

---

## Testing Document

### SWEN90007: Software Design and Architecture

#### The Team

SWEN90007 SM2 2020 Project

Github: <https://github.com/emilylm/swen90007-project.git>

Heroku: <https://swen90007-exam-app.herokuapp.com>

In charge of: Emily Marshall <emilylm@student.unimelb.edu.au>, emilylm, 587580  
Luke Rosa <lrosa@student.unimelb.edu.au>, lrosa, 319522

---



---

SCHOOL OF  
COMPUTING &  
INFORMATION  
SYSTEMS

---

**Revision History**

Date	Version	Description	Author
26/09/2020	01.00-D01	Initial draft	Emily Marshall Luke Rosa
27/09/2020	01.00-D02	Added initial use cases	Luke Rosa
28/09/2020	01.00-D03	Added Administrator use cases	Luke Rosa
9/10/2020	01.00-D04	Added Instructor use cases	Emily Marshall
9/10/2020	01.00-D05	Added Student use cases	Emily Marshall
10/10/2020	01.00-D06	Final review of part 2 testing document	Emily Marshall Luke Rosa
11/10/2020	01.00	Final version of part 2 testing document	Emily Marshall Luke Rosa
30/10/2020	02.00-D01	Added instructor marks exams use cases	Emily Marshall Luke Rosa
01/11/2020	02.00-D02	Final review of part 3 testing document	Emily Marshall Luke Rosa
01/11/2020	02.00	Final version of part 3 testing document	Emily Marshall Luke Rosa

## Table of contents

<b>Introduction .....</b>	<b>4</b>
1.1 Proposal .....	4
1.2 Target users .....	4
1.3 Conventions, terms and abbreviations .....	4
1.4 Glossary of synonyms .....	4
<b>2. Covered requirements .....</b>	<b>5</b>
2.1 Functional requirements summary .....	5
2.2 User authentication functional test cases .....	7
2.2.1 UC002: User logs in .....	7
2.3 Instructor and student functional test cases .....	12
2.3.1 UC003: User views list of subjects .....	12
2.3.1 UC022: User views list of subject enrolments .....	13
2.3.2 UC004: User views list of exams .....	16
2.3.3 UC005: User views an exam .....	17
2.4 Instructor functional test cases .....	19
2.4.1 UC006: Instructor creates an exam .....	19
2.4.2 UC007: Instructor publishes an exam .....	22
2.4.3 UC008: Instructor edits questions on an exam .....	23
2.4.4 UC009: Instructor adds questions to an exam .....	27
2.4.5 UC010: Instructor deletes questions on an exam .....	32
2.4.6 UC011: Instructor deletes an exam .....	36
2.4.7 UC012: Instructor manually closes an exam .....	37
2.4.8 UC013: Instructor views table view of all students and exams for a subject .....	39
2.4.9 UC014: Instructor enters exam marks via table view .....	40
2.4.10 UC015: Instructor views detailed view of student submission .....	41
2.4.11 UC016: Instructor enters exam marks via detailed view of student submission .....	42
2.5 Student functional test cases .....	44
2.5.1 UC017: Student starts and submits an exam .....	44
2.5.2 UC018: Student views results for an exam .....	52
2.6 Administrator functional test cases .....	54
2.6.1 UC019: Administrator creates new subjects .....	54
2.6.2 UC020: Administrator creates new users .....	58
2.6.3 UC021: Administrator creates new subject enrolments .....	62
<b>3. Duplicated functional test cases .....</b>	<b>66</b>
3.1.1 TC022: Instructor is unable to edit questions on an exam .....	66
3.1.2 TC023: Instructor is unable to add questions to an exam .....	66
3.1.3 TC024: Instructor is unable to delete questions on an exam .....	66
3.1.4 TC025: Instructor is unable to delete an exam .....	66
<b>4. Test data .....</b>	<b>67</b>
4.1 User data .....	67
4.2 Subject data .....	68

## Introduction

---

This document specifies the SWEN90007 project system testing data, used to test the functionality implementation.

### 1.1 Proposal

The purpose of this document is to define and present the test cases for project for The Team's Examination Application, covering the test cases for the system use cases.

### 1.2 Target users

This document is mainly designed for those responsible for executing the test cases in this project – the project team members and teaching team of SWEN90007.

### 1.3 Conventions, terms and abbreviations

This section explains the concept of some important terms that will be used throughout this document. These terms are described in the following table, presented in alphabetical order.

Term	Description
Marks	The points attributed to a question or exam
System	The online examination application under development

### 1.4 Glossary of synonyms

Synonyms
Marks, points, grades
Examination Application, system

## 2. Covered requirements

This section lists the system requirements covered in the test cases.

### 2.1 Functional requirements summary

Use case [UC] ID	Use case title	Test case [TC] ID	Test case title
001	Known exception		
002	User logs in	001	Student/Instructor is able to log in
		002	Administrator is able to log in
		003	User is unable to log in
003	User views list of subjects	004	User is able to view list of enrolled subjects
022	User views list of subject enrolments	005	User is able to view list of subject enrolments
004	User views list of exams	006	User is able to view list of exams
005	User views an exam	007	Student is able to view an exam
		008	Instructor is able to view an exam
006	Instructor creates an exam	009	Instructor is able to create an exam
007	Instructor publishes an exam	010	Instructor is able to publish an exam
008	Instructor edits questions on an exam	011	Instructor is able to edit questions on an exam
		022	Instructor is unable to edit questions on an exam
009	Instructor adds questions to an exam	012	Instructor is able to adds questions to an exam
		023	Instructor is unable to add questions to an exam
010	Instructor deletes questions on an exam	013	Instructor is able to delete questions on an exam
		024	Instructor is unable to delete questions on an exam
011	Instructor deletes an exam	014	Instructor is able to delete an exam
		025	Instructor is unable to delete an exam
012	Instructor manually closes an exam	015	Instructor is able to manually close an exam

013	Instructor views table view of all students and exams for a subject	026	Instructor is able to view table view of all students and exams for a subject
014	Instructor enters exam marks via table view	027	Instructor is able to enter exam marks via table view
015	Instructor views detailed view of student submission	028	Instructor is able to view detailed view of student submission
016	Instructor enters exam marks via detailed view of student submission	029	Instructor is able to enter exam marks via detailed view of student submission
017	Student starts and submits exam	016	Student is able to start and submit an exam
		017	Student is able to start and is unable to submit an exam
018	Student views results for an exam	018	Student is able to view marks for an exam they have attempted
019	Administrator creates new subjects	019	Administrator is able to create new subjects
020	Administrator creates new users	020	Administrator is able to create new user
021	Administrator creates new subject enrolments	021	Administrator is able to create new student subject enrolments

## 2.2 User authentication functional test cases

This section describes the test cases that cover the product requirements of the system.

There are further test cases in section 4 that have been grouped together to avoid duplication.

