Project: GameTrakker

Date: 12/7/2016

Authors: Richard Crossley, Bruce Davidson, Nickolas Holden, Robert Woods

Programming Languages, Database, and Markup Languages:

* Programming language = Java
* Database Format = Ms Access 2016
* How to Page = HTML/CSS/JQuery

**Objective:**

The concept behind GameTrakker is for the user to be able to track the scores of multiples game types. Currently, GameTrakker is designed for bowling and allows the user to enter bowling scores either by ending score or frame by frame. The scores can be marked as private tournament or league and are averaged by group and overall total. The scores are dated as when the game was played. The user can view the averages to see where they play best and possibility improve their game. The original idea was to have a PC app, a mobile app and a web site to allow different ways to track the information.

**Design Concept:**

The design of the project is more geared towards the program to work well with a user then have it look awesome. We started off with a navy, cream,red and tangerine orange color scheme that is not bright and is pleasing to look at. We’ll start with the login Page”, A user has the option to login or if he doesn’t have a GameTrakker account, register for an account so they can use the program. Once the user logs in to their account they are directed to “Main Form” where the user has many options to choose from. A user can choose to enter scores either by a general enter score form or a Bowling Scorecard form. If a user is in a Tournament or League they can choose the option to do so. The form has options to logout, delete account, Administration needs and exit program. Next is the “Enter Scores” form. This form allows the user to enter score individually for each type of game, private, tournament, or league play. The user can name a game and the put a manual date for the game. Next is the “BowlingScoreCard” form. This form is a bowling scorecard that the user can enter each frame of their games into the database. Once the user completes the game it’s displayed on the form and entered into the database. Next is the “Tournaments Function” of the program. Here is where a tournament official can enter users or players into a tournament style format and is displayed in a bracket formation. Next is the “Leagues Form”. This is where the user, if part of, can enter scores based on League play. There are options for the type of league they are in. Lastly, there is a form called “DisplayScores”. Here is where the user views all score that they put into the database, either by private, tournament or league. The form displays the game with the name of the game, date of the game, and score of the game. The form also displays an average for all games, and individual type of games. GameTrakker has menus to navigate the user based on the form they are on. Once the user is done using the program they can delete there present from the program and database. On every form is a menu that gives the user a wasy to navigate in in the program. Also, the menu has options to view help and about screens for the programs.

**Design Functions:**

* frm\_LoginPage:

*The Login button on the LoginPage form connects to the database and checks the LoginInfo table if it contains the username and password that were entered into the text boxes. If the username and password have a match in the table then it logs the user in and opens the Main form, passing the playerId into the Main form. If they don’t match then a message is displayed saying the username or password is incorrect and the login fails.*

* + Login button – logs user into the program
  + Register Button – Takes user to a registration page to register for GameTrakker
  + Exit Button – Exit program.
  + Menu Bar
    - File Menu
      * Exit – Exit Program
    - Help Menu
      * About – Shows about program
      * Help – Shows Help on using program functions
* frm\_PlayerRegistration:

*The Add Player button on the Registration form connects to the database and checks the LoginInfo table if it contains the username that was entered into the text box. If the username matches in the table then a message is displayed saying the username has already been registered and the player will not be added to the database. If the username doesn’t have a match then it adds the user’s first name and last name to the PlayerInfo table, and adds the username and password to the LoginInfo table. It also adds the playerId to both tables.*

* + Add Player Button – Takes First Name, Last Name, Username, and password and creates a user in the database so the user can use the program.
  + Clear Form Button – Clears the textboxs if the user has made a mistake on the form
  + Cancel – Cancels creating a user and returns to frm\_LoginPage.
  + Menu Bar
    - File Menu
      * Exit – Exit Program
    - Help Menu
      * About – Shows about program
      * Help – Shows Help on using program functions
* frm\_Main:

*This is the main page that links to all of the following forms*

* + View Score Button – Takes user to frm\_displayScores form.
  + Enter Scores Button – Takes user to frm\_EnterScores form
  + Tournaments – Takes user to frm\_TournamentsSetup form.
  + Logout Button – Logs user out of program and returns to frm\_LoginPage form.
  + ScoreCard Button – Takes user to the frm\_BowlingScoreCard form.
  + Leagues Button – Takes user to the frm\_Leagues form.
  + Exit Button – Exits program.
  + Menu Bar
    - File Menu
      * Delete Account – Deletes user from program and database.
      * Logout – Logs user out of program and returns to frm\_LoginPage
      * Exit – Exit Program
    - Menu
      * View Scores -- Takes user to frm\_displayScores form.
      * ScoreCard -- Takes user to frm\_BowlingScoreCard\_form.
      * Enter Scores -- Takes user to frm\_EnterScores form.
      * Tournaments -- Takes user to frm\_Tournaments form.
      * Leagues -- Takes user to frm\_Leagues form.
      * Player Information -- Takes user to frm\_UpdatePlayer form.
    - Help Menu
      * About – Shows about program
      * Help – Shows Help on using program functions
      * Date – Display current date.
* frm\_DisplayScores:

When the DisplayScores form loads it connects to the database and gets the player’s name from the PlayerInfo table, all their games from the games table, and their overall average from the BowlerAverages table and displays them. When the type combo box is changed it gets the specific game types and average types that are selected by the combo box and displays the games’ dates, names, scores, and gametype.

* + Main Menu Button – Returns user to the frm\_Main form.
  + Exit Button – Exits programs
  + Type ComboBox – Option to select what type of game is displayed.
  + List Area – Area to display the scores.
  + Menu Bar
    - File Menu
      * Delete Account – Deletes user from program and database.
      * Logout – Logs user out of program and returns to frm\_LoginPage
      * Exit – Exit Program
    - Menu
      * View Scores -- Takes user to frm\_displayScores form.
      * ScoreCard -- Takes user to frm\_BowlingScoreCard\_form.
      * Enter Scores -- Takes user to frm\_EnterScores form.
      * Tournaments -- Takes user to frm\_Tournaments form.
      * Leagues -- Takes user to frm\_Leagues form.
      * Player Information -- Takes user to frm\_UpdatePlayer form.
    - Help Menu
      * About – Shows about program
      * Help – Shows Help on using program functions
      * Date – Display current date.
* frm\_EnterBowling Scores:

*The Enter Score button on the EnterBowlingScores form inserts the date, type, name, and score entered by the user into the Games table. If no date is specified then the current date will be entered. If no name is specified then an empty string will be entered. If no score is specified then a score of zero will be entered. It also updates BowlerAverages table with the new overall average of all games and a new average for the specified game type.*

* + Enter Scores Button – Takes information from text and combo boxes and enter scores into database.
  + Clear Button – Clears text and combo boxes if user enters wrong information.
  + Done Button – Returns user to frm\_Main.
  + Menu Bar
    - File Menu
      * Delete Account – Deletes user from program and database.
      * Logout – Logs user out of program and returns to frm\_LoginPage
      * Exit – Exit Program
    - Menu
      * View Scores -- Takes user to frm\_displayScores form.
      * ScoreCard -- Takes user to frm\_BowlingScoreCard\_form.
      * Enter Scores -- Takes user to frm\_EnterScores form.
      * Tournaments -- Takes user to frm\_Tournaments form.
      * Leagues -- Takes user to frm\_Leagues form.
      * Player Information -- Takes user to frm\_UpdatePlayer form.
    - Help Menu
      * About – Shows about program
      * Help – Shows Help on using program functions
      * Date – Display current date.
* frm\_BowlingScoreCard:

*When the BowlingScoreCard loads it connects to the database and gets the player’s name from the PlayerInfo table and their overall average from the BowlerAverages table and displays them. When the type combo box is changed it gets the specific average types that are selected by the combo box and displays it.*

*The Enter Score button on the BowlingScoreCard form inserts the date, type, name, score, and frames entered by the user into the Games table. If no date is specified then the current date will be entered. If no name is specified then an empty string will be entered. It also updates BowlerAverages table with the new overall average of all games and a new average for the specified game type.*

*The Strike button and the Spare button on the BowlingScoreCard form will insert a strike or a spare into the enabled frame and then move on to the next frame.*

