---------------Week 1---------------

Attendance = ✅

**Preliminary Reading**

Project Specifications - 20% - 5pm, Friday, End of Week 6

Final Demonstration - 10% - Week11/Week12

Test Report - 10% - 3pm, Friday, End of Week 12

Workbook and Class Participation - 10% - 3pm, Friday, End of Week 12

Final Report - 50% - 3pm, Friday, End of Week 12

**Referencing Software**

* + Bibtex
  + EndNote

**Referencing Styles Accepted**

* + Apa
  + IEEE
  + Harvard

**Project Notes**

**Project 1**

* + New interface for users with accessibility needs
  + Create keyboard design features
  + Possibility accessibility movements
    - Eye movement
    - Cheek movement
    - Head movement
    - finger tips/hand movement
    - voice interface system
  + Code is assuming process exists

**Project 2**

* + Start with existing phonetic alphabet
  + create base data by recording sounds they make at good quality and then with interference
    - interference can be from
      * sound pollution
      * bad connection/static
      * confused with other words
  + find more efficient words to compare with

**Project 3**

* + How do you define intelligence?
  + what limitations do computers have that we do not, in terms of passing IQ tests.
  + How easily can they be automated at our current technology?

**Project 4**

[www.csse.monash.edu.au/~footy](http://www.csse.monash.edu.au/~footy)

* + Extend this probabilistic competition to other sports/competition
  + Get working for base case (football) and extend to other sports

---------------Week 2---------------

Attendance = ❌

**Preliminary Reading**

**Choosing Project 4**

* + Preliminary references (from competition and Monash library)

References for project

Andersson, P., Edman, J., & Ekman, M. (2005). Predicting the world cup 2002 in soccer: Performance and confidence of experts and non-experts. International Journal of Forecasting, 21(3), 565–576.

Dowe, D., & Lentin, K. (1995). Information-theoretic footy-tipping competition–monash. Computer Science Association Newsletter (Aus- tralia), 55–57.

Dowe, D., Lentin, K., Oliver, J., & Hurst, A. (1996). An information-theoretic and a gaussian footy-tipping competition. FCIT Faculty Newsletter, Monash University, Australia, 2–6.

Dowe, D., et al. (2011). Probabilistic and gaussian football tipping. Vinculum, 48(2), 10.

Wunderlich, F., & Memmert, D. (2016). Analysis of the predictive qualities of betting odds and fifa world ranking: evidence from the 2006, 2010 and 2014 football world cups. Journal of sports sciences, 34(24), 2176–2184.

**Concerns**

**INPUTS**

* + How will data be inputted into system?
    - For Users and results
    - SQL database select statements
    - Other database software
    - Simulated database
    - manual entering of data

* + How will fixture be formed?
    - Import from spreadsheet file
      * .xls
      * .xlsx
      * .csv

**Languages**

* + Python would be ideal for logic side
  + HTML/CSS/JS for webpage
  + PHP if database needed
  + If multiple clients need to access at a time node.js
  + If needed js can be used for logic side

**Risks**

Risks that I can identify now, may change later

* + Changes in fixtures
    - Different sports will have different fixture formats
      * Different number of games/teams
  + Databsae Functionality
    - If a simulated database is used, system cannot be tested as a whole for future use. No tests with database select statements will be implemented.
  + Security
    - Users will have to enter personal details to be stored/displayed on the database.
    - Users must know what data will be stored and concent to each item.

---------------Week 3---------------

Attendence = ✅

Project Plan

**TUTE NOTES**

* + Use example given to generalize to other competition
  + Use bibliography of footy example
  + Start with working for one competition, then try for more
  + Use simulated database
  + Website/end result where users put predictions each week
    - Results put in each week and predicative results given
  + Functionality > physical appearance
  + Read fixture in from file

**Project Requirements**

* + **Functional Requirements**
    - Describes how the system is supposed it work
      * Spreadsheet
      * Hosting platform
      * database
      * results
  + **Non-Functional Requirements**
    - Durability
    - Scalability
    - Security
    - Usability

Software architecture

* + Main classes
    - Tips
    - Results
    - Users
    - Fixtures

UML class using lusid chart

ips 
Tip ID 
User ID 
Team 1 > Team 2 Score 
Users 
User ID 
User Name 
User Tips 
U

---------------Week 4---------------

Attendance = ✅

Project Plan

* + Questions to raise
    - Will results be entered manually? Or can It be read from website
  + Internal specifications
    - Python for main logic on server side
    - HTML for output and display
  + External specifications
    - A fixture to input
    - Predictions from users
    - Results to input if not manual

