

MeHungry

Use Case Document

Revision History

Please keep the latest version on top

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver. | Change Description | Date | Author | Reviewer |
| 1.0 | Initial Commit | 28th Aug 2016 | Gouravmoy, Saswat | Suraj |
| 1.1 | Functionalities added and modifications by Suraj Added | 29th Aug 2016 | Gouravmoy, Saswat | Suraj |

The terms used in the document are explained below:

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| RMA | Restaurant Mobile Application |
| RWA | Restaurant Web Application |
| RSA | Restaurant Super Admin |
| RM | Restaurant Manager |
|  |  |
|  |  |
|  |  |

## Reviewer Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Document Version | Review Comments | Reviewer | Status |
| V 1.1 | Provision for partners for RSA | Suraj | Added in the Document |
| V 1.1 | Provision to assign/revoke roles of a level at par or below to the sub-ordinates. | Suraj | Added in the Document |
| V 1.1 | Provision for the Chef (Let's use Chef instead of Cook), Captain (Let's use Captain instead of Waiter) and Staff to add and edit their personal details; any modification of which, to be notified to the hierarchy above. | Suraj | Changed Cook to Chef. Did not modify Captain as it may be ambiguous |
| V 1.1 | Provision to group hotels. | Suraj | Added in the Document |
| V 1.1 | Provision to change the hierarchy. | Suraj | Added in the Document |
| V 1.1 | Any deletion should not delete anything from the database. Upon any deletion, the verification team shall verify within 7 days and then reflect it on the database. For minor deletion such as item from Menu, etc., the deletion should be reflected on the app instantly but the Manager and upper hierarchy should keep receiving reminders at regular intervals for 7 days, post which, it shall be removed from the database. | Suraj | Added in the Document |

Table of Contents

[Reviewer Comments 3](#_Toc460285968)

[**1.** **Access Control** 5](#_Toc460285969)

[**1.1** **Roles** 5](#_Toc460285970)

[**1.2** **Role Definitions** 5](#_Toc460285971)

[**1.2.1** **RestaurantSuperAdmin (multiple)** 5](#_Toc460285972)

[**1.2.2** **RestrauntManager (multiple)** 6](#_Toc460285973)

[**1.2.3** **Chef/Waiter/Staff** 6](#_Toc460285974)

[**1.2.4** **MeHungrySuperAdmin** 6](#_Toc460285975)

[**2.** **Functionalities** 6](#_Toc460285976)

[**2.1** **RestaurantSuperAdmin** 6](#_Toc460285977)

[**2.2** **RestrauntManager** 7](#_Toc460285978)

[**2.3** **Waiter** 7](#_Toc460285979)

[**2.4** **Chef** 7](#_Toc460285980)

[**2.5** **Staff** 7](#_Toc460285981)

[**3.** **Use Cases** 8](#_Toc460285982)

[**3.1** **Restaurant Super Admin Sign Up** 8](#_Toc460285983)

[**3.2** **Restaurant Super Admin Login** 8](#_Toc460285984)

[**3.3** **Restaurant Creation** 8](#_Toc460285985)

[**3.4** **Restaurant Manager Login** 9](#_Toc460285986)

[**3.5** **Waiter Login** 9](#_Toc460285987)

[**3.6** **Chef Login** 10](#_Toc460285988)

[**4.** **Use Case Customer Side** 10](#_Toc460285989)

[**4.1** **Customer Sign up** 10](#_Toc460285990)

[**4.2** **Customer Login** 10](#_Toc460285991)

[**4.3** **Customer Forgot Password** 11](#_Toc460285992)

[**4.4** **Customer Profile** 11](#_Toc460285993)

[**4.5** **Customer Home Page** 11](#_Toc460285994)

[**4.6** **Inside Restaurant Flow** 11](#_Toc460285995)

[**4.7** **Outside Restaurant Flow (Pre-Order/Book Table)** 11](#_Toc460285996)

[**4.8** **Book Table Flow** 12](#_Toc460285997)

# **Access Control**

# **Roles**

These are the Roles for the Restaurant side of MeHungry

* RestaurantSuperAdmin (multiple)
* RestrauntManager (multiple)
* Chef (multiple)
* Waiter (multiple)
* Staff (multiple)
* MeHungrySuperAdmin

1. In general, the roles and their access are such that any user can assign/revoke roles of a level at par or below to the sub-ordinates.

