THE UNIVERSITY OF NEW SOUTH WALES
SCHOOL OF COMPUTER SCIENCE & ENGINEERING

A New Handbook for UNSW

Robert White Campbell

(Student ID: z3332140)

Thesis Submitted to satisfy requirements of
Bachelor of Software Engineering

Supervisor: John Shepherd
Assessor: Ashesh Mahidadia

Submitted: October 29, 2013

Contents

List of Figures

List of Tables

Abstract

This project was to develop a prototype for a replacement for the current online Handbook [7]. I did so through a PHP-based website connected to the AIMS PostgreSQL database. This report includes information on the existing Handbook, comparable websites, the AIMS database and development of the new website.

The new website is available at `rwca915.srvr.cse.unsw.edu.au`.

Functionality includes searching, browsing, and viewing specific entries. It is also possible to edit the display of a particular Streams page as a proof-of-concept for the External Service, which would allow for external sites to draw upon Handbook information without needing access to AIMS.

Chapter 1

Introduction

The online UNSW Handbook allows students to browse and search program, course and stream information.

A program is a structured program of study leading to the award of a degree, diploma or certificate, and includes one or more streams plus additional rules. A stream is a sequence of study, a list of courses used as the basis for a program. A course is a semester-long class on a topic, and may include lectures, tutorials or laboratories. Courses may be compulsory (core) or selected (elective) to fill the requirements of a stream or program. [7]

The Handbook is primarily used by current and future students looking for enrolment information. When reviewing programs, current students are particularly interested in graduation requirements and course schedules, while future students are interested in descriptions and enrolment requirements. When reviewing courses, students are generally interested in what the course is about, whether or not it is available this semester, and locating the associated School site. The current Handbook does not adequately meet all these needs.

The online Handbook was released in 2005, using static pages to replace the separate books (one per Faculty) that had formerly contained this information. The online Handbook is significantly easier to use than the Faculty books, which had a strict hierarchy and required using multiple books to handle enrolment in a single program, but information is nevertheless occasionally duplicated, contradicted or split across School sites.

The duplication allows School sites to present information in different formats, or provide additional information that is not in the Handbook, but suffers the normal duplication problems of additional work and potential inconsistencies. This opens up the field for a solution that minimizes duplication while still allowing schools to present Handbook information in a form more appropriate to their needs.

The recently developed AIMS (Academic Information Management System) database stores information about a variety of objects, the most relevant being UNSW programs, streams and courses. From this we can construct webpages and a new online UNSW Handbook.

To support School sites and other alternatives, we plan to support the dynamic generation of simple HTML descriptions that contain relevant information presented in a desired format. These can

be inserted into other pages through the use of inline frames, allowing the reproduction of information without duplication of data.

In Chapter 2, I examine in detail the existing Handbook and its flaws. I compare it to similar sites for other Universities, and present and justify an evaluation framework and subsequent evaluation of the existing Handbook.

In Chapter 3, I present my proposed solution to these problems.

In Chapter 4, I discuss what I learned about the AIMS database during my work with it.

In Chapter 5, I discuss my solution and the development and evaluation thereof.

Chapter 2

Background

2.1 The Problem

The UNSW Handbook is intended to provide definitive course and program information for student enrolment. Thus, it should accurately provide all relevant information without contradiction with other School sites. For the purposes of maintenance, data should not be unnecessarily duplicated.

This is not the case.

As discussed in the Introduction, information is currently duplicated, split or contradicted across external sites, particularly School sites. This has both positive and negative effects, and so a solution should aim to preserve the positives while reducing or eliminating the negatives.

Benefits include being able to present information in a manner chosen by the School, rather than according to the standard Handbook format, and being able to provide additional supplementary information that may not be included in the Handbook itself. This includes things like recommended course schedules for programs.

Negatives include the general problems resulting from duplication: it is more work to update an entry, with an increased chance of inconsistencies when doing so.

For example, the Civil Engineering program 3620 demonstrates duplication of information, supplementary information, and minor inconsistencies in figures 2.1 and 2.2 [7]. The school page provides the same information as is provided on the program page, but it is scheduled to make it more useful. However, the school page implies that GMAT1110 and MATS1101 are requirements rather than recommendations.

The Handbook page for Software Engineering 3648 (figure 2.3 [7]) contains little information of its own, preferring to redirect to outside pages. This means the Handbook is no longer a definitive source of information, and students have to refer to additional pages to get a complete understanding of program requirements.

Yuen Wing Kei performed a survey to find out what students thought were the current problems with the Handbook and what people would want from a new Handbook, as part of his 2012 thesis [4]. The raw results can be found in Appendix A of Yuen's report. As there have been no changes since, this information can be assumed to be up-to-date.

The survey respondents were weighted towards (but not completely composed of) CSE students, which should be kept in mind. It also consisted solely of already enrolled students, not having surveyed any high school students looking at enrolling at UNSW.

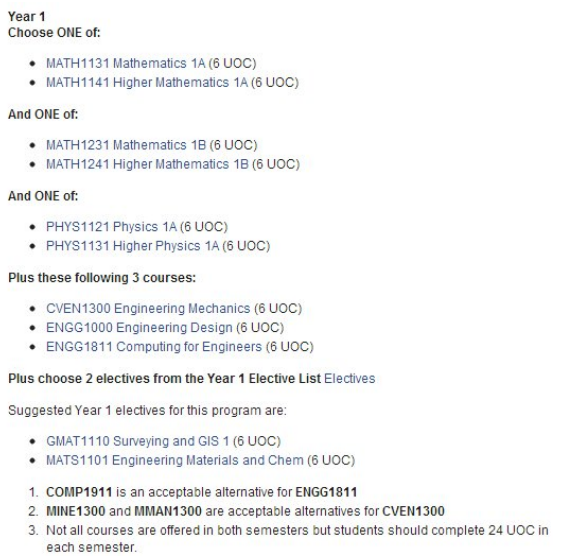
In general, students using the Handbook for course selection particularly wanted to know the requirements of their program, course availability and timetabling, the course outline, and whether or not the course follows on from those they have already taken. They were least interested in student surveys and comments, although even there only 14% and 13% respectively said they were 'not relevant to my decision'.

A specific list of what information students expected from the Handbook pages is provided in the Evaluation section, where it is more easily referenced. This was constructed both from quantitative and qualitative responses.

Otherwise, students thought that the following could be improved:[4]

- Information needs to be more centralised, rather than split across multiple pages (e.g. figure 2.3)
- The search box is confusing (there are currently two separate searches)
- It should be easier to distinguish available courses
- There isn't enough information on double degrees and combined degrees
- It should be easier to distinguish outdated versions of the handbook from the current one
- Handbook descriptions can be improved

Not all of these are directly addressed within the scope of my project. Chapter 3 discusses the proposed scope of my project and Chapter 5 discusses my final solution.



Year 1
Choose ONE of:

- MATH1131 Mathematics 1A (6 UOC)
- MATH1141 Higher Mathematics 1A (6 UOC)

And ONE of:

- MATH1231 Mathematics 1B (6 UOC)
- MATH1241 Higher Mathematics 1B (6 UOC)

And ONE of:

- PHYS1121 Physics 1A (6 UOC)
- PHYS1131 Higher Physics 1A (6 UOC)

Plus these following 3 courses:

- CVEN1300 Engineering Mechanics (6 UOC)
- ENGG1000 Engineering Design (6 UOC)
- ENGG1811 Computing for Engineers (6 UOC)

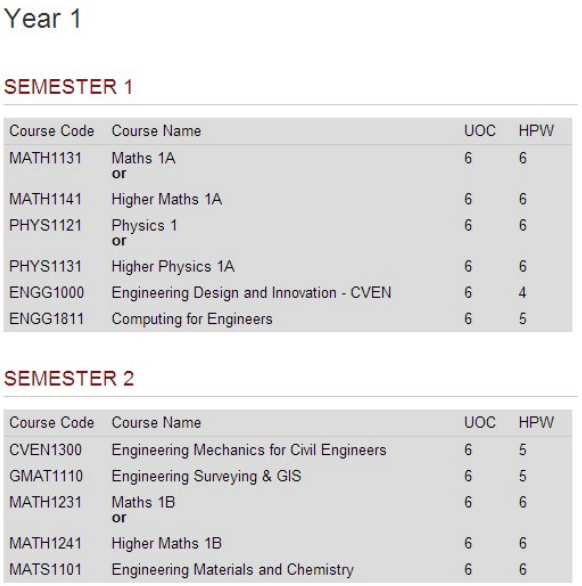
Plus choose 2 electives from the Year 1 Elective List Electives

Suggested Year 1 electives for this program are:

- GMAT1110 Surveying and GIS 1 (6 UOC)
- MATS1101 Engineering Materials and Chem (6 UOC)

1. **COMP1911** is an acceptable alternative for **ENGG1811**
2. **MINE1300** and **MMAN1300** are acceptable alternatives for **CVEN1300**
3. Not all courses are offered in both semesters but students should complete 24 UOC in each semester.

