Venue Bot Spec v1.0.1

SPEC V 1.0.1

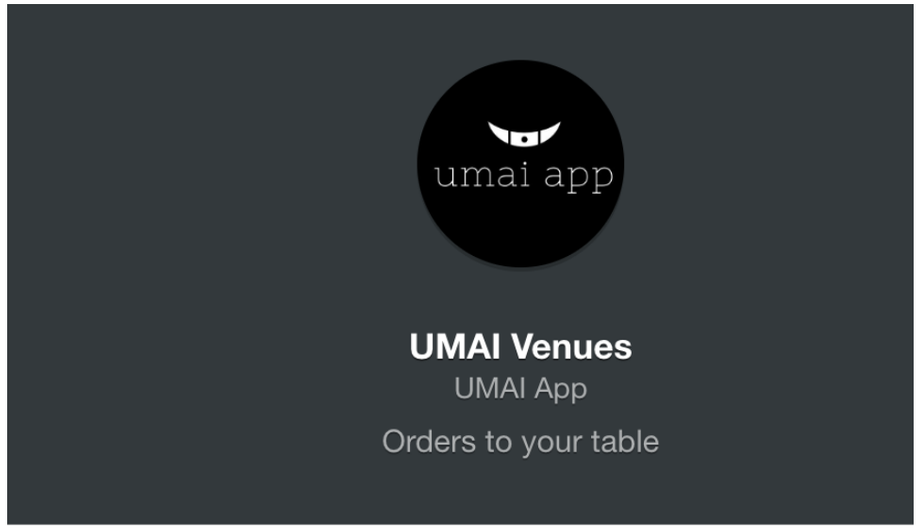
CHAPTER 1

Login and Register

This section describes
- Login process and messaging
- Registration process and messaging

