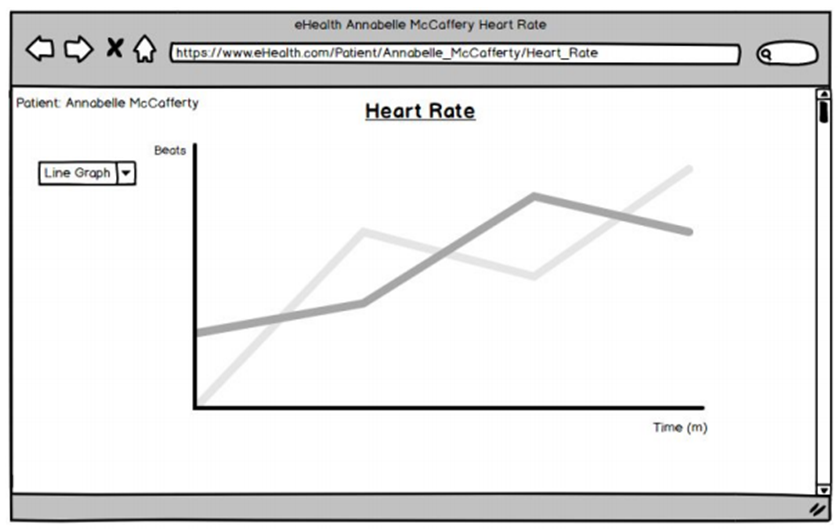
**Critical Reflection**

During the developmental phase, we encountered problems that forced us to change the underlying software structure and user interaction previously proposed in the planning phase of the software development lifecycle.

One major change we had to implement was the presentation of patient data. The original structure was to enable the user to fluidly change between the different data representation graphs (Line, bar, pie) dynamically, however complications during the development of the app forced us to statically change the graphs and not the way it was initially planned. The justification as to why we decided to change to static graphs is because of the predicted time consumption during development as we used an API to display graphs and data, meaning it would be difficult to implement our aforementioned design within the time constraint.



Another major change in our implementation is our slight deviation from the class diagram developed during the planning phase. The original idea was to only have one homepage where both doctors and patients can access however, as we developed the app we decided to separate the doctor and patient homepage from one another as we wanted to have different views and functions for both doctor and patients. Another slight reason was because of UI design, as we wanted to make the UI for the patient view to be as user friendly as possible, whereas the doctor view will include a more professional view.

This change in implementation also inadvertently affected how the structure of the database would be as it was initially designed to have a table for user, patients and doctor but due to the separate views and structure of the app, we had to delete the user table and migrate its attributes to both the patient and doctor table. Also, the attribute “password” and “address” for our tables are reserved words in php, and while running SQL in php using the reserved words are fine, it is better practice to change them to non-reserved words to avoid unforeseen complications.

