



D3: REQUIREMENTS DOCUMENT

Deliverable ID	D3
Deliverable Title	Requirements Document
Project	PSD3 Group Exercise 1
Team	L
Authors	Michael Kilian Dan Tomosoiu Tony Lau Peeranat Fupongsiripan Hector Grebbel
Deliverable Date	12 November 2012
File Name	RequirementsSpecD3`v1.tex
Version	3.0

Contents

1	Introduction	2
1.1	Related Documentation	2
1.2	Purpose and Description of Document	3
1.3	Document Status and Schedule	3
2	System Scope	4
2.1	System Actors	4
2.2	Domain Model	5
3	Use Case Descriptions	7
3.1	Utilities / Account Management	8
3.2	Coordinator Administration	11
3.3	Advert Submission	14
3.4	Advert Viewing and Applications	17
4	Non Functional Requirements	21
5	Conflicts and Resolution	21
A	Change Log	21
A.1	Version 1	21
A.2	Version 2	21
A.3	Version 3	21
B	Glossary	21

1 Introduction

Software Engineering (SE) and Electronic and Software Engineering (ESE) students in the School of Computing Science are required to complete an internship as part of their course, in the summer between level 3 and level 4. An internship is a short period of time that a student spends working within a company in order to gain experience (from as little as a month up to a year). Internships in the software industry are normally paid, although the rate offered can vary from company to company. The School imposes requirements on these internships to ensure that the student receives an appropriate experience for their degree programme. More details of these restrictions can be found on the Software Engineering Summer Placement (SESP) moodle page.

Currently, available internships are advertised to students on an ad-hoc basis through the SESP moodle page. An organisation wishing to recruit an intern submits an advertisement to the course coordinator, who publishes it on the course mailing list. The format and content of the advert can vary widely, including information about the nature of the internship (what the successful applicant will do), duration, expected start date, compensation, person requirements and so on. The course coordinator checks each advert and comments on whether it is suitable for SE/ESE students, as students who are not enrolled on the SE/ESE scheme may also view the advertisements posted on the SESP moodle page in order to obtain information about possible internships.

Sometimes internships applications are managed through the Careers Service's Club21 website; sometimes through the e-Placements scheme; and sometimes the company has its own system of collecting applications. In addition, some advertisements are posted by academics in the school for students to work with them during the summer vacation.

The allocation of SE/ESE students to internships is tracked by the course coordinator separately, using a Microsoft Access Database. Students are required to inform the coordinator when they have secured a placement, which may or may not have been advertised on the SESP page. The coordinator must then approve the internship if it is suitable for the student's course.

The SESP course coordinator has decided that a unified system is necessary for collecting and publishing internship advertisements, and for tracking which SE/ESE students have been successful in securing them. An initial requirements analysis has found that the following features must be supported by the system:

- Submission of internship advertisements
- Review, comment and publication of internship advertisements by the course coordinator
- Review of advertisements by students
- Notification of successful selection for an internship by an SE/ESE student

1.1 Related Documentation

These documents are available in our team repository:

- Client interview questions
- Raw requirements list

- Interview audio recording
- Client consent form

1.2 Purpose and Description of Document

This document displays the use cases we have identified for the system. The long term aim of the document is to add to it as new requirements are identified and to use it as a basis for system design.

1.3 Document Status and Schedule

Naturally this document is under constant revision as new requirements are identified and as stakeholder conflicts are resolved

