
Use Cases

for

MoveInLo

Version 1.0 approved

Prepared by CodeCrafters:

Eugene Wee Jun Lin (Leader) [U2120698A]

Iain Roderick Tay Rong Yu [U221382K]

Joseph Teo Chee Hoe [U2223041J]

Jiang Yifei [U2201092F]

Kim Hyun Bin [U2020316D]

SCSE Lab SCSX Group 3

15 Nov 2023

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

Supply the name of the person who initially documented this use case.

1.3.2 Date Created

Enter the date on which the use case was initially documented.

1.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

2. Use Case Definition

2.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

2.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

2.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

2.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition.

Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

2.5. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

2.6. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

2.8. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course". Example: X.Y.AC.1.

2.9. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by "EX" to indicate "Exception". Example: X.Y.EX.1.

2.10. Includes

List any other use cases that are included ("called") by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

2.11. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

2.12. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

2.13. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Cases

Use Case ID:	A0001		
Use Case Name:	Registration as Customer		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	User
Description:	The user creates a customer account by filling in an email address, username, account type, contact number, age, password.
Preconditions:	User clicks on Register in the Home Page.
Postconditions:	User's customer account successfully created.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the "Sign up" button in the landing page. 2. The user fills in the required information: email address, username, contact number, age, password. 3. The user indicates the account type as "Customer". 4. The user presses the "Sign up" button. 5. App validates the information and sends it to the database. 6. App redirects to the home page UI. 7. Database sends a response and the app displays the message "Account successfully created."
Alternative Flows:	<ol style="list-style-type: none"> 4. App finds the information invalid. 5. Back to step 2.
Exceptions:	The user already has a customer account.
Includes:	-
Special Requirements:	App must be able to check information validity within 2 seconds
Assumptions:	-
Notes and Issues:	<p>The user must input a valid email address that has '@' and '.com'.</p> <p>The user must input a valid password that has at least 8 characters, 1 special character, 1 capital letter, 1 numerical value.</p>

Use Case ID:	A0002		
Use Case Name:	Registration as Job Seeker		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	User
Description:	The user creates a job seeker account by filling in information for email address and password.
Preconditions:	-
Postconditions:	User's job seeker account successfully created.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the "Sign Up" button. 2. The user fills in the required information: email address, username, contact number, age, password. 3. The user indicates the account type as "Job Seeker". 4. The user presses the "Sign up" button. 5. App validates the information and sends it to the database. 6. App redirects to the home page UI. 7. Database sends a response and the app displays the message "Account successfully created."
Alternative Flows:	<ol style="list-style-type: none"> 4. App finds the information invalid. 5. Back to step 2.
Exceptions:	The user already has a job seeker account.
Includes:	-
Special Requirements:	App must be able to check information validity within 2 seconds
Assumptions:	-
Notes and Issues:	<p>The user must input a valid email address that has '@' and '.com'.</p> <p>The user must input a valid password that has at least 8 characters, 1 special character, 1 capital letter, 1 numerical value.</p>

Use Case ID:	A0003		
Use Case Name:	Customer Login		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer logs into the app using their registered email and password.
Preconditions:	Valid email and password.
Postconditions:	The customer's account is successfully logged in.
Priority:	-
Frequency of Use:	By the intention of the customer.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer clicks the "Login" button on the Landing Page. 2. The customer fills in their registered email and password. 3. The customer selects "Customer" for account type. 4. The customer presses the "Login" button. 5. App authenticates the email and password. 6. The customer logs on to their account's home page.
Alternative Flows:	<ol style="list-style-type: none"> 7. App finds credentials are invalid. 8. App displays the error message. 9. Back to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	App must be able to check information validity within 2 seconds
Assumptions:	-
Notes and Issues:	-

Use Case ID:	A0004		
Use Case Name:	Job Seeker Login		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	The job seeker logs into the app using their registered email and password.
Preconditions:	Valid email and password.
Postconditions:	The job seeker's account is successfully logged in.
Priority:	-
Frequency of Use:	By the intention of the job seeker.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer clicks the "Login" button on the Landing Page. 2. The customer fills in their registered email and password. 3. The customer selects "Job Seeker" for account type. 4. The customer presses the "Login" button. 5. App authenticates the email and password. 6. The customer logs on to their account's home page.
Alternative Flows:	<ol style="list-style-type: none"> 7. App finds credentials are invalid. 8. App displays the error message. 9. Back to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	App must be able to check information validity within 2 seconds
Assumptions:	-
Notes and Issues:	-

