

Honeybee Haven



Phase I

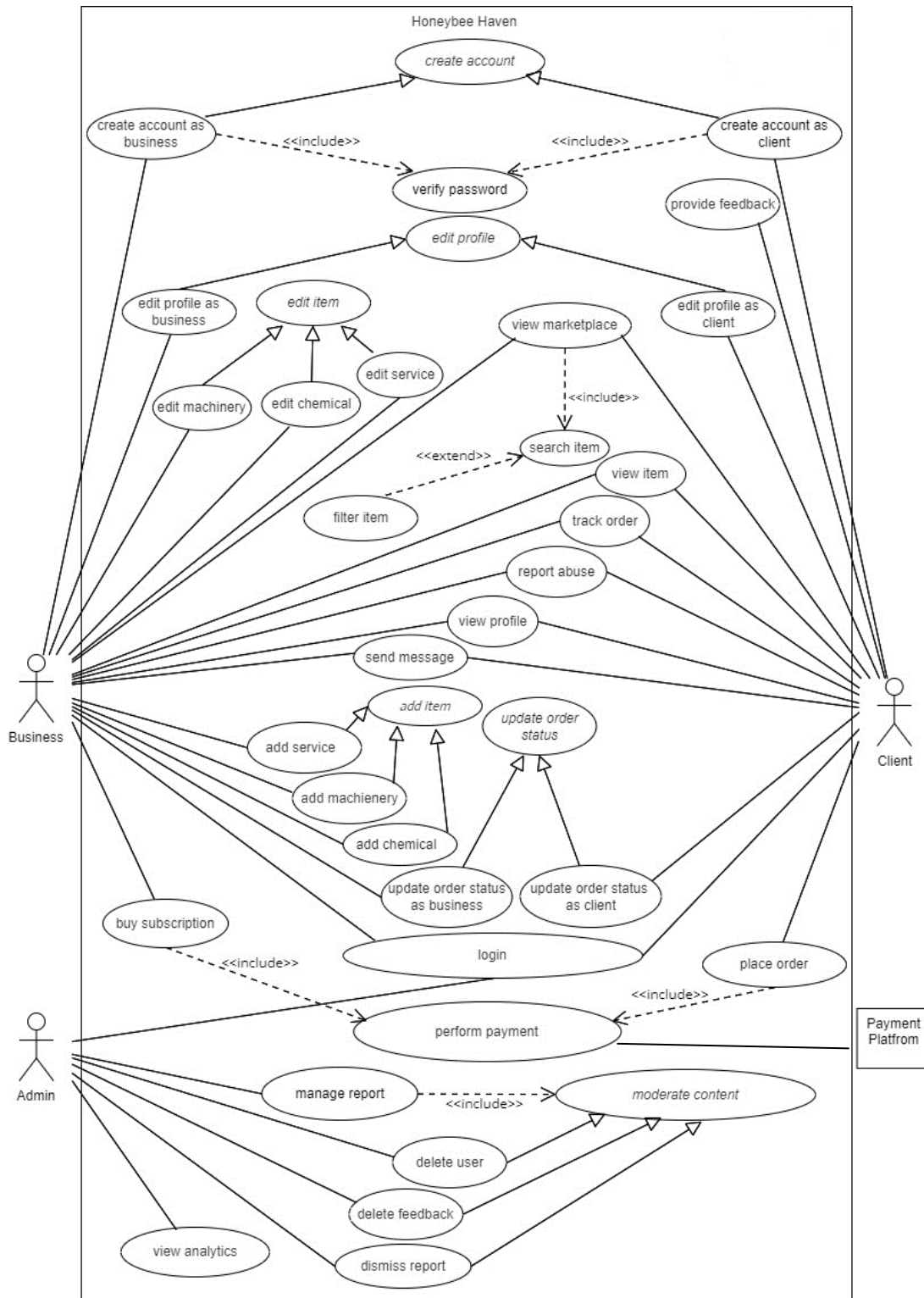
Team I

Member Name	Member Roll #	Primary Responsibility
Anas Nauman	21L-5230	UCD, ACD, UC-27, UC-28, UC-29, UC-30, UC-31, UC-32, UC-33, UC-34
Qasim Naveed	21L-5231	UCD, ACD, Drew AC, UC-1, UC-2, UC-3, UC-4, UC-5, UC-6, UC-7, UC-8
Saad Asif	21L-7532	UCD, ACD, Drew UC, Drew AC, UC-9, UC-18, UC-35, UC-36, UC-37
Wajeeh Ahmed	21L-7565	UCD, ACD, UC-10, UC-11, UC-12, UC-13, UC-14, UC-15, UC-16, UC-17
Muhammad Hammad	21L-5388	UCD, ACD, UC-19, UC-20, UC-21, UC-22, UC-23, UC-24, UC-25, UC-26

Table of Contents

Table of Contents	ii
1. Analysis Class Diagram	1
2. Use Case Diagram	2
3. Use Cases.....	3
3.1 UC-1: create account	3
3.2 UC-2: create account as client	5
3.3 UC-3: create account as business	6
3.4 UC-4: verify password.....	7
3.5 UC-5: edit item	8
3.6 UC-6: provide feedback.....	9
3.7 UC-7: edit profile	10
3.8 UC-8: edit profile as client	11
3.9 UC-9: edit profile as business.....	12
3.10 UC-10: view marketplace	13
3.11 UC-11: search item.....	14
3.12 UC-12: filter item	15
3.13 UC-13: view item	16
3.14 UC-14: track order.....	17
3.15 UC-15: report abuse.....	18
3.16 UC-16: update order status	19
3.17 UC-17: update order status as business	20
3.18 UC-18: update order status as client.....	21
3.19 UC-19: view analytics	22
3.20 UC-20: view profile.....	23
3.21 UC-21: send message	24
3.22 UC-22: add item	25
3.23 UC-23: add service	26
3.24 UC-24: add machinery.....	27
3.25 UC-25: add chemical	28
3.26 UC-26: login.....	29
3.27 UC-27: buy subscription.....	30
3.28 UC-28: place order	31
3.29 UC-29: perform payment.....	33
3.30 UC-30: manage reports.....	34
3.31 UC-31: moderate content.....	35
3.32 UC-32: delete user	36
3.33 UC-33: delete feedback	37
3.34 UC-34: dismiss report.....	38
3.35 UC-35: edit machinery	39
3.36 UC-36: edit service.....	40
3.37 UC-37: edit chemical.....	41

2. Use Case Diagram



3. Use Cases

3.1 UC-1: create account

Identifier		UC-1
Name		create account
Summary		Users can create their account to access functionalities of website
