University of Glasgow | Department of Computing Science

D6: SYSTEM DESIGN & ACCEPTANCE TEST PLAN

Deliverable ID	D6
Deliverable Title	System Design & Acceptance Test Plan
Project	PSD3 Group Exercise 2
Team	W
Authors	Gordon Reid: 1002536R Ryan Wells: 1002253W Kristopher Stewart: 1007175S David Selkirk: 1003646S James Gallagher: 0800899G
Deliverable Date	February 27, 2013
File Name	d6.tex
Version	Final

Contents

1 Introduction

1.1 Identification

This is the design document and test plan for the internship management system being developed by Team W for the Professional Software Engineering 3 course. The internship management system is for Software Engineering (SE) and Electronic and Software Engineering (ESE) students, studying in the school of Computing Science.

1.2 Related Documentation

PSD3 Group Exercise Description:
http://fims.moodle.gla.ac.uk/file.php/128/coursework/psd3-ge-2.pdf
PSD3 Requirements Specification:
http://fims.moodle.gla.ac.uk/file.php/128/coursework/requirements-specificati
pdf
PSD3 Forum Information:
http://http://fims.moodle.gla.ac.uk/mod/forum/view.php?id=20433

1.3 Purpose and Description Of Document

This document serves as the design document containing the overall system architecture diagram with associated state charts. Each component in the system architecture also has its own class diagrams and API specifications specified.
The latter section of the document contains the acceptance test plan for the implementation of the Internship Management System.

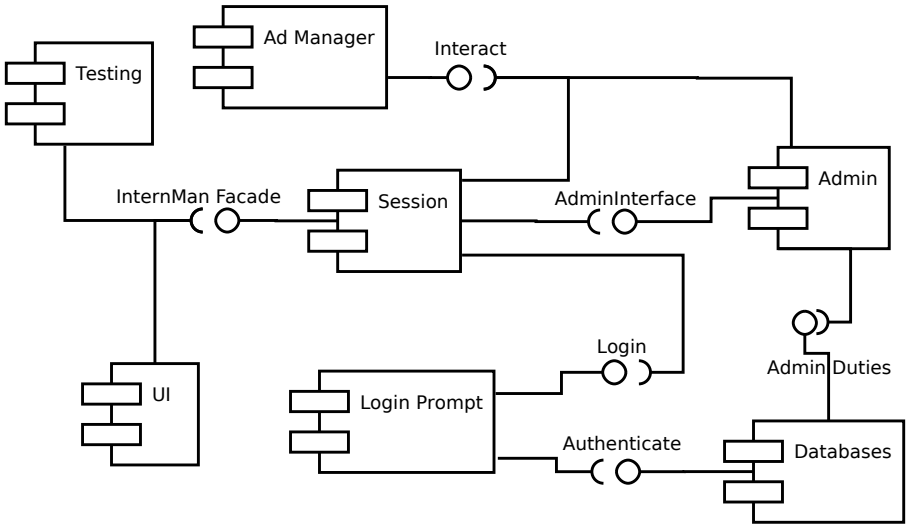
1.4 Document Status and Schedule

This document is the draft version of the deliverable and is subject to major change.
Change log:

1. Updated title page to be consistent with other deliverables.
2. Updated introduction section to include requirements specification document link and states that the document contains the acceptance test plan.
3. Updated system architecture diagram based on feedback.
4. Updated class diagrams and state diagrams.
5. Updated API specification.
6. Updated acceptance test plan to reflect actual test plan.
7. Updated with respect to forum information.

2 System Architecture

2.1 Diagram



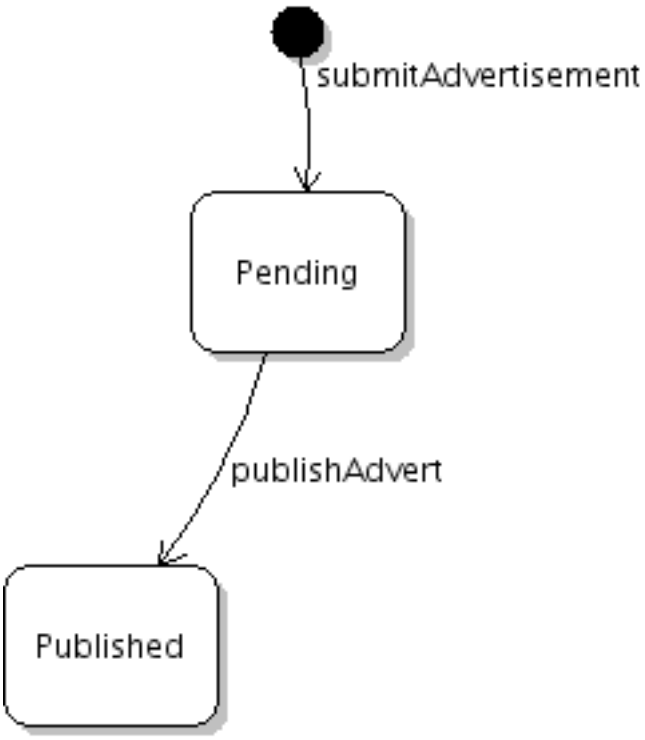
2.2 Rationale

We decided upon this system architecture as any component of the system should have the properties of being interchangeable with another substitutable component (substitutable defined as being having the same API). For this reason, we grouped related functionalities together and made them into a component. The 'Ad Manager' and the 'Databases' components could be combined together into one large 'Database' component, but we did not want to create a coupling between these components - the 'Ad Manager' may well have a database inside it, but the information inside this database is managed in a different way from the other data in the system such as students and companies.
All other components are defined by the primary reason that any component in the system can be replaced with no effect to the overall working of the system.
The 'Session' component is essentially acting as a broker to user requests as it implements the InternMan facade. The facade and corresponding implementation simplifies the user interface's access to the functionality of the system and further aids the ability for components to be swapped out with others as only one component, the 'Session' may have to be modified.