Use Case ID:	A0005		
Use Case Name:	Forget Password		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	User
Description:	On the login page, if the user does not remember their password, they can reset it through email.
Preconditions:	The user has an account.
Postconditions:	The user's current password is changed to the new password.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the "Forget Password" button. 2. App verifies email and the user presses on "Change Password" button. 3. The user is prompted to enter a new password. 4. The user enters a new password. 5. App checks if the new password is valid. 6. App replaces the old password with the new one in the database. 7. Password is reset.
Alternative Flows:	<ol style="list-style-type: none"> 7. App checks that the password is invalid. 8. Back to step 5.
Exceptions:	<ol style="list-style-type: none"> 1. The user's email is not linked to any valid account. 2. The user does not have access to his/her own email.
Includes:	-
Special Requirements:	-
Assumptions:	The email belongs to the user's account.
Notes and Issues:	-

Use Case ID:	A0006		
Use Case Name:	Successful Login		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	12th Sep 2023	Date Last Updated:	12 Sep 2023

Actor:	User
Description:	The application must allow the authenticated user to access their designated home page
Preconditions:	The user has been authenticated during the login.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user is authenticated during successful login or registration. 2. If the user logs in as “Customer”, the user is redirected to the Customer Home Page.
Alternative Flows:	<ol style="list-style-type: none"> 2. If the user logs in as “Job Seeker”, the user is redirected to the Job Seeker Home Page.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	The user account has successfully been authenticated.
Notes and Issues:	-

Use Case ID:	M0007		
Use Case Name:	Selection of moving services		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to select two options of moving services.
Preconditions:	The customer has logged in.
Postconditions:	The customer has selected what moving service he/she wants.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer presses the “Schedule” button. 2. The customer presses the “Moving In” if he/she wants to move in and the “Moving Out” if he/she wants to move out.
Alternative Flows:	<ol style="list-style-type: none"> 3. The customer presses the “Back” button as he/she changes his/her mind.
Exceptions:	-
Includes:	-
Special Requirements:	App is able to show the moving services within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	M0008		
Use Case Name:	Schedule move in services		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to schedule a service to move in.
Preconditions:	The customer has selected a move in service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the customer.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer presses the “Move In” button. 2. The customer is able to select the date and timing of the moving services.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to schedule the move in service within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	M0009		
Use Case Name:	Schedule move out services		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to schedule a service to move out.
Preconditions:	The customer has selected a move out service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the customer.
Flow of Events:	<ol style="list-style-type: none"> 3. The customer presses the “Move Out” button. 4. The customer is able to select the date and timing of the moving services.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to schedule the move out service within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	M0010		
Use Case Name:	Providing location for delivery		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to provide a location for delivery
Preconditions:	The customer has selected a move in/move out service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the customer.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer presses the “Provide Location” button. 2. The customer is able to enter the address of the delivery with a postal code.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to schedule the location within 2 seconds.
Assumptions:	-
Notes and Issues:	App must request the customer to provide a valid 6-digit postal code, SXXXXX within Singapore for the item delivery.

Use Case ID:	M0011		
Use Case Name:	Providing date for delivery		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to provide a date for delivery
Preconditions:	The customer has selected a move in/move out service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the customer.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer presses the “Provide Date” button. 2. The customer is able to enter the date of the delivery.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to schedule the date within 2 seconds.
Assumptions:	-
Notes and Issues:	App must request the customer to provide a date in DD/MM/YYYY format.

Use Case ID:	M0012		
Use Case Name:	Providing time for delivery		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to provide a time for delivery
Preconditions:	The customer has selected a move in/move out service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the customer.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer presses the “Provide Time” button. 2. The customer is able to enter the time of the delivery.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to schedule the time within 2 seconds.
Assumptions:	-
Notes and Issues:	App must request the customer to provide a date in HH:MM 24-hour format.

Use Case ID:	M0013		
Use Case Name:	Display progress tracker		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to track the progress of the delivery.
Preconditions:	The customer has selected a move in/move out service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the customer.
Flow of Events:	<ol style="list-style-type: none"> 1. The customer presses the “Track delivery” button. 2. The customer is able to see the progress of the delivery.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to update the tracker within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	M0014		
Use Case Name:	Update Progress Tracker for Collected		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	The job seeker is able to update the tracker to collected.
Preconditions:	The job seeker has scheduled a move in/move out service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the job seeker..
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker collects the product from the location. 2. The job seeker updates the progress tracker to “Collected”.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to update the tracker within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	M0015		
Use Case Name:	Update Progress Tracker for Delivering		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	The job seeker is able to update the tracker to delivering.
Preconditions:	The progress tracker of the product is collected.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the job seeker..
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker collects the product from the location. 2. The job seeker updates the progress tracker to “In Progress”.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to update the tracker within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	M0016		
Use Case Name:	Update Progress Tracker for Delivered		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	The job seeker is able to update the tracker to delivered.
Preconditions:	The progress tracker of the product is in progress.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the job seeker..
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker delivers the product from the moving service. 2. The job seeker updates the progress tracker to “Delivered”.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to update the tracker within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	M0017		
Use Case Name:	Cancellation of Moving Service		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Customer
Description:	The customer is able to cancel the scheduled service.
Preconditions:	The customer has a scheduled service.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the customer..
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker delivers the product from the moving service. 2. The job seeker updates the progress tracker to “Delivered”.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App is able to update the tracker within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	J0018		
Use Case Name:	Display of job seeker homepage		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job seeker
Description:	When the job seeker successfully logs in, the job seeker homepage is displayed.
Preconditions:	The job seeker manages to successfully log in.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the job seeker.
Flow of Events:	Job seeker homepage is shown.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App must be able to check information validity within 2 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	J0019		
Use Case Name:	Available Job Listings		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job seeker
Description:	The job seeker is able to view the available list of products.
Preconditions:	The job seeker has logged in.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker presses the “View products” button. 2. The job seeker is able to view the list of products available.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App must be able to display listing of products within 2 seconds
Assumptions:	-
Notes and Issues:	-

