SENG2021 Requirements and Design Workshop

Course Code:	SENG2021
Course Title:	Requirements and Design Workshop
Units of Credit:	6
Course Website:	http://www.cse.unsw.edu.au/~se2021/
Handbook Entry:	https://www.handbook.unsw.edu.au/undergraduate/courses/2019/SENG2021

Course Summary

This course is part the series of software engineering workshops designed to teach students to work in teams and apply their knowledge to solve real-life problems. This workshop will offer students the opportunity to concentrate on software requirements analysis and design issues including artefacts produced as well techniques and tools to support this process (brainstorming, problem statements, requirements elicitation, producing design documents and prototyping). In addition, it aims to provide students with some of teamwork skills, requirements engineering and design techniques that an engineer would use in the early stages of the development process. The students will also be getting experience on different aspects of designing a Web application with a major focus on the front-end. The requirements for this course will be determined in collaboration with industry partners and will relate to developing a realistic application. Most of the teaching will be conducted via mentoring of the teams. At the beginning of the course, some lectures will give background on some key concepts and technologies and on how to produce artefacts in general. The course has a number of industry sponsors that include Fairfax Media and Macquarie Bank.

Course Aims

To develop:

- a practical appreciation of the software requirements and design process;
- an understanding of the relation between user requirements, design concepts and implementation considerations;
- an understanding of the quality of project management and the role of managers, users, designers, programmers and analysts throughout the system's development process
- an understanding of Web systems requirements, design and prototyping approaches;

- an ability to produce key requirements and design documents describing how a specified system will be implemented;
- an appreciation of basic usability and Human Computer Interaction (HCI) issues

Student Learning Outcomes

After completing this course, students will:

- reinforce existing knowledge about the concepts and principles in the early stages of the software development life cycle
- experience with the development of project plans, brainstorming, requirement documents, prototyping techniques, issues and tasks management, peer reviews
- learn about the processes of converting requirements to design in a realistic context
- acquire practical design skills, particularly in architectural design and software component integration
- experience the process of implementing a prototype Web system by choosing appropriate languages, libraries and frameworks.
- acquire additional skills involved in working as part of a project team working within strict time constraints.
- learn the process of writing reports and documentation for specific needs.
- the recognition that production of quality software is a task demanding a disciplined approach to all stages of its development
- an appreciation of the many and varied issues involved in the development of software systems and the role and the importance that Software Engineering review processes play in producing quality systems
- develop an awareness of the community of engineering professions and the importance of keeping current through life-long learning and through interacting with that community. Students will also be encouraged to develop their research skills as one of the means of acquiring the necessary knowledge and skills to solve engineering problems

This course contributes to the development of the following graduate attributes:

Graduate Attribute	**Where Acquired**
the skills involved in scholarly enquiry	yes
an in-depth engagement with relevant disciplinary knowledge in its interdisciplinary context	yes
the capacity for analytical and critical thinking and for creative problem solving	yes
the ability to engage in independent and reflective learning	yes

the skills to locate, evaluate and use relevant information (Information Literacy)	yes
the capacity for enterprise, initiative and creativity	yes
an appreciation of and respect for, diversity	no
a capacity to contribute to, and work within, the international community	no
the skills required for collaborative and multidisciplinary work	yes
an appreciation of, and a responsiveness to, change	yes
a respect for ethical practice and social responsibility	no
the skills of effective communication	yes

Assumed Knowledge

Before commencing this course, students should have:

- Basic knowledge of the development process including requirements gathering
- The ability to design and implement general algorithms
- Basic knowledge of essential design concepts and techniques (equivalent to UML class diagrams and ER)
- Basic knowledge of scripting and Web technologies
- Writing and communication skills

These are assumed to have been acquired in the previous software engineering courses and workshops

Teaching Rationale

In this course, students will learn and apply generic skills including requirement elicitation techniques, the design review process, developing team skills, working creatively through group work and brainstorming and managing the varying levels of uncertainty involved in these processes. A primary goal of this course is to teach students the importance and process of group work. Students will learn that both human and technical views of design and implementation are equally important issues. In order to develop long-lasting and efficient software systems—in essence a quality product—Software Engineers need to be aware of and address both these factors in the development process. Less tangible outcomes are for students to also see that specifications are not the panacea of correctness, specification can be wrong and more importantly they can be subtly wrong. In addition, students will see first hand how assumptions and bias can sometimes be embodied in a specification unintentionally.