Priority		High
Actor(s)		Client, Business
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on create account	
2		Display create account page
3	Enter email	
Go to UC-4		
4	Enter name	
5	Enter address	
6	Selects account type as client	
7	Clicks "Create"	
8		Check all fields are filled
9		Checks email in database
10		Stores information in database
Go to UC-2		
Alternate Course of Action (Business account)		
S#	Actor Action	System Response
6	Selects account type as business	
7	Clicks "Create"	
8		Check all fields are filled
9		Checks email in database
10		Stores information in database

Go to UC-3		
Alternate Course of Action (Email already registered)		
S#	Actor Action	System Response
10		Displays error modal
Go to step 2		
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
9		Displays error modal
Go to step 3		

3.2 UC-2: create account as client

Identifier	UC-2	
Name	create account as client	
Summary	Clients create their account to buy items	
Priority	High	
Actor(s)	Client	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Display client account page
2	Enter gender	
3	Enter age	
4	Clicks "Submit"	
5		Check all fields are filled
6		Stores information in database
7		Redirect to client's profile
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
6		Display error modal
Go to step 1		

3.3 UC-3: create account as business

Identifier	UC-3	
Name	create account as business	
Summary	Business can create account to display their items	
Priority	High	
Actor(s)	Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Display business account page
2	Enter business name	
3	Enter business description	
4	Enter contact info	
5	Enter target keywords	
6	Enter bank account number	
7	Clicks "Submit"	
8		Check all fields are filled
9		Checks validity of account number
10		Stores information in database
11		Redirect to profile
Alternate Course of Action (Invalid account number)		
10		Display error modal
Go to step 6		
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
11		Display error modal
Go to step 2		

