

storyname	CV Storage
description	Candidates upload and store their CVs (there may be more than one CV since CVs are designed for a particular job)We may be able to enable the user to group their CVs if they have lots (probably unnecessary). E.g. if a user wants to apply for part time IT
initiating_event	Candidate clicks > 'Manage CVs'
memory_context	Database stores all CV's for candidates.
functional_tests	Uploading new / removing CVs from the databaseEditing offline and uploading new version of same CV (overwriting)
quality_attributes	Candidates must be able to send a CV to apply for a particular jobopening. If the CV management page fails perhaps the messaging tool will allow CV sending/uploading.
observable_result	A page lists your CVs complete with upload_new/remove/edit buttons

storyname	User Login
description	Allow users who already have accounts to access member services and restrict access to non members.
initiating_event	User selects a service only available to members and is not currently logged in.
memory_context	Database access of user accounts.
functional_tests	Test with correct username and password.Test with correct username but incorrect password.Test with incorrect username but correct password.Test with incorrect username and password.
quality_attributes	If a user has already selected a particular member service then they should be presented with that service rather than a general managment screen.Upon login failure a suitable error message should appear along with Password Retrieval
observable_result	Success: User is presented with service they initially requested.Fail: User presented with error message.

storyname	Compose Message
description	The user creates a new message from their account which can be sent to any user on the system or the site's administrator.
initiating_event	User clicks Compose or Reply once logged into the system
memory_context	The composed message is stored in the db
functional_tests	The users need to be able to compose messages and send them to other users. The system needs to check for the existence of the current recipient. The system needs to check that the user is using their own username and not anyone else's to send the message. Su
quality_attributes	Messaging is an important feature of the system and will be the focus of the client's novel system
observable_result	Message passed to send method

storyname	Send Message
description	This task will allow users to send messages from their inbox (using compose) or forward previous messages to another user.
initiating_event	User finished composing a message clicks Send
memory_context	Message stored on server database
functional_tests	We need to check that the IM sending script works correctly. The functional tests will involve checking the recipient's inbox to see if the message was sent correctly.
quality_attributes	IM sending is a very important task and there can be no room for errors here
observable_result	Success message if the mail is sent correctly.

storyname	View inbox
description	Users can view all of their new messages as well as previous messages sent to them on the site.
initiating_event	User logs in clicks Inbox
memory_context	The user's messages are loaded from the messages database
functional_tests	The user must be able to view their inbox and any of the messages stored for them on the system.
quality_attributes	
observable_result	The user is presented with a summary of their messages.

storyname	Read Message
description	Users can view specific messages in their inbox and forward them if necessary
initiating_event	User logs into their account and wishes to read a message entry in their inbox
memory_context	Access the particular message from the database
functional_tests	Users need to be able to view messages from potential employers and if necessary Support. Access tests need to be done on new messages as well as messages stored in the Old/Past messages folder.
quality_attributes	As the Instant Messaging aspect of the system is considered quite important this function needs to function correctly.
observable_result	A page is generated with the desired message and controls allowing access to other account features

storyname	View your job openings
description	Companies can view their job openings in terms of-> Name description of job-> number of candidates shortlisted-> Number of candidates invited to interview -> # accepted # declined # pending
initiating_event	Company logs in job openings are displayed
memory_context	Openings stored in Job Openings table
functional_tests	Add a new job -> The listing will show the new jobDelete a job -> listing with remove the jobShortlist / invite candidates -> status figures will updateCandidate declines / accepts job -> status figures will updateDelete all job openings -> Page displays
quality_attributes	Page should take no longer than 20 seconds to load on a 56k modem
observable_result	Company can see status of all their job openings

storyname	Register User
description	Users register to use additional services that require accounts to enable access. This is also so that personal data can be kept private and not be accessed by unauthorised users.
initiating_event	User attempts to use member services that require registration
memory_context	Database access required for creating a New user
functional_tests	The user needs to be able to create a new account by filling out an online form with the fields : Username Password Name E-Mail.If they attempt to create a username that is already taken the system will output an error message.The system will check if
quality_attributes	User Registration must function correctly as users will not be able to use the core functions of the Recruitment system unless they are registered users in the database.
observable_result	User is now registered