http://localhost:8090/swagger-ui.html#/

Below screen is only for sample purposes

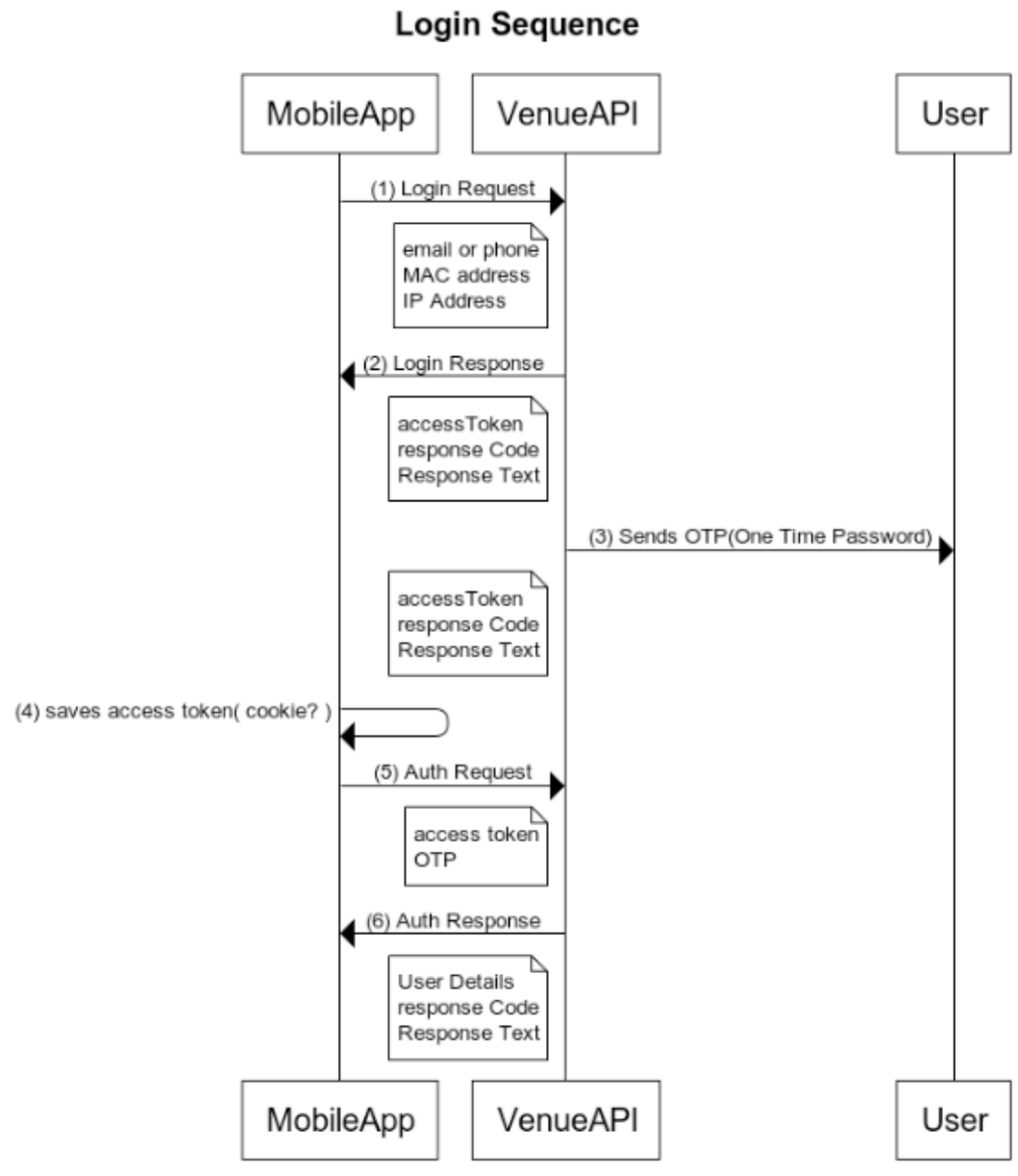


Section 1: Login

This is the sequence for first time login, or when server doesn't validate the access token for the auto login.

Login Process

1. Registered user enters *e-mail or phone* and clicks to login. Mobile app sends this request to APIs (with MAC Address or unique identifier of the device and IP address)
2. API returns an *access token*
3. API sends a text message or email user with OTP(One Time Password)
4. MobileApp stores this access token in the device (will be used to auto login later)
5. User puts the OTP on mobile App and Mobile app sends auth request with access token and OTP
6. API returns the response with User Details

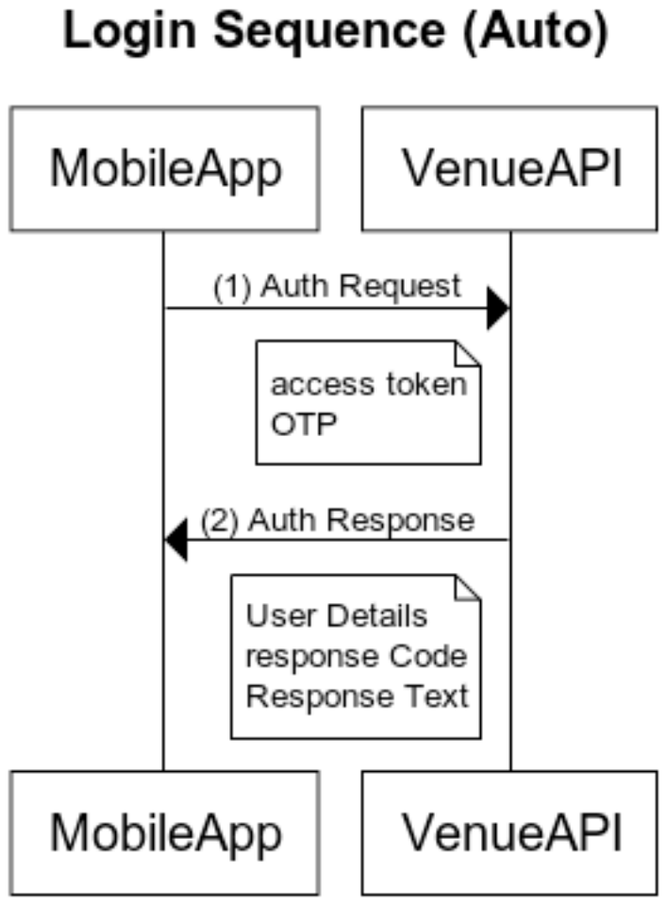


Section 2: Login (REMEMBER ME)

When there is an **access token** stored in the device to remain signed in. (either in cookies or local storage). Then mobile application does NOT ask user to login, but does below sequence to automatically login

Login Process

1. User puts the OTP on mobile App and Mobile app sends auth request with access token and OTP (note OTP here is '*11***' as user will not receive OTP) and mac address of the device and IP address (if available)
2. API returns the response with User Details if the access token is valid by server