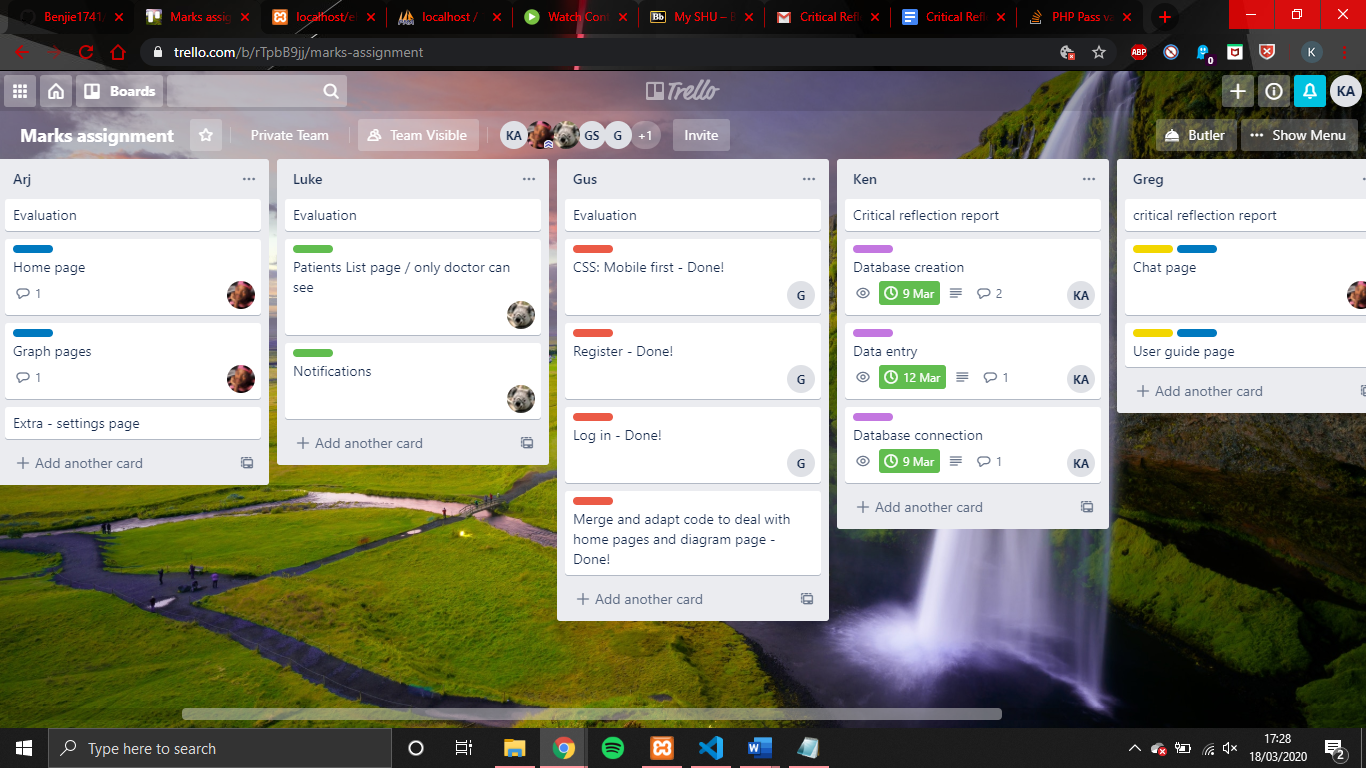
A small change regarding notification is that while we planned to have a notification system for when a patient uploads new health data, we decided that it was impractical and instead changed the notification to be when there is a new chat message between the doctor and patient instead.

Lastly, we did not include a detailed list of attributes in the class diagrams regarding chat as we had little experience in implementing a chat function and because of this we a prototypical class akin to what we thought a class diagram would include.

As for our project management approach, as a group we settled on using Github for storing our repositories and version control, and Trello for workflow management. We chose Github and Trello because all team members are experienced/acquainted with the tools and its functionality.

We used Github mainly for its version control, as it allowed all team members to be able to work concurrently with everyone else on different functions for the app, ensuring there is a continuous flow of work during development, mitigating the chance of inactivity when waiting for a team member to complete work. We extensively used the branch functionality to achieve this as well as including our individual work on a separate branch from the master to avoid cluttering and duplication of functions within the branch.

Trello was used for workflow control. We took advantage of Trello and its functionality by ensuring that every team member is assigned work using the Trello cards. This is important as it allows everyone to set dates and deadlines for each implementation we needed to add, as well as creating notes and comments on each card for transparency between group members. This helped the group by keeping everyone on track and ensuring that everyone has something to do, and with the help of branching in Github, we we’re able to complement the branches with the assigned card we were tasked with on Trello.



In review of our software project management, we believe that our choice of repository, version and workflow control works well in tandem to each other as we were able to translate each functionality on the cards in Trello to a branch in Github during the development of, ensuring a consistent and structured approach to our development of the app.

In conclusion, we had encountered unforeseen problems during the development phase which required us to change part of the underlying structure of the software. However, due to our use of workflow control in Trello and version control in Github, we were able to mitigate much of the problem as it allowed us to be flexible as a group when facing such problems. This allowed us to acknowledge the importance of correct workflow control within the real world as it would be nearly impossible to foresee future problems before actual development and by implementing the branching functionality within Github, we were able to lessen the loss of work efficiency be allowing other members within the group to continue working even if one member faces a problem during development.