Sample Test Plans

Unit Testing - Fixture Input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steps | Description | Expected  Result | Pass/Fail | Comments |
| 1 | Input fixture file into server | Server successfully imports the fixture file | Feedback was given about successful import | Feedback in the way of a JS alert |
| 2 | Server coverts fixture file successfully | File is formatted ready to appear in HTML | Pass if feedback for successful conversion is present |  |
| 3 | HTML page correct displays fixture file, correct teams/dates/matches | All fixture contents are present in HTML page | Pass if information in HTML matches the fixture contents |  |

Unit Testing - Simulated Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steps | Description | Expected  Result | Pass/Fail | Comments |
| 1 | Database successfully connects to the server | Server has connect to the database ready to import users | Pass if feedback for successful connection is reached | Feedback in the way of a JS alert |
| 2 | Server has read in all users and created their profile in the webpage | Users listed in competition as a player | Pass if user can log in and it registers the user as a player | A list of players will be accessible |
| 3 | HTML page gives option to user for them to put in their tips | Once user is logged in the input box will appear for their weeks tips | Pass if user can successfully put in their tips, and they are passed to the HTML for everyone to see |  |

Unit Testing - Logic Side Code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steps | Description | Expected  Result | Pass/Fail | Comments |
| 1 | Server reads all corresponding files (Fixtures, Users, Results) | Server has all necessary information to construct HTML | Pass if server successfully reads all files | Feedback in the way of a JS alert |
| 2 | Server creates calculations for user scores based on their tips | As a user enters tips, server uses log function and has results ready to be presented | Fail if server has an error while calculating tips | Fail will result in JS feedback and server refresh |
| 3 | Server sends all information to client to be displayed | Server sends (users, fixtures, user scores, results) to client | Pass if client successfully reads the information and displays on a webpage | This testing is reliant on the client page so we can identify valid results. |

System test cases to be drafted in later weeks

---------------Week 5---------------

Attendance = NO TUTE

**Project Design**

Answers to questions from week 4

* + *Will results be entered manually? Or can It be read from website*
    - *Will be inputted from a spreadsheet*

Internal Design part of plan

* + UML diagram will be best
    - Lucid chart
    - Paint

Main steps

* + Fixtures inputted into website
  + user logs into website and enters tips
  + server reads user tips
  + results entered into server
  + user scors calculated
  + displayed on website
  + results available for user to view

ixture Input 
Website 
ixtures Inputted ilnto 
User 
User logs into we

**Schedule**

* + Gantt chart
  + PERT chart
  + Kanban

Kanban seems most logical

* + Easy to follow
  + Colour coordinated (use colours for each category)
    - Client
    - Logic
    - Database
    - Fixture
    - Results
    - System whole

kanbantool.com

nputs 
In Progress 
+ add task 
Simulated Database Data 
Unit Tests - Cli

---------------Week 6---------------

Attendance = ✅

**Project Design**

Send in workbook

Security is an issue. Start without and add later

* + Idea = add a page asking for consent

Todays Questions

Hardcopy submission for project specification?

* + Not required

**Mockups for design based on plan so far**

* + User extenal design (lucid chart)
    - User should be able to see all games
    - Option to change the round
    - See other users
    - See all scores
    - input score with tips on how to input

ttp://probabilistic_sport.com 
NFL 
2018 
Season 
Team scores 
bold means 
Victorious 
John Smith 
#1234

* + Admin external design (lucid chart)
    - Should highlight admin is in control on design
    - Admin should be able to create competition
    - Admin should manage competition

itle 
http://probabilistic_sport.com 
Hello Admin 
Hi Admin, What would you like to 