Use Case ID:	J0020		
Use Case Name:	Display of moving service job information		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	The job seeker is able to view the information of the delivery.
Preconditions:	The job seeker has registered for a moving service job.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker presses the “View Information” button. 2. The app displays the date, time and location of the delivery.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App must be able to display information within 2 seconds
Assumptions:	-
Notes and Issues:	-

Use Case ID:	J0021		
Use Case Name:	Display of group buying job information		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	The job seeker is able to view the information of the delivery.
Preconditions:	The job seeker has registered for a group buying job.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker presses the “View Information” button. 2. The app displays the date, time and location of the delivery.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	App must be able to display information within 2 seconds
Assumptions:	-
Notes and Issues:	-

Use Case ID:	J0022		
Use Case Name:	Job Registration		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	Job seekers caan register for available jobs.
Preconditions:	Job seekers have logged in.
Postconditions:	
Priority:	
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. Job seeker clicks register for an available job listed in Job Listings page. 2. The job is registered successfully.
Alternative Flows:	If Job Seeker has already registered for 5 jobs, show “You have registered the maximum number of jobs!” Then refuse the Job Seeker
Exceptions:	
Includes:	
Special Requirements:	Processes of job registration should have a response time of less than 3 seconds.
Assumptions:	
Notes and Issues:	

Use Case ID:	J0023		
Use Case Name:	Registered Jobs Listings		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	Job seeker can view their jobs on their Registered Jobs page
Preconditions:	Job seekers have logged in.
Postconditions:	
Priority:	
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. Job seeker go to their Registered Jobs page 2. Job seeker can see a list of jobs registered by the current job seeker
Alternative Flows:	If the current job seeker has registered no jobs, then show “No job registered.”
Exceptions:	
Includes:	
Special Requirements:	Displaying job listings should take less than 2 seconds.
Assumptions:	
Notes and Issues:	

Use Case ID:	J0024		
Use Case Name:	Completion of job		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	Job seeker can indicate he/she has completed the job.
Preconditions:	Job seekers have a registered job.
Postconditions:	
Priority:	
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. Job seeker has completed the job. 2. Job seeker can change the job status to “Completed”.
Alternative Flows:	If the current job seeker has registered no jobs, then show “No job registered.”
Exceptions:	
Includes:	
Special Requirements:	Displaying job listings should take less than 2 seconds.
Assumptions:	
Notes and Issues:	

Use Case ID:	J0025		
Use Case Name:	Job Payment Completed		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job seekers
Description:	Job seeker will know when the payment is completed.
Preconditions:	<ol style="list-style-type: none"> 1. Job seeker has logged in. 2. Job seeker has completed a registered job. 3. Payment has been paid out by customers.
Postconditions:	
Priority:	
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The job seeker goes to Registered Jobs page. 2. The display of the completed job shows “Payment completed”
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	Displaying job listings should take less than 2 seconds.
Assumptions:	
Notes and Issues:	

Use Case ID:	J0026		
Use Case Name:	Job Withdrawal		
Created By:	CodeCrafters	Last Updated By:	CodeCrafters
Date Created:	10th Sep 2023	Date Last Updated:	11th Sep 2023

Actor:	Job Seeker
Description:	Job Seeker can withdraw a job that the current job seeker successfully registered.
Preconditions:	<ol style="list-style-type: none"> 1. Job seekers have logged in. 2. Job seeker has successfully registered for a job.
Postconditions:	
Priority:	
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 3. Job seeker clicks withdrawal for a certain job in the page of job registered.. 4. Job seeker click “Yes” on the confirmation page. 5. The job is withdrawn successfully.
Alternative Flows:	If the job seeker clicks “No” on the confirmation page, then the job will not be withdrawn, and the application gets back to the job registered page.
Exceptions:	
Includes:	
Special Requirements:	Processes of job withdrawal should have a response time of less than 3 seconds.
Assumptions:	
Notes and Issues:	