

# COMP3001 2012/13 Group Coursework Assignment

James Barnett, Sam Bull, Chris Evans,  
Chris Malton, Patrick Naish, Ryan Tyrrell

January 9, 2013

# 1 Prototype Functionality

The system created was a prototype of a text book selling system for use by university students. The intent is that students, when finishing a module, can put their module text book on the website so that they can sell it to students who will be starting this module after them.

Here is a list of all features implemented in the prototype:

- Sell books;
- Buy books;
- Search books;
- Basket system;
- PayPal checkout;
- Messaging system;
- Registering users;
- Book templates (for selling books that are already known to the system);
- Predictive search.

## 2 Tools and Techniques

The tools used during the development process were the following:

- Editors:
  - Vim
  - Notepad++
  - Sublime Text
- Source Control:
  - Git
- Testing:
  - Google App Engine Local Launcher
  - Google App Engine Runner (described below)

The Google App Engine Runner is a system which updates the App Engine when there is a push to the Git repository. Each branch has its own link to what the final website will look like as the branch currently exists, as well as access to standard out and standard error from the associated Google App Engine instance.

The programming techniques used during the development process were the following:

- Web page templates

## 3 Statistics

### 3.1 Lines of Code Breakdown

Language	Files	Blank	Comments	Code
Python	37	268	410	1633
HTML	32	102	1	667
CSS	2	49	7	246
JavaScript	3	2	5	25
YAML	1	0	0	17
Make	1	1	0	5
Total	76	423	420	2593

### 3.2 External Sources Used

The jQuery framework (<http://jquery.com>) was used for Javascript. jQuery UI (<http://jqueryui.com>) was used for CSS and styling of certain UI elements.

## 4 Design and Implementation

The basic website design was inspired by the appearance of the many popular websites that exist for the selling of books. Wireframes were created for what the main pages would be, along with connections illustrating how these pages were to be connected.

The back end is the Google App Engine Data Store which stores all of the information; Python classes are used as the interface through which the front end interacts with the data store and templates are used to isolate the database and the HTML.

PayPal has been used as the checkout system. However, as this is a prototype system the developer PayPal system was used as opposed to the live one. For testing purposes this can be accessed at “<https://developer.paypal.com>” with the user name “comp3001@lists.cmalton.me.uk” and password “Comp3001!”. Having logged in it will then be possible to checkout using the following details on Payal: user name “comp30\_1357661759\_per@lists.cmalton.me.uk” with password “357661743”. Ignore all quotation marks.

The design which was originally devised was based around the concept of tailoring the system for use in a university environment; it would have been built such that books could be assigned to relevant courses/universities to help users of the system to find books which are appropriate to the courses they are studying at the university they are attending. Unfortunately we had to drop these features due to time constraints.

The following list contains all of the technologies and how they were used:

- Google App Engine Data Store keeps track of users, books, messages between users and sessions;
- Python interfaces are used to interact with the data store for the purposes of retrieving and adding information;
- Templates are used to decouple the database system and the HTML formatting;
- JavaScript is used for predictive search, logging in and listing books;
- Client side validation for forms is performed by HTML5.

## 5 Critical Evaluation

While the basic functionality has been implemented, some of the advanced concepts which were in the initial plan had to be cut due to time constraints. Hence, the final prototype does not fully reflect the intent at the start of the project. This means that the prototype does not necessarily reflect a system for university students but, as stated before, with more time it would easily function as such.

The core functionality that was implemented works well and without error and could, without much difficulty, be ported into a real world scenario. Entries for books are easily created and listed by users; copies of said books can similarly be listed with ease; the checkout system works with our PayPal sandbox and a complete listing of all books a user is selling is stored and visible in their account page.

Beyond the aforementioned features that would uniquely identify this system for use in universities (university details, course and module information, and listings and recommendations based on those), there are other features that could either be expanded or implemented. The messaging system is relatively basic, but could be expanded with the ability to track read and unread messages, and possibly message threads between users; retrieving complete information on a book based on incomplete given information (i.e obtaining the title, author and more from just the ISBN); seller reputation would be a valuable addition; the design of the site could be improved and made more attractive and many other more minor, features.