Project Report

The Ariane 5

The Team

Constatin Dumitrescu
David Geleta
Shane Bruce
Tomasz Oczos
Richard Williams

Contents

People and Roles

The Ariane Five group consisted of five members Shane Bruce, Constatin Dumitrescu, David Geleta, Tomasz Oczos and Richard Williams. All tasks within the requirements stage were carried out as a group in order to give everyone a clear understanding of the system. For the remaining stages of the project tasks were assigned to smaller groups of two or three people.

The following is a breakdown of the major tasks carried out by each group member as well as any additional contributions they made.

Group Member	Major Tasks	Additional Contribution
Shane Bruce	• Requirements elicitation and analysis • Database design and implementation • Website implementation • Collection and entry of sample data	• Artwork design • Documentation preparation and proofing
Constatin Dumitrescu	• Requirements elicitation and analysis • Application Design and Implementation • Website Design and Implementation	• Artwork Design • Came up with the Formula Store idea and provided a lot of useful background information related to the subject
David Geleta	• Requirements elicitation and analysis • Application design and implementation • Acceptance/user testing	• Artwork Design • Designed and produced user guide and on-line help documentation
Tomasz Oczos	• Requirements elicitation and analysis • Database design and implementation • Collection and entry of sample data • Acceptance/user testing	• Translated sample formula into Latex format for entry into database
Richard Williams	• Requirements elicitation and analysis • Database design and implementation • Application implementation • Website implementation • System testing	• Server set-up and maintenance • Project planning and management

Application Overview

The Formula Store is an online database of formulas, algorithms and implementations of these algorithms. Its purpose is to store and manage them with the help of a desktop application. People using the Store can not only search for formulas and algorithms, but also create them and add them to the database. It is designed to work like the Wikipedia website, encouraging collaboration so that the database can grow and more people can use it. Unlike other websites, we are promoting security and correctness when it comes to the formulas/algorithms, which is why people are only allowed to contribute to the database after providing enough qualifications to show they have knowledge in a certain discipline.

The intended audience for the system includes, but is not limited to the academic world. Students, researchers and in the end anyone looking for a formula, or an algorithm would find it useful. The reason they might want to use it is because of the ease of access to information. Instead of searching for a formula or algorithm using Google or some other search engine, and then going through the results, and then through some websites, people can just open this application, or our website, and type in the name of the formula/algorithm and they can find it right away.

The application has three different types of users: regular user, searching for formulae/algorithms; contributor, adding new formulae/algorithms; and the administrators, who manage the database, grant users contributor status and perform database backups. All three types of users can perform typical queries for searching for formulae/algorithms, and in addition the administrators can view different reports for the database, such as system growth, user location, etc.

Project Outcome

The completed Formula Store system has met all the requirements (see requirements specification for full details). The project team created a multi-user, distributed database application complete with fully functional server and both desktop and web-based client applications.

The desktop client is complete providing all required functionality for each user type (Users, Contributors and Administrators). However given the time constraint on the project the web client was scaled back to only provide user-level functionality.

All queries used and reports produced by the system fulfil the specified requirements and the database structure and functionality closely reflects the design specification.

Additional Features

In addition to the required functionality the final system includes some additional features such as the Formula Builder tool, which allowed contributors to build complex formulae using user-extensible toolbars with pre-defined symbols.

