	PROJECT DOCUMENTATION EXPLAINATION
	Document: 10-JLJ-00-01A
	Title: Project Documentation Explanation
	Team: JLJ TEAM
	Effective Date: April-28-2010

Purpose: this document provides the Explanation for each Documentation.

JLJ Team Project Documentation Explaination:

Session 1 is in 10-JLJ-01-01A_Project Selection_.pdf or jljProject(old)

Session 2 is in 10-JLJ-02-01A_User Specification_.pdf and 10-JLJ-02-02A_Techinical Requirement_.pdf or jljProject(old)

Session 3 is in 10-JLJ-02-01A_User Specification_.pdf and 10-JLJ-02-02A_Techinical Requirement_.pdf or jljProject(old)

Session 4,5 is in 10-JLJ-03-02B_Final Concept for Database_.pdf or jljProject(old)

Session 6 is in 10-JLJ-03-02C_Database Implimetation_.pdf or jljProject(old)

Session 7 to 10 is in 10-JLJ-03-02B_Final Concept for Database_.pdf

Session 11 is in 10-JLJ-03-02C_Database Implimetation_.pdf

Session 12 is in 10-JLJ-03-03B_Final Concept in Application_.pdf

Session 13 and 14 is in 10-JLJ-01-03A_Project Team Work by Liyi Li_.pdf and 10-JLJ-01-04A_Project Team Work and Revise By Jon_.pdf and 10-JLJ-01-05A_Project Team Work and Revise By Jacob_.pdf

Session 15 is in 10-JLJ-04-01A_Verification Plan_.pdf and 10-JLJ-04-02A_Verification Protocol_.pdf and 10-JLJ-05-01A_Validatiion_.pdf

Session 16 is 10-JLJ-06-02A_Instruction Manual_.pdf

Session 17 is in 10-JLJ-06-01A_User Manual_.pdf

Session 18 is in 10-JLJ-06-03A_Program Manual_.pdf

Session 19 is in code file

REVISION HISTORY

REVISION	DESCRIPTION
A	Original Document

	PROJECT SELECTION
	Document: 10-JLJ-01-0B
	Title: Project Selection
	Team: JLJ TEAM
	Effective Date: Feb-12-2010

Purpose: this document provides our selection of the project.

JLJ Team Project Selection:

Team members: Jacob Emmert-Aronson, Jonathan Klein, Liyi Li
The project proposal for our team is Fantasy football

Overview:

Fantasy football is a widely popular game that's played during every NFL football season. According to ESPN's Colin Cowherd, there are more than 27 million people who play each year. This game, which is often played online, allows participants to compose a team from a list of NFL players to represent themselves, and points are rewarded to these players based on their statistics each week.

Our project team plans to develop a web-based application for this popular online game, which will include a database to keep track of the points each player earns per week, which players are on whose team, which team earns more points compared to others, and other important aspects needed for the game.

The game in the real world will have these kind of properties:

When a user enters the site, they can create an id and password to access the game, and then creates or joins a league. Once they're part of a league they create a team name. A league is a group of a certain amount of players that will compete against each other over the course of the football season.

At a predetermined date chosen by the users in the league, a draft will occur in order to select players to put on their teams. The users in league are put into a random order, and when it is user's turn to choose, they select from a list of available players. The selected player is placed on that user's team roster and is also removed from the list of available players. This continues until each user's team is full.

After all roster spots are filled, a schedule of matchups pitting one team against another is created round robin style, where the matchups change every week. The user can assign players to certain starting positions that will act as the point-getters for that week. The user can view the following throughout the season:

A, their team roaster as well as other users' rosters
B, their and others' matchup schedules
C, each player's stats and points
D, their wins and losses, as well as their current ranking

The user can also replace a player on their team with a player on the list of available players. Or they can trade a player on their team for a player on another user's team if both agree to the deal. Each week, points are tabulated for each player based on their stats, and these points are added together for each user's team. These point totals are then compared to the team they are matched up against the team with the larger number of points receives a 'win' and the other team receives a 'loss'. Rankings are modified based on the new wins and losses data each week. At the end of the season, the team with the best record wins the league.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document
B	Fixed Document

PROJECT PLAN
Document: 10-JLJ-01-01A
Title: Project Plan
Team: JLJ TEAM
Effective Date: Jan-16-2010

Purpose: this document provides a timeline and general plan for the project

JLJ Team Project Plan:

Week	Tasks
BY Jan 16	Project Selection
BY Feb 12	Project Report One • Overview of our project • User Specification • Technical requirement of the project
BY Feb 19	Database Design and Specification and Project Report Two • ER-model for our Database • Relation Model for our Database
BY March 6	Project Report Three • Make the Database • Run the Database in MySQL • Test if it is correct for each Database and rewrite the wrong one
BY April 12	Finish all our design. • Finish the concept design • Select our programming language • Design the web-page application • Fix the database
BY April 25	Verification Testing • Finish the verification plan • Finish the verification protocol • Test all our component in our web-site
BY April 29	• Validation on Classroom • Submit our project

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

PROJECT DIVISION
Document: 10-JLJ-01-03A
Title: Project Team Work and Revise
Team: JLJ TEAM
Effective Date: April-27-2010

Purpose: this document provides a Project Team Work and Revise for Liyi Li

JLJ Team Project Project Team Work and Revise for Liyi Li:

Session 13:

Discuss any problems encountered overall during the whole project progress. Any scaling down needs to be explained here.
This project is not so hard to finish, but the documentation is quite hard for me, since almost all the documentations are finished by me. I think I manage the documents by software engineering style, but there exist many unexcepted situation in the documentation

What else would you have changed given more resources?

The functional dependency theorem makes our design more easily. I should say it is quite useful.

What else would have been useful to add to the system?
May be a multiple league system, it is quite useful to let our game become a real game.

How could your project evolve into a commonly-used application?
Just give us a web server and make more leagues in our game, it will become a web game immediately.

Explicitly state how much of the code is developed by you as a team member.
Almost all the java code is done by me. generate.java and userLogin.java

Session 14
Liyi Li:

- Project Document design
- Project Report One for User Specification and Database Specification
- Project Report Two for Relation Model
- The entire Project Report Three
- Final Project For Project Plan, Project Selection, User Specification, Technical requirement, the documentation of Three Concept Design, the documentation of two Final Concept, Verification and Validation.
- ALL the JAVA code except SelectWinner and RandomStats

Jon does some of the Java code and the documentation of user manual, also the basic idea of the game is provided by Jon

	PROJECT DIVISION
	Document: 10-JLJ-01-03A
	Title: Project Team Work and Revise
	Team: JLJ TEAM
	Effective Date: April-27-2010

Jacob do all the HTML code, it is hard.

what went well as a team and what did not?
Actually, we really need a team concept in the beginning. The most problem is we lose many time on the communication. Usually, it is hard for us to set up a meeting.

Did you have team meetings, and how often?
We have totally three to four times of meeting.

Did you share information effectively?
We use github to share data, this is not a big problem for us.

Was the division of labor fair?
As a team, it happens usually that some people do a lot, other people do a little, I do not care since I want to get A.

Who did what type of decision making?
All the decisions we have done are by negotiation. We usually communicate by email.

Did you disagree/agree on any topic, and how did you resolve them?
No, anything we did so far is good for me. I should say the first art of a team work is compromise.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

PROJECT DIVISION
Document: 10-JLJ-01-04A
Title: Project Team Work and Revise
Team: JLJ TEAM
Effective Date: April-27-2010

Purpose: this document provides a Section 13 and 14 from Jonathan Klein.

JLJ Team Project Team Work and Revise from Jonathan Klein:

13.

Jonathan Klein:

Our biggest problem with developing the application is that in order to earn points for the game, the NFL season has to be in progress since the points are based on players' stats. We decided that the only way we could show off the application in the off-season was to randomize stats ourselves in the code. Also, there are hundreds of players in the NFL. To input every single player into our database would be very tedious and time consuming, so we ended up including only players that are part of their team's starting lineup.

Given more time and resources, we would like to add the possibility of having multiple leagues running at the same time. This would allow many more people to enjoy using the application. We'd also like to include more NFL players so that users have more options to choose from. The game is set up so that each week the users get to ponder roster changes and trades before the NFL games at the end of the week, but right now, the week only changes at the click of a button as opposed to an actual week's worth of time has passed. Another necessary addition would be to include way for the application to recognize the date to keep track of time.

One way for our application to become more commonly used would be to have it connected to something like an iPhone app. This way, users can access it remotely with their phones instead of having to be in front of a computer.

Code-wise, I developed the randomized stats generator and updater. This code will create stats for every player based on their position, and updates their stats each time the method is called (which would happen each week). I also wrote the code that adds up the points for each player, and then determines the winner of each matchup based on those points.

14.

Jonathan Klein:

I came up with the idea of developing a fantasy football website, and acted as a source of information for members of the group who weren't familiar with fantasy football. In designing the application, I drew up page-by-page screen layouts of what the application should look and function like; then as a team we decided on some changes to the original design. I also worked on the Project Progress Reports, including writing and editing. For the JAVA code, I wrote the methods that randomized stats for the players, figured out the points for those stats, randomized injury statuses, selects the winner of the week's matchups, and updates all of this information to the database.

As far as teamwork is concerned, we all worked together fairly well. We were able to easily agree on any changes during the developmental process, and made sure our work was done on time. We had several meetings during the semester to touch base, but mainly communicated through email. Using email and Github, we were able to share any code or documents.

The division of labor was fairly even. I did much of the report organizing and a large portion of the writing, as well as formulating screen layouts. I also worked on a portion of the JAVA code. Liyi put each progress report together and did quite a bit of writing; he also wrote a large amount of the JAVA code, and the majority of the final project report. Jacob worked some on the progress reports, wrote all of the HTML code, and put together all of the parts of the application to get it in running shape. We all helped each other out with whatever we worked on, providing suggestions for changes and editing.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

PROJECT DIVISION
Document: 10-JLJ-01-05A
Title: Project Team Work and Revise
Team: JLJ TEAM
Effective Date: April-27-2010

Purpose: this document provides a Section 13 and 14 from Jacob Emmert-Aronson.

JLJ Team Project Team Work and Revise from Jacob Emmert-Aronson:

Session 13:

In our original design, we intended to have multiple leagues. We quickly discovered, however, that this would add several layers of complexity to the overall project. We ultimately decided that it was better for each instance of the application to run a single league, and that if multiple leagues are desired, the site admin can install multiple copies of the application (most likely on different servers).

We also decided to allow users to draft players in any order instead of forcing them to choose one player at a time. In a production version of the application, it would be better to enforce drafting order, but for the purposes of a demonstration it is far more convenient to allow each user to select all their players at once than to constantly have to log out of and back in to the application.

Another thing that would be useful to add is an interface to allow the admin to fill out player stats using a web form. This was not a priority for us because it is currently the off-season. We elected to instead write a function that assigns weekly statistics randomly.

There are already several fantasy football leagues on the internet. Our project could become an easy way for people who want to start a new league to install a system to keep track of their statistics.

Because neither of the other team members are comfortable with html, I wrote the java server pages for the application. This includes all the html code, as well as a sizeable amount of java code, most noteably to handle cookies and processing forms. I also integrated the java and mysql code that my teammates wrote into this framework.

Session 14:
Jon came up with the original idea of a fantasy football league, and worked out most of the overall design of what we wanted the application to do. He also mapped out the organization into individual web pages and wrote the java code for a few important functions. Liyi wrote most of the documentation and some of the java functions as well as most of the database specification. I wrote the java server pages which act as the application's front end. This includes all of the html code and significant amounts of java to process the html and handle most of the functions of a webserver. I

also integrated the functions Liyi and Jon wrote into an overall framework.

Overall, we worked well as a team. While there were a few potholes to work out, we never had any serious problems working together. The most difficult issue was communication. Because we only met a few times during the semester, we relied primarily on email to communicate. This could be slow at times, and it is often not as easy to tell whether we understand each other than when talking face-to-face. The division of labor seemed equitable and also allowed each of us to play to our own strengths.

We did have a few disagreements, most notably on which programming language to use initially. I initially favored php because I am more familiar with it in the context of web development, while Jon and Liyi wanted to use Java because they are unfamiliar with php. Ultimately we agreed that it was better to go with the language everyone is familiar with, and one of the homework assignments taught how to use java in conjunction with a webserver, rendering the issue entirely moot.

REVISION HISTORY

REVISION	DESCRIPTION
A	Original Document

	USER SPECIFICATION
	Document: 10-JLJ-02-01A
	Title: User Specification
	Team: JLJ TEAM
	Effective Date: Feb-12-2010

Purpose: this document provides basic requirement of the project, but not including functional and technical specification.

