Backend Speaker Notes

Early in the projects development, we decided that implementing an administrative backend would be a key requirement in order to make the system adaptable and extendable without requiring technical knowledge. The goal is to make the process of adding or editing information in the webapp just as intuitive as using the site itself.

Initially we looked at the pure functionality of the backend and the actions it should be able to perform. Through this flowchart, a few key features became apparent:

* With such an emphasis on ease of management, we would need some kind of interface through which records could be edited.
* We would need a reliable and easily editable way of storing the information such that it could be modified within this backend program.

The first of these points gave rise to Sandman. Sandman is a backend administrative tool designed from the ground up to be tailored to DREAM’s needs. We first created the webpages through which administrators could interact with Sandman, the first of which being a dynamic login page. This page would be a secure method of logging into Sandman with either a username and password or linked social media with options for account creation and approval.

The next webpage we made was the administrative portal, where the files could actually be edited, added and viewed. Executing the three options needed to be intuitive and simple and so we implemented pop-up web forms that allow for the addition of data by filling in the respective fields. Through this portal, the administrator can also manage dontaions made to DREAM through paypal or other 3rd party payment platform.

Upon completing the form to add a new record, the information entered would be instantly propagated to the content management system and thus to the users of the webapp. The same would occur for editing an existing record.

The second requirement raised by the design flowchart was that a reliable way of storing the information would be paramount to the function of the backend. To implement this, we are looking at a few different technologies, of which there are two main classes: JSON and MySQL Databases.

JSON is a notation for storing data that allows for reliable transfer between a backend and a webpage as well as class-based storage of information. This would mean information could be stored in a well-organised, easily transmittable format that allows for grouping of information about identical locations.

MYSQL Databases are a more formal method of storing data such that it can be processed with ease as well as loaded as needed (while JSON can only be loaded all together). A MySQL database is more complex to implement than storage as a JSON file since it cannot innately be sent across a network and would require formatting beforehand. Thus we meet the trade-off between reliability and speed: JSON storage provides a rapid way of delivering data but lacks the precision to access individual records, while MySQL provides reliable and accessible storage but would be more difficult to implement and transmit. In the future, we plan on developing our understanding of both options and possibly a hybrid of the two and utilising the most effective option.