Internship Management System Requirements Specification

Revision 3554, made 21/01/2013 by jsinger

January 28, 2013

Contents

1 Problem Description

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 in 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. The full requirements specification for the system are described in this document.

2 Domain Model

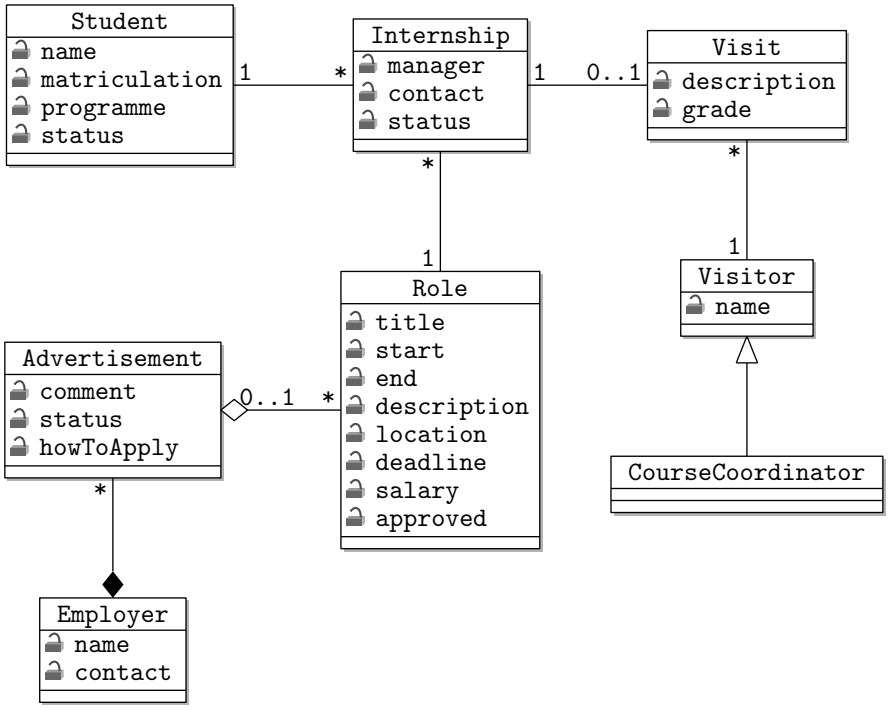


Figure 1: Internship management system domain model

Figure 1 illustrates the major domain elements for the internship management system. The system is centred around the notion of a `Role` describing an internship position with an `Employer`. A number of `Roles` may be placed in an `Advertisement` by an `Employer`. An advertisement describes the common information for all `Roles` with a particular `Employer`.

There may be several positions available for each described role, so more than one student can inform the `CourseCoordinator` that they have secured an `Internship` associated with an advertised `Role`. Alternatively, a `Role` needs to be created if the student records an `Internship` that is not associated with a `Role` advertised on the system.

Each `Internship` is associated with a `Visit` by an academic `Visitor`. This class records details of the visit and the grade assigned by the `Visitor`.

3 Actors

Figure 2 illustrates the relationships between actor roles in the system. A short summary of the actors is given below:

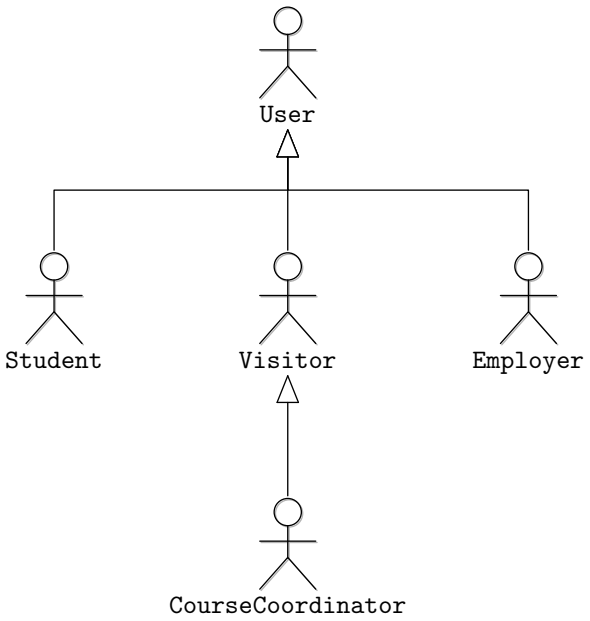


Figure 2: System actors

User a general user of the system, able to login and logout.