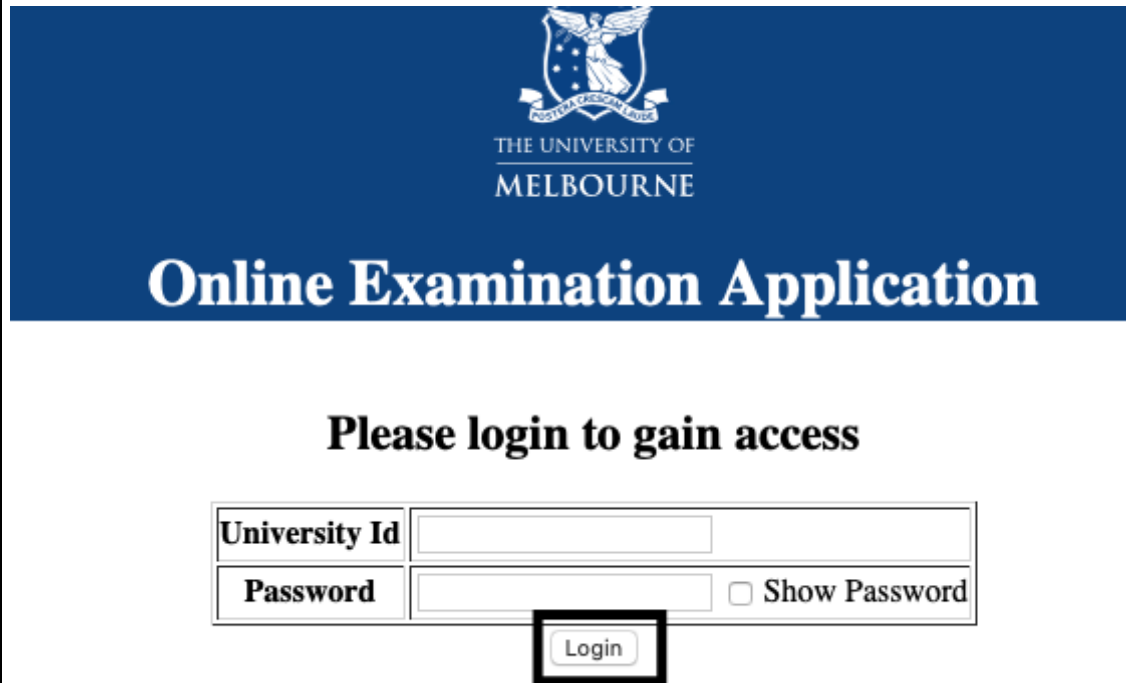
### 2.2.1 UC002: User logs in

#### 2.2.1.1 TC001: Student/Instructor is able to log in

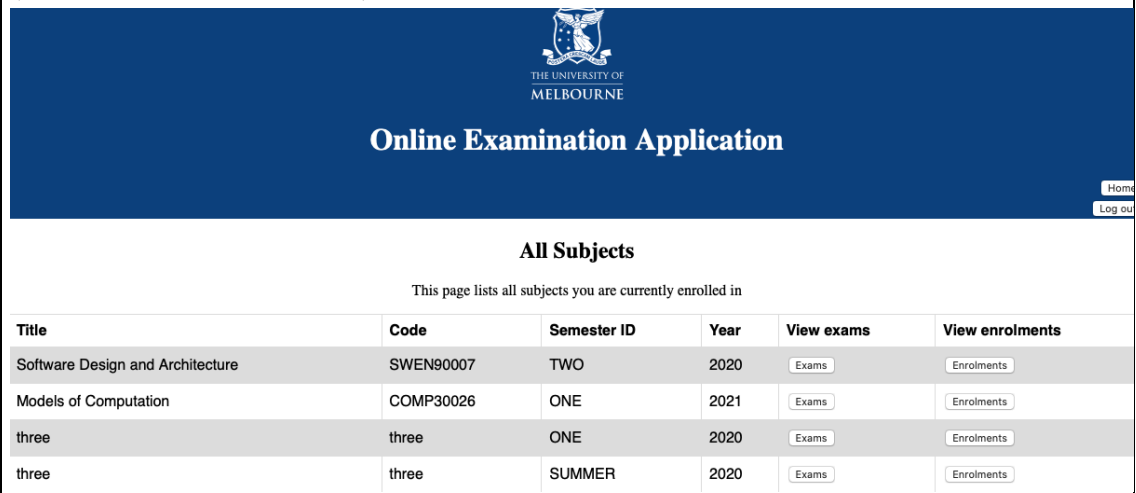
<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that login with correct is performed successfully.	
<b>Setup:</b> <ul style="list-style-type: none"><li>• User account must already exist in the system.</li><li>• User must be a student or instructor.</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• User is authenticated.</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.1 or 4.1.2 for log in details.	

**Steps:**

1. From the application home page, enter valid student or instructor credentials and enter login:



2. Once authenticated, the system will present a list of subjects the user is currently enrolled in (as either student or instructor):



Title	Code	Semester ID	Year	View exams	View enrolments
Software Design and Architecture	SWEN90007	TWO	2020	<a href="#">Exams</a>	<a href="#">Enrolments</a>
Models of Computation	COMP30026	ONE	2021	<a href="#">Exams</a>	<a href="#">Enrolments</a>
three	three	ONE	2020	<a href="#">Exams</a>	<a href="#">Enrolments</a>
three	three	SUMMER	2020	<a href="#">Exams</a>	<a href="#">Enrolments</a>

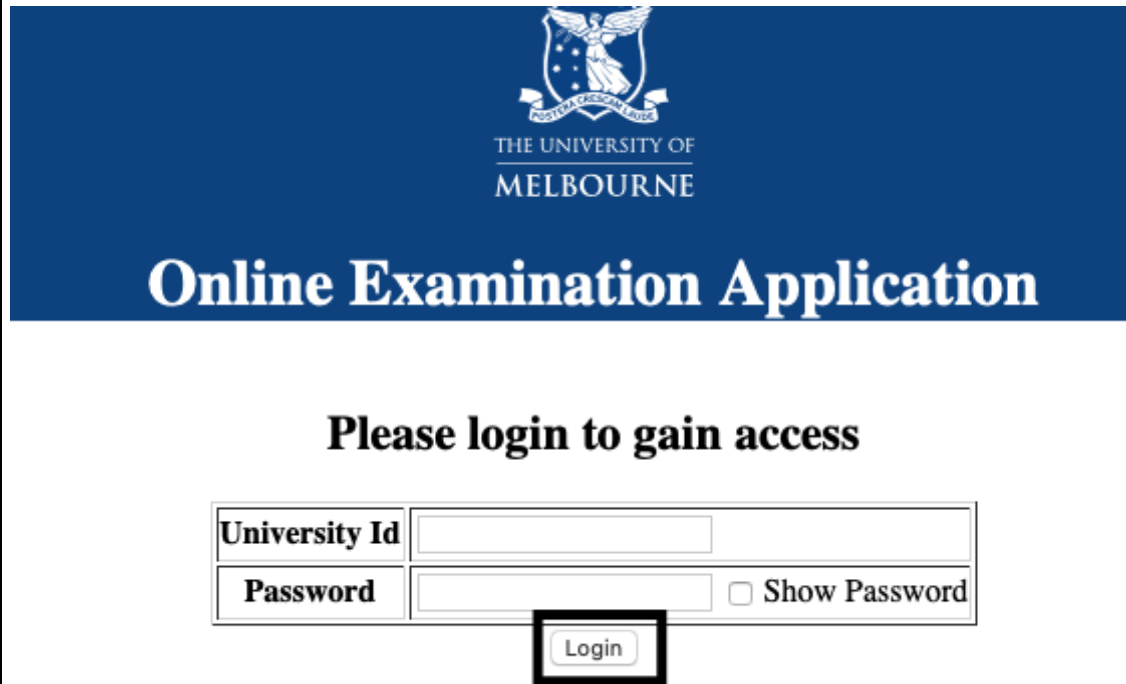


## 2.2.1.2 TC002: Administrator is able to log in

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that login with correct is performed successfully.	
<b>Setup:</b> <ul style="list-style-type: none"><li>• User account must already exist in the system.</li><li>• User must be an administrator.</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• User is authenticated.</li></ul>	
<b>Notes:</b> [1] Please refer to section 4.1.3 for log in details.	

**Steps:**

1. From the application home page, enter valid student or instructor credentials and enter login:





<b>University Id</b>	<input type="text"/>
<b>Password</b>	<input type="password"/> <input type="checkbox"/> Show Password

2. Once authenticated, the system will present the administrator home page:



<b>To view or add subjects</b>	<input type="button" value="View all subjects"/>
<b>To view or add users</b>	<input type="button" value="View all users"/>
<b>To view or add enrolments</b>	


## 2.2.1.3 TC003: User is unable to log in

<b>Test type:</b> Functional	<b>Execution type:</b> Manual				
<b>Objective:</b> Verify that the login is not performed successfully.					
<b>Setup:</b>					
<b>Post-conditions:</b> <ul style="list-style-type: none"> <li>User is not logged in</li> </ul>					
<b>Notes:</b> [1] Try to log in with invalid credentials not in sections 4 or 5.					
<b>Steps:</b> 1. From the home page, enter incorrect credentials: <div data-bbox="236 745 1366 1776" style="border: 1px solid black; padding: 10px; margin-top: 10px;">  <h2 style="text-align: center;">Online Examination Application</h2> <p style="text-align: center;"><b>Please login to gain access</b></p> <div style="display: flex; justify-content: center; align-items: flex-start;"> <table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 5px;"><b>University Id</b></td> <td style="width: 150px; height: 25px;"></td> </tr> <tr> <td style="padding: 5px;"><b>Password</b></td> <td style="width: 150px; height: 25px;"></td> </tr> </table> <div style="margin-left: 10px;"> <input type="checkbox"/> Show Password         </div> </div> <div style="text-align: center; margin-top: 5px;"> <input type="button" value="Login"/> </div> </div>		<b>University Id</b>		<b>Password</b>	
<b>University Id</b>					
<b>Password</b>					
2. The system will prompt the user that the login details are incorrect and will not authenticate user: <div style="text-align: center; margin-top: 20px;">  <h2 style="text-align: center;">Online Examination Application</h2> <p style="text-align: center;"><b>Please login to gain access</b></p> <div style="display: flex; justify-content: center; align-items: flex-start;"> <table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 5px;"><b>University Id</b></td> <td style="width: 150px; height: 25px;"></td> </tr> <tr> <td style="padding: 5px;"><b>Password</b></td> <td style="width: 150px; height: 25px;"></td> </tr> </table> <div style="margin-left: 10px;"> <input type="checkbox"/> Show Password         </div> </div> <div style="text-align: center; margin-top: 5px;"> <input type="button" value="Login"/> </div> <p style="color: red; font-weight: bold; margin-top: 10px;">Invalid username or password</p> </div>		<b>University Id</b>		<b>Password</b>	
<b>University Id</b>					
<b>Password</b>					

## 2.3 Instructor and student functional test cases

### 2.3.1 UC003: User views list of subjects

#### 2.3.1.1 TC004: User is able to view list of enrolled subjects

<b>Test type:</b>	<b>Execution type:</b>			
Functional	Manual			
<b>Objective:</b>				
Verify that student and instructor users are able to view list of subjects.				
<b>Setup:</b>				
<ul style="list-style-type: none"><li>• Student or instructor user is authenticated.</li><li>• User is enrolled in a subject.</li></ul>				
<b>Post-conditions:</b>				
<b>Notes:</b>				
[1] Please refer to sections 4.1.1 or 4.1.2 for log in details.				
<b>Steps:</b>				
1. After authenticating user, the home page presents current list of enrolled (for students) or taught (for instructor) subjects:				
<div><div><p>THE UNIVERSITY OF MELBOURNE</p></div><div><h1>Online Examination Application</h1></div><div><div>Home</div><div>Log out</div></div></div>				
<h2>All Subjects</h2>				
This page lists all subjects you are currently enrolled in				
<b>Title</b>	<b>Code</b>	<b>Semester ID</b>	<b>Year</b>	<b>View exams</b>
Software Design and Architecture	SWEN90007	TWO	2020	<div>Exams</div>
Models of Computation	COMP30026	ONE	2021	<div>Exams</div>

### 2.3.1 UC022: User views list of subject enrolments

#### 2.3.1.1 TC005: Administrator is able to view list of subject enrolments

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that administrator and instructor users are able to view list of subject enrolments.	
<b>Setup:</b> <ul style="list-style-type: none"><li>Administrator user is authenticated.</li></ul>	
<b>Post-conditions:</b>	
<b>Notes:</b> [1] Please refer to section 4.1.3 for log in details.	

