Australian Council of Deans of ICT

ACDICT Learning & Teaching Academy (ALTA) Learning and Teaching Grant Scheme

Final Report 2013–14

1. Project title

A database for storing and retrieving classified and benchmarked exam questions for ICT courses

2. Lead applicant details

Associate Professor **Margaret Hamilton**
School of Computer Science and Information Technology, RMIT University
Email: margaret.hamilton@rmit.edu.au Phone: (03) 9925 2939

3. Project collaborators

Dr Daryl D'Souza, RMIT University
Associate Professor James Harland, RMIT University
Associate Professor Raymond Lister, UTS
Alireza Ahadi, UTS
Donna Teague, QUT

4. Project outcomes

This project developed a database, called RoW for Repository of Wisdom, to enable ICT Lecturers and Tutors to upload exam questions which they have given their cohort of students in the past. The details of the database have been published at ASWEC – Australasian Software Engineering Conference held in Sydney in April 2014, http://www.aswec2014.org/programme/schedule/.

As well as exam questions, Lecturers can also upload performance data of their students on this exam question. This performance data can be for various cohorts, postgraduate or undergraduate, home campus or country or international campus. The data can be in any format, .txt, .xls, .pdf.

Lecturers can compose exams based on questions which have been stored in the database. These are stored as various subsets, and can be retrieved and amended as required. The retrieval of particular exam questions can be based on topic, content, level of course, benchmarked results. There is a search bar, as well as one screen displaying all the attributes. After ASWEC, there were recommendations to include a voting system for questions, so comments and a 5-star rating system have been incorporated. Lecturers are also encouraged to incorporate keywords, such as "rainfall" and the questions can be searched by these keywords, tags and attributes.

This new database extends the work from the BABELnot project (2011-2013) led by A/Prof Raymond Lister, by creating the database to store and share standardized assessments and benchmarked results, as well as incorporating the capability to critically evaluate the questions.

The database is hosted on the Nectar cloud, and maintained by RMIT.

The database can be accessed at: http://115.146.93.78/login.php

5. Project deliverables and budget report.

All academics from around Australia can apply for an account with Nectar and use this to access the database.

Alireza Ahadi designed the database backend required for this system, costing $7000.

At the same time, RMIT employed a summer student to develop the front end, costing $4000.

As part of her PhD, Donna Teague has been designing and testing questions for novice programmers to identify which stages of learning they are currently demonstrating. She was paid $2000 to provide these questions in a suitable format for the database.

The team met face-to-face in Melbourne at the beginning of the project, and met over Skype for the duration of the coding of the system.

The remaining funds were spent to disseminate the project at ASWEC 2014 in April and at ALTA in June, 2014.

6. **Paper delivered at ASWEC, explaining database which is now known as the Repository of Wisdom**

Repository of Wisdom

A database for storing and retrieving classified and benchmarked exam questions for introductory programming courses

Abstract

There is a widespread belief among instructors of programming that summative assessments are valid instruments to test student understanding of programming. Moreover, despite this belief, instructors set exams largely on the basis of intuition, experience and wisdom, and not on the basis of universally accepted theories. This paper presents an attempt to bring a community of ICT academics together to contribute to a repository of questions suited to programming exams, using previously established theories of question difficulty.

Specifically, this paper extends previous work conducted around exam classification (in the BABELnot project) with the aim of engaging the ICT (academic) community in discussions around developing better educational outcomes in standards for setting programming exams. It is about the development of a database in which ICT academics contribute and classify questions that they believe might be useful or interesting to assess their novice programming students. They would also be encouraged to upload exam questions which they have given their cohort of students in the past as well as the performance data of their students on these exam questions. This performance data could be for various cohorts, postgraduate or undergraduate, home campus or country or international campus.

Moreover, while the emphasis here is on building a repository of exam questions, the database may be used to track interesting formative assessment questions in the same way.

Keywords

Software Engineering education; assessment; Computer programming exam questions: style and characteristics.

INTRODUCTION

The aim of the research explained in this paper is to enable lecturers to compose exams based on questions which have been stored in our database, which we have termed, the repository of wisdom. The retrieval of these exam questions could be based on topic, content, level of course, benchmarked results, with interesting and innovative retrieval options related to ranked queries.