JLJ Team User Specification:

Functional specification:

The user specification is obvious, since we have a long description about the Fantasy football game and we just need to simulate the game from real world. However, in order to simulate it in our computer easily, our team decides to change a little bit about the simulation of the game. We will list the three main part of the user specification below including user visible specification, administrator visible specification and the technical specification; then we will start talk about the problem about the problem of the specification.

Ⅰ User visible part:

1, provide an interface for users to login in the game.

2, once the user has enrolled in the game, the system should provide a interface for user to stand and receive the information of the whole game, whatever the user is in predetermined step or in real play step.

3, the system should provide the correct status of the user: predetermined status or real play status.

4, in both the predetermined status and the real play status, the user should have an interface to see these parts:

A, the players have been selected in this user’s team.

B, they both should include other teams’ information icnluding teams’ players and teams’ ranking.

C, they both should include other teams’ player’s information including stats and points.

D, they both should show the available players for the user.

E, the losses and wines for each teams and the whole matchup schedule for the league.

F, for both the predetermined status and the real play status, the system should provide the function that users will not lose their information and chance when they are not enrolled in the network; for example, if a user is choosing some players in the predetermined status (to choose players in the real play status is different), then another user should wait for his turns and he will not lose the players have been chose when he get down from the internet.

5, in the predetermined status, the system should provide an interface for user to select a bunch of players within a random order.

6, in the predetermined status, the new user should wait until all the other old users have finished their session and start the new session.

6, in the predetermined status, when a player has been selected, other users cannot select it again.

7, in the predetermined status, the system should provide each player a default value of their stats and points.

8, during the selection, the user should wait until his turn, the system should tell the user what is the other user’s choice and what is the status of available players.

9, once the game enters the real play step, the system should provide a matchup automatically and the matchup schedule should follow some algorithm which is round robin.

10, user can change one data of the players which is roster positions through the whole session.

11, the system should provide the user to replace players with available players in the available list. If many teams are interesting in same players, the system will give the player to the user according to the first-come-first-serve principle.

12, the system should provide an interface to the user in order to let the user exchange players with other teams within some condition.

13, the system should provide an interface for user to finish their decision for one round of the real play game, and the interface should detect all user have finished the decision for this round then start to get information from the administrator of the game and calculate the result of this week automatically.

Ⅱ Administrator visible part:

1, the system should provide a way for administrator to login in the system and provide a specific interface for administrator in order to provide enough power for administrator to manage the whole system.

2, the system should provide an interface for administrator to insert or delete or manage the data of users, players and teams.

3, the system should provide an interface for administrator to manage the data from players after each round of the user determination; then, calculate the users' ranking automatically based on the current data and some formula.

4, the system should not allow the administrator to change anything except the system have provided, even if the formula to calculate the users' ranking and the framework of the databases. The system should be determined of any aspects except the coder want to show the administrators and users and hide details to any users including administrators.

Technical specification for application:

1, the system should provide visible and graphic interfaces for users or administrators instead of text editor windows.

2, the system should provide a good color for the background of the web-site in order to let the users feel comfortable.

3, the web-site should not surpass a certain size per page in order for user to play the game fast.

4, the web-site should include appropriate links in order for user easy to play the game.

5, the system should provide the web-site a appropriate environment and let the web-site run normally.

Database Specification:

According to the application specification of the whole project, there are several requirements for the database of the project.

1, we should have a user object to store the information about users. It will include these attributes: user name, user's team name (since each user will create his own and unique team), user password, week points, total points for the user, wins data, loss data and the rank data. The 'week points' attribute will have a relation with the weekly stats object and the players' object, and the 'week points' is calculated by a specific formula from weekly stats based on the players' position. The object user should also have these kinds of integrity constraints:

A, 'week points' are created by adding the 'week points' attribute of all of the players on their team.

B, 'total points' are kept track of by adding the 'week points' attribute to the 'total points' attribute.

C, 'wins' and 'losses' are created by taking the points from 'matchup' and the team with the higher, for example, 'week points (1,2)' value will add 1 to wins and the lower will add 1 to losses.

D, 'rank' is determined by having the team with the most wins as 1 and then descend the rank according to how many wins each team has from highest to lowest.

E, when the user wishes to view the rankings, the teams will be sorting according to ‘rank’ in descending order.

2, we should also have a player’s object to store the data of players. It will include these attributes: players name, NFL teams (the real team, not the team in our game), position, total points, week points, injury, owner (the team in our game) and availability. For the position attribute, the user can select the types from QB, RB, WR, TE, DEF, and K. For the injury status attribute, the user will select the types from P, O, and Q. The availability attribute means if the player has been selected by a team in the game, it is a yes/no attribute. Similarly, the week points attribute comes from weekly stats object have has a certain relation with it. There are also four integrity constraints:

A, when user wishes to view available players, only display player with ‘availability’ attribute = ‘Y’

B, when user wishes to view players of a certain position, only display players with their ‘position’ attribute = that which the user selected

C, both of these features can be combined (i.e. a user can view all of the available receivers)

D, user can sort these players by their points

3, we should have a schedule’s object to store the data of the weekly schedule to decide which two teams will match in each week. It includes the attributes from week one to week 2*(n-1) (n is the total number of the teams in the real play status in our game). This object is the object we cannot decide his real attribute at the beginning, and all the attributes must be created once all the users have been decided to enroll in the real play status of the game. Also, for each week, the attribute is a multi-value attribute, and it will includes the all the matchup schedule of this week. It will have a relation with users’ object called matchup relation, which can list the matchup information by each teams instead of by each weeks. There are also three integrity constraints for the schedule object and the matchup relation:

A, created using a round robin format of all the participating teams

B, when user wishes to view their or other teams’ schedules, it will display that row

C, created by using the ‘schedule’ and taking info from ‘participants’, when the user wishes to view their or other teams schedules, it will display that row according to the teams instead of weekly schedule.

4, we should also have two objects to represent the information about players’ weekly stats and total stats. They both have the same attributes: player’s name, pass TD, pass yards, interceptions, rush TD, rush yards, fumbles, receiving TD, receiving yards, points allowed, turnovers, sacks, defensive TD, field goal<40, field goal>40, missed field goal<40, missed field goal>40, PAT, missed PAT. Except player’s name, all the attributes here are the information to calculate the weekly points of the players’ and the users’. There are also two integrity constraints for these two objects:

A, the stats for the week are kept track of here, and are added to the ‘total stats’.

B, once the week is over, the info is erased from the weekly stats object.

C, points are tabulated according to a formula that converts these stats to points, which are then placed into ‘players’.

5, we should have an object called team roster to represent the selection of the user when the game is in the predetermined status. The main function of the table is to store the information of each user’s selection and prevent the other user to select the same players who have been selected. Also, it will be the whole team information of each user’s through the real play status of the game, and calculate the ranking of each user based on this object. The table will include these attributes: team name, QB, RB1, RB2, WR1, WR2, WR3, TE, DEF, K, BN1, BN2, BN3, BN4 and BN5. Except the team name attribute, all the other attribute is the position of the a team. For each user, it should have his own team roster object. There are another three main integrity constraints for this object:

A, the players in the starting spots (QB – K) will be the only ones used for calculated 'week points' and 'total points'

B, the user can switch players between the starting spots and bench (BN) spots anytime before Sunday

C, when a user wishes to view the roster of their own or another team, the application will call that row

Additionally, in the user object, each position on the team is represented by a 1-1 relationship between users and players. In the player object, players take all the individual stats from each week of playing as attributes, as well as one attribute representing their overall score for each week. Matchups between teams are represented by relationships between players, with attributes representing the week of the game and the winning team.

Players will be added to teams at the beginning of the season, and relationships between players and users may be modified during the season if users trade players. Team matchups are also calculated at the beginning of the season. Throughout the season, the database is expected to handle queries looking up the weekly stats of individual players and the win/loss data and ranks of teams. Once per week, players' stats are entered into the database, and team scores are calculated from this as well as wins and losses. Rankings are also recalculated at this time.

As we say before, both the user and the administrator cannot access the database directly, and the system should provide an interface for the user and the administrator to run the database. For some database, the user can just run once in the beginning of the game, just like the team roster object (user can just change the position after predetermined status). For most the database, the user cannot access them, except the user object. The administrator can access the entire database with valid authentication.

In the player object, when the user wishes to sort the players by points, we will use heap sort algorithm to sort the array, and give the feedback to the user, it will take O (nlogn) time. In the user object, when the user wishes to sort the user's rank, we will use card sort algorithm, which means to put the correct to the correct position of the array table, to sort the array, since the ranking is between 1 and n (n is the totally number of the users enrolled in the real play status of the game) and distant. The algorithm will just take O (n) time to finish the job.

REVISION HISTORY

REVISION	DESCRIPTION
A	Original Document

	USER SPECIFICATION
	Document: 10-JLJ-02-02A
	Title: Technical Requirement
	Team: JLJ TEAM
	Effective Date: Feb-12-2010

Purpose: This document provides the technical requirements for our project.

JLJ Team Technical Requirement:

ID	Requirement
TR01	The system should provide visible and graphic interfaces for users or administrators instead of text editor windows.
TR02	The system should provide a good color for the background of the web-site in order to let the users feel comfortable.
TR03	The web-site should not surpass a certain size per page in order for user to play the game fast.
TR04	The web-site should include appropriate links in order for user easy to play the game.
TR05	The system should provide the web-site an appropriate environment and let the web-site run normally.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

	CONCEPT DESIGN
	Document: 10-JLJ-03-01A
	Title: Programming Language Concept
	Team: JLJ TEAM
	Effective Date: March-21-2010

Purpose: this document provides the determination of our programming language.

JLJ Team Programming Language Concept:

To decide which programming language we should use for the project, the choices are between Java and PHP. In order to decide which one is appropriate, our teams decide to use rating matrix for us to solve the problem. Basically, there are four main parts to be considered as factors of choosing programming language. One is easy for us to use this language; the second is easy for us to study, since we may not know anything about any language, we will expected ourselves to study some parts of the language. The third one is easy to connect with the database system, since we will choose the MYSQL to be our DBMS, therefore, the compatibility between the language and MYSQL is an important part. Finally, whatever we choose, it should be easy for the user to use.

For the four aspects, we give each one a weight in our rating matrix. Since this is a project in EECS 341 courses, and it is a simulation from a real world game, the main factor for us to consider is to write it appropriately and easily instead of being respected to the users. Therefore, we decide to give the former three parts 0.3 weights for each, and give a 0.1 weight for the last one.

For the first part, many people will say that PHP is easy to use; however, this part means to choose a language easy for us to use instead of a language easy to use. Due to the fact that we just have one person in our group can write PHP code and we are all computer science major in our group, the PHP language is not easy for us to use though it may really simpler than Java. For the second part, to study a little further from a known language is definitely easier than to study a fully new language. For the third part, by doing some research in the internet, we know that both these two languages are connected with MYSQL well, however, it is true that more web-site designer would like to choose PHP plus MYSQL instead of Java plus MYSQL. For the fourth part, the PHP language is very easy for user definitely since the java program will ask the user to install the java machine usually. The total of Java is higher than PHP; therefore, we will choose Java plus Html as our program language.

The result is listed below:

Ratings Matrix (Track/Motorization)

	weight	JAVA+HTML	PHP
Easy to use	0.3	8	7
Easy to study	0.3	9	6
Connect database	0.3	7	8
Easy for user	0.1	6	9
total	1	7.8	7.2

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

CONCEPT DESIGN
Document: 10-JLJ-03-02A
Title: Database Concept Design
Team: JLJ TEAM
Effective Date: March-28-2010

Purpose: this document provides the determination of our concept of Database design.

JLJ Team Database Concept Design:

Basically, we need to have these kinds of tables or relations in our database: user, players, schedule, teamroster and stats in order to let our program keep going.

There are several choices for the design of the database. First, two opposite design for the stats part: we can do either a whole table to contain the weeklystats and totalstats, or we can have a separated tables to contain weeklystats and totalstats. Combination one can run faster, while separation one can save space and make it easy to use when we need to call either just weeklystats or totalstats.

Second, for the user part, since we have a user name and the team name of the user, therefore, we have two unique attributes where user name and team name can unique determine all other attributes in the table. For these two attributes, we can design make it a separate two attributes table and delete teamname column in the original user table; on the other hand, we can let the two attributes in the same table and keep only one table for user table. The first idea makes the relation more clear; while the second one make the table run faster.

Third, for each column in the teamroster, it is from the name of the players table. There exist two solutions for the teamroster table: either make all columns foreign key from table players, or make just teamname foreign key from user table. The first solution makes the relation between two tables more clear, but we need to deal with null value in the first solution. For the second solution, we can set the null value to be N in order to avoid the null value.

To choose the right selection, we use rating matrix. We set four values for the rating matrix which is complexity, stability, speed and space. Complexity means if the solution is easy to use, we set it 0.35, stability means if the solution can get the correct value without risk, we set it 0.35, speed means if the solution runs fast and space means that if the solution need large or small space, both of this two are set to be 0.15. from the rating matrix below, we decide to choose the solution having two tables for the first problem, the solution to have user table combination for the second problem, the solution to have teamroster table with foreign key in most of the table for the third problem.

