**CS 4750 Project—Football Database**

Yong Jae Kwon(yk7da), Taehyun Kim(tk9at), Andrew Yang(ahy9ng)

Department of Computer Science, University of Virginia

**Abstract:** This project’s main goal is to create a database and corresponding website that contains the data of all football players in the 6 football teams in Virginia.

**1. Project Information.**

**1. Introduction**

Our project’s goal is to create a database for a college football website. The users of this website will be able to search different football players in the 6 NCAA-1 level football teams in Virginia, and to get the match data of those football teams. Each player information stores the player’s height and weight, main class, address, and team. Each match information stores the date, the participating teams, scores for each team, the passing leader, the rushing leader, and the receiving leader. Each users will have an account that stores information of their favorite players. The administrator will have a separate account that can insert and delete players from the players table, which is linked with other school-specific and position-specific tables.

**2. Requirements**

Users should be able to create, sign into and log out accounts. Accounts will only contain the user’s username, password, and their fav\_player list. The information of individual football players will be stored in the Players table in the football database. The details for each player includes their number, position, class, height, weight, hometown, state, team, and their ID (which was generated respective to their team and number). These details will be stored in the main ‘Players’ table and the tables corresponding to the player’s school and team position. The tables have constraints to detect if a player is truly is in a certain school/position. The information of individual matches, which will be stored in the match\_2016 table, include the details of the two teams, the final score, the date and time, and leading players. Users will be able to search certain players either by their names, or by their team, or by their position. They will also be able to insert certain players in their favorite players list, and to delete items from the favorites list. The user can also export the Players table in JSON form. The admin user will be able to insert/delete players from the Players list, which, using triggers, will also change the tables corresponding to the player.

Security is handled so that it is safe against SQL injection attacks. There is a separate user that will be the only user which will be reading that login table. As mentioned above, there is only one admin that can do the insert/delete operations on the entire database, and the rest of the users can only select some tables or insert/delete the favorites table.

**2. Design Process**

**1.** **Design Decisions**

In order to make things easy for users to access and acquire information, we decided to use website as the interface application. While we didn’t used data encryption, for security we separated the data and code and prepared for possible SQL injection threats. We also granted different types of privileges for different users.

Most of the database is consisted of player data. The ‘Players’ table is consisted of the information of football players, including their current team and main position. There exists other tables that includes information of football players in a certain school or in a certain position (Example: table ‘uva’ contains all players in Virginia Cavaliers, table ‘QB’ contains all quarterbacks).

**2. E-R Diagram.**

Overall, we concluded that the system has a very narrow band of distribution, since TransLoc serves one purpose and is exclusively intended for those riding buses in the area.

**3. Database Schema**

. The tasks below apply to all riders regardless of user personas, unless specified.

**Tasks:**

1. **Check announcements**

Upon opening the app, the announcements are at the top of the screen.

**Use Case Description:**

**4. Edit and View Favorites**

* Story: As a rider, I want to be able to edit and view favorite.

**5. Directions from one stop to another**

**9. Alarm / Bus Notification**

* Story: As I rider, I want to be able to set an alarm for a specific line at a specific stop and receive a notification before the bus arrives.

**Environment:**

The typical task environment is the same for all riders.

**Usability Goals**

The usability goals should contribute to an overall better experience with respect to the primary usability goals; effectiveness, efficiency, and utility.

1. **Arrival timing features of the app should be as accurate as possible.** 
   1. This allows users to better manage their time and will drastically improve the application's effectiveness.
2. **The text based interface should contribute to a more efficient user experience.**
   1. Efficiency will stem from the compactness of the newly designed menu.

**Usability Requirements**

1. The interface of the application should be understandable to the user.
   1. This will be accomplished by having consistent transitions throughout the application.

**4. 3NF Proof**

The redesign of this application is focused on bringing a simple, intuitive, text-and-image based method of accessing the bus routes and bus system at the University.

**3. Evaluation of Product**

**1. Testing Procedure.**

As mentioned above, this project concentrates on fixing the main 3 usability goals of effectiveness, efficiency, utility, and the sub usability goals of learnability and memorability.

**2. Sample data and Sample Queries.**