

## <Project Name> Test Case Definitions

Test Case ID: TC-FO-001			Test Description: Verify that a customer can successfully browse a restaurant and food item(s).				
Test Name: Browse Restaurants and Food Items Test			Test Designed by: Muzaffer Berke Savaş				
Use Case Tested : UC-FO-001 (Browse Restaurants and Food Items)			Test Designed Date: 22/03/2025				
Test Priority (Low/Medium/High): High			Test Executed by:<Name-Surname>				
Module Name: Browse Module			Test Execution Date: <dd/mm/yyyy>				
<b>Pre-conditions:</b> <ul style="list-style-type: none"><li>~ The customer must have an active internet connection.</li><li>~ The restaurant and food items must be available in the database.</li><li>~ The restaurant must be open and accepting orders.</li><li>~ The application must have access to the customer's location.</li></ul>							
<b>Dependencies:</b> <ul style="list-style-type: none"><li>~ The database must be properly synced with the latest restaurant and menu data.</li><li>~ The UI components for browsing must be properly loaded.</li></ul>							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Ensure the user is on the home screen or restaurant listing page.	-	The home screen or restaurant listing page is displayed.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

2	Search for a specific restaurant or browse the available restaurants.	Sample restaurant name: TD-1  Valid restaurant address: TD-3  Valid customer address: TD-4	The relevant restaurant (s) appear in the search results or listing with respect to customer city region.				
3	Select a restaurant from the list.	-	The restaurant's menu page loads successfully.				
4	Browse the available food items on the menu.	Sample menu item: TD-2	The menu items with details (name, price, description, etc.) are displayed.				
5	Click on a food item to view its details.	-	The food item's details (name, price, description, add-to-cart option, etc.) are displayed.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

6	Verify that the user can navigate back to the restaurant's main menu.	-	The restaurant's menu is displayed again.				
7	Verify that the user can navigate back to the restaurant listing page.	-	The restaurant listing page is displayed again.				

Test Data Table					
	1	2	3	4	...
	Cezb-Et	T-Bone Steak	99 Finance St, Salem	12 Wall St, Salem	

**Post-conditions:**

- ~ The customer has successfully browsed the restaurant and its food items.
- ~ The customer can navigate back to the home screen or other sections of the app without issues.

<b>Test Case ID:</b> TC-FO-002	<b>Test Description:</b> Verify that a registered customer can successfully place a food order.
<b>Test Name:</b> Place Food Order Test	<b>Test Designed by:</b> Muzaffer Berke Savaş
<b>Use Case Tested :</b> UC-FO-002 (Place Food Order)	<b>Test Designed Date:</b> 25/03/2025
<b>Test Priority (Low/Medium/High):</b> High	<b>Test Executed by:</b> <Name-Surname>
<b>Module Name:</b> Food Ordering Module	<b>Test Execution Date:</b> <dd/mm/yyyy>

**Pre-conditions:**

- ~ The customer must be logged into the system.
- ~ The customer must have at least one saved delivery address.
- ~ The customer must have an active payment method.
- ~ The restaurant selected must be open and accepting orders.

**Dependencies:**

- ~ A database with sample restaurants and menu items must be available.
- ~ The payment gateway should be accessible for payment processing.

Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Search for a restaurant .	Sample restaurant name: TD-1	The restaurant appears in search results.				
2	Select a food item and add it to the cart.	Sample menu item: TD-2	The item is added to the cart.				
3	Proceed to checkout.	-	The checkout page loads with correct order details.				
4	Select a delivery address.	Valid delivery address: TD-3	The address is selected successfully.				
5	Make payment.	Payment details: TD-4	The payment is processed successfully, and the order is confirmed.				

## System Test Case Definitions

Version &lt;x&gt;

Date: &lt;dd/mm/yyyy&gt;

6	Verify order confirmation message.	-	The order confirmation message is displayed .				
Test Data Table							
	1	2	3	4	...		
	Italiano Bro`s	Medium Pepperoni Pizza	123 Main St, Ankara	Credit Card (****0000)			
Post-conditions: ~ The order should be saved in the database. ~ The restaurant should receive the order notification. ~ The customer should be able to track the order status.							

<b>Test Case ID:</b> TC-FO-003				<b>Test Description:</b> Verify that a registered customer can successfully leave a rating and review for a restaurant or courier.			
<b>Test Name:</b> Leave Ratings and Reviews Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested :</b> UC-FO-003 (Leave Ratings and Reviews)				<b>Test Designed Date:</b> 25/03/2025			
<b>Test Priority (Low/Medium/High):</b> Medium				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Review and Ratings Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The customer must be logged into the system. ~ The customer must have completed an order. ~ The review system must be available and functional.							
<b>Dependencies:</b> ~ The customer must have a completed order in their order history. ~ The database must store and retrieve restaurant and courier review data correctly.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

1	Navigate to order history.	-	The order history page loads successfully.				
2	Select an order for review.	Order ID: TD-5	The selected order's details are displayed.				
3	Click on "Leave a Review."	-	The review input form is displayed.				
4	Enter a rating and review comment.	Rating (1-5 stars), review text: TD-6	The review input is accepted.				
5	Submit the review.	-	The system saves the review and displays a confirmation message.				
6	Verify that the review appears on the restaurant/courier profile.	-	The review is displayed under the restaurant/courier reviews section.				
Test Data Table							
1	2	3	4	5	6		

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Italiano Bro`s	Medium Pepperoni Pizza	123 Main St, Ankara	Credit Card (****0000)	Order ID: #456789	Review: "Great food and fast delivery! 5 stars!"
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>~ The review should be stored in the database.</li><li>~ The review should be visible on the restaurant/courier profile.</li><li>~ Other users should be able to see the review and rating.</li></ul>					