Figure 2.1: The Handbook page for program 3620 Civil Engineering



Year 1

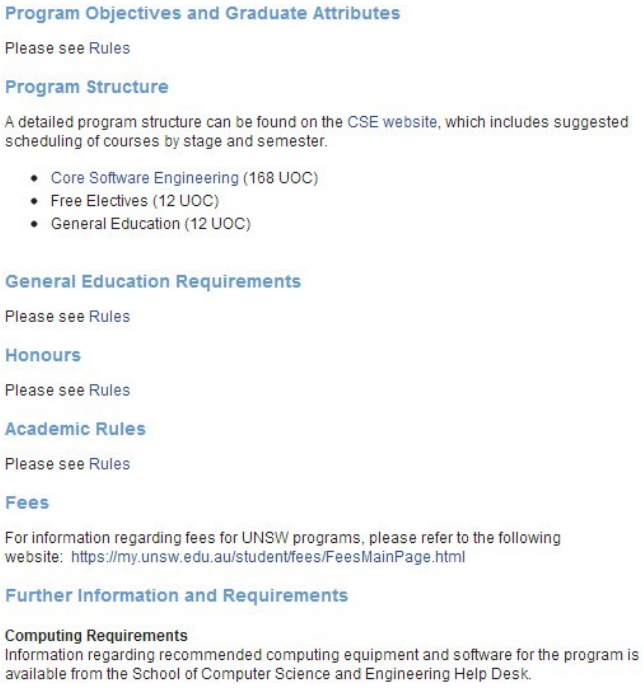
SEMESTER 1

Course Code	Course Name	UOC	HPW
MATH1131	Maths 1A **or**	6	6
MATH1141	Higher Maths 1A	6	6
PHYS1121	Physics 1 **or**	6	6
PHYS1131	Higher Physics 1A	6	6
ENGG1000	Engineering Design and Innovation - CVEN	6	4
ENGG1811	Computing for Engineers	6	5

SEMESTER 2

Course Code	Course Name	UOC	HPW
CVEN1300	Engineering Mechanics for Civil Engineers	6	5
GMAT1110	Engineering Surveying & GIS	6	5
MATH1231	Maths 1B **or**	6	6
MATH1241	Higher Maths 1B	6	6
MATS1101	Engineering Materials and Chemistry	6	6

Figure 2.2: The School page for program 3620 Civil Engineering



Program Objectives and Graduate Attributes

Please see Rules

Program Structure

A detailed program structure can be found on the CSE website, which includes suggested scheduling of courses by stage and semester.

- Core Software Engineering (168 UOC)
- Free Electives (12 UOC)
- General Education (12 UOC)

General Education Requirements

Please see Rules

Honours

Please see Rules

Academic Rules

Please see Rules

Fees

For information regarding fees for UNSW programs, please refer to the following website: https://my.unsw.edu.au/student/fees/FeesMainPage.html

Further Information and Requirements

Computing Requirements
Information regarding recommended computing equipment and software for the program is available from the School of Computer Science and Engineering Help Desk.

Figure 2.3: The Handbook page for program 3648 Software Engineering

2.2 Literature Review

2.2.1 UNSW Handbook

MAPPS/AIMS

The MAPPS database was a course proposal system developed several years ago, storing more information than is currently presented in the Handbook. As it was intended to help handle course proposals through an online system, it aimed to collect a complete set of information about each course. This data was then intended to be made available for multiple purposes, such as building the Handbook or generating Course Outlines. AIMS, the successor to MAPPS, was developed as a system-of-record for all academic information at UNSW.

The Academic Information Management System (AIMS) is a database intended to provide services. It tracks a large amount of data relating to programs, courses and streams, including program rules, course outlines, etc. It contains more information than the MAPPS database does, and is based around records, not proposals.

It is currently in development, and the database is not fully populated. However it is possible to design and build a system around it, since the database schema is essentially complete. This is the database which the new Handbook will be based on.

More information on what I have discovered while working with aims is available in Chapter 4.

Yuen's Handbook

In 2012, Yuen Wing Kei researched and developed a new Handbook based on the MAPPS database.[4] Yuen's project was strictly creating the handbook from the database, and did not include inline frame generation.

Yuen's prototype is viewable at mahler.cse.unsw.edu.au/handbook [3]

I have re-used Yuen's research and surveys on the existing online handbook where applicable. I have referred to his surveys already, and will continue to do so.

2.2.2 Other Universities

Included here are references to how other universities manage similar services. Screenshots are included, and should be referred to for the interface specifics.

Where other universities use 'course' instead of 'program' and 'unit' instead of 'course', this report strictly uses the UNSW terms 'program' and 'course' to maintain internal consistency. When viewing these sites outside this report, keep in mind that these terms may be used differently.

Open University [1]

The Open University website (See Figures 2.4 and 2.5) has a 'study cart', which allows you to mark courses for enrolment while browsing.

Program pages list course requirements and availability and allows adding them to your 'study cart'. Course pages provide lots of relevant information, including textbooks and assessments.

Queensland University of Technology [8]

Specifically relevant and useful: when viewing a program's courses, you can expand a course to view more information on it. This uses AJAX to prevent preloading all of a program's courses. See Figures 2.6, 2.7, 2.8.

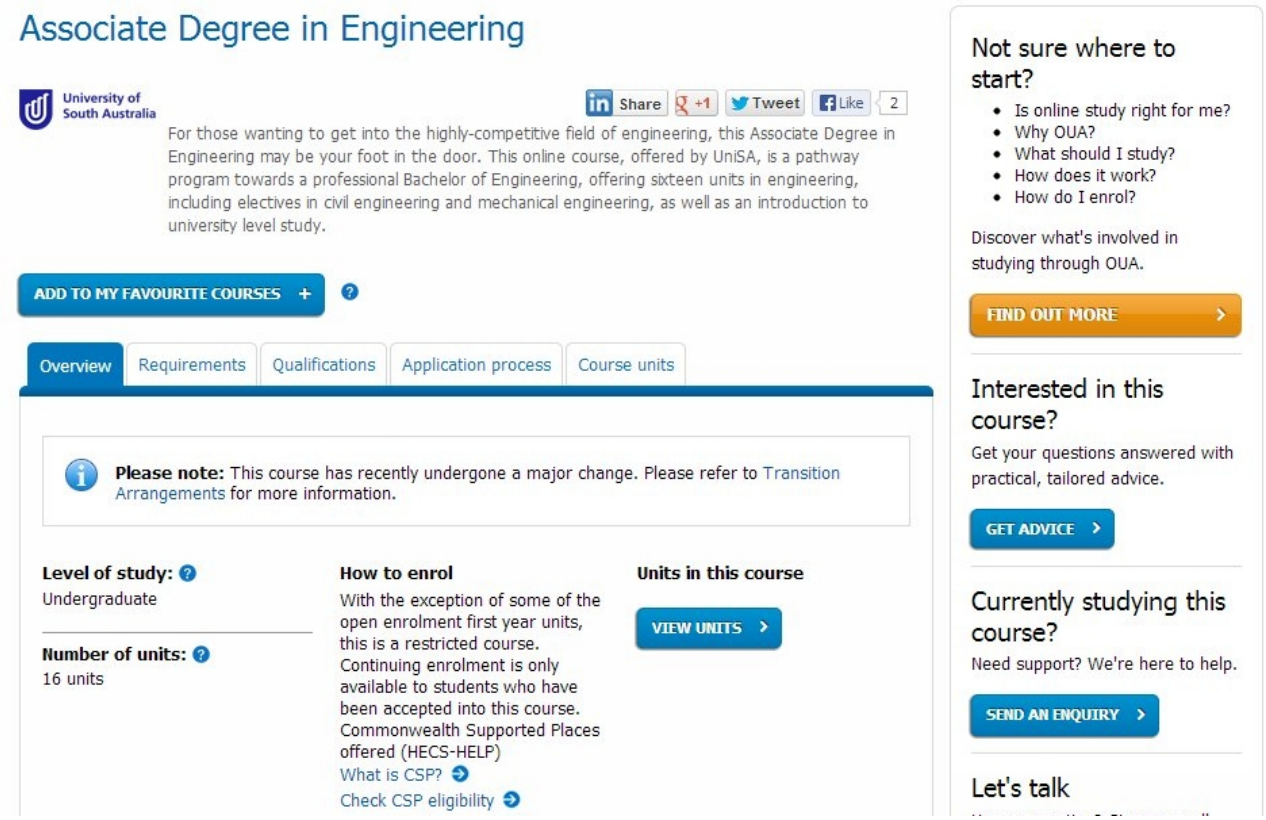


Figure 2.4: An Open University program page[1]

