

# Student Submission Website Design Document

## Group 0

**Members:**

Conor Ryan 17666123

Aidan Murphy 17123312

Sheraz Anjum 17123456

## Introduction

The Student Submission Website (SSW) is an online portal for collating and maintaining group work for students in UL. It provides a way for students to submit weekly reports about their progress, to keep track of their current share of the grade and for faculty members to examine the reports.

## High Level Functionality

There are two key user types in SSW, **students** and **faculty**. Students will be able to log into the system, upload weekly forms, read previous forms and check their current share of the grade. They will also be able to maintain their account by changing their passwords.

Faculty will use SSW to view these reports, specifically to be able to view the reports prior to weekly meetings. They will also be able to view past reports and check the breakdown of grades given to each student in the group. In the event that a report is not submitted, the student responsible will be reported to the faculty so that this can be discussed in the meeting.

# Web Page Mockups

## CS4014/CS4065/ET4243 User Login

ID Number:   
Password:

---

### Notes

Notices and bug reports will be put here. The bug bounty also extends to feature suggestion; that is, if you suggest a new feature for the system that we subsequently use, you will get the extra credit.

### FAQ

- Q: When can I log in?  
• A: Any time from Week 1
- Q: What ID number do I use?  
• A: Your normal college ID
- Q: What is my password?  
• A: Your ID number
- Q: I can't log in and something is due. What should I do?  
• A: Mail it to Conor, making sure to include the [CS4014] tag in the subject line, along with the deliverable name. Make sure you explain that it hasn't been uploaded.

[Back to home page](#)

### Figure 1 Login Screen

## CS4075 User Portal (Week 3)

### CONOR RYAN

### Group 25

Group members

RYAN CONOR (17666123)

MURPHY AIDAN (17123312)

ANJUM SHERAZ (17123456)

### Group Leader Assignments

Week 5, Leader: 17123456

Week 6, Leader: 17666123

Week 7, Leader: 17123312

Week 8, Leader: 17123456

Week 9, Leader: 17666123

Week 10, Leader: 17123312

Week 11, Leader: 17123456

Week 12, Leader: 17666123

[Change password](#)

[Logout](#)

Figure 2 Main Student Screen. No grades are shown as students haven't yet uploaded forms.

# CS4014 User Portal (Week 8)

**BENJAMIN SMITH**

## Entering the weekly form

---

[Back to main portal page](#)

You are updating a weekly report. The previous update was by BENJAMIN SMITH.

Member	Share
MURPHY CIAN	<input type="text" value="25"/>
RYAN MATTHEW	<input type="text" value="0"/>
O'SULLIVAN ALAN	<input type="text" value="25"/>
GRIMES BENJAMIN	<input type="text" value="25"/>
KENT AMY	<input type="text" value="25"/>

## Report for work completed this week (week 8)

This week we met twice and discussed how to fix the login problem. We've decided to use sessions rather than cookies, and Benjamin will produce the code for this. The rest of the work will be...

Figure 3 Updating a weekly form

## CS4014 User Portal (Week 8)

### BENJAMIN SMITH

Group members

MURPHY CIAN (16XXXXXX)

RYAN MATTHEW (16YYYYYY)

CONDELL ALAN (16ZZZZZZ)

SMITH BENJAMIN (16AAAAAA)

KENT AMY (16BBBBBB)

### Group Leader Assignments

Week 8, Leader: 16XXXXXX

Week 9, Leader: 16YYYYYY

Week 10, Leader: 16ZZZZZZ

Week 11, Leader: 16AAAAAA

Week 12, Leader: 16BBBBBB

### Shared marks so far (up to and including week 7)

Student	Week 4	Week 5	Week 6	Week 7
MURPHY CIAN	20	20	18	20
RYAN MATTHEW	20	20	18	20
O'SULLIVAN ALAN	20	20	20	20
SMITH BENJAMIN	20	20	22	20
KENT AMY	20	20	22	20

[Change password](#)

[Enter the weekly form](#)

[View current uploaded form for week 8 \(uploaded by BENJAMIN SMITH\).](#)

[Show all weekly reports so far](#)

[Logout](#)

Figure 4 Main student screen showing grades so far.

