

CSSE3002

The Software Process

Group Assignment 3

Goal:

To develop and test your skill at constructing a software requirements specification document.

Deadlines:

Plan: Thursday, 21 April (week 8) at 5 pm.

SRS document: Friday, 20 May (week 11) at 5 pm.

Scenario:

Three major universities that have campuses in Brisbane realise they have many staff and students who have overlapping interests. There are many seminars on interesting topics at each of the universities, but how does someone on one campus find out about interesting seminars that are happening at one of the other universities? The universities decide that they would like to create a Web-based academic talks listing service that will advertise talks and events to researchers, students, postgraduates, and interested parties from outside academia across Brisbane. The proposal cites <http://www.talks.cam.ac.uk> and <http://www.meetup.com> as examples of talks-listing services, but notes that a site suitable for academic interests across different campuses will have some unique requirements.

Although the system is open to the public to sign up, anyone claiming to have a university affiliation should be verified at least by checking that they have a university email address. Although anyone can create a talk or a series of regular talks in the system, it is expected that most academic users will want to subscribe to the talk lists that will be maintained by various interest groups. Administrators of an interest group should be able to manage which talks and which recurring talks should be advertised as part of their list, remembering that sometimes they will only be interested in one or two talks from a series and sometimes they will be interested in the whole series. As the talks will be across many different campuses, it's important that the location of a talk isn't just, say "78-420", but that it is easy to tell which campus it is on, and to be quickly able to access transport information (generating links that will use Translink's system and Google Maps to find out travel times from the user's location).

The software requirement specification is merely for the software, but it needs to include descriptions of the interfaces for any systems with which it needs to. You may also consider various non-functional requirements for the system.

What you have to do:

1. Prepare a team plan (2 pages max.) for developing the software requirements specification, describing the process you intend to use, the role of the team members, the team schedule and the expected amount of time for each part of the assignment.



The plan is to be submitted in advance. It will not be marked directly (but a plan must have been submitted to get full marks for the *Process Reflection* criterion). Your assignment will compare your actual process and performance with your plan.

2. Develop a Software Requirements Specification document for a multi-campus, multi-field academic talks listing service. This assignment is intended to be open-ended and to provide you with an opportunity to demonstrate your creativity. As a potential user of the website, you may wish to propose creative requirements for the system. To give you a hint for the scope of the system, we expect your system has at least 10 high-level use cases.
3. Your software requirements specification document should contain both a natural language description of the requirements and UML analysis models, including:


Client: Three major universities that have campuses in Brisbane

Project: a Web-based academic talks listing service


Actors

Name	Aliases (Synonyms)	Subsumptions
Anyone	the public	
Use  Non-Academic		Academic, Non-Academic
	interested parties from outside academia	
Academic [User]	anyone with a University Affiliation	Staff, Student
Staff [Member]		Researchers
Student		Postgraduates
Administrator 	Interest Group Administrator	
Geo Service	Google Maps	
Itinerary Service	Translink's System	
Email Service		

Use Cases

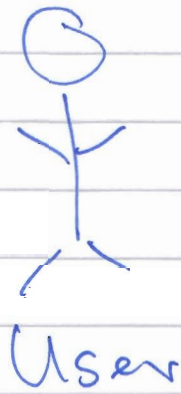
Initiator [Actor]	Use Case	Collaborators [Actor]
Anyone	Read [Talk] Listing	-
Anyone	Show [on-Campus] Location [for Listing]	Geo Service
Anyone	Get Transport Information [for Listing]	Transport Information Service 
Anyone	Sign-up	[University] Email Service
Anyone → User	Create Talk	-
Anyone → User	Create [talk] Series	-
Academic → User	Subscribe to List	-
Administrator	Advertise [on list]	-