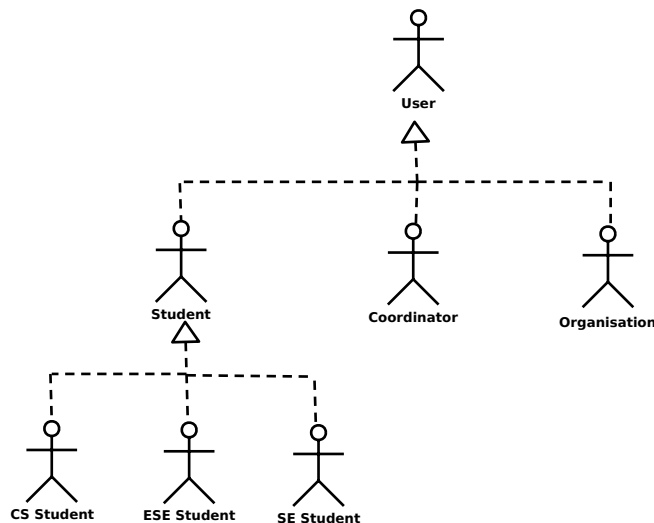
This document should be revised weekly in time for the next PSD team meeting (Tuesday 11am), assuming any changes are required. A change log for this document is visible in the Appendix. Further information about changes can be viewed using github's version tracking features.

2 System Scope

As specified in the initial business case, the system is only to be used within the Department of Computing Science. The only outside interaction comes from companies who submit advertisements to the system.

The process of a student applying for a placement is the most complex with regards to scope. In many cases larger companies will have their own websites or online application processes which they will wish the student to take part in. In this case the student does not apply through the system and will apply by using the details found in the corresponding advert for the placement. However smaller, independent companies may not have their own process. For such companies the system will handle direct applications by the student to the company. Finally the student may also find placements entirely on their own. When this occurs the student should submit information about the placement through the system, but the application to the placement is outside the system scope.

