Web Application using Learning Tools to Teach Software Engineering

Mohammed Yaseen Ali
Computing for Business Applications wIE

Supervisor: Dr Kung-Kiu Lau
April 2013
Abstract

The subjective nature of Software Engineering has most certainly caused ambiguity amongst students. Although Software Engineering is an approach for developing software it embodies a range of different principles and methods. However, Software Engineering is much more than producing code, it entails a range of Unified Modelling Language (UML) artefacts to be produced as prerequisites of the system under analysis or development. After conducting primary research, students highlighted that they did not understand how to capture Use Cases and misunderstood the purpose of Activity Diagrams which are both housed part of the UML.

This report documents the process of developing a web application using learning tools to teach software engineering paying particular attention towards Use Case Diagrams and Activity Diagrams. The web application is to host interactive tutorials, exercises, games and videos to educate students studying the second year software engineering module at The University of Manchester.

The development process entailed background research exploiting how software engineering is currently being taught and primary research to define the problem scope. This is then followed up by a detailed design and implementation stage, explaining key challenges and difficulties. Finally the web application is tested and evaluated to demonstrate the robustness, completeness and quality.
Acknowledgment

First and foremost, I would like to thank my parents for allowing me to freely choose the path I tread and providing the support required to ensure I make a success.

I would like to thank Yunis Ali, Kholsuma Khatun and Asma Khatun for their advice during my time in education and proof reading my report.

Furthermore, I am very grateful to my supervisor Dr Kung-Kiu Lau for his continuous support and guidance throughout the project. His persistence and high expectations has led to me build a web application that was initially beyond my comprehension.

Finally, I would thank the students that have participated in the questionnaires and testing for my project and I am very pleased to say the quality of the project far exceeded my expectations.
Table of Contents

Abstract ................................................................................................................................. 2
Acknowledgment .................................................................................................................. 3
List of Figures ...................................................................................................................... 7
List of Tables ....................................................................................................................... 9
Chapter 1: Introduction ....................................................................................................... 10
  1.1 Problem Summary ......................................................................................................... 11
  1.2 Aims and objectives ...................................................................................................... 11
  1.3 Defined Milestone ........................................................................................................ 11
  1.4 Report Structure .......................................................................................................... 13
Chapter 2: Background ..................................................................................................... 14
  2.1 Chapter Overview ......................................................................................................... 14
  2.2 Learning Styles ........................................................................................................... 14
  2.3 Method of Assessments .............................................................................................. 14
  2.4 Software Engineering .................................................................................................. 15
  2.5 The Unified Modelling Language ............................................................................... 15
  2.6 Existing Web Applications ......................................................................................... 16
Chapter 3 Primary Research .............................................................................................. 19
  3.1 Chapter Overview ......................................................................................................... 19
  3.2 Student Research ......................................................................................................... 19
Chapter 4: Requirement Specification ............................................................................... 23
  4.1 Chapter Overview ......................................................................................................... 23
  4.2 Functional Requirements ............................................................................................ 23
  4.3 Non-Functional Requirements .................................................................................... 23
  4.4 Interface Design Principles ......................................................................................... 23
Chapter 5: Web Application Analysis ................................................................................ 25
  5.1 Chapter Overview ......................................................................................................... 25
  5.2 Software Development Methodology .......................................................................... 25
  5.3 Activity Diagram .......................................................................................................... 25
  5.4 Use Case Diagram ....................................................................................................... 26
  5.5 Hierarchical Task Analysis ........................................................................................ 28
### Chapter 6: Web Application Design

- 6.1 Chapter Overview .................................................. 30
- 6.2 Sitemap Outline View ............................................. 30
- 6.3 Web Template Design ............................................. 31
- 6.4 Interactive Tutorial ................................................ 31
  - 6.4.1 Use Case Diagram Page ..................................... 32
  - 6.4.2 Activity Diagram Page ..................................... 35
- 6.5 Workshop ............................................................ 36
- 6.6 Forum ............................................................... 39
- 6.7 Feedback Page ...................................................... 40

### Chapter 7: Web Application Implementation

- 7.1 Chapter Overview .................................................. 42
- 7.2 Laying down the foundation ..................................... 42
- 7.3 Login and Register Page ......................................... 42
- 7.3 Home Page .......................................................... 43
- 7.4 Interactive Tutorial Page ......................................... 43
  - 7.4.1 Use Case Diagram Page ..................................... 44
  - 7.4.2 Activity Diagram Page ..................................... 47
- 7.5 Workshop Page ...................................................... 49
- 7.6 Forum Page .......................................................... 51
- 7.6 Feedback Page ...................................................... 52
- 7.7 Additional Information and Features .......................... 53

### Chapter 8: Testing

- 8.1 Chapter Overview .................................................. 55
- 8.2 Unit Testing .......................................................... 55
- 8.3 Integration Testing .................................................. 55
- 8.4 System Testing ...................................................... 56
- 8.5 Acceptance Testing .................................................. 56
- 8.6 Manual Testing ...................................................... 56
- 8.7 Evaluating Usability and Learning Experience ............... 57
  - 8.7.1 Usability Results ............................................. 57
  - 8.7.2 Learning Experience Results ............................. 58
Chapter 9 Conclusion

9.1 The Project Objectives

9.2 Issues Encountered

9.3 Potential Future Development

9.4 Personal Achievement

9.5 Closing Comment

Works Cited

Appendices

Appendix A: Questionnaire

Appendix B: Analogy Used

Appendix C: Results

Appendix D: Feedback Questionnaire
List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UML Hierarchical Structure</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>Codecademy Exercise Example</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Code School Exercise</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Topic Students Found Difficult</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>Types of Learning Process</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>Preferred Method of Communication</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Overview Activity Diagram</td>
<td>26</td>
</tr>
<tr>
<td>8</td>
<td>Overview Use Case Diagram</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>Hierarchical Task Analysis</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>ER Diagram</td>
<td>29</td>
</tr>
<tr>
<td>11</td>
<td>Sitemap Overview</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>Template</td>
<td>31</td>
</tr>
<tr>
<td>13</td>
<td>Use Case Page Layout Example</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>Draw Use Case Exercise</td>
<td>33</td>
</tr>
<tr>
<td>15</td>
<td>Key Notation Exercise</td>
<td>33</td>
</tr>
<tr>
<td>16</td>
<td>Capturing Use Case Exercise</td>
<td>34</td>
</tr>
<tr>
<td>17</td>
<td>Hangman Game</td>
<td>34</td>
</tr>
<tr>
<td>18</td>
<td>Word Search Game</td>
<td>35</td>
</tr>
<tr>
<td>19</td>
<td>MCQ Game</td>
<td>37</td>
</tr>
<tr>
<td>20</td>
<td>Fill-in-the-Blanks Game</td>
<td>37</td>
</tr>
<tr>
<td>21</td>
<td>Use Case Spot the Error Game</td>
<td>38</td>
</tr>
<tr>
<td>22</td>
<td>Activity Central Game &amp; Results</td>
<td>38</td>
</tr>
<tr>
<td>23</td>
<td>Forum Category Page</td>
<td>39</td>
</tr>
<tr>
<td>24</td>
<td>Forum Topic Page &amp; Modifications</td>
<td>40</td>
</tr>
<tr>
<td>25</td>
<td>Feedback Graph</td>
<td>40</td>
</tr>
<tr>
<td>26</td>
<td>Start and End Point of the Application</td>
<td>43</td>
</tr>
<tr>
<td>27</td>
<td>Flipboard in Flip Motion</td>
<td>44</td>
</tr>
<tr>
<td>28</td>
<td>Flash File Containing ActionListeners</td>
<td>45</td>
</tr>
<tr>
<td>29</td>
<td>Switching Between Scenes</td>
<td>45</td>
</tr>
<tr>
<td>30</td>
<td>Hangman Image Development</td>
<td>46</td>
</tr>
<tr>
<td>31</td>
<td>Text Field Initialisation</td>
<td>47</td>
</tr>
</tbody>
</table>
List of Tables

Table 1: List of Functional Requirement .......................................................................................... 23
Table 2: List of Non-Functional Requirement ................................................................................. 23
Table 3: Nielson Heuristics & Schneiderman Eight Golden Rules .................................................. 24
Table 4: Design Influences ............................................................................................................. 24
Table 5: Algorithm 1 - CheckInputKey ............................................................................................. 35
Table 6: Algorithm 2 - CheckWord .................................................................................................. 36
Table 7: Algorithm 3 - StoreScoreArray ............................................................................................ 41
Table 8: Structure of the Test Table .................................................................................................. 55
Chapter 1: Introduction

The World Wide Web continues to thrive and has become a new means of engaging students beyond the classroom. It provides a student with a rich learning experience utilising emerging new technologies all at the touch of a button. The advancement of technology has commanded the need to teach disciplines such as Software Engineering dynamically and with precision. Publishing an online web application using learning tools to teach Software Engineering could prove to be a powerful tool.

Furthermore, contribution from all students can be dismal in these ever increasing class sizes. Structuring and teaching an engaging lesson can be moderately challenging. A high level of abstraction of the syllabus is required, which in turn eliminates in-depth aspects of the discipline. However, a web application can be technical and complex as each page is engineered to interact with a user’s request and respond by delivering the desired demands.

The proposition put forth in this paper presents a service that aims to resolve problems encountered by students that studied the 2nd Software Engineering module (COMP23420).

The purpose of this project is to design and develop a web application. It is intended to promote and fully engage students in the teaching and learning of Software Engineering; typically focusing on Use Case Diagram and Activity Diagram, which are housed part of the Unified Modelling Language. Students should be able to perform tasks to enhance understanding and review performance in a logical and clear format. The web application should provide a one-stop introduction and offer a multitude of services to a student.

This report documents the achievement of the online Web Application using Learning Tools to Teach Software Engineering, by describing how the project was initiated and leading onto the designs, implementation and testing.
1.1 Problem Summary
A questionnaire given to third year students revealed two areas of concern about their 2nd year Software Engineering module:

- **Use Case Diagrams** - students are creating one too many use cases which resulted in describing functional entitlement of the system as a whole.
- **Activity Diagrams** - It was highlighted students were penalised when developing Activity Diagram as they modelled how the final software would actually execute as oppose to providing the overview of the flow of activity related to a specific scenario.

Both diagrams had two common problems for students, these are:

- Students had difficulties in understanding how to capture use cases and activities, and
- How they are relevant when developing a software application and/or writing code.

This now provides an opportunity for a web application to be developed which will aid and support the students understanding of these diagrams.

1.2 Aims and objectives
**Aim 1**: Enrich a student’s understanding of Use Case Diagrams and Activity Diagrams

- Objective 1: Develop interactive tutorials to provide comprehensive information to support students understanding
- Objective 2: Create a discussion forum in which students can ask questions and receive answers
- Objective 3: Receive feedback after completion of a task identifying areas of improvements

**Aim 2**: Promote students participation when engaging with the web application

- Objective 4: Develop interactive exercises to allow students to participate and test their subject knowledge
- Objective 5: Develop interactive games to allows students to learn and participate fully with information
- Objective 6: Include text, images, videos and sound to promote multisensory learning
- Objective 7: Include text, images, videos and sound to promote multisensory learning

**Aim 3**: To provide an environment in which students can learn easily

- Objective 8: All information should be accessible with sensible headings
- Objective 9: Good use of colour, contrast, font style and size

1.3 Defined Milestone
In order to manage the complexity of the project it requires some well-defined milestones:

- Research, identify and analyse students concerns.
- Design Graphical User Interface to exhibit all desired features.
- Design appropriate software which provides various features to aid learning.
- Implement the chosen software system.
• Test, evaluate and analyse student’s responses to the developed web application.

The project will be deemed successful under the strict conditions the web application implements all features and objectives described thoroughly in section 1.2 and in Chapter 4 - Requirement Specification of this thesis.

Under the circumstance where all functionalities specified are implemented and depending on the extent the website demonstrates its capabilities in satisfying these demands, the project will be considered a partial success.
1.4 Report Structure

**Chapter 1 – Introduction:**
This chapter starts with a brief explanation and the context of the project. It then provides a summary of the problem, the aims and objectives which is immediately followed up with Defined Milestone.

**Chapter 2 – Background:**
This chapter presents the background information that is necessary to understand what the project is aiming to achieve. It explores the different types of learning styles and methods of assessing student’s achievements. Next, it goes onto explaining how the Software Engineering module is currently being taught and the importance of the Unified Modelling Language. Finally, it evaluates current web applications.

**Chapter 3 – Primary Research:**
This chapter collates the results from the questionnaire conducted to highlight the issue and define the scope of the project.

**Chapter 4 – Requirement Specification:**
This chapter states the Functional and Non-Functional requirements of the purposed system. It goes on to explain the importance of adhering to good design principles with respect to designing and developing a Graphical User Interface followed by defining some optional requirements.

**Chapter 5 – Analysis:**
This chapter explores the backend of the system. This is achieved by adopting a software development methodology and designing the following artefacts: Activity Diagram, Use Case Diagram, Hierarchical Task Analysis and ER Diagram.