The workshop follows a product-based framework to the project-based learning. A set of intermediate deliverables leading to a product are specified by the stakeholder, a role assumed by the lecturer in charge. Some weekly lecture slots will be used to elaborate on the deliverables and answer general questions. During these meetings, teams are encouraged to discuss their progress and demonstrate work-in-progress. Teams can also arrange additional meetings with the stakeholder if required. A tutor will be available to assist with technical matters and answer queries related to the case study.

Teaching Strategies

Early weeks will consist of lectures; afterwards, all teams will meet weekly with their mentors. The Schedule specifies the activities for each week. Teams are offered the possibility to hold additional mentoring sessions if the need arises. Students can also ask for lectures on particular topics.

The Macquarie Second Year Software Engineering prize is awarded to one team from SENG2021 in a particular year. A number of teams usually three are chosen on the basis of their final demonstration and are asked to prepare a 20 to 30 minute presentation explaining their design and prototype implementation of the current project. The presentation is to be made to members of Macquarie Bank.

Assessment

The assessable components for the course are:

- Deliverable 1: Problem Statement/Features/Stories/Mockups: each group need to agree on a set of problems that they are going to solve in their project, defie the high level requirements (we refer to them as features), and express these high level requirements using a controlled language notation (refer to as user stories) where a user story captures features and scenarios. Then each group need to conduct both low fidelity prototype (White board or pen and paper sketching) and high-fidelity prototype (using Mockups tools, HTML CSS). The Prototypes need to illustrate the UI elements and how the UI elements are going to interact. (15%)
- Deliverable 2: Design Report: each Group needs to complete a preliminary design of their product, the technology stack they are going to use, architecture of the system, along with any modelling required (e.g., data model, class diagrams…etc.) (20%)
- Deliverable 3: Pitching Presentations: each group will present their early implementation of the product to their peers, illustrating their idea, why is it important, how they are implementing it, along with a demonstration of what they have (e.g., their Mockups)(15%)
- Deliverable 4: Final Report: each group need to submit a final report describing the whole project in detail (problem statement, design, and implementation) (20%)
- Deliverable 5: Final Demonstrations (30%): each group need to demonstrate a prototype of their product, elaborate on the salient features and reflect on their whole experience.

For more information on these deliverables, see the Course Web page. A detailed requirement for each deliverable will be released in accordance for each phase of the project.

Course Schedule (draft)

Week	Date	Description	Action
1		Lectures: Introduction to the Course, Guest Lectures	Specs available, Form groups
2		Mentoring Meeting (project ideas)	-
3		Mentoring Meeting	**Problem Statement/Features/Stories Del 1 Due**
4		Mentoring Meeting	
5		Mentoring Meeting	
6		Mentoring Meeting	**Design Report Del 2 Due-**
7		**Pitching Presentations (Del 3)**	-
8		Mentoring Meeting	-
9		Mentoring Meeting	-
10		**Final Demonstrations (Del 4)**	**Final Report Del 5 Due**
11		Macquarie Second Year Software Engineering Prize	TBC

Additional details and changes will be posted on the course's noticeboard.

Resources for Students

For domain knowledge, students are encouraged to research appropriate sources of information depending on their needs. They are also expected to learn about the basic concepts using Web data sources.

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

During the last CATEI evaluation, students have raised many issues related to the clarity of the specifications given. Although every effort is made to produce good specifications, students must appreciate that most workshops projects are open ended and leave room for innovation by students. Therefore, it is important that they seek information about the project requirements stakeholder on a continuous basis during mentoring sessions.