Professional Software Development 3

Acceptance Test Plan

Team L

Peeranat Fupongsiripan, Dan Tomosoiu,
Michael Kilian, Tony Lau,
Hector Grebbell

This Document Describes the Acceptance Test Plan for each component within our design of an Internship Management System for the Department of Computing Science at the University of Glasgow. The purpose of this acceptance test is to make sure the system developed meets its system requirements and is suitable for all accepted use cases.

Test Cases Prefixed A can be performed with an automatic test harness. Any Prefixed M must be completed manually.

Authenticator

Test Case	A001
Use Case	The Authenicator is passed a valid username and password for a user registered with the MyCampus System
Input Specification	Multiple known MyCampus Username-Password sets are fed into the authenticator.
Output Specification	For each input the authenticator should return the correct User item associated with the username given.

Test Case	A002
Use Case	The Authenicator is passed a valid username and password for a user registered within the Internship Management System
Input Specification	Multiple known Username-Password sets from the system database are fed into the authenticator.
Output Specification	For each input the authenticator should return the correct User item associated with the username passed in.

Test Case	A003
Use Case	The Authenicator is passed a valid username with incorrect password for a user registered with the MyCampus System
Input Specification	Multiple known MyCampus Usernames are fed into the authenticator with incorrect passwords.
Output Specification	For each input the authenticator should return null.

Test Case	A004
Use Case	The Authenicator is passed a valid username with incorrect password for a user registered within the Internship Management System
Input Specification	Multiple known Usernames from the system database are fed into the authenticator with incorrect passwords.
Output Specification	For each input the authenticator should return null.

Test Case	A005
Use Case	The Authenicator is passed an invalid username with a correct password for a user (with a different username) registered with the MyCampus System
Input Specification	Multiple Usernames known not to exist within the My-Campus System are fed into the authenticator with correct passwords for other users.
Output Specification	For each input the authenticator should return null.

Test Case	A006
Use Case	The Authenicator is passed a valid username with incorrect password for a user registered within the Internship Management System
Input Specification	Multiple Usernames known not to exist within the system database are fed into the authenticator with correct passwords for other users.
Output Specification	For each input the authenticator should return null.

Test Case	A006
Use Case	The Authenicator is passed an invalid username with a password not within the Intern Management or My-Campus systems.
Input Specification	Multiple Username-Password Sets known not to exist within either system are fed into the authenticator.
Output Specification	For each input the authenticator should return null.

UserStore

Test Case	A007
Use Case	A logged in Coordinator wishes to create a new General User Account
Input Specification	addUser() is called with the correct arguments for a specific new user. getUser() (also from this component) and authenticate() (From the authenticator) should then be called with the correct details for the new user.
Output Specification	Both getUser() and authenticate() should return the correct user item (Only valid once tests A002 and A0011).

Test Case	A008
Use Case	A logged in Coordinator wishes to create a new Student User Account
Input Specification	addStudent() is called with the correct arguments for a specific new user. getUser(), getStudent() (also from this component) and authenticate() (From the authenticator) should then be called with the correct details for the new user.
Output Specification	Both getUser() and authenticate() should return the correct user item. getStudent() should return the correct student item (Only valid once tests A002, A011 and A013).

Test Case	A009
Use Case	A logged in Coordinator wishes to create a new Employer User Account
Input Specification	addEmployer() is called with the correct arguments for a specific new user. getUser(), getEmployer() (also from this component) and authenticate() (From the authenticator) should then be called with the correct details for the new user.
Output Specification	Both getUser() and authenticate() should return the correct user item. getEmployer() should return the correct employer item (Only valid once tests A002, A011 and A014).

Test Case	A010
Use Case	A logged in Coordinator wishes to create a new Visitor User Account
Input Specification	addVisitor() is called with the correct arguments for a specific new user. getUser(), getVisitor() (also from this component) and authenticate() (From the authenticator) should then be called with the correct details for the new user.
Output Specification	Both getUser() and authenticate() should return the correct user item. getVisitor() should return the correct visitor item (Only valid once tests A002, A011 and A015).

Test Case	A011
Use Case	The System wishes to retrieve a User item from the UserStore
Input Specification	A valid username is passed to the getUser() function.
Output Specification	The correct User item is returned

Test Case	A012
Use Case	The System wishes to retrieve a Coordinator item from the UserStore
Input Specification	A valid username for the coordinator is passed to the getCoordinator() function.
Output Specification	The correct Coordinator item is returned

Test Case	A013
Use Case	The System wishes to retrieve a Student item from the UserStore
Input Specification	A valid username for the student is passed to the getStudent() function.
Output Specification	The correct Student item is returned

Test Case	A014
Use Case	The System wishes to retrieve an Employer item from the UserStore
Input Specification	A valid username for the Employer is passed to the getEmployer() function.
Output Specification	The correct Employer item is returned

Test Case	A015
Use Case	The System wishes to retrieve a Visitor item from the UserStore
Input Specification	A valid username for the Visitor is passed to the getVisitor() function.
Output Specification	The correct Visitor item is returned

OfferManager

Test Case	A016
Use Case	A student wishes to inform the system of an accepted offer
Input Specification	notifyAcceptedOffer() is called with a range of acceptable variables. getStatus (from the placement store) should then be called on each placement
Output Specification	When getStatus is called on each placement it should return Rejected, not an error

Test Case	A017
Use Case	The Course Co-ordinator wishes to approve a placement
Input Specification	approveAcceptedOffer() should be called on a range of placements. getStatus (from the placement store) should then be called on each placement.
Output Specification	getStatus should return Accepted

Test Case	A018
Use Case	A Student wishes to inform the system of an accepted role from outside of the system.
Input Specification	createNewSelfSourcedRole() is called with a range of acceptable variables.
Output Specification	The function should return the correct role.

PlacementStore

Test Case	A019
Use Case	The user wishes to view all placements
Input Specification	viewPlacements() should be called
Output Specification	The UI should list all placements

Test Case	A020
Use Case	A user wishes to add a placement
Input Specification	addPlacement should be called with a range of placement items
Output Specification	When eviewPlacements() is called, the new placements should also be displayed

Test Case	A006
Use Case	The coordinator wishes to remove a placement
Input Specification	removePlacement is called with the correct placement id
Output Specification	listPlacements no longer displays the placement.