
Use Cases

for

Agora

Version 1.0 approved

Prepared by Horstann

SC2006 – Lab Group 6

7/9/2022

Revision History

Name	Date	Reason For Changes	Version
Horstann Ho	6/9/2022	Start with Use Case Description template	0.0
Horstann Ho	7/9/2022	Complete remaining initial Use Case Descriptions	1.0
Garrick Goh	24/10/2022	Update and add new Use Case Descriptions	2.0

Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

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. Includess

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 Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case Descriptions

Use Case ID:	ACC.01		
Use Case Name:	Sign Up		
Created By:	Horstann Ho	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	

Actor:	Buyer, Seller
Description:	Creates a new account for a user.
Preconditions:	<ol style="list-style-type: none"> 1. The app boots up. and <ol style="list-style-type: none"> 2. The user clicks “Sign up here!”
Postconditions:	<ol style="list-style-type: none"> 1. The user clicks “Continue”. and <ol style="list-style-type: none"> 2. The user’s details are considered valid by the system.
Priority:	
Frequency of Use:	0-1 times per login
Flow of Events:	<ol style="list-style-type: none"> 1. The user enters a username, an email address, a password twice, a location address and a postal code. The user then selects to be either a buyer or seller. 2. The user clicks “Continue”. 3. The system verifies the user’s inputs based on the criteria below: <ol style="list-style-type: none"> a. The username and email address do not already belong to a registered account b. The password has a minimum length of 8 characters containing at least a capital letter and a number c. The 2 passwords match case-sensitively d. The location address and postal code exists 4. If the user’s input are valid, the system creates a new buyer or seller account based on the user’s selection. 5. If the user has selected to be a seller, they will be given the option to provide descriptions of their business. 6. If the user has selected to be a buyer, they will be given the option to state their preference to be a distributor. If they do so, the buyer will have a distributor badge that all other buyers can see.
Alternative Flows:	<p>ACC.01.AC.01: If the username already belongs to a registered account</p> <ol style="list-style-type: none"> 1. The system displays the message “Username belongs to a registered account!” 2. The system returns to step 3 and waits for user to click on “Continue”.

	<p>ACC.01.AC.02: If the email address already belongs to a registered account</p> <ol style="list-style-type: none"> 1. The system displays the message “Email address belongs to a registered account!” 2. The system returns to step 3 and waits for user to click on “Continue”. <p>ACC.01.AC.03: If the password doesn’t have a minimum length of 8 characters containing at least a capital letter and a number</p> <ol style="list-style-type: none"> 1. The system displays the message “Password must have at least 8 characters, with at least 1 capital letter and 1 number!” 2. The system returns to step 3 and waits for user to click on “Continue”. <p>ACC.01.AC.04: If the 2 password don’t match case-sensitively</p> <ol style="list-style-type: none"> 1. The system displays the message “Passwords (case-sensitive) do not match!” 2. The system returns to step 3 and waits for user to click on “Continue”. <p>ACC.01.AC.05: If the location address or postal code cannot be found by the Google Maps API</p> <ol style="list-style-type: none"> 1. The system displays the message “Please re-enter a valid location address or postal!” 2. The system returns to step 3 and waits for user to click on “Continue”.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	ACC.02		
Use Case Name:	Sign In		
Created By:	Horstann Ho	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	

Actor:	Buyer, Seller
Description:	Signs user into an existing account.
Preconditions:	1. The app boots up.
Postconditions:	1. The user clicks “Sign In”. and 2. The user’s details are considered valid by the system.
Priority:	
Frequency of Use:	1 time per login
Flow of Events:	<ol style="list-style-type: none"> 1. The user enters an email address and a password. 2. The user clicks “Sign In”. 3. The system verifies the user’s inputs based on the criteria below: <ol style="list-style-type: none"> a. The email address belongs to a registered account b. The password matches case-sensitively to that of the registered account’s. 4. If the user’s input are valid, the system signs the user into their registered account.
Alternative Flows:	ACC.02.AC.01: If the email address or password are incorrect <ol style="list-style-type: none"> 1. The system displays the message “Email address or password is incorrect!” 2. The system returns to step 3 and waits for user to click on “Sign In”.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	ACC.03		
Use Case Name:	Change Account Settings		
Created By:	Horstann Ho	Last Updated By:	
Date Created:	7/9/2022	Date Last Updated:	

