|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Place Order | |
| **Scenario:** | Place a new food order | |
| **Triggering Event:** | User opens the mobile app to order food from the catalog of various restaurants. | |
| **Brief Description:** | The user selects the order food option on the restaurant profile. The system displays three order food options: Pick-up, Dine-in and Home Delivery. The user selects any one of the options from the three options. System displays the selected page to the user in the app. This use-case is divided into three sub use-cases based on the selection of the order type by the user. The three sub-cases are described below as Pick-up, Book Table and Home Delivery. | |
| **Actors:** | User (Primary) | |
| **Related Use Cases:** | Search Restaurants, Book Table | |
| **Stakeholders:** | *User*: wants to order food  *Restaurant*: wants to prepare and deliver the order/food  *Government Tax Agencies*: want to collect tax from every sale (may be multiple agencies, such as national, state or county  *Payment Authorization Service*: wants to receive digital authorization requests; wants to accurately account for their payables to the system | |
| **Preconditions:** | 1. User should be signed-in on the app 2. Home delivery option should be supported in User’s area. | |
| **Post conditions:** | 1. Unique order number must be generated 2. Order must be associated to a valid user 3. Order must be associated and forwarded to selected restaurant 4. Valid discount coupons must be used 5. Valid payment method must be selected | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. User searches the restaurants through by applying different kinds of filters such as, cuisines, city, ratings etc. 2. User chooses restaurant 3. User views menu and other details 4. User starts order 5. Add item 6. Repeat step 6 until the user wants 7. User ends the order 8. User chooses order type option as Home Delivery 9. User chooses to book a table (if chosen ‘dine-in’ order type and wish to reserve table) 10. Apply Coupon 11. User selects payment details 12. Do payment | 1.1 Displays search results  3.1 Displays restaurant details  5.1 Create order item.  6.1 Add item to order  7.1 Ends the order  9.1 Mark and set the delivery type  10.1 Calculate Total  10.2 Display Payment information  12.1 Do order transaction  12.2 Finish Order |
| **Exception Conditions:** | 1. If the desired restaurant/item in menu isn’t available, then User can:    1. Choose another restaurant/item    2. Choose not to place an order 2. If the discount coupon isn’t valid, then the User can:    1. Pay the full price    2. Choose not to place an order 3. If ‘home delivery’ service is not present in the User’s area, then he/she can:    1. Choose either ‘take away’ or ‘dine in’ as order type    2. Choose not to place an order 4. If User’s card authorization fails, then he/she can:    1. Choose to pay by ‘cash on delivery’ (if ‘home delivery’ is valid)    2. Choose not order | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Rate and review | |
| **Scenario:** | User provides rating and reviews for a restaurant | |
| **Triggering Event:** | Initiate rating and writing review | |
| **Brief Description:** | User selects to Rate & review any restaurant from the restaurant profile of the restaurant. System displays the rate & review screen to the user. User selects the rating out of 10 and writes a review in the review area. User submits the review. System accepts the review and goes to the admin for the verification of review. | |
| **Actors:** | User (Primary)  Admin (Secondary) | |
| **Related Use Cases:** | Verify Review | |
| **Stakeholders:** | *User*: Wants to gives the feedback for the restaurant  *Restaurant*: Will get the feedback from the user, which will help in marketing of the restaurant.  *Gourmand App*: Will get orders from the different users who sees the review and order food from that restaurant. | |
| **Preconditions:** | User must be logged in to the system and opened the restaurant for which the user wants to write a review. | |
| **Post conditions:** | Ratings and reviews should be submitted into the system for approval of system admin | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. User opens the Reviews tab from the restaurant profile page 2. User selects to rate and add a new review for the restaurant 3. User selects a rating from 1 to 10 in the form. 4. User writes a review in the review area inside the review form 5. User submits the review. | 1.1 Displays the list of reviews of the restaurants.  2.1 Display the new rate and review form to the user.  3.1 System displays the number selected by the user as a rating.  4.1 System displays the running review currently written by the user.  5.1 System saves the review and checks for any obscene words in the review. If system finds any such words, the review is sent to the admin to verify review and pending review status is shown to the user. |
| **Exception Conditions:** | 1. User writes some obscene language in the review. Such review will not be immediately posted in the list of reviews for that restaurant. It will be sent to admin for verification. 2. User can rate and review only those restaurants from where he/she has ordered the food. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Book Table | |
| **Scenario:** | User can reserve a table in a restaurant while pre-ordering the food | |
| **Triggering Event:** | User selects ‘dine in’ as order type and opt for reserving a table | |