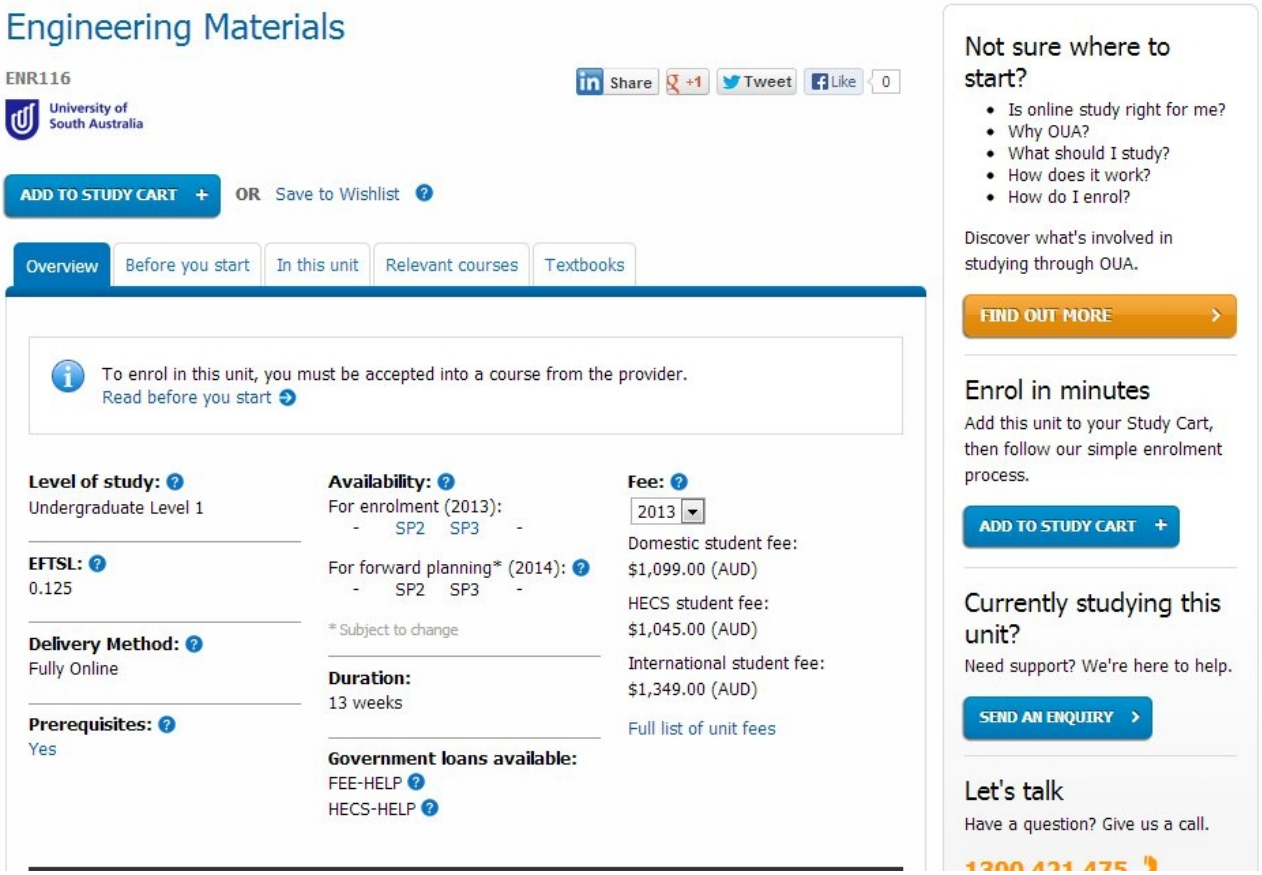


Figure 2.5: An Open University course page[1]

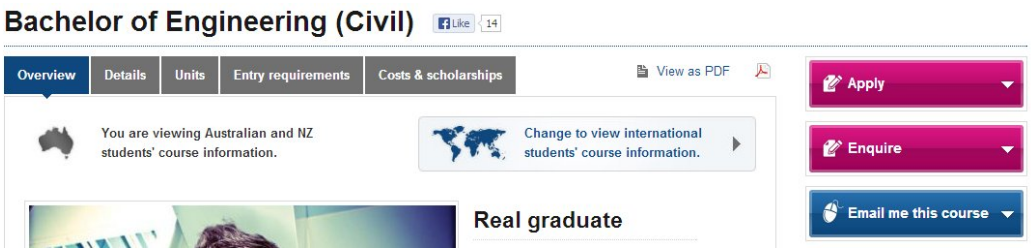
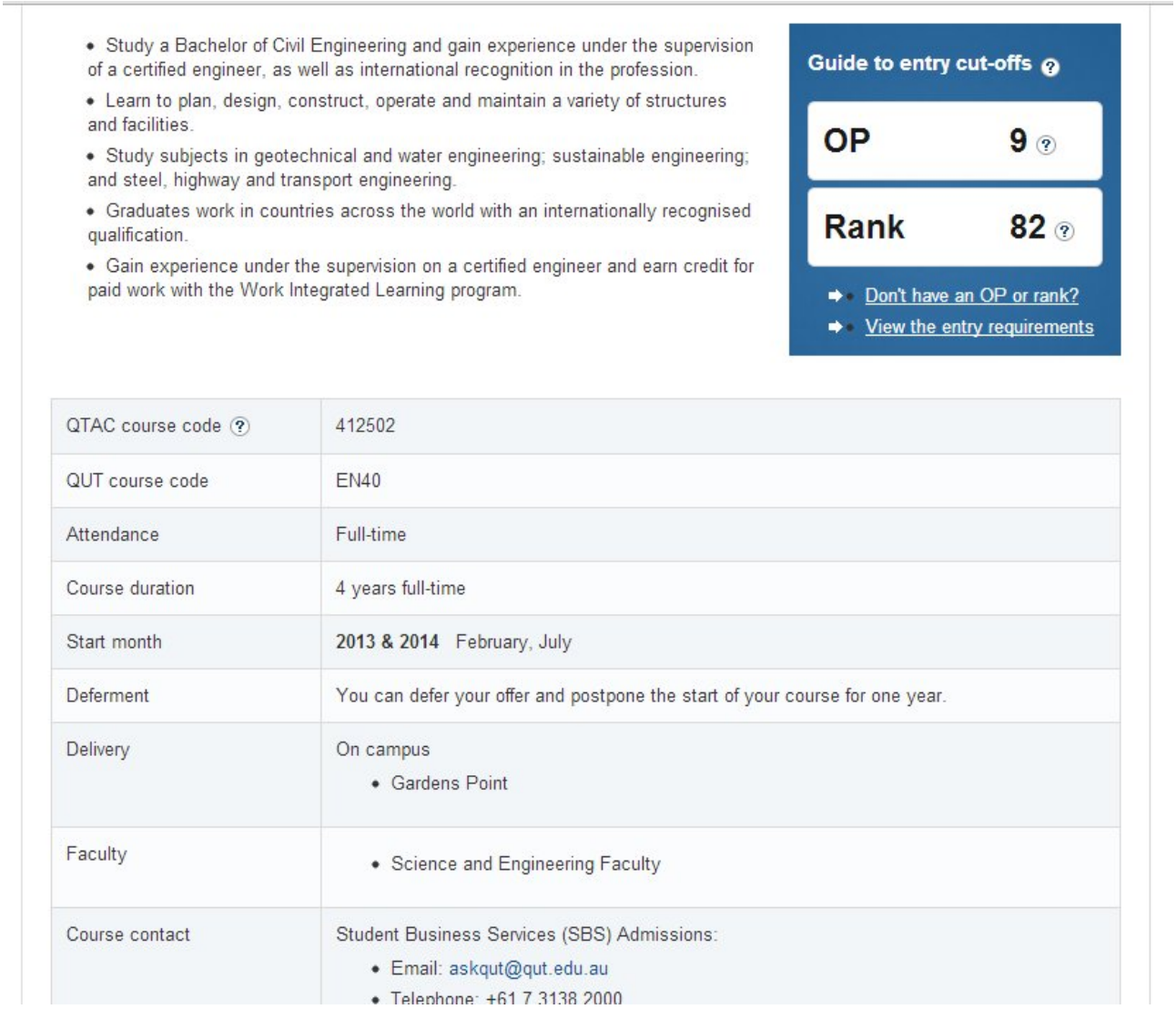


Figure 2.6: QUT program page, menu [8]



- Study a Bachelor of Civil Engineering and gain experience under the supervision of a certified engineer, as well as international recognition in the profession.
- Learn to plan, design, construct, operate and maintain a variety of structures and facilities.
- Study subjects in geotechnical and water engineering; sustainable engineering; and steel, highway and transport engineering.
- Graduates work in countries across the world with an internationally recognised qualification.
- Gain experience under the supervision on a certified engineer and earn credit for paid work with the Work Integrated Learning program.

