**Project Design Document Template**

Divide your design document into sections that address each bullet point in paragraph or visual form,

being as comprehensive as possible. You have the liberty to reorganize these sections as makes sense,

as this generic template may not be perfect for every idea.

**Discovery**

● Idea: -

A web-app to record internal competition results, display competition standings, store member contact details and provide historical competition lookup functionality. Currently the club is using a combination of paper leagues on a notice-board and email notifications.

The web-app should be easy to use, offer easily readable displays and be scalable for mobile use.

● User Personas- Who is using your project? What is your target audience, and why would they bother to start using your project?

○ In describing your problem/solution, be sure to mention any industry/technical

background as well. Assume the reader only knows the code side of things.

Primary users = Club members who play internal comps, they currently email results to other club members that run the comps. It would be a lot easier for them to register results via an app.

Other users = those members currently running the comps

● User Stories- walk through ALL the functionality of your project in paragraph form, as each user

would go through it (e.g. start with login, and go from there to a main menu, etc.)

○ Reference appropriate wireframes/visual mockups as needed

○ Create a separate user story for each user persona (e.g. admins and users would have

different user stories)

‘Player’ user:

* Logs in with email address & password (plus ability to reset password if forgotten via usual scenario of email sent to reg email address, link clicked and password reset) , and then presented with following options (via tabs, button, Menus, etc):

1. Profile/Account: Shows Name, and editable details for: email address; logon password; home phone; mobile phone; membership type (?)
2. Current Comps: Shows comp name and all available matches; gives the ability to record scores against a match, and edit the score if submitted incorrectly. Also shows results already recorded for current comps. Comp name must first be selected eg. listbox. Can only enter/edit own results. Admin user can edit any result
3. Historical Comps: Gives the ability to view historical results with a range of filters eg. player name, comp name, year, etc. Should default to current comps

‘Admin’ user:

* All the ‘Player’ users functionality plus an extra option …

1. Comp Admin: Gives the ability to create/delete/edit comps and add remove players into those comps

‘Super’ user:

* All the ‘Admin’ users functionality plus an extra option …

1. User Admin: Gives the ability to edit the profile of members, ie. make them ‘Player’ or ‘Admin’ users. Also gives the ability to edit member profile details

**Design**

● Wireframes

○ Include visual representations of every single front-end component of your application as

it would appear to the user (computer-created, not hand-drawn)

● Entity Relationship Diagram

○ Identify each of your project’s components, and how they will relate/communicate with

each other. This should be portrayed visually, with arrows showing how each of the

components are linked.

○ Include any database schema (tables, columns, foreign keys, if applicable), as well as

what sort of data storage your project is using (traditional relational database,

nonrelational database, blockchain/distributed storage, etc.).

Possible Relational DB

Tables:

1. User: (User\_Id, Email, Password, First\_Name, Last\_Name, Phone\_1, Phone\_2, Mem\_Type, User\_Profile)
2. Comp (Competition): (Comp\_Id, Comp\_Name, Comp\_Type, Start\_Date, End\_Date)
3. Comp\_Pariticpants (find a better name): (Comp\_Id, User\_Id, Cummulative\_Points\_Total)
4. Matches: (Match\_Id, Match\_Name, Comp\_Id, Player1\_Id, Player2\_Id, Player1\_Games\_Won, Player2\_Games\_Won )

Relationships:

1. A Member can have > 1 Matches, a Match can only have one Member (?)
2. A Comp can have >1 Comp\_Participant, a Comp\_Participant can only have one Comp
3. A Member can have >1 Comp\_Participant, a Comp\_Participant can only have one Member

● List the proposed technologies/languages/frameworks/libraries you plan to use for each specific

component

Some popular technologies that if possible I'd like to incorporate into my projects ....

Flask / Django

React

PySpark, Pandas, NumPy, SciPy, scikit-learn

Jupyter Notebook

Jenkins, Git/Github

AWS / GoogleCloud

I’d like to discuss how we can get the best fit for the above into the 2 projects I’ve got.

I think the Squash-Club app could easily fit into the Flask model we’ve already covered either using SQLite, Postgres or MySQL for storage managed by the Python ORM design with APIs called by a web-app that uses JS/CSS/HTML.

In fact that would probably be a good way for me to cement what I’ve learnt using those technologies.

Although I did have the thought that because I’m so comfortable with relational databases perhaps I should be using some other approach to data storage and retrieval ! I don’t know though, for the purposes of getting something working faster, that I know works and can easily work with perhaps the RDB is better.

The EPL betting strategy app lends itself well to the use of PySpark, Pandas, NumPy, SciPy and Jupiter notebook for data manipulation and modelling but out of the above technologies I mentioned the couple I’d really like to use are React (again to cement what we’ve done) with either AWS or Google Cloud, in fact Google Cloud would be better.

**Development/Deployment**

● Describe what the final project would look like (If you were developing it full-time with unlimited

time). If that is outside the scope of what you could do in Phase 3, describe what paired-down

version you plan to achieve in only 3-4 weeks.

● Describe how you plan to deploy and host your project once it has been developed.

Possibly host video content of matches, for viewing by users, uploaded by me

Potentially offer some stats analysis, ie. avg points conceded per game/match, etc (to be decided)

EARLIER NOTES:

\* Website for squash club members to enter internal league results, and knockout comp results

\* Site should report current league positions and all results for that 6 week period

\* Site should maintain results/final league tables history, searchable by user

\* Should display knockout comp results, current round of comp, and comp history details

\* Potentially offer some stats analysis, ie. avg points conceded per game/match, etc (to be decided)

\* Internal leagues run in 6 weeks blocks, all year round

\* Admin access should allow creation of new leagues with different participants every 6 weeks

\* Admin access should allow creation of new knockout comp, allocation of dates to rounds (comps take a few months to complete)

\* Both male and female sections

\* Website should also display contacts details

\* Each user should have their own username/password, plus ability to recover/reset password

\* Should be simple to use, offer easily readable displays and be scalable for mobile use

\* This will actually be used by my squash club, so I guess I need to figure out what it will cost them to host somewhere ?

Video footage