storyname	Add candidates to shortlist
description	After searching for candidates a company can shortlist the ones they like either to the company generic shortlist or to one/many particular job shortlist(s).
initiating_event	Company selects candidate(s) from search results to add to generic/job shortlist
memory_context	Will be using the candidates table and the job-candidate pair table
functional_tests	Add a candidate to a generic shortlist -> candidate should appear on companies' generic shortlist. Add the candidate to a shortlist he is already on -> error message saying he is already on that shortlist. Add multiple candidates to a generic shortlist -> A
quality_attributes	Companies should be able to add more than one candidate to a shortlist without having to reload the search results page. E.G. tickboxes next to each candidate and a 'ADD ALL SELECTED TO SHORTLIST' button at the bottom/top.
observable_result	Either the companies generic shortlist or a particular job's shortlist will be updated to include the new candidates.

storyname	Edit Calendar
description	Candidates use the calendar to show clients when they are available for interview. E.G july-september. At the moment we are not implementing a system for booking interviews onto the calendar it is just for the companies convenience.
initiating_event	Logged in candidate clicks 'Edit Interview Availability Calendar' option from main user menu
memory_context	Available dates are stored in either the candidates database or a seperate calendar object linked to by the candidates table.
functional_tests	Select a month that you are available for interview -> The calendar turns a different colour for that month. Clients with your CV details check your calendar and they can see the different colour. Select individual days that you are available for interview
quality_attributes	System must be able to function without the calendar system. Interviews can be set up via messaging instead.
observable_result	Calendar days change colour depending on whether the candidate is available during that period or not.

storyname	Pay -5 for a candidate's contact details
description	After a company has messaged a candidate and wants to invite them to interview they must pay -5 and then they receive the candidates contact details and are free to arrange interview times and jobs by themselves.
initiating_event	?
memory_context	?
functional_tests	TODO
quality_attributes	Payment system must not overcharge customers as this is illegal.
observable_result	?

storyname	Search for a candidate
description	A company can search for candidates matching a particular job opening or a generic search. The search will pull up a list of matching candidates and their CVs. (without personal/contact information)
initiating_event	Company clicks 'Search for candidates' from main menu OR from a particular job opening
memory_context	The system searches through the CVs table that match the keywords entered by the user
functional_tests	Enter no information -> Page asks you enter information and reloads Enter information -> Page pulls up possible candidates ordered by whatever the user selected (default green traffic light date last logged on) Enter information and no users returned -> P
quality_attributes	Page should take no more than 30 seconds to load on 56k modem
observable_result	A Results page is generated displaying all job listings found that are related to the keywords entered

storyname	Move message to Trash
description	A User selects a message or group of messages and clicks the Move to trash button. They are prompted to confirm this and then the message is moved to a temporary folder (Trash) from where it can be deleted.
initiating_event	User selects message(s) clicks Move to Trash option
memory_context	The messages selected are flagged as in the Trash Folder but are not deleted from the system
functional_tests	The following checks need to be done:1. The messages that are moved to trash must not be visible in the inbox any longer but still must be accessible.
quality_attributes	The messages must be moved to the Trash Folder in under 30 seconds on any system.
observable_result	The selected messages are no longer visible in the Inbox but are now visible in the Trash folder

storyname	Export Messages
description	The User selects a message or group of messages they wish to export into files they can save on their system. They are presented with two or three format options and can then output the selected messages to file.
initiating_event	User selects messages to export presses Export
memory_context	Messages need to be retrieved from the database
functional_tests	We need to check the following:1. That users can select groups of messages and then choose an output format.2.We need to check that the database accesses all the messages selected.3.We need to check that messages can be output to the various formats available
quality_attributes	The ability to export messages does not have a high priority however both clients and candidates may wish to save messages for their own records and in this respect the function needs to be of a high quality. Tests will need to be done to ensure that all
observable_result	Messages are converted to a file format and the user is prompted to save this file