**Chapter 6 – Web Application Design:**
This chapter explores the frontend of the system by introducing an overview site map of the web application which instantly goes into uncovering some of the key features of the web application and discusses the algorithms used.

**Chapter 7 – Implementation:**
This chapter reviews and justifies the tools that were used in developing the web application and goes onto explaining the implementation process of the exercises, games and algorithms.

**Chapter 8 – Testing:**
This chapter elaborates on the importance of testing the web application and highlights the different methods of testing conducted. This was followed up by the usability and learning experience of the application.

**Chapter 9 – Conclusion:**
This chapter discusses to what degree the web application achieves the aims and objectives. It highlights the issues that affected the project and potential future development.
Chapter 2: Background

2.1 Chapter Overview
The goal of this chapter is to research the different types of learning styles and assessment methods. Also, it focuses on understanding the purpose of Software Engineering and how it is currently being taught at university. Furthermore, it goes onto investigating the importance of The Unified Modelling Language. In addition, emphasis is placed on defining what e-learning is and how others have implemented such systems.

2.2 Learning Styles
Before starting to develop a web application it is important to acknowledge and explore the different types of learning styles.

“Learning styles as the composite of characteristic cognitive, affective, and physiological factors that serve as relatively stable indicators of how a learner perceives, interacts with, and responds to the learning environment. [1]” it is said that there are typically three simplified list of learning styles which are;

1. **Auditory** – this method of learning is traditionally executed by delivering the desired information in auditory formats such as audio recording, lectures and discussions best suited for classroom students where professors lecture.
2. **Visual** – a student with a visual learning style tends to acquire knowledge from hand outs, presentations, books, articles, webpages, images, video and diagrams.
3. **Tactile** – this is best suited for a student when information is convey in a hands-on approach such as workshops, labs and best respond well to touching and creating things.

Creating teaching tools utilising the above learning styles will help to optimise learning and effectively reach and appeal to a wider audience.

2.3 Method of Assessments
An important factor to consider when developing a web application is, understanding how to assess a student’s achievements.

“The primary goal is to choose a method which most effectively assesses the objectives of the unit of study. In addition, choice of assessment methods should be aligned with the overall aims of the program, and may include the development of disciplinary skills and support the development of vocational competencies [2].”

There are eight broad categories listed below:

1. **Thinking critically and making judgements** – this is achieved by students writing essay, reports, developing arguments and evaluating.
2. **Solving problems and developing plans** – this tends to require identifying problems, analysing data and applying information.
Performing procedures and demonstrating techniques – this can be done by computation, using equipment, following protocols and carrying out instructions.

Managing and developing oneself – student may need to work independently, managing their time, learning independently and be organised.

Accessing and managing information – information can be obtained by researching, collecting data, organising information and managing information source.

Demonstrating knowledge and understanding – students can demonstrate by recalling, describing, recognising and relating knowledge to show understanding.

Designing, creating and performing – this is achieved by imagining, designing, innovating and performing.

Communicating – students can communicate one and two ways, within a group, by describing, by presenting and advocating.

A web application can employ various assessment methodologies and tools; to achieve different aims: to support and stretch the student’s analytical skills and thought process, demand logical reasoning and organising ones learning, be a means of communication and development of ideas and finally to measure the progress and achievement of a student.

2.4 Software Engineering

Software Engineering highlights the importance of attention to detail and the significance of using appropriate tools and techniques to deliver the required functionality and performance to the end user. The 2nd Year Software Engineering Module stresses software engineering is much more than producing code: it takes into consideration stakeholder’s budget, deadline, demands and the expectation of software to offer dependability, reliability and consistency (easy to use) in a seamless manner.

A software system can be regarded as the cornerstone of any business; under the conditions such systems are robust, long lived and adaptable in turn, can potentially lead to complex algorithmics. The Software Engineering module teaches students how to apply the principles of requirement gathering and analysis, system design, implementation, testing and maintenance of a software system. Further details about the importance of adopting a Software Engineering Methodology is explained in Chapter 5 Section 5.2.

At Manchester University the School of Computer Science have made Software Engineering a mandatory module for second year students, in response to the fundamental problem of providing students with a practical learning environment.

2.5 The Unified Modelling Language

Within the scope of the Software Engineering module students are taught the importance of developing UML diagrams also known as artefacts to better aid understanding of project before coding. UML is a set of graphical notation for modelling used primarily to provide visual aid to software developers by specifying, constructing and documenting the artefacts of an object oriented software system. UML is referred to as the “de facto standard for modelling object oriented system [3]”. Drawing and developing these graphical notations can be used to illustrate the proposed design and the reasoning about particular system behaviour by documenting key elements to others and be used for future reference by the team.
UML provides a rich variety of diagrams types that can be classified as two major categories such as Structural and Behavioural diagrams which can be further divided into sub-categories (Figure 1):

Structural diagrams aim to model the static aspect of a system such as the physical components, attributes, classes, objects, interfaces, operations, relationship and dependencies between elements.

Behavioural diagrams aim to model the dynamic behaviour of systems overtime, by placing emphasize on the varieties of interaction, showing collaborations between objects and internal instantaneous state change of objects.

The scope of this project is to focus on Use Case Diagram and Activity Diagram which are part of the Behaviour Diagrams. Below is a description of both diagrams.

**Use Case Diagram**

A Use Case Diagram consists of use cases used to visualise and describe high level system functionalities and depicts different types of actors that intend to interact with the system. The purpose of developing a Use Case Diagram is to describe what a system should do, as oppose to describing the process of how it is done.

**Activity Diagram**

A UML Activity Diagram is used to understand the workflow and sequential activity involved in a process typically used for business processes, operational step by step workflows, and complex algorithms in a system component.

**2.6 Existing Web Applications**

An e-learning tool can be used for pedagogical purposes, which is a method and practice of teaching academic subjects. Unlike traditional class room teaching methods, e-learning is a new improved method of delivering desired knowledge and skills using web based technology for students. It can use a range of different media to implement the learning process and provide highly interactive games, step by steps explanations, indexed instructions and guidance through the use of a knowledge database. Unfortunately, there are a few web applications that specifically teach UML...
which mainly contain laborious reams of pages explaining the concepts with images. Nevertheless, below are two examples of teaching tools currently aiming to teach different aspects of Computer Science involving user participation.

**Codecademy**

Codecademy [4] is an online interactive website that teaches a series of programming languages to anyone interested, this includes HTML, CSS, JavaScript, jQuery, Python, Java and much more. These specific topic areas are taught incrementally though a wide range of exercises, allowing users to read a specific scenario, write code and review results on screen. To ensure learners are fully engaged and motivated the website provides feedback, progress bar, badges depending on how well the exercise was completed and provide functions to share user profile with colleagues. The website encourages the user to ask questions and receive answers on a publicly shared forum. Another interesting aspect of the website is that it facilitates and inspires anyone to build their own course quickly and easily (Figure 2).

![Figure 2: Codecademy Exercise Example](image1)

**Code School**

Code School [5] again is very similar to Codecademy with respect to its teaching methods. However, Code School provide different level of difficulties using games and coding exercises that are designed to make new technology, screencast that provides audio for further assistance, easy and flexible coding browser. Furthermore, as learner’s progress newer challenges are unlocked followed by 10-15 minutes videos clip aiming to assist the learner in learning the new subject. To keep users inspired and motivated after the completion of each task, badges are provided and peer to peer learning is encouraged which is similar to Codecademy. However, unlike Codecademy once a learner has completed the entire course a prize is given to the student with the most optimised code for each exercise (Figure 3).

![Figure 3: Code School Exercise](image2)
Identifying and understanding existing web applications aimed to teach similar aspects of software development provided useful insight. They demonstrate how they attempt to engage users though carefully developed interactive exercises and games with videos to enhance students understanding of a particular subject area.

Interestingly, there was an incremental teaching development process that took place in Code School. It allowed students to progress upon completion of a current task unlike Codecademy.

Most importantly both web applications demonstrated strong capabilities of teaching students in a virtual learning environment which was interesting, fun and interactive.
Chapter 3 Primary Research

3.1 Chapter Overview
The goal of this chapter is to demonstrate and discuss the results obtained from the primary research conducted using a questionnaire targeted at third year students. The key findings will be used to build the baseline information for the project.

3.2 Student Research
For data collection purposes a questionnaire was devised (Appendix A). The results were used to define the scope of the project and identify areas students had difficulty with.

The questionnaire was distributed to ten third year students all of whom had completed the subject. However, before issuing the questionnaire it was reviewed by a pilot group consisting of three students who verified validity of each question, identified any obvious errors and reliability before handing out.

The questionnaire consisted of four carefully engineered questions containing two different format types which are:

- Open Format Questions – allowing students to freely express thoughts and opinions without predetermined responses.
- Closed Format Questions – restricting a student options with closed ended questions meant determined response.

The questionnaire started by asking students which UML category they found most difficult, Structural diagrams or Behavioural diagrams. All ten students selected Behavioural diagrams. Two immediate sub questions followed.

The first sub question was: “If you have selected Behavioural Diagrams, please select which topic(s) you found difficult” results are shown in Figure 4.

![Figure 4: Topic Students Found Difficult](image-url)
Student’s identified three out of six topics difficult to understand. 50% of students expressed concerns with Use Case Diagram. 40% of students indicated Activity Diagram and the remaining 10% of student highlighted Sequence diagram.

The Second sub question was “What did you find difficult about the following diagram(s)?” Below are samples of three student responses that had a common recurring theme:

**Student One:** “When I received my feedback on I think Moodle or Blackboard the tutor wrote I created one too many Use Cases. With respect to Activity Diagram I lost marks because I modelled the whole system, like I understand the basic concepts of both diagram but need somewhere I can practice. Apparently I didn’t capture all the right use cases.”

**Student Two:** “Once I looked at my marks I was shocked to see how low it was only because I thought I did the Use Case Diagram right because I understood what it is and considering my diagram corresponded closely to the lecture note. Also on my Activity Diagram I lost marks only because I really don’t understand the purpose of it, I mean how does it relate to writing code or the system design?”

**Student Three:** “I lost marks on my Use Case Diagram because the name of my actors where inconsistent and Activity Diagrams I found rather difficult and lost marks because I modelled the entire scenario and system. Oh yeah, I didn’t define the system boundary on my Use Case Diagram I think”.

The second question asked to student was: “From the list below which method(s) do you think would aid your learning process?” The results are shown in Figure 5.

![Figure 5: Types of Learning Process](image)

Student’s highlighted four methods they thought would aid their learning process. All students agreed that Exercises are a great way to learn new material. 80% of student expressed interest in Games as a means of learning. 60% of students selected Tutorials as a method of learning. 40% of students opted for Videos as an effective method of learning.

The third question asked to student was: “If you don’t understand what is being said during lecture hours, do you ask questions in the lecture?” All ten students answered ‘No’. Two immediate sub questions followed.
The first sub question was: “if answered no why not?” below are five example of student response:

**Student One:** “English isn’t my first language so I don’t want to sound stupid”

**Student Two:** “Some of my questions are rather basic and I don’t want to ask them because I should know them”

**Student Three:** “Am sure like most students I find it embarrassing, what if my question sounds silly”

**Student Four:** “I never raise my hands in lectures and I find it hard to think of questions on the spot am not good like that”

**Student Five:** “I hate it when people ask questions only cause it holding everyone else back so I wouldn’t do it myself for that reason. Plus if I have questions I would much rather look it up on the internet”.

The second sub question was: “would you feel more comfortable asking questions using any of the methods listed below?” The results are shown in Figure 6.

Students were presented with different methods of communication when in doubt about learning material. 60% would prefer to ask questions on a forum. 30% of students prefer directly emailing the respective tutor. The remaining 10% of students would prefer post questions on a message board.

The final question was: “Is there anything you would like to suggest” Surprisingly, this question prompted a response from all of the students who suggested a range of different ideas. Below are five examples of student suggestions:

**Student One:** “I think there should be a website that has an online book to teach students interactively”

**Student Two:** “I think it would be really useful to have exercises that test your knowledge on Use Case and Activity Diagrams”

**Student Three:** “I need a place where I can ask questions if they get stuck”
Student Four: “There should be an automated tutorial that teaches student how to construct an Activity Diagram”

Student Five: “There should be games and exercises that students can complete with feedback of their performance”

Summary Statement

The questionnaire provided useful insight on what topics the application will need to address, in this circumstance Use Case Diagram and Activity Diagram. It was highlighted students lost marks due to creating one too many use cases, not defining the system boundary and incorrect names for the actors in the Use Case Diagrams. It was also stated some of the students did not understand the purpose of these diagrams and how they are related to writing code or the system design.

By asking students to identify the methods which best aided their learning gave rise to a fount of ideas that could be used as types of learning tools. A majority of the students were strongly in favour of exercises, games, tutorials and videos. These suggestions will need to be developed and accommodated in the proposed application.