2. Implementation of custom role to be created by the RSA or RM on the go. The custom roles created will have to give permissions accordingly by the RSA

# **Role Definitions**

# **RestaurantSuperAdmin (multiple)**

* 1. There can be more than 1 RSA in case of partners
  2. The one of the partners who will sign up in the web application
  3. He will create restaurants (one or multiple). The restaurant id will be generated (always unique).
  4. There should be an option to create a group of hotels. Statistics and Analytics will be shown according to the group.
  5. He can create/modify payment credentials for every restaurant.
  6. He can create/modify restaurant’s manager (He needs to provide the managers email id and email id will then be verified after which the manager can login via web application). There has to be an option to make the Super Admin as the Restaurant Manager
  7. He can create/modify Chef, waiter and staff for particular restaurants. For creating the Chef, waiter and staff, only username, password is required.
  8. He can view the statistics and analytics of one or more of his restaurant.
  9. The Restaurant Super Admin is by default the manager of the restaurants he creates. So he can log in to the restaurant mobile application as a manager using the restaurant id.
  10. The RSA can change the hierarchy of other roles.

# **RestrauntManager (multiple)**

1. By default Restaurant Super admin is a manager.
2. He can log in to the web application as well as RMA
3. One or more Restaurant managers will be created by Restaurant Super Admin for a particular restaurant
4. He can create/modify Chef, waiter and staff for particular restaurants. For creating the Chef, waiter and staff, only username, password is required.
5. He can view the statistics and analytics of his own restaurant in web as well as mobile application
6. He can modify/add the roles of a restaurant employee. For example he can give access to a Chef of a waiter.

# **Chef/Waiter/Staff**

* 1. He will be created by RSA or RM
  2. He will log in by mobile application giving Restaurant id, User name and password

# **MeHungrySuperAdmin**

1. Will be created using scripts directly into the database
2. It will have access to all the api’s
3. He will verify when a restaurant is added. Before his verification, some of the functionalities like ordering will not be allowed.
4. Super Admin will authorize the payment credentials

# **Functionalities**

This defines the functionalities of all the stake holders of the Restaurant part.

# **RestaurantSuperAdmin**

1. Creates the Restaurant
2. Creates/Modify the manager
3. He can delete a restaurant
4. He provide the payment credentials of all the restaurants
5. He can perform all the tasks of a manager as he is the default manager of a restaurant
6. He can see the statistics and analytics of all the restaurants under him
7. When the RCA logins in the web application, then he will be treated as a super admin but when he log’s in the RMA with the restaurant id, then he will be treated as a manager.

# **RestrauntManager**

* + 1. Web Application Part

These are the functionalities that a restaurant manager can access in web application.

1. After the restaurant manager has been created by the RSA, the manger needs to verify his email id. Only after that he can login in the web.
2. He creates the staff members, assigns them username and password
3. He can see the analytics and statistics about his restaurant
   * 1. Mobile Application
4. He has to log in into the mobile application using the restaurant id
5. He can cancel an order
6. He can order on behalf of a customer
7. He can change the status of an item as per availability
8. He can check all the order status
9. He can close payments

# **Waiter**

1. Can update his profile details which will notified to the manager
2. Waiter can use only the mobile application
3. Waiter can place order
4. Waiter can cancel order
5. He can close an order created by a customer
6. He can check the status of all orders
7. He can’t close the payment cycle

# **Chef**

1. Can update his profile details which will notified to the manager and waiter
2. Cannot cancel a order
3. Can change the status of a order
4. Can provide an estimated time of arrival of food

# **Staff**

1. Can update his profile details which will notified to the manager and waiter
2. Some api read access

# **Use Cases**

# **Restaurant Super Admin Sign Up**

1. New restaurant owner will provide email id, phone number and password for sign up
2. Once sign up is done, email verification needs to be done.
3. Once email verification is done, mobile number verification should happen.
4. If Email is not verified with in 24hrs, the entry will be deleted from database.