storyname	Retrieve message from Trash
description	The User selects messages currently in the Trash folder and presses the Retrieve button. The messages are then no longer flagged as being in the Trash folder and can be now viewed once more in the Inbox.
initiating_event	The user selects message(s) in the Trash folder and clicks Retrieve from Trash
memory_context	The messages are unflagged but not physically moved
functional_tests	The user needs to be able to Retrieve messages currently in the Trash folder. The following tests need to be done: When a user selects a message or group of messages and clicks retrieve the messages must all be unflagged and made available in the inbox.
quality_attributes	Messages must be retrievable within 2-3 minutes based on the number of messages selected.
observable_result	The messages are no longer visible in the Trash folder but can be viewed once more in the Inbox

storyname	Flag message containing personal info.
description	The User attempts to use personal information in their message such as Telephone numbers email addresses and URL's. When the system detects any of the above the message is flagged and all information related to it is sent to an administrator for manual review.
initiating_event	The user includes Phone Numbers Email addresses or URL's in a message they are sending
memory_context	The message can be stored in the Users sent folder but will not be available by the recipient until it has been verified.
functional_tests	The system needs to be tested with messages containing phone numbers email addresses and URL's. 1. Messages need to be composed from one type of account containing strings of numbers '@'s www.'s etc. 2. The system needs to be able to parse these messages.
quality_attributes	This function needs to be able to parse messages within 30 seconds so the user will not notice any work being done in the background.
observable_result	Details of the message are sent to the administrator for verification/editing before being made visible by the designated recipient

storyname	View candidate details/CV
description	On searching for candidates that fulfill a certain search criteria clients can view their CV details. This is basically a page summarising their skills employment and education histories details of their location desired salaries and positions as well
initiating_event	User searches for a candidate clicks View CV
memory_context	The CV details skills education and employment histories as well as other details must be accessed from the database
functional_tests	Checks:1. The system must be able to access all classes containing data related to the particular candidate whose CV is being accessed.2.The system must be able to generate a summary of this data without showing any personal information related to the ca
quality_attributes	The candidate's CV needs to be generated in under 30 seconds.
observable_result	A summary page of candidate details is generated

storyname	Reply to a message
description	The User clicks a message currently in their Inbox and selects the Reply to option. Details of the current message are then passed directly to the Compose Message function which will fill out the fields related to the current mail automatically.
initiating_event	User selects a message in their inbox and then selects the Reply To option
memory_context	The message details (from etc.) need to be accessed from the database so they can be used to fill in fields in Compose Message
functional_tests	We need to check that the system can automatically fill in fields in the Compose Message function.
quality_attributes	The filled in Compose Message page should take less than 20 seconds to load.
observable_result	A Compose message form is displayed with the required fields already filled

storyname	Register as a Client
description	When a Company wished to use additional member services provided by the site they need to register as a client. They are prompted for their username and password details and then asked for details about their company which will be used in their profile.
initiating_event	A member of a company wishing to use the site clicks Register
memory_context	The companies details are stored in the database
functional_tests	The user needs to be able to create a new account by filling out an online form with the fields : Username Password Name E-Mail and Company Details fieldsIf they attempt to create a username that is already taken the system will output an error messa
quality_attributes	The registration process needs to take less than 3 mintues.
observable_result	The company are now registered as a Client

storyname	Admin approve candidate profile
description	When a candidate submits a profile it is sent to an administrator so that all details can be verified to be true. This involves calling them up companies and checking that the person in question actually works for them etc. When this is done the adminis
initiating_event	The candidate enters details related to their profile and press submit.The administrator then receives the details of this profile for approval.
memory_context	Details of the candidates profile need to be accessed from the database
functional_tests	We need to check that the candidates profile is viewable once it has been approved by the administrator.Besides this the system doesnt have control over the physical real world verification of a candidates details.
quality_attributes	The time between submission and approval should be less than 24 hours.
observable_result	When the admin approves the profile it is viewable by any member of the site