Test Case ID: TC-FO-004				Test Description: Verify that a registered customer can successfully save a restaurant to their favorites for quicker access.			
Test Name: Save Favorite Restaurants Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested : UC-FO-004 (Save Favorite Restaurants)				Test Designed Date: 25/03/2025			
Test Priority (Low/Medium/High): Medium				Test Executed by:<Name-Surname>			
Module Name: Favorite Restaurants Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The customer must be logged into the system. ~ The customer must have access to restaurant search or browsing functionality.							
<b>Dependencies:</b> ~ The system must store and retrieve favorite restaurant data correctly. ~ The user profile must have a "Favorites" section to save and display restaurants.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Search for a restaurant .	Sample restaurant name: TD-1  Sample restaurant address: TD-2	The restaurant appears in the search results.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

2	Select the restaurant from the search results.	TD-1 TD-2	The restaurant details page loads successfully.				
3	Click on "Add to Favorites."	-	The system confirms the restaurant has been added to favorites.				
4	Navigate to the "Favorite Restaurants" section.	-	The saved restaurant appears in the list..				
5	Remove the restaurant from favorites.	-	The restaurant is successfully removed from the favorites list.				

Test Data Table

1	2	3	4	5	6
Sushi Spot	456 Way St, Mersin				

**Post-conditions:**

- ~ The selected restaurant should be saved in the customer's "Favorites" section.
- ~ The customer should be able to access the restaurant quickly from the favorites list.
- ~ The customer should be able to remove the restaurant from the favorites if desired.

**Test Case ID:** TC-FO-005**Test Description:** Verify that a registered customer can request a refund or report an issue with an order.



&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

<b>Test Name:</b> Request Refund or Report Order Issue Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested :</b> UC-FO-005 (Request Refund or Report Order Issue)				<b>Test Designed Date:</b> 25/03/2025			
<b>Test Priority (Low/Medium/High):</b> High				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Refund & Order Issue Handling Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> <ul style="list-style-type: none"><li>~ The customer must be logged into the system.</li><li>~ The customer must have placed an order in the past.</li><li>~ The order status must be "Delivered" or "Completed" to request a refund.</li><li>~ The refund and issue reporting system must be available and functional.</li></ul>							
<b>Dependencies:</b> <ul style="list-style-type: none"><li>~ The system must store and retrieve order history correctly.</li><li>~ The refund processing system must be operational.</li><li>~ The administrator must have access to review refund or issue requests.</li></ul>							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to order history.	-	The order history page loads successfully.				
2	Select a past order.	Order ID: TD-1	The order details page is displayed .				
3	Click on "Report an Issue" or "Request a Refund."	-	The issue reporting/ refund request form appears.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

4	Provide issue details or refund reason.	Issue type: TD-2	The system accepts the input and enables the submit button.				
5	Submit the request.	-	The system confirms submission and notifies the admin.				
6	Verify admin response (if applicable).	-	The customer receives a notification with the admin's decision.				

Test Data Table

1	2	3	4	5	6
Order ID: #987654	Issue Type: "Received wrong order" or "Food was cold"				

**Post-conditions:**

- ~ The request should be stored in the system for admin review.
- ~ The administrator should receive a notification of the request.
- ~ The customer should receive a decision on the refund or issue resolution.

<b>Test Case ID:</b> TC-FO-006	<b>Test Description:</b> Verify that a registered customer can successfully set dietary preferences like vegan mode and allergy alerts.
<b>Test Name:</b> Customer Preferences Test	<b>Test Designed by:</b> Muzaffer Berke Savaş

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Use Case Tested : UC-FO-006 (Customer Preferences)				Test Designed Date: 25/03/2025			
Test Priority (Low/Medium/High): Medium				Test Executed by:<Name-Surname>			
Module Name: Customer Preferences Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The customer must be logged into the system. ~ The customer must have access to the preferences section. ~ The system should be able to store and retrieve dietary preferences.							
<b>Dependencies:</b> ~ The system must correctly apply the preferences when browsing or ordering food. ~ The database should store and retrieve customer dietary preferences properly.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the customer preferences section.	-	The preferences page loads successfully.				
2	Select dietary preferences (e.g., vegan, allergies).	Preference Type: TD-1	The selected preference is marked as active.				
3	Save the preferences.	-	The system confirms that preferences have been updated.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

4	Search for food items.	Cuisine: TD-2	The search results are filtered based on preferences.				
5	Attempt to add an item that conflicts with preferences.	Conflicting Item: TD-3	The system displays a warning message.				
Test Data Table							
1	2	3	4	5	6		
Preference: "Vegan Mode"	Cuisine: "Vegan-friendly restaurants"	Conflicting Item: "Beef Burger"					
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>~ The customer's preferences should be stored in their profile.</li><li>~ The system should filter food items according to the saved preferences.</li><li>~ If the customer selects a conflicting item, the system should provide a warning.</li></ul>							

<b>Test Case ID:</b> TC-FO-007	<b>Test Description:</b> Verify that a registered customer can track the status of an active order.
<b>Test Name:</b> Track Order Status Test	<b>Test Designed by:</b> Muzaffer Berke Savaş
<b>Use Case Tested :</b> UC-FO-007 (Track Order Status)	<b>Test Designed Date:</b> 25/03/2025
<b>Test Priority (Low/Medium/High):</b> Medium	<b>Test Executed by:</b> <Name-Surname>
<b>Module Name:</b> Order Tracking Module	<b>Test Execution Date:</b> <dd/mm/yyyy>
<b>Pre-conditions:</b> <ul style="list-style-type: none"><li>~ The customer must be logged into the system.</li><li>~ The customer must have an active order.</li><li>~ The system must have real-time tracking enabled for orders.</li></ul>	

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

- ~ The system must retrieve real-time order status updates.
- ~ The courier system must update order progress (e.g., Preparing, On the Way, Delivered).

Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the "Track Order" section.	-	The active order list is displayed .				
2	Select an active order.	Order ID: TD-1	The order details page loads with real-time tracking.				
3	Verify order status updates.	Order Status: TD-2	The order progresses through status changes (e.g., Preparing , On the Way).				
4	Check estimated delivery time.	Delivery Time Estimate: TD-3	The system displays an estimated arrival time.				
5	Verify courier location (if applicable).	-	The system shows real-time courier location updates.				
Test Data Table							

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

1	2	3	4	5	6
Order ID: #123456	Order Status: "On the Way"	Delivery Time Estimate: "15 mins remaining"			
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>~ The customer should be able to see real-time updates of their order.</li><li>~ The system should update the order status based on delivery progress.</li><li>~ If available, the system should show the courier's location on a map.</li></ul>					

Test Case ID: TC-MN-001				Test Description: Verify that an admin can successfully log into the system using valid credentials.			
Test Name: Admin Login Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested : UC-MN-001 (Admin Login)				Test Designed Date: 25/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Authentication Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The admin must have valid login credentials (username & password) ~ The admin portal must be accessible. ~ The system must be operational and allow admin authentication.							
<b>Dependencies:</b> ~ The system must store and verify admin credentials securely. ~ The authentication system must support error handling for incorrect login attempts.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the admin login page.	-	The login page loads successfully.				

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

[illegible]

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

<b>Test Case ID:</b> TC-DEL-001				<b>Test Description:</b> Verify that a courier can successfully log into the system using valid credentials.			
<b>Test Name:</b> Courier Login Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested :</b> UC-DEL-001 (Courier Login)				<b>Test Designed Date:</b> 25/03/2025			
<b>Test Priority (Low/Medium/High):</b> High				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Authentication Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The courier must have valid login credentials (username & password). ~ The courier portal must be accessible. ~ The system must be operational and allow courier authentication.							
<b>Dependencies:</b> ~ The system must store and verify courier credentials securely. ~ The authentication system must support error handling for incorrect login attempts.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the courier login page.	-	The login page loads successfully.				
2	Enter valid courier credentials and submit.	Username : TD-1, Password : TD-2	The system authenticates the courier and grants access.				
3	Verify successful login.	-	The courier is redirected to the delivery dashboard.				



&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

4	Attempt login with incorrect credentials.	Invalid Username /Password: TD-3	The system displays an error message.				
5	Check for security features (e.g., CAPTCHA, 2FA).	-	The system enforces security measures as required.				
Test Data Table							
1	2	3	4	5	6		
Courier Username: "courierX"	Courier Password: "Courier@Pass1"	Invalid Login: "wrongCourier / wrongPass"					
<b>Post-conditions:</b> ~ If successful, the courier should be logged in and able to access the delivery dashboard. ~ If unsuccessful, the system should display an appropriate error message. ~ The system should log all login attempts for security monitoring.							

<b>Test Case ID:</b> TC-FO-008	<b>Test Description:</b> Verify that a customer can successfully log into the food delivery platform.
<b>Test Name:</b> Customer Login Test	<b>Test Designed by:</b> Muzaffer Berke Savaş
<b>Use Case Tested :</b> UC-FO-008 (Customer Login)	<b>Test Designed Date:</b> 25/03/2025
<b>Test Priority (Low/Medium/High):</b> High	<b>Test Executed by:</b> <Name-Surname>
<b>Module Name:</b> Authentication Module	<b>Test Execution Date:</b> <dd/mm/yyyy>
<b>Pre-conditions:</b> ~ The customer must have valid login credentials (username/email and password). ~ The system must be online and accessible.	
<b>Dependencies:</b> ~ The system must store and verify customer credentials securely. ~ The authentication module must be functional.	

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the login page.	-	The login page loads successfully.				
2	Enter valid credentials and submit.	Username : TD-1, Password : TD-2	The system authenticates the customer and grants access.				
3	Verify successful login.	-	The customer is redirected to the homepage.				
4	Attempt login with incorrect credentials.	Invalid Username /Password: TD-3	The system displays an error message.				
5	Check for security features (e.g., CAPTCHA, 2FA).	-	The system enforces security measures if necessary.				
Test Data Table							
1	2	3	4	5	6		

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Customer Username: "user123"	Customer Password: "Secure@Pas s1"	Invalid Login: "wrongCourier / wrongPass"			
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>~ If successful, the customer should be logged in and redirected to the homepage.</li><li>~ If unsuccessful, the system should display an appropriate error message.</li><li>~ The system should log all login attempts for security purposes.</li></ul>					

Test Case ID: TC-RES-001				Test Description: Verify that a restaurant owner can successfully log into the food delivery platform.			
Test Name: Restaurant Owner Login Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested : UC-RES-001 (Restaurant Owner Login)				Test Designed Date: 26/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Authentication Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The restaurant owner must have valid login credentials (username/email and password). ~ The system must be online and accessible.							
<b>Dependencies:</b> ~ The system must store and verify restaurant owner credentials securely. ~ The authentication module must be functional.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the restaurant owner login page.	-	The login page loads successfully.				
2	Enter valid credentials and submit.	Username : TD-1,  Password : TD-2	The system authenticates the owner and grants access.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

3	Verify successful login.	-	The restaurant owner is redirected to the management dashboard.				
4	Attempt login with incorrect credentials.	Invalid Username / Password: TD-3	The system displays an error message.				
5	Check for security features (e.g., CAPTCHA, 2FA).	-	The system enforces security measures if necessary.				
Test Data Table							
1	2	3	4	5	6		
Restaurant Owner Username: "owner123"	Restaurant Owner Password: "Owner@Pass1"	Invalid Login: "wrongCourier / wrongPass"					
<b>Post-conditions:</b> ~ If successful, the restaurant owner should be logged in and redirected to the management dashboard. ~ If unsuccessful, the system should display an appropriate error message. ~ The system should log all login attempts for security purposes.							