The rationale for such a project is to encourage a more rigorous consideration of examination (and other) questions. All too often we hear of high failure rates in ICT courses, particularly introductory programming. It is our belief that some of this can be attributed to “tricky”

exam questions which have been intended to challenge the higher achieving students but in doing so have provided a roadblock for the less able students.

We also aim to contribute to the benchmarking of best practices in novice computer programming across the ICT sector, as, once developed, we aim to make this database available to all academics. This project developed from the BABELnot project, which was funded by an Australian Learning and Teaching Council (ALTC) grant. One aim of BABELnot was to develop a rich framework for describing the learning goals associated with programming [1]. The next step was to map syllabus content to this framework, using a system called ProGoSs [2]. One interesting direction of this work was to map exam questions onto this framework [3] and to try to explain what a "bare-pass" student might be able to achieve. The outcomes of the exam classification project have been published in various forums, including a workshop in ACE 2012 [4], and represent an attempt to standardise summative assessment across the ICT discipline, both nationally and internationally.

This new repository research project (which we call the Repository of Wisdom) extends the work from the BABELnot project by creating the database to store and share these standardized assessments and benchmarked results. In this paper we explain the background to our research which lead to the exam classification system we have employed (Section II). In the following section we explain the construction of our repository (Section III), and how to enter and retrieve exam questions (Section IV). We discuss some issues remaining with our work in Section V, and present our conclusions in Section VI.

I. BACKGROUND

Shuhidan et al [5] developed a survey of programming instructors from all over the world and found that "most of the instructors believed that summative assessment is, and is meant to be, a valid measure of a student's ability to program". Many instructors are required to set exam papers as part of their employment, and do not question whether this is the best way to assess their students. The time allocation and submission requirements differ, but many assessors have little training in how to assess their students and follow templates from previous runs of the course. The majority of instructors believed that multiple choice questions were an appropriate way to assess and a significant percentage of these believed they could provide a very good and accurate assessment of the student's programming skills [6]. A repository consisting of purely multiple choice questions has recently been developed as a result of work undertaken at ITiCSE workshop [7].

Efforts to deliver a shared understanding of programming competency have been published by the BABELnot researchers. This was built on earlier BRACElet research projects which investigated exam questions for novice programming students [8]. The BABELnot project documented academic standards associated with the sequence of three programming courses or units found in ICT degrees in Australian universities. The overall goal of this project was to develop a shared understanding of competency in programming so as to arrive at a unified view of, among other things, summative assessment instruments. Two subgoals included the development of a rich framework for describing the learning goals associated with programming and to benchmark exam questions mapped onto this framework.

This framework involved a purpose-built exam question complexity classification scheme to provide an assessment of exam question difficulty [9, 10, 11]. Such a framework is necessary because there is little evidence of the use of learning theories in the writing of exam papers for programming courses [12]. This framework also provides a standard method for understanding questions used in programming exams, and our Repository of Wisdom is based on this same classification system.

II. BUILDING THE SYSTEM

We aimed to design, create, implement and test a database to store exam questions with benchmarked student results. The architecture of such a database needed to include capabilities for the uploading, storing and retrieval of the question, the marking guide, the results of students from various cohorts and other relevant information such as the course where it was delivered, the programming language (if any) required, and the level of the exam (whether for introductory programming novices or advanced expert students).

Through the BRACElet and BABELnot projects, we have already identified many suitable exam questions which have been tested on some cohorts of students in Australia and New Zealand. With permission of the writers, we aim to upload these questions into the database and make them available to other universities who teach similar ICT courses.

Other academics teaching programming who may be interested to contribute or avail themselves of the opportunity to use the questions in the database, or possibly to provide feedback on some of the questions. If they do use the questions we will ask for evidence of their results to extend and further refine our benchmarking inside this database.

We have developed the Repository of Wisdom (ROW) based around the characteristics of the exam question framework published in an earlier paper [10] and summarised below.

