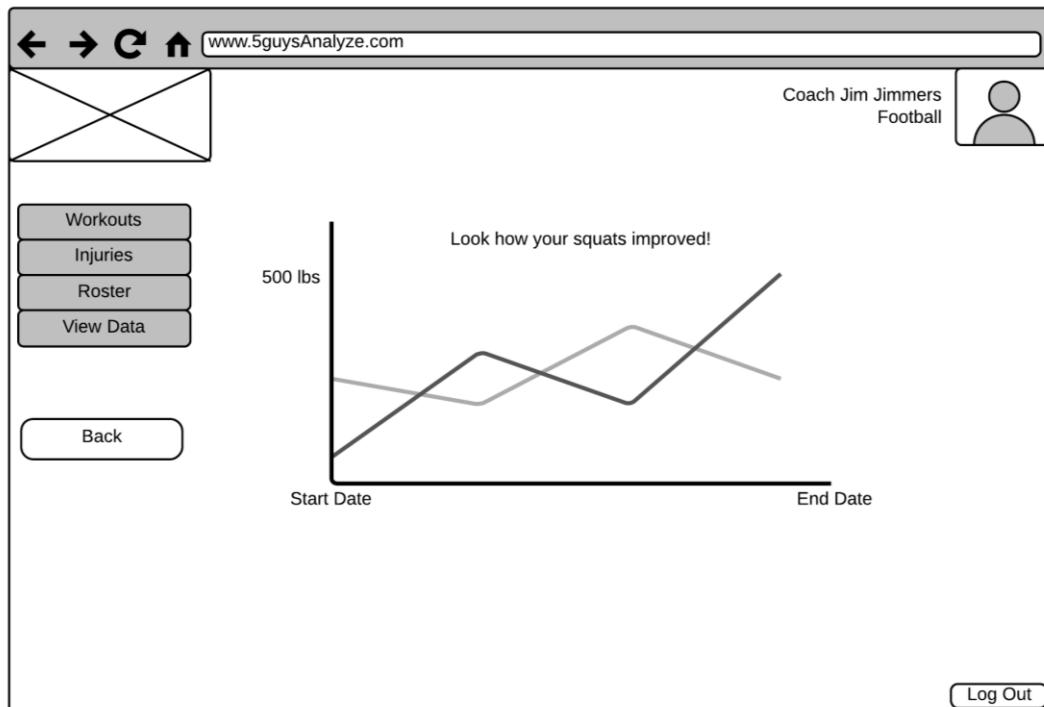
Log in as Sys Admin

Log in as athlete

The screenshot shows a web browser window with the URL www.5guysAnalyze.com. At the top, there are navigation icons for back, forward, and refresh. On the right side, a sidebar displays the user's information: "Mr. Jim Halpert", "CEO Athlead", and "Recruiter" next to a small user icon. Below the sidebar, a vertical menu on the left lists "Workouts", "Roster", and "View Data". To the right, a box contains the user's details: "Jim Halpert", "jmhlapert@athlead.com", and "Professional Recruiter". At the bottom, there are four buttons: "Enter new password", "Enter new email", "change password", and "change email". A "Log Out" button is located at the bottom right.







The screenshot shows a mobile application interface for 'Five Guys Analyze'. At the top, there are navigation icons (back, forward, refresh, home) and the URL 'www.5guysAnalyze.com'. On the right, it displays 'Coach Jim Jimmers' and 'Football' next to a user icon. On the left, there's a sidebar with buttons for 'Workouts', 'Injuries', 'Roster', and 'View Data'. Below the sidebar, there's a table listing injuries:

Name	Injury	Injury Date	Using W.O.	Team	Sport
James	broken neck	7/3/2013	Injured1	Linebacker	Football
Sam	paper cut	7/4/2013	Injured2	Defense	Soccer
Katelyn	leg fracture	6/19/2013	Injured3	Defense	Soccer

At the bottom, a message says 'Injuries was last updated "45 seconds ago"' and there's a 'Log Out' button.