Guide to entry cut-offs

OP	9
Rank	82

Don't have an OP or rank?
View the entry requirements

QTAC course code	412502
QUT course code	EN40
Attendance	Full-time
Course duration	4 years full-time
Start month	**2013 & 2014** February, July
Deferment	You can defer your offer and postpone the start of your course for one year.
Delivery	On campus • Gardens Point
Faculty	• Science and Engineering Faculty
Course contact	Student Business Services (SBS) Admissions: • Email: askqut@qut.edu.au • Telephone: +61 7 3138 2000

Figure 2.7: QUT program page [8]

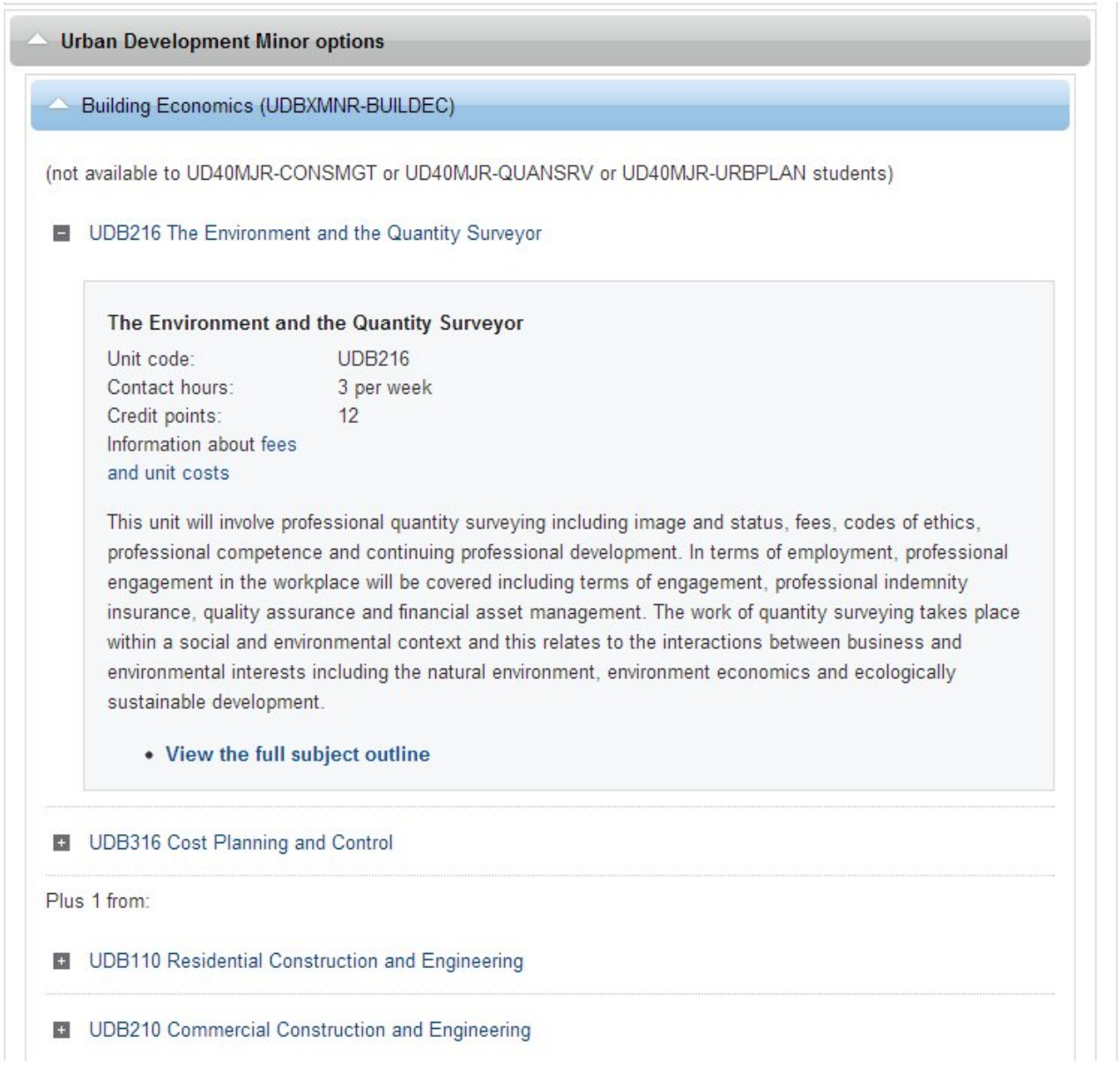


Figure 2.8: QUT course information [8]

2.3 Evaluation

2.3.1 Evaluation Framework

- Accessibility.

  When designing a webpage, the design should support and incorporate accessibility standards. These have been drawn from the W3.org guidelines[2], selecting those which are particularly relevant to this project. In particular, it should be:

  Perceivable: Information should be presented in ways that users can perceive regardless of disabilities. This includes providing text alternatives to non-textual content, making it compatible with screenreaders, braille screens and other textual converters. Content should be able to be presented in different ways, and sufficiently distinguishable, meaning it has properly used colour, sufficient contrast, and is accurately resizable (up to 200 percent) without use of additional technologies and without requiring the user to scroll horizontally to read a line of text.

  Operable: The site can be navigated from the keyboard, and should not contain a 'keyboard trap', where focus can be moved to an area by the keyboard but not away from it. Content should not be timed or flash. Navigability, which is included here by W3, has its own Evaluation section below.

  Understandable: Content should be readable, predictable, and have input assistance where possible.

  Robust: Content should be interpretable by a wide variety of technologies, including assistive technologies.

- Navigability

  Information can be located easily, preferably without leaving the Handbook site. Includes navigation throughout the site as well as across a single page.

- Meaningful and contextual search results

  Search results are specific to the Handbook and UNSW's naming schemes. For instance, 4 digits should return a program where possible (e.g. 3648 should return the 'Software Engineering' program). A string of 4 characters followed by 4 digits should return a course (e.g. COMP3311 should return the 'Database Systems' course).

  Appropriate priority and ranking are given to search results: for instance if I search exactly for a course title, that should be the first result.

- Accuracy and Completeness

Information is present and accurate. Information should not internally conflict or redirect to the wrong sites (e.g. the 2013 handbook should only direct to other 2013 handbook pages, not 2012 pages).

According to Yuen's research, students look for the following information, which should be provided wherever possible: [4]

Programs

- Entry requirements (e.g. ATAR), domestic and international
- Expected attendance, duration
- Description and titles
- Fees
- Enquiries
- Requirements and courses involved
- Alternatives (e.g. double degrees)
- Potentially relevant scholarships

Courses

- Pre- and co-requisites
- Excluded courses
- Description and titles
- Workload
- Availability
- Relevance to program
- Contact hours and timetable
- Fees
- Enquiries/LIC
- Supplementary information (e.g. syllabus, textbooks, assessments)

- Performance

  Speed. The site can perform well under stress testing, as it is expected to perform well when simultaneously serving thousands of students at enrolment time.

- Inline frames

  Specific and relevant only to the new handbook. The new site should be able to create and process inline frame sources which produce accurate information, provide flexibility and granularity of data selection, are customizable, and have a list of predesigned templates. These frames and their information should satisfy the above criteria, particularly performance and accuracy.