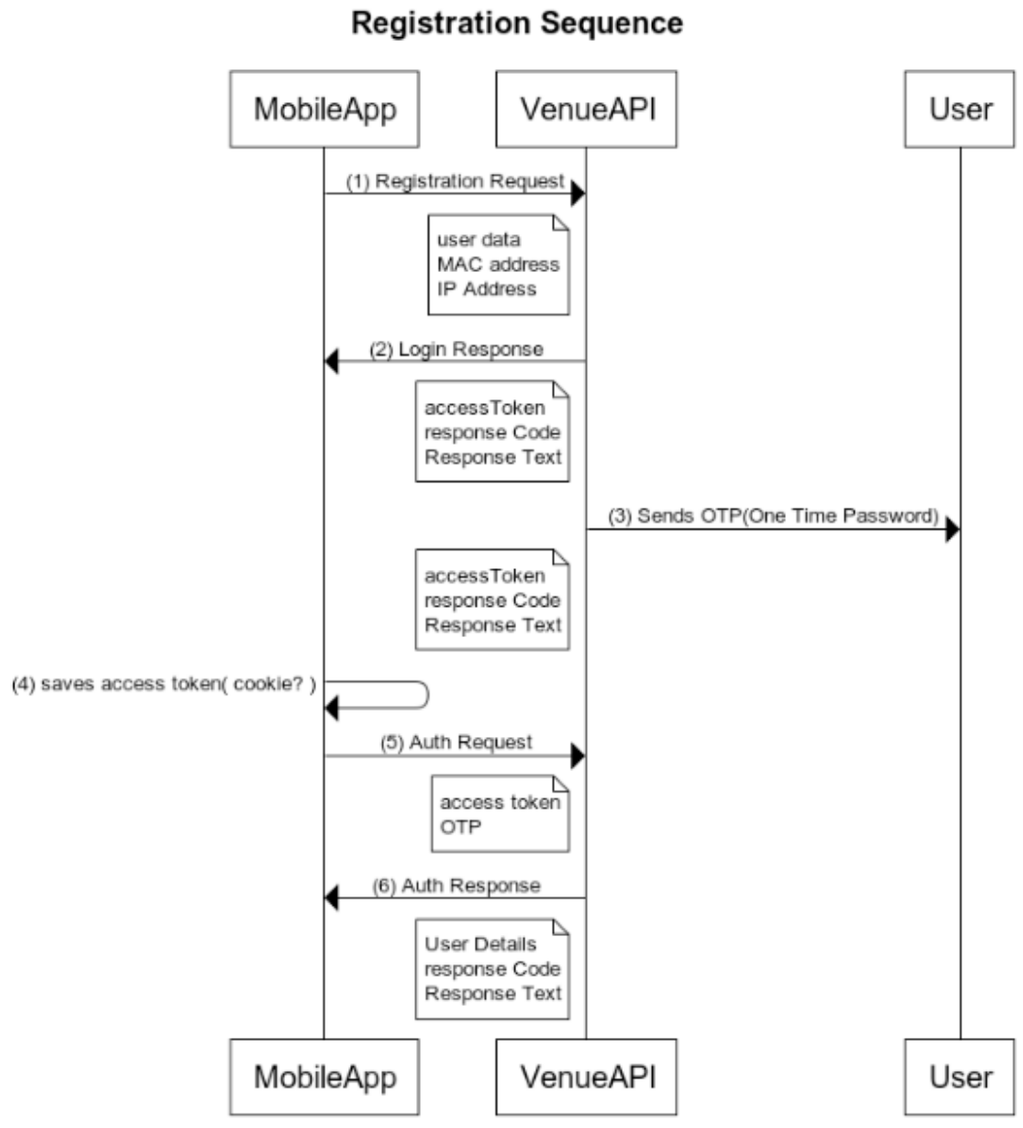


Section 3: Registration

Registration sequence is very much similar to login process. User logins immediately once he/she activated with the OTP

Login Process

1. User puts his details for registration and Mobile App sends it to API
2. API returns an *access token*
3. API sends a text message or email user with OTP(One Time Password)
4. MobileApp stores this access token in the device (will be used to auto login later)
5. User puts the OTP on mobile App and Mobile app sends auth request with access token and OTP
6. API returns the response with User Details



Register Request


```
{
  "user" : {
    "name" : "Bora",
    "lastName" : "Uzun",
    "knownAs" : "Bora",
    "gender" : "MALE",
    "email" : "bora@borauzun.net",
    "phone" : "+447856874102",
    "dobYear" :§ 1979,
    "dobMonth" : 3,
    "dobDay" : 6
  },
  "mac" : "a28:89",
  "loginType" : "CUSTOMER",
  "loginDevice" : "MOBILE"
}
```


HTTP Header

Content-Type: application/json

audience : mobile-customer

* **bold** ones are the mandatory fields

Response

Expected success message


```
{
   "responseCode": 200,
   "accessToken": "d47cc033-a388-49d9-87c6-57a409192bca"
}
```


We will use this accessToken in HTTP Header of the following request,

which is to retreive user details

Expected error message


```
{
   "responseCode": 100,
   "responseText": "Mandatory fields"
}
```

Login Request(by email)

{

"email" : "bora@borauzun.net",

"mac" : "a28:89",

"loginType" : "CUSTOMER",

"loginDevice" : "MOBILE"