## CS4014 User Portal (Week 8)

### BENJAMIN SMITH

Showing all forms (starting in week 4 and finishing in week 7)

Week	16XXXXXX	16YYYYYY	16ZZZZZZ	16AAAAAA	16BBBBBB	Report
4	20	20	20	20	20	<a href="#">Show</a>
5	20	20	20	20	20	<a href="#">Show</a>
6	18	18	20	22	22	<a href="#">Show</a>
7	20	20	20	20	20	<a href="#">Show</a>

[Back to main portal](#)

Figure 5 List all forms submitted so far.

## Group No. 5

### Weekly Report for Week 7

---

Author: PATRICK KEANE

---

This week we started working on a level in creation kit. We had a meeting about the Level Design that we submitted last week. The meeting pointed out problems that we needed to address. After the meeting, we had a playtest with the other group, the playtest was useful as it pointed out the good things in our game and the bad things, it gave us few ideas that we didn't think of before. We are starting to work on an improved version of our Level Design and start making one level for the prototype.

TED WEST 13XXXXXX 25

LIAM POWER 16YYYYYY 25


IAN DUGGAN 16ZZZZZZ 25

PATRICK KEANE 16AAAAAA 25

HENRY SULLIVAN 16BBBBBB 0

**View another group:**

Select a group to view

1  Submit

---

Latest and this are 6 and 7.

Showing all forms (starting in week 4 and finishing in week 6)

Week	13113097	16148983	16151135	16155599	16172639	Report
4	25	25	25	25	0	<a href="#">Show</a>
5	20	20	20	20	20	<a href="#">Show</a>
6	20	20	20	20	20	<a href="#">Show</a>

[Back to main portal](#)

Figure 6 Faculty View of Weekly Reports

## Database Tables

**Table 1 Student Table.** This table describes each student and the group they belong to.

Field	Type	Primary
StudentID	Varchar(10)	Yes
Pass	Binary(2)	
Firstname	Varchar(16)	
Surname	Varchar(16)	
GroupID	Int(11)	

**Table 2 The weekly report table.** The primary key is enforced through a combination of (weekNumber, groupNumber), while the studentID is the ID number of the student who entered the report. Detail contains the actual report. Note that the breakdown of marks is stored in the weeklyShare table, below.

Field	Type	Primary
weekNumber	Int(11)	Yes
groupNumber	Int(11)	Yes
studentID	Varchar(10)	Foreign Key
Detail	text	

**Table 3 The weekly share table;** each entry is for one particular student. The primary key is enforced through a combination of (weekNumber, groupNumber, studentID).

Field	Type	Primary
weekNumber	Int(11)	Yes
groupNumber	Int(11)	Yes
studentID	Varchar(10)	Yes; Foreign Key
Share	Int(11)	

**Table 4 The controls table.** This only ever has a single entry and is used to keep track of the current week number.

Field	Type	Primary
weekNumber	Int(11)	