* + Enter Scores Button – Takes completed score with frames and inserts into database. Also it displays information about game in information part of form.
  + Clear Button – Clears Scorecard and clears labels and combo boxes on form.
  + Done Button – Returns user to frm\_Main form.
  + Strike Button – Enters a strike in the scorecard in each frame on ball 1 if user gets strike.
  + Spare Button – Enters a spare in scorecard on ball 2 if user gets a spare.
  + Options Pane – Gives more options to user for game type, date, and name.
  + Menu Bar
    - File Menu
      * Delete Account – Deletes user from program and database.
      * Logout – Logs user out of program and returns to frm\_LoginPage
      * Exit – Exit Program
    - Menu
      * View Scores -- Takes user to frm\_displayScores form.
      * ScoreCard -- Takes user to frm\_BowlingScoreCard\_form.
      * Enter Scores -- Takes user to frm\_EnterScores form.
      * Tournaments -- Takes user to frm\_Tournaments form.
      * Leagues -- Takes user to frm\_Leagues form.
      * Player Information -- Takes user to frm\_UpdatePlayer form.
    - Help Menu
      * About – Shows about program
      * Help – Shows Help on using program functions
      * Date – Display current date.
* frm\_PlayerInfo:

*The Update Info button on the PlayerInfo form updates the user’s email, phone number, address, city, state, and zip code in the PlayerInfo table with information entered by the user.*

* + Clear Button – Clears text and combo boxes if user makes a mistake.
  + Cancel Button – Cancel and return to frm\_Main.
  + Update Info – Takes information from text and combo boxes and update the users information.
  + Menu Bar
    - File
      * Delete Account – Deletes user from program and database.
      * Main Menu – Returns user to frm\_Main.
      * Logout – Logs user out of program and returns to frm\_LoginPage
      * Exit – Exit Program
    - Menu
      * View Scores -- Takes user to frm\_displayScores form.
      * ScoreCard -- Takes user to frm\_BowlingScoreCard\_form.
      * Enter Scores -- Takes user to frm\_EnterScores form.
      * Tournaments -- Takes user to frm\_Tournaments form.
      * Leagues -- Takes user to frm\_Leagues form.
      * Player Information -- Takes user to frm\_UpdatePlayer form.
    - Help Menu
      * About – Shows about program
      * Help – Shows Help on using program functions
      * Date – Display current date.
* frm\_About:
  + Exit Button – Closes frm\_About

**Database Design:**

The design of the database is composed of a total of 7 tables. Those tables are “LoginInfo”, “PlayerInfo”, “Games”, “BowlerAverages”, “Teams”, “League”, and “Tournament”. The “LoginInfo” table is used to store the username and password to allow each player to have their own account by giving each user a unique player ID. There is also a field to turn on administrative rights (a work in progress) to the developers/moderators of the application. The “PlayerInfo” table stores the user’s first name, last name, DOB, address, city and state of residence, ZIP code, gender, email address, and their sign-up date. While the application does not require this information at start-up, the user has the option to input this information at any time they choose.

The “Games” table is used to store the scores that the user enters from each game. These scores are divided among three different types. They are Private, League, and Tournament. Data can be entered into this in one of two ways. It can only have the final score entered or it can have the score put in frame by frame depending on which method they use in GameTrakker. Finally, each game is given a unique “GameID” number to keep them separate.

The “BowlerAverages” table stores four different averages from the games played and entered into the “Games” table. The user can then view their overall average.  Single game averages, league averages, or tournament averages are also available option if the user would like to see how they did in a particular area of gameplay.

The “Teams” table is used to store the information of bowling teams. Each team is given an ID number and allows for players to be associated with a team. These teams can have a maximum of four active players with alternates depending on the rules. While this is a work in progress, it will allow user to view the teams in the area and see who is on what team. Team logos would also be available to store in this table.

Though it is under construction, the vision for the “Leagues” table is to have the teams in a certain area be able to be grouped for a league season. Each league will be given a unique ID number to differentiate between each season and different leagues. Lastly, the “Tournaments” table will be able to store the information (scores, winner, etc.) of any tournaments played.

The main connection between all of these table is the “PlayerID” table as it is present in almost every other table to connect the user to all of their information from averages, games, tournaments, and league. Almost all relationships relate the player ID’s between the tables.