3 State Charts

3.1 Advertisement

3.1.1 Diagram

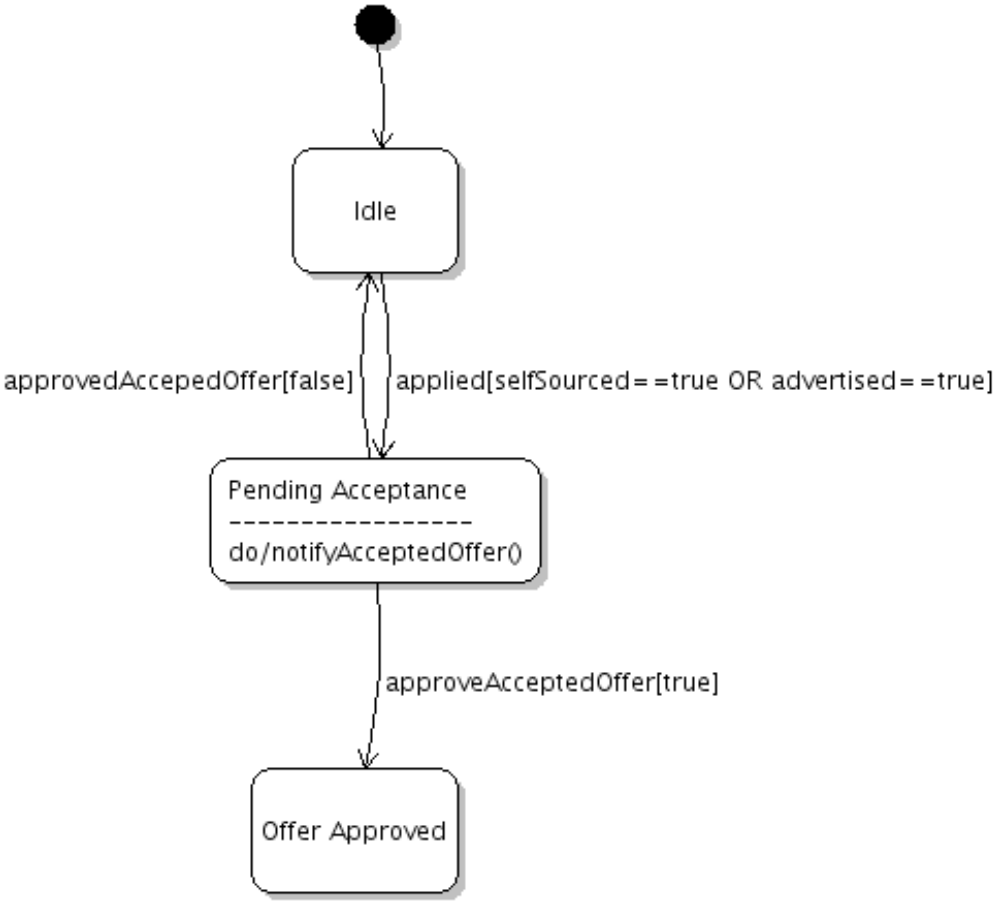


3.1.2 Rationale

The form of the state chart was decided based on the simplistic nature of an advertisements status. All adverts start off life as pending, after a company has made the initial submission. They can be revised at any time prior to either being declined by the course coordinator, where they are removed from the system and feedback sent to the company (outside the scope of the system) or published for viewing.