<b>Test Case ID:</b> TC-FO-009	<b>Test Description:</b> Verify that an admin can successfully manage user accounts (customers, restaurants, and couriers).
<b>Test Name:</b> User Account Management Test	<b>Test Designed by:</b> Muzaffer Berke Savaş
<b>Use Case Tested :</b> UC-FO-009 (User Account Management)	<b>Test Designed Date:</b> 26/03/2025

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: User Management Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The admin must be logged into the system. ~ The admin must have appropriate permissions to manage user accounts. ~ The system must store and retrieve user data correctly.							
<b>Dependencies:</b> ~ The system must support CRUD (Create, Read, Update, Delete) operations on user accounts. ~ The admin dashboard must be functional. ~ Email notifications should be sent when a user account is modified.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the "User Management" section in the admin panel.	-	The user management dashboard loads successfully.				
2	Select a user type (customer, restaurant, or courier) and view the user list.	User Type: TD-1	The system displays the corresponding user accounts.				
3	Edit user details and save changes.	Updated User Data: TD-2	The system updates the user details and confirms the changes.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

4	Delete a user account.	User ID: TD-3	The system removes the user account and updates the database.				
5	Verify user notification upon account update.	-	The user receives an email or in-app notification about the changes.				
Test Data Table							
1	2	3	4	5	6		
User Type: "Customer"	Updated User Data: "Changed Email, Address Update"	User ID: "#54321"					
<b>Post-conditions:</b> ~ The updated user details should be stored in the database. ~ The deleted user should no longer have access to their account. ~ The system should log all admin actions for security and auditing purposes.							

Test Case ID: TC-MN-002	Test Description: Verify that an admin can approve or reject registration requests from restaurants and couriers.
Test Name: Registration Approval/Rejection Test	Test Designed by: Muzaffer Berke Savaş
Use Case Tested : UC-MN-002 (Registration Approval/Rejection)	Test Designed Date: 26/03/2025
Test Priority (Low/Medium/High): High	Test Executed by:<Name-Surname>
Module Name: Registration Management Module	Test Execution Date: <dd/mm/yyyy>

**Pre-conditions:**

- ~ The admin must be logged into the system with valid credentials.
- ~ There must be pending registration requests.
- ~ The system must support the approval/rejection process for registrations.

**Dependencies:**

- ~ The system must allow viewing of pending registration requests.
- ~ The platform must notify applicants about approval or rejection decisions.
- ~ If rejected, the system should store a reason and allow reapplication.

Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the "Pending Registrations" section in the admin panel.	-	The list of pending applications is displayed.				
2	Select a pending registration request.	Registration ID: TD-1	The system loads the details of the request.				
3	Review the provided documents and information.	Applicant Data: TD-2	The admin can view all submitted documents.				
4	Approve the registration request.	-	The system approves the request and grants access.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

5	Reject the registration request and provide a reason.	Rejection Reason: TD-3	The system notifies the applicant and allows resubmission.				
Test Data Table							
1	2	3	4	5	6		
Registration ID: "#78901"	Applicant Data: "Restaurant Name, License, Contact Info"	Rejection Reason: "Incomplete Documents"					
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>~ If approved, the applicant gains access to the system as a registered user.</li><li>~ If rejected, the applicant receives a reason and can reapply.</li><li>~ The system logs all approval/rejection decisions for auditing.</li></ul>							

<b>Test Case ID:</b> TC-MN-003	<b>Test Description:</b> Verify that an admin can successfully handle customer complaints and resolve disputes between customers, restaurants, and couriers.
<b>Test Name:</b> Complaint Handling and Dispute Resolving Test	<b>Test Designed by:</b> Muzafer Berke Savaş
<b>Use Case Tested :</b> UC-MN-003 (Complaint Handling and Dispute Resolving)	<b>Test Designed Date:</b> 26/03/2025
<b>Test Priority (Low/Medium/High):</b> High	<b>Test Executed by:</b> <Name-Surname>
<b>Module Name:</b> Complaint & Dispute Management Module	<b>Test Execution Date:</b> <dd/mm/yyyy>
<b>Pre-conditions:</b> <ul style="list-style-type: none"><li>~ The admin must be logged into the system.</li><li>~ There must be at least one registered complaint or dispute in the system.</li><li>~ The system must support communication between involved parties (customer, restaurant, courier).</li></ul>	
<b>Dependencies:</b> <ul style="list-style-type: none"><li>~ The system must allow users to submit complaints and disputes.</li><li>~ The admin should be able to review complaint details, history, and supporting evidence.</li><li>~ If a refund or resolution is required, the system should facilitate it.</li></ul>	



&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the "Complaints & Disputes" section in the admin panel.	-	The list of registered complaints is displayed.				
2	Select a complaint or dispute.	Complaint ID: TD-1	The system loads the complaint details.				
3	Review complaint details and evidence.	Complaint Details: TD-2	The admin can view user-provided evidence (images, receipts, chat history).				
4	Communicate with involved parties (customer, restaurant, courier).	-	The system enables messaging or ticket-based communication.				
5	Resolve the complaint (approve refund, warn a party, close case).	Resolution Action: TD-3	The system applies the resolution and notifies all parties.				

## System Test Case Definitions

Version &lt;x&gt;

Date: &lt;dd/mm/yyyy&gt;

Test Data Table					
1	2	3	4	5	6
Complaint ID: "#112233"	Complaint Details: "Late delivery, missing items, rude courier"	Resolution Action: "Refund issued, warning given to restaurant"			

**Post-conditions:**

- ~ The complaint is marked as resolved, and users are notified of the resolution.
- ~ If a refund is issued, the transaction is logged in the system.
- ~ The system should track complaint trends for future improvements.