Student uses the system to search for internships, and (depending on programme) notify the course coordinator when they secure a placement.

Employer uses the system to submit advertisements describing available internship roles in their organisation.

CourseCoordinator uses the system to circulate internship advertisements to students and monitor the progress of SE/ESE students in finding a placement. The course coordinator also uses the system to arrange academic visits to students during their internship.

Visitor uses the system to record their assessment of a student's performance during an internship.

4 Use Cases

This section describes the required functionality for the system as a number of use cases (organised into categories):

- Utility use cases
  - Login (MUST HAVE)
  - Logout (SHOULD HAVE)
  - View student status summary (SHOULD HAVE)
  - View advertisement summary (MUST HAVE)
- Advertisement management
  - View advertisement detail (MUST HAVE)
  - Register employer (MUST HAVE)
  - Submit advertisement (MUST HAVE)
  - Revise advertisement (SHOULD HAVE)
  - Publish advertisement (MUST HAVE)
- Offer management
  - Notify accepted offer (MUST HAVE)
  - Approve accepted offer (MUST HAVE)
- Visit management
  - View student detail (MUST HAVE)
  - Record visit assessment (SHOULD HAVE)

Some further use cases have also been identified as potentially useful. Some of these use cases are concerned with the optional extension identified during requirements gathering for allowing employers to manage applications for internships via the system. The clients do not envisage that these use cases should be implemented in the current construction phase and have not been fully described in the previous elaboration phase.

- Schedule academic visit
- Edit accepted internship
- Submit application
- Review applications
- Make offer to student
- Accept offer

4.1 Utility use cases

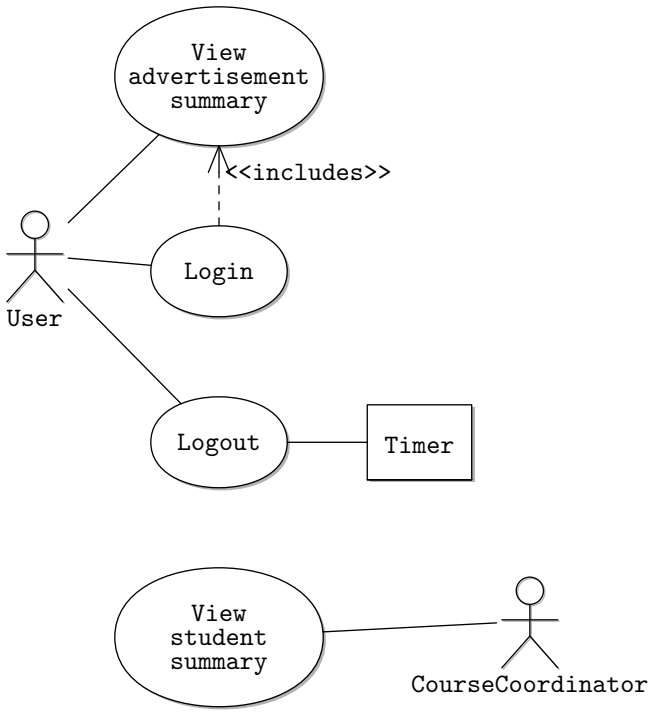


Figure 3: Utility use cases

Figure 3 illustrates the utility use cases for the system. `Users` can log in to the system, which causes the list of advertisements to be displayed. `Users` can also log out manually, or can be logged out by a `Timer`. The `CourseCoordinator` can also view a summary list of `Students` enrolled on the SESP course.

Use case	Login
Description	A user is presented with a prompt into which they must enter their MyCampus username and password. If the combination is valid, the user is logged in and presented with the summary of advertisements view.
Rationale	The system must provide security so that only authorised users are permitted to view privileged information.
Priority	
Status	Not implemented.
Actors	• User
Extensions	
Includes	• View advertisements summary
Conditions	**pre** No user is currently logged in at the same user interface. **post** User is logged in, if username and password combination is correct.
Non-Functional Requirements	Authentication credentials must be checked with the MyCampus single sign-on service.
Scenarios	
Risks	**Realisability** MyCampus authentication system may not be available.
User Interface	