storyname	Register as a Candidate
description	When candidates wish to use services such as the messaging system they are required to register as candidates. They are prompted for username and password and their generalities.
initiating_event	A Candidate wishing to use the messaging system clicks register
memory_context	Candidate details are stored in the database
functional_tests	The user has to be able to create a new account by filling out an online form.If they try to create an existing username the system will output an error message.The system will also check that all the required fields are full. If not it will return an error
quality_attributes	The registration process needs to take less than 3 minutes.
observable_result	The candidate is now registered as a candidate

storyname	Verify Client
description	After clients register their companies will have to be verified before they can use services such as the messaging system.When the administrator wishes to verify a company he will phone them and tick the appropriate check box in the verify clients screen
initiating_event	A new client registers successfully
memory_context	A client will now be tagged as verified
functional_tests	The admin will access the verify clients screen in the admin section.The admin will have the options of changing a clients status to either verified or unverified. He will also be able to eventually delete a client.Once a client's status has been decided
quality_attributes	
observable_result	That client will now be able to message candidates

<i>storyname</i>	Fully register as Client
<i>description</i>	The Client should edit their profile.
<i>initiating_event</i>	Client edits details when registering
<i>memory_context</i>	Details updated to database
<i>functional_tests</i>	See Edit Client details.
<i>quality_attributes</i>	See Edit Client details.
<i>observable_result</i>	Client is now fully registered

<i>storyname</i>	Edit Candidate Details
<i>description</i>	The candidate should go into their homepage click a button that says edit profile and be able to change any details they have already entered.
<i>initiating_event</i>	Candidate enters homepage clicks edit profile
<i>memory_context</i>	Details loaded/updated from database
<i>functional_tests</i>	Candidate details are updated in the database.
<i>quality_attributes</i>	The candidate should be able to the edit their profile in less than 5 seconds under normal circumstances.
<i>observable_result</i>	Candidate details are updated

storyname	Edit Client details
description	The client should go into their homepage click a button that says edit profile and be able to change any details they have already entered.
initiating_event	User logs into account
memory_context	User details loaded from database
functional_tests	The clients updated details should be committed to the database correctly.
quality_attributes	The users details should appear in Edit boxes for them to add or update. It should take no longer than 5 seconds to edit the details in the database under normal circumstances.
observable_result	Client details are updated

storyname	Add bill to clients total invoice
description	When a client requests candidates details or hires a candidate the total cost of services is calculated and updated to the database.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Hire a candidate
description	Client selects a candidate from their shortlist to fill a certain position.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Admin edit message
description	If a message contains any personal details an admin will remove these details and allow the message to pass through to the original recipient.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Pay final invoice for job
description	The Client pays the sum totals of the amount they owe for services.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Admin verify messages
description	Any messages sent between clients and candidate has to be checked by admin for any information that should not be there before it reaches it's destination.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Change Password
description	Users can change their passwords by clicking a button in their profile.
initiating_event	User needs to change password - enters email address
memory_context	Email address matched with username and password
functional_tests	The system will prompt the user for a valid e-mail address. It will then check this against the associated password/user/email data in the database
quality_attributes	A new password must be set within 5 seconds under normal conditions.
observable_result	User receives an e-mail with their login details

storyname	Send introductory job message to all shortlisted candidates
description	Sends a simple job description to all candidates in the clients shortlist for a particular position.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Check interview availability
description	The client can check when a candidate is free and offer them an interview.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Interview a candidate
description	The client will click a button that says I wish to interview this client This is when they will arrange the details
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Interview a candidate
description	The client will click a button that says I wish to interview this client This is when they will arrange the details
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Update logged in times
description	When a user logs into the system it should automatically change the last time they logged in and display it to the user on their home page
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Update statistics about a candidates profile
description	whenever a candidate profile is viewed or shortlisted for a job by a company it should update the statistics associated with the candidates cv
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	

storyname	Admin approves client
description	Admin approves client account.
initiating_event	
memory_context	
functional_tests	
quality_attributes	
observable_result	