A. Questions

Each question is classified into the following properties:

- Language
- Topics
- Style of question: multiple choice, short answer, program code, Parsons problem, graphical representation
- Course level: level 1, level 2, level 3
- Percentage mark: out of 100
- Required skill: skill required to solve the question
- Open / closed: closed if the question only has 1 possible answer; open if the answer can be expressed in many ways
- Difficulty level
- Explicitness level
- Linguistic complexity
- Conceptual complexity

- Intellectual complexity: Bloom's taxonomy
- Code length
- External domain references: refers to a domain beyond what would necessarily be taught in the programming course.

B. Style of Question

Although the classification system refers to it as "style of question", it could more commonly be called "type of question" and really describes the type of student response expected. Our repository stores question styles of:

- multiple choice
- short answer
- program code
- Parson's problem (provides lines of code which have been swapped around and require the student to put them in the correct order),
- graphical representation (eg concept, flowchart, class diagram, data structure)
- report (brief text but not as short as a short answer) and
- essay (which could be considered as text longer than a page).

C. Skill required to answer the question

Skills may refer to abilities required to be demonstrated in the answer, such as writing programming code, tracing code, testing and/or debugging code. We have tried to organize these in a manner often associated with the revised Bloom's taxonomy and/or the Solo taxonomy [13, 14].

- pure knowledge recall
- trace code (includes evaluating expressions)
- explain code
- write code
- modify code (includes rewriting, refactoring)
- debug code
- design program (without coding; if a question involves both design and coding, we will classify it as write code)
- test program.

III. USING THE DATABASE

As part of our teaching, we have, over the years, been designing and testing questions for novice programmers to identify which stages of learning they are currently demonstrating. We aim to identify which questions are suitable to be stored in the database for dissemination to other institutions.

A. Access to the repository

This repository is hosted on the NECTAR cloud [15]. Access to this is free for all academics employed in Universities in Australia. The Shibboleth login enables access to our repository.

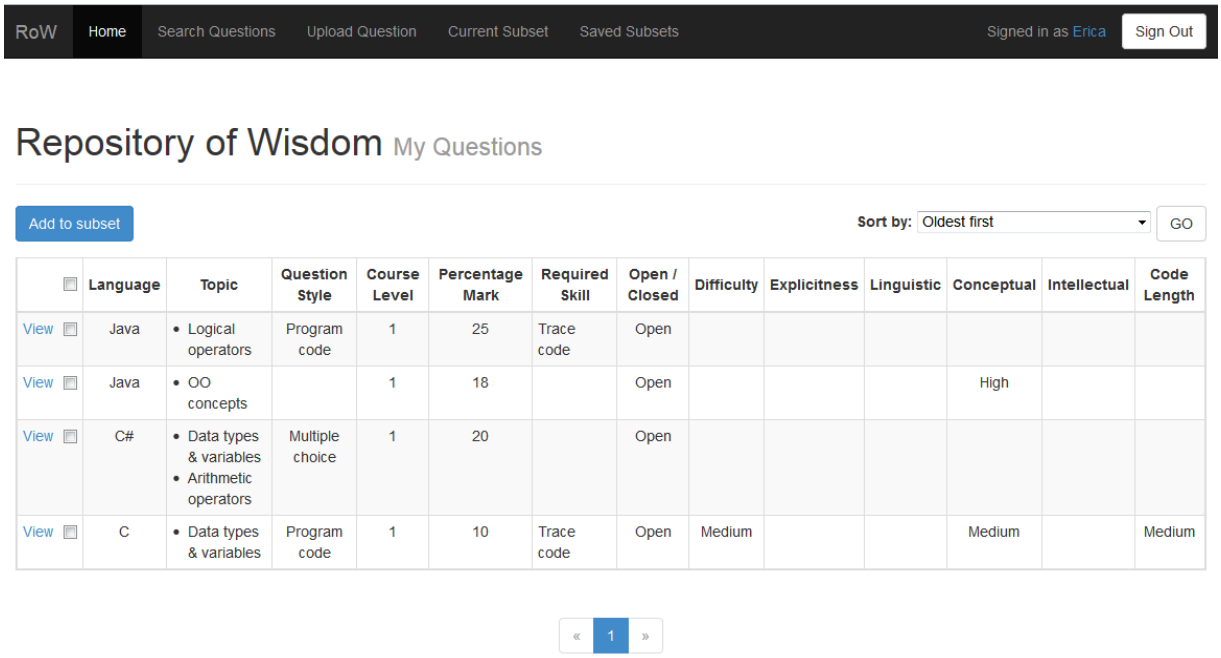

Figure 1: My Questions (Homepage)

Questions can be sorted by various criteria. The implemented criteria are:

- Date created
- Percentage mark
- Course level
- Difficulty.

