

# ***Design Report***

*Healthcare Record Implementing Blockchain*

*3<sup>rd</sup> Year Project*

*Traian Svinti 14432128*

*David Talan 14387991*

*Supervisor: Geoff Hamilton*

## Summary

This report presents the design of a Healthcare Record Application that uses Blockchain as database. This design makes use of HTML/CSS and Javascript for the user interface, with BigchainDB, MongoDB and Django being used in the back end.

For the User Interface, log-in screens were implemented in different places in the case either a doctor or patient wants to use the application. This would then lead to different tabs that they can access to view the medical record.

For the database, the group found a blockchain database, BigchainDB that runs in conjunction with MongoDB. Later, Django was implemented to connect the application to the database.

Overall, the User Interface objectives were met. However, connecting it to the blockchain database posed a big problem for the group.

## Introduction

### User Interface Design

As shown in figure 1, the UI mock-up the group created has a very simple design. This page specifically shows the allergies section of the patient's medical record. On the top side of the page, it has the patient's general information displayed, with the profile tab and log out button on the top right-hand side. Below, to the left, it has the different tabs that lets you navigate through to different aspects of the medical record. The part of the record you choose is then displayed beside it to the right.

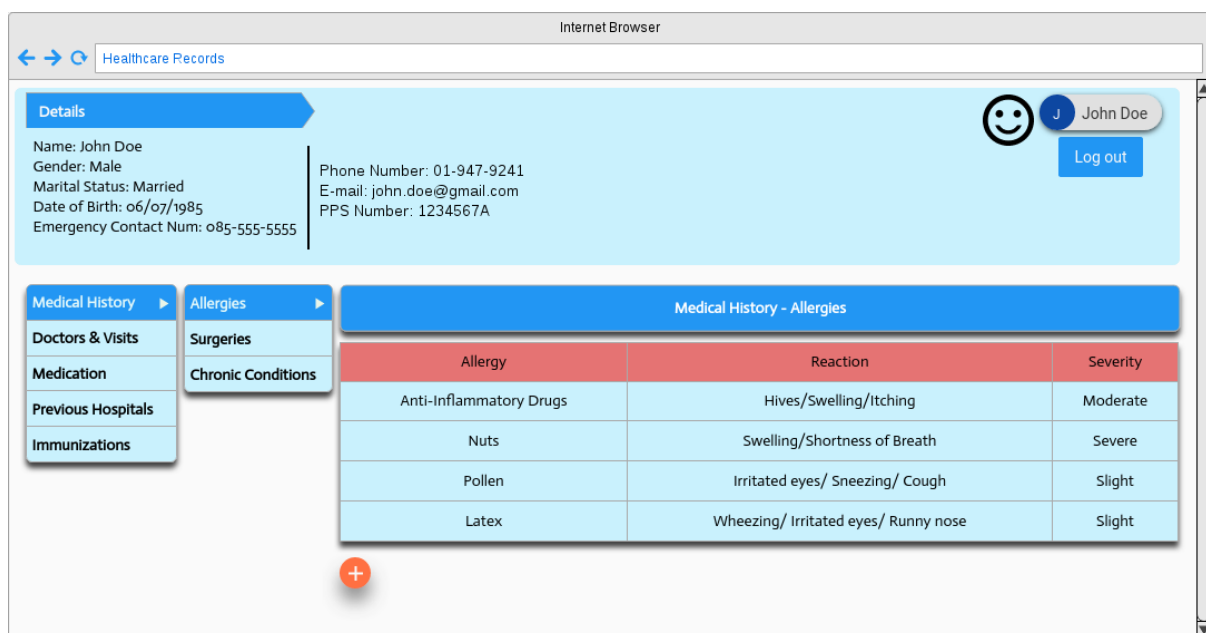


Figure 1. UI Mock-up of the allergy section in a medical record.

With the Requirements Analysis and Data gathered by the group, problems arose from the initial UI mock-up that was made. As shown in figure 1, the feedback from the users stated that there was too much information being displayed all at once.

