

RecDawgs: A Recreational Sports Management System

CSCI 4050 SPRING 2016

SHAEN DEIMLING

ALEX MATTE SANTOS

MOLLY ORR

SPENCER ROBINSON

NINA SMITH

1. Introduction	3
2. Current system.....	4
3. Proposed system	4
3.1 Overview	4
3.2 Functional requirements	4
3.3 Non-functional requirements.....	4
3.4 Constraints	5
3.5 System models.....	5
3.5.1 Scenarios.....	5
3.5.2 Use case model	5
Use case diagram	6
CreateUser.....	7
LogIn	8
LogOut	9
CreateTeam	10
JoinTeam	11
LeaveTeam	12
PromoteCaptain	13
CreateLeague.....	14
UpdateLeague.....	15
DeleteLeague.....	16
CreateMatchups.....	17
DeclareLeagueWinner.....	18
GetStandings	19
Schedule match.....	20
SubmitScore.....	21
DecideScore.....	22
CreateVenue	23
UpdateVenue	24
DeleteVenue	25
3.5.3 Domain object model	26
3.5.3.1 Data Dictionary	26
User.....	26

Team	26
League.....	27
Match.....	27
Schedule.....	28
3.5.3.2 Class diagrams.....	28
3.5.4 Dynamic models.....	28
3.5.5 User interface.....	28
SC-NAV-LOGIN-1	29
SC-CREATE-ACCOUNT-1	30
SC-ADMIN-ACCT-1.....	31
SC-CREATE-TEAM-1	32
SC-CREATE-LEAGUE-1	33
SC-UPDATE-LEAGUE-1	34
SC-DECLARE-WINNER-1.....	35
SC-MANAGE-MATCHES-1	36
SC-UPDATE-VENUE-1	37
SC-LEAVE-TEAM-1	38
SC-JOIN-TEAM-1.....	39
SC-DELETE-TEAM-1	40
SC-DELETE-LEAGUE-1	41
4. Glossary.....	41

1. Introduction

The University of Georgia has several thousand students involved in their recreational sports program. This program, which encompasses several leagues for various sports, has continued to grow, straining the capabilities of the existing system. Currently, administrators must manually manage the leagues, teams, and venues but this arrangement has become too inflexible to accommodate the rapidly growing number of participants.

The new system must reduce the dependence on administrators for manual curation and instead allow students themselves to create and register for teams, matchup schedules should be automatically generated after a league becomes active. It must also allow students to schedule the date and time of their matches, submit the final scores, and access records of teams' current and past performances. Finally, to facilitate the use of the new system it must be accessible over the internet.

2. Current system

3. Proposed system

3.1 Overview

3.2 Functional requirements

Students

1. Students can create their own accounts
2. Students can create their own teams in their desired league
3. Students can join a desired team, provided there is room on the roster
4. Students can leave their team
5. Students can view team win/loss statistics from any league

Team Captains

6. Team captains can schedule the date and time of their teams matches
7. After matches have been played, team captains can enter scores
8. The student which creates a team is automatically designated as the team captain

Administrators

9. Administrators can create leagues
10. Administrators can update league information
11. Administrators can delete leagues
12. If there is a disagreement over the final score of a match then the final score can be set by the administrator.
13. Administrators can create new venues
14. Administrators can update venues
15. Administrators can delete venues
16. Administrators set the venues that are available to a particular league
17. If a team captain leaves their team then an administrator can promote another member of the team to team captain

3.3 Non-functional requirements

System must allow multiple concurrent users.

3.4 Constraints

System must use the MySQL relational database management system and will be implemented using the Java programming language. The system must be accessible over the internet.

3.5 System models

3.5.1 Scenarios

Student joins and leaves a team

1. John logs into the system and looks at the leagues that are still in the registration phase
2. John picks a league that interests him and the system returns all of the teams in that league that are available to join
3. He picks one of these teams and clicks the Join Team button
4. After a few weeks he decides that he no longer wants to be part of the team
5. John logs back into the system, navigates to his team's page and clicks the Leave Team button.