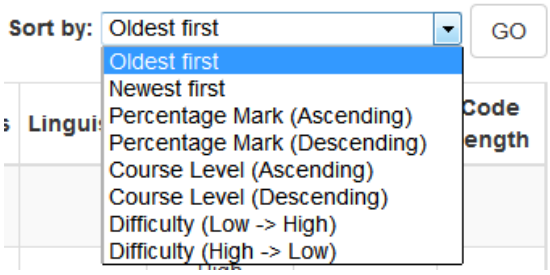

Figure 2: Sorting questions within the repository.

The repository can be searched for questions by very many different criteria, including language, topic, course level, style and skill levels among all the criteria discussed in Section III.

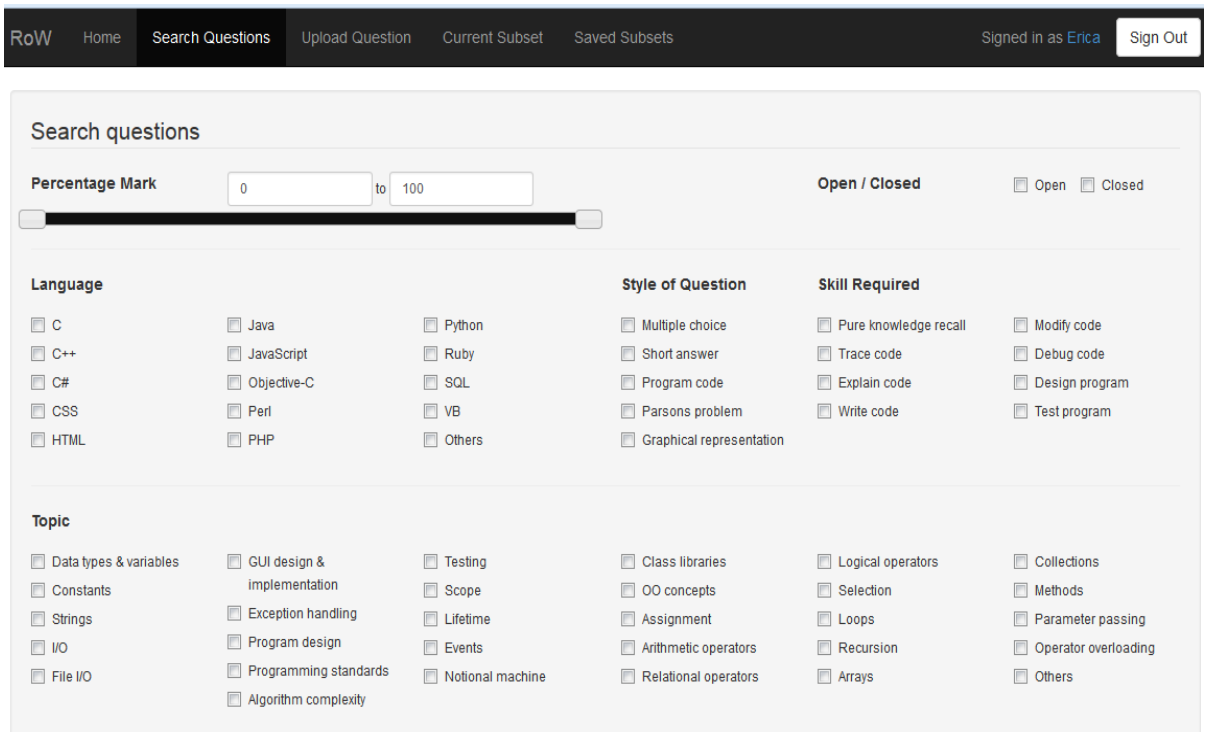

Figure 3: Searching for questions within the repository

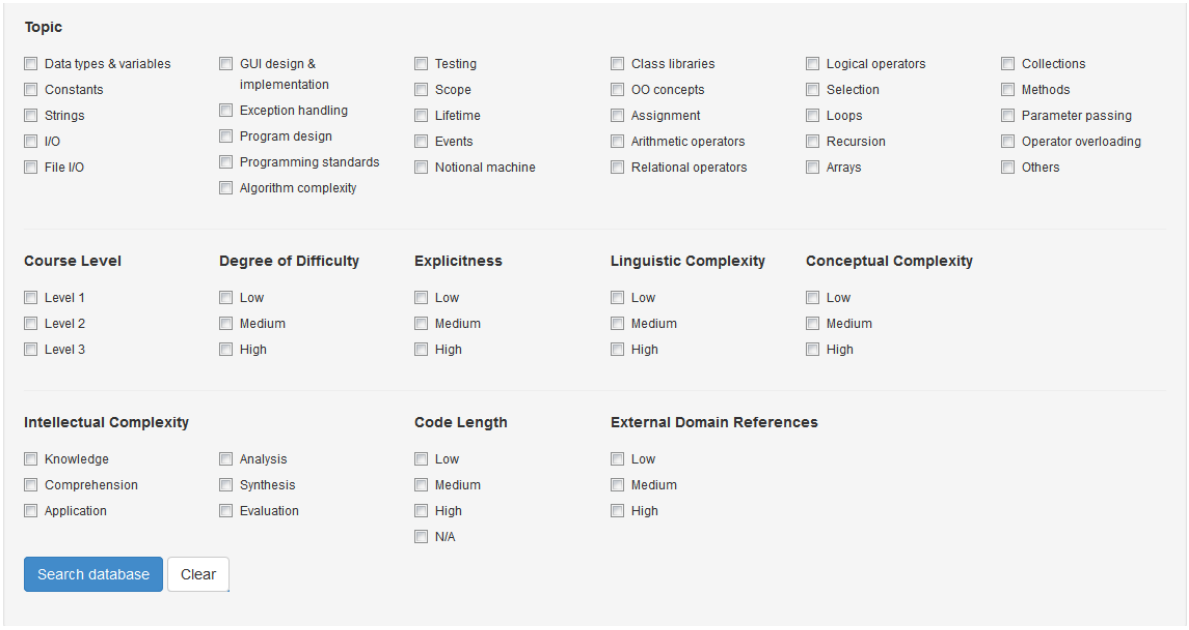

Figure 4: Search questions form (second part)

A subset of questions can be selected, and stored on an exam form and saved for future use, and later retrieved.