It was interesting to note that all ten students stated they do not ask questions during lectures. This was due to either English not being their first language, feeling embarrassed, sounding silly in front the class and/or feeling they will be holding the class up. It was stated that the preferred method of communication would be either a forum or directly asking a question to the tutor via email. Considering 60% of votes are inclined towards a forum this will need to be addressed in the proposed application.

The final question provided an insight of what students would like to see in the proposed system such as an online interactive book, exercises to test their knowledge, a place where they can ask questions, tutorials explaining how to construct the Use Case and Activity Diagram, and feedback on their performance.

All four questions provided in-depth analysis of the type of functionalities that will need to be included in the proposed system. To integrate all of the suggested functionalities is beyond the scope of developing a learning tool. But, a web application utilising learning tools to teach software engineering would be better suited.

Overall the questionnaire was a success, providing much needed information to build an exciting and engaging web application.
Chapter 4: Requirement Specification

4.1 Chapter Overview
During this chapter the aim is derive the Functional and Non-Functional Requirements of the proposed web application. In an attempt to implement these requirements effectively, specific key principles must be adhered too which are included in this chapter.

4.2 Functional Requirements
Firstly, from the research conducted it is important to derive the Functional Requirements (FR) of the system. They are used to describe “what the system should do or provide for users. They include description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system [6]”. The table below (Table 1) identifies all the desired behaviour of the system including the priorities (ranking high to low) and risk evaluation which is based on project objectives.

<table>
<thead>
<tr>
<th>No</th>
<th>Description</th>
<th>Priority</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR1</td>
<td>Student must be able to login into the system</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>FR2</td>
<td>Student must be able to measure their progress</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>FR3</td>
<td>Student must receive task feedback</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>FR4</td>
<td>Student must be able to ask questions</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>FR5</td>
<td>Student must be able to interact with system as a whole</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>FR6</td>
<td>Implement interactive tutorials for students</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>FR7</td>
<td>Student must be able to perform exercises and games</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>FR8</td>
<td>Students must be able to make notes</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 1: List of Functional Requirement

4.3 Non-Functional Requirements
Next, the Non-Functional Requirements (NFR) identified, define the “constraints of the project. They constrain the designer when deciding how the project will deliver the functional requirements [7]”. Attempting to assign priority and risk ranking to Non-Functional Requirement (Table 2) is implausible. However, there are guidelines such as FURPS principle to enable good coding practice. The last three NFRs implemented in this system are part of the FURPS attributes.

<table>
<thead>
<tr>
<th>No</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFR1</td>
<td>The web application should be highly interactive and engaging</td>
</tr>
<tr>
<td>NFR2</td>
<td>The web application should be compatible with different web browsers and devices</td>
</tr>
<tr>
<td>NFR3</td>
<td>The database should be MySQL compatible</td>
</tr>
<tr>
<td>NFR4</td>
<td>Usability -The web application ought be easy to use by ensuring good use of HCI</td>
</tr>
<tr>
<td>NFR5</td>
<td>Reliability - The web application ought to perform its required functions and prone to errors</td>
</tr>
<tr>
<td>NFR6</td>
<td>Performance – The Web application response time should be reasonable</td>
</tr>
</tbody>
</table>

Table 2: List of Non-Functional Requirement

4.4 Interface Design Principles
Now that the underlying requirement of the system have been established, it is crucial to ensure all functionalities are implemented into the web application. Developing a web application would require designing a graphical user interface (GUI) which provides the visual components that uses a
combination of technologies and devices for users to interact with. Taking into account the project has been aimed at students: the web application has been embraced as an inexpensive method of communication to the mass or a focused audience which is traditionally accessed over a network via a web client typically the internet due to the ubiquity of web browsers.

It is important to develop a well-designed GUI as it can determine the acceptance or rejection of a software system. User interface design is also known as Human Computer Interact (HCI) which is a method that accommodates the interactive relation between a user and computer system. To optimise user experience there are two main principles which can are considered called Nielson’s Heuristics (NH) and Schneider’s eight golden rules (SR) listed in Table 3.

<table>
<thead>
<tr>
<th>#No</th>
<th>NH</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Visibility of system status</td>
<td>Strive for consistency</td>
</tr>
<tr>
<td>2</td>
<td>Match between system and the real world</td>
<td>Enable frequent users to use shortcuts</td>
</tr>
<tr>
<td>3</td>
<td>User control and freedom</td>
<td>Offer informative feedback</td>
</tr>
<tr>
<td>4</td>
<td>Consistency and standards</td>
<td>Design dialog to yield closure</td>
</tr>
<tr>
<td>5</td>
<td>Error prevention</td>
<td>Offer simple error handling</td>
</tr>
<tr>
<td>6</td>
<td>Aesthetic and minimalist design</td>
<td>Permit easy reversal of actions</td>
</tr>
<tr>
<td>7</td>
<td>Flexibility and efficiency of use</td>
<td>Support internal locus of control</td>
</tr>
<tr>
<td>8</td>
<td>Recognition rather than recall</td>
<td>Reduce short-term memory load</td>
</tr>
<tr>
<td>9</td>
<td>Help users recognize, diagnose, and recover from errors</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Help and documentation</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Nielson Heuristics & Schneiderman Eight Golden Rules

To further aid the design process of the GUI, research conducted by The University of Victoria was based on the usability of web-based learning tools. Fifty-four students participated in the study to analyse the navigation and tool design aspects between WebCT and Blackboard. “Students’ commented that the tools were positive to use, when they....” had the 7 optional requirements below (Table 4) [8]

<table>
<thead>
<tr>
<th>#No</th>
<th>Optional Requirements (OR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Well-designed, easy to learn, easy to use</td>
</tr>
<tr>
<td>2</td>
<td>Simple to navigate and have a well-designed layout</td>
</tr>
<tr>
<td>3</td>
<td>Compatible with other platforms and programs</td>
</tr>
<tr>
<td>4</td>
<td>Accessible from all places outside of the university</td>
</tr>
<tr>
<td>5</td>
<td>Transparent (tool does not hinder, frustrate the user)</td>
</tr>
<tr>
<td>6</td>
<td>Used as up to date support for the course, not as a replacement of lectures</td>
</tr>
<tr>
<td>7</td>
<td>Relevant to the course and tied into the specific course structure and content</td>
</tr>
</tbody>
</table>

Table 4: Design Influences

In order to design and implement a good GUI, all of the attributes listed above for each principle will need to be given close consideration. Chapter 6 goes onto explain how some of these requirements and principles have been applied.
Chapter 5: Web Application Analysis

5.1 Chapter Overview
In this chapter the aim is to analyse the technical aspect of the project which is referred to as the ‘backend of the system’. This will include formally describing and justifying the purpose of a Software Development Methodology and introducing some key UML design artefacts to better aid understanding of the design decisions for the proposed system.

5.2 Software Development Methodology
Software methodologies have continuously evolved since the traditional waterfall model. Adopting a suitable methodology can be difficult as it has to take into account a number of different and complex factors.

The Incremental and Iterative Development (IID) approach was adopted as the development methodology for the construction of this project. This approach “was developed in response to the weaknesses of the waterfall methodology. It starts with an initial planning and ends with deployment with the cyclic interactions in between [9]”.

Due to the aims and objectives of this project, there was a need for referring back to students in order to measure the success of the project. The IID allowed the development of the web application to go through repeated cycles referred to as iterations. The iterative aspect to the methodology enabled us to “repeat the cycle of design, build and test until the desired functionality [10]” was achieved. During which time smaller portions of the web application was designed, implemented and tested which is referred to as incremental. The incremental aspect to the methodology allowed additional functionalities to be implemented at the end of each stage.

Ultimately the reason for choosing IID was because of the number of high risk FRs outweighed the number of low and medium risk requirements. This was beneficial as the iterative development aspect permitted “early rather than late mitigation of high risks (technical, requirements, objectives and usability) [10]” to be implemented which is known as Risk Driven Development (RDD). Throughout the duration of the project rapid visibility of progress was witnessed due to “early feedback, user engagement, and adaptation, leading to a refined system that more closely meets the real needs of the [10]” students.

5.3 Activity Diagram
The UML Activity Diagram in Figure 7 is aimed to illustrate the sequential activities involved in the web application developed. It is used to demonstrate the overall workflow and concentrate on the crucial functional areas of the application.

To start with students must create a user account permitting them to authenticate to system services, during which point the users credential will either be authorised or declined due to either duplicated user information or invalid email address; in both cases, an error message is displayed and the user will have to validate their details before proceeding.

Under the circumstance the student details are validated they can browse the website. Ideally, if students have logged in for the very first time they will be strongly advised to follow a sequential
order, starting with reading the interactive tutorial which will contain a series of exercises and tasks that will have to be completed.

At this point, if students have understood and comprehended the tutorials provided, they could select and perform games which have been specifically engineered based on the tutorials and lecture slides. There will be four games developed in total. The first two games are generic aimed to test the students overall knowledge of the topics. The remaining two games are specifically based on each topic. Once each game has been completed students will be awarded marks which will be viewable on the results page.

If students at any given time feels the need to ask questions for the simple reasons they did not understand the content in the tutorial or got stuck on an exercise they can freely choose to ask a question on the forum. Once a reply has been confirmed they can view the result.

Finally, once students are pleased with the results and feedback provided they have two options either print or save the results.

![Figure 7: Overview Activity Diagram](image)

5.4 Use Case Diagram

It is important to understand how students will be interacting with the functionalities of the web application being developed. In the web application there are two actors called Student and Database (system actor) which can be best illustrated using a UML Use Case Diagram (UCD) shown below in Figure 8. The Functional requirement listed above will correspond to the UCD which also provides the scope of the web application.
The UCD focuses on six key roles students can perform. Firstly, students can ‘Create Account’ by providing the following: name, password and email address which will then be stored into a Database.

Next, students can ‘Read and Interact with Tutorial’ which will consist of simple visuals and exercises to help aid learning. It is important to note upon completion of exercise no data will be saved into the Database.

Furthermore, students can ‘Select and Perform Game Exercises’ providing four different games to select from. Upon the completion of each game the function will then allow the ‘Results’ to be uploaded into the Database and later be reviewed on ‘Review Individual Performance’.

Moreover, students are entitled to ‘Read or Ask Questions’ on the forum. This will allow students to view posts or ‘Add’, ‘Edit’ and ‘Delete’ old and new posts. If students choose to make any modifications to a post this will be saved into the Database.

In addition, throughout the application students will be entitled to make their own individual notes at any given time which will be saved onto the system cache. The students can choose to ‘Edit’ existing notes or simply ‘Delete’ unwanted note.
Finally, students will be able to ‘Review Individual Performance’ by analysing their marks they received for each game they played. This means all information will be retrieved from the Database and students can either ‘Print’ or ‘Save’ the feedback.

5.5 Hierarchical Task Analysis

For this web application a Hierarchical Task Analysis (HTA) was developed to capture how the tasks should be performed in a systematic and structural fashion. The HTA decomposes the high level tasks in the web application into a hierarchy of subtasks. Thus in turn, provides an insight into the standard flow of events and reveals the core functionality behind the software application. Figure 9 is a visual presentation of the potential states the system could operate under. The system captures the process in which a student login, the process of navigation and eventual logout.

5.6 Database Design

An Entity Relationship Model (ER model) has been developed for the web application as information will need to be stored in a Database shown in Figure 10. For this particular system five tables have been created which are called Users, Results, Categories, Topics and Posts.

The Users table was created to save students detail. The students will need to register with the website and the Database will then automatically assign students with an ID number which is set as the primary key. Finally, the cardinality between Student and Account is 1 to 1 which means students may only have one account. “Cardinality is the number of instances of one entity that can, or must, be associated with each instance of another entity [11]”.

Next, students can play games and the scores received are saved into the Results table. The Results table consisted of the following attributes: ID which is set as a primary key, User_ID set as a foreign key, Attempts_One, Attempts_Two and Attempts_Three. The aim of a foreign key is to cross reference the Users table, enabling the scores to be allocated to each individual student. Finally, the cardinality between Student and Results is 1 to 1…4 which means the Results Table will save between 1 to 4 game results.

The following three tables called Categories, Topics and Posts are related to the Forum. It consists of three predefined categories students could view called Use Case Diagram, Activity Diagram and
Web Application using Learning Tools to Teach Software Engineering

Others. Considering the system is targeted to teach Use Case Diagram and Activity Diagram it is only natural to include them. However, the reason for including Others was to enable students to freely ask questions within the scope of Software Engineering but outside the boundary of both Use Case and Activity Diagrams. The Categories table consisted of the following attributes: ID which is set as the primary key, Category_Title, Category_Description, Last_Post_Date and Last_User_Posted. Finally, the cardinality between Student and Category is 1 to 3.

Once students have selected which category they will like to view, they will have to decide whether to ask a question on an existing Topic or create a new Topic. The Topic table had the following attributes: ID set as the primary key, Category_ID set as the foreign key, Topic_Title, Topic_Creator, Topic_Views, Topic_Date, Topic_Last_User and Topic_Reply_Date. Finally, the cardinality between Categories and Topic is 1 to N, which means a category can consist of a multiple number of topics.