3.2 Student

3.2.1 Diagram



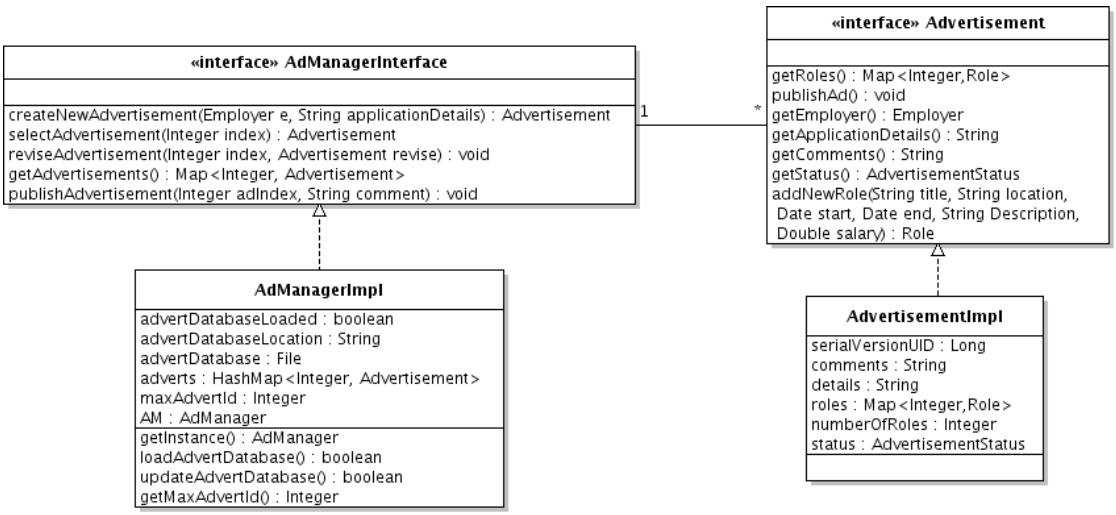
3.2.2 Rationale

We decided to create a state chart for a Student's flow through the system as to clarify the functionality of the system with relation to the student. A student continually changes state between 'Idle' and 'Pending Acceptance' (through either 'Applied for Position' or 'Found Self Sourced Role'). This can potentially happen multiple times until a suitable set of positions has been found that span the required internship time span and only then they will progress on to 'Offer Approved' which is the final resting state for students.

4 Class Diagrams and API Specifications

4.1 Advert Manager

4.1.1 Class Diagram



4.1.2 API Specification

Full name: public abstract interface AdManager

Package: uk.ac.glasgow.internman.AdManager
Each of the 3 available user types will be able to interact with this interface but if the user has course coordinator rights then all available adverts can be manipulated (published, denied etc), if they have registered employer rights they'll be able to create/append and submit adverts for review and if they are of type student, they'll only be able to view all published adverts.
Methods

- **public Advertisement createNewAdvertisement(Employer e, String applicationDetails)**

  Create a new advert with the specified details

  Preconditions: User must be of type Course Coordinator or Employer

  Postconditions: Advert is created and appended to a list of adverts awaiting review within the database.

- **public Advertisement selectAdvertisement(Integer index)**

  Given an index, select and display the appropriate advert from the database.

  Preconditions: Must be a valid index

  Postconditions: Advert is displayed after the index was successfully matched in the database.

- **public void reviseAdvertisement(Integer index, Advertisement revise)**

  Using an index to select an existing advert, change its contents with that of another supplied.

  Preconditions: Must be a valid index

  Postconditions: Advert is successfully replaced with the revised one and the changes can be seen dependant on its state, by others.

- **public Map getAdvertisements()**

  For use mostly by students who wish to see all available published adverts.

  Preconditions: Must be at least one published advert

  Postconditions: Advert(s) are displayed one after another in the UI with their descriptions displayed.

- **public void publishAdvertisement(Integer adIndex, String comment)**

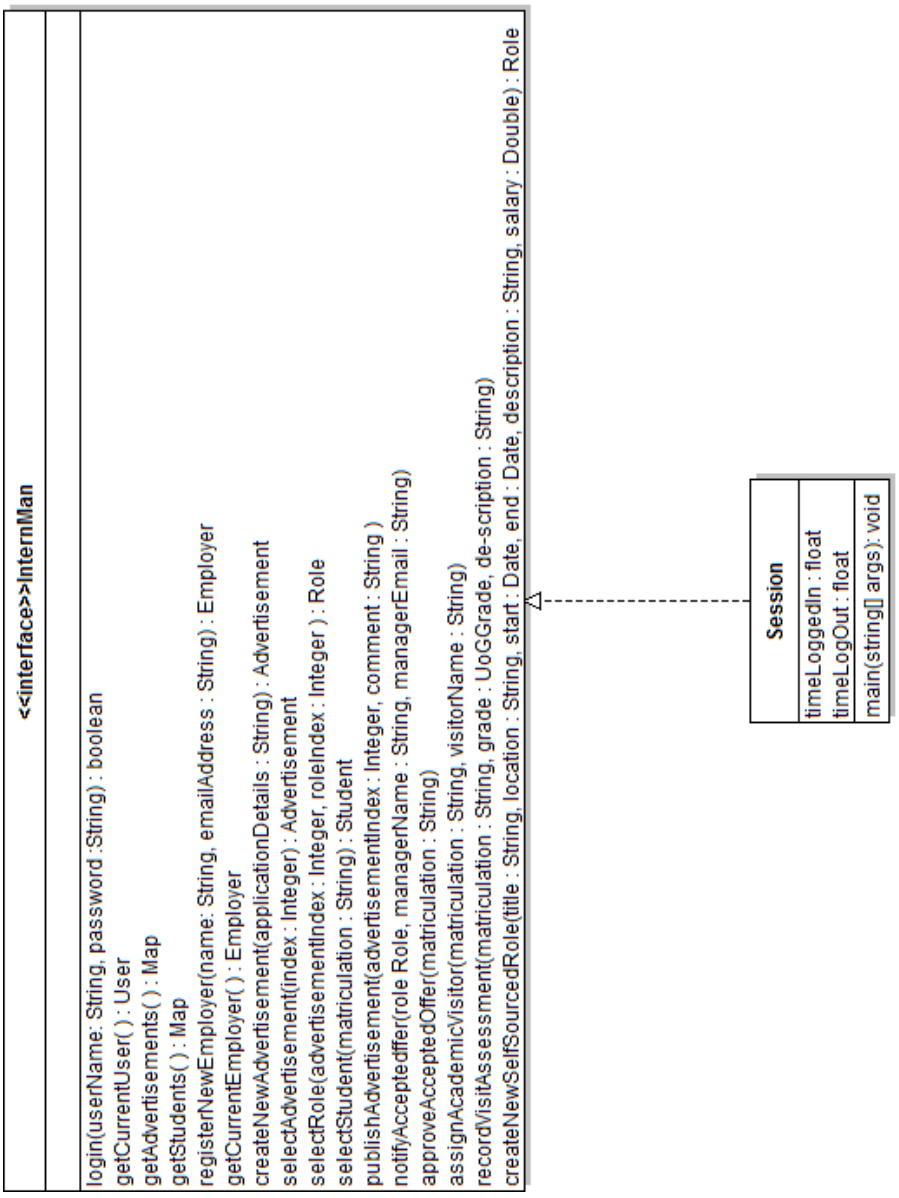
  Course coordinator only, select an advert and give feedback to the employer on why the advert was published.