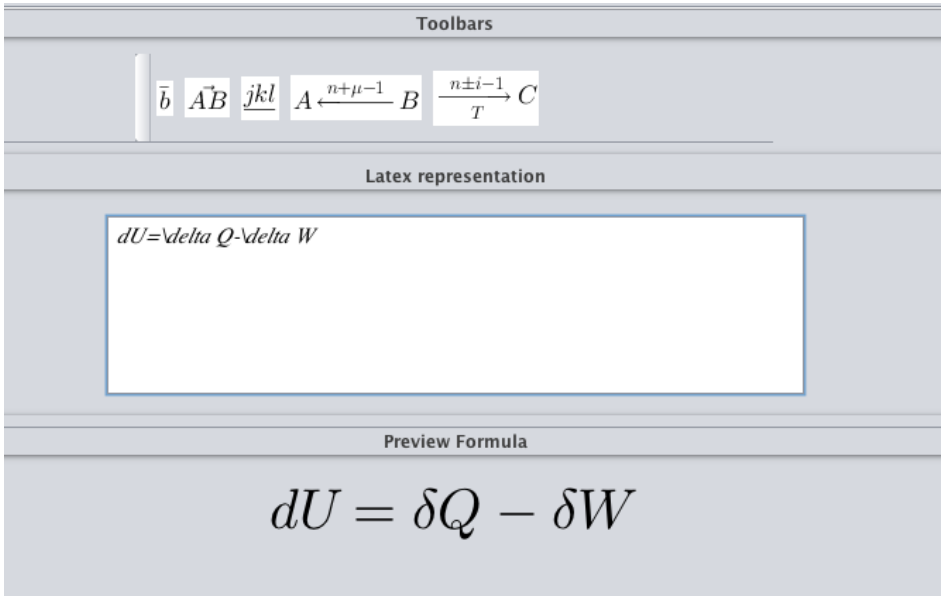

Figure 1 screenshot of the Formula Builder tool

Another addition was the advanced search feature; this allowed users to further define their criteria when searching by providing additional fields. For example when searching for Implementations a user could refine the search by selecting which category to look in or which programming language the implementation should be written in.

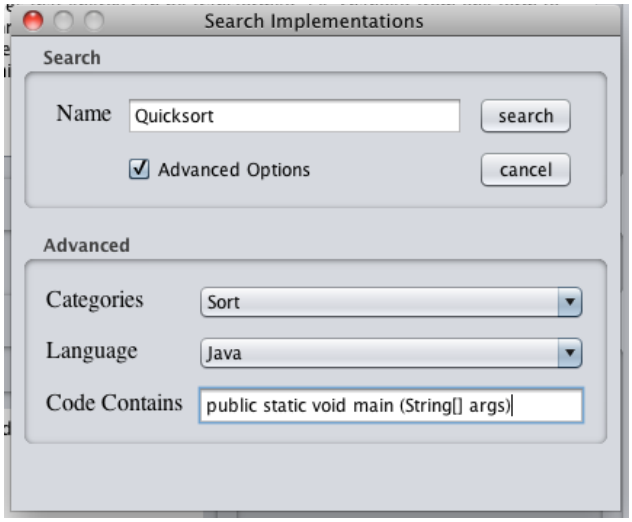

Figure 2 screenshot of the advanced search feature

Another addition involved communicating with users; in order to keep users up to date a number of notification e-mails are sent to their registered e-mail address. These e-mails are sent in order to: confirm registration, notify users

when their requests have been filled or when a user's account has been upgraded.

Project Strengths

Cross platform support

Developing the desktop application using a platform independent language (Java) means the application is not constrained to a single platform; this means it is more accessible to users with a variety of operating systems such as Microsoft Windows, Apple's OSX and Linux. The addition of the web interface further improves the accessibility of the system as it can be accessed on any device with full web browser support (including mobile devices such as tablets and smart-phones).

Accessibility

Both the desktop application and web interface comply with accessibility standards (e.g. W3C standards) to improve the accessibility of the system for users with impaired vision. We made use of tooltips, large fonts and alternate text for images to provide tools such as screen readers (programs which scan web pages and application windows and read the text to users) with useful information to help visually impaired users navigate around our system.

Usability

With a good design and features such as tooltips and standard menus the application user interface is very easy and intuitive to use. The addition of extensive help documentation and an on-line user support facility further improve the usability of the application. This is a key feature of any large application as user satisfaction is a key factor in measuring the success of any system.

Efficiency

A well-designed application that does not perform well would be useless to its users as they would quickly become frustrated and seek alternate means to achieve their goals.

The Formula Store application was designed with efficiency in mind, for example we made use of lightweight, efficient graphical components to ensure the user interface was responsive and did not hog resources.

An additional improvement to aid efficiency of the user interface was to perform computationally complex operations on the application. These included operations such as the population and rendering of graphs.

Robustness and Dependability

The Formula Store system was rigorously tested and performed to specification. Data entry errors are caught where possible and in the event of fatal errors such as connection failure the application will not close unexpectedly. Instead the user is advised of the error and informed of the necessary steps to correct it.

In order to ensure the integrity of data stored within the database daily automated backups are performed and the database state is backed up to a removable storage device. Additionally the administrator can perform a manual backup at any time via the application.

Maintainability

The design of the system is fully documented and the code itself has been well commented. Coding standards such as variable naming conventions and indentation were used to make the code easy to read and understand.