}

HTTP Header

Content-Type: application/json

audience : mobile-customer

* **bold** ones are the mandatory fields

Response

Expected success message

{

"responseCode": 200,

"*accessToken*": "d47cc033-a388-49d9-87c6-57a409192bca"

}

We will use this accessToken in HTTP Header of the following request,

which is to retreive user details

Expected error message

{

"responseCode": 100,

"responseText": "Mandatory fields"

}

Login Request(by phone)


```
{
  "phone" : "+447856874102",
  "mac" : "a28:89",
  "loginType" : "CUSTOMER",
  "loginDevice" : "MOBILE"
}
```


HTTP Header

Content-Type: application/json

audience : mobile-customer

* **bold** ones are the mandatory fields

Response

Expected success message


```
{
   "responseCode": 200,
   "accessToken": "d47cc033-a388-49d9-87c6-57a409192bca"
}
```


We will use this accessToken in HTTP Header of the following request,

which is to retreive user details

Expected error message


```
{
   "responseCode": 100,
   "responseText": "Mandatory fields"
}
```

Auth Request (first time or forgotten)

HTTP Body

```
{
  "otp" : "112233",
  "mac" : "a28:89",
  "loginType" : "CUSTOMER",
  "loginDevice" : "MOBILE"
}
```

HTTP Header

Authorization: d47cc033-a388-49d9-87c6-57a409192bca

Content-Type: application/json

audience : mobile-customer

* **bold** ones are the mandatory fields

Response

Expected success message

```
{
   "responseCode": 200,
   "user": {
      "name": "Bora",
      "lastName": "Uzun",
      "knownAs": "Bora",
      "email": "bora@borauzun.net",
      "phone": "+447856874102",
      "gender": "MALE",
      "dobYear": 1979,
      "dobMonth": 3,
      "dobDay": 6
   }
}
```

Expected error message

```
{
   "responseCode": 100,
   "responseText": "accessToken is not valid"
}
```

Auth Request (auto login- when an accessToken is available on device)

HTTP Body

```
{
  "otp" : "*11***",
  "mac" : "a28:89",
  "loginType" : "CUSTOMER",
  "loginDevice" : "MOBILE"
}
```

HTTP Header

Authorization: d47cc033-a388-49d9-87c6-57a409192bca

Content-Type: application/json

audience : mobile-customer

* **bold** ones are the mandatory fields

Response

Expected success message

```
{
   "responseCode": 200,
   "user": {
      "name": "Bora",
      "lastName": "Uzun",
      "knownAs": "Bora",
      "email": "bora@borauzun.net",
      "phone": "+447856874102",
      "gender": "MALE",
      "dobYear": 1979,
      "dobMonth": 3,
      "dobDay": 6
   }
}
```

Expected error message

```
{
   "responseCode": 100,
   "responseText": "accessToken is not valid"
}
```

Business Registration

Very similar to customer registration. Only difference in request is the field “venueName” . This should. be asked user as “Business name”

Then the Text Result that is received from API should be displayed to user.

Response

Expected success message


```
{
  "responseCode": 200,
  "user": {
    " venueName": "Matyat Hotel",
     "name": "Bora",
     "lastName": "Uzun",
     "knownAs": "Bora",
     "email": "bora@borauzun.net",
     "phone": "+447856874102",
     "gender": "MALE",
     "dobYear": 1979,
     "dobMonth": 3,
     "dobDay": 6
  }
}
```


Expected error message


```
{ "responseCode": 200,
  "responseText": "We received your request. Please check your e-mail to complete your registration "
}
```

CHAPTER 2

Home Screens

This chapter explains the screens after logging in. There will be 4 tabs on the home screen

- Nearby
- Favourites
- Recents
- Search

Section 1: HOME

As soon as user logs in(our auto login) we divert to home screen with user's coordinates.

HOME Process

1. As soon as user logs in. Application calls the API (HTTP PUT) with its location

http://45.40.135.209:8090/general/getVenues

2. API returns a list of Venues available split by

- nearby

- favourites

- recents

- search

3. App displays the 'nearby' tab first, then displays others depending on the user choice.

Sample Request

(gives the coordination, if no access to any, then default to 0


```
{
 "x" : 12,
 "y" : 14,
 "z" : 25
}
```


Sample Response

Please see postman

Section 2: HOME - SEARCH TAB

If user clicks on ‘Search Tab’ on the home screen. In this version, We only support search by VenueId.