Student creates a team

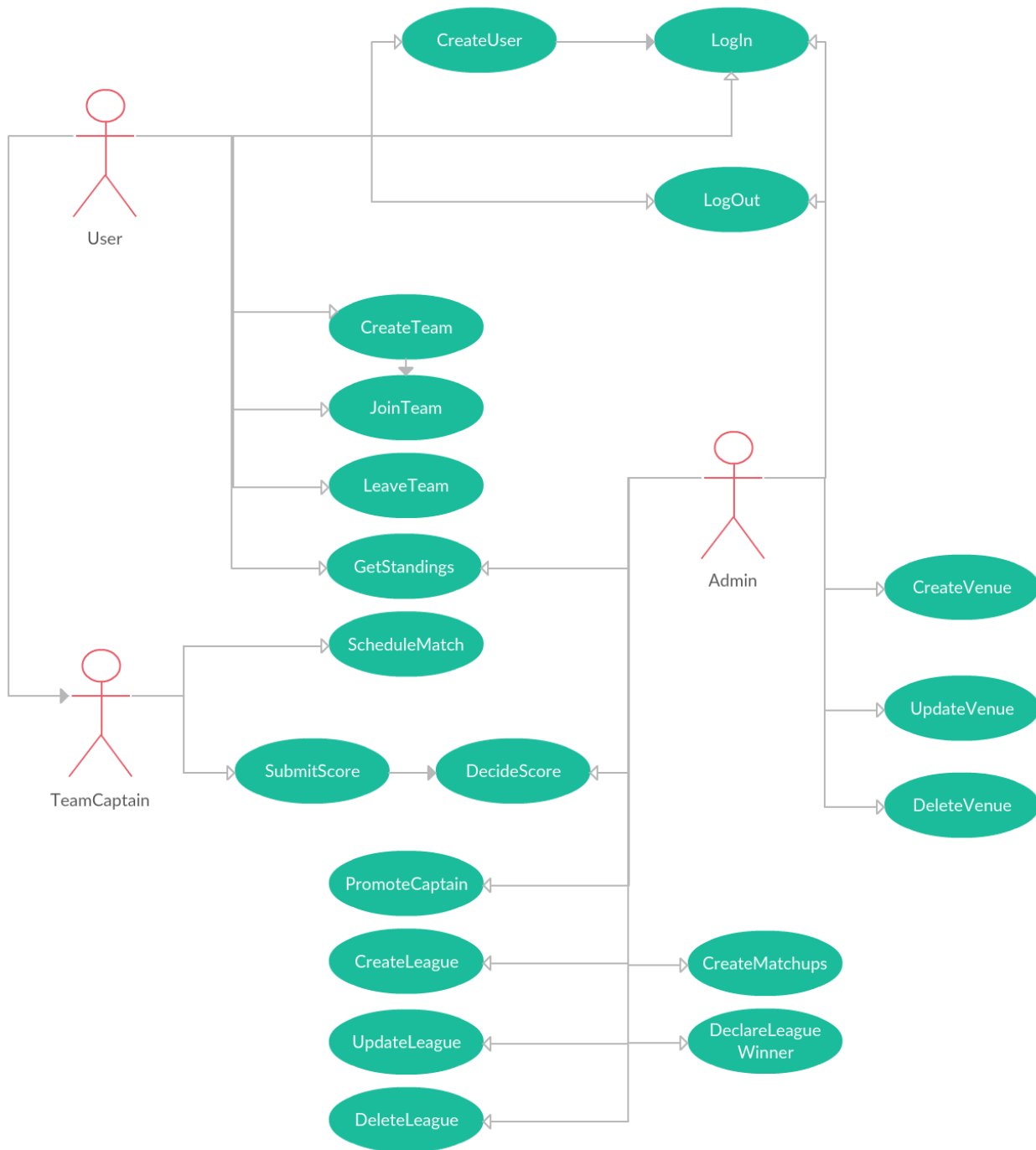
1. Katie logs into the system and looks at the leagues that are still in the registration phase
2. She doesn't find a team she likes and therefore opts to create her own
3. She clicks the Create Team button and is taken to the New Team page
4. She provides a name for her team and clicks the Submit button
5. After creating the new team Katie automatically becomes the new team captain

Administrator creates a new venue

1. The administrator selects the "Add new venue..." link on their account page.
2. The system redirects the admin to a web page requesting the information needed to create a new venue.
3. The admin enters the venue's name, address, and whether the venue is an indoor or outdoor with a drop down list. The admin clicks checkboxes next to each league which they desire to be able to use the venue.
4. The system verifies that the information entered is valid and does not already exist in the system.
5. The admin presses the "Submit" button once they have verified the data entered.
6. The system adds the venue to the current list of venues and notifies the admin. The admin is then redirected to their account page.