In order to ask a question or reply to an existing question students must create a post. The final table is called Posts which consisted of the following attributes: ID set as the primary key, User_ID set as the foreign key, Category_ID set as the foreign key, Topic_ID set as the foreign key, Post_Creator, Post_Content and Post_Date. The cardinality between Topics and Posts is 1 to N meaning a topic can consist of multiple posts.

Figure 10: ER Diagram
Chapter 6: Web Application Design

6.1 Chapter Overview
Here the aim is to focus on the ‘frontend of the system’ starting by describing the Sitemap and the hierarchical structure of the website. Next, the design of the website is explained by defining the purpose and reasoning about particular object positioning which is best demonstrated using a webpage template. Finally, the website is then explored by unfolding what each webpage will contain and justified by referring back to Chapter 4 Requirement Specification.

6.2 Sitemap Outline View
This Sitemap shown in Figure 11 illustrates a linear understanding of all the webpages that are part of the web application. It represents the hierarchical structure and provides an understanding of how students navigate through the website. Following up from the Activity Diagram in Chapter 5 this sitemap provides further clarity to the user navigation aspect of the web application as well as defining the names of each webpage.

![Figure 11: Sitemap Overview](image)

Firstly, students enter the website at Level 0 placing them on the Register Page called register.php. If under the condition students have registered or already have an existing account they will remain on Level 0 but be forwarded to the login.php page which is Level 0.1 in the structure. Once students have login successfully they will then be elevated onto the next Level. However, under the condition students fail to login they will be reverted back to the Register Page.

Next, upon authorisation students are placed on Level 1 the Home Page called index.php, during which point they will have the option to navigate to the following pages: Interactive Tutorial Page called tutorial.php, Workshop Page called workshop.php, Forum Page called forum.php and Feedback Page called feedback.php. As students drill down the structure they will always have the option to revert back to the previous level with a click of a button fulfilling SR6.
Finally, students will not be able to reach Level 4.1 in the structure without formally going through Level 4.0 which is the Forum Page. The subpages of Level 4.0 are: Category Page called form_category.php, Topic Page called form_topic.php and Create Topic Page called forum_create_topic.php.

6.3 Web Template Design
The web application adopted a universal webpage template. This was used as the foundation of the website as it offered a generic and reusable solution. Designing the template required implementing the specified requirements listed in Chapter 4. The basic design of the template is illustrated in Figure 12 which aimed to fulfil SR1 and NH4 as each webpage layout in the website was consistent and standardised.

The webpage template was split into three separate components as placement of information is an important factor to consider. This can affect how well a person absorbs information. The title of the web application was placed on top as this denoted the purpose of the website into key words. Below the title was the navigation bar which was designed as a list format, starting from left to right. Considering the navigation bar was the main point of control to access all regions of the web application all items remained consistent fulfilling SR2.

In addition, below the navigation bar was the subtitle and the body. The purpose of a subtitle was “to break the information into section’s related by topic [12]”, this allowed students to always be informed about which part of the application they are currently situation on.

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation Bar</td>
</tr>
<tr>
<td>Subtitle &amp; Body</td>
</tr>
</tbody>
</table>

Figure 12: Template

There are multiple reasons for the simplistic layout, as this aimed to achieve NH6, NH8 and OR2. Firstly, dividing the webpage template into three sections meant it was easy to maintain and aesthetically it was pleasing. The design of the website remained minimalistic enabling students to become familiar with the layout right away; this automatically triggers the principle of recognition rather than recall.

6.4 Interactive Tutorial
The most important aspect of the project was to ensure that the Interactive Tutorial was interactive, engaging and informative to fulfil FR6. Ideally students will start at the Interactive Tutorial section first to obtain the technical concepts required to advance onto the Workshop. However, this required a great deal of time and innovative thinking. “A creative solution are more than ideas, they must work in the real word. A creative solution has three attributes [13]”

1. Implementing something ‘new’
2. The solution is ‘useful’ in that it solves a problem
3. The new solution is ‘feasible’ overcoming real world constraints like money and time

The aim of the interactive tutorial is to disseminate specific subject knowledge to students as part of the learning process (OR6). It is an easy and effective mechanism to demonstrate the key concepts and ideas of what students found difficult highlighted in Chapter 3, to reiterate Use Case Diagrams and Activity Diagrams. It was crucial to ensure that the tutorial exploited and encouraged all learning styles particularly the use of the tactile approach, the reason being students must be stimulated and engaged at all times. In order to achieve this; the application was designed to be highly interactive (NFR1) and above all allow students to consolidate the information taught in lectures which fulfils OR7.

The Interactive Tutorial was separated into two distinct sections Use Case Diagram Page and Activity Diagram Page.

The tutorials for both pages shared a similar theme and layout in the learning process listed below. An example of the page layout is shown in Figure 13:

- **Step 1**: Introduction to each diagram
- **Step 2**: Explanation of key notations for each diagram
- **Step 3**: Example video of how to construct each diagram
- **Step 4**: Exercise for students to perform in relation to the topic
- **Step 5**: A game for students to perform in relation to the topic

![Figure 13: Use Case Page Layout Example](image)

The objective now is to explore how these five steps where implemented into the Interactive Tutorial.

### 6.4.1 Use Case Diagram Page

Before proceeding it is important to highlight that the technical concepts for both Use Case Diagrams and Activity Diagrams used in the web application were introduced and explained using real world analogies and a range of exercises for students to perform (FR7).

The reason being “an analogy is a way to explain new information to learners by relating it to information they already know. [14]” All analogies used for each exercise can be seen in Appendix B.

**Step 1**: Introduction to each diagram

Firstly to introduce the main concept of a Use Case Diagram, it was important to teach students how to effectively draw a use case. This was taught through the use of an analogy which was of an Ebay buyer bidding for an item with the intention of winning the auction. This was aided by an exercise for students to perform called Draw Use Case Exercise shown in Figure 14.
The aim of this exercise was to promote student engagement by enabling them to freely draw on the canvas. The objective of the task was to encourage a trial and error technique. “*Trial and error is typically good for problems where you have multiple chances to get the correct solution [15]*”. Since the nature of software engineering is highly subjective this technique deemed worthy. Once students have read through the analogy they can spontaneously draw and note down what they think would be a potential use case base on the analogy. Once satisfied with their answer students could click on ‘Show Answer’ and three or four example use cases appeared for them to compare providing students with constructive feedback (FR3).

**Step 2: Explanation of key notations for each diagram**

Now that students have understood the steps required to drawing a use case, it was important to introduce all the key notations required to build a Use Case Diagram successfully. This was achieved by firstly explaining what each notion meant and then matching the use case notation to their respective box, better explained in figure 15.

The objective of the exercise was for students to drag and drop the icons which are going to be placed randomly in the boxes at the top and place the icons to the respective boxes below. The outcome was to test if students understood the definition and the meaning that related to each icon. Students could then click on the ‘Show Answer’ button to check if they have placed the icons in the correct locations.

**Step 3: Example video of how to construct each diagram**

Subsequently, now that students have been exposed to all the notations required to build a Use Case Diagram it was important to illustrate and explain how all these components are composed together to construct an effective Use Case Diagram. This was achieved by developing a video tutorial. The analogy used was of a bank ATM machine and the types of transactions a customer can
perform in a step by step sequence. This provided a different means of interaction that “actively engage the learners’ mental process and creates interest in learning more about the topic [16]”.

**Step 4:** Exercise for students to perform in relation to the topic

After watching the video students at this point got a deeper understanding of the respective domain. During which point it was important to test the students understanding before moving on to the Interactive Tutorial. This was carefully executed by developing an exercise for students to undertake shown in Figure 16 using a library analogy.

![Figure 16: Capturing Use Case Exercise](image)

The aim of the exercise was to test the students understanding of capturing use cases from a particular scenario. The outcome was to ensure that students captured all the correct use cases. This enabled students to test themselves by allowing them to freely choose 3 out of the 6 use cases and writing them down in the following boxes ‘A1, A2 and A3’. Once students are satisfied with their results they could click on the ‘Show Answer’ button which highlight all the right answers in green and all the wrong answers in red.

**Step 5:** A game for students to perform in relation to the topic

Finally to ensure there was a fun factor added to the learning experience a game was developed called Hangman shown in Figure 17. It is a game everyone understands and aimed to “teach problem solving which is an important skill to learn, more important in the age of digital information it helps memorizing certain facts and formulas which can be easily located [17]”.

![Figure 17: Hangman Game](image)

However to develop this game, it required careful planning and attention to detail. The basic details of the Hangman game are listed below:

1. Develop the visibility of the text field
2. Listen for the key press
3. Get the letter pressed
4. Loop through and find the matching letters
5. Match found
6. Update screen
7. Update hangman

To implement these steps into an algorithm is a challenging process and requires precision. To simplify the task it was imperative to write down the pseudo code as “it describes the entire logic of the algorithm so that implementation becomes a rote mechanical task of translating line by line into source code” [18]. The following pseudo code in Algorithm 1 states the logic used to check what letter the users has pressed.

Algorithm 1: CheckInputKey

```
begin dealWithKeyPress() {
    letterPressed = getLetterPressed()
    foundLetter = false
    for each letter in phrase
        if letter = letterpressed
            change shown phrase
            foundLetter = true
    updateOnScreenText()
    if !foundLetter
        numWrong = numWrong + 1
}
```

Table 5: Algorithm 1 - CheckInputKey

6.4.2 Activity Diagram Page
As mentioned above the Activity Diagram Page took the same layout and learning process steps as the Use Case Diagram Page. With respect to step one, two and three the same tasks were used to test the student. However, different analogies and key notation explanations were used.

The example video used an analogy based on process and activities involved when processing an order. In addition, the task used the same method of explanation as the Use Case Diagram. However, the Activity Diagram page did not embed an exercise for students to perform but included a game.

The game developed for the Activity Diagram page was called Word Search shown in Figure 18. This allowed students to find all the buzz words related to an Activity Diagram. The students will be familiar with all the words inserted in the game as it was obtained from step two ‘key notations’.

Figure 18: Word Search Game
The basic detail of the Word Search game is listed in a step by step format below:

1. Create word list and fields
2. Position the words in a grid of letters
3. Add random letters to the grid
4. Find selected letters based on start and end points
5. Check word against word list
6. Word found, remove from list, make outline permanent

Attempting to implement all these steps into pseudo code will be excessive as the most complicated algorithm should be translated which is step 5. This step can be drilled down further into sub-steps which are:

5. Check word against word list
   1.1 Loop though words
   1.2 Compare word
   1.3 Compare word reverse

The following pseudo code in Algorithm 2 states the logic used to check the word against the words stored.

Algorithm 2: CheckWord

```
begin checkWord(word) {
    for each word
        if word in usedWords
            foundWord(word)
            reversedWord = reverseWord(word)
            if reversedWord in usedWords
                foundWord (reversedWord)

Table 6: Algorithm 2 - CheckWord
```

6.5 Workshop

Now that students have completed the Interactive Tutorial they would have gained the knowledge required to proceed onto the Workshop.

“A workshop provides a way to create an intensive educational experience in a short amount of time…. It’s a great way to teach hands-on skills because it offers participants a chance to try out new methods and fail in a safe situation. [19]”

In this case, the Workshop consisted of four different games, testing the student specific and broad knowledge on different aspects of Use Case and Activity Diagrams.

The first two games were generic, testing the student knowledge on both Use Case and Activity diagrams. The aim of the games was to test if a student acquired the basic knowledge of both diagrams by formulating 10 questions per game based on the Interactive Tutorial and lecture slides. The objective of both games was to ensure students have understood the fundamental concepts with respect to these specific domains. The outcome of these games allowed students to proceed onto the next two games that are very specific to each topic.
The first two games developed are called Multiple Choice (MCQ) and Fill-in-the-Blanks. The purpose of these games is to measure the student ability with respect to the information taught and allowed for questions to be proposed in such a way that it required students to recall, interpret and problem solve each question.

“Recall items simply ask candidates to recall or recognize a fact. Interpretive items require candidates to use their base of knowledge to interpret data or other information and come to some conclusion. Problem solving items require the candidate to assess a situation, synthesize with information from their base of knowledge, and then correctly solve a problem or make a decision [20].”

Both these games have proven in the past to be an effective method of assessing student’s knowledge quickly.

The MCQ show in Figure 19 permitted students to choose one correct answer from a list of four possible answers. It offered students a method to obtain results instantly. Though the implementation of a MCQ was initially time consuming to create, it did however facilitate more coverage of challenging learning material. Conversely, “it doesn’t expand a student’s knowledge beyond the choices that are offered for the questions as the right answer. And like true and false quizzes, test takers may be lucky enough to just guess most of the answers correctly [21]”. The second game Fill-in-the-Blanks aimed to test the acquisition and recall of the student.