3.4 UC-4: verify password

Identifier	UC-4	
Name	verify password	
Summary	It ensures that user creates a strong valid password	
Priority	Low	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is running	
Post-condition(s)	Website server is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Enter password	
2	Enter password again to confirm	
3		Checks confirm password is same
4		Checks length of password
5		Checks uppercase character
6		Checks special character
7		Checks numeric character
Alternate Course of Action (Invalid password)		
8		Displays invalid password modal
Go to step 1		
Alternate Course of Action (Wrong confirm password)		
4		Displays invalid confirm password modal
Go to step 2		

3.5 UC-5: edit item

Identifier	UC-5	
Name	edit item	
Summary	Business can edit their item’s details	
Priority	Medium	
Actor(s)	Business	
Pre-condition(s)	Website server is running	
Post-condition(s)	Website server is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on “Edit Item”	
2		Determines type of item
3		Redirects to edit machinery page
Go to UC-35		
Alternate Course of Action (Item type is chemical)		
3		Redirects to edit service page
Go to UC-36		
Alternate Course of Action (Item type is service)		
3		Redirects to edit chemical page
Go to UC-37		

3.6 UC-6: provide feedback

Identifier	UC-6	
Name	provide feedback	
Summary	Clients can provide feedback about business and their items	
Priority	High	
Actor(s)	Client	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks "Feedback" on give item	
2		Displays item feedback form
3	Enter rating	
4	Enter review	
5	Clicks "Submit"	
6		Checks rating is entered
7		Stores information in database
8		Redirects to client's profile
Alternate Course of Action (Rating is missing)		
7		Displays error modal
Go to step 2		

3.7 UC-7: edit profile

Identifier	UC-7	
Name	edit profile	
Summary	Users can edit their profile to keep themselves up to date	
Priority	Medium	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is running	
Post-condition(s)	Website server is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Click on "Edit Profile"	
2		Determines type of account
3		Redirects to edit client profile page
Go to UC-8		
Alternate Course of Action (Type of account is business)		
3		Redirects to edit business profile page
Go to UC-9		

3.8 UC-8: edit profile as client

Identifier		UC-8
Name		edit profile as client
Summary		Clients can edit their profile to keep themselves up to date
Priority		Medium
Actor(s)		Client
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1		Fetch client’s data from database
2		Display data
3		Display edit option
4	Edits name	
5	Edits password	
Go to UC-4		
6	Edits address	
7	Edits gender	
8	Edits age	
9	Clicks “Apply Changes”	
10		Check no field is left empty
11		Stores information in database
12		Redirects to client’s profile
Alternate Course of Action (Field information is missing)		
11		Displays error modal
Go to step 4		

3.9 UC-9: edit profile as business

Identifier	UC-9	
Name	edit profile as business	
Summary	Business can edit their profile to keep themselves up to date	
Priority	Medium	
Actor(s)	Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Fetch businesses from database
2		Display data
3		Displays edit option
4	Edits name	
5	Edits business name	
6	Edits password	
Go to UC-4		
7	Edits address	
8	Edits business description	
9	Edits contact info	
10	Edits target keywords	
11	Edits bank account number	
12	Clicks "Apply Changes"	
13		Check no field is left empty
14		Checks validity of account number
15		Stores information in database
16		Redirects to business's profile
Alternate Course of Action (Invalid account number)		
15		Displays error modal
Go to step 11		
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
14		Displays error modal
Go to step 4		

3.10 UC-10: view marketplace

Identifier	UC-10	
Name	view marketplace	
Summary	Allows the user to surf the page containing all post on items offered by businesses	
Priority	High	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on “Marketplace”	
2		Fetch posts from database
3		Filter posts with old, new, best-sellers and sponsored items
4		Display page containing posts
Go to UC-13		
Alternate Course of Action(Search item)		
S#	Actor Action	System Response
5		
Go to UC-11		