Ratings Matrix

	weight	Stats together	Weekly/total stats	User combination	User separation	Teamroster foreign key	Teamroster without foreign key
Complexity	0.35	7	9	8	8	7	8
Stability	0.35	8	8	7	8	7	9
Speed	0.15	7	7	8	6	8	8
Space	0.15	6	9	8	6	8	8
total	1	7.2	8..35	7.65	7.4	7.3	8.35

REVISION HISTORY

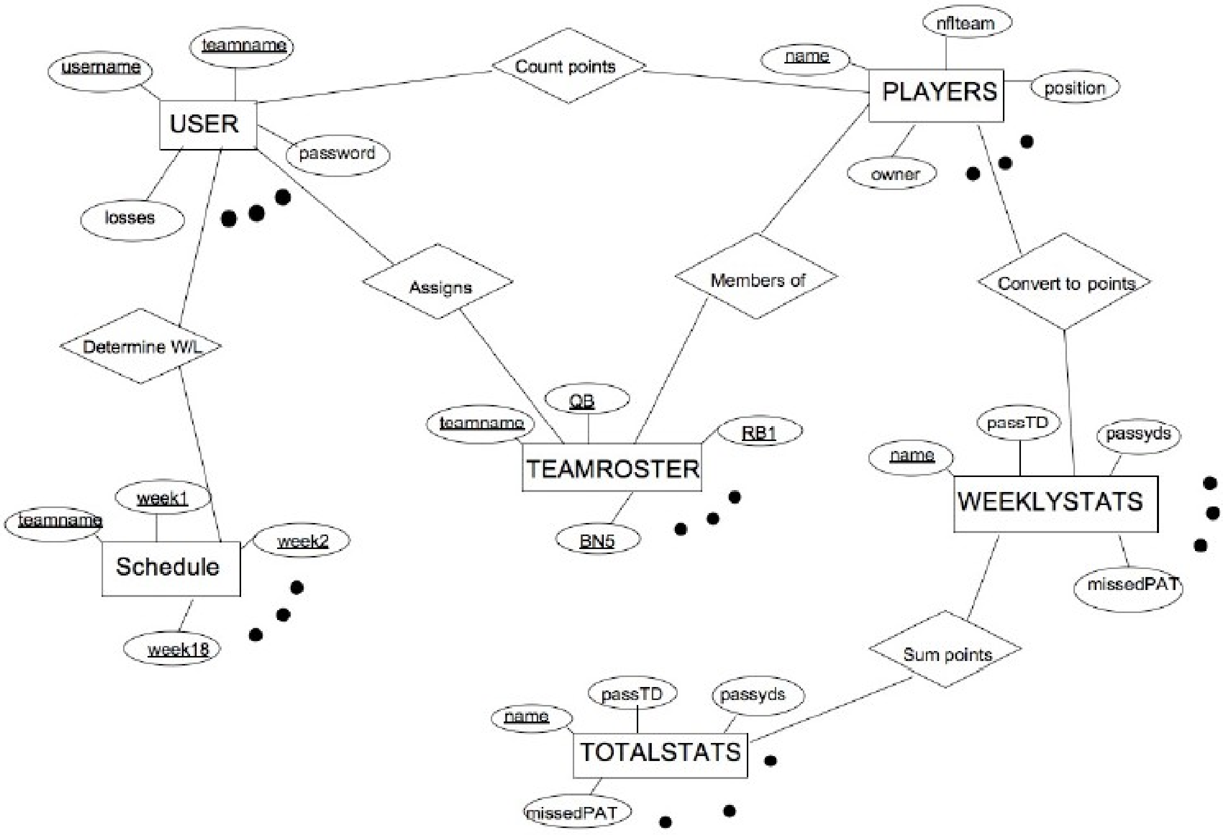
REVISION	DESCRIPTION
A	Original Document

	CONCEPT DESIGN
	Document: 10-JLJ-03-02B
	Title: Final Concept For Database
	Team: JLJ TEAM
	Effective Date: March-28-2010

Purpose: this document provides the description of the final choice of our database design

JLJ Team Database Final Concept for Database:

The ER diagram list below is our final design for our Database:



Constraint:
1, the count points relation between USER and PLAYERS table is 1 to M relation
The count points relation can be put into players table; once we put it into players table, it will become BCNF, since there are not any functional dependency between players and this table, also, originally, we have already have a owner column which is from the username; however, we decide to let this two table separately, since we usually call players table and rarely call count points table, it is waste resource to let them together.
2, the assigns between USER and TEAMROSTER table is 1 to 1 relation
Since, the relation it 1 to 1 relation, we just need to add the username as the primary key in teamroster table. It is in BCNF automatically.

3,the Determine WL relation between USER and Schedule table is 1 to 1 relation
Since, the relation it 1 to 1 relation, we just need to add the username as the primary key in schedule table. It is in BCNF automatically.
4,the Members of relation between PLAYERS and TEAMROSTER table is M to 1 relation
This table is special. All players have a position which is the column name of the teamroster, all column of a single teamname is a player name. we just need to add a player position column in players for the relation.
5,the Convert to points relation between PLAYERS and WEEKLYSTATS table is 1 to 1 relation
Since, the relation it 1 to 1 relation, we just need to add the player name as the primary key in WEEKLYSTATS table. It is in BCNF automatically.
6,the Sum points relation between WEEKLYSTATS and TOTALSTATS table is 1 to 1 relation
Since, the relation it 1 to 1 relation, it is hard to deal with the relation between this two, may be a procedure is a good idea, however, we decide to let the relation deal by program.

The relation model list below:

The first table is to represent the user information

Dependency: since username is primary key and teamname is unique we have username and teamname depends anything. Also rank is candidate key, since if rank is unique determine a user, since windata and lossdata can determine rank, therefore, windata plus lossdata is a candidate key.

Revisiting: The F set is {U→anything,T→U,R→U,WL→R}, the table is in BCNF, we do not need to change it.

```
DROP TABLE IF EXISTS `jlj`.`user`;
CREATE TABLE  `jlj`.`user` (
  `username` char(20) NOT NULL DEFAULT '',
  `teamname` char(30) DEFAULT NULL,
  `password` char(20) DEFAULT NULL,
  `totalpoints` double DEFAULT NULL,
  `weekpoints` double DEFAULT NULL,
  `rank` int(11) DEFAULT NULL,
  `windata` int(11) DEFAULT NULL,
  `lossdata` int(11) DEFAULT NULL,
  `modes` int(11) DEFAULT NULL,
  PRIMARY KEY (`username`),
  UNIQUE KEY `rank` (`rank`),
  UNIQUE KEY `Teamname` (`teamname`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The second table is to represent the relation between user table and players table, which is “rewards points to” relation , it also need the position information to calculate the weekpoints

Dependency: name can unique determine all theres, there are no more dependency

Revisiting: min F set is {playername→anything}. Original we have teamname in the table (from report two), however, we discover the playername can unique determine all others except the teamname, therefore, if we have teamname, the table will be in 1NF, now after we delete the teamname, it is in BCNF

```
DROP TABLE IF EXISTS `jlj`.`countpoints`;
CREATE TABLE  `jlj`.`countpoints` (
  `playername` char(30) NOT NULL DEFAULT '',
  `playerposition` char(3) DEFAULT NULL,
  `sumweekpoints` double DEFAULT NULL,
  `sumtotalpoints` double DEFAULT NULL,
  PRIMARY KEY (`playername`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The third table is to represent the players information:
PLAYERS (name: string, nflteam: string, position: string, totalpoints: real, weekpoints: real, availability: string, injurystate: string, owner: string)

Dependency: name can unique determine anything else, other than name, there are not other denpendency

Revisiting: min set of F{name→anything}, the table is in BCNF, we do not need to change anything.

```
DROP TABLE IF EXISTS `jlj`.`players`;
CREATE TABLE  `jlj`.`players` (
  `name` char(30) NOT NULL DEFAULT '',
  `nflteam` char(30) DEFAULT NULL,
  `position` char(3) DEFAULT NULL,
  `totalpoints` double DEFAULT NULL,
  `weekpoints` double DEFAULT NULL,
  `availability` int(10) unsigned DEFAULT NULL,
  `injurystate` char(1) DEFAULT NULL,
  `owner` char(30) DEFAULT NULL,
  PRIMARY KEY (`name`),
  KEY `FK_players_1` (`owner`),
  CONSTRAINT `FK_players_1` FOREIGN KEY (`owner`) REFERENCES `user` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The 4th table is to represent the weeklystats information and includes the relation between this table and players table which name is “cover to pots for”.

Dependency: name determines anything else. Originally, we think that all other attributes determine calpoints; however, once we see the function of how to calculate calpoints, we find that it is impossible to calculate calpoints without the player’s position information, therefore, other than name, there are not other dependency

Revisiting: min set of F{name→anything}, the table is in BCNF, we do not need to change anything.

WEEKLYSTATS(name: string, passTD: real, passyards: real, interceptions: real, rushTD: real, rushyards: real, fumbles: real, receivingTD: real, receivingyards: real, pointsallowed: real, turnovers: real, sacks: real, defensiveTD: real, fieldgoal<40: real, fieldgoal>40: real, missedfieldgoal<40: real, missedfieldgoal>40: real, PAT: real, missedPAT: real, calpoints: real)

```
DROP TABLE IF EXISTS `jlj`.`weeklystats`;
CREATE TABLE  `jlj`.`weeklystats` (
  `name` char(30) NOT NULL DEFAULT '',
  `passTD` double DEFAULT NULL,
  `passyards` double DEFAULT NULL,
  `interceptions` double DEFAULT NULL,
  `rushTD` double DEFAULT NULL,
  `rushyards` double DEFAULT NULL,
  `fumbles` double DEFAULT NULL,
  `receivingTD` double DEFAULT NULL,
  `receivingyards` double DEFAULT NULL,
  `pointsallowed` double DEFAULT NULL,
  `turnovers` double DEFAULT NULL,
  `sacks` double DEFAULT NULL,
  `defensiveTD` double DEFAULT NULL,
  `fieldgoalless40` double DEFAULT NULL,
  `fieldgoalgreater40` double DEFAULT NULL,
  `missedfieldgoaless40` double DEFAULT NULL,
  `missedfieldgoalgreater40` double DEFAULT NULL,
  `PAT` double DEFAULT NULL,
  `missedPAT` double DEFAULT NULL,
  `calpoints` double DEFAULT NULL,
  PRIMARY KEY (`name`),
  CONSTRAINT `weeklystats_ibfk_1` FOREIGN KEY (`name`) REFERENCES `players` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The 5th table is just the same as 4th, and the function is to store the sum value of each weekly stats.

Dependency: the same as No.4 table

Revisiting: the same as No.4 table

TOTALSTATS(name: string, passTD: real, passyards: real, interceptions: real, rushTD: real, rushyards: real, fumbles: real, receivingTD: real, receivingyards: real, pointsallowed: real, turnovers: real, sacks: real, defensiveTD: real, fieldgoal<40: real, fieldgoal>40: real, missedfieldgoal<40: real, missedfieldgoal>40: real, PAT: real, missedPAT: real, totalpoints: real)

```
DROP TABLE IF EXISTS `jlj`.`totalstats`;
CREATE TABLE  `jlj`.`totalstats` (
  `name` char(30) NOT NULL DEFAULT '',
  `passTD` double DEFAULT NULL,
  `passyards` double DEFAULT NULL,
  `interceptions` double DEFAULT NULL,
  `rushTD` double DEFAULT NULL,
  `rushyards` double DEFAULT NULL,
  `fumbles` double DEFAULT NULL,
  `receivingTD` double DEFAULT NULL,
  `receivingyards` double DEFAULT NULL,
  `pointsallowed` double DEFAULT NULL,
  `turnovers` double DEFAULT NULL,
  `sacks` double DEFAULT NULL,
  `defensiveTD` double DEFAULT NULL,
  `fieldgoalless40` double DEFAULT NULL,
  `fieldgoalgreater40` double DEFAULT NULL,
  `missedfieldgoaless40` double DEFAULT NULL,
  `missedfieldgoalgreater40` double DEFAULT NULL,
  `PAT` double DEFAULT NULL,
  `missedPAT` double DEFAULT NULL,
  `calpoints` double DEFAULT NULL,
  PRIMARY KEY (`name`),
  CONSTRAINT `totalstats_ibfk_1` FOREIGN KEY (`name`) REFERENCES `players` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The 6th table is the TEAMROSTER table, it will allow the user to assign all the player to specific position in his team, in this table, one thing must be mentioned is that if a player want to assign a player to a different position, the system will change the data in players' position column automatically, since different positions have different calculated points method. Actually, the table can use a check constraint, just like:CHECK ('QB'=(SELECT P.position
FROM PLAYERS P
WHERE P.name=TEAMROSTER.QB))
However, for 14 column in the TEAMROSTER table, it must check the constraints for 14 times, and the changing position work is done by our Java program instead of real user, we will assume that our program will not have program and avoid to use the constraints in order not to waste the resources.

Dependency: teamname can unique determine all other attributes, there are no more functional dependency

Revising: min F set is {teamname→anything}, the table is in BCNF, do not need to change anything

TEAMROSTER (teamname: string, QB: string, RB1: string, RB2: string, WR1: string, WR2: string, WR3: string, TE: string, DEF: string, K: string, BN1: string, BN2: string, BN3: string, BN4: string, BN5: string)