---------------Week 7---------------

Attendance = ✅

**Test Plan**

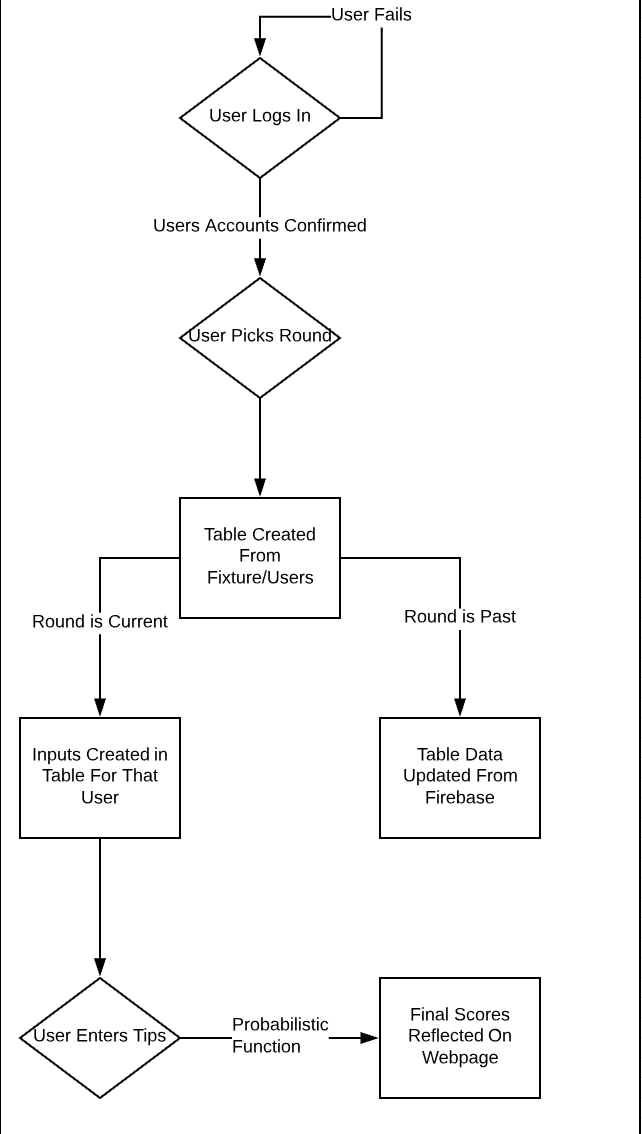
Test case ideas

* Unit Test – Fixtures Firebase
* Unit Test – Users Firebase
* Unit Test – Round Scores
* Unit Test – Total Scores
* Numeric tests for each above
* Unit Test – Create table
* System test

Test strategy

* Each module that is implemented provide extensive unit testing. Final system test may have many things to test, less chances of things failing is better.