**Steps:**

1. From administrator home page, user clicks *View all subjects*

## Welcome back, Admin

<b>To view or add subjects</b>  <b>To view or add users</b> <b>To view or add enrolments</b>	View all subjects
	View all users

2. User selects *Enrolments*:

### All Subjects

This page lists all subjects on the application

Add New Subjects					
Title	Code	Semester ID	Year	View exams	View enrolments
Software Design and Architecture	SWEN90007	TWO	2020	Exams	Enrolments
Models of Computation	COMP30026	ONE	2021	Exams	Enrolments

3. System presents lists of student and instructor enrolments:

### All Subject Enrolments for SWEN90007

This page lists all users enrolled in the subject



Add New Enrolments				
University ID	Role	First Name	Last name	Email
11	INSTRUCTOR	emily	marshall	instructor
33	STUDENT	biggles	tasha	student
44	STUDENT	albus	dumbledog	student2
319534	STUDENT	Luke	Rosa	lrosa@unimelb.edu.au

## 2.3.1.2 TC006: Instructor is able to view list of subject enrolments

<b>Test type:</b> Functional	<b>Execution type:</b> Manual																									
<b>Objective:</b> Verify that instructor users are able to view list of subject enrolments.																										
<b>Setup:</b> <ul style="list-style-type: none"><li>Instructor user is authenticated.</li><li>Instructor is enrolled in a subject.</li></ul>																										
<b>Post-conditions:</b>																										
<b>Notes:</b> [1] Please refer to section 4.1.2 for log in details.																										
<b>Steps:</b> 1. From instructor home page, user selects <i>Enrolments</i> : <div><div>All Subjects</div><div>This page lists all subjects on the application</div><div><div>Add New Subjects</div><table><tr><th>Title</th><th>Code</th><th>Semester ID</th><th>Year</th><th>View exams</th><th>View enrolments</th></tr><tr><td>Software Design and Architecture</td><td>SWEN90007</td><td>TWO</td><td>2020</td><td>Exams</td><td>Enrolments</td></tr><tr><td>Models of Computation</td><td>COMP30026</td><td>ONE</td><td>2021</td><td>Exams</td><td>Enrolments</td></tr></table></div></div>		Title	Code	Semester ID	Year	View exams	View enrolments	Software Design and Architecture	SWEN90007	TWO	2020	Exams	Enrolments	Models of Computation	COMP30026	ONE	2021	Exams	Enrolments							
Title	Code	Semester ID	Year	View exams	View enrolments																					
Software Design and Architecture	SWEN90007	TWO	2020	Exams	Enrolments																					
Models of Computation	COMP30026	ONE	2021	Exams	Enrolments																					
2. System presents lists of student and instructor enrolments:																										
<div><div>All Subject Enrolments for SWEN90007</div><div>This page lists all users enrolled in the subject</div><div><div>Add New Enrolments</div><table><tr><th>University ID</th><th>Role</th><th>First Name</th><th>Last name</th><th>Email</th></tr><tr><td>11</td><td>INSTRUCTOR</td><td>emily</td><td>marshall</td><td>instructor</td></tr><tr><td>33</td><td>STUDENT</td><td>biggles</td><td>tasha</td><td>student</td></tr><tr><td>44</td><td>STUDENT</td><td>albus</td><td>dumbledog</td><td>student2</td></tr><tr><td>319534</td><td>STUDENT</td><td>Luke</td><td>Rosa</td><td>lrosa@unimelb.edu.au</td></tr></table></div></div>		University ID	Role	First Name	Last name	Email	11	INSTRUCTOR	emily	marshall	instructor	33	STUDENT	biggles	tasha	student	44	STUDENT	albus	dumbledog	student2	319534	STUDENT	Luke	Rosa	lrosa@unimelb.edu.au
University ID	Role	First Name	Last name	Email																						
11	INSTRUCTOR	emily	marshall	instructor																						
33	STUDENT	biggles	tasha	student																						
44	STUDENT	albus	dumbledog	student2																						
319534	STUDENT	Luke	Rosa	lrosa@unimelb.edu.au																						

### 2.3.2 UC004: User views list of exams

#### 2.3.2.1 TC007: User is able to view list of exams

<b>Test type:</b> Functional	<b>Execution type:</b> Manual															
<b>Objective:</b> Verify that student and instructor users are able to view a list of exams.																
<b>Setup:</b> <ul style="list-style-type: none"><li>• Student or instructor user is authenticated.</li><li>• User is enrolled in a subject.</li><li>• Exam exists in the system.</li><li>• User is on the exam page.</li></ul>																
<b>Post-conditions:</b>																
<b>Notes:</b> [1] Please refer to sections 4.1.1 or 4.1.2 for log in details.																
<b>Steps:</b> 1. User clicks on 'Exams' to view the exams for the corresponding subject:																
<div><div> THE UNIVERSITY OF MELBOURNE</div><div>Online Examination Application</div><div>Home Log out</div><div>All Subjects</div><div>This page lists all subjects you are currently enrolled in</div><table><thead><tr><th>Title</th><th>Code</th><th>Semester ID</th><th>Year</th><th>View exams</th></tr></thead><tbody><tr><td>Software Design and Architecture</td><td>SWEN90007</td><td>TWO</td><td>2020</td><td>Exams</td></tr><tr><td>Models of Computation</td><td>COMP30026</td><td>ONE</td><td>2021</td><td>Exams</td></tr></tbody></table></div>		Title	Code	Semester ID	Year	View exams	Software Design and Architecture	SWEN90007	TWO	2020	Exams	Models of Computation	COMP30026	ONE	2021	Exams
Title	Code	Semester ID	Year	View exams												
Software Design and Architecture	SWEN90007	TWO	2020	Exams												
Models of Computation	COMP30026	ONE	2021	Exams												
2. The system will display a list of all published or closed exams for that subject:																
<div><div> THE UNIVERSITY OF MELBOURNE</div><div>Online Examination Application</div><div>All Exams for SWEN90007</div><table><thead><tr><th>Title</th><th>Status</th><th>View Exam</th></tr></thead><tbody><tr><td>quiz 2</td><td>PUBLISHED</td><td>View Exam</td></tr></tbody></table></div>		Title	Status	View Exam	quiz 2	PUBLISHED	View Exam									
Title	Status	View Exam														
quiz 2	PUBLISHED	View Exam														



### 2.3.3 UC005: User views an exam

#### 2.3.3.1 TC008: User is able to view an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that student and instructor users are able to view an exam.	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Student or instructor user is authenticated.</li><li>• User is enrolled in a subject.</li><li>• Exam exists in the system.</li><li>• User is on the exam page.</li></ul>	
<b>Post-conditions:</b>	
<b>Notes:</b> [1] Please refer to sections 4.1.1 or 4.1.2 for log in details.	

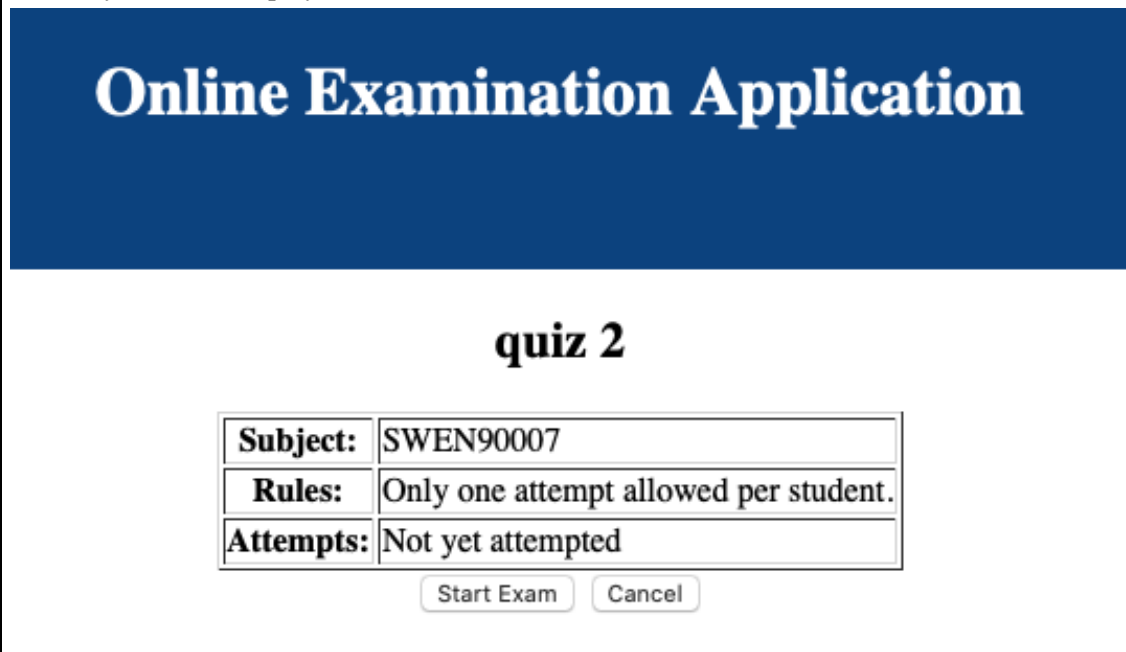
**Steps:**

1. User clicks on 'View exam'



Title	Status	
quiz 2	PUBLISHED	<a href="#">View Exam</a>

2. The system will display the exam



<b>Subject:</b>	SWEN90007
<b>Rules:</b>	Only one attempt allowed per student.
<b>Attempts:</b>	Not yet attempted