Actor:	Buyer, Seller, Google Login API, Google Maps API
Description:	Allows users to change their account details like location address and password.
Preconditions:	1. The user selects "Account".
Postconditions:	1. The user's new details are considered valid by the system. or 2. The user leaves the app
Priority:	
Frequency of Use:	0-1 time per login
Flow of Events:	1. The system lets the user to select between multiple options as below a. Change Location Address b. Change Password c. Forgot Password
Alternative Flows:	ACC.03.AC.01: If the username or password are incorrect 1. The system displays the message "Username or password is incorrect!" 2. The system returns to step 3 and waits for user to click on "Sign In".
Exceptions:	ACC.03.EX.01: If the password doesn't have a minimum length of 8 characters containing at least a capital letter and a number 1. The system displays the message "Password must have at least 8 characters, with at least 1 capital letter and 1 number!" 2. The system returns to step 3 and waits for user to click on "Continue". ACC.03.EX.02: If the 2 password don't match case-sensitively 1. The system displays the message "Passwords (case-sensitive) do not match!" 2. The system returns to step 3 and waits for user to click on "Continue". ACC.03.EX.03: If the location address or postal code cannot be found by the Google Maps API 1. The system displays the message "Please re-enter a valid location address or postal!" 2. The system returns to step 3 and waits for user to click on "Continue".
Includes:	Change Location Address 1. The system prompts the user to sign in again 2. Once successfully signed in, the system lets the user enter a new location address and postal code 3. The system verifies whether the new address and postal code exists

	<ol style="list-style-type: none"> 4. If the user's input are valid, the system updates the user account's location 5. If the user is in any group chats, the user's location pins will change in the corresponding group maps <p>Change Password</p> <ol style="list-style-type: none"> 1. The system prompts the user to sign in again 2. Once successfully signed in, the system prompts the user enter a new password 3. The system prompts the user to enter the same password again for confirmation 4. If the 2 passwords match case-sensitively, the system updates the user account's password <p>Forgot Password</p> <ol style="list-style-type: none"> 1. The system sends an email to the user's specified email address, with a private link to change their password. 2. The user logs into their email account, and clicks into the link 3. The system prompts the user to enter a new password 4. The system prompts the user to enter the same password again for confirmation 5. If the 2 passwords match case-sensitively, the system updates the user account's password
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	PROD.01		
Use Case Name:	Browse Products		
Created By:	Horstann Ho	Last Updated By:	
Date Created:	7/9/2022	Date Last Updated:	

Actor:	Buyer
Description:	Displays available products to users based on location.
Preconditions:	1. The user signs in or signs up successfully as a buyer.
Postconditions:	1. The user selects to browse group chats or 2. The user leaves the app
Priority:	
Frequency of Use:	1-3 times per login
Flow of Events:	<ol style="list-style-type: none"> The system displays a list of available products closest to the user's specified address location. Each product displays the below information. <ol style="list-style-type: none"> Product name Product image Product price thresholds Number of people in queue Total available units The system lets the user scroll through the list. The system lets the user expand a particular product to view more information about it as below <ol style="list-style-type: none"> Queue duration Product image Product price thresholds Number of people in queue Total available units Product name Product description
Alternative Flows:	ACC.03.AC.01: If the username or password are incorrect <ol style="list-style-type: none"> The system displays the message "Username or password is incorrect!" The system returns to step 3 and waits for user to click on "Sign In".
Exceptions:	EX.CONN: If the app fails to connect to the server <ol style="list-style-type: none"> The system displays the message "Unable to connect!"
Includes:	Buy Product <ol style="list-style-type: none"> The user expands a particular product If the product's queue is open and the number buys in queue is below the product's total available units, the system lets the user buy the product The user selects a quantity to buy The user clicks on "Buy Now!" If the selected quantity doesn't exceed the product's available unit, the system adds the user into the product's queue and group chat Drop Product