<b>Test Case ID:</b> TC-MN-004				<b>Test Description:</b> Verify that an admin can successfully update platform-wide configurations, such as delivery fees and promotional campaigns.			
<b>Test Name:</b> Platform Configurations Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested :</b> UC-MN-004 (Platform Configurations)				<b>Test Designed Date:</b> 27/03/2025			
<b>Test Priority (Low/Medium/High):</b> High				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Configuration Management Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The admin must be logged into the system. ~ The admin must have the necessary permissions to modify platform settings. ~ The system must allow real-time updates to platform configurations.							
<b>Dependencies:</b> ~ The system must support dynamic configuration updates. ~ The changes should apply across all users and modules where relevant. ~ The system should notify users if the configuration affects them (e.g., changes in delivery fees).							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

1	Navigate to the "Platform Configurations" section in the admin panel.	-	The admin configuration dashboard loads successfully.				
2	Select a configuration to update.	Configuration Type: TD-1	The system loads the selected configuration settings.				
3	Modify the configuration values.	New Values: TD-2	The system validates and applies the updates.				
4	Save the updated configuration.	-	The system confirms the update and applies changes.				
5	Verify that affected users are notified (if applicable).	-	Users receive an email or in-app notification about the change.				
Test Data Table							
1	2	3	4	5	6		

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Configuration Type: "Delivery Fee Policy"	New Values: "Base delivery fee increased from \$2.99 to \$3.49"				
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>~ The updated platform configurations should be applied system-wide.</li><li>~ Users affected by the changes should be notified if necessary.</li><li>~ The system should log configuration updates for auditing purposes.</li></ul>					

Test Case ID: TC-FO-005				Test Description: Verify that an admin can suspend or permanently ban restaurants and couriers who violate platform policies.			
Test Name: Suspension of Restaurants or Couriers Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested : UC-FO-005 (Suspension of Restaurants or Couriers)				Test Designed Date: 27/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Account Management & Policy Enforcement Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The admin must be logged into the system. ~ The admin must have permission to manage user suspensions. ~ The restaurant or courier must have committed a policy violation (e.g., repeated late deliveries, fraudulent activity).							
<b>Dependencies:</b> ~ The system must allow viewing of restaurant/courier profiles and violation history. ~ The suspension system must log the action and notify the affected party. ~ If an appeal system is in place, it must allow suspended users to submit a review request.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the "Policy Violations" section in the admin panel.	-	The admin can see a list of flagged restaurants and couriers.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

2	Select a restaurant or courier to suspend.	User ID: TD-1	The system loads the selected user's profile and violation history.				
3	Choose the suspension type (temporary or permanent) and provide a reason.	Suspension Type: TD-2	The system records the action and updates the user's status.				
4	Notify the suspended user via email or in-app notification.	-	The suspended user receives a notification with details.				
5	Verify that the suspended user cannot access their account.	-	The user is blocked from logging in or using the platform.				

Test Data Table

1	2	3	4	5	6
User ID: "Courier_987" or "Restaurant_456"	Suspension Type: "Temporary (7 days) due to repeated late deliveries"				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

**Post-conditions:**

- ~ The restaurant/courier should be restricted from accessing their account during suspension.
- ~ The admin should have the ability to lift the suspension if needed.
- ~ If an appeal system exists, the suspended user should have an option to request a review.

Test Case ID: TC-RES-002				Test Description: Verify that the restaurant owner can add, update, or delete menu items including price, descriptions, and availability.			
Test Name: Menu Management Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested : UC-RES-002 (Menu Management)				Test Designed Date: 27/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Menu Management Module				Test Execution Date: <dd/mm/yyyy>			
Pre-conditions: ~ The restaurant owner must be logged into the system. ~ The restaurant must have a valid profile.							
Dependencies: ~ The platform must allow item creation, editing, and removal with live updates. ~ Menu updates must be reflected immediately for customers.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to "Menu Management" in the dashboard.	-	The restaurant's current menu is displayed.				
2	Click "Add New Item."	TD-1	A form to add a new menu item appears.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

3	Enter item details (name, price, description, availability).	TD-2	The system validates inputs.				
4	Submit the new item.	-	The item is added to the menu and visible on the platform.				
5	Edit an existing item (e.g., update price).	TD-3	The updated item replaces the old version.				
6	Delete an item from the menu.	TD-4	The item is removed and no longer visible to customers.				
7	Attempt to add an item with missing price.	TD-5	The system shows a validation error and blocks submission.				
Test Data Table							
1	2	3	4	5	6		

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Item Name: "Spicy Chicken Wrap"	Price: \$7.99, Description: "Grilled chicken with spicy mayo", Availability: Available	Updated Price: \$8.49	Item to Delete: "Old Veggie Burger"	Missing Price (left blank)	
<b>Post-conditions:</b> ~ Menu is updated with the latest changes. ~ Customers can only see the updated, available menu items. ~ System stores logs of menu changes for record-keeping.					

Test Case ID: TC-RES-003				Test Description: Verify that the restaurant owner can process customer orders and update order status as food is prepared.			
Test Name: Order Processing Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested: UC-RES-003 (Order Processing)				Test Designed Date: 27/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Restaurant Order Management				Test Execution Date: <dd/mm/yyyy>			
Pre-conditions: ~ The restaurant owner must be logged in. ~ There must be at least one incoming order in the system.							
Dependencies: ~ The order placement system must be working properly. ~ Real-time notification must reach the restaurant upon new order. ~ Courier assignment will follow once order is marked “Ready for Pickup.”							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to "Incoming Orders" in the dashboard.	-	Incoming orders are displayed .				



&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

2	Select an incoming order to review.	TD-1	Order details are shown including items, instructions, and delivery info.				
3	Click "Accept Order" to begin food prep.	-	System confirms acceptance and updates status to "In Preparation".				
4	Mark order as "Ready for Pickup" when preparation is complete.	-	Order status updates and becomes visible for courier assignment.				
5	Reject an order and provide a reason.	TD-2	The customer is notified about the rejection via email.				
6	Try to accept an already expired or cancelled order.	TD-3	System prevents action and displays an error.				
Test Data Table							
1	2	3	4	5	6		

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Order ID: #A132	Rejection Reason: "Item out of stock"	Cancelled Order ID: #A129			
<b>Post-conditions:</b> <ul style="list-style-type: none"><li>~ Accepted orders are marked as "Ready for Pickup" when prepared.</li><li>~ Rejected orders are canceled and the customer is notified.</li><li>~ Courier assignment is triggered after marking an order ready.</li></ul>					