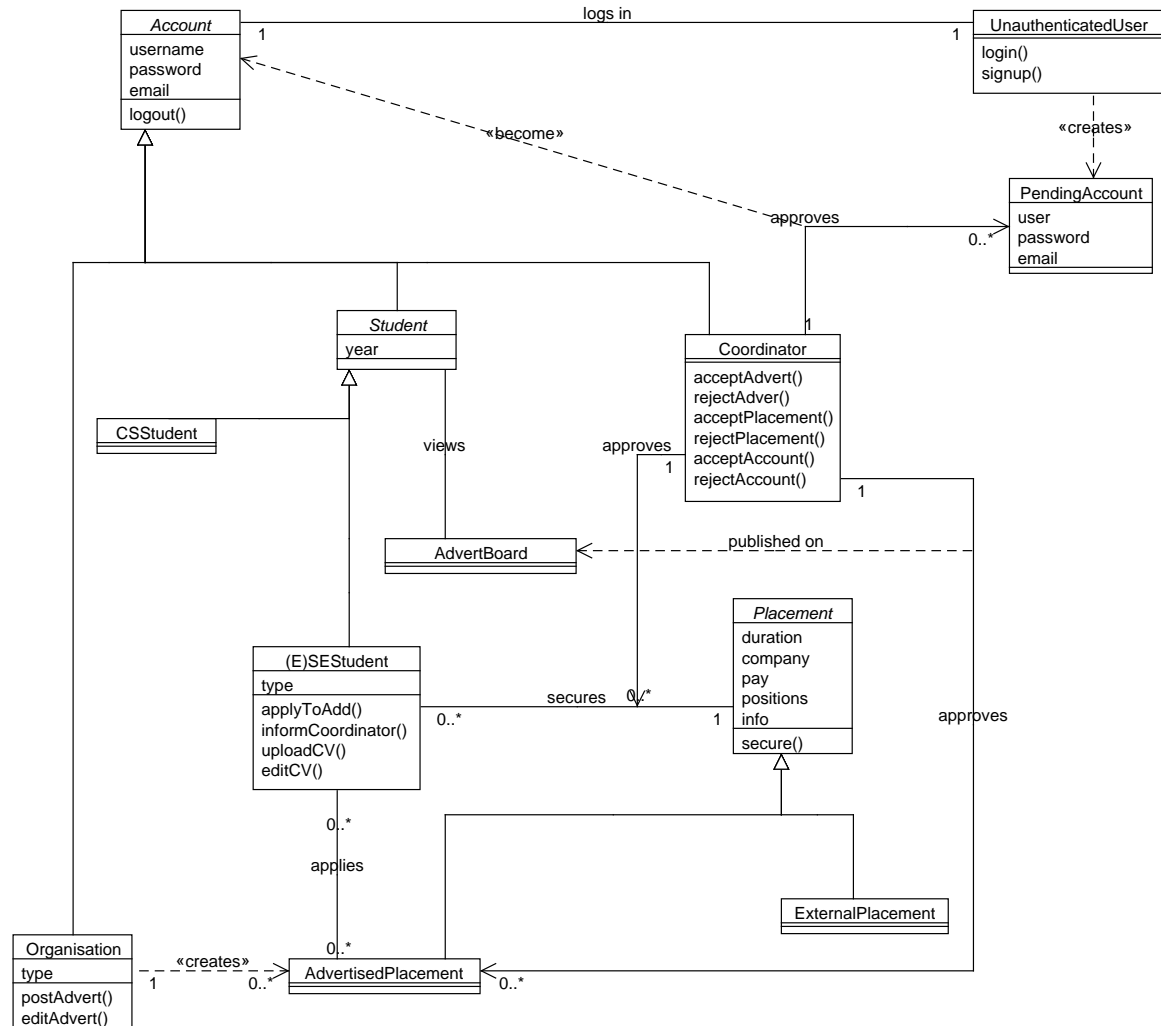
2.1 System Actors



The above diagram shows the actor roles present in the system:

- **Student:** all students enrolled in the system who can apply for placements. This can be further broken down into CS, SE and ESE students, who will have different rules on what constitutes a suitable placement.
- **Coordinator:** the course coordinator will have privileges to approve and reject files, view student progress, etc.
- **Organisation:** any group who advertises placements to the system. This may include external companies, internship schemes and university groups.

2.2 Domain Model



Unauthenticated users represents the guests that visit the system, prior to logging in.

Unauthenticated users can opt to sign up, which creates a **PendingAccount** instance, that needs to be approved by a **Coordinator**. Once the **Coordinator** approves it, the **PendingAccount** becomes one of the three types of **Account**, and can then be used for logging in.

Account is a generalisation representing user accounts. This branches off to three types of account - **Organisation**, **Coordinator** and **Student**.

Organisations represent either Companies or Academics (distinguished by type). They create **AdvertisedPlacements** that, once approved by a **Coordinator**, become visible on the **AdvertBoard**.

Students are either ordinary Computing Science students, in which case they can only view adverts, or (Electronic) Software Engineering students, which can also apply to the placements that are visible on the **Advert Board**.

Students can secure placements, either from the ones visible on the AdvertBoard, or from external sources. Coordinators need to approve the securing of placements.

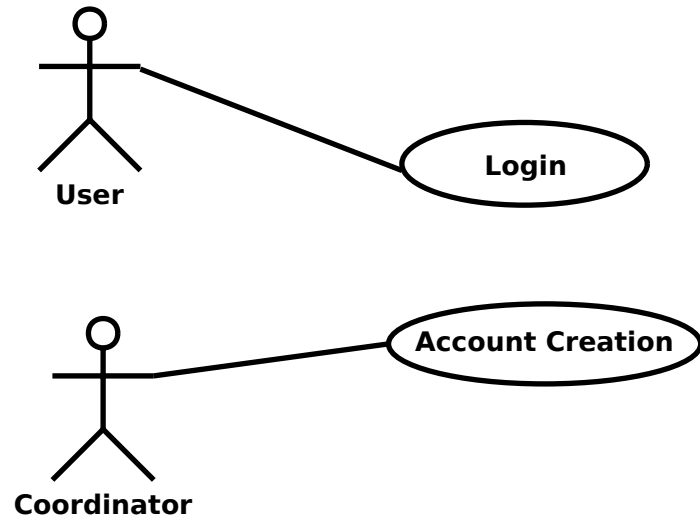
3 Use Case Descriptions

This section describes the key features which are required for the system. These can be broken into four categories as follows:

- **Utilities / Account Management**
 - Login
 - Account creation
- **Coordinator Administration**
 - Advert approval
 - Mark placements as filled
- **Advert Submission**
 - Submission of adverts by organisations
 - Submission of external placements by students
- **Advert viewing and applications**
 - Viewing of adverts
 - Application for placement through system
 - Notification of successfully securing a placement