3.11 UC-11: search item

Identifier		UC-11
Name		search item
Summary		Page displays product(s) relevant to the product searched by the user in search bar
Priority		Medium
Actor(s)		Client, Business
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1	Enter search title	
2	Clicks "Search"	
3		Fetch matching data from database
4		Displays relevant posts
Alternate Course of Action (No matching title)		
S#	Actor Action	System Response
4		Display No match found modal
Go to step 1		
Alternate Course of Action (Filters showed)		
S#	Actor Action	System Response
5	Clicks on "Add Filter"	
6		Display filters options
Go to UC-12		

3.12 UC-12: filter item

Identifier		UC-12
Name		filter item
Summary		Page displays product(s) relevant to the product searched for in the search bar. Moreover, allows user to apply filters on their search
Priority		High
Actor(s)		Client, Business
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1	Enter recent products filter	
2	Enter price range filter	
3	Enter item type filter	
4	Enter categories filter	
5	Clicks "Search"	
6		Searches matching filters from database
7		Displays relevant posts
Alternate Course of Action (No matching title)		
S#	Actor Action	System Response
7		Display No match found modal
Go to UC-10		

3.13 UC-13: view item

Identifier	UC-13	
Name	view item	
Summary	Displays a page showing all details of the selected item’s price, feedbacks and pictures	
Priority	High	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks “View Details” of a post	
2		Fetch item details from database
3		Displays details of the post

3.14 UC-14: track order

Identifier	UC-14	
Name	track order	
Summary	Allows the user to track their order during the delivery process. E.g., they can check the dispatch date, expected delivery date/time and status of delivery	
Priority	High	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Click on "Track Order"	
2		Fetch orders in transit from database
3		Display list of orders
4	Select order from list	
5		Display details of order modal

3.15 UC-15: report abuse

Identifier	UC-15	
Name	report abuse	
Summary	Allows the user to report any offensive content, posts or conversations	
Priority	High	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on "Report"	
2		Ask reason for report
3	Mention reason	
4		Enqueue in report queue in database
5		Display report recorded successfully message

3.16 UC-16: update order status

Identifier		UC-16
Name		update order status
Summary		Allows the user to update the status of the order being delivered i.e., has it been dispatched or it has been delivered.
Priority		Medium
Actor(s)		Client, Business
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on “Track Order”	
2		Fetch orders in transit from database
3		Display list of orders
4	Select order from list	
5		Display details of order modal
6	Click on “Update Order”	
Go to UC-17		
Alternate Course of Action (Update order status as client)		
S#	Actor Action	System Response
Go to UC-18		

3.17 UC-17: update order status as business

Identifier		UC-17
Name		update order status as business
Summary		Allows the user to update the dispatch status of the order
Priority		Medium
Actor(s)		Business
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1		Prompt for dispatch date
2	Enter date	
3	Clicks "Submit"	
4		Check field is filled
5		Checks validity of date
6		Stores information in database
7		Redirect to Business order list
Alternate Course of Action (Date field is empty)		
S#	Actor Action	System Response
5		Display Error modal
	Go to step 1	
Alternate Course of Action (Date is invalid)		
	Actor Action	System Response
6		Display error modal
	Go to step 1	

3.18 UC-18: update order status as client

Identifier		UC-18
Name		update order status as client
Summary		Allows the user to update the delivery status of the order
Priority		Medium
Actor(s)		Client
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1		Prompt for delivery date
2	Enter date	
3	Clicks "Submit"	
4		Check field is filled
5		Checks validity of date
6		Stores information in database
7		Redirect to Business order list
Alternate Course of Action (Date field is empty)		
S#	Actor Action	System Response
5		Display Error modal
	Go to step 1	
Alternate Course of Action (Date is invalid)		
	Actor Action	System Response
6		Display error modal
	Go to step 1	