3.5.2 Use case model

Use case diagram



Name	CreateUser
ID	CREATE-USER-1
Version	1.0
Author	Shaen Deimling
Date	02/02/2016
Summary	Allows an actor to create a new account.
Basic Path	<ol style="list-style-type: none"> 1. The user will be prompted for their name, password, student number, college major, email address, and residence address. 2. The system will verify that the information entered by the student is valid. 3. The student will hit the "Submit" button. 4. The system will create a new account with the information provided and notify the user of the account's successful creation. 5. The user will be redirected to the login screen.
Alternative Paths	<ol style="list-style-type: none"> 1. In the case that the user presses the Cancel button, the system will terminate the current transaction and discard any information the user has input.
Exception Paths	<ol style="list-style-type: none"> 1. If an account has already been created with the information entered, the system will notify the user and will not create a new account.
Extension Points	
Triggers	The user has selected "New User" from the login screen.
Assumption	The user does not already have an account.
Pre-conditions	
Post-conditions	The user has an account with their information and a username/password combination with which they can access said account.

Name	Login
ID	LOGIN-1
Version	1.2
Author	Spencer Robinson
Date	02/02/2016
Summary	Allows existing user to gain access to system
Basic Path	<ol style="list-style-type: none"> 1. System displays prompts for the user.email and user.password. The login button is displayed. 2. User enters their user.email and user.password and presses login button. 3. System locates User's data and redirects them to the main page.
Alternative Paths	None
Exception Paths	<ol style="list-style-type: none"> 1. If in step 3 the system fails to verify the user's access rights, System will display UnknownUserMessage and terminate the use case.
Extension Points	
Triggers	User presses the "Login" button on the web page.
Assumption	
Pre-conditions	The user already has an account.
Post-conditions	The user is now able to access user-specific actions through the website.

Name	LogOut
ID	LOGOUT-1
Version	1.1
Author	Nina Smith
Date	02/02/2016
Summary	This use case details the logout steps of a user.
Basic Path	1. User presses the Logout button on the web page.
Alternative Paths	1. If the user is inactive for more than 15 minutes, the system will automatically trigger this use case.
Exception Paths	
Extension Points	
Triggers	The user elects to Logout or they have been idle for 15 minutes.
Assumption	The user has an account.
Pre-conditions	The user is Logged In
Post-conditions	The user is Logged Out and will no longer be able to access user specific functions through the website.

Name	CreateTeam
ID	CREATE-TEAM-1
Version	1.1
Author	Alex Matte Santos
Date	02/02/2016
Summary	A user creates a new team for one of the available leagues and becomes its captain.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks the “Create new team...” link from their account page. 2. The system displays the team creation page (SC-CREATE-TEAM-1) and provides a list of leagues in which a new team can be created. 3. User selects a league from a drop down list 4. User provides a valid team.name. 5. The system notifies the user if the team name is not valid or is already in use. 6. User clicks on Submit button once the system notifies them that the name is valid. 7. The system registers the new team in the selected league and automatically makes the user the team captain.
Alternative Paths	<ol style="list-style-type: none"> 1. If the user clicks the Cancel button the use case is terminated and the user is navigated to their account page.
Exception Paths	<ol style="list-style-type: none"> 1. If a league becomes full during the process, and the user presses the Submit button, the system will notify the user that the team was unable to be created and refresh the page. 2. If the user presses the Submit button while the team name is invalid, the system will notify the user of the issue and remain on the Create Team page.
Extension Points	
Triggers	User elects to create a new team from their account page.
Assumption	There is at least one league with an empty spot for a team.
Pre-conditions	User has logged into the system and is a student.
Post-conditions	New team has been created for the given league and the user is the team’s captain.