# **Restaurant Super Admin Login**

1. Registered RSA should be able to login by giving email Id or phone number and password.
2. If email is not verified then show error.
3. If email is verified and phone number is not verified, ask the RSA to verify the phone number.
4. Forgot Password facility should be provided, if email is verified.
5. Once Logged in RSA can create restaurants.

# **Restaurant Creation**

(In future we will provide personalization for the restaurant app and let the RSA choose what the template and logo for the app)

1. Once email and phone number is verified, the RSA can create one or more restaurants.
2. The Create Restaurant Work Flow
   * 1. Name, Address, Location, phone number. This is a mandatory step.
   1. Once these fields are given a thumbnail for the restaurant will be created .The RSA clicks on the thumbnail .After that the following steps can be performed.
      1. Upload Menu, Here we will provide import menu as csv file or create menu individually. The csv template to import is also being provided .This is not a mandatory step.
      2. Create a manager. By default the RSA is the manager of all the restaurants he owns. He can add one or more managers. When adding a manager, he has to provide manager’s email id, phone number and password. This is not a mandatory step.
      3. Create other staffs like waiters, cooks, staffs. RSA only needs to provide username and password for them. This is not a mandatory step.
      4. Provide payment details for monitory transaction like pan card, bank account number.RSA only has the privilege to update these fields.
      5. The RSA will be able to see the analytics and statistics of the entire restaurant as well as overall data.
   2. After the creation of the restaurant, the verification process starts. The verification will be done by MeHungry Admin. Once the verification is complete, QR code/Beacon assignment is to be started.

# **Restaurant Manager Login**

* + 1. Web Application Part
    2. RM can login after his email has been verified
    3. The manager can see all the details of the restaurant/s he is the manager of. He cannot see the restaurants in which he is not the manager.
    4. Upload Menu, Here we will provide import menu as csv file or create menu individually. The csv template to import is also being provided .This is not a mandatory step.
    5. Create other staffs like waiters, cooks, staffs. RSA only needs to provide username and password for them. The RM can also change the roles of any of its staff. This is not a mandatory step.
    6. Payment details for monitory transaction are not accessible to the Manager of the restaurant.
    7. The RM will be able to see the analytics and statistics of the restaurant.
    8. Mobile Application Part

1. The RM will login to the RMA using the restaurant ID, email and password for the first time. After that it will auto login
2. Create other staffs like waiters, cooks, staffs. RSA only needs to provide username and password for them. The RM can also change the roles of any of its staff. This is not a mandatory step
3. He should have all active orders and history of all orders.
4. He can cancel an order
5. He can order on behalf of a customer
6. He can change the status of an item as per availability. We will add inventory functionality to check the availability of the product.
7. He can check all the order status
8. He can close payments

# **Waiter Login**

1. The waiter will login using restaurant id, its username and password
2. He can see some statistics (Like his own order of the day, etc)
3. He should have all active orders, My active orders, My status and all status for that particular day.
4. Waiter can place order
5. Waiter can cancel order only if the order is not in progress state
6. The waiter can add more items any time but cannot delete any item if the order is in “In progress” state.
7. He can close an order created by a customer
8. He can check the status of all orders
9. He can’t close the payment cycle

# **Chef Login**

1. Chef Logs In using restaurant id, its username and password.
2. Chef can change the status of an order
3. Chef cannot cancel an order
4. Chef can notify the RM and the waiter if an item is finished.

Any deletion should not delete anything from the database. Upon any deletion, the verification team shall verify within 7 days and then reflect it on the database. For minor deletion such as item from Menu, etc., the deletion should be reflected on the app instantly but the Manager and upper hierarchy should keep receiving reminders at regular intervals for 7 days, post which, it shall be removed from the database.

# **Use Case Customer Side**

# **Customer Sign up**

1. The customer should be able to sign up using social media login like Facebook, Google
2. If it a social media signup then, the user can register and verify his phone number at later point of time.
3. If the user chooses to register, then he has to provide email, phone number and password.
4. User will not be allowed enter into the mobile app until email or phone number is verified.
5. We will try to verify both email (A link will be sent to mail id) and phone number (Auto verify).
6. If either one is verified, then he will able to login in the mobile application.
7. Phone number and email Id combination should be unique. Email Id or phone number with a different combination will not be allowed to signup.

