Use Cases

Use case	Compose an order
Scope	EPS
Level	User-goal
Intention in context	The customer wants to compose an order
Primary actor	Customer
Support actor	-
Stakeholders' interest	Farmer wants to sell its products
Precondition	There are products available The order is entered within Sunday, 11pm
Minimum guarantees	-
Success guarantees	The order is saved
Trigger	-
Main success scenario	1. The customer enters the system 2. The system shows the list of products 3. The customer selects the product 4. The system checks the availability 5. The customer inserts the desired quantity 6. The system checks the minimum quantity constraints *Steps 5 and 6 can be repeated* 7. The customer confirms the order 8. The system saves the order and changes the status to entered
Extensions	2.a there aren't products in the list: use case terminates with an error 3.a the customer exits: use case terminates with an error 3.b the customer goes back: use case returns to 1 4.a the products aren't available: use case terminates with an error 5.a the customer exits: use case terminates with an error 5.b the customer goes back: use case returns to 3 6.a quantity constraints aren't specified: use case terminates with an error 7.a the customer exits: use case terminates with an error 7.b the customer goes back: use case returns to 5 8.a the system doesn't save: use case goes back to step 7

Use case	Manage modify
Scope	EPS
Level	User-goal
Intention in context	The customer wants to manage the modifies
Primary actor	Customer
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	There are products available The modifies are accepted within Monday evening
Minimum guarantees	**-**
Success guarantees	The order is modified and correctly collected
Trigger	The quantities requested aren’t available
Main success scenario	1. The system shows the modified order 2. The customer visualise the modifies 3. The system asks for confirmation to the customer 4. The customer accepts the modify 5. The system saves Use case terminates with success
Extensions	4.a the customer denies modify: the system changes the order status to cancelled and notify the customer 4.b the customer goes back: use case returns to step 2 4.c the customer cancels: use case terminates with an error 5.a the system cannot save: use case returns to step 4

Scope	EPS
Level	User-goal
Intention in context	The customer wants to fix an appointment for the pick up of the products
Primary actor	Customer
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	The customer has made an order
Minimum guarantees	-
Success guarantees	The appointment is scheduled
Trigger	-
Main success scenario	1. The customer wants to book an appointment 2. The system shows the available days and times 3. The customer chooses a day and a time 4. The system saves the choices Use case terminates with success
Extensions	2.a there aren't available appointments: use case terminates with an error 3.a the customer exits: use case terminates with an error 3.b the customer goes back: use case returns to 1 4.a the system doesn't save: use case goes back to step 3

Use case	Compose an order
Scope	EPS
Level	User-goal
Intention in context	The employee wants to compose an order
Primary actor	Employee
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	There are products available The order is entered within Sunday, 11pm
Minimum guarantees	-
Success guarantees	The order is saved
Trigger	-
Main success scenario	1. The employee enters the system 2. The system shows the list of products 3. The employee selects the product 4. The system checks the availability 5. The employee inserts the desired quantity 6. The system checks the minimum quantity constraints 7. The employee confirms the order 8. The system saves the order and changes the status to entered
Extensions	2.a there aren’t products in the list: use case terminates with an error 3.a the employee exits: use case terminates with an error 3.b the employee goes back: use case returns to 1 4.a the products aren’t available: use case terminates with an error 5.a the employee exits: use case terminates with an error 5.b the employee goes back: use case returns to 3 6.a quantity constraints aren’t met: use case terminates with an error 7.a the employee exits: use case terminates with an error 7.b the employee goes back: use case returns to 5 8.a the system doesn’t save: use case goes back to step 7

Use case	Recharge virtual wallet
Scope	EPS
Level	User-goal
Intention in context	The employee wants to charge the customer’s virtual wallet
Primary actor	Employee
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	
Minimum guarantees	-
Success guarantees	The wallet is charged
Trigger	-
Main success scenario	1. The employee enters the system 2. The system asks the amount to charge 3. The employee inserts the amount 4. The system saves the new amount of money on the virtual wallet Use case terminates with success
Extensions	3.a the employee exits: use case terminates with an error 3.b the employee goes back: use case returns to 1 4.a the system doesn’t save: use case goes back to step 4

Use case	View pick up schedule
Scope	EPS
Level	User-goal
Intention in context	The employee wants to view the pick up schedule
Primary actor	Employee
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	There are booked appointments
Minimum guarantees	-
Success guarantees	The schedule Is showed
Trigger	-
Main success scenario	1. The employee enters the system 2. The system shows the pick up schedule Use case terminates with success
Extensions	2.a the pick up list is empty: use case terminates with an error

Use case	Confirm order ready to pick up
Scope	EPS
Level	User-goal
Intention in context	The employee confirms that the order is ready to be picked up by the customer
Primary actor	Employee
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	There are orders and products available
Minimum guarantees	**-**
Success guarantees	The order is ready
Trigger	-
Main success scenario	1. The employee selects an order 2. The system shows the order details 3. The employee prepares the order and confirms the readiness 4. The system saves Use case terminates with success
Extensions	2.a the system cannot show the order details: use case terminates with an error 3.a the employee goes back: use case returns to step 1 3.b the employee cancels: use case terminates with an error 4.a the system cannot save: use case returns to step 3