## Process List

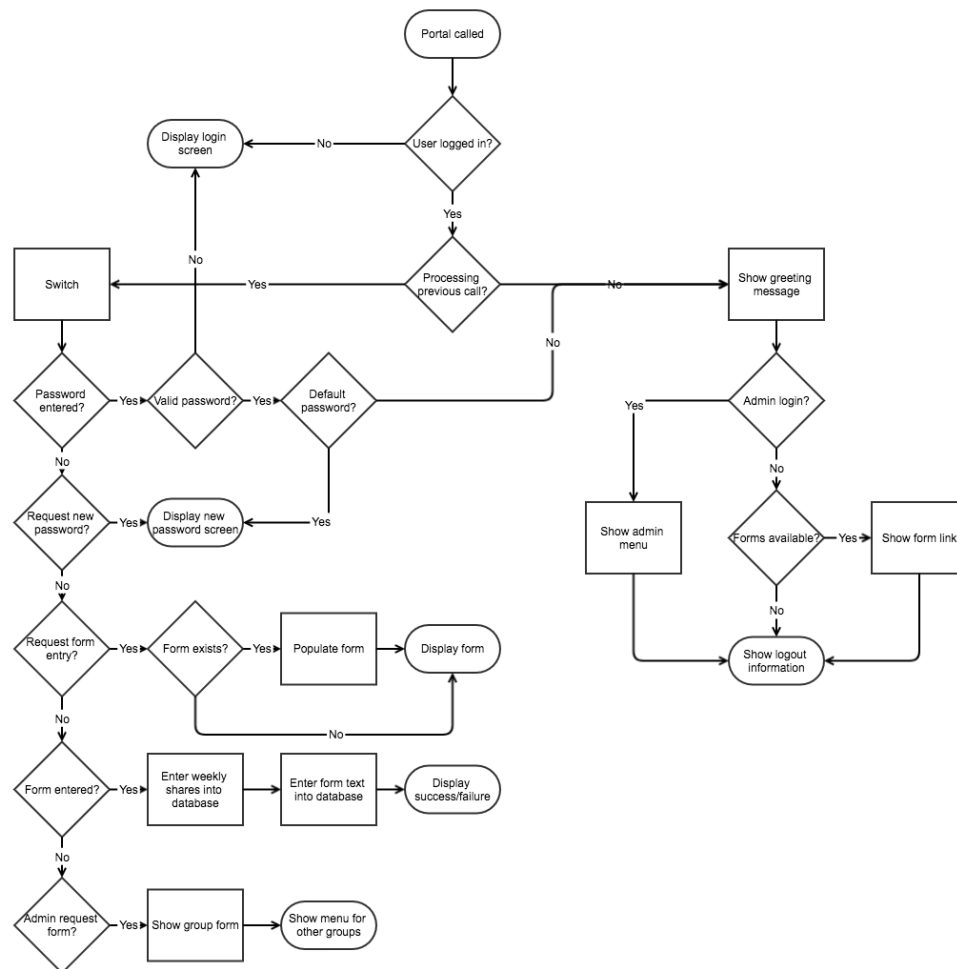


Figure 7 Flowchart for system

<b>Process #</b>	1
<b>Title</b>	Get Group Members
<b>Brief Descr.</b>	List the group members
<b>Inputs</b>	Group number
<b>Detailed Descr.</b>	Create an array of student members, consisting of firstname, surname and student ID
<b>Output</b>	A 2D array of student details



<b>Process #</b>	2
<b>Title</b>	Find Group Leader
<b>Brief Descr.</b>	Identify who is the group leader for a particular week
<b>Inputs</b>	Week number, group
<b>Detailed Descr.</b>	Generate the list of group leaders and identify who is this week's one
<b>Output</b>	A student ID number

<b>Process #</b>	3
<b>Title</b>	Get Week
<b>Brief Descr.</b>	Find out what week this is
<b>Inputs</b>	None
<b>Detailed Descr.</b>	Return the week number from the controls
<b>Output</b>	The current week number

<b>Process #</b>	4
<b>Title</b>	Update password
<b>Brief Descr.</b>	Update a password
<b>Inputs</b>	Student ID, new password
<b>Detailed Descr.</b>	Update the password in the student table
<b>Output</b>	None

<b>Process #</b>	5
<b>Title</b>	Get user first name
<b>Brief Descr.</b>	Return a user's first name
<b>Inputs</b>	Student ID
<b>Detailed Descr.</b>	Query the student table for a first name
<b>Output</b>	The user's first name

<b>Process #</b>	6
<b>Title</b>	Get user surname
<b>Brief Descr.</b>	Return a user's surname
<b>Inputs</b>	Student ID
<b>Detailed Descr.</b>	Query the student table for a surname
<b>Output</b>	The user's surname