  Preconditions: Must be a valid index, comment must be supplied.

  Postconditions: State of the selected advert is changed to published and can be successfully viewed by students.

4.2 Session

4.2.1 Class Diagram

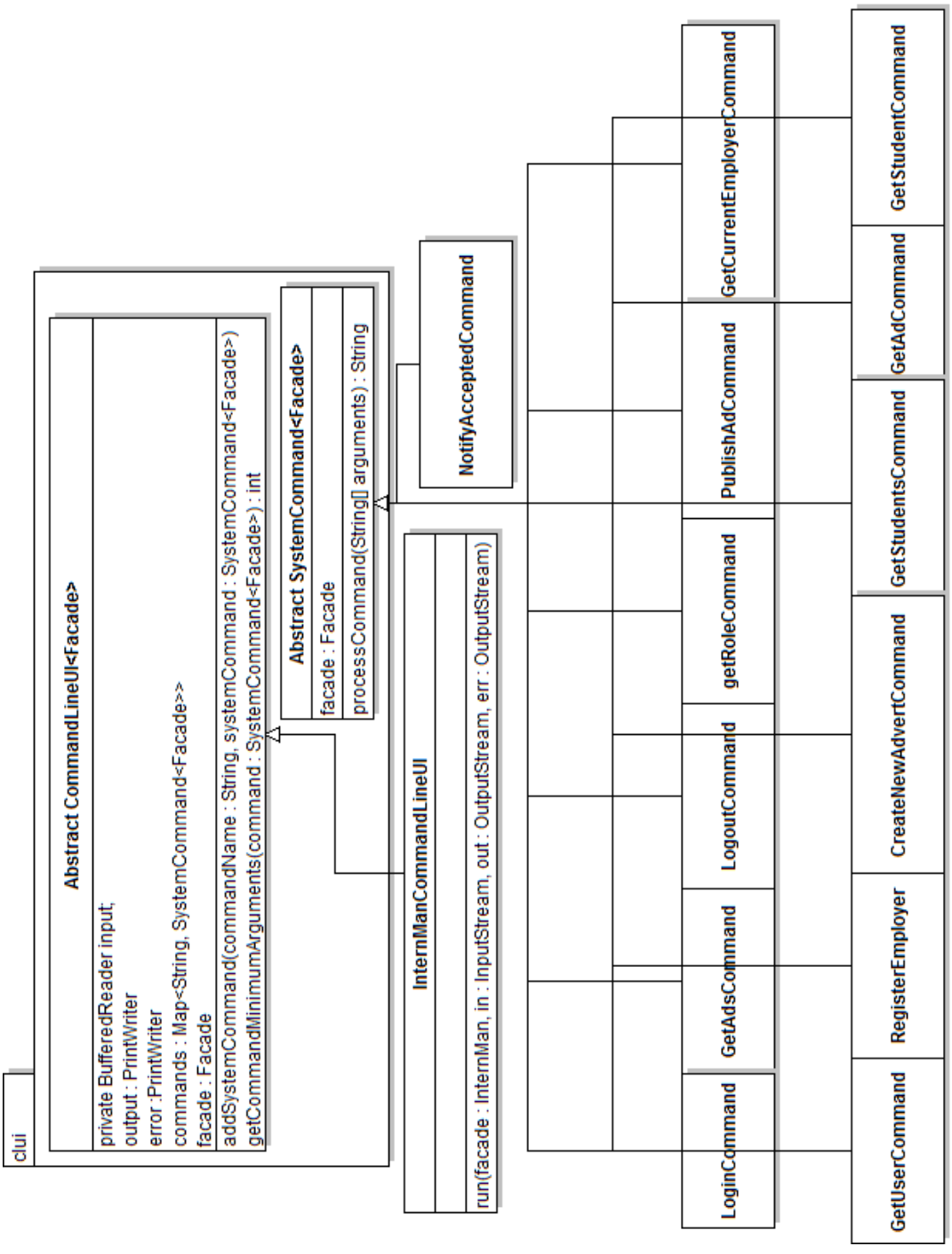


4.2.2 API Specification

Session is purely the implementation of the supplied internship management system API and doesn't supply functionality in its own right.

4.3 User Interface

4.3.1 Class Diagram



4.3.2 API Specification

Full name: public abstract interface InternManCommanLineUI

Package: uk.ac.glasgow.internman.ui

This is the interface for the graphical user interface.

- **public void run(Facade facade, InputStream in, OutputStream out, OutputStream err)**

  Runs the UI.