Name	JoinTeam
ID	JOIN-TEAM-1
Version	1.1
Author	Shaen Deimling
Date	02/02/16
Summary	A user can join a pre-existing team if the team has not already reached maximum capacity and the user is not already on a team in that particular league.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks the “Join Team” button on their account page. 2. The system will redirect the use to a page where they are able to select what team they would like to join. The system will retrieve a list of leagues with teams that the user can join. 3. The user will select a league from the current list of leagues presented by the system as a drop down box. This list will not include any leagues in which the user is currently participating. 4. The system will retrieve the list of teams that the user is able to join in the league that was selected. This list will be present as another drop down box. 5. The user will select a team from the list and confirm their choice by pressing the “Submit” button at the bottom of the screen. 6. The system will add the player to the team roster and notify the user of their successful addition to the team. The user will then be directed back to their account web page.
Alternative Paths	<ol style="list-style-type: none"> 1. If at any point the user wishes not to join a team, then they may opt to press the “Cancel” button at the bottom of the screen. This will cancel the operation and redirect the user back to their account page.
Exception Paths	<ol style="list-style-type: none"> 1. If a team is filled during the operation, the system will will not be able to add the player to the team. Step 6 will instead notify the user that the team has been filled and refresh the page.
Extension Points	
Triggers	The user clicks the “Join Team” link on their account page.
Assumption	There is at least one team available for the user to join.
Pre-conditions	The user is a student, has an account, and is logged in.
Post-conditions	The user is now a member of the team they selected.

Name	LeaveTeam
ID	LEAVE-TEAM-1
Version	1.1
Author	Molly Orr
Date	02/02/16
Summary	A user can leave a team that they have joined.
Basic Path	<ol style="list-style-type: none"> 1. User clicks the "Leave Team" button on their account page. 2. User is redirected to a page with a dropdown list showing the teams of which they are members 3. User is able to select the team they wish to leave and presses submit to confirm their selection. 4. System will remove user from selected team roster and notify user of a successful removal. 5. System will redirect user to their account page.
Alternative Paths	<ol style="list-style-type: none"> 1. If at any point the user wishes not to leave a team, then they may opt to press the "Cancel" button at the bottom of the screen. This will cancel the operation and redirect the user back to their account page.
Exception Paths	<ol style="list-style-type: none"> 1. If the user is not a member of any teams, the system will display an error message and redirect back to the user's account page
Extension Points	If the user is the captain of the team they wish to leave, extend to the PROMOTE-CAPTAIN-1 UC. If the team is below the league minimum, the team becomes inactive.
Triggers	User clicks "Leave Team" on their account page.
Assumption	
Pre-conditions	User must be a student, be logged in, and be a member of the team they wish to leave.
Post-conditions	User is no longer a member of the team they left.

Name	PromoteCaptain
ID	PROMOTE-CAPTAIN-1
Version	1.1
Author	Nina Smith
Date	02/02/2016
Summary	This use case details the steps necessary for the ADMIN to promote a TEAM MEMBER to the TEAM CAPTAIN in the event that a new team without a captain is created or a team captain leaves a team.
Basic Path	<ol style="list-style-type: none"> 1. ADMIN presses a Change Team Captain button on their account page 2. This button will prompt the ADMIN to choose a league in which the team is a part of in a dropdown list 3. The ADMIN will then choose the team in which the member whom they wish to promote to captain is in from another dropdown list 4. The ADMIN will then choose the team member from a dropdown list 5. After choosing the member the ADMIN must select either the Submit button or the Cancel button at the bottom of the page to confirm their selection
Alternative Paths	<ol style="list-style-type: none"> 1. If the ADMIN presses the Cancel button then they are redirected to the team member dropdown list
Exception Paths	
Extension Points	
Triggers	ADMIN elects to promote a team member to the team captain
Assumption	
Pre-conditions	User is ADMIN There are team members
Post-conditions	The team has a CAPTAIN