| **Brief Description:** | User selects the number of persons to book a table. System displays the table availability to the user. User selects the table and books the table. System displays the confirmation of table being booked and also displays the confirmation number to the user. | |
| **Actors:** | User (Primary) | |
| **Related Use Cases:** | Place order, Pay for Order | |
| **Stakeholders:** | User: wants to reserve a table while pre-ordering the food  Restaurant: A restaurant will get table booked through the app which will increase their business. | |
| **Preconditions:** | 1. User should place an order 2. Restaurant should have registered with the system for booking tables | |
| **Post conditions:** | Table at the restaurant should be reserved | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. User places an order 2. User selects ‘dine in’ as order type 3. User enters the number of persons, date and time for the table booking 4. User selects the table to book. | 1.1 Creates order  2.1 Display table details  3.1 System displays the available tables in the time and date selected by the user.  4.1 System displays the table information and book table receipt to the user. |
| **Exception Conditions:** | If there are no tables available for booking, then user can:   * 1. Choose to continue to place order as ‘dine in’ (without table reservation)   2. Choose to continue to place order by opting for other order types   3. Choose to not proceed with placing the order | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Track order | |
| **Scenario:** | User can track the order placed earlier by the user | |
| **Triggering Event:** | User selects to track the previously placed order | |
| **Brief Description:** | User selects the Track Order feature in the system. System displays the track order screen to the user. User enters the order id in the order id field. System displays the current order status to the user. | |
| **Actors:** | User (Primary) | |
| **Related Use Cases:** | None | |
| **Stakeholders:** | *User*: Wants to track the status of the order placed  *Restaurant*: Will let the user know about the current status of the order and thus increase the customer service. | |
| **Preconditions:** | User should have placed the order | |
| **Post conditions:** | Current status of the order placed should be displayed | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. User opens the tracking function in the app 2. User enter the order id in the input text. | 1.1 Displays the input to enter the order id of the order.  2.1 Displays the current status of the order being tracked |
| **Exception Conditions:** | If user provided the invalid order details, then user is prompted to enter the correct/valid details. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Search Restaurant | |
| **Scenario:** | Search restaurant by applying different filters such as cuisine, city and ratings etc. | |
| **Triggering Event:** | User selects to search some restaurants from the app search functionality | |
| **Brief Description:** | User selects the filter options from the filter bar and filters the search based on cuisine, rating, price range, distance, and other features like coupons available, order tracking, newly opened, open now, free delivery, express delivery and rapid pick-up. System refines the list and shows a new list of restaurant based on the new filter criteria entered by the user. | |
| **Actors:** | User (Primary) | |
| **Related Use Cases:** | Sort Restaurant, View Restaurant | |
| **Stakeholders:** | *User*: Wants to search various restaurants from the catalog based on some search criteria.  *Restaurants*: Will benefit if the user selects to search with their name and increase awareness to the user. | |
| **Preconditions:** | User should be signed into the app | |
| **Post conditions:** | Should display the search results as per the search criteria | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. User selects the search functionality from the app 2. User enters the name of the city in the search bar 3. User selects the filter option in the search bar 4. User selects certain filtering options and clicks on the search button again | 1.1 Displays the search bar in the app  2.1 Displays the list of restaurants available in the selected city  3.1 Displays the filter options like cuisine, rating, price, range etc  4.1. Displays the filtered list if restaurants matching the filter criteria |
| **Exception Conditions:** | 1. If the search result is null, then user can choose to search by providing a different search criteria.  2. If the filter selected is not available for any list of restaurants, then the same list as previous will be displayed by the system. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Event | |
| **Scenario:** | Restaurant can add events on an occasion | |
| **Triggering Event:** | Restaurant adds event | |
| **Brief Description:** | Restaurant selects to add a new event. New event screen is displayed by the system. Restaurant enters event name, date, time and location of the event and submits the event. System saves the event and updates the events on the dashboard of the restaurant | |
| **Actors:** | Restaurant (Primary) | |
| **Related Use Cases:** | None | |
| **Stakeholders:** | *Restaurant*: Wants to add event provided in the app  *User*: Will get notified for the event if the user is already subscribed with the restaurant. | |
| **Preconditions:** | Restaurant should be registered with the app/system and have a dashboard | |
