Software Service Design and Engineering

Course Details

Course Code	**COMP9322**
Course Title	**Software Service Design and Engineering**
Conveners	Shayan Zamani, Madhushi Bandara, Mortada Al-Banna
Classes	See timetable
Consultations	See timetable
Units of Credit	**6**
Course Website	http://cse.unsw.edu.au/~cs9322
Handbook Entry	http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9322.html

Course Summary

Today's modern era permeates with phenomenal growth and advancement across a wide range of areas: from science and technology, to medicine and art, as well the social sciences. Amongst all, the notion of "services" plays a critical role to abstract and make available snippets of innovation, that can thereby be used cross-disciplinary to create bigger, better and more interesting innovations. This course arms students with the knowledge and expertise to "design" and "engineer" services using modern Web technology.

Students will appreciate the importance of the Service Oriented Architecture (SOA) as a way to design an enterprise environment that features cross-platform compatibility, agility and cost-efficiency. In particular, microservices variant will be taught and serverless computing will be introduced. Also students will learn about cognitive services and how conversational Bots can be used to allow natural interactions between Humans and services.

In addition, students will be introduced to principles of semantic web services and information modeling using semantic web technologies. The course will strengthen students' data modeling expertise, covering topics on semantic information modelling of both simple and complex structures,

their usage in business and the technical standards and technologies that support semantic data modelling.

Course Aims and Learning Outcomes

After completing this course, students will:

- Describe different architectural design approaches and their role in engineering software.
- Apply modelling of services in different settings
- Be competent in designing, managing and documenting APIs.
- Understand techniques for semantic information modelling of simple and complex structures, as well as their associated technical standards and technologies.

This course is part of a three-part series of the Data and API Engineering stream alongside COMP9321 and COMP9323.

Prerequisites and Assumed Knowledge

To take this course, a student must have programmed in a programming language (preferably Python), have learnt how to organize and query the data in a relational data model (e.g. SQL), and have basic knowledge of the workings of the Internet (TCP/IP, sockets, etc.) and Web Technologies (HTML, CSS, JavaScript, XML and AJAX).

Formally, the prerequisites are: Undergrad - COMP1531 and COMP2041, Postgrad - COMP9021 and COMP9311

Teaching Strategies

- Lectures: introduce concepts, show examples
- Online Quizzes: revision of the concepts introduced in Lectures.
- Labs: perform practical exercises
- Assignments: solve significant problems (individual assignments)

Weekly Activities

Schedule of topics, organised by week:

W	Lecture-1	Lecture-2	Lab	Activities
1	**Introduction & Service Orientation Design**	**Web APIs** *Principles of RESTful design*	*No Lab activity*	
2	**Web APIs (cont'd)** *Accessing and using a simple Web Service, Documenting*	**Web APIs (Cont'd)** *Advanced topics in designing RESTful APIs (Authentication & Scaling)*	*No Lab activity*	Quiz-1

		REST API with Swagger		
3	**Containers** *How to run a service in a container (Docker)*	**Cloud Computing** *SaaS, PaaS*	Lab-1: *RESTful Activity*	Quiz-2
4	**Cognitive Services Part1** *Introduction, Chatbots, concepts, design*	**Rule-based Chatbots** *Building a simple chatbot using some tools*	Lab-2: *Docker*	Lab-1 activity deadline
5	**Cognitive Services Part2** *Machine learning Chatbot development and tooling*	**Machine learning-based Chatbots** *Building a simple smart chatbot*	Lab-3: *Rule-based chatbots*	Assignment 1 deadline Lab-2 activity deadline
Week 6: Flexibility Week – No Lecture/Lab				
7	**Cognitive Services Part3** *Introduction to prototyping and sampling*	**Prototyping Chatbots** *Building a prototype for a simple chatbot*	Lab-4: *Machine learning-based chatbots*	Quiz-3 Lab-3 activity deadline
8	**Introduction to Semantic Web** *Knowledge graphs and principles of semantic web*	**Understanding Graph Data** *Exploring and designing graph data models.*	*Lab-5: Explore graph database (Neo4j)*	Quiz-4 Lab-4 activity deadline
9	**Semantic Web Technology Stack** *RDF, RDFS and OWL*	**Semantic Web Services and Applications** *Exploring applications of semantic web technologies.*	*Lab-6:* *Explore semantic web application.*	Quiz-5 Lab-5 activity deadline
10	**Querying linked data with SPARQL.**	**Exam Warm-up Session**	*No Lab activity*	Assignment 2 deadline Lab-6 activity deadline

Assessment

The assessment of the following components:

- **Quizzes (10%)**: this component help in the revision of the concepts introduced in the lectures. There will be 5 quizzes in total.
- **Assignments (30%)**: This component assesses the modelling skills related to learning outcomes. Students are given a major case study involving building an information system for an organisation. There will be 2 assignments in total each constructing 10% (assignment 1) and 20% (assignment 2) of your final mark.
- **Practical Labs (10%)**: various assessments based on the lab activities; you need to attend 5 out of 6 labs showing the activity represented during the lectures.
- **Written final exam (50%)**: individual assessment. This component is going to assess the various facts-and-knowledge level learning outcomes. The exam is a mixture of multiple choice questions and written answer questions.

Resources for Students

There is no formally prescribed textbook for this course. Recommended reading lists will be provided through the online Q/A (online course forum) throughout the semester. However, the following books are highly recommended and relevant for the topics covered in this course:

- SOA: Principles of Service Design, Thomas Erl, 2008, Prentice Hall.
- Web Services: Concepts, Architectures and Applications, Alonso, Casati, Kuno, Machiraju, , Springer 2004
- RESTful Web APIs, Richardson and Amundsen, O'Reilly, 2013
- Web Service Implementation and Composition Techniques by H-Y Paik, A. Lemos, M. Barukh, B. Benatallah and A. Natarajan, Springer, late 2017.
- Building Microservices, Newman, Sam, 2015, O'Reilly Media.
- Reactive Microsystems, Jonas Boner, 2017, Lightbend, Inc.
- DevOps: A Software Architecture's Perspective by L.Bass, I. Weber and L. Zhu, Addison-Wesley, 2016
- Semantic web for the working ontologist: effective modeling in RDFS and OWL by Dean Allemang and James Hendler. Elsevier, 2011.

Student Conduct

The Student Code of Conduct (Information , Policy) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all

external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer , Grievance Officer , or one of the student representatives.

Plagiarism is defined as using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several online sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity
- UNSW Plagiarism Procedure

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism

The pages below describe the policies and procedures in more detail:

- Student Code Policy
- Student Misconduct Procedure
- Plagiarism Policy Statement
- Plagiarism Procedure

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students

Course Evaluation and Development

This course is evaluated each session using the standard UNSW course evaluation system.

Return Campus and Health Advice

- Safe Return Campus FAQs
- NSW Health Advice