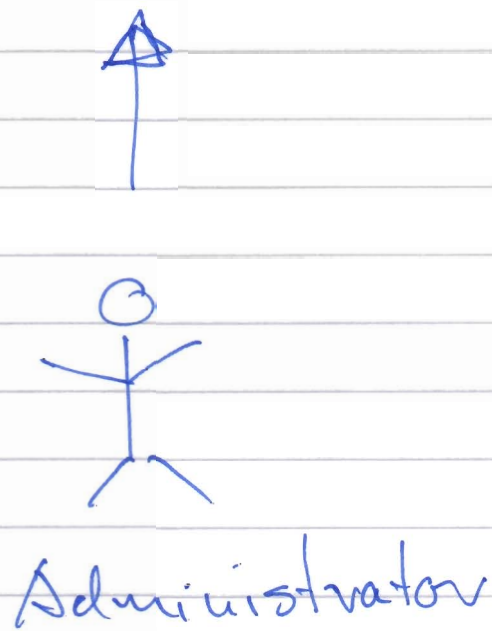
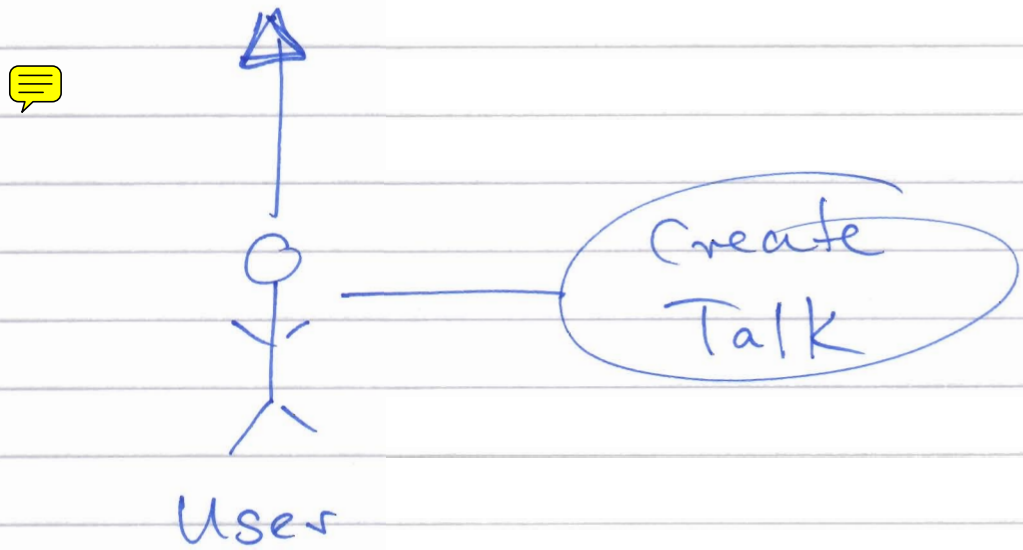
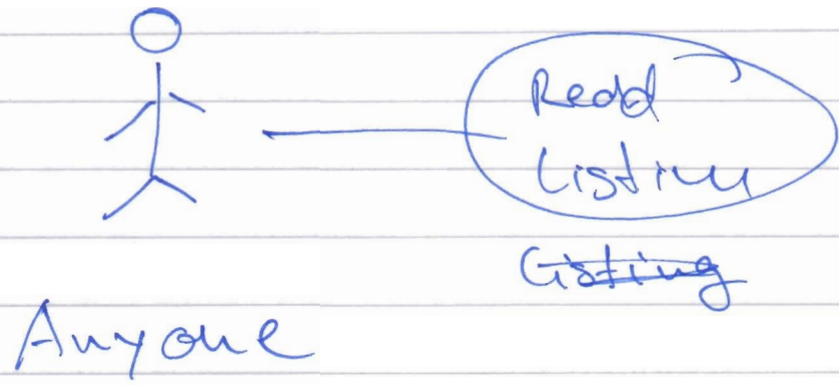
Obligations on Use Cases