www.5guysAnalyze.com

Coach Jim Jimmers
Football

Workouts

Injuries

Roster

View Data

Name	Injury	Injury Date	Using W.O.	Team	Sport
James	broken neck	7/3/2013	Injured1	Linebacker	Football
Sam	paper cut	7/4/2013	Injured2	Defense	Soccer
Katelyn	leg fracture	6/19/2013	Injured3	Defense	Soccer
	Type Here				
	broken neck				
	dislocated arm				

 Cancel Changes Log Out

www.5guysAnalyze.com

Coach Jim Jimmers
Football

Workouts

Injuries

Roster

View Data

Name	Team	Sport	injured	current workout
James	Linebacker	Football	<input checked="" type="radio"/>	injured 1
Sam	Defense	Soccer	<input type="radio"/>	cycle 1
Katelyn	Defense	Soccer	<input type="radio"/>	cycle 1

Log Out

www.5guysAnalyze.com

Coach Jim Jimmers
Football

Name	Team	Sport	injured	current workout
James	Linebacker	Football	<input checked="" type="radio"/>	injured 1
Sam	Defense	Soccer	<input type="radio"/>	cycle 1
Katelyn	Defense	Soccer	<input type="radio"/>	cycle 1
		Football	<input checked="" type="radio"/>	
		Soccer	<input type="radio"/>	
		Baseball	<input type="radio"/>	
			<input type="radio"/>	

Cancel Changes Log Out

www.5guysAnalyze.com

Coach Jim Jimmers
Football

Workout Health Attendance

Workouts Athlete ...

Injuries Enter Start Date Enter End Date Or:

Roster Cycle 1 Cycle 2

View Data Last Week Last Month Last Cycle

Search! Look at Graphs

Log Out

The screenshot shows a web-based application interface for 'Five Guys Analyze' at www.5guysAnalyze.com. The top right corner displays the user information 'Coach Jim Jimmers Football' next to a profile icon. The main content area shows a summary for 'Matt Boyd' under the heading 'Workouts Missed'. It lists two entries: 'Monday July 7 cycle 1' and 'Friday August 13 cycle 2'. On the left sidebar, there are buttons for 'Workouts', 'Injuries', 'Roster', and 'View Data'. Below the sidebar is a button labeled 'Look at Graphs'. At the bottom right is a 'Log Out' button.

The screenshot shows a search interface for the 'Five Guys Analyze' application. At the top right, the user 'Coach Jim Jimmers Football' is logged in. A search bar contains the placeholder text 'Athlete'. Below the search bar are three input fields: 'Enter Start Date', 'Enter End Date', and a dropdown menu labeled 'Or:' with options 'Cycle 1' and 'Cycle 2'. Underneath these fields are three buttons: 'Last Week', 'Last Month', and 'Last Cycle'. On the left side, there is a sidebar with buttons for 'Workouts', 'Injuries', 'Roster', and 'View Data', along with a 'Search!' button and a 'Look at Graphs' button. At the bottom right is a 'Log Out' button.

Sleep / Nutrition

Sleep	Bedtime:	Hours:	
Meals	B <input checked="" type="checkbox"/>	L <input checked="" type="checkbox"/>	D <input checked="" type="checkbox"/>
Snacks	M <input type="checkbox"/>	A <input checked="" type="checkbox"/>	N <input checked="" type="checkbox"/>
Weight	Pre 160	Post 160	
Monday Tuesday Wednesday Thursday Friday Saturday			

Log Out

Workout **Health** **Attendance**

Coach Jim Jimmers Football

Workouts Injuries Roster View Data

Athlete Or: Linebacker
 Offensive Line

Search!

Look at Graphs

Enter Start Date Enter End Date Or: Cycle 1
 Cycle 2

Last Week Last Month Last Cycle

Log Out

The screenshot shows a software interface for managing player workouts. At the top, there are navigation icons (back, forward, search) and the URL "www.5guysAnalyze.com". On the right, it displays "Coach Jim Jimmers" and "Football" next to a user profile icon. A sidebar on the left contains links for "Workouts", "Injuries", "Roster", "View Data", and "Look at Graphs". The main content area is titled "Football: Linebacker Matt Boyd". It shows a timeline with four tabs: "weeks 1-3", "weeks 4-5" (which is selected), "week 6", and "pre-game". Below the timeline is a table of exercises with columns for "Exercises", "Weight", "Sets", "Reps", "Time", and "Level". The table includes rows for Bench Press (220, 3, 4), an unnamed exercise (230, 1, 4), Pull Ups (10), and Treadmill (30min, 8mph). At the bottom of the table are buttons for "Monday", "Tuesday" (which is highlighted in yellow), "Wednesday", "Thursday", "Friday", and "Saturday". A "Log Out" button is located at the bottom right.

