COS301: Possible Technologies

# Suggestions for technologies

Firstly we advise that the system be completely thin client web based. This enables various platforms and devices to make use of the system from the get go. Using thick clients or platform specific implementations will negatively affect usability.

## Front end

* HTML5 (Fully compliant to latest standards)
* JavaScript and CSS3 for styling
* BootStrap for styling and responsive design (this enables devices of all screen sizes to make use of the system, and improves maintainability significantly).
* Jquery Mobile will probably also help a lot for mobile web development.
* AJAX should be employed to enhance the user experience.
* The AngularJS JavaScript framework can be used to create apps on the various pages of the front end. AngularJS follows the MVC architectural pattern, which we believe is a very good way to design and implement a front end.
* Naturally, most of the technologies mentioned will make use of the latest JQuery library.

## Back end

### Database technology

### As an object-relational database management system we recommend considering PostgreSQL. There are many reasons to recommend postgres, but most importantly it is not difficult to use and:

* + It is ACID compliant
  + It uses Multiversion Concurrency Control (MVCC) to avoid deadlock and allow for many concurrent users which enhances scalability.
  + Uses SQL based queries which most students have experience with.
  + It supports Binary and textual large-object storage, which will enable students to upload larger files to the system, which will then be managed by the database.

### Application Framework

We are looking at the Spring Framework that works as a layer on top of Java EE. Quite a few reasons exist for this, but mainly because Spring:

* + Supports NoSQL databases like PostgreSQL.
  + Employs Model-View-Controller (MVC) architecture.
  + Has a remote access framework that supports SOAP and Cobra.
  + Has Classes for writing unit and integration test.
  + Is not too difficult to learn.

An alternative to Spring, is Django. A comparison can be seen here: <http://vschart.com/compare/spring-framework/vs/django-framework>

One quite significant advantage of Spring is that it supports Domain Driven Design, which works very well with the way the BuzzSpace mini project is being structured.

### Application Server

The applications that will run on the BuzzSpace website have to have some sort of server to run on. One of the best Application servers that integrates well with the Spring framework, is the GlassFish Application server. Read more here: <https://glassfish.java.net/>

### Dependency Management

We recommend the use of apache maven for dependency management. Apache Maven integrates well with the Spring Framework.