Use case	Logout
Description	This use case is invoked either manually by a User, or by a Timer if the system is idle for a configurable period of time. When the logout option is selected, the current session with the user is closed.
Rationale	The system must provide security to ensure that confidential Student and Employer information is protected.
Priority	
Status	Not implemented.
Actors	• User • Timer
Extensions	
Includes	
Conditions	**pre** User is logged in. **post** User is logged out and the session is ended.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	View advertisement summary
Description	Displays summary information for all published, advertised roles on the system and the associated roles in a tabular format. If the logged in user is an `Employer`, then only their own advertisements should be included in the summary. If the logged on `User` is the `CourseCoordinator`, then all `Advertisements` that have not been published are included in the summary view. The following information should be published in tabular format for each advertised role: • Employer name • Title • Start date • End date • Coordinator approved • Location • Salary
Rationale	Users need to be able to survey the available roles.
Priority	
Status	Not implemented.
Actors	• User
Extensions	
Includes	
Conditions	**pre** User is logged in.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	View student summary
Description	Displays student summary information regarding placements. The following information about all students enrolled on the SESP course in the current session should be displayed: • Name • Matriculation • Status • Student contact email address • Manager name • Manager contact email name • Whether a visit has completed
Rationale	The `CourseCoordinator` needs to be able to track the progress of students in securing an internship to ensure that all students meet their course requirements.
Priority	
Status	Not implemented.
Actors	• `CourseCoordinator`
Extensions	
Includes	
Conditions	**pre** `User` is logged in.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

4.2 Advertisement Management

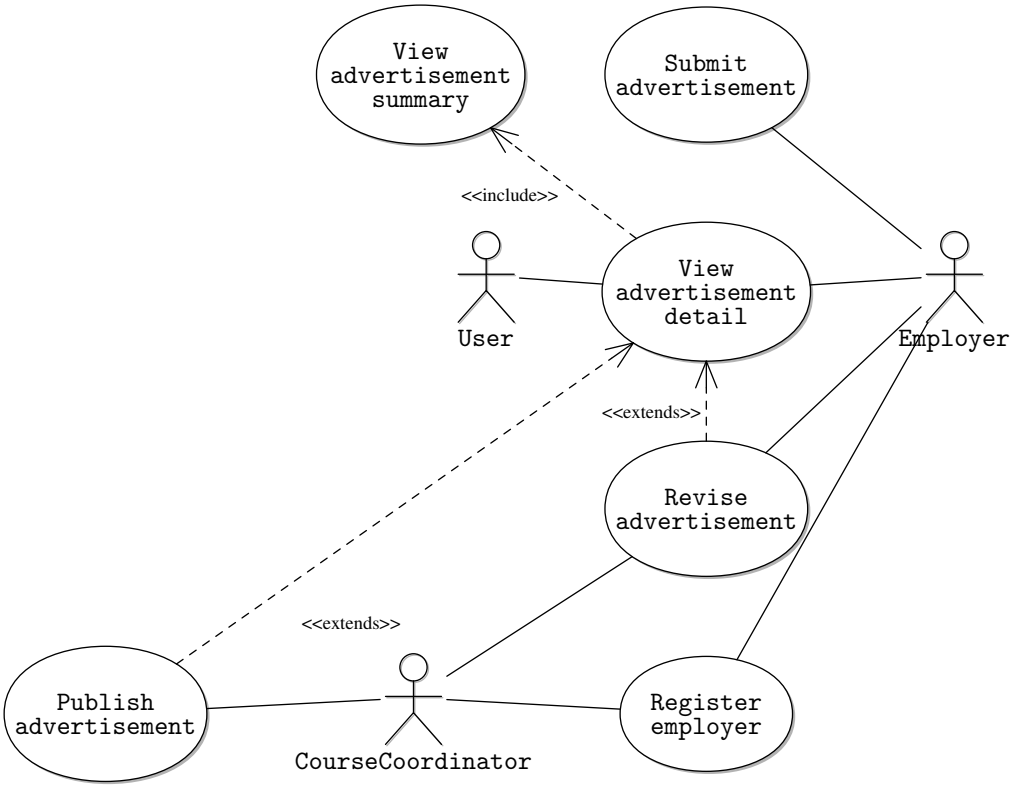
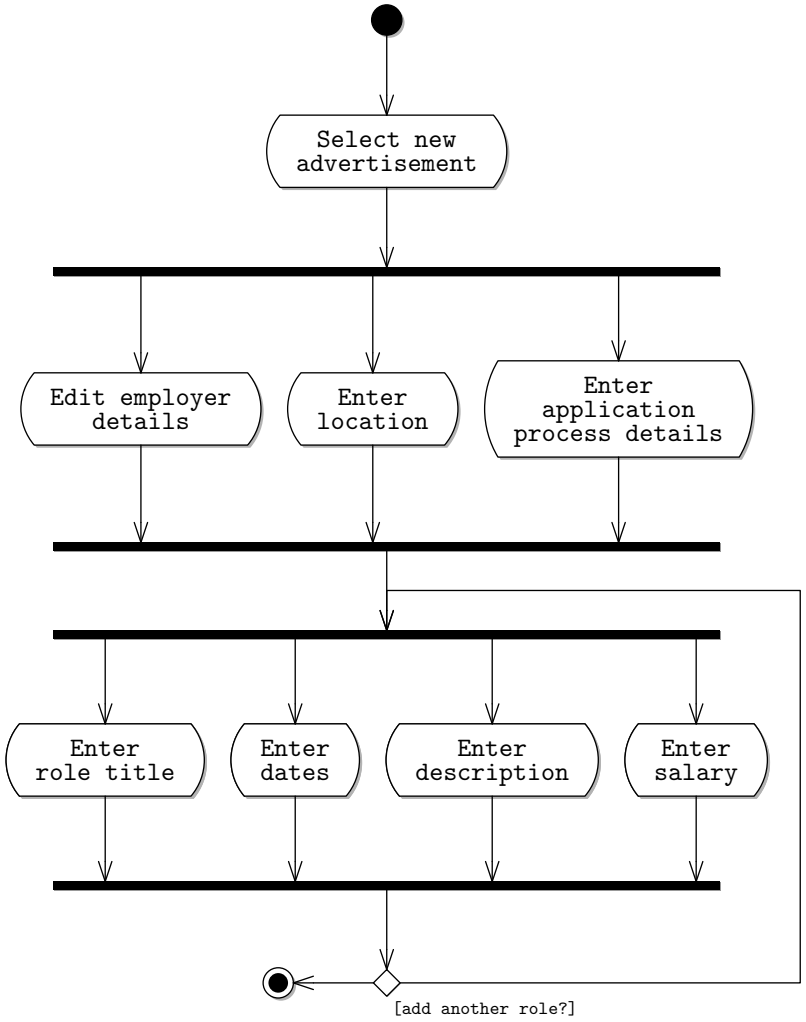


