**Technology choices**

This section describes the considerations taken into account during the process of selecting the technologies for this Level 4 project. The final choices for the technologies used were selected on the grounds of these considerations, the constraints of this project and maximum coverage of the requirements by these technologies.

**Considerations taken into account**

Due to the size and the nature of this project, the technologies for it were selected carefully based on the following criteria:

* Ease of development. This criteria is important due to the constraints of the Level 4 project. It was preferred that the selected technologies allow quick and easy implementation and testing.
* Maximum coverage of the requirements. The selected technologies should allow the satisfaction of the requirements in an efficient way and they should allow flexibility for any possible changes.
* Abstraction of other components. Separating different components is a well-recognised software engineering practice and would improve the maintainability of the code.
* Sufficient documentation. Learning or improving skills in the chosen technologies is one important benefit of this project so the presence of clear and detailed documentation is desired.
* Prior experience. Having good knowledge in the technologies used will result in a smaller learning curve and maximum effort could be concentrated in development, testing and improvement of the product.

**Choices**

**The backend**

Based on the criteria listed above, the Python programming language in complement with the Django web framework were selected for implementing the backend component. Django provides a well-developed and easy to use database abstraction layer. Each entity in the database is represented as a Python object and there is no need for writing SQL. All the data needed by Weave can be stored in the form of objects and can be accessed, modified, and deleted easily both programmatically and via an administrator interface. This serves well for both debugging and implementing the required functionality of the application. Abstraction is achieved due to the fact that if a different database component was needed the only change would be to change the Django database backend. Sufficient prior experience coming from studying Python in the first year at the University of Glasgow and the Django framework in the Distributed Information Management 3 module enable better and faster development. Furthermore, both Python and Django are very well-documented and developing web based application using them has turned into an enjoyable activity thanks to the award-nominated beginner’s guide to web development with Python and Django “How to Tango with Django”(reference) available online.

**The web interface**

There are two groups of technologies for the web interface- for the server and the client side.

**Server Side**

The Django web framework has a sufficient support for serving clients’ requests. This is achieved via direct communication with the database. Django allows clear separation between presentation and business logic due to the usage of the model-view-template software design pattern which guarantees better maintainability and readability of the code.(expand+add figure)

**Client side**

An important role in the selection of technologies for the client side of the web interface plays the compatibility with the browsers which are expected to be used in schools, mainly Internet Explorer, Google Chrome and Mozilla Firefox. Typically, pupils in schools are constrained to use the browsers that are installed on the school machine and acceptable appearance and behaviour is crucial.

**HTML5.** This is the markup language chosen for the generation of the web pages. It is preferred to other markup languages due to multiple benefits it provides. Firstly, HTML5 provides an easy access to contents and elements which helps for design and debugging purposes. Secondly, it allows for writing of cleaner code where style and content are separated. Last but not least, HTML5 supports excellent cross-browser compatibility.

**CSS** Cascading Style Sheets (CSS) is used for defining the appearance of the web pages. It enforces separation of concerns between the HTML elements and their presentation. Furthermore, the use of CSS allows the control and flexibility over the appearance of different elements and results in a cleaner HTML code.

**JavaScript.** Being free, open-source and supported by the majority of browsers, this scripting language is used to deal with the interactions of the client and the webpage.

**JQuery** Having the same benefits as JavaScript, JQuery is an excellent solution for simplifying client-side scripting. It is used for defining the behaviour of different components on user interaction with the website.

**Ajax.** Asynchronous JavaScript and XML (AJAX) is used for sending asynchronous requests to the server side avoiding the need for reloading the web page. This reduces the network overhead and the behaviour of the application “feels” closer to a desktop one.

**Bootstrap.** This is the most popular framework for developing responsive design of web applications. WEAVE is intended for school computers and the screen sizes may vary across schools. Furthermore, it is not guaranteed that students will use full screen size at all times of interaction. Due to the nature of WEAVE, students may need to open a different window with more information needed for solving a problem. These needs lead to the responsiveness of the application being crucial.

**Font Awesome.** This is an open source library used to simplify the user interface through the use of familiar icons for visualising possible means of interactions with the webpage.

**HighCharts.** This free library is chosen because it allows the easy creation and control of interactive charts which are needed for the teacher interface of WEAVE.