Name	CreateLeague
ID	CREATE-LEAGUE-1
Version	1.1
Author	Molly Orr
Date	02/02/2016
Summary	This use case details the necessary steps for an ADMIN to create a league for a certain sport.
Basic Path	<ol style="list-style-type: none"> 1. The ADMIN selects “Create League” from their account page. 2. The admin enters the sport for the league, the minimum number of teams to be valid, the maximum number of teams that can enter, and league rules (SC-CREATE-LEAGUE-1). 3. The admin hits the submit button to create the league and receives confirmation of successful league creation. 4. The admin is returned to their account page.
Alternative Paths	<ol style="list-style-type: none"> 1. The admin selects the Cancel button rather than the Submit button. 2. The action is canceled. 3. The admin is returned to their account page.
Exception Paths	<ol style="list-style-type: none"> 1. If the league for a certain sport already exists, and error message is displayed. 2. The league is not able to be submitted until a new, valid league is submitted.
Extension Points	
Triggers	ADMIN selects “Create League” from account page.
Assumption	
Pre-conditions	A league for the sport does not exist, the user is an Administrator, and is logged in.
Post-conditions	A new league exists, open for teams to join.

Name	UpdateLeague
ID	UPDATE-LEAGUE-1
Version	1.0
Author	Molly Orr
Date	02/02/2016
Summary	This use case details the steps necessary for an ADMIN to update league information.
Basic Path	<ol style="list-style-type: none"> 1. The ADMIN selects "Update League" from their account page. 2. The admin selects a league from a dropdown menu 5. The admin can change the minimum number of teams to be valid, the maximum number of teams that can enter, and league rules. 6. The admin hits the submit button to update the league and receives confirmation of successful league creation. 7. The admin is returned to his account page.
Alternative Paths	<ol style="list-style-type: none"> 1. The admin can select to "Cancel" the update. 2. The admin is returned to his account page.
Exception Paths	<ol style="list-style-type: none"> 1. If the admin enters any invalid information, the system will not allow an update until the information is valid.
Extension Points	
Triggers	The "Update League" button is selected by an admin.
Assumption	
Pre-conditions	The league existed, the user is an admin, and is logged in.
Post-conditions	The league has new information.

Name	DeleteLeague
ID	DELETE-LEAGUE-1
Version	1.1
Author	Molly Orr
Date	02/02/2016
Summary	A use case to describe the steps for an ADMIN to delete a league from the system.
Basic Path	<ol style="list-style-type: none"> 1. The ADMIN selects “Update League” from their account page. 2. The ADMIN presses the Delete League button to delete the league and receives confirmation of successful league deletion. 3. The ADMIN is returned to their account page.
Alternative Paths	
Exception Paths	<ol style="list-style-type: none"> 1. The user selects “Submit” rather than “Delete League.” 2. The league is updated rather than deleted.
Extension Points	<ol style="list-style-type: none"> 1. UpdateLeagueUC if submit selected.
Triggers	<ol style="list-style-type: none"> 1. The “Delete” button is selected after “Update League” is selected.
Assumption	
Pre-conditions	<ol style="list-style-type: none"> 1. The league exists, the user is an admin, is logged in and has selected “Update League”.
Post-conditions	<ol style="list-style-type: none"> 1. The league no longer exists.

Name	CreateMatchups
ID	CREATE-MATCHUPS-1
Version	1.1
Author	Alex Matte Santos
Date	02/02/2016
Summary	After a league becomes active an admin can create a league's matchup schedule.
Basic Path	<ol style="list-style-type: none"> 1. The system shows the CREATE MATCHUPS page 2. The admin selects a league from the dropdown menu 3. The admin clicks the CREATE MATCHUPS button 4. The system creates the schedule and populates it with the appropriate number of matches. Each match will have two teams and a venue.
Alternative Paths	<ol style="list-style-type: none"> 1. If there are an odd number of teams, each team will have a bye week.
Exception Paths	<ol style="list-style-type: none"> 1. If there are no venues available, the location listed for each match will be stated as "Unknown".
Extension Points	
Triggers	The admin selects the "Create league matchups..." link from their account page.
Assumption	There is at least one venue available for the league.
Pre-conditions	There are an even number of teams in the league and there are at least 2 teams. The league is active.
Post-conditions	league.registration is set to closed. schedule for a league is created and populated with multiple matches .