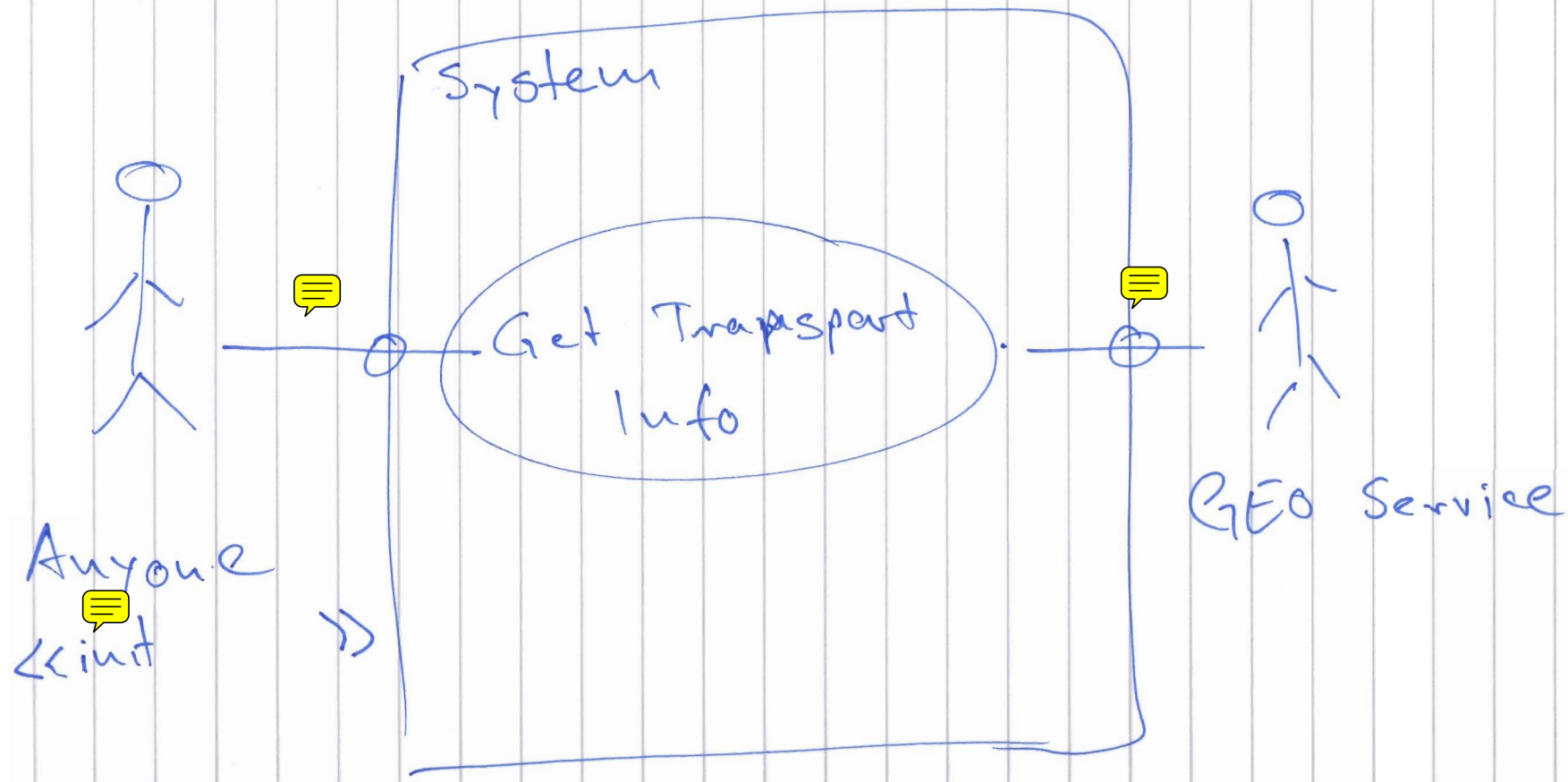
Use Case 	Obligation <i>Anyone</i>
Sign-up	For Academic check that uni email exists
Advertise	Allow single talk or series

Additional Inclusions









- Interfaces Descriptions of for all connected systems
- NFPs







Objects

Term	Aliases (Synonyms),	Persistent	Attributes	Associations	Generalisation	Methods
System 	academic talks listing service	No 	-	-	-	readListing() showLocation(Listing) signUp(email) createTalk(topic*) createSeries(Talk*) subscribe(List) advertise(Listing)
Anyone		No		System	-	-
User 		Yes	Email	System, Listing	Anyone	-
Administrator		Yes		System, List	User	-
Geo Service		No	-	System	-	getMap(Location): Map
Itinerary Service 		No	-	System	-	getItinerary(Location): Itinerary 
Email Service		No	-	System	-	confirmEmail(Anyone): boolean sendEmail(email)
Listing	Talk	Yes		Anyone		
Itinerary 	-	No		Listing		
Location	-	No		Listing 		
Itinerary	-	No		Listing		
Listing	Talk	Yes				
Talk					Listing	
Series				Talk	Listing	
Location	-	No		Listing		
List				 Administrator		advertise (Listing)