Fill-in-the-Blanks shown in Figure 20 required students to complete a sentence that has the key word omitted, forcing students to rely on their memory to fill in the correct answer to complete the sentence. Unlike MCQ, where the students had the options to randomly guess the answer, in this case if the student does not know the answer he/she will be unable to complete the sentence. In addition, if students happen to skip reading lecture slides, the Interactive Tutorial or do not understand the topics Fill-in-the-Blanks will be a good indicator to highlight shortcomings in their understanding.

The third game developed was called Use Case Spot the Error shown in Figure 21 which was devised to help students identify common errors when constructing Use Case Diagrams. The example Use
Case Diagram was deliberately implemented with errors for students to identify and most importantly justify. The game was carefully developed using the feedback obtained from the questionnaire, for instance student three lost marks for naming the actor incorrectly so the example Use Case diagram incorporated this. The aim of the game was to help students identify common errors but justify why they think it is an error using MCQ. The objective of the game was for students to not only spot all errors but understand why it is an error and will be appointed marks for each time they get the MCQ right. The outcome of the game was for it to be interactive, engaging and most importantly help students understand and recognise common mistakes when developing Use Case Diagrams.

The final game was called Activity Central shown in Figure 22 which intended to encourage students to understand how the flow of activities will be modelled in a system. This was achieved by providing students an example scenario of the formal processes required when enrolling into University.
The aim of the game was to help students identify key activities and assist in identifying the direction of flow from one activity to the next. The object of the game was for students to drag and drop each activity into correct position in the predefined flow. The outcome was for students to grasp a better understanding of Activity Diagrams, fundamentally the aspect of capturing the activities and the direction in which it flows. Once students have completed the game and are satisfied with the outcome, they had to click on submit button and the correct version of the diagram appeared on the top right corner with the score beside it (highlighted in red). This allowed students to understand where they have gone wrong fulfilling FR3.

6.6 Forum
Having completed the Interactive Tutorial and the Workshop, students at this point may have questions (FR4) which warranted the development of an online forum. Furthermore, the purpose of the forum was to address the issue of those students that have answered ‘no’ in the questionnaire regarding asking questions during class hours.

The objective of the forum was to facilitate all students to openly ask, discuss and respond to questions within the scope of software engineering. The outcome is for students to receive feedback with regards to the question they have posted and to inspire peer-to-peer learning by forming an online community.

To ensure that the forum was effective it was based on a three tier system starting with three predefined categories for students to choose from which are: Activity Diagram Category Page, Use Case Diagram Category Page and Others Category Page (Figure 23). The reason for this design decision was that it prevented the forum from becoming cluttered and unnecessary categories being created with no replies, hence eliminating any confusion.

<table>
<thead>
<tr>
<th>Forum Category Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Diagram Category Page</td>
</tr>
<tr>
<td>Activity Diagram Category Page</td>
</tr>
<tr>
<td>Others Category Page</td>
</tr>
</tbody>
</table>

Figure 23: Forum Category Page

Once students have decided which category to choose they will proceed onto the second tier which is the Topic Page shown in Figure 24. This entitled students to ask questions by creating a topic or check to see if the question has been asked already. The students will only enter the third tier if under conditions students wanted to create, edit or delete a post. If students decided to create a topic or a post, it will be stamped with a User’s ID, Date and Time. Finally, each topic kept an account of the number of students that have viewed and replied to that topic.
6.7 Feedback Page

A fundamental aspect to this project was to design and develop a feedback method. This enabled students to keep track of their scores and ultimately allow students to measure their progress (FR2). The feedback for each individual game was published at the end of the game allowing the student to reflect and understand how to improve for next time. However, the score awarded for each game was displayed on a line graph shown in Figure 25 as it is “especially useful in the fields of statistics and science and are more popular than all other graphs combined because their visual characteristics reveal data trends clearly [22]”. The graph recorded the first three attempts and then restarted to attempt one. This allowed for the graph to be clear and concise but more importantly reduce the number of entries into the database.

Taking into account line graphs can show specific values of data clearly and the relation between data. Students were able to compare trends between games and compare their previous score for a specific game. It provided a quantitative approach to understanding how well they have understood the content. An underlying strategy of this chart was to motivate students; if for example a student has performed well in one game they can see which game they need to work on (i.e. area of development) and if they did poorly on their previous attempt then students may retake that game to try and improve their score.
The scores that are plotted on this graph are dynamically generated from a database. The underlying AJAX method performed a call to the database and the data received is then stored in a well formed data structure that is JSON encoded and used to populate the graph. Synchronized calls are made to the AJAX method to display multiple score lines. The following pseudo code in Algorithm 3 states the logic used to store the score into structured array and returned as JSON.

**Algorithm 3: StoreScore**

Check Login
If successful{
    Select users Id from the database
    If users Id exists
        Get all scores based on the users ID
        store scores into a structured array
        Encode the structured array to JSON
        Return JSON Encoded Object
    }
Else{
    Return No Data
}
Else{
    Return No Data
}

**Table 7: Algorithm 3 - StoreScoreArray**
Chapter 7 Web Application Implementation

7.1 Chapter Overview
Following on from the Design Chapter some thought was given to how the system was implemented. This involved becoming acquainted with various unfamiliar software programs. The implementation had to enhance the experience for the intended user. This Chapter looks at how the programs were used to create use of ease for the students, the storage of data, creation of illustrations, encouragement of user interaction and the implementation of animation and multimedia tutorials, games and videos with the provision of meaningful feedback to students. It required a range of new technologies and tools to be used. The outcome of the web application can viewed in Appendix C.

7.2 laying down the foundation
To ensure the web application entailed a professional look a website template called Rifle was obtained from Free CSS Templates [23]. The website required some restructuring and modification to ensure it was fit for purpose. By choosing a web template it helped encouraged good coding practice and was easily editable. The template was programmed in HTML5 and CSS3, this help me understand that HTML5 is a mark-up language which enables a website to form a structure and allows content to be presented onto the World Wide Web.

Adopting HTML5 for the project requires no installation of software but rather the latest web browser. The main reason for using a HTML5 based template is because of the dynamic properties such as enabling the web application to re-size and alter itself for any type of device (interoperability). The list includes smartphones, tablets and browsers. It facilitated students to interact with the website on the move thus tapping into the widest possible market but also fulfils NFR2 and OR3.

Finally, CSS3 allowed the formatting of a webpage and concentrated on the presentational semantics. The primary objective of CSS3 is that it enabled a way to control the following on the web application: text fonts, size, colour, basic object positioning and identification.

7.3 Login and Register Page
After this implementing the login and register page was completed simultaneously to fulfil FR1. The learning curve was extremely steep as it required PHP and MySQL programming languages to function concurrently to obtain desired results.

PHP scripts allowed student information to be gathered when interacting with the website. The data was parsed to a database and retrieved using SQL Statements. Once there was a connection between PHP and MySQL the combination was a powerful tool that allowed dynamic webpages to interact with a student while collecting data and creating specific content on the fly (NFR3).

The reason for using PHP is because it is a scripting language that has been specifically designed to be used alongside HTML5, enhancing the functionality and abilities of the website. Finally, MySQL is a server-side scripting language which was used to store all user information such as login detail,
forum information and game scores and enabled students to retrieve data from the server on the move.

To develop the login and register page a HTML form was implemented. The form data is parsed onto the PHP file using a POST method. The PHP processes the information and MySQL inserts a new row of data into the database using an ‘Insert’ statement. However, before students can register there is a server side validation to check if a student already exists. This will provide students with an error message on the same page as the form. This will then allow students to easily handle the error and type in a different username. There is also a user input validation to reduce the server load to fulfil SR3, SR5 and HN9.

To ensure the students password was secured the first level of encryption referred to as md5 hash encryption was used. This translated a student password into a hash by using the PHP mcrypt_encrypt library. “However, it is a one-way transaction and as such it is almost impossible to reverse engineer an MD5 hash to retrieve the original string [24].”

7.3 Home Page
After the completion of the login and register page it was imperative to guide students though the web application. This was achieved by carefully designing four self-explanatory images using Adobe Illustrator.

“Adobe Illustrator is computer software that enables users to design, modify and edit vector graphics images from a computer. Contained within the software are a range of tools designed to allow a smooth interface between the user and the program [25].”

Learning the range of different tools such as pen and lasso tool was a difficult and time consuming process. However, the final outcome shown in Figure 26 is aimed to help students acknowledge the sequence in which the application should be executed. A point to note is that if a student decides not to use the images to navigate they have the option to revert to the navigation bar on top.

7.4 Interactive Tutorial Page
To ensure students are fully engaged an application program interface (API) called Filpboard was obtained and integrated.

“API is a set of routines, protocols, and tools for building software applications. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together [26].”

Integrating Flipboard was aimed to encouraged interaction by allowing a student to drag the page from right to left like a book shown in Figure 27. This in turn, will engage a student as it “transform how people discover, view and share content by combining the beauty and ease of print [27].”
Another beneficial factor of using Flipboard was the extensive use of JavaScript. HTML provides the content, while CSS provides the structure and JavaScript was added to provide the behaviour. Even though HTML and CSS can provide simple interaction to a webpage such as a user form this would however be considered a static webpage. The use of JavaScript enabled superior interaction with the website without them having to wait for the next page to load each time the webpage send for a request. The final reason for using JavaScript was that it improved the overall experience for students.

Having now put together the foundation for the Interactivity Tutorial page, it is important to note how all exercises and games were implemented by only explaining the most difficult and interesting aspects.

All exercises were developed in Adobe Flash Professional. The software is dedicated to creating interactive animation and multimedia content that can be used across different desktops and multiple devices, including tablets, smartphones. This was aided by ActionScript (AS3) which is an object-oriented programming (OOP) language typically used for developing website animation.

“ActionScript makes it possible for developers to create onscreen environments (such as games, tutorials, and e-commerce applications) that can respond to user input through the keyboard or mouse. ActionScript is an event-based language: just as is the case in real life, actions are triggered by events [28].”

Once all exercises were developed it was exported as .SWF format ready to be integrated into the website.

7.4.1 Use Case Diagram Page
The first exercise that was implemented is called Draw Use Case Exercise. The most difficult aspect of the exercise was to capture user input as they draw across the stage. The most effective method was to create an AS3 file that called the Flash File. The AS3 file defined the width, height and colour of the object that will be printed on the stage. The Flash File contained three ActionListeners, one to listen to the drawing initiation and another to listen to when the drawing has stopped. The stage mapped the drawing objects using x and y coordinate positioning and stored the value into an array. The final actionlistener was used to listen to the click of the erase button to delete the drawing objects off the stage by emptying the array show in Figure 28.
Implementing the second exercise took a slightly different approach to the first exercise. At this point a proficient level of knowledge was required to execute the transitions between scenes in Adobe Flash.

“A scene is like a clip of a movie, which can be treated as an entire single unit all on its own and arranged around other clips [29]”.

For this exercise two movie clips were recorded one with the answers and the second without answers. Two ActionListeners were initiated per object on screen. Once a student has dragged the object from the top table to the table below they can click on a button call ‘Show Answer’ which will trigger scene 2 and vice versa if the student clicks on a button called ‘Hide Answer’ shown in Figure 29.

As oppose to writing out a lengthy piece of ActionScript aided by a large number of scenes, using Adobe Timeline to develop a tutorial video proved to be an effective and efficient method.

“The Timeline in Adobe Flash Professional organizes and controls a document’s content over time in layers and frames [30]”.

M o h a m m e d  Y a s e e n  A l i

45 | P a g e
Adobe Flash provides a range of different types of techniques to create animation such as:

1. **Motion Tweens** – sets the properties of an object such as the position and is particular useful for animation that consists of continuous motion or transformation of an object.
2. **Classic Tweens** – similar to motion tween but useful for specific animation. The effect is usually not used for continuous motion.
3. **Inverse Kinematic Poses** – permits for objects to be stretched and shaped during motion and more complex to create.
4. **Shape Tweens** – allows for a shape to be drawn at one specific frame and allows for the shape to be changed at another specific frame. Flash “*then interpolates the intermediate shapes for the frames in between, creating the animation of one shape morphing into another* [31].”
5. **Frame-by-Frame Animation** – this allows for graphic elements of each frame to be different by permitting different images for different frames.

When developing both the tutorial videos the following three tweens were applied: motion tween, classic tween and shape tween.

Shape Tween was applied once the animation started. This enabled objects to move between different frames. A classic tween was applied to change the colour of the text to red as the video was being executed. Simultaneously a motion tween was applied to indicate which step the tutorial was on, this enabled objects to appear on stage.

The following three tweens were applied continuously to complete both the tutorial videos. It was a very time consuming and confusing process.