2.3.2 Evaluation of current UNSW Handbook

- Accessibility

  Good. Little to no non-textual content, colour-blind safe. Could perhaps benefit from a bit more contrast between text and background at points, such as subtitles on course pages (slightly-

darker blue against light blue background). Can zoom up to 200% and read the main section without needing to scroll.

- Navigability

  Poor. The handbook sometimes redirects to outside sites (e.g. SENG 3648 in figure 2.3). Occasionally, these outside sites then redirect back to the Handbook for other information.

  Additionally, some links are broken (e.g. 'Plan Structure' for COMPAS8543 as of 27th May 2013).

  Students report accidentally finding themselves on earlier versions of the Handbook and not noticing.

  Beyond this, the basic design is sound, although there's a little confusion resulting from having two separate searches.

- Search results

  Search results are reasonable; contextual and ranked, although the ranking is imperfect (multiple matches within the description can overrule a perfect match on the course name, for instance a search for 'Database Systems').

- Accuracy

  Information is not always present or may conflict with school sites or suffer broken links (see figures 2.1, 2.2, 2.3 and the Handbook page for COMPAS8543).

  Generally, program pages do not directly list alternatives or potential scholarships, but provide all other relevant information. Course pages always list pre- and co-requisites, excluded courses, the workload and a description, but either do not always list or require referring to another site to find everything else.

- Performance

  Good. Given that the existing site uses static pages, any proposal I make will certainly be unable to surpass it, given the technical requirements of dynamically generated pages and inline frames.

2.3.3 Evaluation of other sites

Open University [1]

- Accessibility

  Good. Has meaningful text alternatives to important images and buttons, can zoom up to 200% etc.

- Navigability

  Good. In some respects better than UNSW's, especially as course names in program lists are hyperlinked so you don't need to do a separate search for them.

- Search results

  Good. Not only has meaningful results, but when using the search-bar it will provide a drop-down list of likely results (e.g. autocompleting a course name).

- Accuracy

  Does not list alternative programs on the program page (they may not exist), and the course outlines given are rather brief (generally just a list of topics and provided material types and textbooks). Beyond that, the information provided is impressively complete.

- Performance

  Seems fine. Not able to stress-test, obviously, but no apparent latency. Everything seems preloaded, unlike QUT below.

Queensland University of Technology [8]

- Accessibility

  Should be fine, assuming the javascript/AJAX is properly set up to handle screenreaders. Can only zoom up to around 175% before having to scroll, whereas W3C expects 200% [2].

- Navigability

  Only allows viewing a course from the program page. While the ability to do so is useful, only being able to do so may cause problems. This is certainly not something that I would restrict the UNSW Handbook to, as UNSW Handbook users have adapted to separate program and course pages.

- Search results

  Cannot easily search for a course, as courses are strictly accessed through programs. There may be another site I cannot access which handles enrolled course information.

- Accuracy

  Seems to provide all relevant program information. Does not appear to list pre- and co-requisites for courses, but the strict scheduling may be seen to imply these. Does not appear to provide contact information or LIC for courses. Some of this information might be on another University site, but I am strictly evaluating their Handbook equivalent.

- Performance

  Some of the AJAX queries (such as when loading information on a course from a program page) can take some time to process.

Chapter 3

Proposed Solution

I aim to produce a web server which reproduces the existing functionality and additionally provides more information and the 'external service'.

To address problems with the existing Handbook, the new Handbook shall source its data from the AIMS database. Properly done, this reduces the chance of data duplication or inconsistencies. The AIMS database also provides significantly more information than is currently presented in the Handbook, which may be useful.

The 'external service' provides the means to embed this AIMS information in other pages. This can be accomplished using inline frames, which direct the browser to insert part of another HTML document into this page.

To properly allow customization of the external service results, the server shall parse specifically formatted URLs and produce a specific dynamic result. As such, the system shall also provide a means to create these URLs and the associated tags.

This allows information to be presented on multiple sites without data duplication, which is especially useful for school sites. This information can then be presented in different ways or with supporting information.

In this respect, the web server also provides a service beyond the handbook itself, with the external service acting as a secure gateway to the relevant information in the AIMS database and allowing users to customise the presentation of this information.

By pulling information from the database rather than manually creating each page, data only needs to be updated in one place, reducing workload and improving consistency. This is further improved by allowing schools and other sites to draw upon handbook information for their own pages, and encourages security by not requiring direct database access for these external sites. The template should also be more user-friendly than writing and handling your own database queries.

The web server shall be written in PHP, with PostgreSQL to access AIMS. As AIMS is already written in PostgreSQL, it is the simpler and more natural choice. I am already familiar with PHP and comfortable with similar languages, and it is well-suited to the proposal.

The basic webpages are written in standards-compliant HTML5 and CSS. More complicated expansions can be done through the use of Javascript and AJAX, such as is done by QUT[8].

Chapter 4

AIMS Database

The AIMS database is provided and maintained by the University of New South Wales. Weng Lok Mok is in the process of writing scripts and moving data from the old MAPPS database to AIMS, as well as reworking the schema. The mapping is currently incomplete. My use of the database has turned up the following:

A record, for this text, refers to a row in the 'record_instances' table and corresponding rows in related tables. An entry refers to an actual course, program or stream found in the current Handbook, which is presumed to be the most up-to-date official documentation.

There is no single column which determines whether a record is 'current' or not in referring to an entry. A current record is thus defined as an entry in records with an 'AP' approved proposal status, flagged is_published, not flagged as a skeleton record and with no enddate. Current records are translated into entries in the new Handbook. The r_records view initialized in reset.php applies these filters. There are 4977 current records out of 14972 as of this writing.

```
CREATE VIEW r_records AS
      SELECT id, code as fullcode, name as recordname
      FROM record_instances
      WHERE enddate IS NULL
        AND is_published IS TRUE
        AND (proposal_status = 'AP')
        AND is_skeleton_record IS FALSE;
```

There currently exist approximately 317 entries with more than two current records, covering a total 657 rows. They may be determined by running the following query on the rwcaAIMS database (which has the above r_records view):

```
SELECT fullcode, count(*)
FROM r_records
GROUP BY fullcode
HAVING count(*) > 1
```

In cases where a table is completely empty in my version of AIMS, I have only included functionality relying on this table if I can clearly deduce what it is and how this information is represented (e.g. course prerequisites were displayed before any were present in the database).

The following query was run to test for empty tables, and recovered 64 empty tables in the AIMS schema. These include pr_scholarships (program scholarships), pr_pgrd_admissions (postgraduate program admission requirements), co_exclusion_groups (which courses exclude other courses) and others.

```
VACUUM ANALYZE;
select relname, relnamespace from pg_class
where relkind = 'r'
  and reltuples = 0;
```

In addition to this, a list of empty columns was generated by running:

```
VACUUM ANALYZE;
select schemaname, tablename, attname
from pg_stats
where most_common_vals is null
  and most_common_freqs is null
  and histogram_bounds is null
  and correlation is null
  and null_frac = 1;
```

This results in 64 non-empty tables having 144 empty columns between them. Notably, the pr_ugrd_admissions table (undergraduate program admissions requirements) contains 10 empty columns of its total 13. The highly relevant program_records, course_records and stream_records contain 35, 13 and 5 empty columns respectively. These have been listed below 4.1.

Additional idiosyncrasies include:

- Every assessment having the same semester category (Semester 1).
- Some column titles are inconsistently applied, e.g. 'descr' versus 'description'.
- There are several different 'name' columns available, many of which are identical. This may be a result of backwards compatibility in some cases.
- There does not appear to be any formatting regulations or requirements for various text sections. For example, some outcomes are given as one outcome per dot-point (as apparently intended), while in other cases an entire HTML list is kept in just one row. At other times, individual outcomes are stored on rows, but each one is wrapped in HTML paragraph tags, interfering with proper display of these records.