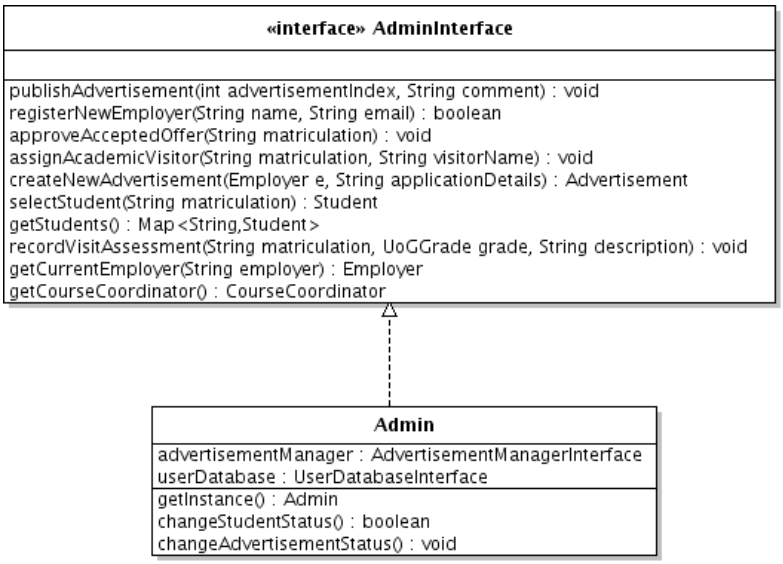
  Preconditions: Facade must be constructed.

  Invariants:

  Postconditions: UI is displayed.

4.4 Admin

4.4.1 Class Diagram



4.4.2 API Specification

Full name: public abstract interface Admin

Package: uk.ac.glasgow.internman.admin
This is the interface for the administration part of the database, accessible by only the course coordinator. This assumes that the user using this interface has Course Coordinator rights.

- **public void publishAdvertisement(Integer advertisementIndex, String comment)**

  Approves an advertisement and makes it visible to students. Sends comment as feedback to the employer as an email.

  Preconditions: Advertisement must be in advertisement database.

  Invariants:

  Postconditions: Advertisement is now approved.

- **public boolean registerNewEmployer(String name, String email)**

  Interface to add a company to the database, functionality is delegated to the database component. Error checking is done inside this function, but not whether or not a company currently exists inside the companies database.

  Preconditions:

  Invariants:

  Postconditions: The return value of the delegated function indicates the success of this function.

- **public void approveAcceptedOffer(String matriculation)**

  Approves the offer most recently accepted by the student with this matriculation id. **Preconditions:** Student with this matriculation must have a successful application and must have notified the Course Coordinator of their success.

  Invariants: Student must not accept another successful application during this review process.

  Postconditions: Students status is changed to a success status.

- **public void assignAcademicVisitor(String matriculation, String visitorName)** Sends email to student, visitor and employer manager to let them know about assignment.

  Preconditions: An academic visit cannot be currently assigned.

  Invariants:

  Postconditions:

- **public Advertisement createNewAdvertisement(Employer e, String applicationDetails)** Adds a new (not-yet-approved) advertisement into the system.

  Preconditions: Advertisement should not all ready exist in the system.

  Invariants:

  Postconditions:

- **public Student selectStudent(String matriculation)** Gets handle on Student from supplied matriculation.

  Preconditions: Null returned if Student is not in the map.

  Invariants:

  Postconditions:

- **public void recordVisitAssessment(String matriculation, UoGGrade grade, String description)** Records grade for visit. Email to student and employer.

  Preconditions: A visit assessment should not currently be assigned for this student

  Invariants:

  Postconditions:

- **public Map <String,Student >getStudents()** Returns a Mapping from matriculation to Student for every student in the system.

  Preconditions:

  Invariants:

  Postconditions:

- **public Employer getCurrentEmployer(String employer)** Selects the employer given by the String “Employer”.

  Preconditions: Employer employer should exist within the system.

  Invariants:

  Postconditions:

- **public CourseCoordinator getCourseCoordinator()** Gets the systems current Course Coordinator.

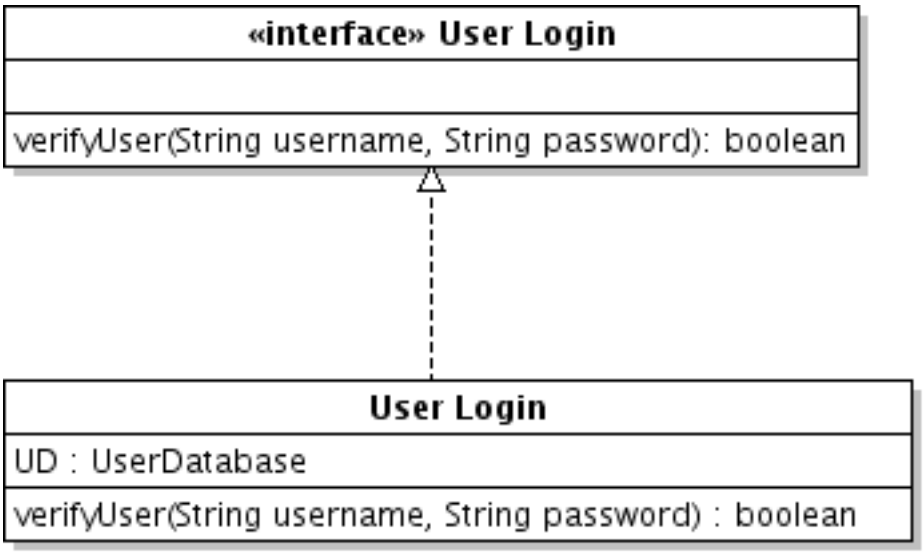
  Preconditions:

  Invariants:

  Postconditions:

4.5 Login Prompt

4.5.1 Class Diagram



4.5.2 API Specification

Full name: public abstract interface Login

Package: uk.ac.glasgow.internman.login
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.