A complex aspect was the development and implementation of the Hangman Game for the Use Case Diagram page:

**Step 1: Developing the hangman image (Figure 30)**
- Develop a Frame-by-Frame Animation of the hangman failing off
- Add ‘X’ beside the image to indicate how many guesses the student got wrong
- Print off the outcome
- Create a new scene with an image so when the student get the answer right

![Figure 30: Hangman Image Development](image)

**Step 2: Setting up the visible text field (Figure 31)**
- Define the width and height dimensions of text field
- Define x and y positioning of text field
- Wrap text to ensure it is within the defined text field
Web Application using Learning Tools to Teach Software Engineering

- Text field instance was added to the stage via addChild() method

```javascript
public function Hangman() {
    // create a copy of text with _ for each letter
    shown = phrase.replace(/[a-zA-Z]/g, "_");
    numWrong = 0;
    message_txt.visible = false;
    // set up the visible text field
    textDisplay = new TextField();
    textDisplay.defaultTextFormat = new TextFormat("Courier", 30);
    textDisplay.width = 400;
    textDisplay.height = 200;
    textDisplay.x = 50;
    textDisplay.y = 50;
    textDisplay.wordWrap = true;
    textDisplay.selectable = false;
    textDisplay.text = shown;
    addChild(textDisplay);

    // listen for key presses
    stage.addEventListener(KeyboardEvent.KEY_DOWN, pressKey);
}
```

Figure 31: Text Field Initialisation

**Step 3: Implement Algorithm 1 CheckInputKey**

After writing down the pseudo code for getting the keyboard input, careful consideration and precaution was taken when implementing the code shown in Figure 32. During the testing phase the game failed to match letters if the key inputted was uppercase letters. A single line of code was added to resolve the issue which converted the alphabetical characters in the string to lowercases. This was achieved by calling upon a .toLowerCase() method.

```javascript
public function pressKey(event:KeyboardEvent)
{
    // get letter pressed
    var charPressed:String = (String.fromCharCode(event.charCode));

    // loop through and find matching letters
    var foundLetter:Boolean = false;
    for(var i:int=0;i<phrase.length;i++) {
        if (phrase.charAt(i).toLowerCase() === charPressed) {
            // match found, change phrase
            shown = shown.substring(0,i)+phrase.charAt(i)+shown.substring(i+1);
            foundLetter = true;
        }
    }

    // update on-screen text
    textDisplay.text = shown;

    if (textDisplay.text == "Actor"){
        message_txt.visible = true;
    }

    // update hangman
    if (!foundLetter) {
        numWrong++;
        character.gotoAndStop(numWrong+1);
    }
}
```

Figure 32: Algorithm 1 Implementation

**7.4.2 Activity Diagram Page**

The Word Search game developed for the Activity Diagram Page was another interesting aspect of system build. The game was implemented within three incremental steps:
Step 1: Create a two dimensional array (Figure 33)
- Create an empty grid
- Insert words into the grid
- And fill the rest of the grid with random letters using Math.Random() method

```java
// place the words in a grid of letters
public function placeLetters() {
    // create empty grid
    var letters:Array = new Array();
    for(var x=0; x<gridSize;x++) {
        letters[x] = new Array();
    }
    // insert word into grid
    insertLoopFor (10, word.length++) {
        letters[(x=(x%d))+y] = word,choked();
    }
    // fill rest of grid with random letters
    for(var x=0; x<gridSize;x++) {
        for(var y=0;y<gridSize,y++) {
            if [letters[x][y] = ""] {
                letters[x][y] = String.fromCharCode(65+Math.floor(Math.random()+26));
            }
        }
    }
    return letters;
}
```

Figure 33: Initialise and Fill Array

Step 2: Record student input as they hover over the word grid (Figure 34)
- Player clicks down on a letter to start
- Player dragging over letters
- If invalid range, change colour to red
- Mouse released
- Get word and pass it on to check function()

```java
// player clicks down on a letter to start
public function clickLetter(event:MouseEvent) {
    var letter: string = event.currentTarget.addEventListener[0].text;
    start point = findGridPoint(event.currentTarget);
    dragged: = "drag";
}
// player dragging over letters
public function overLetter(event:MouseEvent) {
    if (dragmode == "drag") {
        endPoint = findGridPoint(event.currentTarget);
        // if valid range, show outline
        outlineSprite.graphics.clear();
        if (isValidRange(startPoint, endPoint)) {
            drawOutline(outlineSprite, startPoint, endPoint, 0xFF0000);
        }
    }
    // mouse released
    public function mouseRelease(event:MouseEvent) {
        if (dragmode == "drag") {
            dragmode: = "mouse";
            outlineSprite.graphics.clear();
            // get word and check it
            if (isValidRange(startPoint, endPoint)) {
                var word = getSelectedWord();
                checkWord(word);
            }
        }
    }
    return word;
}
```

Figure 34: Word Grid Hover Input

Step 3: Implement Algorithm 2 CheckWord (Figure 35)
- Check word against the word list
- Compare the words

Mohammed Yaseen Ali
- If word is found remove word from list
- If valid range, change colour to green
- Change colour of word in the word list to gray
- End game

```javascript
// check word in set word list
public function checkWord(word string) {
    // loop through words
    for (i = 0; i < wordList.length; i++) {
        // compare word
        if (word == wordList[i].toLowerCase()) {
            foundWord(word);
        }
        // compare word reversed
        var revWord = word.split('').reverse().join('');
        if (revWord == wordList[i].toLowerCase()) {
            foundWord(revWord);
        }
    }
    // word found, remove from list, make outline permanent
    public function displayWord(word string) {
        // draw outlines (beginOutlineSprite, startSprite, endSprite, endOutline)
        // find text field and set it to gray
        for (i = 0; i < document.querySelectorAll('#theclass').length; i++) {
            textField = document.querySelectorAll('#theclass').item(i);
            textField.className += ' textColor = #000000';
        }
        // see if all have been found
        if (found == true) {
            endGame();
        }
    }
}
```

### Figure 35: Get Word and Compare

#### 7.5 Workshop Page
To make the Workshop engaging Fancybox API was used which was built using jQuery library.

"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript [32]."

All games were placed into a fancybox, allowing them to float overtop of the webpage reducing page reload.

The problem encountered when using fancybox was that it did not support flash file (.SWF) documents, so a new function had to be manually written. The function was set to trigger on `click` with the width and height of the fancybox auto scaling to the flash file created shown in Figure 36.

```javascript
$(".fancybox").click(function() {
    $this = $(this);
    autoScale: false,
    'title': this.title,
    'width': 1000,
    'height': 1000,
    'href': this.href,
    'type': 'iframes',
    'src': // pass your flashvars here:
            'widths': 'transparent',
            'allowfullscreen': 'true'
}));
```

### Figure 36: Fancybox Function
Web Application using Learning Tools to Teach Software Engineering

The first game called MCQ was implemented in HTML5 using a form. This theoretically worked very similar to the login and register page. However, the difference was as oppose to allowing students to manually type in the information, an option element was used to produce a list of options for the student to choose from, effectively a dropdown menu. The form data is parsed onto the PHP file using a POST method.

During Chapter 6, Section 6.7 Feedback, it was mentioned a student should be able to keep a track of their scores. On completion of the MCQ game the student score is parsed onto PHP and inserted into the database. Before proceeding it is important to highlight after the third attempt all three scores will be overwritten with the new scores. For example a student obtained 5, 7 and 6 in their first three attempts. However, in their fifth and sixth attempts they got 6 and 5 so their previous scores will be overwritten and will appear as 6, 5 and 0. This is to ensure that that database does not become excessive and the graph remains simple to read. In order to enforce the rule an if-statement was written stating, if ‘attempt three’ field has a value update ‘attempt one’ field and clear the score values of attempt two and three.

Use Case Spot the Error game proved to be a lot more complicated than originally expected. The complexity arose during the implementation of the ‘circle error’ indicator that should appear on screen. The circle is used to show an error has been found. The implementation of the game was achieved in the following incremental steps:

**Step 1: Develop the stage**
- Create all the Use Case shape objects within Flash
- Define all the object names
- Set up the MCQ question holder and answer buttons

**Step 2: Implementing the errors indicator**
- Hiding certain objects by making them transparent
- Define the x and y positioning of mouse click
- If clicked on valid range, circle error indicator will appear on screen
- Passing the information onto the MCQ by the hitTestPoint() method

**Step 3: Implementing the MCQ**
- Question will appear presenting options to choose
- If answer is correct pass value onto the Correct() Function
  - With a value of 1
- If answer is incorrect pass value onto the Incorrect() Function
  - With a value of 0

**Step 4: Show Result**
- Show incremental results
- Show end of game result

The final game implemented was Activity Central which contained drag and drop features. The important aspect of the game was to specify the collision point between the moveable object and its destination. This was done by implementing the objects to be dragged with specific coordinates. However, the moveable objects would be implanted with another smaller object which would be
transparent. The reason being smaller sized target points are going to be placed on stage with specific coordinates, which are going to correspond to the hidden size objects contained within the moveable objects. This would result in a more precise object positioning.

### 7.6 Forum Page

As mentioned in Section 7.3 Login and Register Page, the learning curve for PHP and MySQL was substantial. To implement a forum containing categories, topics, and posts also required CSS3 to create tables so the data retrieved from the database can be presented clearly.

Considering the categories where predefined it was important to query the database by using a ‘Select’ statement and appending the data from the categories table into a list of links and order them by category title. This was then styled using CSS3. However, before attempting to retrieve the data from the Categories table an if statement was performed to check if the Categories table has data in it. If there is no data an error message is returned stating “no categories are available”.

A complicated aspect of the forum was creating a topic within a category. Again a HTML5 form was created obtaining the title and content of the topic. The data was parsed onto a PHP file which used an SQL ‘Insert’ statement to add a new row of data into the posts table. However, this also meant executing an ‘Update’ statement on both the Categories table and Topics table that are associated with this topic shown in Figure 37.

```php
// Check to see if the person accessing this page is logged in
if (! $SESSION['logged_in']) {
    exit;
}
// Connect to the database
include('connect.php');
// Begin the SQL statement
$sql = "SELECT * FROM categories ORDER BY category_title DESC;"
// Execute the SQL statement
$result = $con->query($sql)
// Get the results
$categories = $result->fetch_all(MYSQL_ASSOC);
// Display the results
foreach ($categories as $category) {
    echo '<a href="view_category.php?id="'. $category['category_id'] . '"'>
        ' . $category['category_name'] . '</a>
    <br />
}
// End the SQL statement
$con->close();
```

Now that a topic has been initiated, creating a thread of multiple posts per topic was a complex process. This meant assigning a category id, topic id, post creator, post content and postdate to each post in the topic and concurrently updating the Category and Topic tables by executing the ‘Update’ statement.

The option for a post to be deleted was relatively straightforward to implement. All that was required was to write a SQL ‘Delete’ Statement which was used to delete a record from the posts table.

---

**Figure 37: Insert Topic into the Database**
7.6 Feedback Page

Implementing the feedback page was done using Highcharts API which uses a Charting Library. This offered "intuitive, interactive charts to the web application [33]". As mentioned earlier the scores that are inserted into the Database are now going to be retrieved and displayed on Highcharts. During the implementation of the chart, AJAX and JSON (JavaScript Object Notation) were used.

On one hand, AJAX was used to exchange data with the server and updating the webpage without reloading the whole page. On the other hand, JSON is a lightweight data exchange format, to parse and generate data.

JSON is built on two structures:

1. “A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
2. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence [34].”

This is advantageous as both JSON data structures are universal meaning they are interchangeable with most modern programming languages.

To populate the highchart, an AJAX function was written which used a GET method to obtain desired results in JSON. This was achieved using a block method, which means the script will block all request until the initial request has been fulfilled. Show in Figure 38.

```javascript
$(document).ready(function() {
    var myAJAXVariable = (function () {
        var myAjaxValue = null;

        $.ajax({
            'async': false,
            'global': false,
            'url': 'Charts/validate.php',
            success: function (returnedData) {
                myAjaxValue = returnedData;
            }
        });
        return myAjaxValue;
    })();
});
```

Figure 38: AJAX Request

To obtain the results from the results table, Algorithm 3: StoreScore was implemented. A data structure was used in PHP to store the retrieved data which is a specialised format to organise and store data. In this case all individual marks are retrieved from the results table and pushed into an array. The data structure in then encoded into a JSON format (using universal data structure one mentioned above) and returned to the AJAX call shown in Figure 39. An example of the JSON output is shown in Figure 40.
The JSON encoded variable is parsed into JavaScript and iterated through to acquire the scores and stored into global array (thereafter, the array is used to generate the Highchart data) shown in Figure 41.

```
      if (mysqli_num_rows($res) == 1) {
        $row=mysqli_fetch_array($res);
        $id=$row['id'];
        //Retrieve Scores from the Database
        $sql2 = "SELECT attempts_one, attempts_two, attempts_three | FROM results WHERE users_id=".$id;
        $row2 = mysql_query($sql2);
        $one = $row2['attempts_one'];
        $two = $row2['attempts_two'];
        $three = $row2['attempts_three'];
        $array=[ ];
        //Function inserts one or more elements to the end of an array.
        array_push($array,$one);
        array_push($array,$two);
        array_push($array,$three);
        //Convert an array into JSON format
        echo json_encode($array);
      }
      else{
        echo json_encode('No data!');
      }
```

Figure 39: Data Structure

The JSON encoded variable is parsed into JavaScript and iterated through to acquire the scores and stored into global array (thereafter, the array is used to generate the Highchart data) shown in Figure 41.

```
var myAJAXArray = JSON.parse(myAJAXVariable);
var array = new Array();
for(var i=0; i < myAJAXArray.length; i++){
  if (myAJAXArray[i]!="""){
      array.push(parseInt(myAJAXArray[i]));
    }
}
```

Figure 41: JSON to AJAX

### 7.7 Additional Information and Features

#### Additional Security

A feature added to the web application was a Session ID to add a layer of security.