- Various records, although approved and published, are not complete entries in and of themselves, e.g. they note 'No changes' to the outcomes rather than listing them. Others are incomplete for other reasons. Presumably most of this will be fixed as more data is moved from MAPPS.

program_records:	**course_records:**	**stream_records:**
cohort_of_students_details	key_transition_pt_details	replaces_existing
uac_code_type	intensive_mode_start_sum	campus
last_semester_offered	intensive_mode_num_days	delivery_reasons
has_different_english_requirements	intensive_mode_contact_hours	websites
ielts_score	intensive_mode_comments	resource_commitments
uni_strategic_alignment	pre_req	
fac_strategic_alignment	websites	
program_advantages	proposal_justification	
expected_demand	lnt_discussion_details	
marketing_requirements	next_review_date	
campus_percentage	teaching_mode	
cohort_of_students	intensive_mode_start	
justification_for_english_reqts	other_lms_used	
volume_of_learning		
recognised_program		
include_capstone_experience		
research_component		
academic_performance_required		
can_exit_programs		
exit_programs_details		
credit_transfer		
websites		
must_complete_work_based_trng		
intl_stu_placement_process_details		
intl_stu_placement_supervision_details		
impact_on_workload		
has_twinning_arrangements		
has_formal_partnerships_with_ext_orgs		
ext_org_partnership_details		
requires_use_of_cats		
requires_use_of_additional_facilities		
additional_facilities_details		
requires_specialised_equipment		
specialised_equipment_details		
offers_coop_scholarship		

Table 4.1: Empty columns in the three major tables.

Empty tables	
co_intensive_offering_details	pr_pathways
co_pre_reqs	pr_ext_org_partnership_memorandums
co_exclusion_groups	pr_websites
co_learning_management_systems	pr_twinning_arrangements
co_rich_media	pr_work_based_training_attachments
co_equivalence_groups	shared_consultations
co_learning_outcomes_mapping	shared_consultants
co_equiv_group_members	shared_consultation_attachments
co_excl_group_members	shared_documentation
co_offered_in_programs	shared_disestab_info
co_offerings	shared_documentation_attachments
co_rules	system_notices
dual_award_program_records	st_related_streams
dp_constituent_programs	shared_workflow_convenors
comments	shared_learning_outcomes_mapping
co_websites	st_articulated_streams
job_classes	st_modes_of_delivery
item_edits	st_postgrad_pathways
new_external_users	st_stream_replacements
orgunit_info_history	subject_area_orgunit
orgunit_replacements	pr_faculties_sharing_teaching_load
pr_competitor_program_attribs	pr_twinning_memorandums
pr_delivery_locations	affiliation_requests
pr_partnerships_with_ext_orgs	pr_future_career_opportunities
pr_memorandum	co_course_replacements
pr_pgrd_admissions	notifications
pr_related_programs	pr_similar_progs_offering_coop_schol
pr_scholarships	pr_articulation_rules
pr_cognate_disciplines_required	media_uploads
pr_domestic_fee_options	shared_occupations_available
pr_exit_programs	shared_media
pr_expected_demand_docs	shared_records_to_be_disestab_with

Table 4.2: Completely empty tables

Chapter 5

My Solution

My solution uses PostgreSQL and an Apache PHP server to serve the website. Both are currently hosted by UNSW CSE, available at `rwca915.srvr.cse.unsw.edu.au`.

The web server accesses my copy of the AIMS database (rwcaAIMS), to which I have made some additions (four views and one table) but no other alterations: it may be easily generated from an AIMS dump.

Resetting the database can be accomplished by simply dropping the rwcaAIMS database, recreating it from the AIMS dump, and running `rwca915.srvr.cse.unsw.edu.au/reset.php` to import the class schedule and create several important views.

I have made use of two Javascript plugins: Menucool Tabbed Content[5] and Mottie's Tablesorter[6]. The former was used in all places where tabbed content is available; the latter was used for all sortable tables. Tables which are scrollable but not sortable, such as those found on the Search page, were not created with Mottie's Tablesorter due to compatibility problems.

The UNSW logo and program, stream and course icons were drawn from the UNSW Handbook[7], and I edited in an alpha channel to each of them.

5.1 Product

5.1.1 Page Guide

The **outer frame** is present on each page, and contains a "UNSW AIMS Handbook" header and a sidebar. The header is a link to the home page. The sidebar has links to the Programs, Streams and Courses Lists and a searchbox, plus a link to the Advanced Search page. An example of this is visible in Figure 5.1.

The **home page** lists a brief glossary.

The **list pages** (**Programs**, **Streams**, **Courses**) list all recognized and valid programs, streams and courses in the database. These are provided in a sortable scrolling table, and link to the respective information pages. An example of this is shown in Figure 5.1.

The **Search page** includes a number of advanced search options, plus a results section that is only

visible if a search has been performed. An screenshot of a results search page is shown in Figure 5.2. A search for * or % will return every known entry, much like the list pages. The search page has basic protection against SQL-injection attacks.

The advanced search options have slightly different modifications depending upon the row. Some increase the search range if ticked, others decrease it (e.g. allowing courses versus requiring courses to be available). I have attempted to make it clear what each box will do through appropriate labelling and separation. These search options are turned into queries made to the database: no filtering is performed by the PHP layer.

The results section is separated into three tabs, one each for courses, streams and programs, with each tab only displayed if these results have been searched for. Each results list is a scrollable table, but the table is not sortable due to how the Mottie's Tablesorter[6] plugin interacts with hidden-on-load divs. It may be possible to overcome this through further use of javascript.

The **Information** pages (**Program**, **Stream**, **Courses**) include information specific to that type of record. Each record is identified by the ?id= URL argument. This means that in cases of duplicate entries (discussed in the AIMS section) only the first valid entry is accessible through an information page. This would not be a problem with a complete and checked database, and makes the URLs significantly more human-readable than linking through an AIMS internal primary key identifier.

Green text on the information pages indicates that this was a 'text' field drawn from the database. This can be seen in figure 5.3.

The Stream pages have a brief proof-of-concept external service. See Figures 5.4 and 5.5 for the example options and result, respectively.

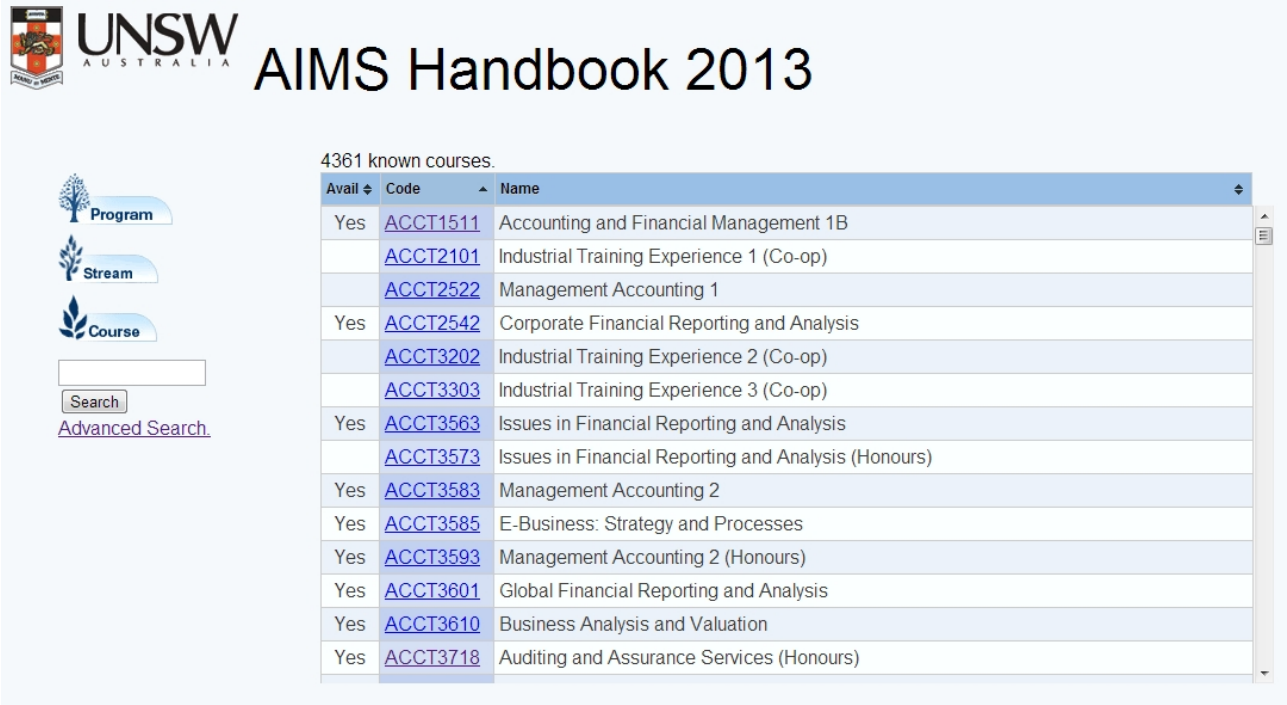


Figure 5.1: The courses list page, showing both the outer frame and a scrollable/sortable table