Test Case ID: TC-RES-004				Test Description: Verify that the restaurant owner can assign available couriers to orders that are marked as "Ready for Pickup."			
Test Name: Courier Assignment Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested: UC-RES-004 (Courier Assignment)				Test Designed Date: 27/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Courier Assignment Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ The restaurant owner must be logged into the system. ~ There must be at least one order marked as "Ready for Pickup.” ~ At least one available courier must be listed.							
<b>Dependencies:</b> ~ The courier availability system must be operational. ~ Order status should be updated correctly after courier assignment. ~ The assigned courier must receive a notification.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Go to "Orders Ready for Pickup" section in the dashboard.	-	All orders ready for courier assignment are displayed .				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

2	Select an order to assign a courier.	TD-1	Courier assignment screen is displayed.				
3	Choose a courier from the available list.	TD-2	Courier is selected for the delivery.				
4	Click "Assign Courier."	-	System updates the order status to "Out for Delivery" and notifies the courier.				
5	Try to assign a courier when none are available.	TD-3	System shows a warning or suggests retrying later.				
6	Assigned courier rejects the order.	TD-4	System prompts the restaurant owner to assign a different courier.				
Test Data Table							
1	2	3	4	5	6		
Order ID: #A133	Courier: "Courier_001"	No couriers available at assignment time	Assigned courier declined the order				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

**Post-conditions:**

- ~ Order status changes to "Out for Delivery" after courier assignment.
- ~ The assigned courier is notified and marked as "Unavailable" until delivery is completed.
- ~ If courier rejects the delivery, the system allows reassignment.

<b>Test Case ID:</b> TC-RES-005				<b>Test Description:</b> Verify that the restaurant owner can view current deliveries and track the status of ongoing or past orders.			
<b>Test Name:</b> Order Tracking (Restaurant Side) Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested:</b> UC-RES-005 (Order Tracking - Restaurant Owner)				<b>Test Designed Date:</b> 28/03/2025			
<b>Test Priority (Low/Medium/High):</b> Medium				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Order History & Tracking Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ Restaurant owner must be logged in. ~ There must be at least one current or past order in the system.							
<b>Dependencies:</b> ~ Real-time order updates must be working. ~ The system must integrate with courier location tracking (if applicable). ~ Restaurant must have access to order history and live tracking modules.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to "Order History" or "Current Deliveries."	-	A list of past and ongoing orders is shown.				
2	Select an active order to track its delivery.	TD-1	Order details and tracking info are displayed.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

3	View courier location and estimated delivery time.	-	System shows real-time location and ETA.				
4	View completed order details from history.	TD-2	Past order info is displayed including items and delivery status.				
5	Check filtering options (by date, delivery status, etc.).	TD-3	The order list filters correctly.				
6	View empty order history for new restaurant.	TD-4	System displays "No orders yet" message.				

Test Data Table

1	2	3	4	5	6
Order ID: #A135 (Status: "Out for Delivery")	Completed Order ID: #A101	Filter: Last 7 Days, Status: Delivered	Test Restaurant Account with no order history		

**Post-conditions:**

- ~ The restaurant owner can track all current deliveries in real-time.
- ~ Completed orders are available for reference and reporting.
- ~ Any unclear statuses can be manually investigated via contact with couriers.

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

<b>Test Case ID:</b> TC-RES-006				<b>Test Description:</b> Verify that the restaurant owner can view and respond to customer reviews and inquiries.			
<b>Test Name:</b> Review Processing Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested:</b> UC-RES-006 (Review Processing)				<b>Test Designed Date:</b> 28/03/2025			
<b>Test Priority (Low/Medium/High):</b> Medium				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Review Management Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ Restaurant owner must be logged in. ~ There must be customer reviews or inquiries visible in the system.							
<b>Dependencies:</b> ~ Customers must be able to leave reviews after completing orders. ~ The system must allow the restaurant owner to respond via dashboard. ~ Admin moderation system should filter inappropriate content.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to "Reviews & Inquiries" section in dashboard.	-	Reviews and inquiries are listed.				
2	Select a review to respond to.	TD-1	Review details are displayed.				
3	Type and submit a response.	TD-2	Response is saved and displayed under the review.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

4	Attempt to submit an empty or offensive response.	TD-3	System blocks submission and shows validation error.				
5	Customer receives notification about the response.	-	Customer is informed that their review was replied to.				
Test Data Table							
1	2	3	4	5	6		
Review ID: #R101 - "Pizza was cold, took too long."	Response: "We're really sorry! We're improving our delivery time. Thanks for the feedback!"	Response Attempt: Empty or contains inappropriate language (e.g., profanity)					
<b>Post-conditions:</b> ~ Review responses are visible publicly under the original review. ~ Customers are notified of replies. ~ Inappropriate responses are blocked and/or flagged for admin moderation.							

<b>Test Case ID:</b> TC-RES-007	<b>Test Description:</b> Verify that a restaurant owner can successfully view and update their restaurant's profile information.
<b>Test Name:</b> Restaurant Profile Management Test	<b>Test Designed by:</b> Muzaffer Berke Savaş
<b>Use Case Tested:</b> UC-RES-007 (Restaurant Profile Management)	<b>Test Designed Date:</b> 28/03/2025
<b>Test Priority (Low/Medium/High):</b> Medium	<b>Test Executed by:</b> <Name-Surname>
<b>Module Name:</b> Restaurant Profile Module	<b>Test Execution Date:</b> <dd/mm/yyyy>
<b>Pre-conditions:</b> ~ The restaurant owner must be logged into the system. ~ A valid restaurant profile must exist in the system.	

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

### Dependencies:

- ~ The system must validate fields like contact info and delivery range.
- ~ Customers should see updates in real-time when browsing restaurants.

Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Go to “Restaurant Profile” section.	-	Current profile details are displayed .				
2	Edit business hours and save.	TD-1	System saves and confirms the new hours.				
3	Update contact info (email/phone) and save.	TD-2	System validates input and updates profile.				
4	Modify delivery range and save.	TD-3	System checks range feasibility and saves it.				
5	Try saving invalid phone number format.	TD-4	System displays error and blocks submission.				
6	Set hours to unavailable (e.g., closed day).	TD-5	Restaurant is marked closed on customer side.				
Test Data Table							



&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

1	2	3	4	5	6
New Business Hours: Mon–Fri, 11:00–22:00	New Contact Info: Email – <a href="mailto:info@burgerhub.com">info@burgerhub.com</a> , Phone – (555) 123-4567	New Delivery Range: 8 km	Invalid Phone Number: “1234”	Set Sunday as “Closed”	
<b>Post-conditions:</b> <ul style="list-style-type: none"> <li>~ Profile changes are saved and reflected in the system.</li> <li>~ Invalid entries are rejected with proper feedback.</li> <li>~ Updated info is visible to customers browsing the restaurant.</li> </ul>					

<b>Test Case ID:</b> TC-DEL-002				<b>Test Description:</b> Verify that couriers can register with restaurants on the platform to be eligible for delivery assignments.			
<b>Test Name:</b> Courier Restaurant Registration Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested:</b> UC-DEL-002 (Courier Restaurant Registration)				<b>Test Designed Date:</b> 28/03/2025			
<b>Test Priority (Low/Medium/High):</b> High				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Courier-Partner Registration Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ Courier must be registered on the platform and logged in. ~ Courier profile must be complete (e.g., background check, vehicle info).							
<b>Dependencies:</b> ~ Restaurants must be visible for registration based on courier’s location. ~ Courier must be able to submit required info (availability, range). ~ System must allow restaurants to approve/reject registration requests.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Courier logs into the courier dashboard.	-	Courier dashboard is displayed.				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

2	Navigate to "Restaurant Registration" section.	-	List of restaurants available for registration is displayed.				
3	Select a restaurant and fill in registration details.	TD-1	Registration form is completed with required info.				
4	Submit registration request.	-	System sends request to restaurant for approval.				
5	Restaurant approves registration.	-	Courier profile is updated with new restaurant partnership.				
6	Attempt registration with incomplete profile (e.g., missing vehicle info).	TD-2	System blocks submission and prompts for missing info.				
7	Restaurant rejects registration.	TD-3	Courier is notified of rejection and reason.				

&lt;Project Name&gt;

Version &lt;x&gt;

System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

Test Data Table					
1	2	3	4	5	6
Courier Availability: 5pm–9pm, Delivery Range: 10km	Missing Info: No vehicle details provided	Rejection Reason: "Too many couriers registered already"			
<b>Post-conditions:</b> <ul style="list-style-type: none"> <li>~ Approved couriers are linked to restaurants and visible in their courier pool.</li> <li>~ Rejected couriers receive notification and feedback.</li> <li>~ System logs registration actions for transparency and auditing.</li> </ul>					

Test Case ID: TC-DEL-003				Test Description: Verify that couriers can update their availability status to receive or pause delivery requests.			
Test Name: Availability Status Update Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested: UC-DEL-003 (Availability Status Update)				Test Designed Date: 28/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Courier Dashboard – Availability Module				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ Courier must be logged into the system. ~ Courier must be registered with at least one restaurant.							
<b>Dependencies:</b> ~ Status must update in real-time for restaurant owners to see. ~ Courier with active delivery must not be able to mark themselves “Available” again. ~ Restaurants must only assign deliveries to “Available” couriers.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Courier logs into the courier dashboard.	-	Dashboar d loads successfully.				

## System Test Case Definitions

Version &lt;x&gt;

Date: &lt;dd/mm/yyyy&gt;

