

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the text 'Grade 12 Key 3'. Below the banner, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

Grade 12
Key 3

PAT Project Specifications Document

Five a Side Football

Nikhar Ramlakhan

TRINITYHOUSE HIGH SCHOOL 2020

Table of Contents

<u>Content</u>	<u>Page</u>
Project Summary	2
Program Functions	2
Goals and Success Criteria	3
UI Design	4
Help Menu	5
Data Storage	6
System Requirements	8

Project Summary

Using Java as a programming language, I will create a 5 a side football game where players will select a team and play off against other opponents in a simulation against 5 other teams in a tournament.

The user will select players based on an algorithm that allows fairness in the team selection that will ensure that there is a balance in the team. This is done using a "level of quality" number for each position with 5 players in each category. It will range from Level 1 being the worst possible players too Level 5 being the best possible player. The program will automatically assign a level of quality number to each position ensuring that players cannot pick only the best players. The program will then generate 5 random teams, using the same algorithm with the level of quality check for the user team to play against.

The tournament will then consist of a user picked team and 5 program generated teams. The user's team will then face a program-generated team with the other 4 teams playing in the background. After each game, the user will play against the team that they haven't yet played against.

The team who deserves to win will be decided using an algorithm that requires each player to have 3 statistics, an overall rating, an offensive stat and a defensive stat. A database will be used to store 125 players with 25 players per position.

After each game simulation, the user will be shown the results of the game such as their points acquired, goals scored and the form of the team indicating how many games were won, lost or drawn.

Program Functions

1. The user must be able to start the game and go to the home screen where they must be able to play the game, or access the leaderboard or help menu.
2. If the user starts the game:
 - The user must be able to enter their username, team name and tournament name as per the validation rules.
 - The user must be able to select a player for all 5 positions.
 - The user must be able to refresh to get a new set of selectable options.

- The user must be able to change a player before confirming their team.
 - The user must be able to select tactics for each game.
 - The user must be able to click a button to play each game.
 - The user must be able to play all 5 games.
 - The program must simulate 2 games in the background of the teams that aren't playing the user and must ensure that each team has only played another team once.
 - The user must receive the game result after each game that they have played.
 - The user must be able to progress to the next screen after all 5 games have been played.
 - The program must determine the total points acquired by each team. Points are given based on the following outcomes:
 - Team wins game: +1000 points
 - Team draws game: +500 points
 - Team loses game: +200 points
 - Every goal scored: +100 points
 - The user must receive information regarding their overall team performance such as total goals scored and trends with their match outcomes.
 - The user must be able to see which team acquired the most points (hence the winner) and the rankings of all 6 teams.
 - The user must be able to save their results and go to the leaderboard to see if they placed in the top 10.
3. If the user accesses the leaderboard:
- The user must be able to view the top 10 user teams ranked in order of highest points acquired.
 - If a team is selected by the user, they must be able to see the players, stats and rankings of the team.
4. If the user triggers the help menu:
- The user must be able to select the category of help they require and the program must display the appropriate content.

Goals and Success Criteria

- α Movement between menus and frames is fluid.
- α The theme and graphics of the program are clean and simplistic.
- α There are no spelling errors.
- α The database is normalized and well organized.

- α The database connects and communicates with the program successfully and the program is able to access and store data in the database.
- α The various help menus are helpful and understandable.
- α Validation checks work correctly and do not cause any program errors.
- α The player selection works flawlessly adhering to the player level quality fairness selection algorithm.
- α The refresh button works and randomly generates players again.
- α Each player has a face ID photo to make the game more graphically pleasing.
- α The program is able to store the database and images in such a way that there is no need to always transfer data across when running the program on another computer.
- α The buttons work correctly and take the user to the correct screen.
- α The simulated team is randomly generated under the same fairness algorithm (using the level of quality approach) to ensure each team is different.
- α The program can simulate the other games and perform the necessary calculations in the background
- α The handling for player vs player is done correctly ensuring that the game is fair when determining the result of each game and goals scored by each team.
- α The leaderboard updates when accessed ensuring that the rankings are accurate and based on the most recent game played.
- α Progress in development is consistent and deadlines are met.
- α No “crashes” occur during program testing.

UI Design

1. Start Screen and the Home Screen

- There is an initial small screen with a button to start the game and go to the home screen.

- There will be 3 options to select on the home screen in the form of buttons:
 - Play Game
 - Leaderboard
 - Help Menu

2. Play Game

- The Input Page will consist of 3 text fields that will capture the:
 - Name of user
 - Name of tournament
 - Name of team
- Select Team Screen:
 - 5 combo boxes will be used for the user to select each player.
 - Players' faces and their offensive and defensive statistics will be displayed below each combo box respectively.
 - Refresh and cancel buttons will be on the bottom of the screen.
 - A confirm button will be used to proceed to the next screen.
- Tournament Matches
 - When accessed, a pop-up message dialogue will appear prompting the user of how the points system works.
 - The screen is split up into two parts with the user's team on the left and the generated opposition on the right-hand side. Each teams display will include the player's face, name, position and stats.
 - A play match button is located in the centre at the bottom.
 - A help button is located below the play match button.
 - A message dialog box will appear after each game has been simulated to indicate if the user's team won, lost or drew.
 - Labels are displayed in the middle of the screen and are used to show the current rankings of all the participating teams and results of the user's team.
- Tournament Results
 - The user's team members face, name and position will be displayed.