The application was implemented following an object-oriented approach and is therefore very modular. Thus revisions to the specification will require changes to only a small number of modules. Additionally each module has very high cohesion i.e. each module performs a single self-contained function making them easier to understand.

Project Weaknesses

Security

Given the limited time and resources (mainly the knowledge within the team) only the most basic security measures (encrypted passwords, user authentication) have been put in place. Unfortunately this means a determined attacker may be able to gain unauthorised access to the system and thus more stringent security measures would be required to ensure the integrity of the system.

Scalability

Given the very limited budget the scalability of our application is very limited. This is primarily down to the hardware limitations of our server and it's network connection speed. The server is a laptop with limited memory capacity and computation resources. Whereas a more suitable solution would a high-end dedicated server. Additionally the servers network connection is standard home broadband with very limited data transfer speeds.

Even with these limitations the application still performs well within current requirements. However the server will quickly become overloaded as more users start to use the application.

Reliance on User Base

As all content within the Formula Store system is user-generated the quality of the whole user experience is heavily reliant on the actual quality of the content. Thus if the system does not attract a large enough number of qualified users to add formulae, algorithms and implementations to the database the Formula Store will not be a very useful tool.

We have attempted to ensure the quality of the system by requiring users to apply for a Contributor account before they are allowed to add content to the system.

Application Requirements

The desktop application is not a stand-alone application and requires the user to first install the latest version of the Java Runtime Environment. While this is not a major issue it may be an obstacle for certain users with security concerns.

Know Issue

As with all distributed systems the Formula Store is reliant on internet connectivity. Any interruption to the internet connection on the users' computer (caused, for example, by a poor wireless signal) can lead to errors within the application. When this occurs and error message is displayed prompting the user to check their internet connection.

The application has been designed to allow the user to resubmit the request once the connectivity issues have been resolved.

Future Developments

Relations
Information on relations between different formula, algorithms or implementations could be used to improve search results, suggest related formula, algorithms or implementations to users.

This could be achieved using a similar approach to tagging web pages with keywords that reflect the content on the page. For example a sorting algorithm could have tags such as sort, sorting, computer science and logarithmic complexity.

Expand Implementations

To improve the accuracy of implementations the system could be expanded to include more rigorous syntax checking. This could be achieved by creating a module to parse the given code and check for syntax errors depending upon the programming language selected.

An alternate approach would be to integrate the system with existing compilers for all the common programming languages. This would allow users to compile and execute their code allowing them to verify both the syntax and the semantics of the code.

Algorithm Animation

To provide users with a better understanding of how a specific algorithm works the system could be expanded to include an animation feature. This would provide a visual representation of a how the algorithm actually works and make it easier to understand. The feature would be most useful in an academic environment as a learning tool for students.

Completing Website Functionality

To complete the website, the functionality of the two remaining user views (contributor and admin) should be implemented. This includes functions such as adding content, filling requests and viewing reports.

Content Moderator

In order to improve the quality and accuracy of the content within the database an additional user view could be included; that of Content Moderator. These specialised users would receive notifications every time a formula, algorithm or implementation has been created or modified.

They would then be able to review the new content for accuracy and correctness, marking it as verified where all information is correct or either making the necessary changes themselves or referring it back to the contributor (with feedback on the issues).

BCS Code of Conduct – Professional Issues

The BCS code of conduct outlines a guideline to professional standards required in IT. As the group project was designed in order to develop professional skills that members would use in real development environments (amongst other skills) it was important that the project was not taken in a light hearted fashion and industry standards were followed/obeyed where necessary.

Although it was primarily a module project the implementation and the ethical issues surrounding it were taken into serious consideration as if the project were designed for the public domain. It became clear that it was important to reference these guidelines in the design and implementation of the group software project in order to develop an ethically bound and expertly produced system.

The main concerns surrounding our software project were the regard's to the legitimate rights of third parties. This was due to the ability of users to register and therefore submit personal information. The team attempted to comply with these guidelines by invoking the use of a password registration system and the secure storage of user information.

The obligations to data protection under the Data Protection act (1998) were also sighted as we felt that they were relevant to the storage of data within our database system and the transmission of such data from the application across the server architecture.

The BCS code of conduct also communicates the importance of the understanding of relevant legislation, regulations and standards, and that you comply with such requirements. Intellectual property rights were one such legislation that fell into our project domain. In the implementation of our software project we made use of third party libraries, which were open source. To ensure we adhere to the third party open source licence we sighted a list of the libraries we used in the application documentation for the application, which provides links to the libraries (figure1).