Name	DeclareLeagueWinner
ID	DECLARE-LEAGUE-WINNER-1
Version	1.1
Author	Alex Matte Santos
Date	02/02/2016
Summary	After all of the particular matches have been played the admin chooses a team as the overall winner of the league in the event of a tie or dispute.
Basic Path	<ol style="list-style-type: none"> 1. The administrator presses the “Declare league winners...” link on their account page. 2. System shows Declare Winner Page (SC-DECLARE-WINNER-1) 3. Admin selects desired league from dropdown box 4. Admin selects winning team from dropdown box 5. Admin clicks Submit button 6. league.winner is set to the chosen team 7. league.status is set to closed
Alternative Paths	<ol style="list-style-type: none"> 1. At any point, the Administrator may choose to click the “Cancel” button. This will redirect the user to their account page.
Exception Paths	
Extension Points	
Triggers	Admin wishes to declare a winner for a particular league
Assumption	All matches in a league have been played
Pre-conditions	The league is active and all matches have been played.
Post-conditions	A team is declared the winner of a league, and the league becomes inactive.

Name	GetStandings
ID	GET-STANDINGS-1
Version	1.0
Author	Alex Matte Santos
Date	02/02/2016
Summary	A user can get a summary of the teams and their statistics in a league.
Basic Path	<ol style="list-style-type: none"> 1. System shows users home page 2. Users clicks on View Leagues button 3. User selects desired league from dropdown menu 4. User clicks on View Standings button
Alternative Paths	<ol style="list-style-type: none"> 1. If after step 2 the user navigates away from the page then the use case terminates
Exception Paths	
Extension Points	
Triggers	User wishes to see the standings for a particular league
Assumption	
Pre-conditions	
Post-conditions	

Name	Schedule match
ID	SCHEDULE-MATCH-1
Version	1.0
Author	Spencer Robinson
Date	02/02/2016
Summary	Details the steps a Captain may take to schedule a match between his team and a team determined by CreateMatchups
Basic Path	<ol style="list-style-type: none"> 1. System displays the Manage Matches page with a calendar populated with dates and times available for a match to be scheduled, dates and times of scheduled and previous matches, calendar navigation buttons (to move to next week, month, etc.), and a “Schedule Match” button. 2. Captain decides on an appropriate date and time for the match to take place, and clicks to select that time in the calendar. 3. Captain presses “Schedule Match” to submit the request to schedule a match at the chosen time.
Alternative Paths	
Exception Paths	<ol style="list-style-type: none"> 1. If the Captain attempts to select a time that is unavailable, the System will display a message stating the chosen time is unavailable for play.
Extension Points	
Triggers	Captain presses the “Schedule Match” button.
Assumption	
Pre-conditions	<ol style="list-style-type: none"> 1. CreateMatchups has successfully run. 2. User is a Captain.
Post-conditions	Match is scheduled at the indicated date and time, between the Captain ’s team and the team decided by CreateMatchups.

Name	SubmitScore
ID	SUBMIT-SCORE-1
Version	1.0
Author	Spencer Robinson
Date	02/03/2016
Summary	Allows team captains to submit the final score of a game they have participated in. Both captains must submit the same score for that score to be verified. In the event of a dispute, the administrator may enter the final score.
Basic Path	<ol style="list-style-type: none"> 1. System displays the Manage Matches page with a calendar populated with dates and times available for a match to be scheduled, dates and times of scheduled and previous matches, calendar navigation buttons (to move to next week, month, etc.), and a "Schedule Match" button. 2. Captain navigates to a match his team has previously participated in. 3. System displays information relevant to the match such as date and time, venue, participating teams, and a "Submit Score" button. 4. Team Captain enters the final score of the game, and presses "Submit Score". 5. The captain of the opposing team will receive a notification that a score has been submitted, and will be prompted to verify that score.
Alternative Paths	
Exception Paths	<ol style="list-style-type: none"> 1. If Captain enters an invalid score (outside the range of possible scores for a match of that particular sport), System displays message indicating an impossible score. 2. If clicking Submit Score for a game which has already had a score verified, System displays message that game score has been finalized and is unchangeable. 3. If User clicking Submit Score is not a Captain, System displays message that User must be a captain to Submit Score.
Extension Points	
Triggers	Captain hits the "Submit Score" button.
Assumption	
Pre-conditions	User is a Captain.
Post-conditions	Score is submitted for verification.