**General Logic**



---------------Week 8---------------

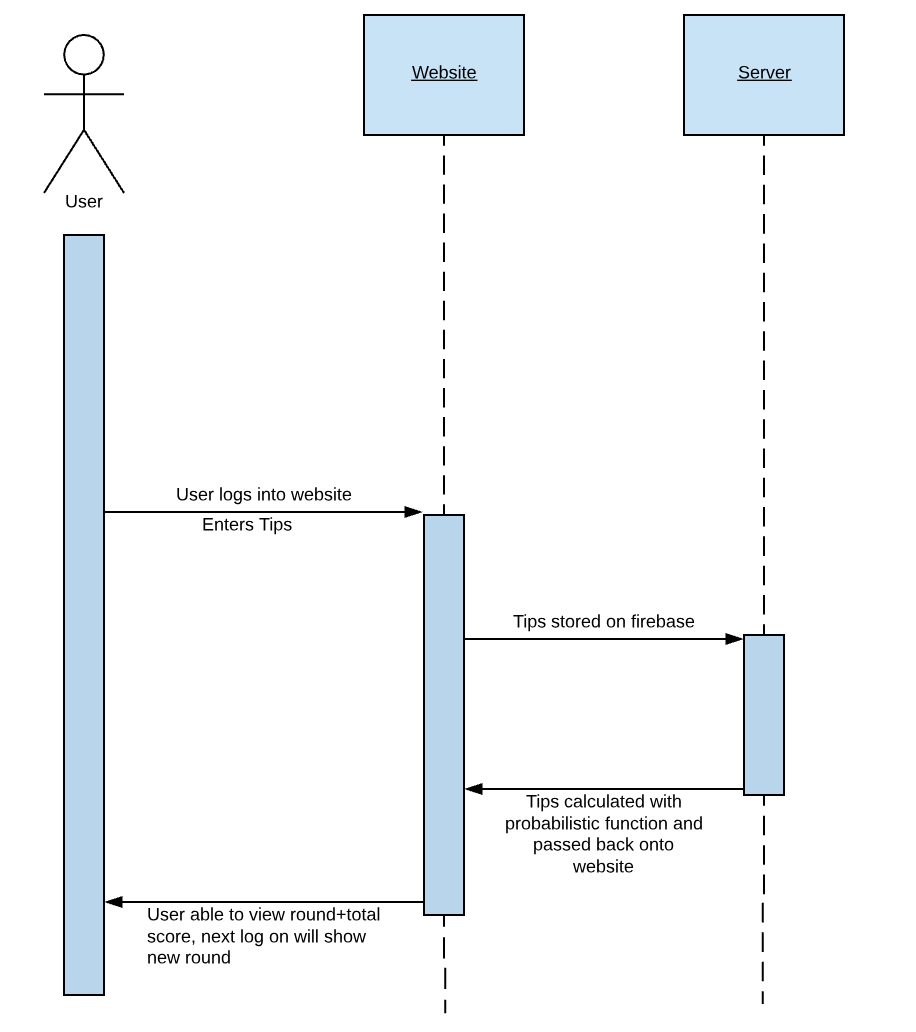
Attendance = ✅

**Implementation & Testing**

Modules Required

* SocketIO
* SocketIO-file-uploader
* Express
* Firebase-admin
* FS
* XLSX

Basic UML diagram



---------------Week 9---------------

Attendance = ✅

**Implementation & Testing**

Firebase for index Page

let config = {

apiKey: "AIzaSyDxnzTFBAMHLu9GAyDqGd5DctZ-hFDs6oA",

authDomain: "compsciproject-1dacd.firebaseapp.com",

databaseURL: "https://compsciproject-1dacd.firebaseio.com",

projectId: "compsciproject-1dacd",

storageBucket: "compsciproject-1dacd.appspot.com",

messagingSenderId: "887968300100"

};

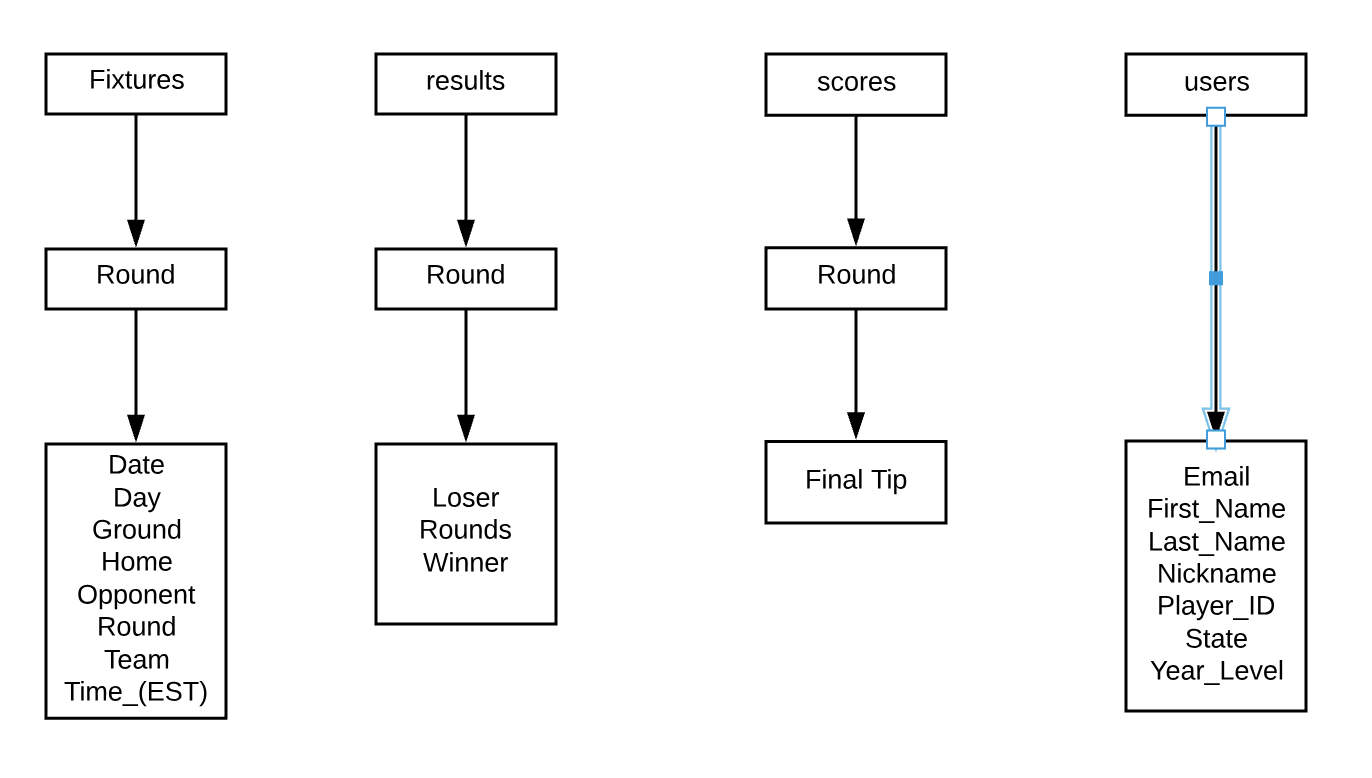
Firebase Structure



Useful code

// Initialise firebase references  
**let** ref = firebase.database().ref("/");  
**let** usersChild = ref.child("users");  
**let** fixtureChild = ref.child("fixtures);  
**let** gameDataChild = ref.child("gameData");