Exercises	Weight	Sets	Reps	Time	Level
Bench Press	220	3	4		
—	230	1	4		
Pull Ups			10		
Treadmill				30min	8mph

This screenshot shows the same software interface as the first one, but with a dropdown menu open. The dropdown is titled "Select a Workout Group" and has a downward arrow icon. It also includes "Add" and "Assign" buttons. The rest of the interface is identical to the first screenshot, including the sidebar, timeline, and exercise table.

Exercises	Weight	Sets	Reps	Time	Level
Bench Press	220	3	4		
—	230	1	4		
Pull Ups			10		
Treadmill				30min	8mph

At the bottom of the interface, a message says "Workout was last updated "3 days ago"" and a "Log Out" button is located at the bottom right.

The screenshot shows a software interface for managing sports training. At the top right, it displays "Coach Jim Jimmers Football" with a user icon. On the left, there's a sidebar with buttons for "Workouts", "Injuries", "Roster", and "View Data". Below that is a list of "Cycle 1" through "Cycle 6" and an "Injuries" button. The main area has a "Workout Name" input field and dropdown menus for "Sport" (set to Football), "Team" (Linebacker, Offensive), and "Player" (Matt Boyd, James Smith). Below these are buttons for "Monday", "Tuesday" (which is highlighted in yellow), "Wednesday", "Thursday", "Friday", and "Saturday". At the bottom are "Save" and "Cancel Changes" buttons, along with a "Log Out" link.

This screenshot shows a specific workout setup. The title at the top right is "Football Saturday Special Workout". The "Sport" dropdown is set to Football, and the "Team" dropdown is set to Linebacker, Offensive. The "Player" dropdown shows Matt Boyd and James Smith. The sidebar and other interface elements are identical to the first screenshot, including the "Assign" button at the top right.

The screenshot shows a web-based application for creating a football workout plan. At the top right, it says "Coach Jim Jimmers Football" next to a user icon. On the left, there's a sidebar with links like "Workouts", "Injuries", "Roster", "View Data", "Cycle 1" through "Cycle 6", and "Injuries". The main area is titled "Football: Linebacker" and has tabs for "weeks 1-3", "weeks 4-5", "week 6", "pre-game", and "Assign". Below these tabs is a table for exercises:

Exercises	Weight	Sets	Reps	Time	Level
Bench Press <input checked="" type="checkbox"/>	220	3	4		
<input checked="" type="checkbox"/>	230	1	4		
Pull Ups <input checked="" type="checkbox"/>			10		
Treadmill <input checked="" type="checkbox"/>				30min	8mph
Type Here <input checked="" type="checkbox"/>					
Military Press					
Arm Curl					
Squats					

At the bottom, there are buttons for "Monday", "Tuesday" (highlighted in blue), "Wednesday", "Thursday", "Friday", and "Saturday". Below the table are buttons for "Cancel Changes" and "Log Out".

The screenshot shows the "Home" page of the application. At the top right, it says "Matt Boyd Football" next to a user icon. The main area has a title "Select a Week" and a list of week ranges:

- 7/1 - Week 1 - 7/7
- 7/8 - Week 2 - 7/14
- 7/15 - Week 3 - 7/21
- 7/22 - Week 4 - 7/29

Below this is a "Month Slider" icon. To the right, there's a large empty box with the text "Your workout is loading, thanks for waiting". At the bottom right is a "Log Out" button.