Name	DecideScore
ID	DECIDE-SCORE-1
Version	1.0
Author	Spencer Robinson
Date	02/03/2016
Summary	In the event of a dispute between team captains as to the final score of a match, DecideScore allows the administrator to submit the final score of a match.
Basic Path	<ol style="list-style-type: none"> 1. System displays the Manage Matches page with a calendar populated with dates and times available for a match to be scheduled, dates and times of scheduled and previous matches, calendar navigation buttons (to move to next week, month, etc.), and a “Schedule Match” button. 2. Administrator navigates to the match in question. 3. System displays information relevant to the match such as date and time, venue, participating teams, and a “Decide Score” button (only displayed to users with administrator permissions). 4. Administrator presses “Decide Score” to submit the final score. 5. Administrator enters the final score.
Alternative Paths	
Exception Paths	<ol style="list-style-type: none"> 1. If Administrator enters an invalid score (outside the range of possible scores for a match of that particular sport), System displays message indicating an impossible score.
Extension Points	
Triggers	Administrator hits the “Decide Score” button.
Assumption	
Pre-conditions	The user is an administrator and is logged in.
Post-conditions	A final score is verified for the match in question.

Name	CreateVenue
ID	CREATE-VENUE-1
Version	1.0
Author	Shaen Deimling
Date	02/02/2016
Summary	Allows the ADMIN to enter a new venue where matches can be played.
Basic Path	<ol style="list-style-type: none"> 1. The administrator selects the “Add new venue...” link on their account page. 2. The system redirects the user to a web page requesting the information needed to create a new venue. 3. The user enters the venue’s name, address, and whether the venue is an indoor or outdoor with a drop down list. The user clicks checkboxes next to each league which they desire to be able to use the venue. 4. The system verifies that the information entered is valid and does not already exist in the system. 5. The user presses the “Submit” button once they have verified the data entered. 6. The system adds the venue to the current list of venues and notifies the user. The user is then redirected to their account page.
Alternative Paths	<ol style="list-style-type: none"> 1. If the user decides not to create a new venue, they may press the Cancel button, which redirect them to their account page.
Exception Paths	<ol style="list-style-type: none"> 1. If the user presses the Submit button while any of the input boxes are empty or contain invalid values, the system will notify the user and remain on the same page.
Extension Points	
Triggers	An administrator clicks the “Add new venue...” link.
Assumption	
Pre-conditions	The user is an administrator and is logged in.
Post-conditions	The new venue is added to the current list of venues.

Name	UpdateVenue
ID	UPDATE-VENUE-1
Version	1.0
Author	Shaen Deimling
Date	02/02/2016
Summary	The administrator can update any information about a pre-existing venue.
Basic Path	<ol style="list-style-type: none"> 1. The administrator selects the "Update venue information..." link on their account page. 2. The system redirects the user to a page where the league information is displayed and able to be modified. 3. The user is able to change the venue name and address. The user can also change the venue from indoor to outdoor. The user can also change what leagues are able to use the venue by clicking the checkboxes at the bottom of the page. 4. The system verifies all user input and provides immediate feedback. 5. The user presses the Submit button once all information is valid and correct. 6. The system updates the information in the database.
Alternative Paths	<ol style="list-style-type: none"> 1. At any time, the user may select to cancel the operation by pressing the Cancel button. This will redirect the user to their account page.
Exception Paths	<ol style="list-style-type: none"> 1. If the venue is deleted during this operation, and the submit button is pressed, the user will be notified and redirected to their account page. 2. If the user presses the submit button while there is invalid information on the page, the system will notify the user and allow them to correct the information before pressing the Submit button.
Extension Points	
Triggers	The administrator selects the "Update venue information..." link on their account page.
Assumption	
Pre-conditions	The user is an administrator and is logged in. The venue must already exist.
Post-conditions	The venue information is updated according to the changes made by the administrator.