Use case	Manage modifies
Scope	EPS
Level	User-goal
Intention in context	The employee confirms the modifies on behalf of the customer
Primary actor	Employee
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	There are orders and products available
Minimum guarantees	**-**
Success guarantees	The modify is accepted
Trigger	-
Main success scenario	1. The employee selects an order 2. The system shows the order details and ask if he wants to accept the modifies 3. The employee confirms 4. The system saves Use case terminates with success
Extensions	2.a the system cannot show the order details: use case terminates with an error 3.a the employee goes back: use case returns to step 1 3.b the employee cancels: use case terminates with an error 4.a the system cannot save: use case returns to step 3

Use case	Register pick up
Scope	EPS
Level	User-goal
Intention in context	The employee wants to register the pick up
Primary actor	Employee
Support actor	
Stakeholders' interest	Farmer wants to sell its product
Precondition	The pick up is registered within Friday evening
Minimum guarantees	-
Success guarantees	The pickup is registered
Trigger	-
Main success scenario	1. The employee enters the system 2. The system shows the pick up schedule 3. The employee selects the appointment 4. The system asks if the pick up occured 5. The employee confirms 6. The system saves and changes the order status to completed Use case terminates with success
Extensions	2.a the pick up schedule is empty: use case terminates with an error 3.a the employee exits: use case terminates with an error 3.b the employee goes back: use case returns to 1 5.a the employee exits: use case terminates with an error 5.b the employee goes back: use case returns to 1 5.c the employee doesn’t confirms: the use case terminates 4.a the system doesn’t save: use case goes back to step 3

Use case	Register delivery
Scope	EPS
Level	User-goal
Intention in context	The employee wants to register the receipt of the products
Primary actor	Employee
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	The delivery is confirmed within Tuesday, 8 pm
Minimum guarantees	-
Success guarantees	The products are delivered
Trigger	-
Main success scenario	1. The employee enters the system 2. The system shows the list of deliveries 3. The employee selects one delivery to confirm the receipt of the products 4. The system asks for confirmation 5. The employee confirms 6. The system saves Use case terminates with success
Extensions	2.a the list of deliveries is empty: use case terminates with an error 3.a the employee cancels: use case terminates with an error 3.b the employee goes back: use case returns to step 1 5.a the employee doesn’t confirm: use case terminates with an error 6.a the system doesn’t save: use case goes back to step 5

Use case	Insert products
Scope	EPS
Level	User-goal
Intention in context	The farmer wants to insert the products
Primary actor	Farmer
Support actor	Payment system
Stakeholders' interest	Farmer wants to sell its products
Precondition	The products are uploaded within Saturday, 9 am
Minimum guarantees	-
Success guarantees	The products are uploaded
Trigger	-
Main success scenario	1. The farmer asks to register new products 2. The system asks to insert information about the products 3. The farmer inserts the information and confirms 4. The system saves the new amount of money on the virtual wallet Use case terminates with success
Extensions	3.a the farmer exits: use case terminates with an error 3.b the farmer goes back: use case returns to 1 4.a the system cannot save: use case goes back to step 3

Use case	Confirm availability
Scope	EPS
Level	User-goal
Intention in context	The farmer wants to confirm the availability
Primary actor	Farmer
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	The availability is confirmed within Monday, 9am
Minimum guarantees	-
Success guarantees	The quantity is confirmed
Trigger	-
Main success scenario	1. The farmer asks to confirm the products quantities 2. The system shows the previous amount and ask if he wants to confirm or change the quantities 3. The farmer confirms the amount 4. The system saves Use case terminates with success
Extensions	3.a the farmer exits: use case terminates with an error 3.b the farmer goes back: use case returns to 1 4.a the system cannot save: use case goes back to step 3

Use case	Modify availability
Scope	EPS
Level	User-goal
Intention in context	The farmer wants to confirm the availability
Primary actor	Farmer
Support actor	
Stakeholders' interest	Farmer wants to sell its products
Precondition	The availability is modified within Monday, 9am
Minimum guarantees	-
Success guarantees	The quantity is confirmed
Trigger	-
Main success scenario	1. The farmer asks to confirm the products quantities 2. The system shows the previous amount and ask if he wants to confirm or change the quantities 3. The farmer selects to modify the amount 4. The system asks to insert the new amount 5. The farmer inserts it and cofirms 6. The system checks that the new amount is lower than the sum of the ordered quantity and saves *Steps 5 and 6 can be repeated* Use case terminates with success
Extensions	3.a the farmer exits: use case terminates with an error 3.b the farmer goes back: use case returns to 1 5.a the new amount is higher: use case returns to step 3 5.b the farmer exits: use case terminates with an error 5.c the farmer goes back: use case returns to 3 6.a the system cannot save: use case goes back to step 3