```
DROP TABLE IF EXISTS `jlj`.`teamroster`;
CREATE TABLE  `jlj`.`teamroster` (
  `teamname` char(30) NOT NULL DEFAULT '',
  `QB` char(30) DEFAULT NULL,
  `RB1` char(30) DEFAULT NULL,
  `RB2` char(30) DEFAULT NULL,
  `WR1` char(30) DEFAULT NULL,
  `WR2` char(30) DEFAULT NULL,
  `WR3` char(30) DEFAULT NULL,
  `TE` char(30) DEFAULT NULL,
  `DEF` char(30) DEFAULT NULL,
  `K` char(30) DEFAULT NULL,
  `BN1` char(30) DEFAULT NULL,
  `BN2` char(30) DEFAULT NULL,
  `BN3` char(30) DEFAULT NULL,
  `BN4` char(30) DEFAULT NULL,
  `BN5` char(30) DEFAULT NULL,
  PRIMARY KEY (`teamname`),
  KEY `QB` (`QB`),
  KEY `RB1` (`RB1`),
  KEY `RB2` (`RB2`),
  KEY `WR1` (`WR1`),
  KEY `WR2` (`WR2`),
  KEY `WR3` (`WR3`),
  KEY `TE` (`TE`),
  KEY `DEF` (`DEF`),
  KEY `K` (`K`),
  KEY `BN1` (`BN1`),
  KEY `BN2` (`BN2`),
  KEY `BN3` (`BN3`),
  KEY `BN4` (`BN4`),
  KEY `BN5` (`BN5`),
  CONSTRAINT `FK_teamroster_1` FOREIGN KEY (`teamname`) REFERENCES `user` (`teamname`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The 7th table is the SCHEDULE table, it is really a dynamic table after the game enter real play status then the SCHEDULE table is created and the total columns in the table depends on the total users enrolled in the real play status of the game. Since, each users should fight with the other users for twice in a session, therefore, the total weeks is 2*(n-1) (n is the total user number), when we do the Java program, we will find a way to decide the total column. All the content of the week from one to 2*(n-1) are from the teamname of USER table, and signed to the teamname in this table with round robin method.

Dependency: since it is dynamic table, it is hard to say exactly how many columns in the table, however, other than username column, all other columns is about the opposite team in each week, there exist only one functional dependency which is teamname determine anything else.

Revising: min F set is {teamname→anything}, the table is in BCNF, do not need to change anything

SCHEDULE (teamname: string, week1: string, week2: string, …..week2*(n-1): string)

```
DROP TABLE IF EXISTS `jlj`.`schedule`;
CREATE TABLE  `jlj`.`schedule` (
  `username` char(20) NOT NULL,
  `week1` char(20) DEFAULT NULL,
  `week2` char(20) DEFAULT NULL,
  PRIMARY KEY (`username`),
  CONSTRAINT `schedule_ibfk_1` FOREIGN KEY (`username`) REFERENCES `user` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Revising No.8 table: We decide to delete the original 8th table in report two which is sumwin table. Since we have already have the user table in BCNF, we do not need a extra table to store the win and loss data, it is redundancy. Therefore, we decide to delete the table.
SUMWIN (teamname: string, week1: string, week2: string, …..week2*(n-1): string, wins, integer, loss, integer)

SQL QUERY:

Choose all the players which is unavailable.

SQL: SELECT * from players p where p.availability=0
RA: $\sigma_{availability=0}$(Players)
TRC: $\{t \mid \exists p(t=p \wedge Players(p) \wedge p[6]=0)\}$

Choose the name and injury of the players where their name is some specific by temp

SQL: SELECT p.name,p.injurystate from player p where p.name='"+temp+"'”
RA: $\pi_{name,injurystate}(\ \sigma_{\ name=temp}(Players))$
TRC:$\{t^2 | \exists p(Players(p) \wedge p[1]=temp \wedge p[1]=t[1] \wedge p[7]=t[2])\}$

List all the totalstats where the players of the totalstats is belong to specific user

SQL: SELECT * FROM totalstats t where t.name IN (select p.name from players p where p.owner='"+username+"')
RA: $totalstats \bowtie \pi_{name}(\ \sigma_{\ owner=username}(players))$
TRC:$\{t | \exists s \exists p(totalstats(s) \wedge players(p) \wedge s[1]=p[1] \wedge t=s \wedge s[8]=username)\}$

Choose the team which belong to specific user

SQL: SELECT u.teamname FROM user u where u.username='"+username+"'
RA: $\pi_{teamname}(\ \sigma_{\ username='\ username'}\ (user))$
TRC:$\{t^1 | \exists u(user(u) \wedge t[1]=u[2] \wedge u[1]=username)\}$

Choose the user where the username and password is specific

SQL: SELECT * FROM user u where u.username='"+username+"' AND u.password='"+password+"'
RA: $\sigma_{\ username='\ username'\ \wedge password='\ password'}\ (user)$
TRC$\{t | \exists u(user(u) \wedge u=t \wedge u[1]=username \wedge u[3]=password)\}$

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document
B	Final Concept

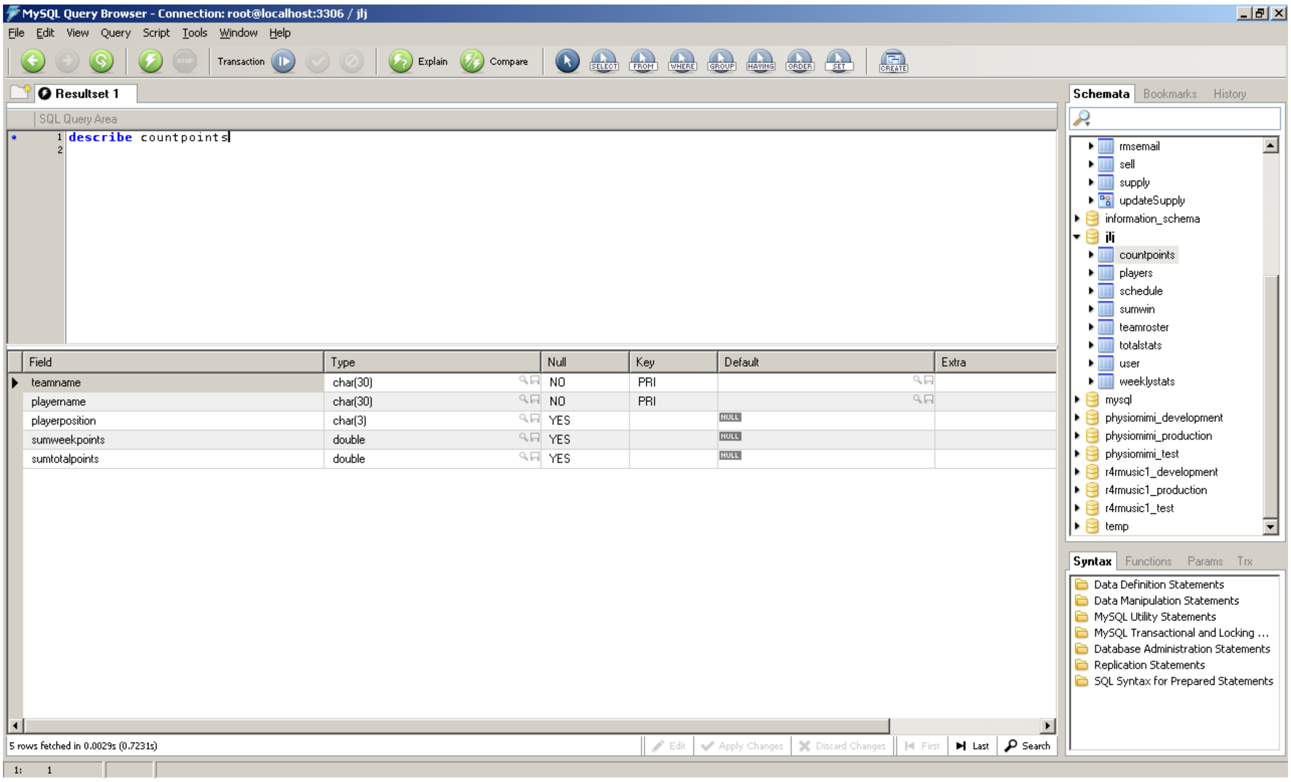
	CONCEPT DESIGN
	Document: 10-JLJ-03-02C
	Title: Database Implementation
	Team: JLJ TEAM
	Effective Date: March-28-2010

Purpose: this document provides the description of the final choice of our database design

JLJ Team Database Implementation:

First table countpoints:

```
DROP TABLE IF EXISTS `jlj`.`countpoints`;
CREATE TABLE  `jlj`.`countpoints` (
  `teamname` char(30) NOT NULL DEFAULT '',
  `playername` char(30) NOT NULL DEFAULT '',
  `playerposition` char(3) DEFAULT NULL,
  `sumweekpoints` double DEFAULT NULL,
  `sumtotalpoints` double DEFAULT NULL,
  PRIMARY KEY (`teamname`,`playername`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



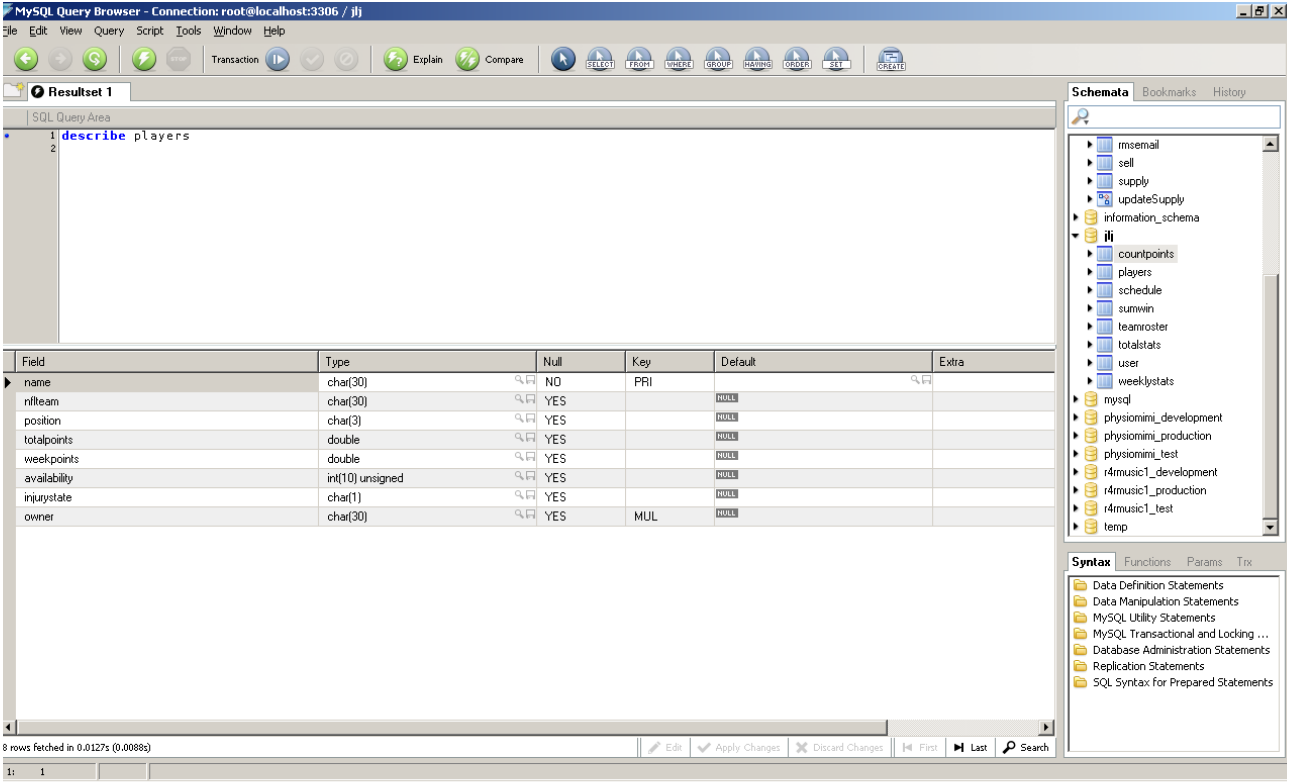
```
describe countpoints
```

Field	Type	Null	Key	Default	Extra
teamname	char(30)	NO	PRI		
playername	char(30)	NO	PRI		
playerposition	char(3)	YES		NULL	
sumweekpoints	double	YES		NULL	
sumtotalpoints	double	YES		NULL	