[Start Exam](#) [Cancel](#)

## 2.4 Instructor functional test cases

### 2.4.1 UC006: Instructor creates an exam

#### 2.4.1.1 TC009: Instructor is able to create an exam

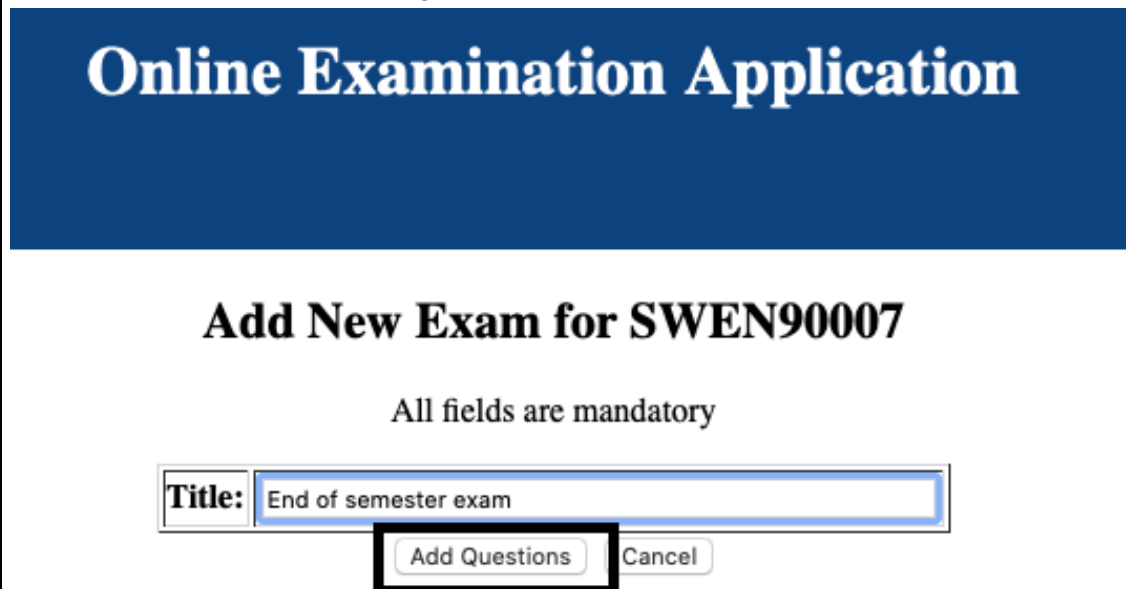
<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an instructor is able to create an exam	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Instructor user is authenticated.</li><li>• User instructs a subject.</li><li>• User is on the Exams page.</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Exam exists in the system.</li><li>• Exam is in unpublished state.</li></ul>	
<b>Notes:</b> [1] Please refer to section 4.1.2 for log in details.	

**Steps:**

1. User clicks on 'Add New Exam':



2. Enter exam title and click 'Add Questions':



3. Enter a short answer or multiple choice question

# Online Examination Application

## Add New Question to

<b>Question Type:</b>	<input checked="" type="radio"/> Short Answer <input type="radio"/> Multiple Choice
<b>Description:</b>	What is the capital of Aus?
<b>Marks:</b>	40

5. Repeat adding questions until finished

5. Save the exam

6. The system will redirect to the 'Exams' page where the newly added exam can be found. The new exam will have status 'UNPUBLISHED' and is not yet visible to students:

End of semester exam	UNPUBLISHED	false	<input type="button" value="Edit"/> <input type="button" value="View"/> <input type="button" value="Delete"/> <input type="button" value="Publish"/>
----------------------	-------------	-------	--

## 2.4.2 UC007: Instructor publishes an exam

### 2.4.2.1 TC010: Instructor is able to publish an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual								
<b>Objective:</b> Verify that an instructor is able to publish an exam.									
<b>Setup:</b> <ul style="list-style-type: none"><li>• Instructor user is authenticated.</li><li>• User instructs a subject.</li><li>• User is on the Exams page.</li><li>• Exam exists in the system.</li><li>• Exam is in unpublished state.</li></ul>									
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Exam is in published state.</li></ul>									
<b>Notes:</b> <p>[1] Please refer to sections 4.1.2 for log in details.</p>									
<b>Steps:</b> <p>1. User clicks ‘Publish’ corresponding to an exam:</p> <table><tr><td>End of semester exam</td><td>UNPUBLISHED</td><td>false</td><td><div>EditViewDeletePublish</div></td></tr></table> <p>2. The exam status is now ‘PUBLISHED’:</p> <table><tr><td>End of semester exam</td><td>PUBLISHED</td><td>false</td><td><div>EditViewDeleteClose</div></td></tr></table>		End of semester exam	UNPUBLISHED	false	<div>EditViewDeletePublish</div>	End of semester exam	PUBLISHED	false	<div>EditViewDeleteClose</div>
End of semester exam	UNPUBLISHED	false	<div>EditViewDeletePublish</div>						
End of semester exam	PUBLISHED	false	<div>EditViewDeleteClose</div>						

### 2.4.3 UC008: Instructor edits questions on an exam

#### 2.4.3.1 TC011: Instructor is able to edit questions on an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an instructor is able to edit questions on an exam.	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Instructor user is authenticated.</li><li>• User instructs a subject.</li><li>• User is on the Exams page.</li><li>• Exam exists in the system.</li><li>• Exam is in published or unpublished state.</li><li>• There have been no student attempts on the exam.</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Exam questions are edited.</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.	

**Steps:**

1. Click on '*Edit*' corresponding to an exam.

End of semester exam	PUBLISHED	false	<b>Edit</b>
			View
			Delete
			Close

2. The system will display an editing window, split across two 'Details' and 'Questions' tabs. User clicks '*Edit*' and change the details.

## Edit End of semester exam

Details Questions

**Edit**

**Title:** End of semester exam

**Subject:** SWEN90007

Back to Exams

3. Click '*Questions*' to view the question pane:

## Edit End of semester exam

Details **Questions**

Update

**Title:** End of semester exam

**Subject:** SWEN90007

Back to Exams

4. Click *Edit* on the questions you wish to edit:



## Edit End of semester exam

[Details](#)[Questions](#)[Edit](#)

### Question 0

**Question Type:** ☒ Short Answer ☐ Multiple Choice**Description:** What is the capital of Aus?**Marks:** 40[Edit](#)

### Question 1

**Question Type:** ☒ Short Answer ☐ Multiple Choice

5. Edit as many questions as required.
6. Once complete, click *Save all changes*:

## Edit End of semester exam

[Details](#)[Questions](#)[Edit](#)

### Question 0

**Question Type:** ☒ Short Answer ☐ Multiple Choice**Description:** What is the capital of Aus?**Marks:** 40[Add Question](#)[Back to Exams](#)[Save All Changes](#)[Cancel](#)

## Edit End of semester exam

[Details](#)[Questions](#)[Update](#)

<b>Title:</b>	End of semester exam
<b>Subject:</b>	SWEN90007

[Back to Exams](#)

7. The changes will be committed, the system redirects to the exams page, and displays a success message:

# Online Examination Application

## All Exams for SWEN90007

**Exam has been saved.**

**Note:** If no updates are made, there will be no option to 'Save All Changes'. If changes are cancelled, the instructor is redirected to the exams page. Re-entering the exam, it will have remained the same.

**Note:** If an instructor commences editing an exam and, before the changes are saved, a student commences the exam, all changes will be rejected by the system when the instructor clicks *Save all changes*.

#### 2.4.4 UC009: Instructor adds questions to an exam

##### 2.4.4.1 TC012: Instructor is able to add questions to an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an instructor is able to add questions to an exam.	
<b>Pre-conditions:</b> <ul style="list-style-type: none"><li>• Instructor user is authenticated</li><li>• User instructs a subject.</li><li>• User is on the Exams page</li><li>• Exam exists in the system</li><li>• Exam is in published or unpublished state</li><li>• There have been no student attempts on the exam</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Questions are added to exam.</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.	

**Steps:**

1. User clicks *Edit* on exam:

End of semester exam	PUBLISHED	false	<div><div>Edit</div><div>View</div><div>Delete</div><div>Close</div></div>
----------------------	-----------	-------	--

2. User click *Edit*:

## Edit End of semester exam

Details

Questions

Edit

Title:

End of semester exam

Subject:

SWEN90007

Back to Exams

3. User clicks *Questions*:

## Edit End of semester exam

Details

Questions

Update

Title:

End of semester exam

Subject:

SWEN90007

Back to Exams

4. User clicks *Add question*:

## Edit End of semester exam

[Details](#)[Questions](#)[Edit](#)

<b>Question 0</b>	
<b>Question Type:</b>	<input checked="" type="radio"/> Short Answer <input type="radio"/> Multiple Choice
<b>Description:</b>	What is the capital of Aus?
<b>Marks:</b>	40 <input type="text"/>

[Add Question](#)

5. Enter question and click *Save*:

## Add New Question to End of semester exam

<b>Question Type:</b>	<input checked="" type="radio"/> Short Answer <input type="radio"/> Multiple Choice
<b>Description:</b>	What is the capital of Italy?
<b>Marks:</b>	20 <input type="text"/>

[Save](#)[Cancel](#)

6. Enter as many questions as necessary by clicking *Add Question*:

## Edit End of semester exam

[Details](#)[Questions](#)[Edit](#)

### Question 0

**Question Type:** ☒ Short Answer ☐ Multiple Choice**Description:** What is the capital of Aus?**Marks:** 40[Edit](#)

### Question 1

**Question Type:** ☒ Short Answer ☐ Multiple Choice**Description:** What is the capital of Italy?**Marks:** 20[Add Question](#)

7. Once done, select *Save all changes*:

[Edit](#)

### Question 1

**Question Type:** ☒ Short Answer ☐ Multiple Choice**Description:** What is the capital of Italy?**Marks:** 20[Add Question](#)[Back to Exams](#)[Save All Changes](#)[Cancel](#)

8. The system will display a message that the exam has been updated successfully:

# All Exams for SWEN90007

Exam has been saved.