HOME Process

1. User selects ‘Search’ Tab on home screen
2. User puts the VenueId on press for search button
3. APP calls API, APP returns the search (it will be only 1 venue, or nothing as the search is with VenueId)

Sample Request

(gives the coordination, if no access to any, then default to 0

```
{
 "x" : 12,
 "y" : 14,
 "z" : 25
}
```

Sample Response

Please see postman for the full response

Venue Details in the response

```
{
        "venueId": 1,
        "name": "A Bar",
        "address": "120 Moorgate, London EC2M 1A",
        "distance": 0.1,
        "distanceUnit": "miles",
        "imgUri": "https://merkezlokantasi.com/wp-content/uploads/2018/05/merkez-logo-2.png"
    }
```

* We don’t need to show Venue Id to user when displaying venues.

CHAPTER 3

Venue Screens-1 (intro)

This section describes
- Main Screen for the selected Venue Menu
- Registration process and messaging

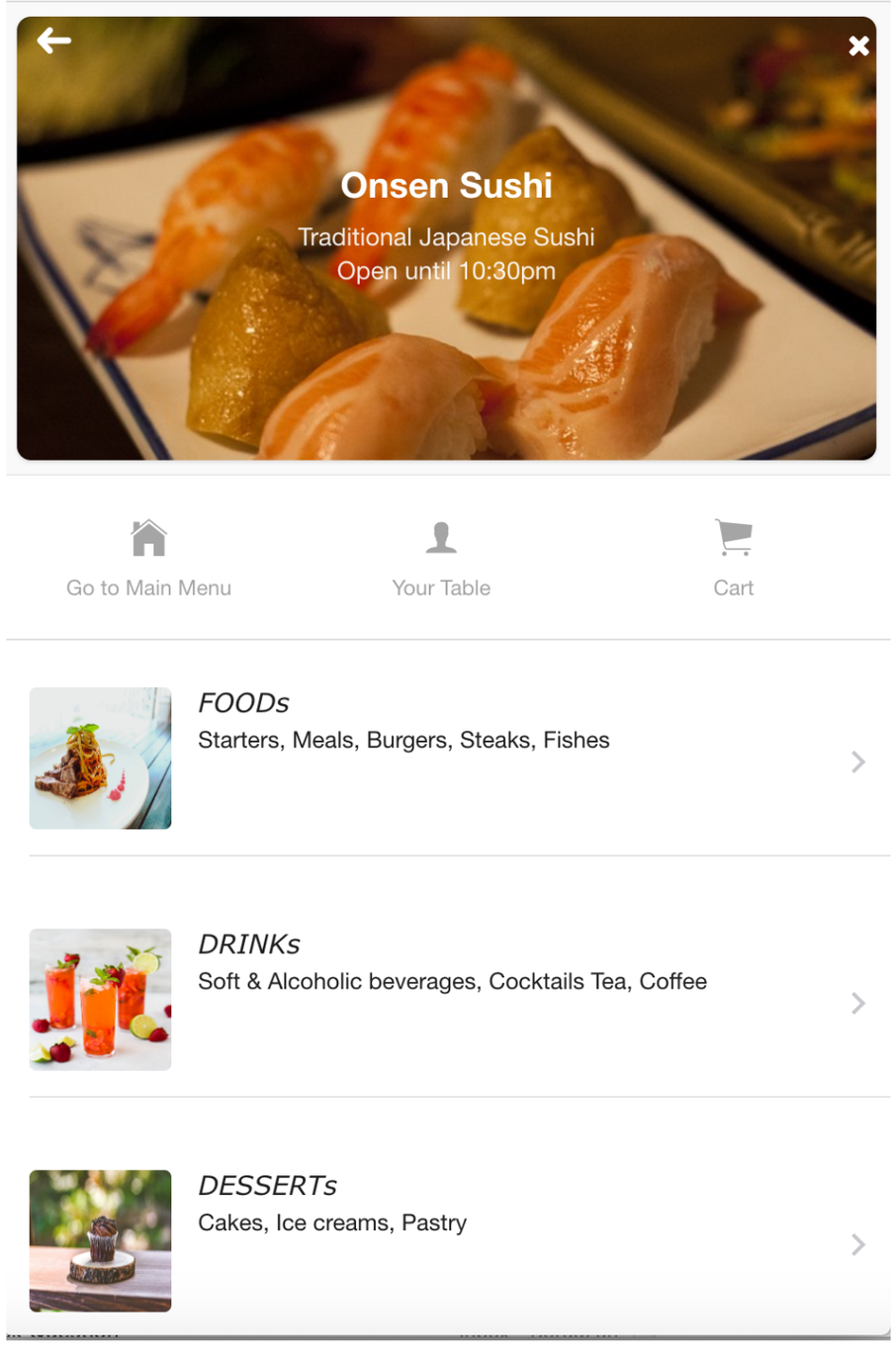
Below screen is only for sample purposes

Section 1: Inside The Venue (Venue Main Menu)

Once the user selects the Venue among the list of venues.

