## Progress so far

1. Storing the whiteboard server-side meaning when a new user goes to access an edited whiteboard, they are able to see previous changes
2. Allowing for the functionality to undo changes made to the whiteboard, as well as across networked users

# Storing the whiteboard to the server

This, has been a partial success so far. Currently a whiteboard is stored on the server which records any changes that are made, and if you access a whiteboard with changes it will appear on your device, however I am yet to store it into a database. The changes that are made to a whiteboard only last as long as the server is active, and once the process is closed the progress is destroyed. This is because I am now at a point where I have to make a major decision in my project – which database management system to use.

After some research from a long list of default MySQL, Oracle, Microsoft SQL server, PostgreSQL, MongoDB & MariaDB I concluded MongoDB is the way forward. It was a tough tie between MongoDB and PostgreSQL, with MongoDB seeming quite fit for purpose considering the nature of the whiteboard and what might be stored in the whiteboard (shapes, images etc) which could be rapidly changing and with all having different requirements on what data needs to be stored for them. PostgreSQL was quite attractive for its scalability however but that was not compelling enough to settle on it.

|  |  |  |
| --- | --- | --- |
| **Database management software** | **Pros** | **Cons** |
| MySQL | Very standard, lots of documentation. Have experience with this before. | Not particularly suitable for my purpose not for a dynamic website. |
| Orcale | Nothing, its awful! | Its slow & costs money. |
| Microsoft SQL Server | Developed by Microsoft, fast and stable. | Suitable for a business environment, not for a dynamic website like the whiteboard. Also costs money! |
| PostgreSQL | Incredibly scalable for if there was a surge on hits to the website | Limited documentation |
| MongoDB | Incredibly versatile, fast | Bit of hassle to setup, doesn’t use SQL so little bit of a learning curve |
| MariaDB | Fast & stable | Quite a new SQL type, but a better version of MySQL essentially. |

# Undo

This was something I assumed would be a quick thing to add, but it ended growing into quite a headache. My first attempt to implement an Undo function was multi part. Two things had to be achieved

1. The line being undone
2. It happening across all clients connected to the whiteboard.

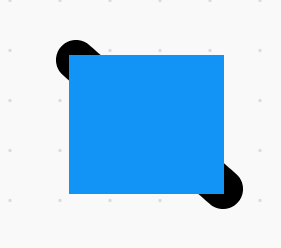
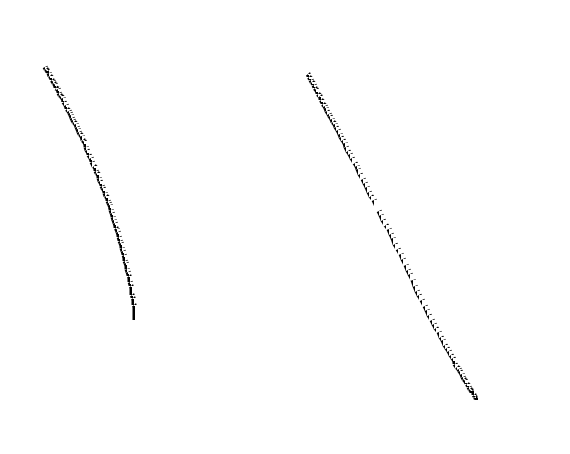
**The networking part**

First was removing a line – this I thought would be easy, however it turned out to be the opposite. To be able to keep track of objects on the whiteboard, I had to store every coordinate of the line in a table as its being drawn, and once the line is completed move it to a master table which keep track of all the lines on the whiteboard. This needed to also support if other users were drawing a line at the same time. To do this, whilst the line is being drawn, for both if the user is doing it or a networked user the line is stored in a special table identified by either “self” (if it’s the user drawing) or their socket ID (if it’s a networked user drawing), and once the line is completed (which meant I had to fire a network event for once the line is completed too) it is moved to a master table keeping track of all lines on the whiteboard and the special table is wiped clean ready for the next line. The database will mirror this function server-side (currently just done in a table but achieves the same effect).

**The drawing part**

The whiteboard uses the HTML canvas element, and whilst it offers a drawLine function, the only way to remove lines is using clearRect (clears a rectangle with supplied dimensions). This presented 2 problems. The first being lines aren’t rectangles – they can be diagonal in shape, meaning it wasn’t deleting the line as expected. In the first image below, you can see the blue box being a clearRect, and the black line being the drawLine. It was causing a weird effect removing partial bits of the lines, but not as intended. You can see the effect in 2nd image below. The other problem was if you had 2 lines overlapping, it would delete any line under or on top of the lines you were trying to undo.

I went back to the drawing board and tried a new approach. I ended up removing the last line drawn from the table that tracks all the lines mentioned previously and redrew the entire canvas. From an optimization standpoint, it’s not ideal nor the way I wanted to do it – but far as I can see it’s the only way, and it gets the job done.

# What’s next?

1. Implementing the database
2. Adding permissions to whiteboard
3. Implementing zooming and dragging around the whiteboard
4. Adding more options, for example shapes & images
5. Fleshed out UI & mobile support (including live viewer list, etc)