We did receive positive feedback however on the second UI mock-up that we presented in the questionnaire. As shown in figure 2, the current tab is on 'Doctors & Visits' which shows a page with the relevant information. The objective was to have easy to read cards of the previous doctors that a patient may have dealt with before and their contact information. There is also the information of when they last visited those doctors. Some of the users stated that the cards were "very nice looking, presented information concisely".

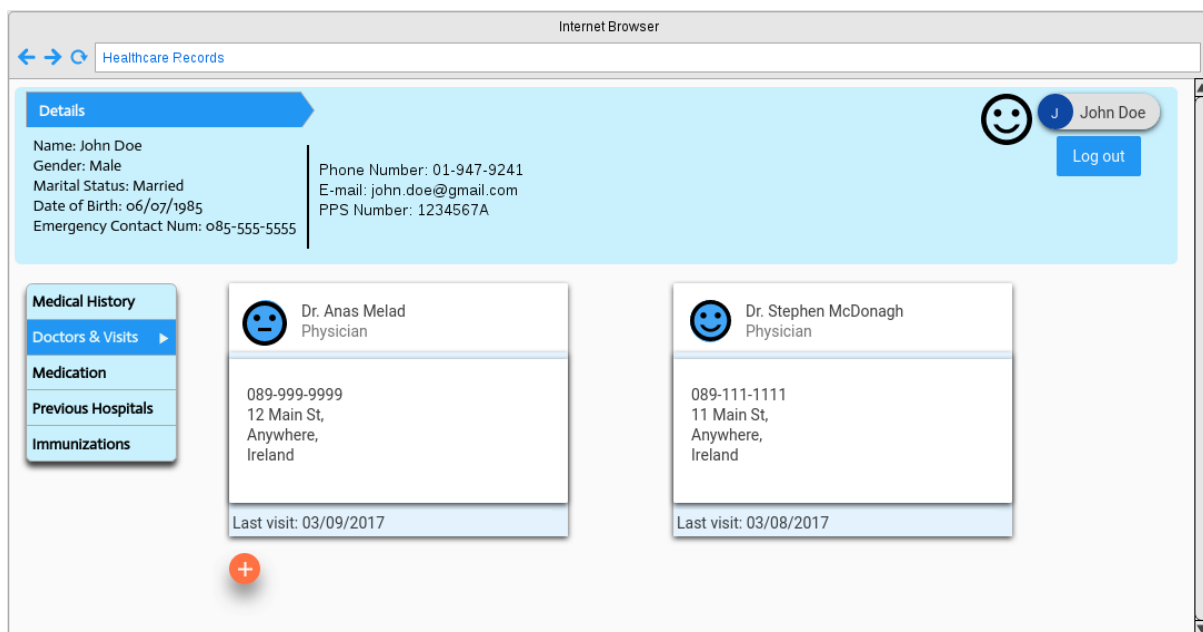


Figure 2. Page showing the patient's previous doctors and the last visit.

## Database Design

For the database design, figure 3 shows an architecture diagram of how it would work. You have the client machine (computer) where the user can view the records and it is connected to the internet, which then accesses Django. Django would then authenticate the users, which would then let them access their records in the blockchain database.