**Firebase Data Structure**



---------------Week 10---------------

Attendance = ✅

**Implementation & Testing**

Presentation notes

**Introduction/Aims**

\*\*\*Next Slide\*\*\*

Sport is an integral part of our culture and always had a strong role in how societies interact. Guessing which team will win is a natural consequence from this. In today’s society, many competitions exist where we play against each other to see who is better at picking the correct team to win. Some studies have shown past data can be an important factor for predicting which teams in sport will win. While others have shown experts and non-experts are almost just as likely to pick the winner. There are many methods you can approach tipping and the main purpose in this project is to extend an existing competition to a generalised form for all sports. With the completion of this project, a more inclusive and generalised version will be created. The idea is anyone who has a fixture can log onto the page and create their own competition to share to their friends via a link.

What makes this competition unique is instead of a standard Team A/Team B will win, a probabilistic function is used which rewards winners much more then losers.

\*\*\*Talk about function\*\*\*

A drawback to this scoring system is there is no great disadvantage for joining the competition late. Each week no tips are entered, the default score of 0 will be given to the participant. Seeing as scores of negatives are given should you guess wrong; many players will be in the negatives and score of 0 would place you in front of them.

**Method**

\*\*\*Next Slide\*\*\*

The server was implemented using node.js. This allowed the server to constantly be active while clients could join and leave as they needed. All calculations and data storage done on the server side. Player tips are stored in a spreadsheet, therefore data isn’t lost when server shuts down. At current implementation, the client page and entering tips cannot be done while the server is not running.

* Creating simulated Database
  + Creating a list of random names and data in an excel spreadsheet
* Creating the table In HTML page
  + Used a module called XLSX for Javascript which pulls data from a spreadsheet. The fixtures were a saved spreadsheet file of the 2018 system
  + On first implementation, it was required the fixture follow this input of rounds, matches etc.
  + This implementation will be achieved with an algorithm that searches the first row for key words such as matches or games. If it is unable to verify which column is needed, manual entering of the column number is possible. Using neural networks in the future would be useful to make the algorithm better at importing different spreadsheets
* Adding tips from players
  + In this case, the server saves the data into a spreadsheet, the use of database and server less functions in the future would allow the server to be turned on and off
* Import results into table
  + Similar process for importing fixtures, using XLSX module in Javascript
  + Also, implemented so manual results can be entered
* Calculating scores based on tips
  + Uses probability function defined earlier and stores data for client to see

\*\*\*Next Slide\*\*\*

**Screen shots**

* Admin create/manage page
* New client page
* Idea of how table will look

\*\*\*Next Slide\*\*\*

**Drawbacks**

* Simulated Database
  + Further work will be required when implementing this in modern example.
  + The use of google firebase in the future will provide a constant database for data to be extracted from, this will allow users to constantly update tips, change their details or remove themselves from the competition.
  + The logic for this will remain the same, just importing users from firebase will be the difference
* Spreadsheet inputs
  + Due to the multiple different types of spreadsheets one can receive for use, the algorithm can struggle to work out the rounds, teams. As mentioned before neural networks would be an interesting way to improve this.
* Centralised Server Needs To Be Running
  + At time of implementation the website will require the server to be running on a local machine to process the logic for website.
  + No data can be entered or fixtures viewed while server is off.
  + Future implementation would use server less functions such as google firebase functions and firebase database. This would allow a consistent server to be running, so users can log in, enter tips and the server is there to receive it.
* Security
  + Seeing as SQL will not be used SQL injections are not something that will need to be worried about, firebase uses their own security and data validation techniques which can be implemented alongside firebase code.

\*\*\*Next Slide\*\*\*

**Conclusion**

With the use of firebase, the extent to which this competition can be used is endless. Neural networks could also be used to make the competition smarter about its inputs and more predictive, this project would be a great demonstration of how future technologies can make a sizable impact in the way we implement programs and share them around the world.