5 rows fetched in 0.0029s (0.7231s)

Second table players

```
DROP TABLE IF EXISTS `jlj`.`players`;
CREATE TABLE  `jlj`.`players` (
  `name` char(30) NOT NULL DEFAULT '',
  `nflteam` char(30) DEFAULT NULL,
  `position` char(3) DEFAULT NULL,
  `totalpoints` double DEFAULT NULL,
  `weekpoints` double DEFAULT NULL,
  `availability` int(10) unsigned DEFAULT NULL,
  `injurystate` char(1) DEFAULT NULL,
  `owner` char(30) DEFAULT NULL,
  PRIMARY KEY (`name`),
  KEY `FK_players_1` (`owner`),
  CONSTRAINT `FK_players_1` FOREIGN KEY (`owner`) REFERENCES `user` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```


MySQL Query Browser - Connection: root@localhost:3306 / jlj
File Edit View Query Script Tools Window Help
Transaction
Explain
Compare
SELECT FROM WHERE GROUP HAVING ORDER SET CREATE
Resultset 1
SQL Query Area
Schemata Bookmarks History
rmsemail
sell
supply
updateSupply
information_schema
jlj
countpoints
players
schedule
sumwin
teamroster
totalstats
user
weeklystats
mysql
physiomimi_development
physiomimi_production
physiomimi_test
r4music1_development
r4music1_production
r4music1_test
temp
Syntax Functions Params Trx
Data Definition Statements
Data Manipulation Statements
MySQL Utility Statements
MySQL Transactional and Locking ...
Database Administration Statements
Replication Statements
SQL Syntax for Prepared Statements
Edit Apply Changes Discard Changes First Last Search
1: 1

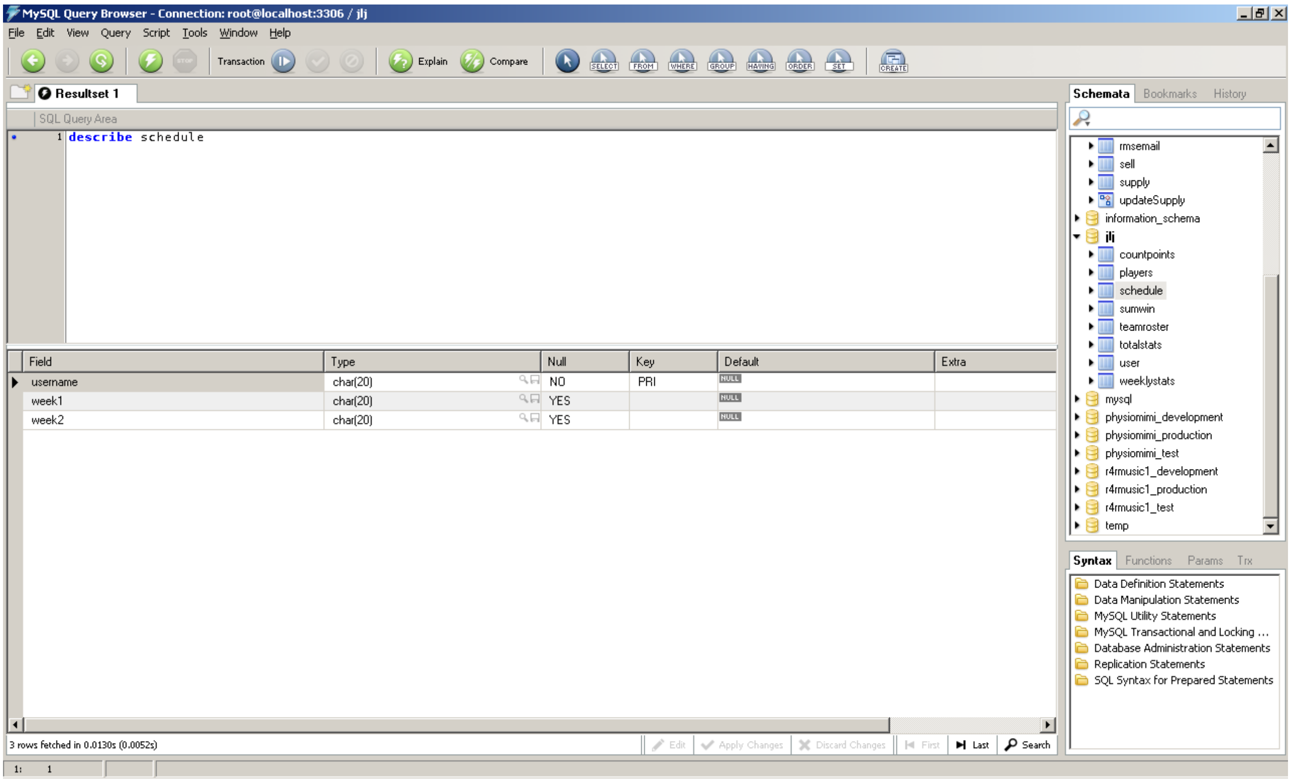
```
describe players
```

Field	Type	Null	Key	Default	Extra
name	char(30)	NO	PRI		
nflteam	char(30)	YES		NULL	
position	char(3)	YES		NULL	
totalpoints	double	YES		NULL	
weekpoints	double	YES		NULL	
availability	int(10) unsigned	YES		NULL	
injurystate	char(1)	YES		NULL	
owner	char(30)	YES	MUL	NULL	

8 rows fetched in 0.0127s (0.0088s)

Third table schedule

```
DROP TABLE IF EXISTS `jlj`.`schedule`;
CREATE TABLE  `jlj`.`schedule` (
  `username` char(20) NOT NULL,
  `week1` char(20) DEFAULT NULL,
  `week2` char(20) DEFAULT NULL,
  PRIMARY KEY (`username`),
  CONSTRAINT `schedule_ibfk_1` FOREIGN KEY (`username`) REFERENCES `user` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



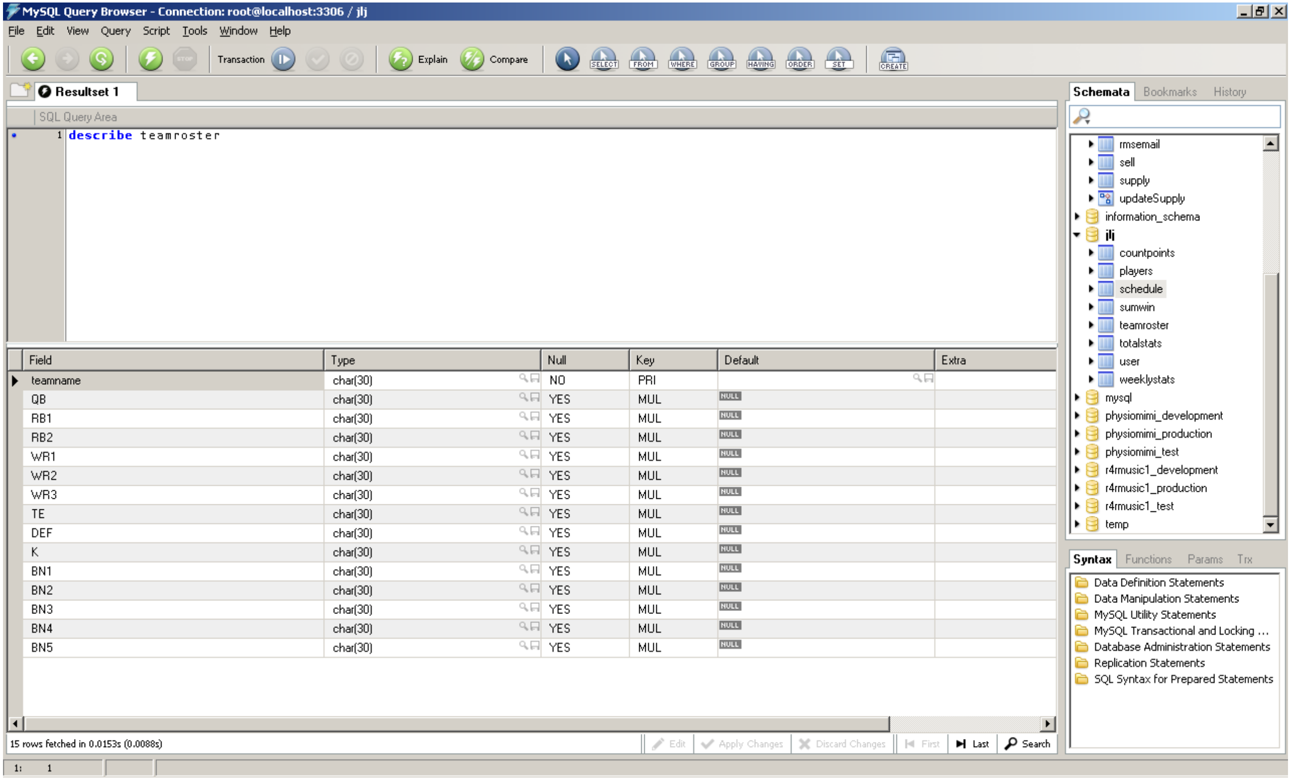
```
describe schedule
```

Field	Type	Null	Key	Default	Extra
username	char(20)	NO	PRI	NULL	
week1	char(20)	YES		NULL	
week2	char(20)	YES		NULL	

3 rows fetched in 0.0130s (0.0052s)

Fourth table teamroster

```
 DROP TABLE IF EXISTS `jlj`.`teamroster`;
CREATE TABLE  `jlj`.`teamroster` (
  `teamname` char(30) NOT NULL DEFAULT '',
  `QB` char(30) DEFAULT NULL,
  `RB1` char(30) DEFAULT NULL,
  `RB2` char(30) DEFAULT NULL,
  `WR1` char(30) DEFAULT NULL,
  `WR2` char(30) DEFAULT NULL,
  `WR3` char(30) DEFAULT NULL,
  `TE` char(30) DEFAULT NULL,
  `DEF` char(30) DEFAULT NULL,
  `K` char(30) DEFAULT NULL,
  `BN1` char(30) DEFAULT NULL,
  `BN2` char(30) DEFAULT NULL,
  `BN3` char(30) DEFAULT NULL,
  `BN4` char(30) DEFAULT NULL,
  `BN5` char(30) DEFAULT NULL,
  PRIMARY KEY (`teamname`),
  KEY `QB` (`QB`),
  KEY `RB1` (`RB1`),
  KEY `RB2` (`RB2`),
  KEY `WR1` (`WR1`),
  KEY `WR2` (`WR2`),
  KEY `WR3` (`WR3`),
  KEY `TE` (`TE`),
  KEY `DEF` (`DEF`),
  KEY `K` (`K`),
  KEY `BN1` (`BN1`),
  KEY `BN2` (`BN2`),
  KEY `BN3` (`BN3`),
  KEY `BN4` (`BN4`),
  KEY `BN5` (`BN5`),
  CONSTRAINT `FK_teamroster_1` FOREIGN KEY (`teamname`) REFERENCES `user` (`teamname`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



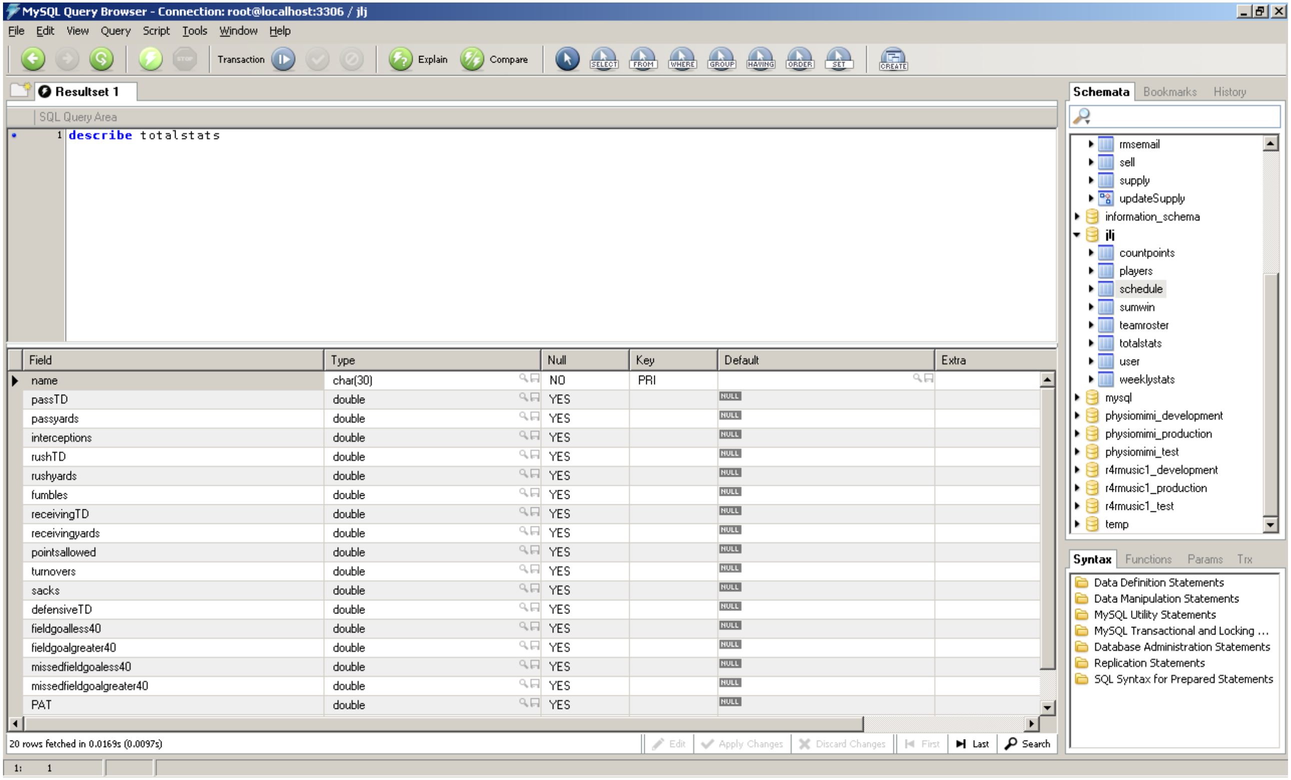
```
describe teamroster
```

Field	Type	Null	Key	Default	Extra
teamname	char(30)	NO	PRI		
QB	char(30)	YES	MUL	NULL	
RB1	char(30)	YES	MUL	NULL	
RB2	char(30)	YES	MUL	NULL	
WR1	char(30)	YES	MUL	NULL	
WR2	char(30)	YES	MUL	NULL	
WR3	char(30)	YES	MUL	NULL	
TE	char(30)	YES	MUL	NULL	
DEF	char(30)	YES	MUL	NULL	
K	char(30)	YES	MUL	NULL	
BN1	char(30)	YES	MUL	NULL	
BN2	char(30)	YES	MUL	NULL	
BN3	char(30)	YES	MUL	NULL	
BN4	char(30)	YES	MUL	NULL	
BN5	char(30)	YES	MUL	NULL	

15 rows fetched in 0.0153s (0.0088s)

Fifth table totalstats