Methods

- **public Boolean verifyUser(String Id,String password)**

  Check if a user is in database

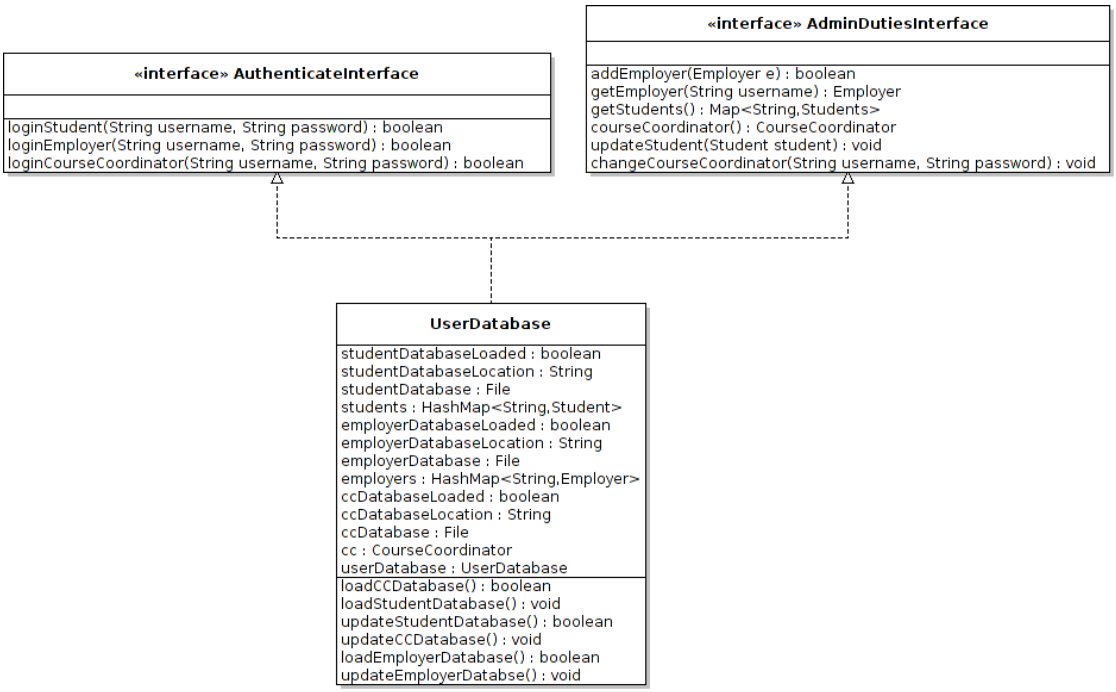
  Preconditions: No user is currently logged in at the same user interface.

  Invariants:

  Postconditions: User is logged in, if username and password combination is correct.

4.6 Database

4.6.1 Class Diagram



4.6.2 API Specification

Full name: public abstract interface UserDatabase

Package: uk.ac.glasgow.internman.users
This is the interface for the user database. This database will hold information on students, companies, and the course coordinator. For University students and staff members, the database will access the MyCampus GUID system for login. For companies, a separate store for their user information will be used.

Methods

- **public boolean addEmployer(Employer e)**

  Add information about an employer to the database.

  Preconditions: Database must not already contain details of the employer.

  Invariants:

  Postconditions: Database now contains information about an employer including their login password.

- **public Employer getEmployer(String username)**

  Gets the employer with the given username.

  Preconditions:

  Invariants:

  Postconditions: If a valid name is given, the object associated with the company is returned, otherwise null is returned.

- **public Map¡String,Student¿ getStudents()**

  Returns the mapping of String matriculation to Student's in the system.

  Preconditions:

  Invariants:

  Postconditions:

- **public void updateStudent(Student student)**

  Updates the user specific data for the supplied student. For example, when a student obtains an internship, or has had their internship assessed.

  Preconditions: Student must be a valid Computing Science student and is known to the application's user database.

  Invariants:

  Postconditions: Student's user specific information is up to date.

- **public CourseCoordinator courseCoordinator()**

  Returns the current course coordinator of the system.

  Preconditions:

  Invariants:

  Postconditions: If a course coordinator exists within the system then they are returned (providing that the current user has sufficient rights to do so) and if it does not then null is returned.

- **public void changeCourseCoordinator(String username, String password)** Changes the current Course Coordinators login details to the ones supplied by the arguments username and password.

  Preconditions:

  Invariants:

  Postconditions:

- **public boolean loginStudent(String username, String password)** Returns whether or not the given username and password are valid entries in the respective database.

  Preconditions:

Invariants:

Postconditions:

- **public boolean loginEmployer(String username, String password)** Returns whether or not the given username and password are valid entries in the respective database.