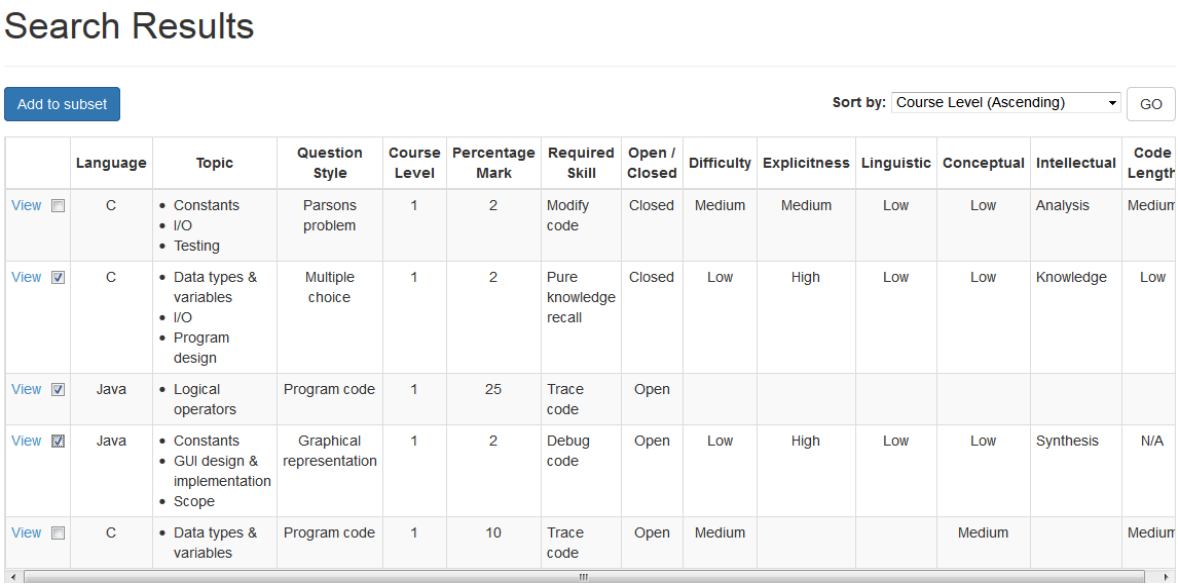

Figure 5: Search results with some of the questions selected

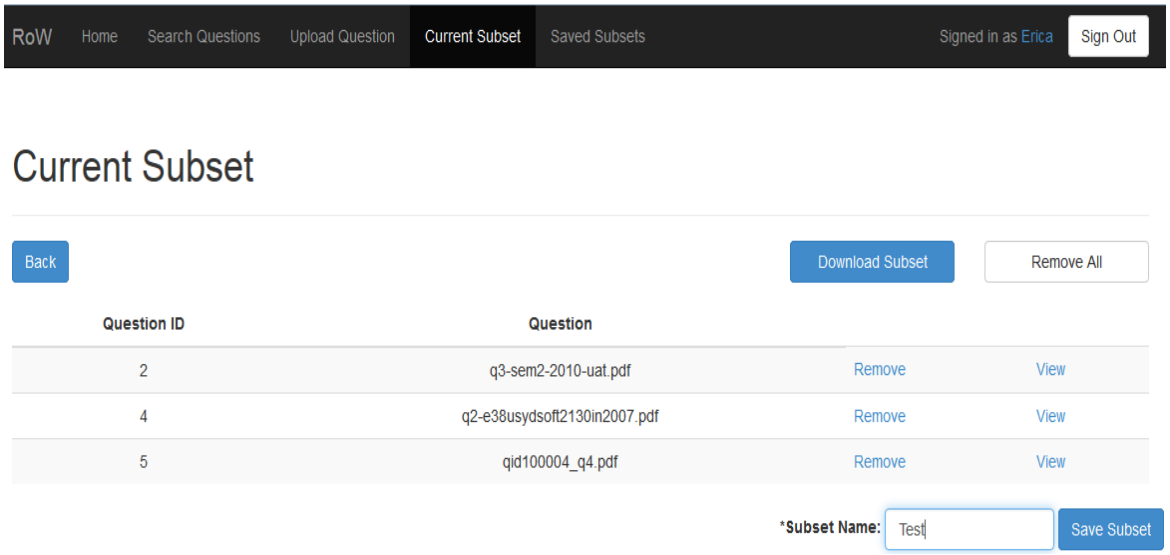

Figure 6: Saving current subset of questions

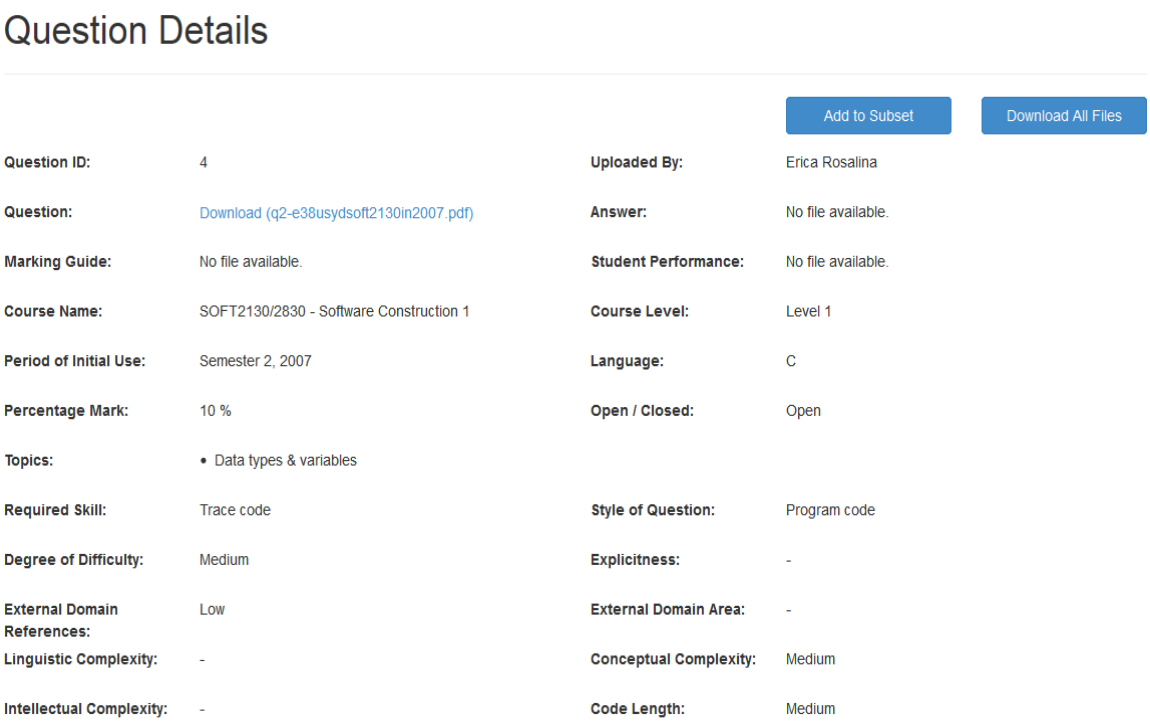

Figure 7: Show detailed information of a question

Questions can be viewed individually, or together as an exam paper in a standard tabular format which would need to be massaged into any required template. They can be downloaded separately, or together and stored with an answer file, marking guide, and student performance data if available and if required.

IV. INTERESTING FEATURES AND ISSUES WITH THE REPOSITORY

As with Facebook and social media-types of repositories, we are implementing capabilities to write comments and rate questions within the repository. So far, this is untested or trialled, but is merely encouraged. We are not sure how academics might feel about their "favourite" exam questions being rated as poor, or their "easy" questions being pulled apart and explained as difficult or obscure.

We have fixed an arbitrary maximum file upload size as being 20MB for a file. When uploading a new question, users can upload 4 files (question file, answer file, marking guide and student performance data). If each of these files is 20MB (total of 80MB), this will be fine. As long as none of the files exceed 20MB, the system will still allow the uploading.

Some parts of the website interface might not render properly in some browsers (e.g. older versions of Internet Explorer 10). It is advised to use browsers other than Internet Explorer.

Disabling JavaScript might cause some performance issues and some parts might not work properly since a thorough testing on this has not been done yet.