3.1 Utilities / Account Management

Use Case Diagram

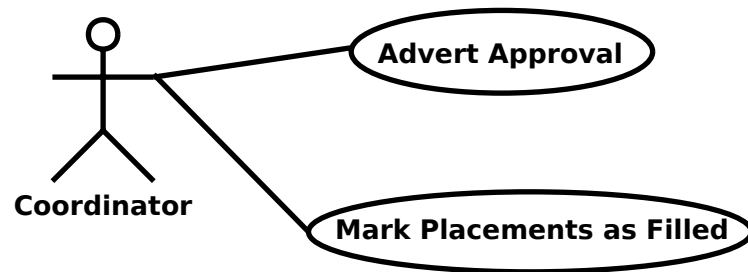


Use case	Login
Description	<ol style="list-style-type: none"> 1. User enters account name and password into appropriate field. 2. User presses enter/confirm. 3. IF (details are correct) THEN GOTO 4 ELSE GOTO 5. 4. Prompt user reenter details. GOTO 1. 5. END
Rationale	Identified a need for different classes of user to have a different range of functionality presented. For example only the coordinator should be able to approve adverts. Identified at client interview.
Priority	Must Have
Status	Implemented
Actors	User
Extensions	
Includes	
Conditions	<ul style="list-style-type: none"> • pre user must have an account created in the system
Non-Functional Requirements	<ul style="list-style-type: none"> • Login should be GUID for students and generated for companies.
Scenarios	<ul style="list-style-type: none"> • Primary: Mark enters his username and password into the corresponding fields and selects login. He is logged into the system successfully. • Alternate 1: Ross makes a typing mistake whilst entering his password and attempts to login. The system reports this error and prompts him to reenter his password. • Alternate 2: Craig wishes to login to the system but has forgotten his details. He chooses to have his password sent to him. He enters his email to which the account is registered and selects send. Using the details emailed to him he successfully logs in.
Risks	
User Interface	Username/Password input fields

Use case	Account Creation
Description	<ol style="list-style-type: none"> 1. User chooses an account type (Organisation or Student) . 2. User is prompted for an account username. IF (the account name is taken) THEN prompt the user to enter another name and repeat. 3. Prompt user to enter password and then enter it again to confirm. IF (passwords don't match) THEN alert the user and repeat. 4. Continue to prompt user for relevant data in this fashion. 5. END
Rationale	To interact on an individual basis with the system, it must have some way of telling who is using it. Identified in business cases / client interview.
Priority	Must Have
Status	Not Implemented
Actors	Coordinator
Extensions	
Includes	
Conditions	<ul style="list-style-type: none"> • pre A user exists for whom an account must be created.
Non-Functional Requirements	<ul style="list-style-type: none"> • For students their account details should match their GUID login. • Usernames must be unique.
Scenarios	<ul style="list-style-type: none"> • Primary: Tim wishes to create a new account. He selects an option to create a new account and enters a username and initial password for the user. He then confirms his selection. The system informs him that his choice of username is valid and the account is created. • Alternate 1: Rose attempts to confirm a new account, but the username she has entered is already in use. The system informs her of this and she chooses an alternative username instead. She confirms this new username and the account is successfully created.
Risks	
User Interface	A field based input for each value required