<b>Process #</b>	7
<b>Title</b>	Get Group Size
<b>Brief Descr.</b>	Return the number of students in a group
<b>Inputs</b>	Group number
<b>Detailed Descr.</b>	Identify the size of the group
<b>Output</b>	The number of students in a group

<b>Process #</b>	8
<b>Title</b>	Check Force Change
<b>Brief Descr.</b>	Check if the user needs to change their password
<b>Inputs</b>	Student ID
<b>Detailed Descr.</b>	Query the student table to check if the user still has their default password. If they do, call the changepassword() function (Process #9)
<b>Output</b>	None

<b>Process #</b>	9
<b>Title</b>	Change password
<b>Brief Descr.</b>	Show a form for changing the password
<b>Inputs</b>	None
<b>Detailed Descr.</b>	Create a form to prompt the user for their current password and new password, making sure they retype the new password. This will be processed by Process #4.
<b>Output</b>	None

<b>Process #</b>	10
<b>Title</b>	Make choice
<b>Brief Descr.</b>	Choose what functionality the user wants
<b>Inputs</b>	\$_REQUEST array
<b>Detailed Descr.</b>	This is the main control flow of the system. It will use the \$_REQUEST array to figure out what choice the user made on the previous menu and then call the appropriate function, e.g. change password, enter form, etc.
<b>Output</b>	None

<b>Process #</b>	11
<b>Title</b>	Log user in
<b>Brief Descr.</b>	Log a user in
<b>Inputs</b>	Student ID, Password
<b>Detailed Descr.</b>	Use the check password function (Process #12) to verify the password. If it is correct, create a session, if not, report the error.
<b>Output</b>	None

<b>Process #</b>	12
<b>Title</b>	Check password
<b>Brief Descr.</b>	Verify user password
<b>Inputs</b>	Student ID, Password
<b>Detailed Descr.</b>	Query the database to verify the user has entered the correct password
<b>Output</b>	Boolean

<b>Process #</b>	13
<b>Title</b>	Show Menu
<b>Brief Descr.</b>	Show the actions that a user can select
<b>Inputs</b>	Student ID, week number, group number
<b>Detailed Descr.</b>	Show the various actions a user can perform (change password etc.) and check if they can enter a form. If older forms exist, show a link to view these. If the user is a faculty, then call process 14 instead. The link clicked on the list given will be identified by Process #10.
<b>Output</b>	None

<b>Process #</b>	14
<b>Title</b>	Show Admin Menu
<b>Brief Descr.</b>	Show faculty/admin actions
<b>Inputs</b>	Week number
<b>Detailed Descr.</b>	Show all the actions a faculty can perform, such as change a user's password or view a group
<b>Output</b>	None.

<b>Process #</b>	15
<b>Title</b>	Show header
<b>Brief Descr.</b>	Show website headers
<b>Inputs</b>	Week number
<b>Detailed Descr.</b>	Output the standard HTML for the site (including links to CSS etc.) and a friendly note at the top showing the week
<b>Output</b>	None

<b>Process #</b>	16
<b>Title</b>	Validate login
<b>Brief Descr.</b>	Check that a user is actually logged in
<b>Inputs</b>	\$_SESSION
<b>Detailed Descr.</b>	Check if the session is still valid, if not, make the user log in again using Process #17
<b>Output</b>	None

<b>Process #</b>	17
<b>Title</b>	Report not logged in
<b>Brief Descr.</b>	Show an error message and redirect the user
<b>Inputs</b>	None
<b>Detailed Descr.</b>	Show an error message indicating that the user doesn't appear to be logged in. In ten seconds, redirect to the login page.
<b>Output</b>	None.