“A session ID is a unique number that a Web site's server assigns a specific user for the duration of that user's visit (session) [35].”

By including a Session ID to each webpage it prevented unauthorized access. However, once a student logs out of the website the session is destroyed.

#### Integrating Notepad

Chapter 4, Section 4.2 Functional Requirement stated students must be able to make notes (FR8) however this deemed a low priority for this project. A last minute addition to web application was made by integrating a HTML5 based notepad. This was obtained from a website called Dynamic Drive [36] which uses DOM Storage capability to allow students to save their notes onto their hard drive.
“The content does not ever expire until explicitly cleared by the user. It works in browsers that support DOM Storage”.

The notepad was triggered dynamically via a link which was positioned as the first item on the navigation bar.

**Database Normalization**

Chapter 5, Section 5.6 ER Diagram explained the structure of the database. It introduced all the tables and described the purpose of each table and attribute. To elaborate further, a database can be normalized to reduce problems with respect to data consistency and duplication by eliminating redundant data. This is executed by dividing large tables into smaller tables which can potentially be normalized to the 6NF (Normal Form). However, to normalize a table to the 6NF is only applicable depending on the data contained within the table.

In most cases the third normal form is usually considered highest level necessary. Below is a description of the 1NF, 2NF and 3NF:

- **First Normal Form**: eliminates repeating groups in each table
- **Second Normal Form**: create new tables for set of value that apply to multiple records
- **Third Normal Form**: eliminate fields that do not depend on the key

Relating to the database developed for this particular web application, the third normalization form was applied shown in Figure 42.

![Database Schema (3NF)](image-url)
Chapter 8 Testing

8.1 Chapter Overview
This chapter describes the testing methodologies conducted to evaluate the system’s compliance with Chapter 4 - Requirement Specification. It is important to perform testing in order to identify the usability, correctness, robustness, completeness and quality of the system software. It was mentioned in Chapter 5 – Analysis, that IID will be used to help reduce the risk of implementing high risk Functional Requirements. Nevertheless, it is crucial to ensure a range of different testing was performed to eliminate all possible errors.

Testing was conducted after creating a test plan which is a “document describing the testing scope and activities. It is the basis for formally testing any software/product in a project [37]”.

The following test plan template was created shown in Table 8.

<table>
<thead>
<tr>
<th>Test ID</th>
<th>Test Case</th>
<th>Expected Outcome</th>
<th>Pass/Fail</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity Number</td>
<td>A description explaining the purpose of the test</td>
<td>A prediction of the outcome of what the test should be</td>
<td>The result of the outcome was denoted as Pass or Fail</td>
<td>If the test failed how was it rectified</td>
</tr>
</tbody>
</table>

Table 8: Structure of the Test Table

8.2 Unit Testing
Unit testing was conducted during the implementation of each exercise and game for the web application.

“Primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use [38]”.

An example of unit testing was highlighted in Chapter 7 – Implementation, Section 7.4.1 Use Case Diagram Page, when implementing “Algorithm 1 CheckInputKey” for the Hangman Game. The game failed to match the letters. The remedy was to add .toLowerCase() method.

8.3 Integration Testing
Integration testing was conducted straight after unit testing. This was undertaken by testing a series of units that are combined to demonstrate that different pieces of the application work together. During the integration stage common errors occurred such as syntax errors, bugs which were unclear in turn making it a time consuming process to debug. To track and eliminate these issues a method called White Box testing was used. This permitted breakpoints and extra print statements to be used to detect and deal with the issues quickly.
8.4 System Testing
After the completion of the integration testing this was followed up with system testing. This was done by a fellow student with the intent of breaking the code and finding bugs. This was advantageous as the tester tested all functions of the application as a whole and all errors were recorded in the error log. All known errors were promptly fixed ready for acceptance testing. The test cases developed for system testing was to validate the non-functional requirements and to verify that the application also met the function requirements.

8.5 Acceptance Testing
Before handing over the system to the students it was important to preform acceptance testing. This was undertaken as a formal test to determine if the system satisfies all the acceptance criteria. This was performed by an end user, but a plan was developed to assist the user. This was a crucial aspect of the testing phase as it helped identify important bugs that affected the usability of the application.

The testing method used was called Black Box testing which is a “software testing techniques where the functionality is tested without looking at the internal code structure, execution information and knowledge of internal paths of the software. This kind of testing is wholly based on the software requirements and specifications [39]”.

8.6 Manual Testing
When making modification and tweaks to the website template mentioned earlier it was important to perform manual testing. This was difficult and time consuming to test as no automated tools or scripts were used as a result the testing involved writing short pieces of code and refreshing the screen to view results. Making modification to one aspect of the website did have an effect on another part of the website which was recorded in the error log. All known errors were fixed immediately on the same day.

All the different types of testing mention above were used during the development and implementation of the web application. However, this does not measure how effective the student’s usability and learning experience of the web application was, so a questionnaire was devised.
8.7 Evaluating Usability and Learning Experience

To evaluate the usability and learning experience of the web application a questionnaire was developed. This was given to ten second year students. The questionnaire was divided into two sections the first part was aimed to determine how effective the usability of the web application was. The second part was to determine how effective the web application contributed towards the student’s Software Engineering module. Below are summary statements of the questionnaire finding. The questionnaire can be found in Appendix D.

8.7.1 Usability Results

In order to measure the usability of the web application five questions were asked to students shown in Figure 43.

<table>
<thead>
<tr>
<th>Usability Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look and feel of the website</td>
</tr>
<tr>
<td>Organization of the content</td>
</tr>
<tr>
<td>Ease of finding information</td>
</tr>
<tr>
<td>Content (clarity of language used, spelling and grammar)</td>
</tr>
<tr>
<td>Easy to navigate</td>
</tr>
</tbody>
</table>

![Figure 43: Usability Results](image)

The first question asked students to rate the “look and feel of the website”. 70% of students thought the website looked and felt very good. 20% deeming it good and the remaining 10% thought it was reasonable. This was achieved using a website template as each webpage contained the same design and layout.

The second question asked students to rate the “organisation of the content” of the website. 80% of student thought the organisation of content was very good. 10% of students declared it good and the remaining 10% of students answered reasonable. This was achieved by presenting the information in a structured fashion and grouping related elements accordingly.

The third question asked students the “ease of finding information” on the website. 90% of the students thought the website was very good for finding relevant information. The remaining 10% of students expressed that the website was good to find information. This was achieved by situating the main navigation bar consistently on the top of each web page. This was to ensure all information was easily accessible by students.
The fourth question asked students what they thought of the “Content (clarity of language used, spelling and grammar)” in the website. 60% of students thought the website content was very good, 30% highlighted it was good and the remaining 10% indicated it was reasonable. Though the average number of students scored the website content relatively high, it was unfortunately below expectations. However, this could be easily improved by reviewing the web content and ensuring all spelling and grammar are removed immediately.

The final question asked students if the website was “easy to navigate”. All students highlighted the navigational aspect of the website was very good. This was achieved by using the same font and colour. All hyperlinks contained within the website had obvious names such as ‘Interactive Tutorial’ making it clear to understand. All links were organised according to the order of importance.

The reason the usability aspect of the web application was highly successful is because it embodied good use of HCI principles. Adhering to Nielson Heuristics and Schneiderman Eight Golden Rules was important and it provided the theories to understand the importance of colour, positioning, sizes of objects, patterns and consistency which are all contributing factors that influenced the student’s decisions.

8.7.2 Learning Experience Results

Measuring the learning experience of the students is a key element to this project. The pressing need for measuring and understanding how effectively the web application has contributed toward students learning is crucial. This was achieved using the Likert Scale, which is a rating scale developed to measure students attitudes towards the web application directly using series of short statements. This was a direct approach “tapping into the students cognitive and affective components of attitude” [40].

In order to measure the learning experience of the web application multiple questions were asked to students. The questionnaire was divided into three sub sections. The first sub section assessed the learning outcome shown in Figure 44. The second sub section assessed the effectiveness of the exercises and games shown in Figure 45 and the third sub section measured the effectiveness of notepad shown in Figure 46.

First sub section – Learning Outcome

![Learning Outcome Results](image_url)
The results from the previous questionnaire stated some students did not understand the purpose of both Use Case Diagrams and Activity Diagrams. The first question asked students if the “Interactive Tutorial helped you understand the purpose of both UC and AD?” 70% of students strongly agreed it did and the remaining 30% of students agreed.

The second question asked students if the “Interactive Tutorial helped you understand when and how to apply UC and AD?” 80% of students agreed it helped them understand and the remaining 20% strongly agreed.

The third question asked if the “Concepts were introduced well for both UC and AD” 80% of students strongly agreed it introduced the concepts well and the remaining 20% of students agreed.

Based on the first three questions all ten students agreed the interactive tutorial helped them understand the purpose of these diagrams and how it links into the software development process. The reason being both pages explained the concepts in a clear and logical format.

The fourth question asked if the “Material successfully tied up well with the Software engineering module?” It was highlighted all ten students found the material in the tutorial pages relevant and the information tied up really well with the Software engineering Module. This could be based on the fact the information was extracted from reliable sources such as lecture notes and Applying UML Book.

The fifth question was if “The forum will help you get the answers you need when you don’t understand something?” 60% of strongly agreed it will help them get the answers they need. 20% of students agreed it would help obtain the information they are seeking and the remaining 20% of students stated they neither agree nor disagree.

The final question was “Is there anything you would like to suggest” there were a few interesting responses.

Integrating the forum was widely appreciated by the students giving them the option to explore other avenues if they lacked understanding or were unsure of the tutorial content. Students pointed out that it was easy to use allowing them to post a question when they are unsure of something.

On the one hand, during the testing phase no student posted on the forum as they believed the Interactive Tutorial provided them with adequate and reliable information. On the other hand, a student did state they liked the fact each post tallied up the number of views and replies. He stated further this is vital information as it would allow him to determine how difficult the question was. The higher number of views with no reply could potentially mean the question is difficult/uncommon problem. In addition, he stated this would also indicate how activate the forum was.

Though the forum was fully functional a drawback during the testing phase was the fact the sample size of ten was inadequate to measure the forum’s success.
Second sub section – Exercises and Games Results

The first question asked to students was if “The range of tutorials, exercises, games and videos enhanced your understanding?” 90% of the students strongly agree the range of methods enhanced their understanding in these domains and 10% of students agreed.

The second question asked to students was if the “Exercises and Games developed were easy to play and understand?” 80% of students strongly agreed they were easy to comprehend and play, leaving the remaining 20% of students to agree.

Analysing the results for the first two questions indicated by carefully designing and implementing a range of different learning tools it positively impacted on the student’s learning process. This was achieved by ensuring the exercises and games adhere to good HCI principles which resulted in them being easy to play and understand.

The third question asked students if “The exercises and games developed will help you learn and revise for exams?” 70% of the student strongly agreed it will help them during their revision period and the remaining 30% agreed it will.

The forth question asked students if “The games tied in well with the Interactive Tutorial” 90% of the students strongly agreed the games reflected the interactive tutorial well. The remaining 10% agreed the game linked in with the interactive tutorial.

Under the condition the web application was heavily dependent on the games to test a student’s knowledge. All students approved of the quality of games as the majority stated it will be beneficial to them during their revision period. This was achieved by carefully ensuring the games intertwined with the Interactive Tutorial closely.

The fifth question asked students if “The feedback provided was constructive on each exercise and game?” All students strongly agreed that the games provided enough feedback for them to understand why and how they went wrong on each exercise and game.

The sixth question asked students if “The line chart was a helpful feedback method that motivated you to do better next time?” 70% of student strongly agreed it would motivate them. 20% of student
agreed the feedback on a line graph is helpful. The remaining 10% of students neither agree nor disagree.

The final question was “*Is there anything you would like to suggest*” below are the summary of the responses.

Two students thought the games were difficult to complete. Though, they did agree the games were easy to play and understand. However, all students believed the games were fun, enjoyable and interactive, leaving them satisfied as it will help them revise for during their exam period.

Three students stated they enjoyed the ‘Drawing Use Case Exercise’ as it gave them a chance to practice and see the answer (trial and error technique).

Six students stated they really enjoyed and understood how to capture use cases through the tutorial videos.

One student thoroughly enjoyed the Hangman Game simple based on the fact it offered a fun factor.