Add New Exam

## 2.4.5 UC010: Instructor deletes questions on an exam

### 2.4.5.1 TC013: Instructor is able to delete questions on an exam

<b>Test Type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that the user is able to view list of subjects	
<b>Pre-conditions:</b> <ul style="list-style-type: none"><li>• Instructor user is authenticated</li><li>• User instructs a subject</li><li>• User is on the Exams page</li><li>• Exam exists in the system</li><li>• There have been no student attempts on the exam</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Questions are deleted from the exam.</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.	



**Steps:**1. User clicks *Edit*:

End of semester exam	PUBLISHED	false	<b>Edit</b>
			View
			Delete Close

2. User clicks *Questions*:

## Edit End of semester exam

[Details](#)**[Questions](#)**

[Edit](#)

<b>Title:</b>	End of semester exam
<b>Subject:</b>	SWEN90007

[Back to Exams](#)

3. User clicks *Edit* on the question they wish to edit:

## Edit End of semester exam

[Details](#)[Questions](#)

**[Edit](#)**

<b>Question 0</b>	
<b>Question Type:</b>	<input checked="" type="radio"/> Short Answer <input type="radio"/> Multiple Choice
<b>Description:</b>	What is the capital of Aus?
<b>Marks:</b>	40 <input type="button" value="v"/>

[Edit](#)

4. User clicks *Delete*:

## Edit Question

<b>Question Type:</b>	<input checked="" type="radio"/> Short Answer <input type="radio"/> Multiple Choice
<b>Description:</b>	What is the capital of Aus?
<b>Marks:</b>	40

Delete

Update Question

Cancel

5. User repeats steps 3 and 4 for all the questions they wish to delete  
6. Once done, user clicks *Save all changes*:

## Edit End of semester exam

Details

Questions

Edit

<b>Question 0</b>	
<b>Question Type:</b>	<input checked="" type="radio"/> Short Answer <input type="radio"/> Multiple Choice
<b>Description:</b>	What is the capital of Italy?
<b>Marks:</b>	20

Add Question

Back to Exams

Save All Changes

Cancel

7. The system confirms changes have been saved:

## All Exams for SWEN90007


Exam has been saved.

Add New Exam

Status	Attempted
--------	-----------

## 2.4.6 UC011: Instructor deletes an exam

### 2.4.6.1 TC014: Instructor is able to delete an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an instructor is able to delete an exam.	
<b>Setup:</b> <ul style="list-style-type: none"> <li>• Instructor user is authenticated</li> <li>• User instructs a subject</li> <li>• Exam exists in the system</li> <li>• There have been no student attempts on the exam</li> <li>• User is on the Exams page</li> </ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"> <li>• Exam is deleted from the system.</li> </ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.	
<b>Steps:</b> 1. User clicks <i>Delete</i> next to an exam: <div data-bbox="228 1211 1359 1294" data-label="Image">  </div>	
2. Exam is deleted from the system.	

## 2.4.7 UC012: Instructor manually closes an exam

### 2.4.7.1 TC015: Instructor is able to manually close an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an instructor is able to close an exam.	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Instructor user is authenticated</li><li>• User instructs a subject</li><li>• Exam exists in the system</li><li>• Exam is in published state</li><li>• User is on the Exams page</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Exam is in published state.</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.	

**Steps:**

1. Click the 'CLOSE' button for the exam:

quiz 2	PUBLISHED	true	<a href="#">Edit</a> <a href="#">View</a> <a href="#">Delete</a> <a href="#">Close</a>
--------	-----------	------	---

2. The status of the exam is now 'CLOSED':

quiz 2	CLOSED	true	<a href="#">Edit</a> <a href="#">View</a> <a href="#">Delete</a>
--------	--------	------	--

**Note:** Logging in as a student of the subject who has not attempted the exam, the responses will be recorded as blank:

## Mid-semester exam

**Grade:** Not yet graded

<b>Question:</b>	What is a Java pattern
<b>Marks available:</b>	20
<b>Your Answer:</b>	

*You did not answer the above question.*

[Back to Exams](#)

## 2.4.8 UC013: Instructor views table view of all students and exams for a subject

2.4.8.1 TC026: Instructor is able to view table view of all students and exams for a subject

<b>Test type:</b> Functional	<b>Execution type:</b> Manual																		
<b>Objective:</b> Verify that an instructor is able to view table view of all students and exams for a subject.																			
<b>Setup:</b> <ul style="list-style-type: none"><li>Instructor user is authenticated</li><li>User instructs a subject</li><li>User is on the Subjects page</li></ul>																			
<b>Post-conditions:</b>																			
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.																			
<b>Steps:</b> 1. User selects <i>View / Edit Marks</i> : <div><div>All Subjects</div><div>This page lists all subjects you are currently instructing</div><table><tr><th>Title</th><th>Code</th><th>Semester ID</th><th>Year</th><th>Exams</th><th>Enrolments</th></tr><tr><td>Software Requirements Analysis</td><td>SWEN90007</td><td>ONE</td><td>2020</td><td><a href="#">View Exams</a></td><td><a href="#">View Enrolments</a></td></tr><tr><td colspan="4"></td><td colspan="2"><a href="#">View/Edit Marks</a></td></tr></table></div>		Title	Code	Semester ID	Year	Exams	Enrolments	Software Requirements Analysis	SWEN90007	ONE	2020	<a href="#">View Exams</a>	<a href="#">View Enrolments</a>					<a href="#">View/Edit Marks</a>	
Title	Code	Semester ID	Year	Exams	Enrolments														
Software Requirements Analysis	SWEN90007	ONE	2020	<a href="#">View Exams</a>	<a href="#">View Enrolments</a>														
				<a href="#">View/Edit Marks</a>															
2. The system presents marks for all students and all exams in the database for that subject:																			

All Student Marks for SWEN90007



Show 10 entries

Search:

Student	Mid-semester exam
Andrew Castle	-
Jackson Hamilton	-
Carmen Khor	-

## 2.4.9 UC014: Instructor enters exam marks via table view







### 2.4.9.1 TC027: Instructor is able to enter exam marks via table view

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an instructor is able to enter exam marks via the table view.	
<b>Setup:</b> <ul style="list-style-type: none"> <li>Instructor user is authenticated</li> <li>User instructs a subject</li> <li>Exam exists in the system</li> <li>User is on the table view of an exam</li> </ul>	
<b>Post-conditions:</b>	
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.	
<b>Steps:</b> <ol style="list-style-type: none"> <li>User selects <i>Edit</i> next on the row of a student exam submission:</li> </ol> <div style="text-align: center;"> <b>All Student Marks for SWEN90007</b> </div>  <ol style="list-style-type: none"> <li>The user enters the marks for the exam and presses save:</li> </ol>  <ol style="list-style-type: none"> <li>The marks are then saved in the system.</li> </ol>	









### 2.4.10 UC015: Instructor views detailed view of student submission

#### 2.4.10.1 TC028: Instructor is able to view detailed view of student submission

<b>Test type:</b> Functional	<b>Execution type:</b> Manual									
<b>Objective:</b> Verify that an instructor is able to view detailed view of a student's submission.										
<b>Setup:</b> <ul style="list-style-type: none"> <li>• Instructor user is authenticated</li> <li>• User instructs a subject</li> <li>• Exam exists in the system</li> <li>• User submission exists in the system</li> <li>• User is on the table view of an exam</li> </ul>										
<b>Post-conditions:</b>										
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.										
<b>Steps:</b> 1. User selects View on the row of a student's exam: <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center;"><b>All Student Marks for SWEN90007</b></p> <p>Show <input type="text" value="10"/> entries <span style="float: right;">Search: <input type="text"/></span></p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Student</th> <th>Mid-semester exam</th> <th></th> </tr> </thead> <tbody> <tr> <td>Andrew Castle</td> <td>-</td> <td></td> </tr> <tr> <td>Jackson Hamilton</td> <td>-</td> <td></td> </tr> </tbody> </table> </div>		Student	Mid-semester exam		Andrew Castle	-		Jackson Hamilton	-	
Student	Mid-semester exam									
Andrew Castle	-									
Jackson Hamilton	-									
2. The system displays the detailed submission: <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0; text-align: center;"> <p><b>Mid-semester exam</b></p> <p><b>Total Grade:</b> Not yet graded</p> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> <b>Question:</b> </div> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> <b>Marks available:</b> </div> <p><i>This question was not answered.</i></p> <p>Mark: <input type="text"/></p> <div style="display: flex; justify-content: center; gap: 10px; margin-top: 10px;"> <span>Cancel</span> <span>Save Changes</span> </div> </div>										