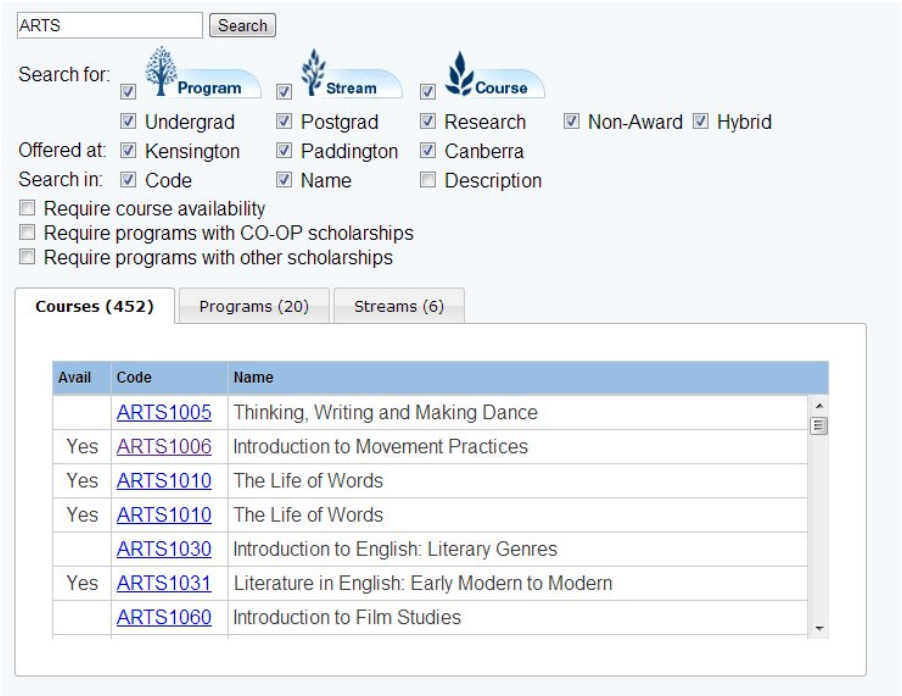


Figure 5.2: The result of a basic search for 'ARTS'

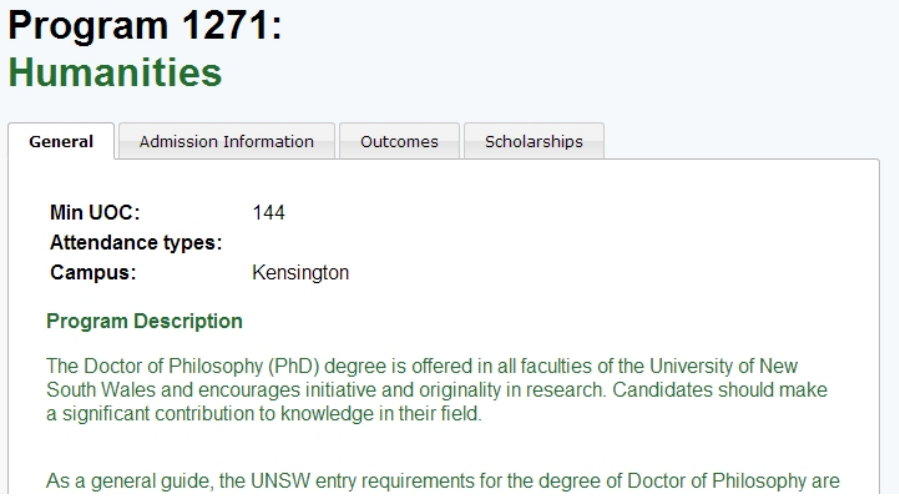


Figure 5.3: An example program page.

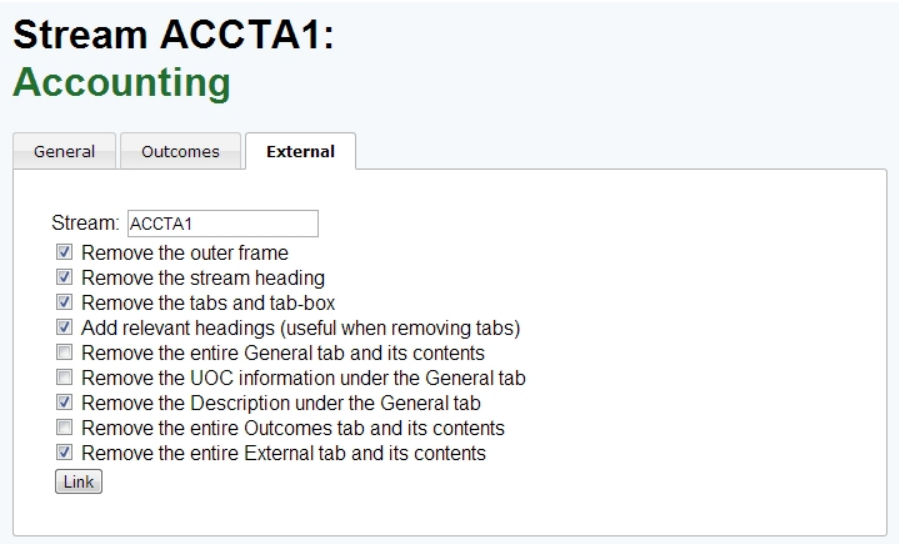


Figure 5.4: An example of the external service options under a Stream page.

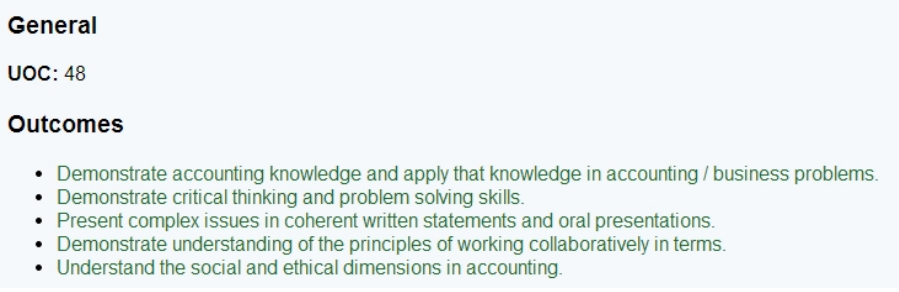


Figure 5.5: The result of clicking ’link’ in the above example.

5.2 Development

I have a local development server, and regularly exported through FTP to the CSE server for testing. The only change necessary is redirecting the PHP-Postgres database connection to target the appropriate database.

I received an updated version of the AIMS database every few weeks, and integrated it. This occasionally involved schema changes, but generally only added data.

The three-part structure of the website (Programs, Courses and Streams) was based on both the existing Handbook and the AIMS Database itself. AIMS has one overarching table (record_instances) to store information that is common between types, then three linked tables (program_records, stream_records and course_records) which store information specific to these types.

Initially, I developed basic browsing and information functionality (i.e. the 'courses' and 'course' pages). Information presented was largely based on what was found in AIMS as well as the results of Yuen's surveys.

Then I improved the display of these pages (introduction of tabs, sidebar, etc.), and automatically inserted links on course codes and redirected links to the new Handbook rather than the old. The tabs structure was inspired both by Yuen's site and by dissatisfaction with the current Handbook display (requiring either scrolling or clicking through to other pages). The information available was limited by what was currently in AIMS: for instance, I could not .

Next I introduced searching functionality and expanded this to advanced search. Search options were based upon AIMS information and expected use. Having experienced the current Handbook's myriad distinct and confusing search engines, I chose to implement the 'basic' search on the sidebar as simply a preselected list of options in the advanced search, so as to have a standard search engine across my entire site.

I reworked pages to ensure they would display well and maintain as much functionality as possible with javascript disabled. For the list page tables this meant ensuring they would retain scrolling functionality without javascript. For the search and information pages, the tab functionality degrades: all of the text is visible in one long page, and clicking a tab simply jumps you to the appropriate line. I added headings to the relevant points to clearly separate sections.

Finally, I added in a brief, proof-of-concept external service to the Streams page. A new tab is added with a number of options, and a redirect button. Each option modifies the URL to add or remove something from the Streams page, essentially allowing you to specify which information you want to view and (to a limited extent) how you want it presented.