```
DROP TABLE IF EXISTS `jlj`.`totalstats`;
CREATE TABLE  `jlj`.`totalstats` (
  `name` char(30) NOT NULL DEFAULT '',
  `passTD` double DEFAULT NULL,
  `passyards` double DEFAULT NULL,
  `interceptions` double DEFAULT NULL,
  `rushTD` double DEFAULT NULL,
  `rushyards` double DEFAULT NULL,
  `fumbles` double DEFAULT NULL,
  `receivingTD` double DEFAULT NULL,
  `receivingyards` double DEFAULT NULL,
  `pointsallowed` double DEFAULT NULL,
  `turnovers` double DEFAULT NULL,
  `sacks` double DEFAULT NULL,
  `defensiveTD` double DEFAULT NULL,
  `fieldgoalless40` double DEFAULT NULL,
  `fieldgoalgreater40` double DEFAULT NULL,
  `missedfieldgoaless40` double DEFAULT NULL,
  `missedfieldgoalgreater40` double DEFAULT NULL,
  `PAT` double DEFAULT NULL,
  `missedPAT` double DEFAULT NULL,
  `calpoints` double DEFAULT NULL,
  PRIMARY KEY (`name`),
  CONSTRAINT `totalstats_ibfk_1` FOREIGN KEY (`name`) REFERENCES `players` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



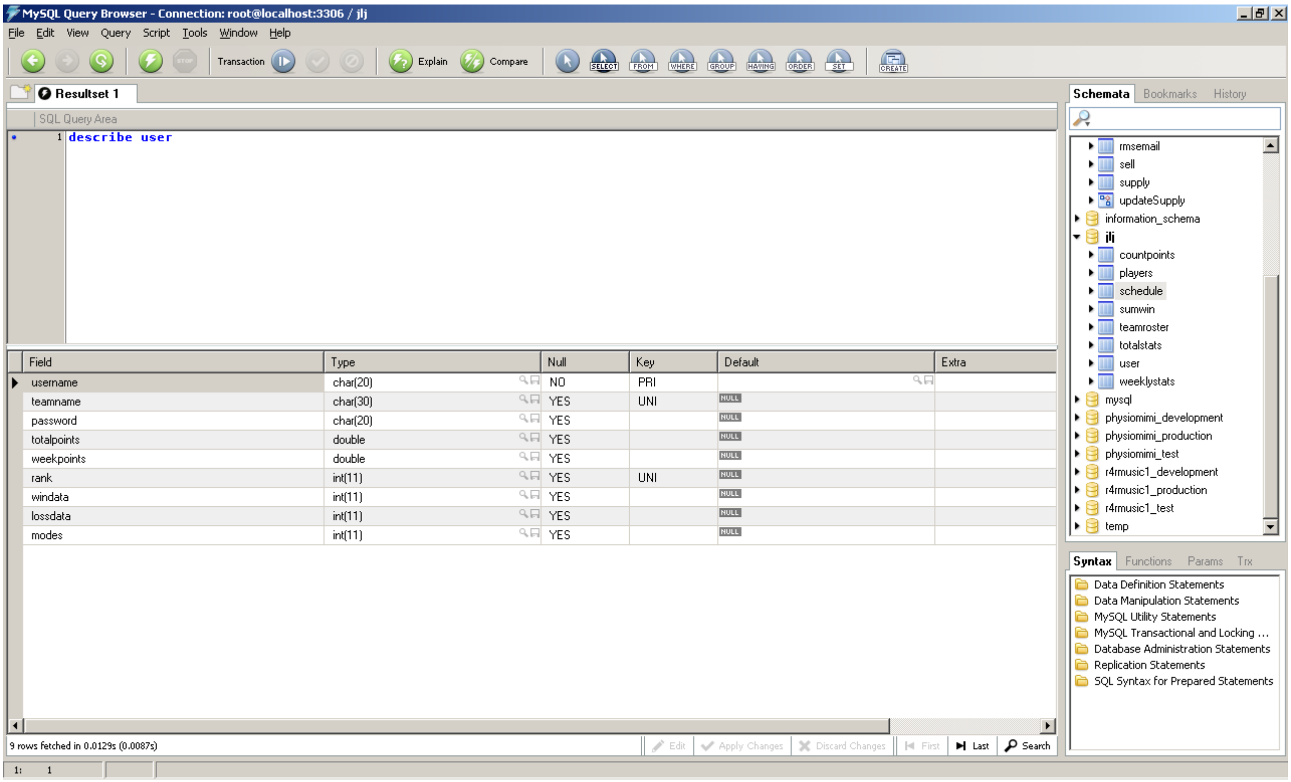
```
describe totalstats
```

Field	Type	Null	Key	Default	Extra
name	char(30)	NO	PRI		
passTD	double	YES		NULL	
passyards	double	YES		NULL	
interceptions	double	YES		NULL	
rushTD	double	YES		NULL	
rushyards	double	YES		NULL	
fumbles	double	YES		NULL	
receivingTD	double	YES		NULL	
receivingyards	double	YES		NULL	
pointsallowed	double	YES		NULL	
turnovers	double	YES		NULL	
sacks	double	YES		NULL	
defensiveTD	double	YES		NULL	
fieldgoalless40	double	YES		NULL	
fieldgoalgreater40	double	YES		NULL	
missedfieldgoaless40	double	YES		NULL	
missedfieldgoalgreater40	double	YES		NULL	
PAT	double	YES		NULL	

20 rows fetched in 0.0169s (0.0097s)

Sixth table user

```
DROP TABLE IF EXISTS `jlj`.`user`;
CREATE TABLE  `jlj`.`user` (
  `username` char(20) NOT NULL DEFAULT '',
  `teamname` char(30) DEFAULT NULL,
  `password` char(20) DEFAULT NULL,
  `totalpoints` double DEFAULT NULL,
  `weekpoints` double DEFAULT NULL,
  `rank` int(11) DEFAULT NULL,
  `windata` int(11) DEFAULT NULL,
  `lossdata` int(11) DEFAULT NULL,
  `modes` int(11) DEFAULT NULL,
  PRIMARY KEY (`username`),
  UNIQUE KEY `rank` (`rank`),
  UNIQUE KEY `Teamname` (`teamname`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



MySQL Query Browser - Connection: root@localhost:3306 / jlj

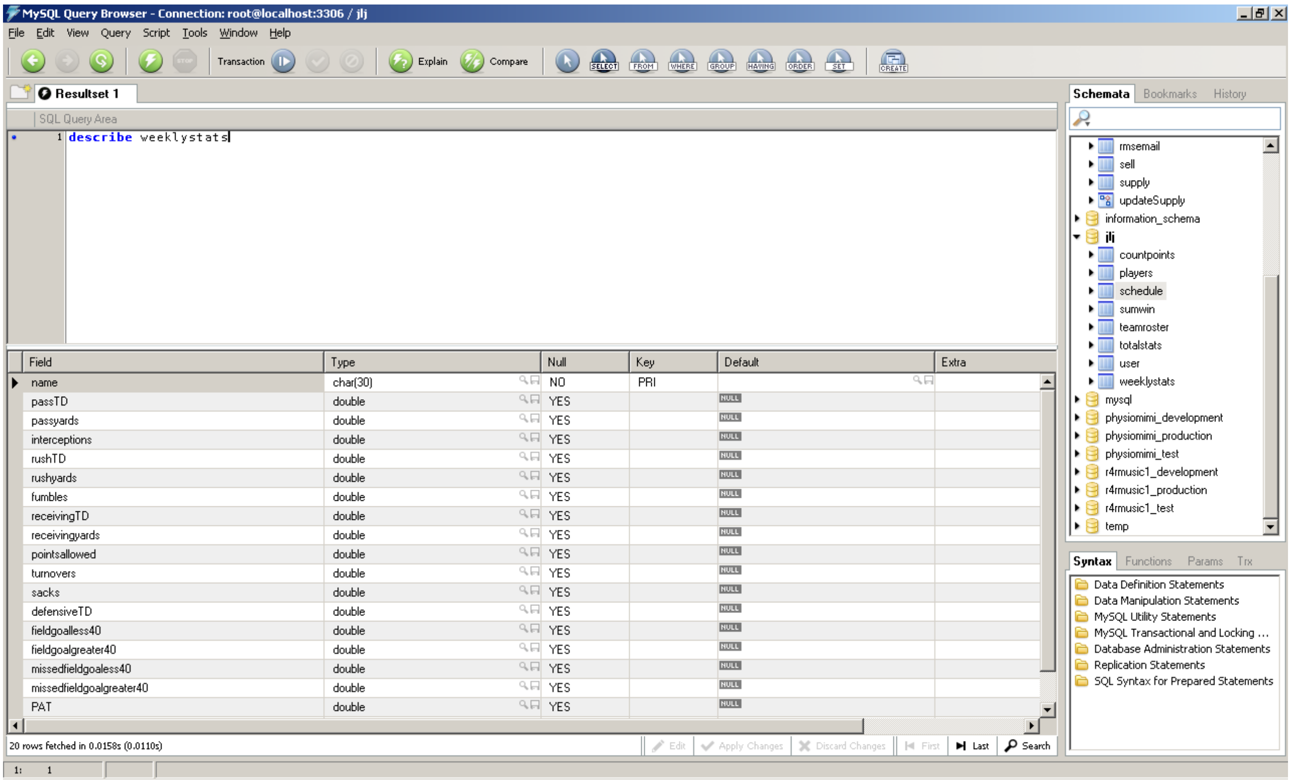
```
describe user
```

Field	Type	Null	Key	Default	Extra
username	char(20)	NO	PRI		
teamname	char(30)	YES	UNI	NULL	
password	char(20)	YES		NULL	
totalpoints	double	YES		NULL	
weekpoints	double	YES		NULL	
rank	int(11)	YES	UNI	NULL	
windata	int(11)	YES		NULL	
lossdata	int(11)	YES		NULL	
modes	int(11)	YES		NULL	

9 rows fetched in 0.0129s (0.0087s)

Seventh table weeklystats