### 2.4.11 UC016: Instructor enters exam marks via detailed view of student submission

2.4.11.1 TC029: Instructor is able to enter exam marks via detailed view of student submission

<b>Test type:</b> Functional	<b>Execution type:</b> Manual						
<b>Objective:</b> Verify that an instructor is able to enter exam marks via detailed view of student submission.							
<b>Setup:</b> <ul style="list-style-type: none"> <li>• Instructor user is authenticated</li> <li>• User instructs a subject</li> <li>• Exam exists in the system</li> <li>• User submission exists in the system</li> <li>• User is on the table view of an exam</li> </ul>							
<b>Post-conditions:</b>							
<b>Notes:</b> [1] Please refer to sections 4.1.2 for log in details.							
<b>Steps:</b> <ol style="list-style-type: none"> <li>1. User selects View on the row of a student's exam:</li> </ol> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center;"><b>All Student Marks for SWEN90007</b></p> <p>Show <span style="border: 1px solid #ccc; padding: 2px 5px;">10</span> entries <span style="float: right;">Search: <input type="text"/></span></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Student</th><th style="text-align: left;">Mid-semester exam</th></tr> </thead> <tbody> <tr> <td>Andrew Castle</td><td>- </td></tr> <tr> <td>Jackson Hamilton</td><td>- </td></tr> </tbody> </table> </div> <ol style="list-style-type: none"> <li>2. The system displays the detailed submission where the user can enter marks for each exam question to mark the exam.</li> </ol>		Student	Mid-semester exam	Andrew Castle	- 	Jackson Hamilton	- 
Student	Mid-semester exam						
Andrew Castle	- 						
Jackson Hamilton	- 						

## Mid-semester exam

**Total Grade:** Not yet graded

**Question:**

**Marks available:**

*This question was not answered.*

**Mark:**

30

Cancel

Save Changes

3. Repeat this for each question in the exam.

## 2.5 Student functional test cases

### 2.5.1 UC017: Student starts and submits an exam

#### 2.5.1.1 TC016: Student is able to start and submit an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify a student is able to start and submit an exam.	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Student user is authenticated</li><li>• User is enrolled in a subject</li><li>• Exam exists in the system</li><li>• User is on the Exams page</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Student has an exam response saved in the system.</li></ul>	
<b>Notes:</b> [1] Please refer to section 4.1.1 for test data.	

**Steps:**1. User clicks *View exam*

Mid-semester exam

PUBLISHED

View Exam

2. Click *Start exam*

## Mid-semester exam

<b>Subject:</b>	SWEN90007
<b>Rules:</b>	Only one attempt allowed per student.
<b>Attempts:</b>	Not yet attempted

Start Exam

Cancel

3. Enter an answer and click *Save and next*

## Question 1 of 2

### Short Answer

10 marks available

**Description:** Explain lazy load**Response:**

Save and Next

4. Enter responses for each question

5. At any point, click Save and back to modify any responses:

## Question 2 of 2

### Short Answer

10 marks available

**Description:** Explain unit of work

**Response:**

It is a Java pattern...

Save and Back

Save and Submit Exam

6. On questions with responses, click *Edit response*:

## Question 1 of 2

### Short Answer

10 marks available

Edit Response

**Description:** Explain lazy load

**Response:**

It is a Java pattern...

Next

7. Edit the response and click *Save and next* to continue taking the exam:

## Question 1 of 2

### Short Answer

10 marks available

**Description:** Explain lazy load

**Response:**

Save and Next

8. Once all questions are answered, select Save and submit exam on the last question:

## Question 2 of 2

### Short Answer

10 marks available

**Description:** Explain unit of work

**Response:**

Save and Back

Save and Submit Exam

9. The system will display all questions and responses submitted:

## Mid-semester exam

**Grade:** Not yet graded

<b>Question:</b>	Explain lazy load
<b>Marks available:</b>	10
<b>Your Answer:</b>	It is a Java pattern...
<b>Question:</b>	Explain unit of work
<b>Marks available:</b>	10
<b>Your Answer:</b>	It is a Java pattern...

[Back to Exams](#)

**Note:** When you navigate back to a question you have already answered, the buttons change from 'Save and Back' and 'Save and Forward' to 'Back' and 'Forward'. The latter buttons do not update any information or objects in the server.

**Note:** Any answers left unanswered will be recorded as blank in the system:

## Mid-semester exam

**Grade:** Not yet graded

<b>Question:</b>	What is a Java pattern
<b>Marks available:</b>	20
<b>Your Answer:</b>	

*You did not answer the above question.*

[Back to Exams](#)



## 2.5.1.2 TC017: Student is able to start and is unable to submit an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b>	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Student user is authenticated</li><li>• User is enrolled in a subject</li><li>• Exam exists in the system</li><li>• User is on the Exams page</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• There is no saved student response in the system</li></ul>	
<b>Notes:</b> <p>[1] Start an exam while it is in state “PUBLISHED” and attempt to submit it after it has been changed to status “CLOSED” by an instructor</p> <p>[2] This cannot have data set up for it as it requires two simultaneous actions – please choose from test data and attempt to recreate.</p>	

**Steps:**

1. User clicks *View exam*

Mid-semester exam

PUBLISHED

View Exam

2. Click *Start exam*

## Mid-semester exam

<b>Subject:</b>	SWEN90007
<b>Rules:</b>	Only one attempt allowed per student.
<b>Attempts:</b>	Not yet attempted

Start Exam

Cancel

3. Enter responses (or leave blank) all answers:

4. On the final page, select *Save and submit exam*:

## Question 1 of 1

### Multiple Choice

**20 marks available****Description:** What is a lazy load?**Options:**☐ A type of camel☐ A Java pattern

Save and Submit Exam

5. During the time the student has been taking the exam, the instructor has closed the exam. The student response will not be saved and the system will present an error message that the exam has been closed and the submission has not been saved:

## All Exams for SWEN90007

Submission unsuccessful. Exam has closed.

Status
--------

## 2.5.2 UC018: Student views results for an exam

2.5.2.1 TC018: Student is able to view marks awarded for an exam they have submitted

<b>Test type:</b> Functional	<b>Execution type:</b> Manual						
<b>Objective:</b> To retrieve marks for an exam attempt.							
<b>Setup:</b> <ul style="list-style-type: none"> <li>• Student user is authenticated</li> <li>• User is enrolled in a subject</li> <li>• Student has submitted an exam which has been graded by an Instructor</li> <li>• Student is on the exams page</li> </ul>							
<b>Notes:</b> [1] Please refer to sections 4.1.3 for log in details.							
<b>Steps:</b> <ol style="list-style-type: none"> <li>1. User clicks <i>View Exam</i> for the exam they wish to see results for.</li> <li>2. User clicks <i>View Submission</i>:           <div style="text-align: center; margin: 20px 0;"> <b>Quiz 2</b>  <table border="1" style="margin: auto;"> <tr> <td><b>Subject:</b></td><td>SWEN90007</td></tr> <tr> <td><b>Rules:</b></td><td>Only one attempt allowed per student.</td></tr> <tr> <td><b>Attempts:</b></td><td>Exam already attempted</td></tr> </table> <div style="display: flex; justify-content: center; gap: 10px;"> <div style="border: 2px solid black; padding: 2px 5px;">View Submission</div> <div>Cancel</div> </div> </div> </li> <li>3. User is shown their attempt alongside any marks awarded, and a grade for the entire exam if one exists:</li> </ol>		<b>Subject:</b>	SWEN90007	<b>Rules:</b>	Only one attempt allowed per student.	<b>Attempts:</b>	Exam already attempted
<b>Subject:</b>	SWEN90007						
<b>Rules:</b>	Only one attempt allowed per student.						
<b>Attempts:</b>	Exam already attempted						

## Quiz 2

**Grade:** 25 out of 30 marks

<b>Question:</b>	Discuss the benefits and drawbacks of the Unit of Work pattern.
<b>Marks available:</b>	20
<b>Marks awarded:</b>	10
<b>Your Answer:</b>	It is good for saving only what you need.

<b>Question:</b>	Which of the following may be included in a Unit of Work implementation?
<b>Marks available:</b>	10
<b>Options:</b>	<ul style="list-style-type: none"><li><input type="radio"/> A list of new objects</li><li><input type="radio"/> A list of dirty objects</li><li><input type="radio"/> A list of deleted objects</li><li><input checked="" type="radio"/> All of the above</li></ul>

[Back to Exams](#)

## 2.6 Administrator functional test cases

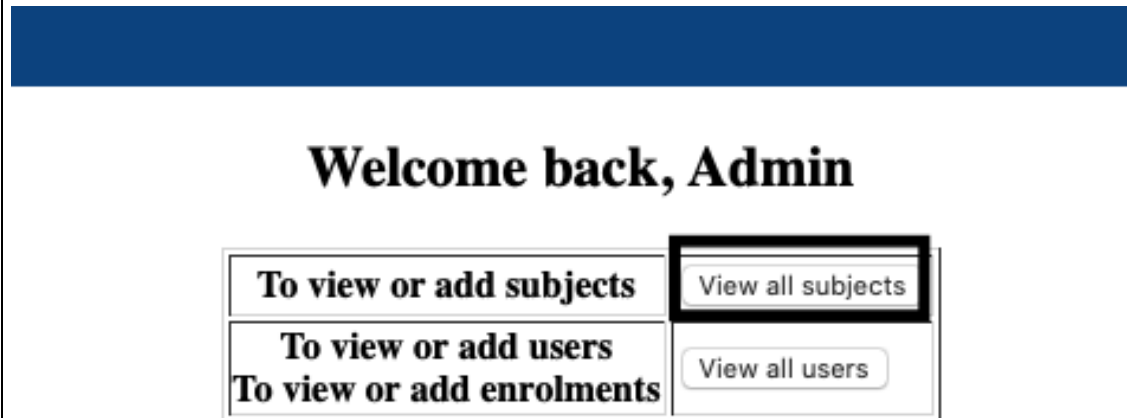
### 2.6.1 UC019: Administrator creates new subjects

#### 2.6.1.1 TC019: Administrator is able to create new subjects

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an administrator is able to create new subjects	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Administrator user is authenticated</li><li>• User is on the administrator home page</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Subject is created in the system</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.3 for log in details.	