  Preconditions:

  Invariants:

  Postconditions:

- **public boolean loginCourseCoordinator(String username, String password)** Returns whether or not the given username and password are valid entries in the respective database.

  Preconditions:

  Invariants:

  Postconditions:

5 Acceptance Test Plan

Identifier	UtilityAAO1
Use Case	Approve Accepted Offer
Description	Login as course coordinator and accept a student's internship.
Setup	A course coordinator and a student must be users in the system before this test case is to be executed. If these users do not exist, they must be added prior to execution. A successful application for the referred student must also exist, and should be added and assigned if it does not.
Interface	uk/ac/glasgow/internman
Includes	UtilitTCLogin7
Expected Outcome	Student now has their internship approved.

Identifier	UtilityAAO2
Use Case	Approve Accepted Offer
Description	Login as student and accept a student's internship.
Setup	A student with a successful application must exist in the system. If this user does not exist, or if the successful application does not exist then they should be created prior to execution.
Interface	uk/ac/glasgow/internman
Includes	UtilityTCLogin1
Expected Outcome	Student 100000 has their internship still as ACCEPTED.

Identifier	UtilityAAO3
Use Case	Approve Accepted Offer
Description	Login as course coordinator and accept a student's internship. The student does not exist.
Setup	A course coordinator should exist within the system. If this user does not exist then it should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	No change and program doesn't crash.

Identifier	UtilityTCLogin1
Use Case	Login
Description	Login a valid student.
Setup	A Student must exist in the system. If this user does not exist in the system then it should be created prior to theo execution.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Student logged in, Student is now the current user in the system.

Identifier	UtilityTCLogin2
Use Case	Login
Description	Login an invalid student. Valid username, invalid password
Setup	Student should exist in the system, if this is not the case then this user should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Student not logged in, current user not changed.

Identifier	UtilityTCLogin3
Use Case	Login
Description	Login an invalid student. Invalid username
Setup	No setup required.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Student not logged in, current user not changed.

Identifier	UtilityTCLogin4
Use Case	Login
Description	Login a valid employer.
Setup	Employer should exist in the system, if this is not the case then this user should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Employer logged in, Employer is set as current user in the system.

Identifier	UtilityTCLogin5
Use Case	Login
Description	Login an invalid employer. Valid username, invalid password
Setup	Employer should exist in the system, if this is not the case then this user should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Employer not logged in, current user is not changed.

Identifier	UtilityTCLogin6
Use Case	Login
Description	Login an invalid employer. Invalid username.
Setup	No setup required.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Employer not logged in, current user is not changed.

Identifier	UtilityTCLogin7
Use Case	Login
Description	Login a valid course coordinator.
Setup	Course Coordinator must exist inside the system, if this is not the case then this user should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Course coordinator logged in, Course Coordinator is now current user.

Identifier	UtilityTCLogin8
Use Case	Login
Description	Login an invalid course coordinator. Valid username, invalid password.
Setup	Course Coordinator must exist inside the system, if this is not the case then this user should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Course coordinator not logged in, current user is not changed.

Identifier	UtilityTCLogin9
Use Case	Login
Description	Login an invalid course coordinator. Invalid username.
Setup	No setup required.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Course coordinator not logged in, current user is not changed.

Identifier	UtilityNAO1
Use Case	Notify Accepted Offer
Description	Login as course coordinator and notify course coordinator of an internship.
Setup	Course coordinator should exist within the system and an internship should also exist within the system. If these entities do not exist then they should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Attempt should fail, course coordinators do not do internships.

Identifier	UtilityNAO2
Use Case	Notify Accepted Offer
Description	Login as an employer and notify course coordinator of an internship.
Setup	An employer should exist within the system. If this is not the case then it should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	Nothing, employers do not do internships.

Identifier	UtilityNAO3
Use Case	Notify Accepted Offer
Description	Login as course coordinator and accept a student's internship. The student does not exist.
Setup	A course coordinator and a student with a valid and successful internship should exist within the system, if this is not the case then these entities should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	No change and program doesn't crash.

Identifier	UtilityPA1
Use Case	Publish Advertisement
Description	Login as course coordinator publish an advertisement. View as a student.
Setup	A course coordinator, a student and a pending advertisement must exist within the system, if these entities do not exist then they should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7, UtilityTCLogin1
Expected Outcome	Advertisement is published and available to be viewed by a student.

Identifier	UtilityPA2
Use Case	Publish Advertisement
Description	Login as course coordinator and publish an advertisement. View as course coordinator.
Setup	A course coordinator and a pending advertisement must exist within the system. If these entities do not exist then they should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	Advertisement is published and available to be viewed by the course coordinator.

Identifier	UtilityPA3
Use Case	Publish Advertisement
Description	Login as course coordinator publish an advertisement. View as employer who owns advertisement.
Setup	A course coordinator, employer and a pending advertisement must exist within the system. If these entites do not exist then they should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7, UtilityTCLogin4
Expected Outcome	Advertisement is published and available to be viewed by the owner employer.

Identifier	UtilityPA4
Use Case	Publish Advertisement
Description	Login as course coordinator and publish an advertisement that doesn't exist, then try to get a student to view it.
Setup	A course coordinator and a student must exist within the system. If these entites do not exist then they should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7, UtilityTCLogin1
Expected Outcome	Nothing happens as advert doesn't exist.