Name	DeleteVenue
ID	DELETE-VENUE-1
Version	1.1
Author	Nina Smith
Date	02/02/2016
Summary	This use case details the steps the ADMIN goes through to delete a venue
Basic Path	<ol style="list-style-type: none"> 1. The administrator selects the "Delete venue" link on their account page. 2. The administrator selects a venue from a drop down list. 3. The administrator presses the Submit button at the bottom of the page. 4. The system removes the venue from the current list of venues and redirects the administrator to their account page.
Alternative Paths	<ol style="list-style-type: none"> 1. If the user presses the Cancel button, they will be redirected to their account page.
Exception Paths	
Extension Points	
Triggers	The ADMIN elects to delete a previously created venue.
Assumption	There is no league with matches scheduled at the venue.
Pre-conditions	The venue exists. User is logged in as an administrator.
Post-conditions	The venue is removed from the current list of venues.

3.5.3 Domain object model

3.5.3.1 Data Dictionary

Name	User
Description	Represents a student registered with the system
Attributes	<ul style="list-style-type: none">• name• password• studentNumber• major• email• address
Static relationships	<ul style="list-style-type: none">• A user can belong to 0 or many teams

Name	Team
Description	Represents a team of players registered with a particular league
Attributes	<ul style="list-style-type: none">• name• league• captain• members• wins• losses• pointsFor• pointsAgainst
Static relationships	<ul style="list-style-type: none">• A team can have 1 to many players• A team can only belong to 1 league

Name	League
Description	Represents a group of teams playing a particular sport
Attributes	<ul style="list-style-type: none"> ● name ● venues ● schedule ● registration ● status
Static relationships	<ul style="list-style-type: none"> ● A league can contain 0 to many teams ● A league can use 1 to many venues

Name	Match
Description	Represents a match that was played between two teams in a league
Attributes	<ul style="list-style-type: none"> ● date ● time ● venue ● homeTeam ● awayTeam ● homeTeamScore ● awayTeamScore ● winner ● loser
Static relationships	<ul style="list-style-type: none"> ● A match must be part of 1 schedule ● A match must have 1 venue ● A match has 2 teams

Name	Schedule
Description	Represents all of the matches that will be played during the season of a league
Attributes	<ul style="list-style-type: none">● league● rounds● matches
Static relationships	<ul style="list-style-type: none">● A schedule is used by one league

3.5.3.2 Class diagrams

3.5.4 Dynamic models

3.5.5 User interface

Logo and Website Navigation Pane

Username:

Password:

Logo and Website Navigation Pane

Name:	<input type="text"/>
Password:	<input type="password"/>
Confirm Password:	<input type="password"/>
Student Number:	<input type="text"/>
Email Address:	<input type="text"/>
College Major:	<input type="text"/>
Residential Address:	<input type="text"/>

Logo and Website Navigation Pane

Name: User.name

Change account information...

Create venue...

Update venue...

Delete venue...

Create league...

Delete league...

Decide league winner...

Decide match score...

Decide team captain...

Logo and Website Navigation Pane

Create New Team

Select League:

Please select a league

Team Name:

Logo and Website Navigation Pane

Create New League

League Name:

Minimum # of Teams:

Maximum # of Teams:

League Rules:

Submit

Cancel

Logo and Website Navigation Pane

Update League

League Sport:

League Name:

Minimum # of Teams:

Maximum # of Teams:

League Rules:

Logo and Website Navigation Pane

Declare League Winner

Select League:

Please select a league

Select Team:

Please select a team

Cancel

Submit

Logo and Website Navigation Pane

February 2016						

Schedule Match

Submit Score

Decide Score

Logo and Website Navigation Pane

Update League

League Sport:

League Name:

Minimum # of Teams:

Maximum # of Teams:

League Rules:

Logo and Website Navigation Pane

Leave Team

Select team:

Please select a team


Cancel

Leave

Logo and Website Navigation Pane

Join Team

Select team:

Please select a team 

Cancel

Join

Logo and Website Navigation Pane

Delete Team

Select team:

 ▾

Logo and Website Navigation Pane

Delete League

Select league:

Please select a league

Cancel

Submit

4. Glossary