**Steps:**

1. User clicks *View all subjects*:



2. User clicks *Add new subjects*:





3. User enters subject details and clicks *Submit*:

## Add New Subject

All fields are mandatory

[View All Subjects](#)

<b>Subject Title:</b>	COMP20003
<b>Subject Code:</b>	Algebra 2
<b>Semester:</b>	ONE 
<b>Year:</b>	2020 

[Submit](#)

4. System will prompt the user that the subject has been created:

## Add New Subject

All fields are mandatory

[View All Subjects](#)

<b>Subject Title:</b>	
<b>Subject Code:</b>	
<b>Semester:</b>	ONE 
<b>Year:</b>	2020 

[Submit](#)

Subject created

5. Click *View all subjects*:



## Add New Subject

All fields are mandatory

[View All Subjects](#)

<b>Subject Title:</b>	<input type="text"/>
<b>Subject Code:</b>	<input type="text"/>
<b>Semester:</b>	ONE 
<b>Year:</b>	2020 

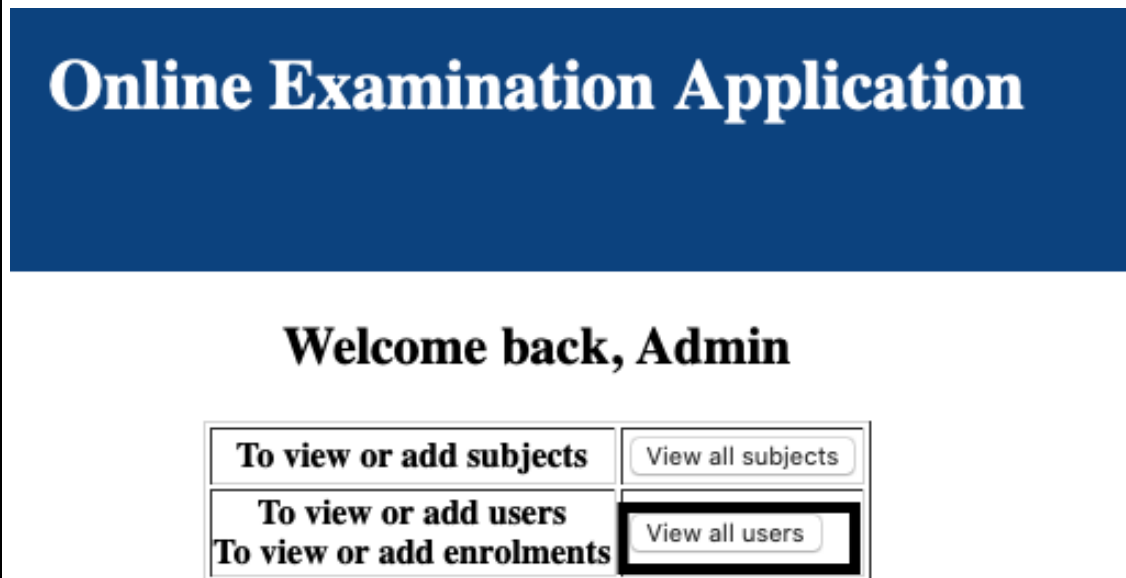
6. Created subject will appear in list of subjects:

COMP20003	Algebra 2	ONE	2020	<a href="#">Exams</a>	<a href="#">Enrolments</a>
-----------	-----------	-----	------	-----------------------	----------------------------

## 2.6.2 UC020: Administrator creates new users

### 2.6.2.1 TC020: Administrator is able to create new user

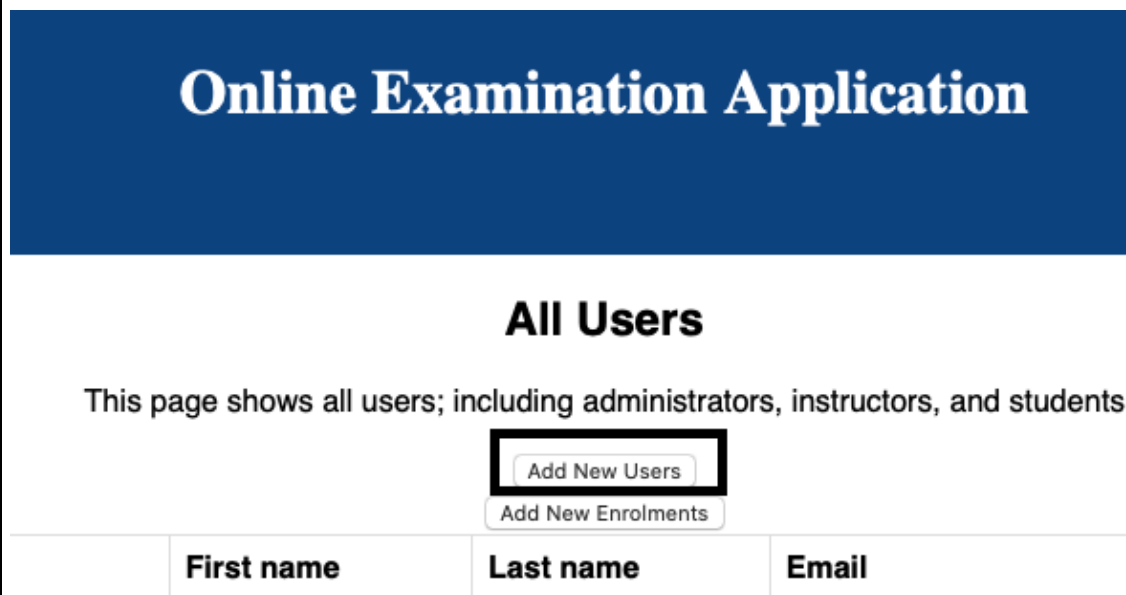
<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an administrator is able to create new users	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Administrator user is authenticated</li><li>• User is on the administrator home page</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• User is created in the system</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.3 for log in details.	

**Steps:**1. User clicks *View all users*:

**Online Examination Application**

**Welcome back, Admin**

To view or add subjects	<input type="button" value="View all subjects"/>
To view or add users To view or add enrolments	<input type="button" value="View all users"/>

2. User clicks *Add new users*:

**Online Examination Application**

**All Users**

This page shows all users; including administrators, instructors, and students

First name	Last name	Email
------------	-----------	-------

3. Enter user details and select role from the drop down:

## Add New User

Add new administrator, instructor or student users here  
All fields are mandatory

View All Users

University ID	319522
Role	<div>✓ ADMIN STUDENT INSTRUCTOR</div>
First name	
Last name	Rosa

4. Select *Create new user*:

## Add New User

Add new administrator, instructor or student users here  
All fields are mandatory

View All Users

University ID	319522
Role	ADMIN
First name	Luke
Last name	Rosa
Email	lukearosa@me.com
Password	.... <input type="checkbox"/> Show Password


Create New User

5. The system will prompt that the user has been created:

## Add New User

Add new administrator, instructor or student users here  
All fields are mandatory

[View All Users](#)

University ID	<input type="text"/>
Role	ADMIN 
First name	<input type="text"/>
Last name	<input type="text"/>
Email	<input type="text"/>
Password	<input type="password"/> <input type="checkbox"/> Show Password

[Create New User](#)


User created

6. User clicks *View all users*:

## Add New User

Add new administrator, instructor or student users here  
All fields are mandatory

[View All Users](#)

University ID	<input type="text"/>
Role	ADMIN 
First name	<input type="text"/>
Last name	<input type="text"/>
Email	<input type="text"/>
Password	<input type="password"/> <input type="checkbox"/> Show Password

[Create New User](#)

User created

7. User is visible in the list of users:

319522	ADMIN	Luke	Rosa	lukearosa@me.com	<a href="#">View / Edit</a>
--------	-------	------	------	------------------	-----------------------------

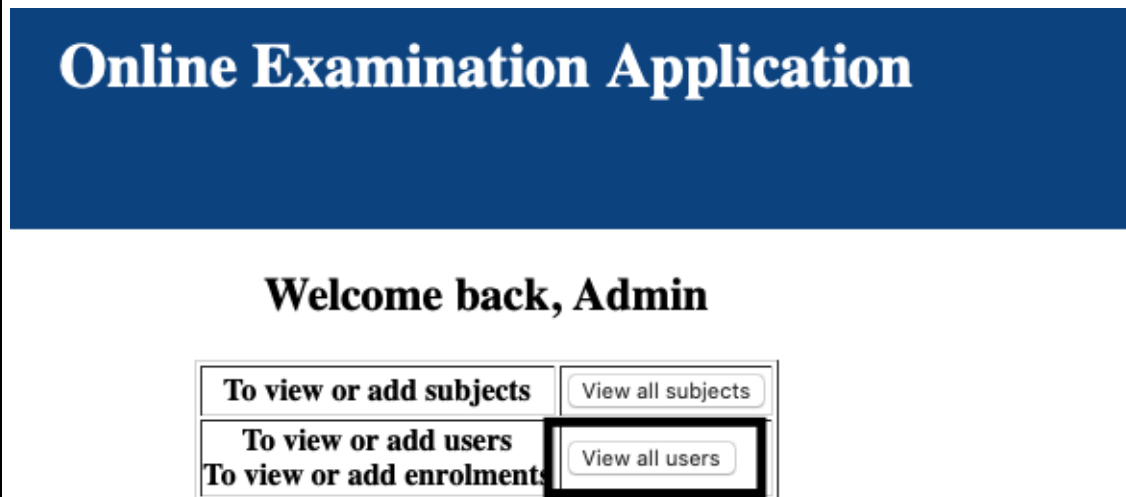
### 2.6.3 UC021: Administrator creates new subject enrolments

#### 2.6.3.1 TC021: Administrator is able to create new subject enrolments

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that an administrator is able to create new subject enrolments.	
<b>Setup:</b> <ul style="list-style-type: none"><li>• Administrator user is authenticated</li><li>• User is on the administrator home page</li></ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>• Subject enrolment exists in the system</li></ul>	
<b>Notes:</b> [1] Please refer to sections 4.1.3 for log in details.	