```
DROP TABLE IF EXISTS `jlj`.`weeklystats`;
CREATE TABLE  `jlj`.`weeklystats` (
  `name` char(30) NOT NULL DEFAULT '',
  `passTD` double DEFAULT NULL,
  `passyards` double DEFAULT NULL,
  `interceptions` double DEFAULT NULL,
  `rushTD` double DEFAULT NULL,
  `rushyards` double DEFAULT NULL,
  `fumbles` double DEFAULT NULL,
  `receivingTD` double DEFAULT NULL,
  `receivingyards` double DEFAULT NULL,
  `pointsallowed` double DEFAULT NULL,
  `turnovers` double DEFAULT NULL,
  `sacks` double DEFAULT NULL,
  `defensiveTD` double DEFAULT NULL,
  `fieldgoalless40` double DEFAULT NULL,
  `fieldgoalgreater40` double DEFAULT NULL,
  `missedfieldgoaless40` double DEFAULT NULL,
  `missedfieldgoalgreater40` double DEFAULT NULL,
  `PAT` double DEFAULT NULL,
  `missedPAT` double DEFAULT NULL,
  `calpoints` double DEFAULT NULL,
  PRIMARY KEY (`name`),
  CONSTRAINT `weeklystats_ibfk_1` FOREIGN KEY (`name`) REFERENCES `players` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
describe weeklystats
```

Field	Type	Null	Key	Default	Extra
name	char(30)	NO	PRI		
passTD	double	YES		NULL	
passyards	double	YES		NULL	
interceptions	double	YES		NULL	
rushTD	double	YES		NULL	
rushyards	double	YES		NULL	
fumbles	double	YES		NULL	
receivingTD	double	YES		NULL	
receivingyards	double	YES		NULL	
pointsallowed	double	YES		NULL	
turnovers	double	YES		NULL	
sacks	double	YES		NULL	
defensiveTD	double	YES		NULL	
fieldgoalless40	double	YES		NULL	
fieldgoalgreater40	double	YES		NULL	
missedfieldgoaless40	double	YES		NULL	
missedfieldgoalgreater40	double	YES		NULL	
PAT	double	YES		NULL	

20 rows fetched in 0.0158s (0.0110s)

REVISION HISTORY

REVISION	DESCRIPTION
A	Original Document
B	Final Concept
C	Database Implementation

	CONCEPT DESIGN
	Document: 10-JLJ-03-03A
	Title: Application Concept Design
	Team: JLJ TEAM
	Effective Date: March-26-2010

Purpose: this document provides the determination of our concept of Application design.

JLJ Team Application Concept Design:

From Programming Concept Design, we will use Java plus Html to finish our program design. Basically, our design will divide the program into four parts: the login part, the selection roster part, the matchup roster part and the admin part.

We also have some arguments in the design of application. First, we will have either one league or multiple leagues for the whole game, for only one league, it is easy to apply; while it is more similar to the original game if we have multiple leagues. Second, we design the assigning value to weeklystats by hand-writing or by randomly assigning from computer. If we do it by hand-writing, it is more similar to the original game; while it is easier for admin to control if we do it by randomly assigning. Third, we argued if we should use an extra admin table or column to show who the admin is, or we just use the key word “admin” to show who the admin is. The former one make the database clearer, the later one reduce the complexity.

We also use the rating matrix to decide our choice for our design. The rating matrix includes these parts: complexity for coding, complexity for user, stability, database connection, similarity to original game. We give complexity for coding 0.3 score in the rating matrix, since our simulation must be easy to code. We give complexity for user 0.2 score in the rating matrix, since it is as important as complexity for coding, but it is also important for simulation of our program. One thing that must be mention is that this property includes complexity for real user and admin. We give stability and database connection 0.20 score in the rating matrix, and we give similarity to original game 0.1 score.

The final result lists below. Our selection is single league, do the weeklystats by random assigning and show admin in key word.

Ratings Matrix

	weight	Single league	Multiple league	Weeklystats by hand-writing	Weeklystats by computer	Admin in extra column	Admin in key word
Complexity for coding	0.3	9	6	9	8	6	9
Complexity for user	0.2	9	8	6	9	7	7
stability	0.2	7	7	7	7	8	7
database connection	0.2	9	9	7	8	9	8
similarity to original game	0.1	5	9	9	6	9	8
total	1	8.2	7.5	7.6	7.8	7.5	7.9

REVISION HISTORY

REVISION	DESCRIPTION
A	Original Document

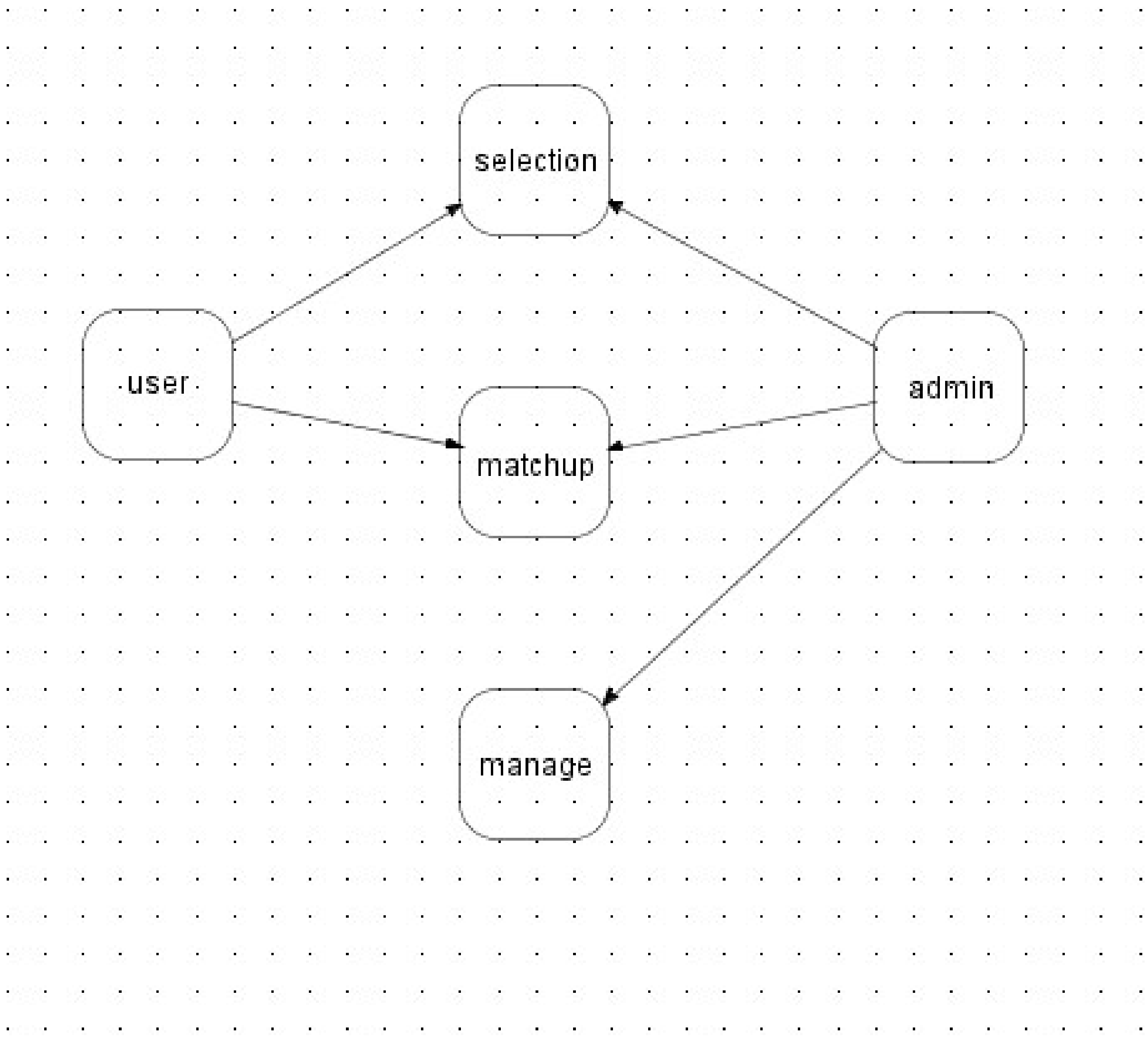
	CONCEPT DESIGN
	Document: 10-JLJ-03-03B
	Title: Application Final Concept
	Team: JLJ TEAM
	Effective Date: March-26-2010

Purpose: this document provides the final concept of our application.

JLJ Team Application Final Concept:

1. Final Concept diagram:

The diagram shows the basic relation of our application.



2. Final Concept in each page

A: introduction page

user has the ability to type their account name and password into boxes, and then

click/press enter to login.
-a proper name and password takes the user to page B: league list page the user also has the option to click on ‘register’ to register a new name and pasword
-clicking ‘register’ takes the user to page D: registration page

B: league list page

This page displays a list of all of the league that that person is a member of, with the ability to select any from that list
-clicking on a league name will take the user to that leagues page C: league page the user can also click on ‘sign up’ to create or join another league
-clicking ‘sign up’ takes the user to page F: league select page

C: league page

The league page has a task bar at the top with the following selection options:
‘my team’ K
‘my matchup’ L
‘available players’ M
‘draft’ N
‘logout’ O
This page will display all of the team names in the league with their respective win/loss record, and rank them in order
-clicking on any of the team names will take that user to that team’s page P: other roster page next, this page will display the weekly matchups for that week
-clicking on any of the matchups will take the user to those teams’ page Q: other matchup page

D: registration page

The user has the option to type an account name and password, and then click/press enter to check the availability of that account name
-if the account name is not taken, then clicking/pressing enter will take the user to page E: registry confirmation page
-if the name is taken, then the user will be prompted to type another name and password

E: registry confirmation page

Tells the user that the registration is complete and that their name and password are confirmed.
-clicking continue take the user to their page B: league list page

F: league select page

user can select either ‘create’ or ‘join’
-slecting ‘create’ takes the user to page G: create league page
-selecting ‘join’ takes user to page I: join league page

G: create league page

user can type in a new league name
-if the league name is not taken, then that league is created and the user is taken to page H: creation confirm page
-if the name is take, the user will be prompted to type another name

H: creation confirm page

Tells the user that their new league has been created and confirmed

-clicking continue takes user to that league's page C: league page

I: join league page

user can type in an existing league name

-if the league name matches an already existing leage name, the user is taken to page J: join confirm page

-if the name doesn't exist, the user will be told that that league doesn't exist, and allows user to enter the name again

J: join confirm page

Tells the user that they are confirmed as members of the leage name they typed in

-clicking continue takes the user to that league's page C: league page

K: my team page

The my team page has a task bar at the top with the following selection options:

'my team' K
'my matchup' L
'available players' M
'draft' N
'logout' O

the team name is displayed at the top

the user's starting lineup and bench are displayed along with their:

position
player name
total stats
total points

L: my matchup page

the my matchup page has a task bar at the top with the following selection options:

'my team' K
'my matchup' L
'available players' M
'draft' N
'logout' O

one column has the user's team name on top, with their starting lineup displayed beneath with the following attributes shown:

position
player name
week points
total weekly points

second column has the opponent's team name on top, with their starting lineup displayed beneath with the following attributes shown:

position
player name
week points
total weekly points

for the sake of showing functionality off-season, there will also be a 'next week button'

-'next week' button will basically simulate the league for that week perhaps we should have this on page C: league page instead so that it simulates everything for the week

M: available players page

the available players page has a task bar at the top with the following selection options:

'my team' K
'my matchup' L
'available players' M
'draft' N
'logout' O

this page will display 4 columns with the following attributes for available players:

player name
position
total stats
total points

a 5th column will be included either before the player name or after the total points that will consist of buttons labeled 'select' or 'acquire'

-selecting the button for a certain player will move that player to the user's roster, and remove that player from the available players list

N: draft page

the draft page has a task bar at the top with the following selection options:

'my team' K
'my matchup' L
'available players' M
'draft' N
'logout' O

this page will display 3 columns with the following attributes for available players:

player name
position
nfl team

a fourth column withh be included either before player name or after nfl team that will consist of buttons labeled 'select'

-selecting the button for a certain player will move that player to the user's roster, and remove the player from the available players list a fifth column at the end of the page will display the draft order of the teams in the league, with an arrow or some other indication of who's turn it is to select.

-while it's one user's turn, only they can click on 'select' buttons. the other users cannot do so once every team has their roster filled, the draft page will display 'draft finished' rather than the player info and draft order, and will remain this way for the remainder of the season (perhaps if we have time, we could instead display the players chosen in the order they were chosen and which user selected them)

O: logout page

Selecting logout from any of the taskbars will take the user to a page that displayed 'you have successfully logged out'
-clicking on 'continue' will take the user to page A: introduction page

P: other rosters page

The other rosters page has a task bar at the top with the following selection options:
'my team' K
'my matchup' L
'available players' M
'draft' N
'logout' O
the team name is displayed at the top
the user's starting lineup and bench are displayed along with their:
position
player name
total stats
total points
this page will also have a button labeled 'matchup' at the bottom of the page
-clicking on 'matchup' will take the user to page Q: other matchup page

Q: other matchup page

the other matchup page has a task bar at the top with the following selection options:
'my team' K
'my matchup' L
'available players' M
'draft' N
'logout' O
one column has one member's team name on top (the team who's matchup the user is viewing), with their starting lineup displayed beneath with the following attributes shown:
position
player name
week points
total weekly points
second column has the opponent's team name on top, with their starting lineup displayed beneath with the following attributes shown:
position
player name
week points
total weekly points

	CONCEPT DESIGN
	Document: 10-JLJ-03-03B
	Title: Application Final Concept
	Team: JLJ TEAM
	Effective Date: March-26-2010

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document
B	Final Application Concept

	VERIFICATION
	Document: 10-JLJ-04-01A
	Title: Verification Plan
	Team: JLJ TEAM
	Effective Date: April-10-2010

Purpose: To show the outline of the verification plan.

JLJ Team Verification Plan:

The verification test will be performed on April 20th, 2010 to April 27th, 2010.

- The JLJ team will test the program to see whether it fits the user requirement and technical requirement.

- The test will include the following
  - Test the login system and see if it will allow wrong user to login.
  - Test the Selection system of the user and see if it will run correct, specifically, we will test and see if it will give us the correct feedback after we select a player.
  - Test the matchup system and wee if it will allow the user do his application correctly, includes select his own teamroster, select his new team player and refuse he put the unavailable players to his team.
  - Test the matchup move on system, and see if all user click the start button, the system will move to next week.
  - Test the admin system, and see if the administrator can do his job correctly.

- Five trials of the test will be performed and evaluated, the acceptance criteria will be performed on the verification protocol.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

VERIFICATION
Document: 10-JLJ-04-02A
Title: Verification Protocol
Team: JLJ TEAM
Effective Date: April-10-2010

Purpose: This document provides a protocol for verifying that the design fits the technical requirements for the design.

JLJ Team Verification Protocol:

Verification ID	Tech. Req. Tested	Test Protocol	Success Criteria
VR01	TR01, TR02, TR03 TR04,TR05	Test the login system and see if it will allow wrong user to login.	Right user success login and wrong user cannot, also admin user will login into a special page.