Figure 4: Offer management use cases

Figure 4 illustrates the use cases of the workflow for publishing a `Advertisement`. The `CourseCoordinator` is responsible for registering new `Employers` with the system. `Employers` can submit and revise `Advertisements` as many times as they wish, until the `Advertisement` is marked as approved by the `CourseCoordinator`. `Advertisement` details can be viewed by all `Users`, although `Employers` can only view their own advertisements.

Use case	Register employer
Description	New `Employers` who wish to advertise placements are registered with the system. The user must supply an `Employer` name and contact information (including an email address). The `Employer` is then notified of the new registration and issued with a username and password for logging on to the system.
Rationale	Employers need to be able to access the system to submit adverts.
Priority	
Status	Not implemented.
Actors	• `CourseCoordinator` • `Employer`
Extensions	
Includes	
Conditions	**pre** The user is logged in **post** a new `Employer` registration is added to the system.
Non-Functional Requirements	Usernames issued to employers must not conflict with usernames issued to students via MyCampus.
Scenarios	
Risks	**Necessity** Employer login may not be required if all adverts are submitted by the `CourseCoordinator`.
User Interface	

Use case	Submit advertisement
Description	 The activity diagram shows the flow of activities for the use case. Employer details should be pre-completed by the system, based on the information provided when the `Employer` is registered with the system.
Rationale	`Employers` need to be able to create new `Advertisements` describing available roles on the system. The `CourseCoordinator` may do this on behalf of some employers.
Priority	
Status	Not implemented.
Actors	• `Employer` • `CourseCoordinator`
Extensions	
Includes	
Conditions	**pre** User is logged in **post** A new advertisement describing one or more roles is added to the system with status set to `PENDING`.
Non-Functional Requirements	