Anyone can register to the website for access to the repository. Since the main purpose of this repository is to store exam questions, students should not be able to access it. This may be difficult to implement but the ideal solution for this problem may be to have another form of authentication mechanism, similar to the login system used in NeCTAR (https://dashboard.rc.nectar.org.au/), to only allow lecturers to register.

The system accepts any format for the file uploads (since it is not clear yet what the file format requirement might be). In the future this may be limited to secure the system by restricting the file formats to only those commonly used. In additional to that, an antivirus program may be installed on the server to make it even secure.

V. CONCLUSIONS

This is a work in progress and, to date, we have only stored information about our own exam questions. However, we are aiming to complete the coding and testing of the Repository of Wisdom by June 2014, after which we plan to advertise it to Universities who deliver degree programs in Computer Science and Software Engineering around Australia.

However, in the meantime, the authors are happy to make this repository available to academics who email them and request access for testing purposes, or who have suggestions for particular types/styles of questions and benchmarking information they might like to share and access.

REFERENCES

[1] Lister, R., Corney, M., Curran, J., D'Souza, D., Fidge, C., Gluga, R., Hamilton, M., Harland, J., Hogan, J., Kay, J., Murphy, T., Roggenkamp, M., Sheard, J., Simon and Teague, D. (2012): Toward a shared understanding of competency in programming: An invitation to the BABELnot project. In proceedings of the 14th Australasian Computing Education Conference (ACE2012), Melbourne, Australia.