**Steps:**

1. User clicks *View all users*:

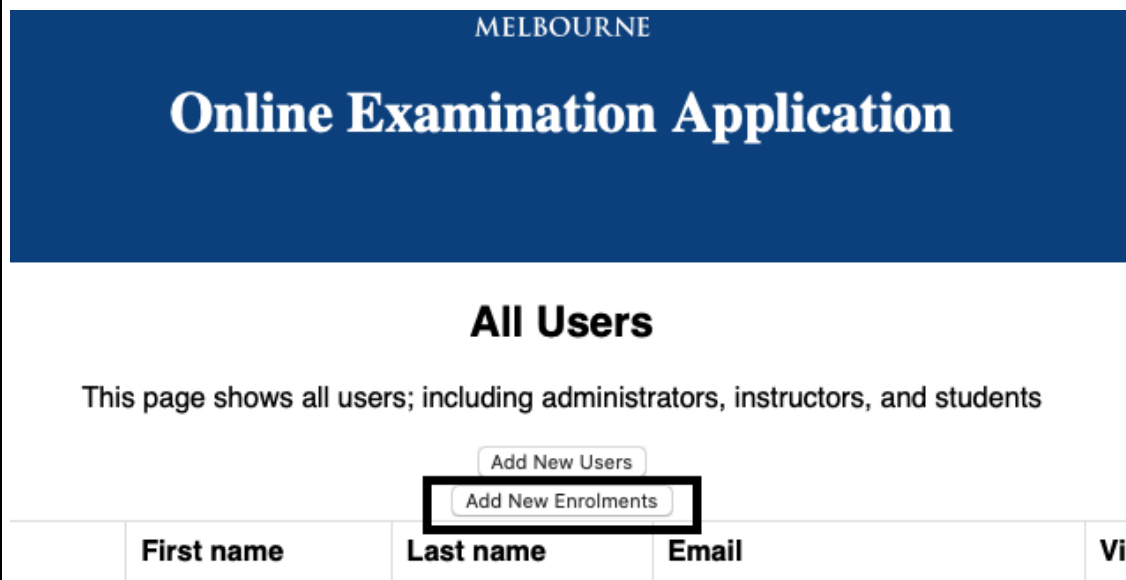


**Online Examination Application**

**Welcome back, Admin**

To view or add subjects	View all subjects
To view or add users	<b>View all users</b>
To view or add enrolments	

2. User clicks *Add new enrolments*:



MELBOURNE

**Online Examination Application**

**All Users**

This page shows all users; including administrators, instructors, and students

Add New Users

**Add New Enrolments**

	First name	Last name	Email	View
--	------------	-----------	-------	------

3. User enters student and subject details and clicks *Create new enrolments*:

## Add New Enrolment

Create student or instructor enrolments on this page  
All fields are mandatory

[View All Users](#)

University Id:	319534
Subject Code:	SWEN90007
Semester:	ONE
Year:	2020

[Create New Enrolment](#)

4. System will prompt user that the enrolment was successful:

## Add New Enrolment

Create student or instructor enrolments on this page  
All fields are mandatory

[View All Users](#)

University Id:	
Subject Code:	
Semester:	ONE
Year:	2020

[Create New Enrolment](#)

**Enrolment successful**

5. User selects *View all users*:



## Add New Enrolment

Create student or instructor enrolments on this page  
All fields are mandatory

[View All Users](#)

University Id:	<input type="text"/>
Subject Code:	<input type="text"/>

6. User selects *View / edit* in enrolments column:

319534	STUDENT	Luke	Rosa	lrosa@unimelb.edu.au	<a href="#">View / Edit</a>
--------	---------	------	------	----------------------	-----------------------------

7. The system presents the user's enrolments:

Home  
Log out

### All Enrolments

[Add New Enrolment](#)

Title	Code	Semester	Year
Software Design and Architecture	SWEN90007	TWO	2020

### 3. Duplicated functional test cases

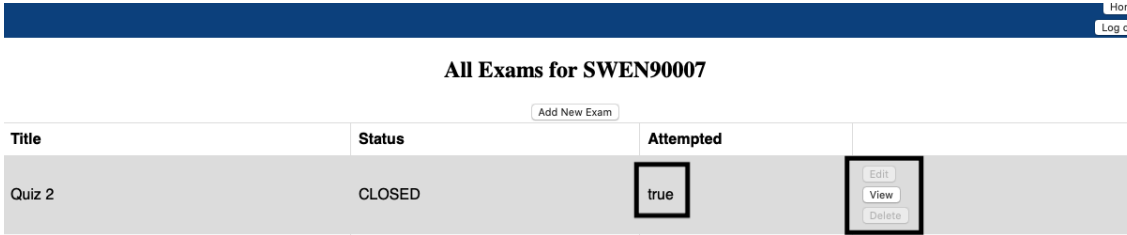
This section describes the general test cases that will be related to more than one requirement or assumption/constraint, avoiding data duplication.

#### 3.1.1 TC022: Instructor is unable to edit questions on an exam

#### 3.1.2 TC023: Instructor is unable to add questions to an exam

#### 3.1.3 TC024: Instructor is unable to delete questions on an exam

#### 3.1.4 TC025: Instructor is unable to delete an exam

<b>Test type:</b> Functional	<b>Execution type:</b> Manual
<b>Objective:</b> Verify that the user is unable to make any amendments to an exam once it has one or more student responses.	
<b>Setup:</b> <ul style="list-style-type: none"> <li>• Instructor user is authenticated</li> <li>• User instructs a subject</li> <li>• Exam exists in the system</li> <li>• Exam is in published or closed state</li> <li>• Exam has one or more student responses</li> <li>• User is on the Exams page</li> </ul>	
<b>Post-conditions:</b> <ul style="list-style-type: none"> <li>• Exam is unchanged</li> </ul>	
<b>Notes:</b> [1] Try to edit questions for an exam that has student responses. [2] Please refer to sections 4.1.2 for log in details.	
<b>Steps:</b> 1. For exams that have been started by students (with or without a response), the attempted column will display true. The system will grey out the options to edit, add or delete questions on the exam and to delete the exam:	
	

## 4. Test data

To support testing, please find data that has been used to populate the database. Test data for instructor and student enrolments, exams, grades, responses, etc have also been populated and are viewable by logging into the system as a user below.

### 4.1 User data

Student ID	First name	Last name	Email	Password	Role
crosa	Christina	Rosa	c.rosa@unimelb.edu.au	DfeRT	ADMIN
jsmith	Jane	Smith	j.smith@unimelb.edu.au	g3S2qR	STUDENT
aklein	Angie	Klein	a.klein@student.unimelb.edu.au	5ufFqx	STUDENT
marterton	Matthew	Arterton	m.arterton@student.unimelb.edu.au	bFe9qD	STUDENT
lrosa	Luke	Rosa	l.rosa@student.unimelb.edu.au	WJq9h9	STUDENT
ewilliams	Emma	Williams	e.williams@student.unimelb.edu.au	kJ3aY5	STUDENT
cwilliamson	Clair	Williamson	c.williamson@student.unimelb.edu.au	7nwCPM	STUDENT
bpitt	Brad	Pitt	b.pitt@student.unimelb.edu.au	Q8Rdak	STUDENT
jstuart	Jennifer	Stuart	j.stuart@student.unimelb.edu.au	wvRe6E	STUDENT
dmahn	Dohber	Mahn	d.mahn@student.unimelb.edu.au	907GH	STUDENT
cwawah	Chi	Wahwah	c.wahwah@student.unimelb.edu.au	lO0pd	STUDENT
cstewart	Claire	Stewart	c.stewart@unimelb.edu.au	V5K9xc	INSTRUCTOR
wbottel	Walter	Bottel	w.bottel@unimelb.edu.au	RnefG	INSTRUCTOR
scairn	So-Da	Cairn	s.cairn@unimelb.edu.au	Sgf56	INSTRUCTOR
mbimba	Mark	Bimba	m.bimba@unimelb.edu.au	Vcg5Lk	INSTRUCTOR
acastle	Andrew	Castle	a.castle@unimelb.edu.au	H52Mew	INSTRUCTOR
mridolfi	Marcel	Ridolfi	m.ridolfi@unimelb.edu.au	zTT2QY	INSTRUCTOR
jhamilton	Jackson	Hamilton	j.hamilton@unimelb.edu.au	e4mWtE	INSTRUCTOR
kly	Kaylin	Ly	k.ly@unimelb.edu.au	U7NnLa	INSTRUCTOR

ckhor	Carmen	Khor	c.khor@unimelb.edu.au	2czGTP	INSTRUCTOR
cskourletos	Christina	Skourletos	c.skourletos@unimelb.edu.a	qVz7Mf	INSTRUCTOR

#### 4.2 Subject data

Subject title	Subject code	Semester	Year
Software Requirements Analysis	SWEN90007	ONE	2020
Software Modelling and Design	SWEN90010	TWO	2020
Models of Computation	COMP30025	SUMMER	2020
Database Systems	COMP20010	ONE	2020
Introduction to Databases	COMP10010	TWO	2020
Introduction to Mathematics	MAST10010	TWO	2020
Algebra 1	MAST10030	ONE	2020
Algebra 2	MAST20030	ONE	2020
Calculus 1	MAST10025	ONE	2020
Calculus 2	MAST20050	TWO	2020