Scenarios	
Risks	
User Interface	

Use case	View advertisement detail
Description	Presents the full details of a selected `Advertisement` (from a list presented in the `View advertisement summary` use case), including associated roles to the `User`. All `Users`, except `Employers` can view the details of all `Advertisements`. `Employers` can view the details of their own `Advertisements` only. The full details for the advertisement are: • `Employer` name and contact • The location where the roles will be based • How to apply for the listed roles • `CourseCoordinator` comment The details for each `Role` associated with the `Advertisement` are as follows: • Title • Start and end dates • Description • Salary • Whether the `Role` has been approved as appropriate for SESP
Rationale	`Users` must be able to access the details of a `Advertisement`.
Priority	
Status	Not implemented.
Actors	• User
Extensions	• `Revise advertisement`
Includes	• `View Advertisement summary`
Conditions	**pre** User is logged in.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Revise advertisement
Description	Any of the details of an `Advertisement` that is already in the system can be modified by the `Advertisement`'s `Employer`, except the association between the `Advertisement` and the `Employer`. The use case is an extension of `View advertisement detail`.
Rationale	`Employers` may take several attempts to get the details of an `Advertisement` correct.
Priority	
Status	Not implemented.
Actors	• `Employer` • `CourseCoordinator`
Extensions	
Conditions	**pre** The current session `User` is the `CourseCoordinator` or the `Employer` who created the `Advertisement` **pre** The advertisement status is not `PUBLISHED` **post** The modified `Advertisement` is saved in the system.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Publish advertisement
Description	The `CourseCoordinator` selects an `Advertisement` from the summary list to view the `Advertisement`'s detail. They choose the option to publish the `Advertisement` and indicate for each role in the `Advertisement` whether it is suitable for SESP. They also enter an optional comment for the `Advertisement` as a whole. Finally, they confirm that the `Advertisement` should be published. This changes the `Advertisement`'s status from `PENDING` to `PUBLISHED`.
Rationale	The `CourseCoordinator` indicated that `Advertisements` should not be published to `Students` until they are approved.
Priority	
Status	Not implemented.
Actors	• `CourseCoordinator`
Extensions	
Includes	
Conditions	**pre** The user is logged in **post** The `Advertisement` viewed is `PENDING` **post** The `Advertisement` status is changed from `PENDING` to `PUBLISHED`
Non-Functional Requirements	
Scenarios	
Risks	**Necessity** The `CourseCoordinator` may prefer some `Advertisements` to be automatically published to save on time.
User Interface	

4.3 Offer Management

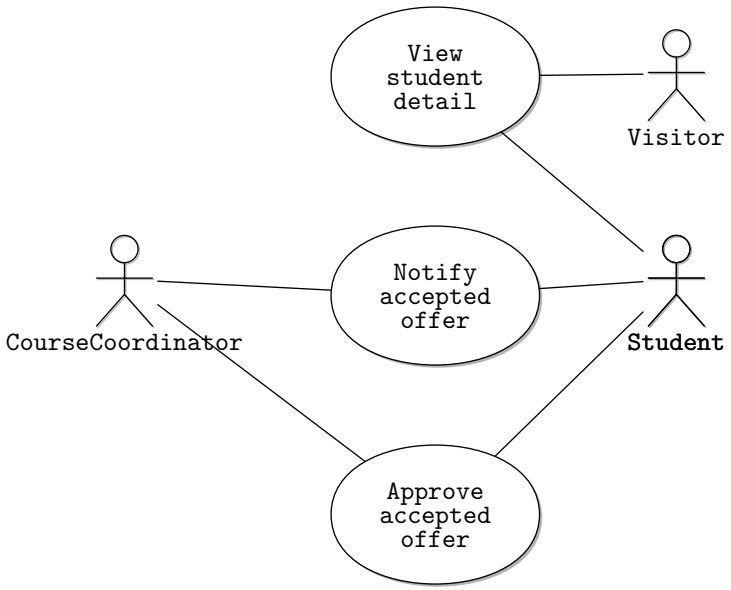
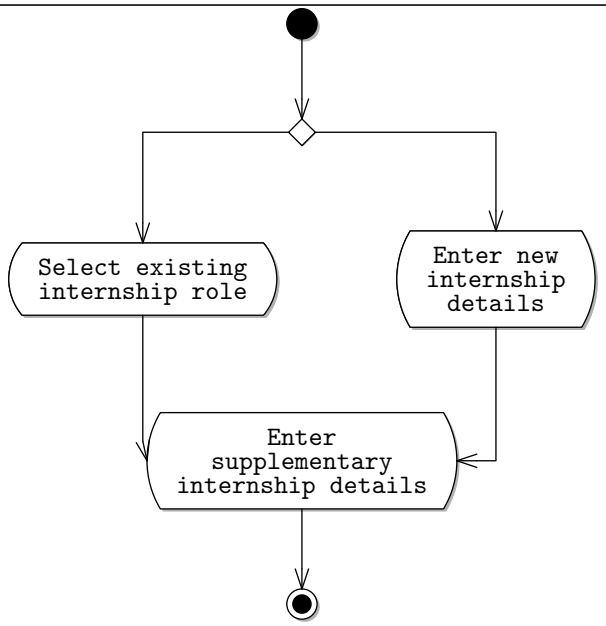