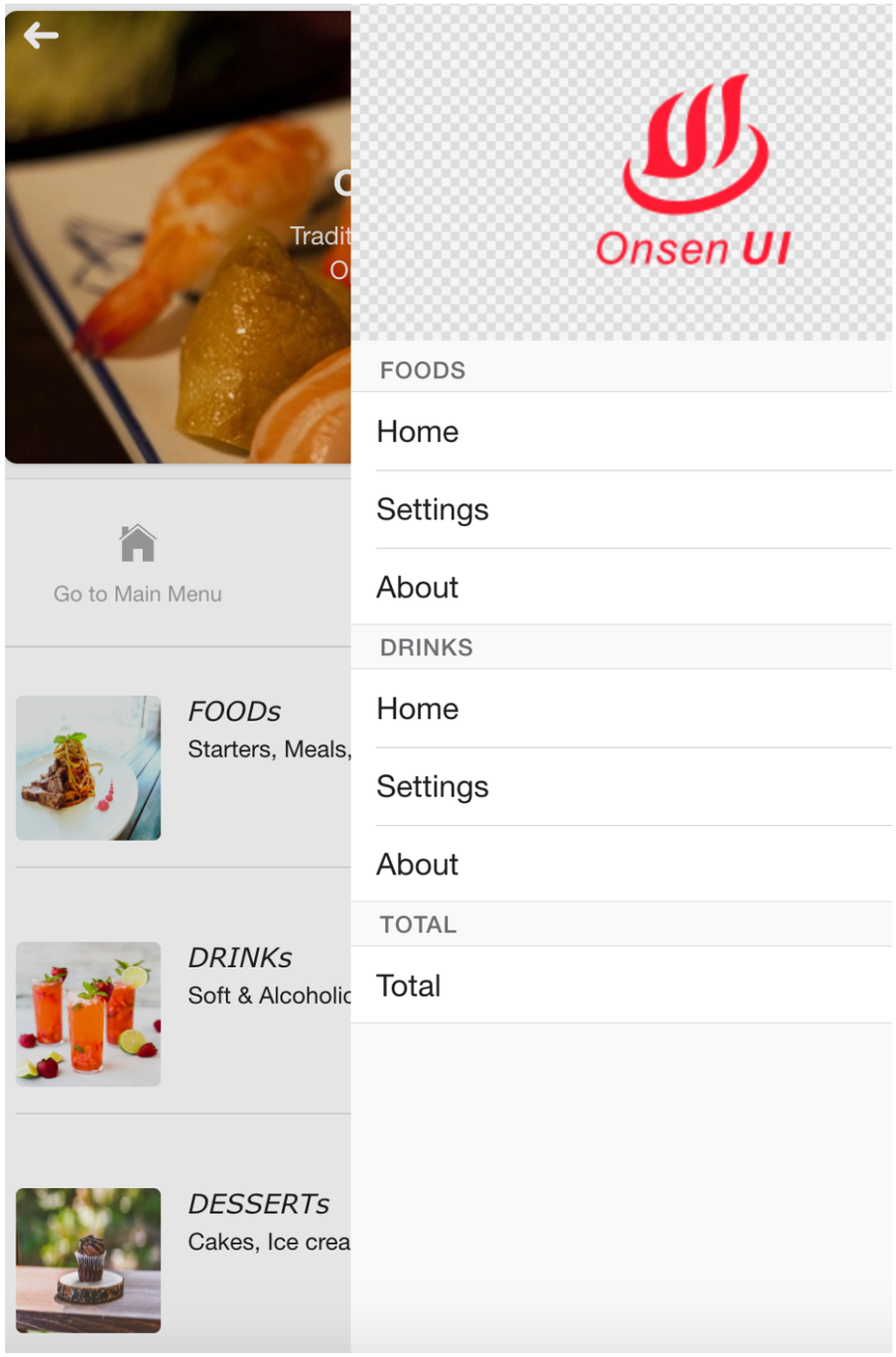
Login Process

1. User selects the Venue
2. MobileApp calls API
3. API returns the products/items of the Venue
4. Mobile APP saves the menu in order to not to call API again.
5. MobileApp displays the main menu (Foods, Drinks, Desserts, Deals)
6. On the main menu, user should be able to “select the table”, “pre-order” and “take away” options.
7. User must be able to see cart/basket at all times.
8. Cart/Basket must be collapsable item, so user will not lose where he/she was.



when pressed Cart/Basket display current items in the basket by main categories (Foods, Drinks,etc) and total.