| **Post conditions:** | System should display the event on its page & profile. | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. Restaurant sign into the app 2. Restaurant selects to add a new event from the dashboard 3. Restaurant enters details like event name, date and time for the event 4. Restaurant submits the event to the system | 1.1 Displays the restaurant’s dashboard  2.1 Displays the add new event form to the restaurant.  3.1 Displays the filled form by the restaurant  4.1 Saves the event and displays the event on the restaurant dashboard as well as the restaurant profile which is viewed by the users. Users who are subscribed with the restaurants also get notified about the event. |
| **Exception Conditions:** | 1. If restaurant’s account does not exist, then restaurant can:    1. Choose to register and create an account 2. If restaurant forgets username, then it can:    1. Choose to request username to be sent on restaurant’s registered email-id 3. If restaurant forgets password, then it can:    1. Choose to reset the password | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | User Login | |
| **Scenario:** | Sign-in the user into the app | |
| **Triggering Event:** | User enters the username and password | |
| **Brief Description:** | User enters username (email address) and password and clicks on Login button. The system verifies the username and password and if correct, logs the user into the system and displays the dashboard to the user else it shows invalid username/password. | |
| **Actors:** | User (Primary) | |
| **Related Use Cases:** | None | |
| **Stakeholders:** | *User*: wants to access and manage account; wants to order food | |
| **Preconditions:** | User should have an active account | |
| **Post conditions:** | Should display user’s dashboard after successfully logging into the system. | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. User enters the username and password in the input fields on the login screen 2. User selects the login option from the login page | 1.1 Displays the username and displays the password as asterisks  1.2 System checks the username and password and authenticates the user. The dashboard of the user is displayed to the user |
| **Exception Conditions:** | 1. If user’s account does not exist, then user can:    1. Choose to create an account 2. If user forgets username, then user can:    1. Choose to request username to be sent on user’s registered email-id 3. If user forgets password, then user can:    1. Choose to reset the password | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add Restaurant Coupon | |
| **Scenario:** | Restaurant can add their coupons in the app | |
| **Triggering Event:** | Restaurant adds coupon | |
| **Brief Description:** | Restaurant select to add a new coupon. System displays the new coupon page. Restaurant enters the coupon fields like coupon name, start date, end date, discount and description and submits the coupon. System saves the coupon and updates the coupon in the system. The restaurant profile will now display this coupon which can be viewed by all the users. | |
| **Actors:** | Restaurant (Primary) | |
| **Related Use Cases:** | None | |
| **Stakeholders:** | *Restaurant*: Wants to add their coupon in the app  *User*: Can use the coupon while ordering food for that restaurant to get some discount. | |
| **Preconditions:** | Restaurant should be registered with the app/system and have a dashboard | |
| **Post conditions:** | System should display the coupon on its page & profile. | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. Restaurant selects to add a new coupon to their page 2. Restaurant enters the details like coupon name, start date, end date, discount and description of the coupon. 3. Restaurant submits the coupon to the system. | * 1. Displays the new coupon page   2.1 Displays the filled new coupon form  3.1 Saves the coupon and updates it to the list of coupons available for the restaurant. Restaurant profile is also updated where the users can view the newly added coupon |
| **Exception Conditions:** | 1. Restaurant is not a registered restaurant. In such case, the restaurant cannot add coupons to their page | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Manage Restaurants | |
| **Scenario:** | Gourmand admin can manage the restaurant in the app | |
| **Triggering Event:** | Gourmand admin initiates the restaurant management | |
| **Brief Description:** | Admin selects to add/update/delete a restaurant. System opens the selected page based on the selection of the admin. In case of a new restaurant, a New Restaurant page is opened and admin enters information of the restaurant and submits. System saves the restaurant and updates it to the list of restaurants. | |
| **Actors:** | Admin (Primary) | |
| **Related Use Cases:** | None | |
| **Stakeholders:** | *Admin*: wants to manage restaurants in the app | |
| **Preconditions:** | 1. For adding restaurant, it should not be present in the system 2. For removing restaurant, it should be present in the system 3. For managing restaurant, it should be registered and have a valid dashboard | |
| **Post conditions:** | Should update the app with the latest information | |
| **Flow of Events:** | **User (actor)** | **System** |
| 1. Admin opens the app 2. Admin manages the restaurant inventory by performing all or either of the following:    1. Admin adds a new restaurant    2. Admin removes an existing restaurant    3. Admin modifies the restaurant dashboard | 1.1 Displays the app content  2.1 Updates the app with latest information |
| **Exception Conditions:** | 1. While adding restaurant, if it is already present in the system then admin can choose to update its details. 2. If a restaurant is not having a valid dashboard, then admin can provide/tag a valid dashboard to the restaurant. | |