Figure 5: Offer management use cases

Figure 5 illustrates the two use cases in the manage offer work flow. Students indicate that they have accepted an offer for an internship from an employer. If the internship was advertised on the system and approved for SESP by the `CourseCoordinator`, then the `Internship` is automatically approved. If the `Internship` is not for a pre-approved `Role`, then the `Student` must create a new `Role` and this must be approved by the `CourseCoordinator`.

Use case	View student detail
Description	Presents a detailed view of a `Students` status. `Students` can access their own view. The user enters a `student`'s matriculation number to view their details. `Visitors` can access the details of any student they have been assigned to visit. The `CourseCoordinator` can access any `Student`'s details. The information to be presented about a `Student` in this view is: • Name • Matriculation • Programme • Internship status (`APPROVED`, `ACCEPTED`, `WITHDRAWN` or `PENDING`) If a student has accepted an `APPROVED Internship` role, the following details should be displayed: • Internship role title • Internship start and end dates • Internship location • Internship salary • Internship employer name • Internship contact details • Internship manager details The following details should also be displayed if they have been entered into the system: • The name of the `Visitor` • The grade and assessment assigned by the `Visitor`
Rationale	The `CourseCoordinator` needs to track where students will undertake their `Internship` in order to assign an academic `Visitor`. The `Visitors` need to know the context of an `Internship`.
Priority	
Status	Not implemented.
Actors	• `CourseCoordinator` • `Visitor` • `Student`
Extensions	• `Approve accepted offer` • `Assign academic visitor`
Includes	
Conditions	**pre** User is logged in

Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Notify accepted offer
Description	 The activity diagram shows the flow of activities for a `Student` when they use the system to confirm that they have accepted an offer for an `Internship`. If the `Internship` was not advertised on the system, the `Student` must enter these details to create a new `Internship`: • Employer organisation name • Role Title • Salary • Description of activities • Location • Start date • End date The following supplementary details must be entered by a student to confirm that they have accepted an offer: • Manager name • Manager contact email address
Rationale	The `CourseCoordinator` needs to track when SESP enrolled `Students` accept internship offers.
Priority	• Main use case: • Email notifications:
Status	Not implemented.
Actors	• `Student` • `CourseCoordinator`

Extensions	
Includes	
Conditions	**pre** The user is logged in. **post** The `Student` is recorded as having accepted a `PENDING` or `APPROVED` `Internship`. An `Internship` is `APPROVED` if it was a previously advertised role on the system that the `CourseCoordinator` had approved. **post** The `CourseCoordinator` is sent a notification stating that a `Student` has accepted an offer that may need to be confirmed.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Approve accepted offer
Description	A `CourseCoordinator` changes the status of a `Student`'s accepted `Internship` from PENDING to APPROVED, if the `Internship` was not a pre-approved `Role` advertised in the system.
Rationale	SESP `Students` cannot accept offers without the approval of the `CourseCoordinator`
Priority	• Main use case: • Email notifications:
Status	Not implemented.
Actors	• `CourseCoordinator`
Extensions	
Includes	
Conditions	**pre** User is logged in **pre** The selected user has a `PENDING` accepted `Internship` offer. **post** The `Intership` status is changed to `APPROVED`. **post** The `Student` is sent a notification confirming that the offer has been a approved.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