It must be collapsable as seen below.So user will not lose the page he/she is on



CHAPTER 4

Venue Screens-2 (Foods,Drinks,etc)

This section describes
- Login process and messaging
- Registration process and messaging

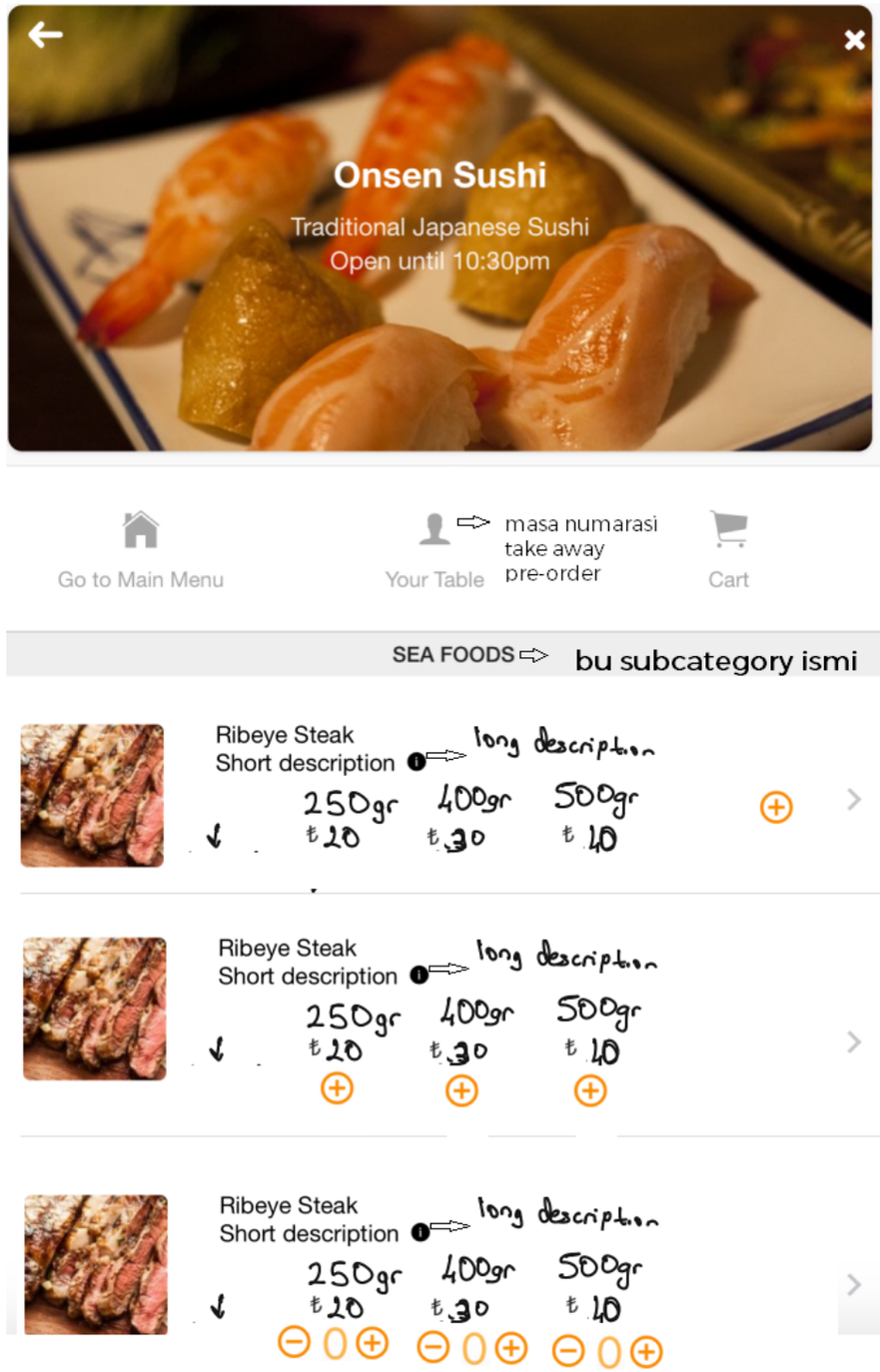
Below screen is only for sample purposes

Section 1: Food Menu

On the main menu, user selects “Foods”, then Foods menu is displayed

Login Process

1. Foods are displayed by Mobile App
2. MobileApp calls API
3. API returns the products/items of the Venue
4. Mobile APP saves the menu in order to not to call API again.
5. MobileApp displays the main menu (Foods, Drinks, Desserts, Deals)
6. On the main menu, user should be able to “select the table”, “pre-order” and “take away” options.
7. User must be able to see cart/basket at all times.
8. Cart/Basket must be collapsable item, so user will not lose where he/she was.
9. When listing, if ***inStock*** flag is false then display with a note (not available currently(