	<ol style="list-style-type: none"> 1. The user expands a particular product 2. If the user has previously entered the product queue and the queue is still open, allow the user to revert their actions 3. The user clicks on “Drop Product” 4. The system removes the user from the product queue 5. The user remains in the product group chat, unless the user opts to leave <p>Like Product</p> <ol style="list-style-type: none"> 1. The system lets the user save a product into a personal list of “Liked Products” 2. The user clicks on “Liked Products” 3. The system shows the user’s personal list of Liked Products 4. The system lets the user remove any items from Liked Products <p>Search Products</p> <ol style="list-style-type: none"> 1. The user selects the search bar 2. The user types a product name to be searched 3. The system returns a list of products, ranked based on the closest match and geographical distance of the user to the product’s seller <p>Enter Group Chat for Product</p> <ol style="list-style-type: none"> 1. If the user has entered the product’s queue, the system lets the user to enter that product’s group chat
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	PROD.02		
Use Case Name:	Manage Products		
Created By:	Horstann Ho	Last Updated By:	
Date Created:	7/9/2022	Date Last Updated:	

Actor:	Seller
Description:	Displays all of the seller's products that are currently on sale.
Preconditions:	1. The user signs in or signs up successfully as a seller.
Postconditions:	1. The user selects to browse group chats or 2. The user leaves the app
Priority:	
Frequency of Use:	1-5 times per login
Flow of Events:	<ol style="list-style-type: none"> The system displays a list of the user's own products that are currently on sale. Each product displays the below information. <ol style="list-style-type: none"> Product name Product image Product price thresholds Number of people in queue Total available units The system lets the user scroll through the list. The system lets the user expand a particular product to view more information about it as below <ol style="list-style-type: none"> Queue duration Product image Product price thresholds Number of people in queue Total available units Product name Product description
Alternative Flows:	
Exceptions:	EX.CONN: If the app fails to connect to the server 2. The system displays the message "Unable to connect!"
Includes:	<p>Add Product</p> <ol style="list-style-type: none"> The user selects to add a new product The user fills in the details below regarding the product <ol style="list-style-type: none"> Queue duration Product image Product price thresholds Total available units Product name Product description The user clicks on "Add Product" If all details are valid, the system adds the product, opens a queue for the product and makes it publicly viewable to all users <p>Edit Product</p> <ol style="list-style-type: none"> The user expands a particular product

	<ol style="list-style-type: none"> 2. The system lets the user modify any of the product's details below anytime before the product's queue closes <ol style="list-style-type: none"> a. Queue duration b. Total available units 3. The system also lets the user remove the product completely anytime before the product's queue closes. If this is done, the product's queue is closed and its group chat removed. <p>Search Products</p> <ol style="list-style-type: none"> 1. The user selects the search bar 2. The user types a product name to be searched 3. The system returns a list of products, ranked based on the closest match and geographical distance of the user to the product's seller 4. The system lets the user filter between his own products and other sellers' products <p>Enter Group Chat for Product</p> <ol style="list-style-type: none"> 1. If the user owns the product, the system lets the user to enter that product's group chat
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	CHAT.01		
Use Case Name:	Browse Group Chats		
Created By:	Horstann Ho	Last Updated By:	
Date Created:	7/9/2022	Date Last Updated:	

Actor:	Buyer, Seller
Description:	Displays all group chats that the user is currently in.
Preconditions:	1. The user selects to browse group chats.
Postconditions:	1. The user selects to browse/manage products or 2. The user leaves the app
Priority:	
Frequency of Use:	1-3 times per login
Flow of Events:	1. The system displays a list of group chats the user is in, ranked based on those with latest messages. 2. The system lets the user scroll through the list.
Alternative Flows:	
Exceptions:	EX.CONN: If the app fails to connect to the server 1. The system displays the message “Unable to connect!”
Includes:	Enter Group Chat for Product 1. The system lets the user to enter a selected group chat.
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	CHAT.02		
Use Case Name:	Enter Group Chat for Product		
Created By:	Horstann Ho	Last Updated By:	
Date Created:	7/9/2022	Date Last Updated:	