<b>2</b>	Navigate to "Availability Status" section.	-	Current availability status is displayed .				
<b>3</b>	Set status to “Available.”	TD-1	Courier status updates in system and becomes visible to restaurants.				
-	~	~~~~~	~				
-	~~~~~	~~~~~	~				
-	~~~~~	~~~~~	~				
		~~~~~	~~~~~				

Test Case ID: TC-DEL-004				Test Description: Verify that couriers can accept or reject delivery requests and the system handles both outcomes appropriately.			
Test Name: Accept/Reject Delivery Requests Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested: UC-DEL-004 (Accepting/Rejecting Delivery Requests)				Test Designed Date: 28/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Execution Status	
						Passed	Failed
	1. Login as Courier						
	2. View Delivery Requests						
	3. Accept Request						
	4. Reject Request						
	5. Logout						
	6. Verify Request Status						
	7. Repeat steps for another request						
Tester:		Reviewed By:		Date:			

## System Test Case Definitions

Version &lt;x&gt;

Date: &lt;dd/mm/yyyy&gt;

Test Case ID: TC-DEL-005				Test Description: Verify that couriers can view the details of assigned orders and navigate to the restaurant for pickup.			
Test Name: Courier Order Viewing Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested: UC-DEL-005 (Courier Order Viewing)				Test Designed Date: 28/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Courier Delivery Dashboard Module				Test Execution Date: <dd/mm/yyyy>			
Pre-conditions: ~ Courier must be logged into the system. ~ Courier must have at least one assigned delivery order.							
Test Data							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes		
1	Log in		Successful				
2	View orders		Orders listed				
3	Filter by status	Assigned	Shows assigned				
4	Click on order	Order ID	Order details shown				
5	View map	Address	Map displayed				
6	Click on restaurant	Restaurant ID	Restaurant details shown				
Test Results							
Test Case ID	Test Case Name	Test Case Description	Test Case Status	Test Case Date	Test Case Author		
TC-DEL-005	Courier Order Viewing Test	Verify that couriers can view the details of assigned orders and navigate to the restaurant for pickup.	Passed	28/03/2025	Muzaffer Berke Savaş		

Test Case ID: TC-DEL-006	Test Description: Verify that couriers can mark orders as "Picked Up" or "Delivered" and system updates status accordingly.
Test Name: Courier Order Status Update Test	Test Designed by: Muzaffer Berke Savaş
Use Case Tested: UC-DEL-006 (Order Status Update)	Test Designed Date: 28/03/2025
Test Priority (Low/Medium/High): High	Test Executed by:<Name-Surname>
Module Name: Delivery Flow Module	Test Execution Date: <dd/mm/yyyy>

<b>Pre-conditions:</b> ~ Courier must be logged into the platform. ~ Courier must have an active, assigned delivery task.													
<b>Dependencies:</b> ~ System must sync status changes in real time to restaurant and customer. ~ Courier must only be able to update status at the correct location and stage. ~ If customer is unavailable, system should allow alternate flow.													
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status							
						Pass	Fail						
1	Log in as a courier		Successful login										
2	Click on 'My Deliveries'		Displays active deliveries										
3	Click on a delivery		Shows delivery details										
4	Click on 'View Earnings'		Shows earnings history										
5	Click on 'Delivery History'		Shows past delivery records										
6	Click on 'Customer Ratings'		Shows customer feedback										
7	Log out		Successful logout										
Test Case ID: TC-DEL-007		Test Description: Verify that couriers can view their earnings, delivery history, and customer ratings through the dashboard.											
Test Name: Courier Statistic Viewing Test		Test Designed by: Muzaffer Berke Savaş											
Use Case Tested: UC-DEL-007 (Courier Statistic Viewing)		Test Designed Date: 28/03/2025											
Test Priority (Low/Medium/High): Medium		Test Executed by:<Name-Surname>											
Module Name: Courier Performance Dashboard		Test Execution Date: <dd/mm/yyyy>											
<b>Pre-conditions:</b> ~ Courier must be logged into the system. ~ Courier must have completed at least one delivery.													
<b>Dependencies:</b> ~ System must track and store data on courier activity. ~ Dashboard must display up-to-date performance metrics. ~ Ratings must be tied to verified deliveries.													
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status							
						Pass	Fail						
1	Log in as a courier		Successful login										
2	Click on 'My Deliveries'		Displays active deliveries										

Test Case ID: TC-DEL-007				Test Description: Verify that couriers can view their earnings, delivery history, and customer ratings through the dashboard.			
Test Name: Courier Statistic Viewing Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested: UC-DEL-007 (Courier Statistic Viewing)				Test Designed Date: 28/03/2025			
Test Priority (Low/Medium/High): Medium				Test Executed by:<Name-Surname>			
Module Name: Courier Performance Dashboard				Test Execution Date: <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ Courier must be logged into the system. ~ Courier must have completed at least one delivery.							
<b>Dependencies:</b> ~ System must track and store data on courier activity. ~ Dashboard must display up-to-date performance metrics. ~ Ratings must be tied to verified deliveries.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						—	—
1	Log in as a courier		Successful login				
2	Click on 'My Deliveries'		Displays active deliveries				

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

~	~	~	~				
~	~	~	~				
~	~	~	~				
~	~	~	~				
~	~	~	~				
~	~	~	~	~	~	~	
~	~	~	~	~	~	~	

Test Case ID: TC-DEL-008				Test Description: Verify that couriers can register on the platform by submitting required personal and vehicle information.			
Test Name: Courier Registration Test				Test Designed by: Muzaffer Berke Savaş			
Use Case Tested: UC-DEL-008 (Courier Registration)				Test Designed Date: 28/03/2025			
Test Priority (Low/Medium/High): High				Test Executed by:<Name-Surname>			
Module Name: Courier Registration Module				Test Execution Date: <dd/mm/yyyy>			
Pre-conditions: ~ Courier has access to the internet and platform registration page. ~ Courier has a valid email and required documents ready.							
Dependencies: ~ System must validate submitted documents and credentials. ~ Admin must receive request for approval. ~ System must check for duplicate emails.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	1.1		1.1				
2	2.1	2.1	2.1				
3	3.1	3.1	3.1				
4	4.1		4.1				
5	5.1	5.1	5.1				
6	6.1	6.1	6.1				
7	7.1	7.1	7.1				
8	8.1	8.1	8.1				

<b>Test Case ID:</b> TC-RES-009				<b>Test Description:</b> Verify that restaurant owners can register on the platform by providing necessary business and contact information.			
<b>Test Name:</b> Restaurant Owner Registration Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
<b>Use Case Tested:</b> UC-RES-009 (Restaurant Owner Registration)				<b>Test Designed Date:</b> 28/03/2025			

&lt;Project Name&gt;

Version &lt;x&gt;

## System Test Case Definitions

Date: &lt;dd/mm/yyyy&gt;

<b>Test Priority (Low/Medium/High):</b> High				<b>Test Executed by:</b> <Name-Surname>			
<b>Module Name:</b> Restaurant Registration Module				<b>Test Execution Date:</b> <dd/mm/yyyy>			
<b>Pre-conditions:</b> ~ Restaurant owner must have access to the registration form and a stable internet connection. ~ Owner must have all required documents and info ready (e.g., business license).							
<b>Dependencies:</b> ~ System must validate all required fields. ~ Registration info must be stored and sent to admin for review. ~ Duplicate accounts (same email) must be detected and flagged.							
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F
1	Navigate to the "Register as Restaurant" page.	-	Registration form loads properly.				
2	Enter valid business details.	TD-1	All required fields are accepted.				
-	-	-	-				
-	-	-	-				
-	-	-	-				
-	-	-	-				
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

<b>Test Case ID:</b> TC-FO-010				<b>Test Description:</b> Verify that new customers can successfully register on the platform using valid credentials and receive verification.			
<b>Test Name:</b> Customer Registration Test				<b>Test Designed by:</b> Muzaffer Berke Savaş			
Test Case ID: TC-FO-010-01				Test Case ID: TC-FO-010-02			
Test Case ID: TC-FO-010-03				Test Case ID: TC-FO-010-04			
Test Case ID: TC-FO-010-05				Test Case ID: TC-FO-010-06			
Step	Test Steps	Test Data	Expected Results	Actual Results	Notes	Status	
						P	F




AI PROMPTS



testcasegpt.txt