3.19 UC-19: view analytics

Identifier	UC-19	
Name	view analytics	
Summary	Admins can view statistical reports	
Priority	High	
Actor(s)	Admin	
Pre-condition(s)	Website server is up, Database is running	
Post-condition(s)	Website server is up, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on the “View Analytics”	
2		Fetch relevant data from database
3		Generate statistic report 1
4		Generate statistic report 2
5		Generate statistic report 3
6		Generate statistic report 4
7		Generate statistic report 5
8		Apply visual illustrations
9		Display final report

3.20 UC-20: view profile

Identifier	UC-20	
Name	view profile	
Summary	User can view the profiles of other clients and businesses	
Priority	Low	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is up, Database is running	
Post-condition(s)	Website server is up, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on "Show Profile" for a user	
2		Fetches the data from database
3		Display profile details

3.21 UC-21: send message

Identifier	UC-21	
Name	send message	
Summary	User can send messages to other users	
Priority	High	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is up, Database is running	
Post-condition(s)	Website server is up, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on message button	
2		Fetches the ID number of sender from database
3		Fetches the ID number of receiver from database
4		Fetches the ID of conversation from database
5		Load previous chat
6		Prompts sender to enter text
7	Enters message text	
8	Clicks "send"	
9		Adds message to the database
10		Shows message to the receiver
Alternate Course of Action (receiver is unavailable)		
S#	Actor Action	System Response
10		Add to receiver's notifications

3.22 UC-22: add item

Identifier	UC-22	
Name	add item	
Summary	Businesses can add the product they want to launch on the platform	
Priority	High	
Actor(s)	Business	
Pre-condition(s)	Website server is running	
Post-condition(s)	Website server is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on “Add Item”	
2		Displays add item page
3	Enters item name	
4	Enters item price	
5	Enter item description	
6	Selects item type as service	
7		Check fields are filled
Go to UC-23		
Alternate Course of Action (Machinery item)		
S#	Actor Action	System Response
6	Selects item type as machinery	
7		Check fields are filled
Go to UC-24		
Alternate Course of Action (Chemical item)		
S#	Actor Action	System Response
6	Selects item type as chemical	
7		Check fields are filled
Go to UC-25		
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
8		Display error modal
Go to Step 2		

3.23 UC-23: add service

Identifier	UC-23	
Name	add service	
Summary	Businesses can add a service that they would like to provide on the platform	
Priority	High	
Actor(s)	Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Displays the service entry form
2	Enters the service type	
3	Enters the base charges	
4	Clicks "Create"	
5		Checks fields are filled
6		Stores information in database
7		Displays success modal
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
7		Displays error modal
Go to Step 1		

3.24 UC-24: add machinery

Identifier	UC-24	
Name	add machinery	
Summary	Businesses can add a machine item that they would like to sell on the platform	
Priority	High	
Actor(s)	Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Displays the machine entry form
2	Enters the machine type	
3	Enters the dimensions	
4	Enters the weight	
5	Enters the power details	
6	Enters the warranty details	
7	Enters the quantity in stock	
8	Clicks "Create"	
9		Checks fields are filled
10		Stores information in database
11		Displays success modal
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
11		Displays error modal
Go to Step 1		

3.25 UC-25: add chemical

Identifier		UC-25
Name		add chemical
Summary		Businesses can add a chemical item that they would like to sell on the platform
Priority		High
Actor(s)		Business
Pre-condition(s)		Website server is running, Database is running
Post-condition(s)		Website server is running, Database is running
Typical Course of Action		
S#	Actor Action	System Response
1		Displays the chemical entry form
2	Enters the metric system	
3	Enters the quantity	
4	Enters the quantity in stock	
5	Enters the expiry date	
6	Enters the hazard level	
7	Enters the chemical type	
8	Clicks "Create"	
9		Checks fields are filled
10		Stores info in database
11		Displays success modal
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
11		Displays error modal
Go to Step 1		