Actor:	Buyer, Seller, Google Maps API
Description:	Displays all messages between the seller and buyer of a particular product, and also allows users to view a map of the locations of these people.
Preconditions:	1. The user selects to enter a particular group chat.
Postconditions:	1. The user selects to exit the group chat or 2. The user leaves the app
Priority:	
Frequency of Use:	1-4 times per login
Flow of Events:	<ol style="list-style-type: none"> 1. The system shows all the messages in the group chat from newest to oldest, and their corresponding senders' names. 2. The system lets the user to view all of the group chat's participants, distinguishing the seller and buyers who opt to be distributors (possibly via badges). 3. The system lets the user to send either text messages or images. 4. If the user is a buyer who has dropped the product, the system lets the user leave the group chat.
Alternative Flows:	CHAT.02.AC.01: Once the queue of the group chat's product closes <ol style="list-style-type: none"> 1. The system sends a message to the group chat that says "The duration of the product queue is over. The queue is now closed!" 2. The system lets the users use the group chat as usual.
Exceptions:	EX.CONN: If the app fails to connect to the server <ol style="list-style-type: none"> 1. The system displays the message "Unable to connect!"
Includes:	View Map of Buyers and Sellers <ol style="list-style-type: none"> 1. If a user is in a product's group chat, the system lets the user view a map of all the product's buyers and sellers 2. The system differentiates pins of <ol style="list-style-type: none"> a. Sellers b. Buyers (normal buyers who aren't distributors) c. Distributors (buyers who have opted to be distributors) d. Non-buyers (users who were previously buyers of the product, but have since dropped the product before its queue closes) 3. The system lets the user click on each pin to view the full address location of each pin, but doesn't show the user's name in that address (for privacy's sake).
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	VIEWLIKED.01		
Use Case Name:	View Liked Contacts		
Created By:	Garrick Goh	Last Updated By:	
Date Created:	24/10/2022	Date Last Updated:	

Actor:	Buyer
Description:	Displays the user liked contacts
Preconditions:	1. The user chooses to view his list of liked contacts or add liked contacts
Postconditions:	1. The user selects to view like contacts or 2. The user adds new liked contacts or 3. The user leaves the list
Priority:	
Frequency of Use:	1-5 times per login
Flow of Events:	1. The system displays a list of contacts that are liked 2. The system let the user scroll through the list
Alternative Flows:	LIKE.01.AC.01: User does not have like contacts 1. If the user does not have liked contacts, 'There is no liked contacts' will be displayed 2. The system lets the users use the list as usual
Exceptions:	EX.CONN: If the app fails to connect to the server 1. The system displays the message "Unable to connect!"
Includes:	Liking Contact 1. The system lets the user save a contact into a personal list of "Liked Contacts" 2. The user clicks on "Liked Contacts" 3. The system shows the user's personal list of Liked Contacts 4. The system lets the user have the option to remove contacts from Liked Contacts
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	REQUEST.01		
Use Case Name:	Send Buy Request		
Created By:	Garrick Goh	Last Updated By:	
Date Created:	24/10/2022	Date Last Updated:	

Actor:	Buyer
Description:	Allows the user to send buy request to other users
Preconditions:	1. The user selects to send buy requests
Postconditions:	1. The user sends the buy request and chooses to leave the app or 2. The user leaves the app
Priority:	
Frequency of Use:	1-5 times per login
Flow of Events:	1. The system allows the user to select the group of people to send buy requests to 2. The system then proceeds to send out the buy requests
Alternative Flows:	REQUEST.01.AC.01: User does not have like contacts 1. If the user does not have liked contacts, it will prompt to ask the user if he wants to send to nearby users 2. The system lets the users use the app as usual
Exceptions:	EX.CONN: If the app fails to connect to the server 1. The system displays the message “Unable to connect!”
Includes:	Liked Contact 1. The user selects to send buy requests to liked contacts 2. The user chooses from the liked contacts who he wants to send to 3. The system will send a message “Buy request sent to liked contacts” telling the user that the buy request has been successfully sent Get Nearby Users 1. The user selects to send buy requests to users nearby 2. The system sends out to users within a 1km radius 3. The system will send a message “Buy request sent to nearby users” telling the user that the buy request has been successfully sent
Special Requirements:	
Assumptions:	
Notes and Issues:	

Notes and Issues:	