A key aspect to the web application was how effectively it measures a learners understanding of each game played. It was highlighted four students thought the feedback page was extremely useful as it clearly laid out their scores for each game. They found the line graph easy to understand and stated considering each series on the graph is formulated based on their scores this would motivated them to do better next time under the conditions if the line is infrequent or below an average score of 8.

However, it was suggested the feedback page should also include the how long it took them to complete each game.

**Third sub section – Note Making**

![Figure 46: Notes Helped Recall Information Quickly](image)

Students were asked if “*Making notes helped you recall information quickly?*” 20% of the students strongly agreed it did help them recall information quickly. 40% agree it was useful and helped recall information. 30% of the students neither agree nor disagreed and the remaining 10% disagreed.

Final question was “*Is there anything you would like to suggest*” below are the statement summaries.
Web Application using Learning Tools to Teach Software Engineering

Three students stated the concept behind the note making was creative and interesting. However, they believed it did not work effectively in the web application based on the fact the notes could not be printed. Overall, the concept behind the note making was useful as it helped students recall information and would have been more useful if the notes were printable.
Chapter 9 Conclusion

9.1 The Project Objectives
Designing, implementing, and testing a web application which consisted of a vast number of well-defined objectives was a demanding and challenging process. Though confronted with time constraints and submitting to learning a range of different programming languages and designing software all objectives were achieved. From the very beginning it was crucial to ensure each and every objective was satisfied subsequently achieving the three aims of the project.

To proclaim the project a success based on overcoming the aims and objectives is implausible. The specification of the project stipulated that all features described thoroughly in Chapter 4 - Requirement Specification of this thesis are met. Considering all core functionalities have been implemented satisfying all Functional, Non-Functional and Optional Requirements including all design principles, this has judiciously deemed the web application a complete success.

9.2 Issues Encountered
The methods in which people learn is highly subjective, so aiming to enrich a person’s understanding using a web application was troublesome. Attempting to incorporate all methods of assessment and catering for all different types of learners is a time consuming process. Even though, the application did consist of a range of different exercises and games which appealed to the majority, a small fraction believed some of the games where difficult leaving them discouraged.

Implementing the Activity Central game which consisted of drag and drop features was undeniably a mind boggling process. The reason being when initially designed the collision point between the object being dragged to the destination point was the same size. Designing the size of the destination point to the moveable object seemed reasonable. However, the initial node, decision node and final node are of different sizes which were not taken into consideration. Keeping the design the way it was made the game a lot easier as the drop points for each object where of varying sizes. The only way of overcoming this problem was creating smaller objects of the same size and attaching them to each object and making the destination point the same size.

The navigational aspect was a minor issue encountered when developing the forum. This prevented a student from navigating back and forth between webpages. Initially the implementation for the back button simply included the url of the previous page. However, this was swiftly resolved by including a GET method that obtained both the Category and Topic ID.

9.3 Potential Future Development
The web application developed did contain a range of innovative learning tools allowing students to be full engaged and interested. However, a potential future develop could be designing and implementing a notepad. This could be achieved using the same mechanism as the Login and Register Page. A HTML5 form could have been implemented which would use a POST method parsing the data onto a PHP file and inserting the data into a database. CSS3 could have been used to make the form look like a notepad. Finally, once implemented and tested the notepad could have been integrated into Fancybox making it dynamic and easy to access from each and every page.
An interesting future development could be creating an administrator account for the following reasons listed below:

1. It will allow tutors to monitor student progress
2. It would allow tutors to monitor which games students find most attractive
3. If the site does prove to be popular it could facilitate tutors to update the tutorial page by dynamically adding content and adding a new category to the forum

Integrating all features would potentially lead to a very powerful learning platform as future development will be based on student trends.

Under the circumstances a student wanted to change their personal details e.g. username or forgets their password there are currently no provisions implemented to deal such situations. A future development could be allowing a student to changed their personal details or even recover their password. This was not achieved in this project due to time constraints and because it was beyond the core purpose of this project.

A limitation of the web application was the fact all exercises and most games were developed using Flash. This originally meant the web application will not work with any iOS devices. However, Adobe Flash Professional CS6 has introduced an extension toolkit which can transform current flash files into HTML5.

The toolkit “empowers designers and animators working with Flash Professional to create new assets for HTML5-based games and other rich interactive content. The extension will support many of the core animation and drawing capabilities of Flash Professional and allow users to export content in a format that targets the open source CreateJS framework [41].”

A future development could be using the new Flash Professional CS6 to develop future games.

The games such as MCQ and Fill-in-the-Blanks can be simple expanded upon by obtaining more information from the lecture notes and Applying UML book. The MCQ may take a little longer than Fill-in-the-Blank game since Fill-in-the-Blank is a simple of case of copying and pasting the sentence and omitting the key word.

The games that have been developed such as Use Case Spot the Error and Activity Central can easily be expanded upon by adding new improved diagrams which embed errors that are easier to identify.

With respect to Activity Central additional scenarios can be included with more complex levels.

9.4 Personal Achievement

The complexity of this project and the level of innovating thinking that was demanded were far beyond my comprehension. To overcome all the challenges encountered involved in this project could only be achieved through determination and self-discipline. Though the project was challenging and diverse it did offer a wide range of opportunities to learn the importance of technological advancement in the field of education. This fascination has spawned an interest which will be researching further in the future.

The project also explored and implemented an array of languages allowing me to enhance my skills in the following areas: HTML5, CSS3, JavaScript, JQuery, AJAX, JSON, PHP, MySQL, Adobe Flash
Professional and ActionScript 3. By becoming proficient in these languages it has enabled me to grasp a good understanding of the complexities involved in designing, implementing and testing a web application.

In addition, the project also involved the use of Adobe Illustrator. After developing images in this software I have come to understand the combination of being able to use a design package fluently with the comprehensive list of languages mentioned above could potentially lead to developing weird and wonderful web applications.

9.5 Closing Comment
Developing a Web Application using Learning Tools to Teach Software Engineering to this level of magnitude was a personal satisfaction. It will provide valuable support to 2nd year students enabling them to hone into their UML diagraming skills. In addition, others studying the Software Engineering module will see the vision instilled into this web application and hopefully make future enhancement through new up and coming means of interactions. This web application can be considered the first of it kinds with respect to research and development. Undoubtedly, there are a lot more radical changes and ideas that could be integrated to this web application but will require a large amount of planning, designing but most importantly testing.
Works Cited


Web Application using Learning Tools to Teach Software Engineering

[Accessed 22 February 2013].


Appendices

Appendix A: Questionnaire

UML diagrams can be classified as two categories which are Structural and Behavioural diagrams.

1. Which category did you find most difficult?
   - Structural Diagrams
   - Behavioural Diagrams

1.1. If you have selected Structural Diagrams, please select which topic(s) you found difficult
   - Class Diagram
   - Object Diagram
   - Component Diagram

1.2. If you have selected Structural Diagrams, please select which topic(s) you found difficult
   - Use Case Diagram
   - Interaction Diagram
   - Sequence Diagram
   - Activity Diagram
   - State Machine Diagram
   - Communication Diagram

1.3. What did you find difficult about the following diagram(s)?

2. From the list below which method(s) do you think would aid your learning process?
   - Tutorials
   - Exercises
   - Games
   - None
   - Others
   - Videos

2.1. If you have selected either none or others could you please specify?

3. If you don’t understand what is being said during lecture hours, do you ask questions in the lecture?
   - Yes
   - No

3.1. If answered ‘No’ why not?

3.2. Would you feel more comfortable asking questions using any of the methods listed below?
Web Application using Learning Tools to Teach Software Engineering

4. Is there anything you would like to suggest?

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________
Appendix B: Analogy Used

Draw Use Case Exercise Analogy

An EBAY buyer has identified an item they wish to buy, so they will place a bid for an item with the intent of winning the auction and paying for the item.

Use Case Diagram Video Tutorial Analogy

Customer uses bank ATM to check balances of his/her bank accounts, deposit funds, withdraw cash and/or transfer funds. All these use cases also involve Bank due to it relating to customer transactions [42].

Capturing Use Case Exercise Analogy

An Online Public Access Catalogue (OPAC) is e-Library website which is part of Integrated Library System (ILS), also known as a Library Management System (LMS), and managed by a library or group of libraries [43].

Activity Diagram Video Tutorial Analogy

Once the order is received the activities split into two parallel sets of activities. One side fills and sends the order. On the fill order side, the method of delivery is decided. Depending on the condition either the overnight delivery activity or the regular delivery activity is performed. While the other handles the billing. Finally the parallel activates combine to close the order [44].

Activity Central Game Analogy

An applicant wants to enrol in the university. The applicant hands a filled out copy of form University Application Form to the registrar. The registrar inspects the forms. The registrar determines that the forms have been filled out properly. The registrar informs student to attend in university overview presentation. The registrar helps the student to enrol in seminars. The registrar asks the student to pay the initial [44].
Appendix C: Results

1. Login and Register Page
   This is where students can either login or register to the Website

   ![Login and Register Page](image)

   1.1 Data Validation
   Example of a student entering incorrect data and the error message being displayed

   ![Data Validation](image)

2. Home Page
   This home page is simple yet elegant. The images that helped students navigate on the website. They have been aligned using CSS3

   ![Home Page](image)
3. Interactive Tutorial Page
This is where students have to flip the page over like a book from right to left. This was implemented using Flipboard API.

3.1 Use Case Diagram Page
The Use Case Diagram page following the learning process explained in Chapter 6, Section 6.4 Interactive Tutorial

3.1.1 Draw Use Case Exercise
Example of the Draw Use Case Exercise and drawing a potential use case on stage based on the analogy and showing the possible correct answers.
3.1.2 Key Notation Exercise  Example of all the key notations objects being dragged from the table above to the table below

3.1.3 Use Case Diagram Tutorial Video  
Example of the example video being played

3.1.4 Capturing Case Diagram Exercise  
Example of student capturing use cases from the given scenario into the text boxes. The ‘See Answer’ button after the three text boxes have been filled

3.1.5 Hangman Game  
Example of the Hangman being played and the ‘X’ appearing to identify how many wrong guesses the student has had
3.2 Activity Diagram Page

The layout of the Activity Diagram page

3.2.1 Word Search Game

Example of the Word Search game being played. Once students find a word it turn screen on the grid and grey on the word list

4. Workshop Page

A well-spaced out game page with clear title of the name of the names
4.1 MCQ Game

Example of the MCQ game being played on a Fancybox to reduce page load time

![MCQ Game Example](image)

4.2 Fill-in-the-Blanks Game

Example of Fill-in-the-Blanks game being played. Students can click on the ‘Hint’ button to get help and click on ‘Check’ to see if they got the answer right which turn right or red and a score is provided.

![Fill-in-the-Blanks Example](image)

4.3 Use Case Spot the Error Game

Example of the use case game being played. Student has identified an error a questionnaire appears on the side. The student gets the answer right and it informs the student that there are still four errors remaining. Once the game is completed, it notifies the student about the score they have received.

![Use Case Spot the Error Example](image)
4.4 Activity Central Game
The Final Game that was implemented. This allowed the student to drag and drop the objects where they thought is right. They could also click on the ‘Hint’ button which highlights the key achievements in the scenario in red. They can click on the ‘Submit’ button where the score and the right answer will appear.

5. Forum
Example of the forum page with the three predefined categories

5.1 Creating a Topic
5.2 Topic Page

5.3 Post Page

5.4 Edit and Delete a Post
6. Feedback Page

6.1 Feedback analysis

7. Notepad
This is an example of a student being able to create a note. Once the student clicks on the ‘Save’ button the title of the notes changes corresponding to what the student entered.
Appendix D: Feedback Questionnaire

Website Usability Questions

1. How would you rate the web application on the following parameters?

<table>
<thead>
<tr>
<th></th>
<th>Very Good</th>
<th>Good</th>
<th>Reasonable</th>
<th>Poor</th>
<th>Very Poor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look and feel of the website</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization of the content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ease of finding information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content (clarity of language used, spelling and grammar)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to navigate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.1. Is there anything you would like to suggest?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Learning Experience

<table>
<thead>
<tr>
<th></th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither agree nor disagree</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interactive Tutorial Page</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive Tutorial helped you understand the purpose of both UC and AD?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive Tutorial helped you understand when and how to apply UC and AD?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concepts were introduced well for both UC and AD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material successfully tied up well with the Software engineering Module?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The forum will help you get the answers you need when you don’t understand something?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Game Page</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The range of tutorials, exercises, games and videos enhanced your understanding?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exercises and Games developed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Making Notes
Making notes helped you recall information quickly?

| were easy to play and understand? | | | |
| The exercises and games developed will help you learn and revise for exams? | | | |
| The games developed tied in well with the Interactive Tutorial? | | | |
| The feedback provided was constructive on each exercise and game? | | | |
| The line chart was a helpful feedback method that motivated you to do better next time? | | | |

2.1. Is there anything you would like to suggest?

_________________________________________________________________
_________________________________________________________________
_________________________________________________________________