Business Requirements

Document: Sports Score Tracker

1. Users should be greeted with a menu of options of sports to choose from, which navigate to new pages for the sport the user selects.
2. All pages should display background images for both UWP and Android versions of the application
3. Sports pages for each sport in design document which allows users to implement scores for a home and away team.
   1. Soccer scores should be implemented with a button which increments each team’s score by 1.
   2. Basketball scores should be implemented allowing users to increase score by 1, 2 or 3 in accordance with the rules of basketball.
   3. Tennis scores should be implemented in points of 15 for the 1st two scores and a score worth 10 points for the next score.
      1. Tennis score cannot go above 40.
      2. If tennis scores are level at 40, then a deuce should be initiated.
      3. Winner of a deuce should be made clear.
   4. Ice Hockey scores should be implemented with a button which increments each team’s score by 1.
   5. GAA game’s scores should be implemented using a goal and point system, allowing users to implement both home and away scores by goals and points in separate increments.
   6. Rugby should be implemented using increments of 2, 3 and 5 for a conversion, goal kick and try respectively.
4. Users should be allowed decrease the score for each team on the sport pages.
5. Each sports page should display headers for the sport, as well as a home team and away team header so user can differentiate between teams.
6. The games should have a match name in order to be identified for later use.
7. Sound Icon sound be available for user, which when double tapped either mutes sounds playing or allows sounds to play (different icon displayed for each, to show user if muted or not).
8. Sport pages should contain a save button which allows users to save game for later use.
   1. Users should not be allowed to save game unless a unique name is entered and cannot be left blank.
   2. When a game is saved a sound should play if the user chooses to leave sound on.
   3. If user chooses to mute sound on page, then when a game is saved then no sound should be played on saving a game.
   4. Saving should navigate back to the main/home page
9. A view saved games page should be accessible from the main/home page where saved games by user can be viewed.
   1. A list of the games saved should be displayed to the user on the saved games page.
   2. When a game is selected from the list the user should be able to update the name and score of the game (when updated should be saved), users should also see the data be dynamically updated in real time in their list.
   3. Users should be able to delete the game they have selected from the list.

Issues from Testing Process

**Issue Reference: 4.0 (compromise made)** When applying button’s to minus scores to correspond the increment buttons, when building the android version, I realised that there was a very limited amount of room available with which to work with. When making buttons much smaller to accommodate them, resulted in text being very small for the user. Minus button worked fine on sports where they only had one increment button for each team, but those with 3 for each, resulted in 12 buttons on the page, not alone causing the problem with text size mentioned above as an issue, but also making the page look overcrowded and took away from user experience and aesthetics drastically. As a result, I compromised by added a reset button which resets all scores to zero for all scores on each page. It also matches the void opposite the sound button making the pages look slightly more symmetrical in their element ratios.

**Issue Reference: 3.2 (fixed)** When looking back at my testing plan to test my project, I realised I had forgot to add a 1 pointer to the score for basketball, as this is possible in basketball when a player has 2 free throw shots caused by a foul in play (worth 1 each) and they miss one while scoring the other, resulting in one point, which I had documented in the test plan but forgot during implementation. As a result, I added that to the code.

**Note: All plan parts above have been tested on application and have passed**