- Labels are used to show the rankings of all the teams (in a ranking table) from the tournament sorted by highest points acquired.
- The team stats such as goals scored, games won, games lost and games drew are visible using labels.
- There is a save button that also goes to the leaderboard.

3. Leaderboard

- 10 buttons will be used to rank the top 10 teams on the left side of the screen.
- Buttons are used so that the user can click the team and view the relevant details of the team and tournament on the right side of the screen in the form of tables and text panes.
- There is a back button to go back to the home screen at the top left.

4. Help Menu

- There will be 3 buttons at the top of the screen for each type of help category:
 - Basics
 - Team Selection
 - Points and Ranks
- After a button is selected, an image is displayed below the buttons showing the relevant content for each help category.

Help Menu

The help menu is triggered using a button that is visible in the home screen. When it is clicked, the help menu screen appears prompting the user to select a help category.

Basics will go over:

- What Five a Side football is.
- How to start the game.
- The rules for name and team and tournament name entries.

Teams Selection will go over:

- How player selections are determined and set by the program.
- What statistics are and how they are used.
- The acronyms for each position and their strongest stat.

Points and Ranks will go over:

- How points work and how they are assigned.
- How leaderboard rankings work.

Data Storage

- Images of Players and Backgrounds used will be stored in the **Netbeans Project Folder** to be able to run the game on another computer and show all necessary graphics.
- There will be one main **Microsoft Access Database** that will consist of multiple tables:

1) Players – Fields:

- Player ID
- Position
- First Name
- Surname
- LOQ (*Level of Quality*)
- LOQN (*Level of Quality Number*)
- Overall (*Overall Rating of the Player*)
- Off Stat (*Offensive Stat*)
- Def Stat (*Defensive Stat*)
- Image Location

When are records created?

Records are created by the programmer.

When are records accessed?

When the user is required to select players for the team.

When the program needs to generate opposition teams.

When the leaderboard is used.

A class in the program will access all the records and put them into an array.

When are records updated?

Records are updated by programmer when necessary.

2) Teams – Fields:

- Tournament ID
- Username
- Team Name
- Tournament Name
- GK ID (*Goalkeeper ID*)
- CB ID (*Centre back ID*)
- CDM ID (*Central Defensive Midfielder ID*)
- CAM ID (*Central Attacking Midfielder ID*)
- ST ID (*Striker ID*)

When are records created?

When a user enters their details and confirms their team.

When a user selects their players and confirms their team.

When are records accessed?

When the user accesses a top 10 team from the leaderboard.

When are records updated?

Records are not updated once the user has confirmed their team and relevant details.

3) Opponent – Fields:

- Tournament ID
- Team1 Points
- Team2 Points
- Team3 Points
- Team4 Points
- Team5 Points

When are records created?

When a user saves their game after reviewing results and statistics.

When are records accessed?

When the leaderboard requires the tournament rankings of a top 10 team.

When are records updated?

Records are not updated once the game has ended.

4) Leaderboard – Fields:

- Tournament ID
- Points
- Wins
- Draws
- Loses
- Goals Scored

When are records created?

When a user saves their game after reviewing their team stats.

When are records accessed?

When the leaderboard requires the top 10 list.

When are records updated?

Records are not updated but are refreshed when the leaderboard is accessed to get only the top 10 teams. The program will access a query that pre-determines the top 10 teams based on the highest points and uses the most current game before finding the top 10 teams.

System Requirements

Hardware:

- Intel Core i7-7500U @2.70Ghz
- 8,00GB RAM
- Nvidia GeForce 940MX (4GB) Graphics Card
- 1TB SSHD
- Screen Resolution: 1920 x 1080

Software:

- Windows 10 Pro (64bit) – *The Operating System of the computer that will be used for the project.*
- NetBeans IDE v8.2 and Java JDK 8 – *Used to create the GUI and programming language that will be used to code the program.*
- Microsoft Access 2016 – *Used to create the database that will be used to store data such as the players, user details and tournament rankings and stats.*
- Microsoft Word 2016 – *Used to create final documents relating to the project.*
- Google Chrome Version 84.0.4147.89– *Used for research and help purposes and getting image resources and player details.*
- PicsArt 8.9.0 – *A lightweight photo editor used for designing each screens GUI background.*
- Over 7.0.7 – *An iOS app used to create text and headings.*