Home www.5guysAnalyze.com

The interface shows a navigation bar with arrows and a home link. A user profile for "Matt Boyd Football" is displayed. On the left, a sidebar titled "Select a Week" lists weeks from "7/1 - Week 1 - 7/7" to "7/22 - Week 4 - 7/29". Below it is a "Month Slider" and a "Look at Graphs" button. The main content area has tabs for "Today's Workout" and "Sleep / Nutrition", with "Today's Workout" selected. It displays a table of exercises with columns for Exercise, Weight, Sets, Reps, Time, and Level. Exercises listed include Bench Press, Pull Ups, and Treadmill. At the bottom are links for the days of the week: Monday, Tuesday (highlighted), Wednesday, Thursday, Friday, and Saturday. A "Log Out" button is at the bottom right.

Exercises	Weight	Sets	Reps	Time	Level
Bench Press	220	0/3 <input type="button" value="+"/>	0/4 <input type="button" value="+"/>		
—		0/1 <input type="button" value="+"/>	0/4 <input type="button" value="+"/>		
Pull Ups			10		
Treadmill				30min	8mph

Monday | **Tuesday** | Wednesday | Thursday | Friday | Saturday

Log Out

Home www.5guysAnalyze.com

The interface shows a navigation bar with arrows and a home link. A user profile for "Matt Boyd Football" is displayed. On the left, a sidebar titled "Select a Week" lists weeks from "7/1 - Week 1 - 7/7" to "7/22 - Week 4 - 7/29". Below it is a "Month Slider" and a "Look at Graphs" button. The main content area has tabs for "Today's Workout" and "Sleep / Nutrition", with "Sleep / Nutrition" selected. It displays a table with sections for Sleep, Meals, Snacks, and Weight. Under Sleep, there are fields for Bedtime and Hours. Under Meals, there are checkboxes for B, L, and D. Under Snacks, there are checkboxes for M, A, and N. Under Weight, there are fields for Pre and Post. At the bottom are links for the days of the week: Monday, Tuesday (highlighted), Wednesday, Thursday, Friday, and Saturday. A "Log Out" button is at the bottom right.

Sleep	Bedtime: <input type="text"/>	Hours: <input type="text"/>	
Meals	B <input type="checkbox"/>	L <input type="checkbox"/>	D <input type="checkbox"/>
Snacks	M <input type="checkbox"/>	A <input type="checkbox"/>	N <input type="checkbox"/>
Weight	Pre <input type="text"/>	Post <input type="text"/>	

Monday | **Tuesday** | Wednesday | Thursday | Friday | Saturday

Log Out

Home www.5guysAnalyze.com



Matt Boyd
Football 

Select a Week

7/1 - Week 1 - 7/7

7/8 - Week 2 - 7/14

7/15 - Week 3 - 7/21

7/22 - Week 4 - 7/29

Bench Press 

Squats

...

Month Slider

Enter Start Date Enter End Date Or:

Last Week Last Month Last Cycle

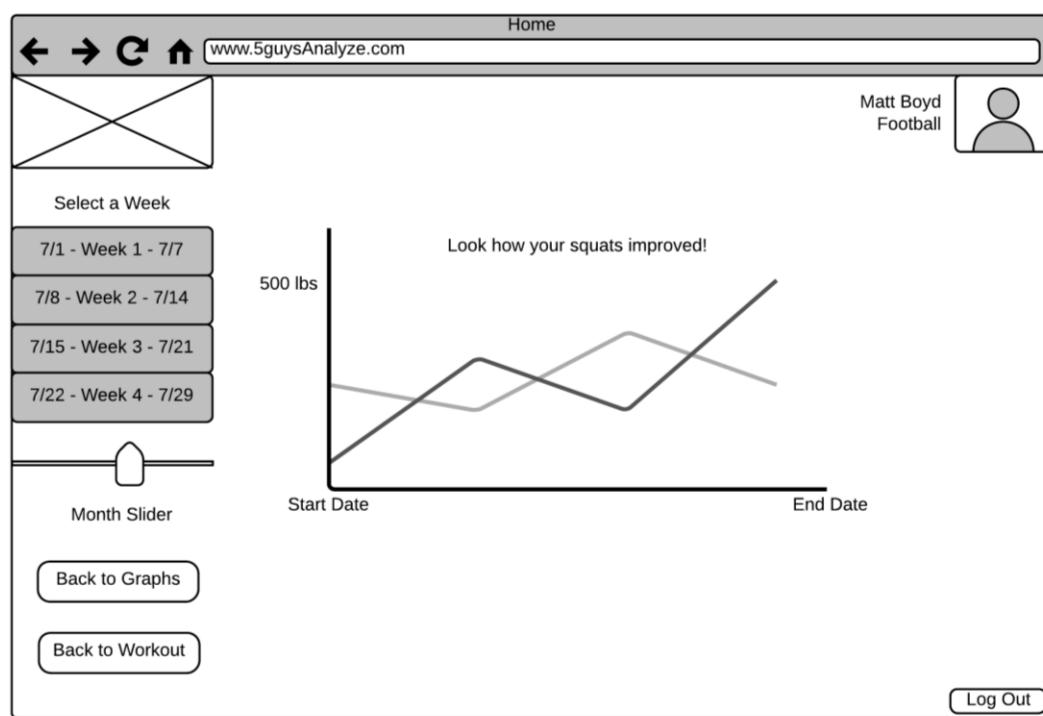
Cycle 1 

Cycle 2

Graph!

Back to Workout

Log Out





A wireframe of a user management page. The header and sidebar are identical to the previous wireframe. The main content area features a title 'Add / Remove Users' above a table. The table has three columns: Name, Email, and Role. It lists four users:

Name	Email	Role
Jim Jimmers	Jim@jimmers.com	Coach
Matt Boyd	matt@me.com	athlete
April Shoures	april@me.com	Athletic Trainer
katelyn	katelyn@me.com	athlete

At the bottom right of the main area is a 'Log Out' button.

www.5guysAnalyze.com

Admin Bob Katz

Workouts
Injuries
Roster
View Data
Accounts
Sports

Add / Remove Users

Name	Email	Role
Jim Jimmers	jim@jimmers.com	Coach
Matt Boyd	matt@me.com	athlete
April Shoures	april@me.com	Athletic Trainer
katelyn	katelyn@me.com	athlete
<input type="text" value="type name"/>	<input type="text" value="type email"/>	<input checked="" type="button" value="Athlete"/> <input type="button" value="View Only"/> <input type="button" value="Athletic Trainer"/> <input type="button" value="Coach"/> <input type="button" value="System Admin"/>

Cancel Changes Log Out

www.5guysAnalyze.com

Admin Bob Katz

Workouts
Injuries
Roster
View Data
Accounts
Sports

Add additional permissions for player: Matt Boyd

Roster
 Attendance
 Workout Progress
 Update Injuries

Cancel Changes Log Out

www.5guysAnalyze.com

Admin Bob Katz

Workouts
Injuries
Roster
View Data
Accounts
Sports

Add / Remove

Sports	Number of Coaches
Football	8
Soccer	2
Basketball	1
Baseball	1

Log Out

www.5guysAnalyze.com

Admin Bob Katz

Workouts
Injuries
Roster
View Data
Accounts
Sports

Sports	Number of Coaches
Football	8
Soccer	2
Basketball	1
Baseball	1
Xtreme Hockey	7

Cancel Changes

Log Out

5.0 Non-Functional Requirements

5.1 Usability

When the Coach is interacting with the system to analyze data, it was requested that the program look and feel like Microsoft Excel. Many users already know how to use Excel, so this will make the program easier to learn. The browser application will present the data in a manner similar to the Excel format. The result will make the program easy to learn for novice users with little to no technology experience. Coaches can add data by clicking a specific cell, which opens the program's edit mode.

Athletes will be able to record their data with a mobile application. The application will display each activity for the daily workout in a form containing boxes in which to input sets, reps, and weights for each exercise. The system will also have functionality for including nutrition information and sleep records.

5.2 Accessibility

To access the system, the user must have a registered account within the system database. There are several levels of accessibility that determine which users are allowed to access certain features of the system. For example, an athlete accessing the system will only be able to add workout data, while a coach is able to create and modify workouts. The system includes customization to support injuries, allowing for the creation of special workout routines solely available to an injured athlete. At this time, our system does not support other languages.

5.3 Availability

The system will be available 24/7 for athletes and coaches to access on their mobile devices and web-browser. The system is dependent on coaches updating workout routines for their athletes. However, past workout data and injuries should always be available. If the system goes down due to maintenance or upgrades, athletes could still log the data at another time if they could record their workout on paper.