3.2 Coordinator Administration

Use Case Diagram

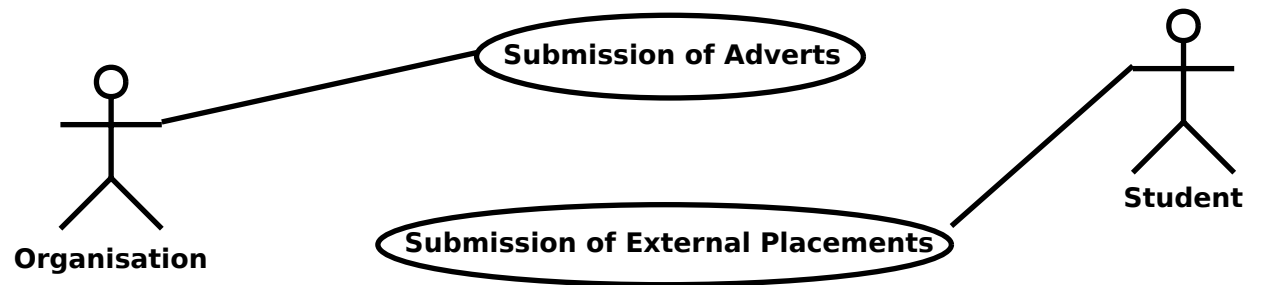


Use case	Advert Approval
Description	<ol style="list-style-type: none"> 1. User selects an advert. 2. User selects 'approve' and confirms. 3. Advert is made visible to students. 4. END
Rationale	Course coordinator needs to be able to assess advert to be visible for course coordinator. Identified in business case / client interview.
Priority	Must Have
Status	Not Implemented
Actors	Coordinator
Extensions	Email approved adverts to students
Includes	<ul style="list-style-type: none"> • Login • Viewing of adverts
Conditions	<ul style="list-style-type: none"> • pre Advert must exist. • post Advert is marked as approved and becomes viewable to students.
Non-Functional Requirements	Prompt for confirm before approving advert.
Scenarios	<ul style="list-style-type: none"> • Primary: John logs in and searches for unapproved adverts on the advert board. He finds one such advert and determines whether the advert is suitable for the course. Then, he marks the advert as approved, confirming his choice when prompted. • Alternate 1: Kate logs in and search for unapproved adverts. Then she accidentally clicks approve advert without determining the purpose of the course carefully. At the confirm prompt, she cancels the approval.
Risks	Advert may be accidentally approved before editing is complete or when advert is unsuitable. Even with error checking prompts this is still a possibility.
User Interface	

Use case	Mark Placements as Filled
Description	<ol style="list-style-type: none"> 1. User selects an advert 2. User chooses to mark the placement as filled and confirms 3. Advert now appears as taken in advert board 4. END
Rationale	There is obviously a need to communicate that a placement has been successfully filled so that other students don't waste their time making applications to it. However the client requested that in this situation the advert should not be removed, in case the placement becomes available again. Identified at client interview.
Priority	Should Have
Status	Not Implemented
Actors	Coordinator
Extensions	
Includes	Viewing of Adverts
Conditions	<ul style="list-style-type: none"> • pre an advert exists for a placement which a student has filled. • post this advert is marked as taken on the advert board.
Non-Functional Requirements	Persistence of advert after marking it as filled
Scenarios	<ul style="list-style-type: none"> • Primary: Valerie logs in and has a placement she wishes to mark as fulfilled by a student. She scrolls through the advert board until she finds the advert, selects in and chooses an option to mark it as taken. • Alternate 1: Margaret logs in and searches the advert board for a placement she wishes to mark as taken. She mistakenly selects the wrong placement and proceed to mark it as taken. Upon realising her mistake, she reselects the advert and chooses to reopen the advert for submissions.
Risks	User could mistakenly mark an advert as taken and leave the system, making the advert appear closed to students (and therefore most likely leave a taken advert marked as unfulfilled), failing to notice their mistake.
User Interface	