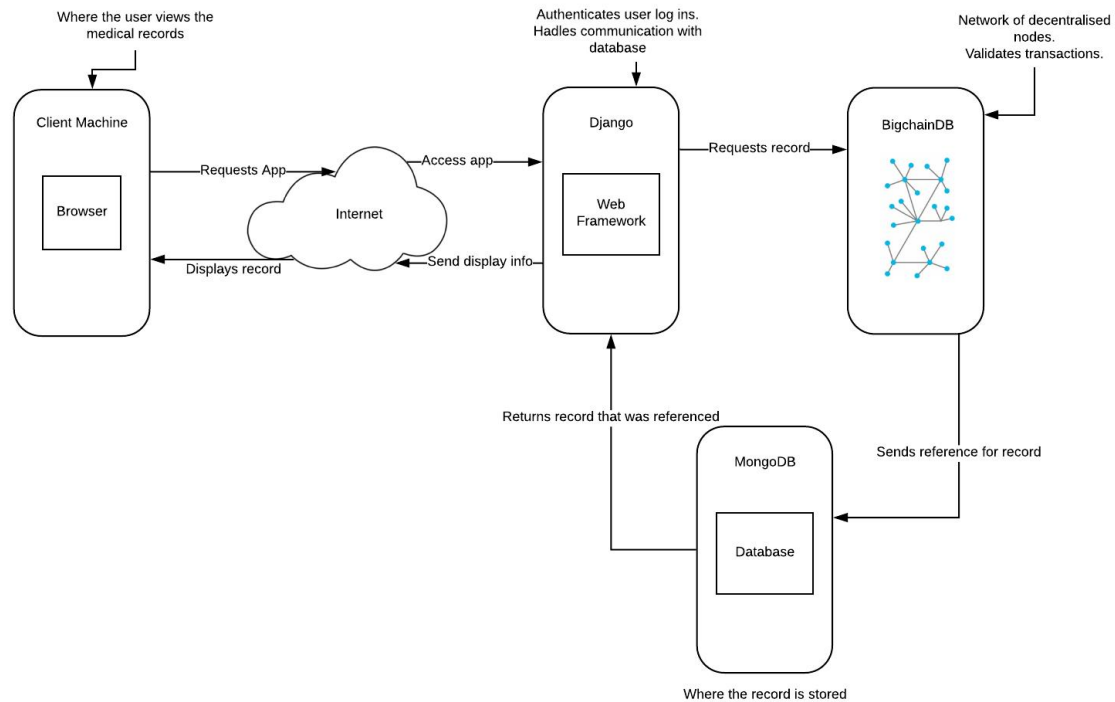


Figure 3. Revised system architecture.

The blockchain database handles the storage of the medical records, incorporating public and private keys to encrypt the transactions. When a new block is added, the different nodes in the blockchain network must approve of the transaction and validate it.

Initially, the database ended up being more complicated than what the group anticipated. The group initially thought that the data being stored is being stored directly into the blockchain. However, while dealing with the blockchain application itself, the group realised that the blockchain stores references to the data being stored in MongoDB as shown in the figure 3.

## Resolving Design Issues

### User Interface

The initial design concept for the UI mock-up was to constantly display the patient's general information on the top of the page. The idea was that the doctor would need easy access to their details in case they need it. With the feedback received from the questionnaire, the consensus was that the page presented too much. It made it seem very overwhelming to the user. Having it placed in the top position meant that they are more likely to read it first, and the group also intended it to stay there in every page. Additional feedback received also disproved the group's idea that the information is needed to be displayed all the time. One user stated that it was "unnecessary and distracting".

The group resolved this issue by simply moving the patient's general information to a specific page. The tabs were also moved on to the top of the page instead to allow more space when displaying medical information.

The questionnaire also raised a problem that the text can be quite difficult to read so the group opted to use a different font. The change in font style, size and the use of contrasting colours now makes it easily readable to any user. The buttons that were created also incorporate these properties so that a user won't miss them.

## Database

A problem that the group faced during the design of the application was connecting the web application to the blockchain database itself. Django was chosen to be implemented to do this and to query a non-relational database. The group was hoping for it resolve said issue but the group faced another problem. When Django was finally set up and running, it was found out that Django doesn't have a streamline way of connecting to BigchainDB. Also, when the BigchainDB, MongoDB and Django servers are running together, the group discovered that it is unable to add blocks to the blockchain. This is problematic in case for example, when needing to modify a patient's details is needed. The group discovered that Django caused this problem.

The group couldn't resolve this problem. This is probably due to BigchainDB being a relatively new application and has little support at the moment.

## Conclusion

---

This report has discussed development of a healthcare record application implementing a blockchain database. The objective of the development is to provide a user interface that is easily usable by our target demographic and a blockchain database for its advantageous properties. Only one of these objectives were accomplished. However, the blockchain database itself is functional, but the group is unable to implement it fully with the web application. Right now, there isn't a solution online that the group found to fix the Django and BigchainDB connection but surely in the future, when there is further development and support for BigchainDB, this problem should be resolved.

This project has introduced the group to a range of different applications that the group hasn't encountered before. The knowledge of how a blockchain works, how it can be used as a database, and using Django as a web framework will surely help us in the future.