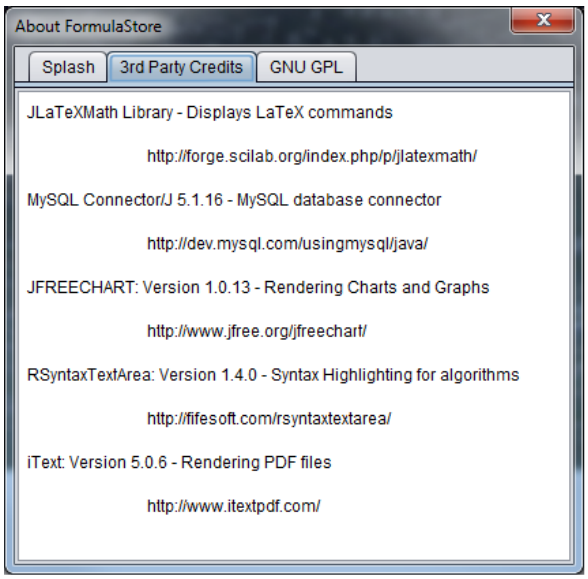

Figure 1. Links to 3rd party libraries used.

Also users of the system provided information other than personal such as algorithms and formulae. These too fell under the law relating to intellectual property. In order to adhere to our professional responsibilities a 'terms and conditions' of use and a 'code of conduct' were produced. These documents outlined the use of third party content and the privacy policy. We made it clear that the formula store does not own any uploaded content and that if users uploaded content which had intellectual property rights it would be removed if complaints arose.

> *"The material and content uploaded by any user is not owned by the formula store, we do not provide any warranty or guarantee as to the accuracy, timeliness, performance, completeness or suitability of the information and materials found or offered on this website for any particular purpose." [1]*

> *"All occurrences of intellectual property theft will be the sole responsibility of the user account to which it was uploaded, all occurrences/complaints*

> *will be investigated and content will be removed accordingly, this may result in user termination"[2]*

The BCS code of conduct lists a number of guidelines, which are related to the supply of expertise to a customer. There were therefore many guidelines, which were not relevant to the production of our project.

---

[1], [2] Bruce, Shane (2011). *Formula Store Terms and Condition*. Available From: http://formulastore.co.uk/help.termsandconditions.php (9th May 2011)

Project Bibliography

Books

Sommerville Ian, (2007). *Software Engineering Eighth Edition*, Addison-Wesley

Connolly T, Begg C, (2004). *Database Solutions Second Edition,* Pearson & Addison-Wesley

Lecture Notes

Bell Paul, (2010). *Software Engineering I,* Available from VITAL: https://vital.liv.ac.uk/ (9th May 2011)

Malcolm Grant, (2010). *Advanced Object Oriented Programming,* Available from: http://www.csc.liv.ac.uk/~grant/Teaching/COMP213/ (9th May 2011)

Websites

LaTeX Project Team, (2010). *LaTeX – A document preparation system*, Available from: http://www.latex-project.org/ (9th May 2011)

Oracle and/or its affiliates, (2011). *MySQL 5.5 Reference Manual,* Available from: http://dev.mysql.com/doc/refman/5.5/en/ (9th May 2011)

Oracle and/or its affiliates, (2011). *Creating a GUI With JFC/Swing,* Available from: http://download.oracle.com/javase/tutorial/uiswing/ (9th May 2011)

Oracle and/or its affiliates, (2011). *Java™ Platform, Standard Edition 6 API Specification,* Available from: http://download.oracle.com/javase/6/docs/api/ (9th May 2011)

Achour M, Betz F et al, (2011). *PHP Manual,* Available from: http://www.php.net/manual/en/index.php (9th May 2011)

W3Schools, (2011). *HTML Tutorial,* Available from: http://www.w3schools.com/html/default.asp (9th May 2011)

BCS, the Chartered Institute for IT, (2011). *BCS Code of Conduct V3,* Available from: http://www.bcs.org/upload/pdf/conduct.pdf (9th May 2011)

Free Software Foundation, (2007). *GNU General Public License,* Available from: http://www.gnu.org/licenses/gpl.html (9th May 2011)

United Kingdom Parliament, (1998). *Data Protection Act,* Available from: http://www.legislation.gov.uk/ukpga/1998/29/contents (9th May 2011)