4.4 Visit Management

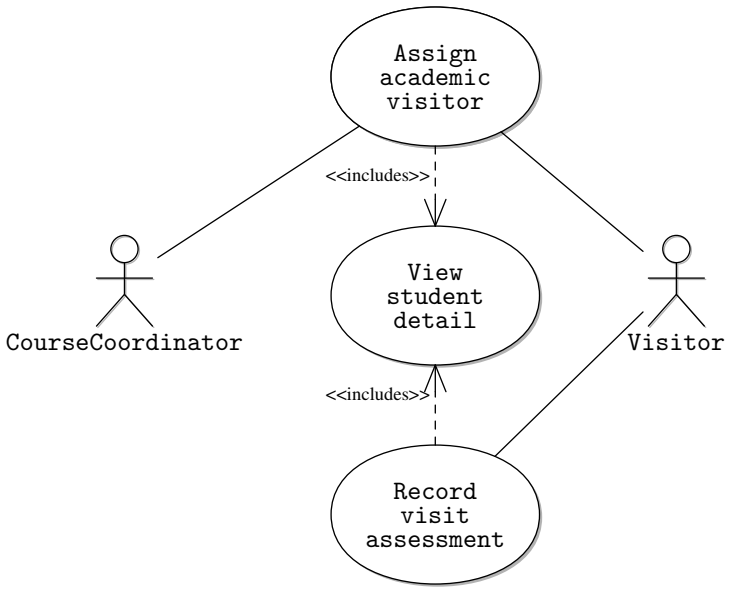


Figure 6: Visit management use cases

Two tasks are undertaken under visit management: assigning a visitor (by the `CourseCoordinator`) and record the `Visitor`'s assessment of the student.

Use case	Assign academic visitor
Description	The `CourseCoordinator` selects views for the record of a `Student` who has accepted an `Internship`. They select an option to set the `Visitor` for the student. They then enter the `Visitor`'s name and email address. The `Visitor` is notified by email of the `Student` they have been assigned to visit. After the use case is complete, the name and email address of the assigned visitor appears on the `Student`'s `Visitor` appears on their detailed record.
Rationale	The `CourseCoordinator` stated they needed a means of managing the assignment of `Visitor`'s to `Students`.
Priority	• Main use case: • Email notifications:
Status	Not implemented.
Actors	• `CourseCoordinator` • `Visitor`
Extensions	
Includes	• View student detail
Conditions	**pre** The viewed `Student` has accepted an Internship offer. **pre** The viewed `Student` does not have a `Visitor` assigned. **post** The association between the `Student`'s `Internship` and the `Visitor` assigned is recorded in the system.
Non-Functional Requirements	
Scenarios	
Risks	**Completeness** It is not clear from current requirements gathering whether a `CourseCoordinator` will need to able to change an assigned `Visitor`. **Realisable** The current proposed specification makes it difficult to track when a `Visitor` has been assigned multiple student `Visits`, since the `CourseCoordinator` may not enter their name consistently.
User Interface	

Use case	Record visit assessment
Description	An academic `Visitor` chooses to view the details for a `Student` that they have been assigned to visit. They choose an option to record a `Visit` assessment. They enter a textual description of the `Student`'s performance, and select a `grade` on the University of Glasgow's grading system (A1-H).
Rationale	The `CourseCoordinator` stated that it is necessary to have a means of collating visit assessments in the system.
Priority	
Status	Not implemented.
Actors	• `Visitor`
Extensions	
Includes	• View student detail
Conditions	**pre** The logged in user is a `Visitor` **pre** The selected `Student` has been assigned to the `Visitor` for a `Visit` **post** A grade and textual description of the `Visit` is recorded in the system.
Non-Functional Requirements	The system shall be able to record assessment reports of up to 2000 words.
Scenarios	
Risks	**Completeness** The specification provides for a less structured report format than is employed in the current system.
User Interface	

5 Non Functional Requirements

1. (security/0) The system shall authenticate users via the MyCampus single sign-on service.
2. (interoperability/0) The system shall provide a command line user interface, using the CLI library provided.
3. (interoperability/1) The system shall realise the Facade interface provided to ensure compatability with other user interface implementations.
4. (performance/0)The system shall support at least 50 advertisements with at least one role per advertisement.
5. (performance/1) The system shall support at least 50 employers.
6. (performance/2) The system shall support at least 50 registered SESP students.
7. (performance/3) The system shall support at least 100 concurrent users.

6 Unresolved Issues

1.