3.26 UC-26: login

Identifier	UC-26	
Name	login	
Summary	User enters email and password to access their accounts	
Priority	High	
Actor(s)	Client, Business	
Pre-condition(s)	Website server is up, Database is running	
Post-condition(s)	Website server is up, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Displays the login form
2	Enters email	
3	Enters password	
4		Fetch credentials from database
5		Matches user-entered credentials
6		Redirect to profile
Alternate Course of Action (Wrong Credentials)		
S#	Actor Action	System Response
6		Display error modal
Go to step 1		

3.27 UC-27: buy subscription

Identifier	UC-27	
Name	buy subscription	
Summary	Business can choose a pricing plan and subscribe to a plan to increase their targeted customers reach.	
Priority	High	
Actor(s)	Business	
Pre-condition(s)	Web server is running, Database is running	
Post-condition(s)	Web server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks "Show Offers"	
2		Display pricing plans and features
3	Selects a plan	
4		Redirects to payment page
Go to UC-29		
5		Apply the features to items
		Update item status in database
6		Display success modal
7		Redirect to homepage
Alternate Course of Action (Error in payment method)		
S#	Actor Action	System Response
8		Display Error modal
Go to step 2		

3.28 UC-28: place order

Identifier		UC-28
Name		place order
Summary		Client places an order of an item of their liking and Honeybee Haven places the order.
Priority		High
Actor(s)		Client, Business
Pre-condition(s)		Web server is up
Post-condition(s)		Web server is up
Typical Course of Action		
S#	Actor Action	System Response
1	Client clicks “Place Order” for an item	
2		Fetch information about item from database
3		Redirect Actor to place order page
4		Display input form
5	Enter contact info	
6	Enter shipping address	
7	Enter required amount	
8	Click “Order”	
9		Calculate and display total cost of order
10		Redirect to payment page
	Go to UC-29	
11		Notify business of new order
12		Redirect Client to homepage
Alternate Course of Action (Error in payment method)		
S#	Actor Action	System Response
13		Display Error modal
14	Client closes modal	
	Go to step 3	
Alternate Course of Action (Client gives invalid input)		
S#	Actor Action	System Response
15		Display Error modal

16	Client closes modal	
	Go to step 3	
	Alternate Course of Action (Item is out of stock)	
S#	Actor Action	System Response
17		Display Error modal
18	Client closes modal	
19		Redirect to homepage

3.29 UC-29: perform payment

Identifier	UC-29	
Name	perform payment	
Summary	Uses our payment partners API platform to conduct real time transaction.	
Priority	High	
Actor(s)	Payment Platform, Client, Business	
Pre-condition(s)	Web server is running	
Post-condition(s)	Web server is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Fill out desired input form boxes
2		Display input form
3	Enter account number	
4	Enter contact info	
5	Accept T&C's	
6	Clicks "Pay"	
7		Check fields are filled
8		Extract required information
9		Send POST request to payment platform using their API
10		Wait for payment platform response
11		Successful transaction response
12		Display success modal
13		Redirect to homepage
Alternate Course of Action (Failed transaction response)		
S#	Actor Action	System Response
14		Redirect to sender page with error
Alternate Course of Action (Field information is missing)		
	Actor Action	System Response
15		Display error modal
16	Go to step 2	

3.30 UC-30: manage reports

Identifier	UC-30	
Name	manage reports	
Summary	Admin can manage reports queue and act accordingly	
Priority	Medium	
Actor(s)	Admin	
Pre-condition(s)	Web server is running, Database is running	
Post-condition(s)	Web server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1	Admin clicks "show reports"	
2		Fetch reports from database
3		Display all reports in queue
4	Admin selects a particular report	
5		Fetch details regarding report
6		Display details of the report
Go to UC-31		

3.31 UC-31: moderate content

Identifier	UC-31	
Name	moderate content	
Summary	Admin analyzes the content of report and takes appropriate action	
Priority	High	
Actor(s)	Admin	
Pre-condition(s)	Web server is running	
Post-condition(s)	Web server is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Display the details, the reason and the context of report
2		Display “Actions” feature for the report
3	Clicks on “Delete User”	
Go to UC-32		
4		Display success modal
5		Redirect main reports page
Alternate Course of Action (Actor selects Delete Feedback)		
S#	Actor Action	System Response
3	Clicks on “Delete feedback”	
Go to UC-33		
Go to step 4		
Alternate Course of Action (Actor selects Dismiss Report)		
	Actor Action	System Response
3	Actor clicks on “Dismiss report”	
Go to UC-34		
Go to step 4		
Alternate Course of Action (Actor selects Close Report)		
	Actor Action	System Response
3	Actor clicks on “Close report”	
Go to step 5		