3.3 Advert Submission

Use Case Diagram

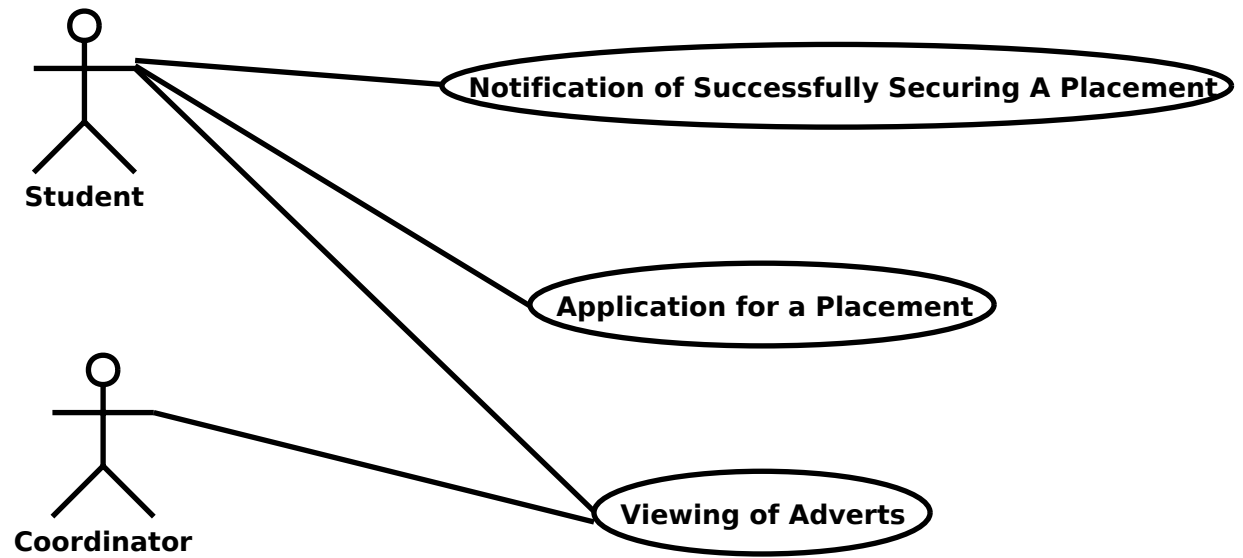


Use case	Submission of Adverts
Description	<ol style="list-style-type: none"> 1. User chooses 'submit advert'. 2. Prompt for input into advert fields. 3. User confirms their input. 4. Advert is added to system, pending approval. 5. END
Rationale	This is a core feature of the system. Identified in the business case
Priority	Must Have
Status	Not Implemented
Actors	Organisation
Extensions	
Includes	Login
Conditions	<ul style="list-style-type: none"> • pre The organisation must have already obtained a login for the system. The organisation must be logged in. • post The advert is queued to be reviewed by the course coordinator before being visible to the users of the system.
Non-Functional Requirements	<ul style="list-style-type: none"> • Normally for the established organisations, a link to the application page of the organisation should be included in the description. Also organisations can post links to more info in the ad.
Scenarios	<ul style="list-style-type: none"> • Primary: Apple Inc. logs in successfully. They select the Submit Advertisement menu item and are taken to the correct page. They fill in all fields and click the Confirm button at the bottom of the page. A confirmation is displayed saying that the advert submission was successful. • Alternate 1: Dr. Jones logs in successfully. He selects the Submit Advertisement menu item and is taken to the correct page. He enters details in the fields and clicks Confirm. An error message is displayed saying that not all fields have been filled out. Dr. Jones fill in the missing fields and clicks Confirm again. A confirmation is displayed saying the advert submission was successful.
Risks	
User Interface	Structure Input Form

Use case	Submission of External Placement
Description	<ol style="list-style-type: none"> 1. User chooses 'submit advert'. 2. Prompt for input into advert fields 3. User confirms their input. 4. Advert is added to system, pending approval. 5. END
Rationale	A student must get an internship approved by submitting the internship details. Identified at client interview.
Priority	Should Have
Status	Not Implemented
Actors	Student
Extensions	
Includes	Login
Conditions	<ul style="list-style-type: none"> • pre The student must have already obtained a login for the system (by automatic assignment or from the course coordinator). The student must be logged in. • post The advert is queued to be reviewed by the course coordinator.
Non-Functional Requirements	Structured input form
Scenarios	<ul style="list-style-type: none"> • Primary: Barry wants to submit an advert to be reviewed. He logs in to the system, clicks the submit advert button under the review section and fills in the details. It gives him a message confirming the success of the operation.
Risks	
User Interface	