While developing the information pages' database querying functionality, I struggled with minimizing and offloading my queries onto the database and handling some of the internal workings there. My initial impulse was to reduce the database, removing unnecessary rows or tables to increase efficiency and simplify queries. However, I was stymied by non-cascading foreign keys linking the tables. A script to overcome this would have been unnecessarily complex and made updating the database more difficult.

Instead, I developed reset.php which creates several views and also loads the class schedule into the database. It can take several minutes to run, depending upon the server load, but is intended to only be run when the database or class schedule is updated. These views both filter out unwanted records and rename certain columns to a standard format shared between views, making for simpler queries.

Properly integrating the class schedule (.sched file) took some time: it involved applying various PHP functions to turn the .sched files from multi-row entries into a single row per course, then converting this into an appropriate table. The code to do this is found on the reset.php page. The timetable is a single table containing lots of information, much of which is not currently used, in order to simplify the importing process and leave open the option of further developments using this information.

5.3 Testing

Display was primarily tested on Chrome and Firefox. Minor cosmetic issues are present at extreme resolutions or zooms (things bleed into one another or wrap oddly).

The site is mostly HTML5 compliant. This was validated through use of the W3C Markup Validation Service at `http://validator.w3.org/check`. Known problems:

- Use of <noscript>tags inside tables is apparently forbidden. Looking for a compliant way to have tables retain scrolling functionality when javascript is disabled.
- There may be some lines where htmlspecialchars() (a function to rework certain characters as HTML-printable, such as <, >, &) is not being used.
- There were some errors relating to markup drawn directly from the database, such as in course descriptions, where the supplied text is not HTML5 compliant (e.g. using the 'name' attribute).

Corrected issues include some HTML links not being placed in quotations, incorrect use of some tags or parameters, and empty rows at the end of some loop-generated tables.

Validity testing included checking output of pages versus data in database and the original class schedule file, and comparing results (where possible) to the current Handbook. Data was always a match to the database, and was correct in major respects when compared to the Handbook - occasionally names and the like were slightly different, but they were obviously the same entries.

Notably, there were some mismatches between my schedule file and classes at `http://www.timetable.unsw.edu.au/` - compare ACCT2542, where classes that I have marked as Monday are given as Tuesday by the .unsw website. The Monday times were accurate with respect to the .sched file I was given.

Usability testing was performed by asking classmates and roommates to use my Handbook as they would the actual Handbook, with the caveat that many entries were missing and that green text is unedited. Feedback from this primarily went into shaping and clarifying some of the search options.

Other feedback included that they were happy my Handbook showed availability when browsing, but wished the timetable display was more like a timetable. Some felt it would be useful to see UOC when browsing or searching.

Speed testing was performed using `http://fpt.pingdom.com/` to get an average load time across several requests. Performance grade is an evaluation of how much has been done to reduce load times, and was reduced primarily due to the lack of proper caching declarations. Results varied depending upon server load: these tests were performed on a weekend evening, and the most recent results are provided. These numbers include client-side actions such as javascript.

http://rwca915.srvr.cse.unsw.edu.au/

Unlike other entries, this is a static page with no javascript and is thus useful as a baseline.

5 requests. Performance grade: 84/100. Load time: 1.25s.

programs.php

13 requests. Performance grade: 78/100. Load time: 5.34s.

streams.php

13 requests. Performance grade: 78/100. Load time: 2.94s.

courses.php

12 requests. Performance grade: 86/100. Load time: 7.36s.

program.php?id=3403

8 requests. Performance grade: 84/100. Load time: 1.81s.

stream.php?id=ECONFS

8 requests. Performance grade: 84/100. Load time: 1.35s.

course.php?id=ACCT2542

9 requests. Performance grade: 83/100. Load time: 1.86s.

search.php

12 requests. Performance grade: 78/100. Load time: 2.02s.

search.php?query=ARTS

13 requests. Performance grade: 80/100. Load time: 2.55s.

search.php?query=*

13 requests. Performance grade: 80/100. Load time: 5.09s.

Basic SQL-injection attacks were tested against the search and information pages (which apply user input to queries). Current prevention techniques simply sanitize inputs before placing them in queries. More advanced SQL-injection protection could be introduced; I tried to update to pg_query_params but ran into weird errors and lacked time to work them out and convert over.

The queries tested were inserted into the search box or placed in the URL in different encodings. Many were drawn from `http://ha.ckers.org/sqlinjection/` or other websites. They included:

- `1'1`
- `1\'1`
- `';DROP TABLE timetable;--`
- `'";`

5.4 Evaluation

- Accessibility.

  There are some issues with sections bleeding into each other at extreme zooms.

  What few images there are have alt-texts as appropriate. All other information is textual.

  There are no 'keyboard traps'.

- Navigability

  As descriptions are drawn directly from the AIMS database, certain external links still persist. Links that would direct to certain Handbook pages are redirected to corresponding internal pages, however this redirection does not currently distinguish between entries that are in AIMS (and thus the new Handbook) and entries that have yet to be copied over to AIMS. This results in 'broken' links.

- Meaningful and contextual search results

  Search includes basic and advanced search. Results have reasonable contextual interpretation: they are separated into courses, programs and streams and ordered first by matching query to record code (e.g. COMP1927 or SENGA1) and then alphabetically by record code. This could be improved: attempts to integrate dynamically sortable javascript tables failed.

- Performance

  Refer to the 'speed testing' section of 5.3 for hard numbers. Generally, the pages load very quickly; slowdowns are seen in pages with large tables such as the list pages or a very generic search. The list pages are particularly slow due to the use of the tablesorter plugin, which has additional client-side overhead: compare the speed of the universal search to the speed of the list pages.

- External Service

  Very basic, only supplied for Streams, limited options.

- Accuracy and Completeness

  Some of these are not included because the database does not currently contain the relevant information, such as alternative programs (the pr_related_programs is not populated).

  Programs

  - ✓ Entry requirements (e.g. ATAR), domestic and international
  - ✗ Expected attendance, duration
  - ✓ Description and titles
  - ✗ Fees
  - ✗ Enquiries
  - ✗ Requirements and courses involved
  - ✗ Alternatives (e.g. double degrees)
  - ✓ Potentially relevant scholarships

  Courses

  - ✓ Pre- and co-requisites
  - ✓ Excluded courses
  - ✓ Description and titles
  - ✓ Workload
  - ✓ Availability
  - ✗ Relevance to program
  - ✓ Contact hours and timetable
  - ✗ Fees
  - ✗ Enquiries/LIC
  - ✓ Supplementary information (e.g. syllabus, textbooks, assessments)

5.5 Comparison

The old Handbook is, obviously, currently more complete in terms of number of entries. However my new Handbook provides more information within the Handbook itself, where the old Handbook often relies upon redirecting the user to separate sites. This makes the new Handbook more navigable than the old.

The new Handbook provides the aims and outcomes where relevant, plus the timetable for courses. The latter allows my Handbook to be searched and sorted according to which courses are currently available in the given schedule, a very useful option for students.

The new Handbook also splits ”Indicative Contact Hours per Week” into the known types such as ”Lecture”, ”Tutorial”, ”Laboratory”, etc. and provides a brief overview of assessments (with the amount of information provided dependent, of course, upon how much is stored in the database).

The new Handbook’s search engine allows for a few more choices than the old Handbook’s, based upon the information stored in the new Handbook - such as searching for available courses, or programs that provide scholarships.

The ’External Service’ provided for streams, although currently a brief proof-of-concept, could serve an important role in external pages that wish to make use of Handbook information.

Bibliography

[1] Open Universities Australia. Open Universities Australia Courses and Units. http://www.open.edu.au/courses/. Last accessed 25th April 2013.

[2] World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 2.0. http://www.w3.org/TR/WCAG20/. Last accessed 20th May 2013.

[3] Yuen Wing Kei. Handbook Prototype. mahler.cse.unsw.edu.au/handbook. Last accessed 25th April 2013.

[4] Yuen Wing Kei. A New Student Handbook for UNSW. Undergraduate Thesis A, University of New South Wales, 2012.

[5] Menucool. Menucool Tabbed Content. http://www.menucool.com/tabbed-content. Last accessed 24th October 2013.

[6] Mottie and Christian Bach. Mottie's Tablesorter. https://github.com/Mottie/tablesorter. Last accessed 24th October 2013.

[7] University of NSW. UNSW Handbook. http://www.handbook.unsw.edu.au/. Last accessed 25th April 2013.

[8] Queensland University of Technology. QUT Study. http://www.qut.edu.au/study/. Last accessed 25th April 2013.