5.4 Documentation and Training

The system will include proper Documentation and Training to assure its complete implementation and usage.

5.4.1 Documentation

Our system will include documentation in the form of a manual for users to learn how to properly use the system. The manual will include the necessary information for each user role to perform each of their respective functions. For example, when a coach wants to create a workout, he or she will be able to read a document that details the necessary steps of the creation process.

5.4.2 Training

We seek to create a system that is very accessible to its users and as such hope to minimize any training that may be required. However, the future System Administrator may require some degree of training from the original system creators in order to have adequate knowledge of the system's functionality. We do not expect that athletes, athletic trainers, or coaches will require any training beyond the documentation that has been supplied to them.

5.5 Performance

Data entered into the application should appear as the user's types. When athletes enter data, the data will be stored instantaneously. Other system users who modify data will click a "save" button to submit the data changes. The amount of time for this operation should be minimal depending on the amount of traffic going through the system.

As described in the Availability section of our report, the system will be available 24/7 for athletes and coaches. However, there may be certain time periods of the day when more users are likely to access the system. For example, if a Coach requires his/her Athletes to workout together at a specified time, there will be many users accessing the system at the same time. Therefore, the system needs to be scaled to allow for groups of users roughly the size of an athletic team to simultaneously access the system.

5.6 Capacity

The system should be able to support any number of given athletes at a school. Each additional student athlete should not cause the system to slow down or become unresponsive, assuming adequate hardware is provided.

The system should be able to accommodate for updating the sports as new student athletes are added each year. This will be the main area of growth within the system. However, if the University adds a sport to the athletic department, or if another coach within the University decides to use the system, there needs to be space to accommodate the addition of new sports and all the coaches, athletes, and athletic trainers involved.

5.7 Security

The most crucial data that will be stored in the system is the workout data of Athletes. The customer has expressed his desire for this information to be protected from any changes once submitted. Therefore, once an athlete submits workout data, no user will be given permission to modify this data in any way. There are other areas of data security involved as well. To meet security needs, permissions have been created and are assigned to users according to their position. For example, only the Athletic Trainer will be given permission to create or modify injury information. These permissions will restrict users from accessing or modifying data they are not supposed to interact with.

Network security will be maintained by only allowing authorized users to create accounts and log in. Account authorization is only available to System Admins. Individual users are not allowed to change personal permissions. The system should be accessible off campus.

Privacy and data confidentiality are enforced through the permissions given to each user. Athletes will not be given permission to see another athlete's workout data or personal information. The System Admin will be given the ability to set these permissions accordingly. Since permissions will be given to each user specifically, unauthorized access will not be allowed to happen in the first place.

5.8 Longevity

The customer may expect this system to remain in service as long as the final product remains accessible in the Taylor University network. Due to the nature of being academic assignment, further modification, updating, and possible bug fixing will not be able to be carried out by the original creators of the system. Any future maintenance will have to be carried out by the System Administrator.

6.0 Collaborating Systems and Software

Excluding Operating Systems and Internet Browsers, this system will only need to interact with Microsoft Excel. Workout data will need to be exported to Excel as often as a Coach desires to analyze data more thoroughly. This workout data can include workout information from an Athlete, a group of Athletes, or a whole team. The user interface will include a button for the Coach to download the Athlete's data. This will pull data from the database and compile an Excel file. This will be one-way connectivity. Data modified in the Excel file will not be able to upload back to the database.

7.0 Schedule

Milestone	Estimated Time to Complete (Weeks)	Estimated Completion Date	Total Time Elapsed (Weeks)
Webpage framework created	3	2/24/2014	3
Test for Mobile Design Compatibility	0.75	3/1/2014	3.75
Database framework created	2.5	3/18/2014	6.25
Implement Permissions	2.5	4/4/2014	8.75
Test Accounts Created for Trial Run	1	4/11/2014	9.75
Database / Website expanded to accommodate multiple sports	2	4/25/2014	11.75
Send to Customer for Functionality Approval	1	5/2/2014	12.75
Test Security	0.75	5/7/2014	13.5
Bug Fixing	2	5/21/2014	15.5
Final Implementation		5/21/2014	15.5