3.32 UC-32: delete user

Identifier	UC-32	
Name	delete user	
Summary	Admin has the ability to permanently delete account of a violating user	
Priority	Medium	
Actor(s)	Admin	
Pre-condition(s)	Web server is running, Database is running	
Post-condition(s)	Web server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Show confirmation modal
2	Clicks "Confirm"	
3		Delete all relevant data of the user from database
4		Redirect to main reports page
Alternate Course of Action (Actor doesn't click confirm)		
S#	Actor Action	System Response
2		Redirect to main reports page

3.33 UC-33: delete feedback

Identifier	UC-33	
Name	delete feedback	
Summary	Admin has the ability to permanently delete feedback of a violating user	
Priority	Medium	
Actor(s)	Admin	
Pre-condition(s)	Web server is running, Database is running	
Post-condition(s)	Web server is up, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Show confirmation modal
2	Clicks "Confirm"	
3		Delete all relevant data of the feedback from database
4		Update the rating and review of the respective item
5		Notify the Business of the action
6		Notify the Client of the action
7		Redirect to main reports page
Alternate Course of Action (Actor doesn't click confirm)		
S#	Actor Action	System Response
2		Redirect to main reports page

3.34 UC-34: dismiss report

Identifier	UC-34	
Name	dismiss report	
Summary	Admin has the ability to dismiss the report of a violating user	
Priority	Medium	
Actor(s)	Admin	
Pre-condition(s)	Web server is running, Database is running	
Post-condition(s)	Web server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Show confirmation modal
2	Clicks "Confirm"	
3		Update database to remove report details
4		Remove the report from reports queue
		Redirect to main reports page
Alternate Course of Action (Actor doesn't click confirm)		
S#	Actor Action	System Response
2		Redirect to main reports page

3.35 UC-35: edit machinery

Identifier	UC-35	
Name	edit machinery	
Summary	Businesses can edit machinery item to update its details	
Priority	Medium	
Actor(s)	Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Fetches machinery data
2		Displays data
3		Display edit option
4	Edits machine type	
5	Edits dimensions	
6	Edits weight	
5	Edits power details	
6	Edits warranty details	
7	Edits the quantity in stock	
8	Clicks “Edit”	
9		Check fields are filled
10		Stores information in database
11		Redirects to business’s item page
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
10		Displays error modal
Go to step 4		

3.36 UC-36: edit service

Identifier	UC-36	
Name	edit service	
Summary	Businesses can edit service to update its details	
Priority	Medium	
Actor(s)	Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Fetches service data from database
2		Displays data
3		Display edit option
4	Enters the service type	
5	Enters the base charges	
6	Clicks "Edit"	
7		Check fields are filled
8		Stores information in database
9		Redirects to business's item page
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
8		Displays error modal
Go to step 4		

3.37 UC-37: edit chemical

Identifier	UC-37	
Name	edit chemical	
Summary	Businesses can edit chemical item to update its details	
Priority	Medium	
Actor(s)	Business	
Pre-condition(s)	Website server is running, Database is running	
Post-condition(s)	Website server is running, Database is running	
Typical Course of Action		
S#	Actor Action	System Response
1		Fetches chemical data
2		Displays data
3		Display edit option
4	Edits metric system	
5	Edits quantity	
6	Edits quantity in stock	
5	Edits expiry date	
6	Edits hazard level	
7	Edits the chemical type	
8	Clicks “Edit”	
9		Check fields are filled
10		Stores information in database
11		Redirects to business’s item page
Alternate Course of Action (Field information is missing)		
S#	Actor Action	System Response
10		Displays error modal
Go to step 4		