VR02	TR01, TR02, TR03 TR04, TR05	Test the Selection system of the user and see if it will run correct, specifically, we will test and see if it will give us the correct feedback after we select a player.	The user can select available players, and cannot select unavailable players; once the players have been selected, the system will show the player is in specific team and let he become unavailable
VR03	TR01, TR02, TR03 TR04, TR05	Test the matchup system and wee if it will allow the user do his application correctly, includes select his own teamroster, select his new team player and refuse he put the unavailable players to his team.	The user can fix his teamroster, but cannot put injury player to his starting lineup, the links between each page are correct, the color of each page look comfortable.
VR04	TR04, TR05	Test the matchup move on system, and see if all user click the start button, the system will move to next week.	Once all the users click the start button, the admin should know it and move the system to go to next week.
VR05	TR01, TR03, TR04, TR05	Test the admin system, and see if the administrator can do his job correctly.	The admin should include all function the admin can do and the links must be correct.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

	VERIFICATION
	Document: 10-JLJ-04-03A
	Title: Verification Data
	Team: JLJ TEAM
	Effective Date: April-26-2010

Purpose: This document provides the data from the verification tests as well as data processing required by the protocols.

JLJ Team Verification Data:

Verification ID		Date	Person(s)
		04/26/2010	All
VR01			
Trial #	Success or Failure		
1	Failure	Color in the page looks not so good	
2	Success		
3	Success		
4	Success		

Verification ID		Date	Person(s)
		04/26/2010	All
VR02			
Trial #	Success or Failure		
1	Failure	Pages surpass the bottom line	
2	Success		
3	Success		
4	Success		

Verification ID		Date	Person(s)
		04/26/2010	All
VR03			
Trial #	Success or Failure		
1	Success		
2	Success		
3	Success		
4	Success		

	VERIFICATION
	Document: 10-JLJ-04-03A
	Title: Verification Data
	Team: JLJ TEAM
	Effective Date: April-26-2010

Verification ID		Date	Person(s)
		04/26/2010	All
VR04			
Trial #	Success or Failure		
1	Failure	Random assign code have some problem	
2	Failure	Forget to put random injury status assign	
3	Success		
4	Success		

Verification ID		Date	Person(s)
		04/26/2010	All
VR05			
Trial #	Success or Failure		
1	Success		
2	Success		
3	Success		
4	Success		

Revision History

Revision	Description
A	Original Document

	VALIDATION
	Document: 10-JLJ-05-01A
	Title: Validation Plan
	Team: JLJ TEAM
	Effective Date: April-26-2010

Purpose: To show the outline of the validation plan.

JLJ Team Validation Plan:

The validation test will be performed on April 30th, 2010.

- The designers will perform a presentation first.
- The user will test the device to see whether it fits the specifications.
- The test will include the following:
  - Test the login system and see if it will allow wrong user to login.
  - Test the Selection system of the user and see if it will run correct, specifically, we will test and see if it will give us the correct feedback after we select a player.
  - Test the matchup system and wee if it will allow the user do his application correctly, includes select his own teamroster, select his new team player and refuse he put the unavailable players to his team.
  - Test the matchup move on system, and see if all user click the start button, the system will move to next week.
  - Test the admin system, and see if the administrator can do his job correctly.
- Three trials of the test will be performed and evaluated

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

USER MANUAL
Document: 10-JLJ-06-01A
Title: JLJ Team User Manual
Team: JLJ TEAM
Effective Date: April-27-2010

Purpose: this document provides a User Manual.

JLJ TeamUser Manual:

Getting Started:

To begin your fantasy football experience, go to website you have set up during the installation process. This is the starting page for the application.

Introduction Page:

Start by clicking on “Register” to begin the registration process. Here, you may enter an account name and password of your choosing granted that the name is not already taken. Click on the “Enter” button to continue to the *League Page*.

League Page:

On the *League Page*, all of the teams in the league will be displayed. Up to 8 teams may join a single league.

Clicking on the “Logout” button will log you out of the application.

Logging in:

After you have gone through the registration process and have joined or created a league, you can get back into the application at any time by once again visiting the site.

Introduction Page:

Enter your account name and password into the proper fields and click on “Enter” to go to the *League Page*.

The Draft:

The draft is a very important part of the fantasy football process. The player draft allows each team to select NFL players to populate their roster. It’s these players that will earn them the points used for competition. The members of the league must first agree on a date to execute the player draft.

League Page:

On the date agreed upon, you need to click on the "Draft" button on the League Page. This takes you to the *Draft Page*.

Draft Page:

The Draft Page will display all of the available players (players that are not already chosen to be on another team). The draft allows the teams to select players one by one in a randomized order. The ultimate goal is to obtain the best players possible to fill your roster; your roster will contain a certain number of players from each position. You will get to choose 1 quarterback (QB), 2 running backs (RB), 3 wide-receivers (WR), 1 tight-end (TE), 1 defense (DEF), 1 kicker (K), and 5 bench players (BN) which can be from any of the previously mentioned positions. These bench players will be used as replacements in case a player on your roster is not earning enough points or is injured.

When it becomes a team's turn to select they must choose one of the players from the list, and once their selection has been made, it is the next team's turn to select a player. When it is your turn to select a player, examine the list of players; if you see a player that you believe will earn you a lot of points or that you need to fill a roster position, click on the button next to their name and they will be removed from the available players list and be placed on your team. The draft will last 15 rounds, which will fill each team's roster and bench. When the draft ends, the season will begin.

The Fantasy Football Season:

After you have completed the draft, you will have a complete team of players that will earn you points every week of the NFL football season based on the stats they earn for that week.

Week by week, teams in the league will be pitted against each other in a head-to-head competition. The points earned by one team's roster will be compared to the points earned by the opposing team's roster. The team with the most points wins the matchup and gets a 'win'; the other team will get a 'loss'. Teams will be ranked according to the number of their wins and losses. At the end of the season, the team with the number 1 rank will be the league's winner.

League Page:

At the top of the page there's menu options including: My Team, My Matchup, Available Players, Draft, and Logout.

Selecting "My Team" will take you to the *My Team Page*.
Selecting "My Matchup" will take you to the *My Matchup Page*.
Selecting "Available Players" will take you to the *Available Players Page*.
Selecting "Draft" will take you to the *Draft Page* (however, since the draft is already over, this page would not be of any use other than seeing that the draft has been completed)
Selecting "Logout" will log you out of the application and bring you back to the *Introduction Page*.

On the league page, the teams will be listed according to their ranking. By clicking on a team name, you can view that team's *Team Page*. Below that, the matchups for the week will be displayed. Clicking on one of the matchups will bring you to that *Matchup Page*.

My Team Page:

This page will display your roster of players, along with their stats for the year and the points they have earned for those stats. On this page, you have the option to switch players from your roster with bench players, or you may drop players from your team entirely so that you can select a new player from the list of available players. These kinds of moves are typically done if a player on your roster is injured or underperforming to your expectations.

My Matchup Page:

The *My Matchup Page* will display the rosters of your team beside the roster of the team you will be facing off against that week. The points each player earns will be displayed next to them, and then all players' points will be totalled and displayed at the bottom of each roster.

Available Players Page:

On this page, you can view all players that are not currently on another team's roster. If you were to delete a player from your roster, you can go to this page and select a new player by clicking on the button next to that player's name. That player will then become a part of your roster.

Team Page:

This page acts the same as the *My Team Page* except that it shows the selected team's roster instead of yours.

Matchup Page:

This page acts the same as the *My Matchup Page* except that it shows the selected matchup rosters and points instead of the matchup you will be participating in.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

USER MANUAL

Document: 10-JLJ-06-02A

Title: JLJ Team Instruction Manual

Team: JLJ TEAM

Effective Date: April-27-2010

Purpose: this document provides a Instruction Manual.

JLJ Team Instruction Manual:

This application requires working installations of the MySQL dbms, the Java development environment, and the Apache web server with the Tomcat extension. Each of these products are available for every major operating system. The application itself comes as a zip folder. To install it, simply unzip this folder somewhere within the webaapps/ROOT directory of your tomcat installation. Then login to a mysql shell as root and execute the install.sql file. This will set up the database jlj as well as a user also named jlj with full permissions on this database but nowhere else. All mysql queries performed by the application use the permissions of this user.

Now navigate to the web page in any web browser and you will be taken to the login page. Register with username admin to access to the admin control page, which allows you to generate the weekly schedule and simulate each week of play, in addition to the privileges of any other user. Once the application is fully installed, it is recommended that you remove any file ending with a .sql or .jsp extension from the application directory.

REVISION HISTORY

REVISION	**DESCRIPTION**
A	Original Document

	USER MANUAL
	Document: 10-JLJ-06-02A
	Title: JLJ Team Program Manual
	Team: JLJ TEAM
	Effective Date: April-27-2010

Purpose: this document provides a Program Manual.

JLJ Team Program Manual:

The application itself is split into files primarily based on web page. index.jsp exists solely as a redirect page. The user will be directed to login.jsp, the login page. login.jsp first checks whether the user has a valid login cookie. If they do, it presents a redirect to the main page, user.jsp, or to admin.jsp if the user is logged in as the admin. Otherwise, it presents a simple login prompt and a link to the registration page. If the user enters a valid login, i.e., the username/password combination matches an entry in the user table, it generates a cookie and then redirects them.

user.jsp is the main user page. It contains the generic code shared by all user pages and uses include directives to display the portion of the page that changes as the user navigates. The files that can be included are league.jsp (the default, which gives information about every team in the league), team.jsp, which gives the current statistics of the user's team, matchup.jsp, which allows a comparison of the user's team to the team they played against on any given week, players.jsp, a listing of players that have not been selected by any team, draft.jsp, which allows the user to draft players for their team, and roster.jsp and other_match.jsp, which provide the same functionality as team.jsp and match.jsp except that the user can view other teams instead of their own.

The other page reachable from the main page is the log-out page (logout.jsp), which clears the user's session cookie and redirects them to the login page.

If the user is admin, they also have access to the admin page (admin.jsp), which contains functions to generate the schedule of games and simulate each week of play.

The remaining files (those with .sql and .csv extensions) are used in the initial installation of the application to create the database and populate the tables.

The application enforces security on multiple levels. First, the user and admin pages are only viewable by users who have logged in. If the user does not have a valid username and password, a redirect page is displayed instead. Because this is handled on the server side, this works regardless of whether users have scripting enabled. Secondly, any mysql call that has the potential to change the database in any way has been implemented as a stored procedure, thus ensuring that the only changes users can make are those that the application specifically intends them to.

REVISION HISTORY

REVISION	DESCRIPTION
A	Original Document