# **Customer Login**

1. Either email or phone number has to be verified for the customer to login successfully.
2. The customer can provide email or phone number and password to login.
3. If email is not verified then customer should login through the verified phone number and vice versa. If both are not verified the customer should not be able to login and the entry in the database will be deleted after 24hrs.

# **Customer Forgot Password**

1. In case of forgot password, the user should enter verified email id or phone number.

# **Customer Profile**

1. Basic information of customer will be given here.
2. Customer can update information like name, DOB, sex, profile picture etc.
3. If phone number or email id not verified, then customer can verify here.
4. Customer can update its phone number or email id here. And It is compulsory to verify the same else can not be updated.

# **Customer Home Page**

1. There will be two options i.e. Per Order and Inside Restaurant
2. Below these two options, It will show all the restaurants who all are subscribed to MeHungry.
3. Listing of restaurants in home Page will follow some criteria i.e. popular restaurants, restaurants near by the customer etc.
4. If you click any restaurant on the Home page, it will follow the Pre order flow.

# **Inside Restaurant Flow**

1. Again there will be two options i.e. Scan QR Code and Search Menu
2. Both the option will give you the restaurant menu.
3. Then you can select items and add them to cart.
4. Then you go to check out where the summary of the order is given. If accepted, then the user places the order.
5. Once the order is placed the user is redirected to the Track order page where he can track his order at real time.
6. The user after receiving the order can also place ad-hoc orders of food
7. Once the user has done with all the ordering and has completed the food, he can complete the order upon which he is redirected to payment option page
8. In the payment option page the user has an option to pay in cash or payment gateway. In case the user selects cash, the payment can be closed by the manager.
9. Once the foodie has eaten, he will be asked to rate the food he has eaten, give some comments on the food if required. He can also rate the restaurant and comment about the restaurant too.

# **Outside Restaurant Flow (Pre-Order/Book Table)**

1. This flow will be initiated minimum before 1 hour. A user cannot preorder food less than 1 hour.
2. This flow is generally used when the user is not present in the restaurant and wants to pre order his dining before visiting the restaurant.
3. The user selects the Pre Order option upon which he is redirected to the list of nearby supported restaurants.
4. Only those restaurants that have opted for pre-order facility will be shown in the list to the user. The restaurant will have an option to choose if they want to go for pre order. The restaurant will provide the days of the week and time and how many free tables they have and when they can allow pre order. If they have opted then they have to book a table for the customer.
5. Once the user selects the preorder option, he is asked the time and the number of guests he is bringing.
6. Then the user is asked to order the food. It is the normal ordering process as the one in the Inside the restaurant flow.
7. Payment for the food is mandatory. Once the payment is done a confirmation is sent to the mail/phone no. A request is also triggered to the restaurant. The restaurant starts preparing the food.
8. If the user decided to cancel the pre-order before 45 minutes of the order time, then no money is deducted and the whole money is refunded. After 45 minutes, some money is deducted and the rest is refunded
9. If the user is not reaching after 45 minutes of the time of booking, then we will not guarantee the table or preorder of food anymore and the refund as per policy will be initiated after getting confirmation from the user. If the user decides to cancel then the money as per policy will be refunded. Else the user can go and have food in the restaurant for which he has already paid.
10. Once the foodie has eaten, he will be asked to rate the food he has eaten, give some comments on the food if required. He can also rate the restaurant and comment about the restaurant too.

# **Book Table Flow**

1. This flow is generally used when the user is not present in the restaurant and wants to book a table before hand.

1. The user selects the Pre Order option upon which he is redirected to the list of nearby supported restaurants.
2. Only those restaurants that have opted for pre-order facility will be shown in the list to the user. The restaurant will have an option to choose if they want to go for pre order. The restaurant will provide the days of the week and time and how many free tables they have and when they can allow pre order. If they have opted then they have to book a table for the customer.
3. Once the user selects the preorder option, the user is given an option to preorder the food or just book the table. If book the table is selects book a table option then we will check the availability of the tables for the restaurant at that time and book the table.
4. We will guarantee the booked table for 45 minutes after the dead line after which we will not guarantee the table to the customer.
5. We will share the details of the customer to the restaurant. The restaurant will try to keep in contact with the customer whether he will come or not.