3.4 Advert Viewing and Applications

Use Case Diagram



Use case	Viewing of Adverts
Description	<ol style="list-style-type: none"> 1. Adverts are displayed on screen. 2. END
Rationale	Fundamental to system description. Initial need specified in business case and exact method for viewing adverts elaborated in client interviews
Priority	Must Have
Status	Not Implemented
Actors	<ul style="list-style-type: none"> • Organisation • Student • Coordinator
Extensions	
Includes	Login
Conditions	None
Non-Functional Requirements	
Scenarios	<ul style="list-style-type: none"> • Primary: John logs in and attempts to find a suitable placement on the advert board. He scrolls down through the advert board until he finds a suitable placement, or there are no adverts left to view. He closes the interface.
Risks	
User Interface	

Use case	Application for Placement
Description	<ol style="list-style-type: none"> 1. User finds advert they wish to apply for on the advert board 2. They select this advert and choose 'Apply' 3. Prompt user for relevant info. 4. User confirms and sends application. 5. END
Rationale	Some smaller companies may not have their own application process so in this scenario, the client has expressed a preference for applications to be handled through the system. Current version identified as a result of requirements forum.
Priority	Could Have
Status	Not Implemented
Actors	Student
Extensions	
Includes	Viewing of Adverts
Conditions	<ul style="list-style-type: none"> • post application has been submitted and made visible to company
Non-Functional Requirements	Structured input form
Scenarios	<ul style="list-style-type: none"> • Primary: Karl wishes to submit an application for an advert a friend told him about. He finds the advert on the board and chooses to submit an application. He enters all relevant extra data and sends off the application. He exits the system. • Alternate 1: Ian selects an advert and chooses to submit an application. When prompted to enter any extra details he accidentally clicks send. He is prompted to confirm he wishes to submit the application. Thanks to this warning he recovers the error and enters data. He submits the application
Risks	Student could send away incomplete or erroneous applications accidentally. No method exists to retrieve/edit such an application
User Interface	

Use case	Notification of Successfully Securing a Placement
Description	<ol style="list-style-type: none"> 1. Student selects a placement from the advert board 2. They choose to notify that they have secured this placement 3. IF (the placement is already filled) THEN user is notified and submission is ignored. GOTO 2 4. Notification is received and put into system for confirmation by coordinator 5. END
Rationale	Course coordinator must ensure that every course student has their work placement.
Priority	Could Have
Status	Not Implemented
Actors	Student
Extensions	
Includes	Login
Conditions	<ul style="list-style-type: none"> • pre Advert must exist. • pre Advert must be approved by course coordinator. • pre Organisation has to secure the placement position for student. • post Course student's status is marked as secure
Non-Functional Requirements	
Scenarios	<ul style="list-style-type: none"> • Primary: Andrew logs in and look for desirable and approved advert. He clicks on secure the placement button and sends an email about the selected placement to course coordinator. • Alternate 1: Margaret logs in and select desirable advert but she cannot click on secure the placement button because the organisation she applies for has not accepted her position.
Risks	It may take time for the organisation to secure student's position.
User Interface	

4 Non Functional Requirements

- When the deadline for a placement has been reached, the placement should remain visible for one week after the deadline before being removed from the advert board
- There should be a single interface, but different dashboards available to the user type once the user logs in.
- Getting login details to the company could be done by email – an email could be sent to the company with registration details and the course coordinator would approve the request for the company to be part of the system.

5 Conflicts and Resolution

Conflict: Originally in client interview it was established that students could be able to make applications through the system directly to companies. However at the stakeholder panel it was suggested that this feature was outside of the system's scope. **Resolution - 2012/11/08:** The question of whether this feature should or should not be implemented was raised by the team in the Requirements Forum. compromise the team chose to keep this feature as a potential use case but relegated it from a must have to a could have status.

A Change Log

A.1 Version 1

0. Initial Draft – 2012/11/01
1. Minor edits / cleanup – 2012/11/09

A.2 Version 2

0. Revision of use cases. Addition of use cases diagrams – 2012/11/11

A.3 Version 3

0. Addition of pseudocode descriptions and Conflict and Resolution section – 2012/11/12

B Glossary

PSD/PSD3: Professional Software Development 3

CS: Computing Science

SE: Software Engineering

ESE: Electronic Software Engineering

SESP: Software Engineering Summer Placement

GUID: Glasgow University Identification