[2] Gluga, R., Kay, J., Lister, R.F., Simon, and Kleitman, S. (2013), 'Mastering Cognitive Development Theory in Computer Science Education', Computer Science Education, vol. 23, no. 1, pp. 24-57 "ProGoSs", https://progoss.com/user/view-login-page/, last accessed: 24:02:14.

[3] Gluga, R., Kay, J., Lister, R., (2012) "ProGoSs: Mastering the Curriculum", Proceedings of the Australian Conference on Science and Mathematics Education, (ACSME2012), Sydney, Australia, September 2012 in Australian Conference on Science and Mathematics Education (ACSME2012), ed M. Sharma and A. Yeung, UniServe Science, Sydney, Australia, pp.92-98 http://ojs-prod.library.usyd.edu.au/index.php/IISME /article/ view/5914 last accessed: 13.12.13.

[4] Simon, Sheard, J., Carbone, A., Chinn, D., Laakso, M.-J., Clear, T., de Raadt, M., D'Souza, D., Lister, R., Philpott, A., Skene, J. and Warburton, G. (2012), "Introductory programming: Examining the exams". In proceedings of the 14th Australasian Computing Education Conference (ACE2012), Melbourne, Australia.

[5] Shuhidan, S., Hamilton, M., D'Souza, D., (2010), "Instructor Perspectives of Multiple Choice Questions in Summative Assessment for Novice Programmers", Computer Science Education, 20(2):220-259.

[6] Shuhaida Shuhidan, Margaret Hamilton, Daryl D'Souza, (2009), "A Study of Novice Programmer Responses in Summative Assessment" Australasian Computing Education Conference, ACE2009, Wellington NZ, January, 2009, CRPIT, Volume 95.

[7] Lister, R. and Sanders, K. (2013), "One Thousand and Twenty Four Multiple Choice Questions for CS1/CS2". ICER 2013, http://www-staff.it.uts.edu.au/~raymond/OneThousandAndTwentyFourMultipleChoiceQuestions.pdf, last accessed: 24:02:14.

[8] Whalley, J., Clear, T. & Lister, R.F. (2007), "The Many Ways of the BRACElet Project", Bulletin of Applied Computing and Information Technology, vol. 5, no. 1, pp. 1-16.

[9] Simon, Sheard, J., Carbone, A., D'Souza, D., Harland, J. and Laakso, M.-J. (2012), "Can computing academics assess the difficulty of programming examination questions?", in proceedings of the 11th Koli Calling International Conference on Computing Education Research, Finland.

[10] Sheard, J., Simon, Carbone, A., Chinn, D., Clear, T., Corney, M., D'Souza, D., Fenwick, J., Harland, J., Laakso, M. and Teague, D. (2013), "How difficult are exams? A framework for assessing the complexity of introductory programming exams", in proceedings of the 15th Australasian Computing Education Conference (ACE2013), Adelaide, Australia.

[11] Sheard, J., Simon, Carbone, A., D'Souza, D. and Hamilton, M. (2013), "Assessment of programming: Pedagogical foundations of exams", in proceedings of the Innovation and Technology in Computer Science Education, ITiCSE 2013, Canterbury, UK.

[12] Simon, Sheard, J., Carbone, A., Chinn, D., Laalso, M., Clear, T., de Raadt, M., D'Souza, D., Lister, R., Philpott, A., Skene, J. and Warburton, G. (2012), "Introductory programming: examining the exams", in Fourteenth Australasian Computing Education Conference (ACE2012), pp.61-70, http://www.crpit.com/confpapers/CRPITV123 Simon.pdf, last accessed: 24:02:14.

[13] Shuhidan, S., Hamilton, M., D'Souza, D, (2009), "A taxonomic study of novice programming summative assessment", Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95, Pages 147-156.

[14] Shuhaida Shuhidan, Margaret Hamilton, Daryl D'Souza, (2011), "Understanding Novice Programmer Difficulties via Guided Learning" in proceedings of the 16th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2011) Darmstadt, Germany, June 27-29.

[15] Nectar research cloud: http://nectar.org.au/about-nectar, last accessed: 24:02:14.