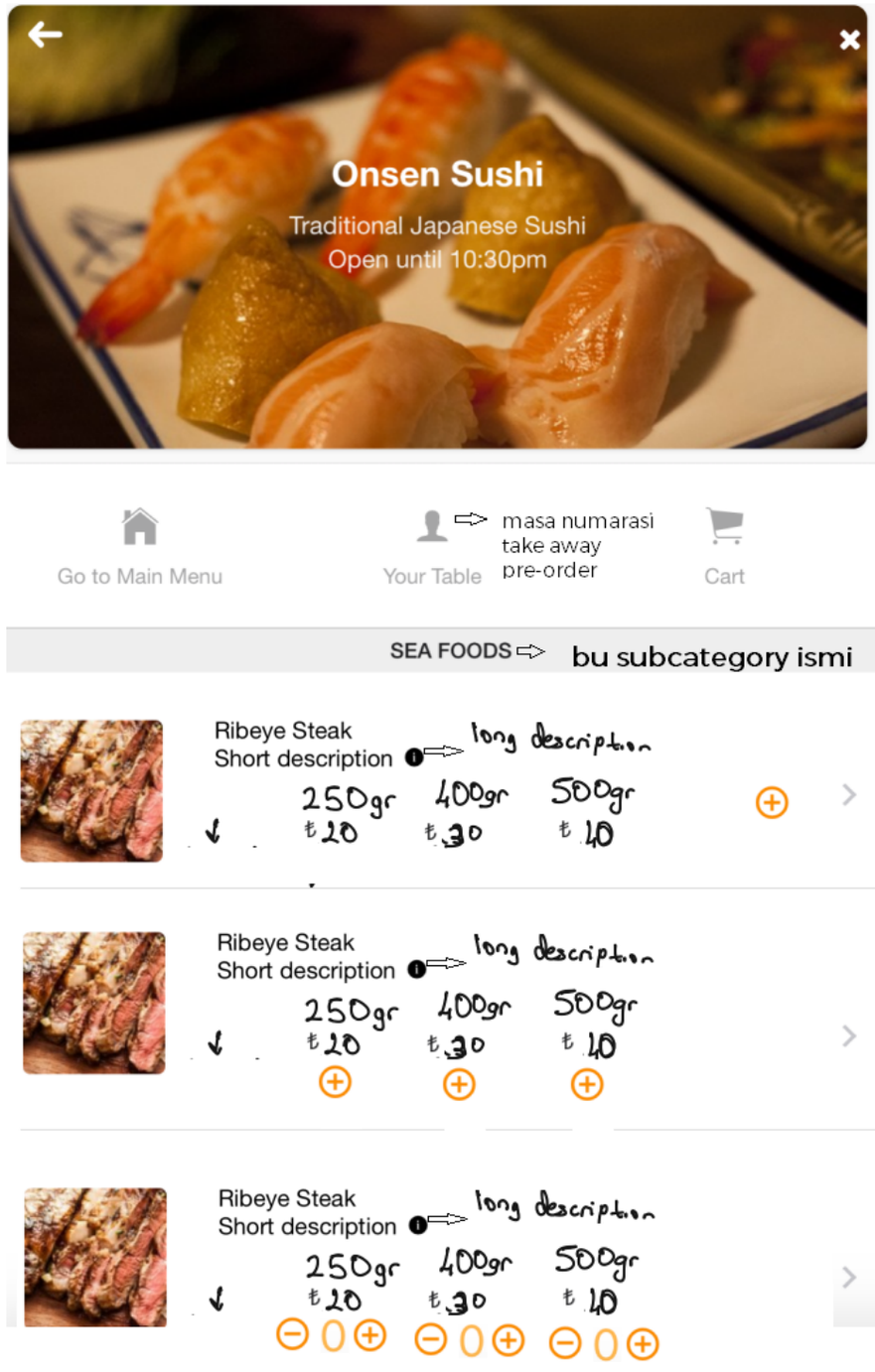


Section 2: Food Menu Details

On the main menu, user selects “Foods”, then Foods menu is displayed

Important ~Fields

1. inStock field is to highlight if that product is served at the moment. If it is not in stock, still display in the menu but put “not in stock at the moment”
2. sku - is a product Id
3.



Food Menu Details - 2

A product can have below options. Think about it for a design

Size Options (Minimum 1)

1. If there is a different size/price combination for the product. (example: 33cl ,50cl for beer)

Preference (min 0)

1. If there is a preference question. Example: Rare, Medium, Well-done for steak.
2. If there is a preference this should be asked

Add Ons (min 0)

1. These are the add ons available for particular product. (e.g : Ketchup, Mayo, etc for Burger)
2. It can be free, as well as with a fee

Suggestions (min 0)

1. If there is a different size/price combination for

Promotions (min 0)

1. Any promotion data such as ‘happy hour’ between 17.00 - 19.00 , or “Lunch Time”