

MyGamePlan - Full Stack – Assignment : Timeline

1. Preparation of the project (40min):

- Storyboard - creation of a sketch of the application: 20min
- Analysis of the database – MongoDB Compass: 20min

2. Backend – Nodes.js & MongoDB (2h30)

- Connection to the MongoDB database: 15min
- Creation of the backend routes + Testing with Postman: 2h15

3. Frontend – React Components, Styled Components, React Hooks (5h30)

- Creation and design of the components : 4h30
- Queries and displays of data (teams and formations) for dropdown inputs: 1h

4. Management of data useful to several components – Redux (2h)

- Creation of the redux store : 10min
- Creation of the inputValuesReducer and associated files : 40min
- Creation of the crossesReducer and associated files : 40min
- Queries and storage of crosses using Redux Thunk : 30min

5. Retrieving data from the Redux Store and processing the data to display the required information – Data Processing/Algos, React Hooks, React Components, Redux (4h30)

- Display the statistics about the club (2 charts): 1h
- Display the statistics of the 3 best crosses takers: 1h
- Display all the crosses of the club on the field: 2h
- Display a pop-up when the user hovers over a cross: 30min

6. Making the site responsive – CSS Media Queries (1h30)

7. Switching the front-end from JavaScript to TypeScript (4h)

(since I still had some time, I thought I would show you my TypeScript skills)

8. Update - When the user clicks on a player, show only the crosses made by this player on the field (1h00)

9. Deploy the application on Heroku (10min)

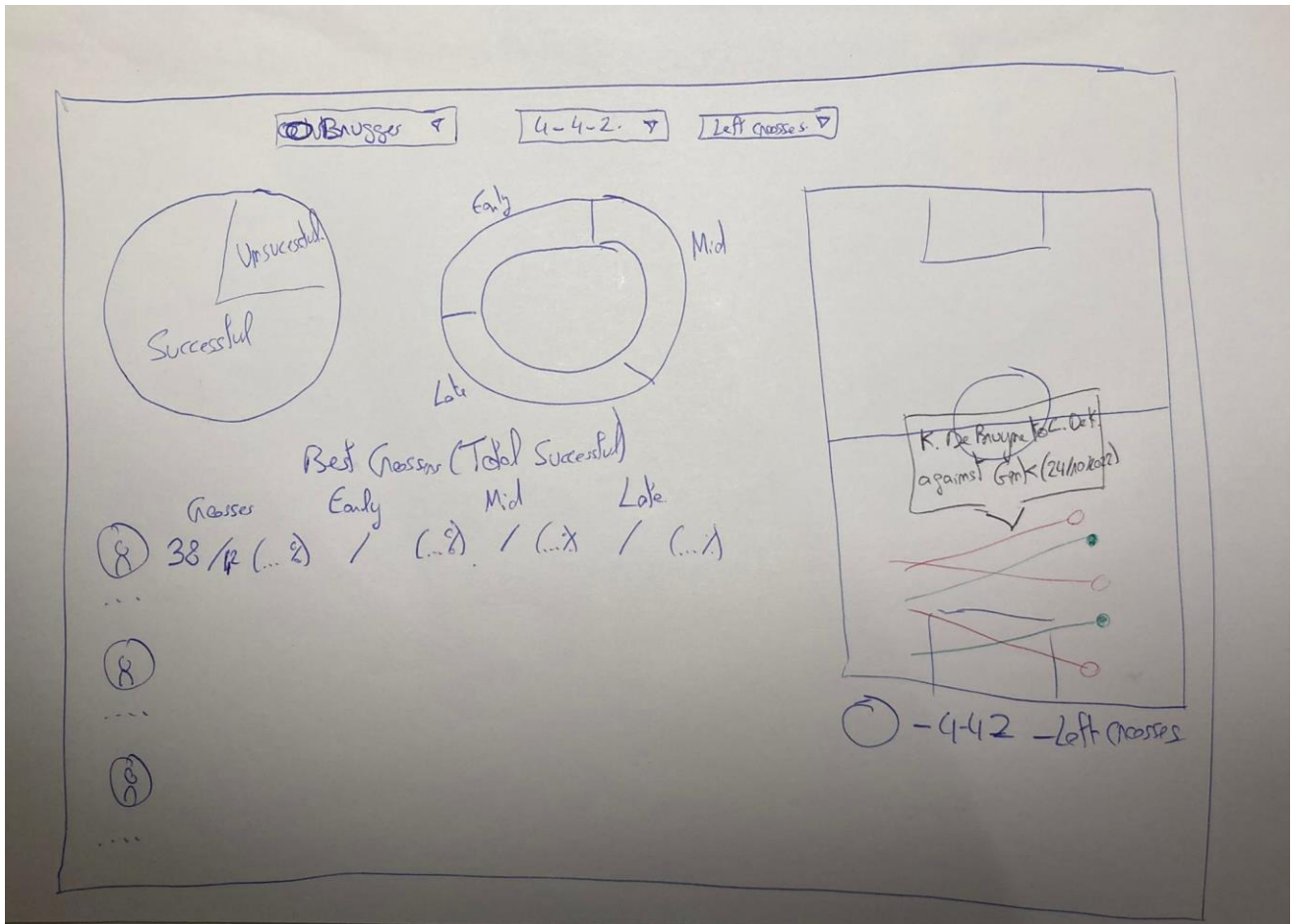
10. Update 2 – Adding Redux Persist to store the last consulted club in local storage and redisplay it the next time you open the application (10min)

Total time spent on the project:

- To meet the instructions (items 1-6): 16h40
- To improve the project (items 7-9): 5h20

=> Total time spent: 22h

Sketch of the application:



Calculation of the information needed to display a cross:

To display a cross (represented by a div element and positioned using CSS), you need:

- A starting point
- A length
- A rotation angle (rotation from the center of the element because that's how CSS works)
- A translation according to y and a translation according to x (to reposition the starting point at the right place)