Identifier	UtilityPA5
Use Case	Publish Advertisement
Description	Login as course coordinator publish an advertisement that doesn't exist, then try to get the course coordinator to view it.
Setup	A course coordinator must exist within the system. If this is not the case then this entity must be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	Nothing happens as advert doesn't exist.

Identifier	UtilityPA6
Use Case	Publish Advertisement
Description	Login as course coordinator publish an advertisement that doesn't exist, then try to get an employer to view it.
Setup	A course coordinator and an employer must exist within this system. If this is not the case then these entities must be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7, UtilityTCLogin4
Expected Outcome	Nothing happens as advert doesn't exist.

Identifier	UtilityRE1
Use Case	Register Employer
Description	Login as course coordinator and add a new employer.
Setup	A course coordinator must exist within the system. If this is not the case then this entity must be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	New employer added to the database.

Identifier	UtilityRE2
Use Case	Register Employer
Description	Login as student and add a new employer.
Setup	A student must exist within the system. If this is not the case then this entity must be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin1
Expected Outcome	Nothing, student cannot add employers.

Identifier	UtilityRE3
Description	Login as employer and add a new employer.
Setup	An employer must exist within the system. If this is not the case then this entity must be created within the system prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	Nothing, employers cannot add employers.

Identifier	UtilitySA1
Use Case Description	Submit Advertisement Login as employer and submit an advertisement.
Setup	An employer must exist within the system. If this is not the case then this entity must be created within the system prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	New advertisement to be published added to the database.

Identifier	UtilitySA2
Use Case Description	Submit Advertisement Login as course coordinator and submit an advertisement.
Setup	A course coordinator must exist within the system. If this is not the case then this entity must be created within the system prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	New advertisement to be published added to the database.

Identifier	UtilitySA3
Use Case	Submit Advertisement
Description	Login as student and submit an advertisement.
Setup	A student must exist within the system. If this is not the case then this entity must be created within the system prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin1
Expected Outcome	Nothing, students cannot submit advertisements.

Identifier	UtilitySA4
Use Case	Submit Advertisement
Description	Don't login and submit an advertisement.
Setup	No setup is required.
Interface	/uk/ac/glasgow/internman
Includes	
Expected Outcome	Nothing, you need to login to submit advertisements.

Identifier	UtilityVA1
Use Case	View Advertisement
Description	Login as student and view a published advertisement.
Setup	A student and a published advertisement must exist within the system. If this is not the case then these entites should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin1
Expected Outcome	Advertisement details available.

Identifier	UtilityVA2
Use Case	View Advertisement
Description	Login as course coordinator and view a published advertisement.
Setup	A course coordinator and a published advertisement must exist within the system. If this is not the case then these entities should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	Advertisement details available.

Identifier	UtilityVA3
Use Case	View Advertisement
Description	Login as owner company and view their published advertisement.
Setup	An employer who owns a published advertisement, and the published advertisement must exist within the system. If this is not the case then these entities should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	Advertisement details available.

Identifier	UtilityVA4
Use Case	View Advertisement
Description	Login as an employer and view another employers published advertisement.
Setup	An employer and a published advertisement belonging to another employer should exist within the system. If this is not the case then these entities should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	Advertisement details unavailable.

Identifier	UtilityVA5
Use Case	View Advertisement
Description	Login as course coordinator and view a pending advertisement.
Setup	A course coordinator and a pending advertisement should exist within the system. If this is not the case then these entites should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	Advertisement details available.

Identifier	UtilityVA6
Use Case	View Advertisement
Description	Login as student and view a pending advertisement.
Setup	A student and a pending advertisement should exist within the system. If this is not the case then these entities should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin1
Expected Outcome	Advertisement details unavailable.

Identifier	UtilityVA7
Use Case	View Advertisement
Description	Login as owner company and view their pending advertisement.
Setup	An employer and a pending advertisement belonging to the employer must exist within the system. If this is not the case then these entities should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	Advertisement details available.

Identifier	UtilityVA8
Use Case	View Advertisement
Date Tested	18/02/2013
Description	Login as some employer and view another's pending advertisement.
Setup	An employer and a pending advertisement belonging to another employer must exist within the system. If this is not the case then these entities should be created prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	Advertisement details available.

Identifier	UtilityVS1
Use Case	View Student
Description	Login as course coordinator and view a student.
Setup	A course coordinator and a student should exist within the system. If this is not the case thene these entities should be created within the system prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	Student details available.

Identifier	UtilityVS2
Use Case	View Student
Description	Login as course coordinator and view a nonexistent student.
Setup	A course coordinator should exist within the system. If this is not the case then these entities should be created within the system prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin7
Expected Outcome	Nothing, student doesn't exist.

Identifier	UtilityVS3
Use Case	View Student
Description	Login as employer and view a student.
Setup	An employer and a student should exist within the system. If this is not the case then these entities should be created within the system prior to execution.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin4
Expected Outcome	Student details unavailable, employers cannot view students.

Identifier	UtilityVS4
Use Case	View Student
Description	Login as student and view a student.
Setup	Two students should exist within the system. If this is not the case then these entities should be created within the system.
Interface	/uk/ac/glasgow/internman
Includes	UtilityTCLogin1
Expected Outcome	Student details unavailable, students cannot view students.