<b>Process #</b>	18
<b>Title</b>	Show Group
<b>Brief Descr.</b>	Show the members of a group
<b>Inputs</b>	Group Number
<b>Detailed Descr.</b>	Get a list of the group members from Process #1 and display it on the page.
<b>Output</b>	A table containing all the members of a group.

<b>Process #</b>	19
<b>Title</b>	Show Weekly Form
<b>Brief Descr.</b>	Create a HTML form for a weekly report
<b>Inputs</b>	Student ID
<b>Detailed Descr.</b>	Check if a form already exists for this week, if so, populate everything with this. Create the key parts of the form, that is, the list of students and mark (if it exists) and then the text part (populating it if it exists). This will create the form header using Process #21, the form itself using Process #22 and the footer (submit button etc.) with Process #23. It also calls Process #20 to generate some group-specific JavaScript code.
<b>Output</b>	A form to be filled out.

<b>Process #</b>	20
<b>Title</b>	Create JavaScript Functions
<b>Brief Descr.</b>	Create form checking JavaScript functions
<b>Inputs</b>	Group Size
<b>Detailed Descr.</b>	Write a JavaScript function that will be used to validate the shares entered, i.e. they should sum to 100 (or 99) for a group of 3. ( <b>Note to students:</b> This is a fairly elaborate piece of code, but an example of the output can be seen by viewing the source of the weekly form page and looking at the <i>validateForm</i> function).
<b>Output</b>	A JavaScript function to be embedded in the weekly form page.

<b>Process #</b>	21
<b>Title</b>	Build form preamble
<b>Brief Descr.</b>	Create the opening tags for the weekly submission form
<b>Inputs</b>	None
<b>Detailed Descr.</b>	Create the opening tags, including a call to the <i>validateForm</i> JavaScript function created by Process #20
<b>Output</b>	Opening HTML tags to create form

<b>Process #</b>	22
<b>Title</b>	Build Form
<b>Brief Descr.</b>	Show the parts of the form the user will interact with
<b>Inputs</b>	Group ID, week number, Student ID
<b>Detailed Descr.</b>	If a form already exists, use it to populate all the fields, otherwise show blank entries for the shares and the work done
<b>Output</b>	A (possibly populated) HTML form

<b>Process #</b>	23
<b>Title</b>	Build form postamble
<b>Brief Descr.</b>	Create closing tags for the weekly submission form
<b>Inputs</b>	Name
<b>Detailed Descr.</b>	Show the closing HTML tags and hide a field in it to identify to Process #10 that a weekly form has been submitted
<b>Output</b>	The closing HTML tags

<b>Process #</b>	24
<b>Title</b>	Process weekly report
<b>Brief Descr.</b>	Enter the form and weekly shares into the database
<b>Inputs</b>	Week, Student ID, the weekly form
<b>Detailed Descr.</b>	Clean the form by removing escape strings and enter the text into the weeklyReport table. Enter the student share for each student for that week
<b>Output</b>	A confirmation message that the form has been processed or an error message.

<b>Process #</b>	25
<b>Title</b>	Show all weekly reports
<b>Brief Descr.</b>	Show all the forms and breakdowns so far
<b>Inputs</b>	Week number, Group number
<b>Detailed Descr.</b>	Create a table showing the share for each student for each week so far, and create a link to the form submitted
<b>Output</b>	A table of results with embedded links to older forms

<b>Process #</b>	26
<b>Title</b>	Show report admin
<b>Brief Descr.</b>	Show all reports for a group
<b>Inputs</b>	Week number, group number
<b>Detailed Descr.</b>	This is for faculty only. Show the current report as well as a list of existing reports. Include a menu to choose a different group
<b>Output</b>	The current weekly report as well as a list of all reports submitted by this group.

<b>Process #</b>	27
<b>Title</b>	Existing form
<b>Brief Descr.</b>	Does a form exist for this group for this week?
<b>Inputs</b>	Week number, group number
<b>Detailed Descr.</b>	Check if a form has already been submitted for this group
<b>Output</b>	An existing form, if it exists.