Technical Documentation - MMS
Version: 1.0
Group 3
Apr 6, 2022

Table of Contents

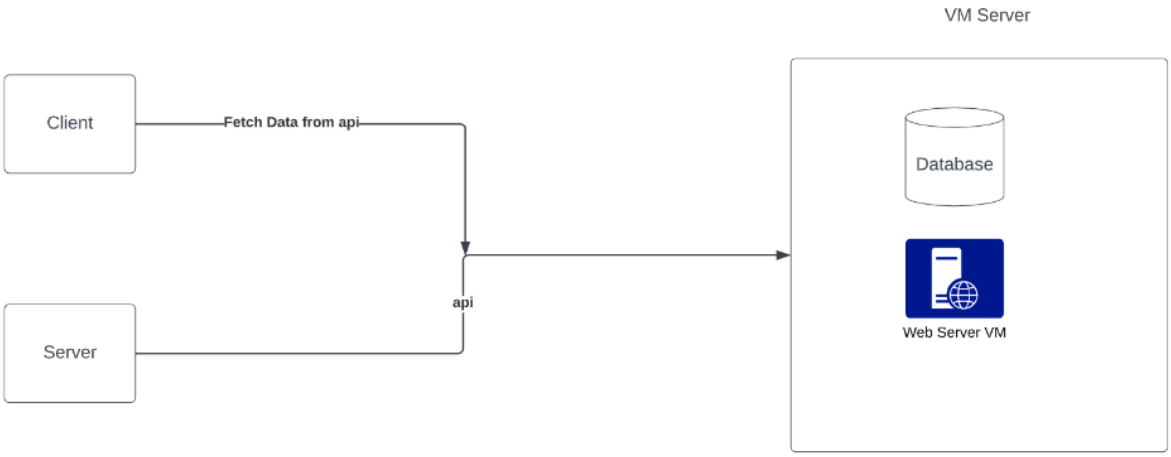
Overview

This document will provide crucial documentation of program code implementation of the platform at the level of the testing stage, production, and deployment. It will cover each module and component of the system.

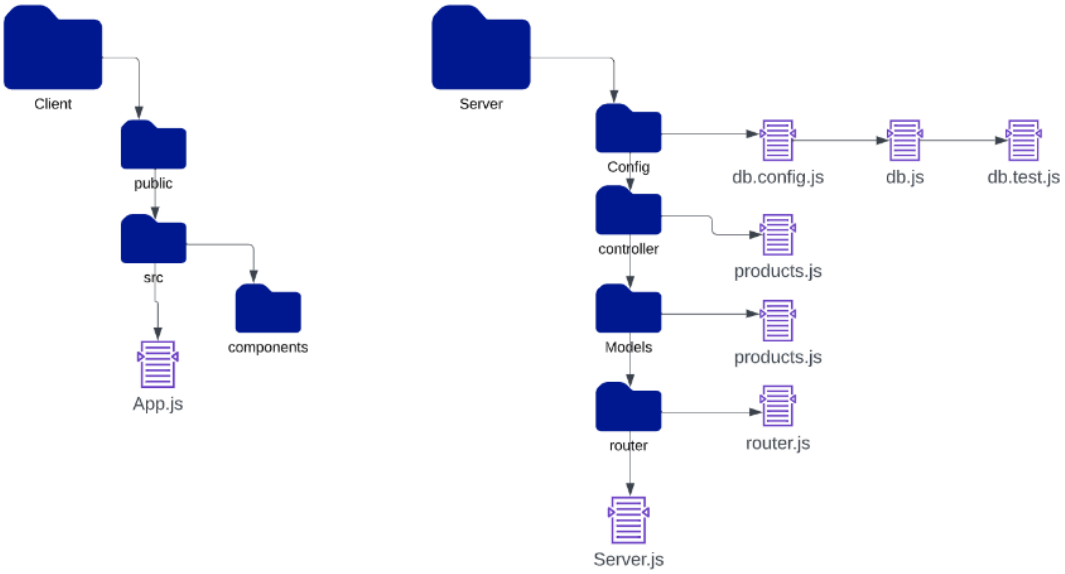
Objective

The main objective of this documentation is to provide the developer with critical information on the code implemented in the development of the system. It aims to facilitate the maintenance and expansion of the system.

Program Structure

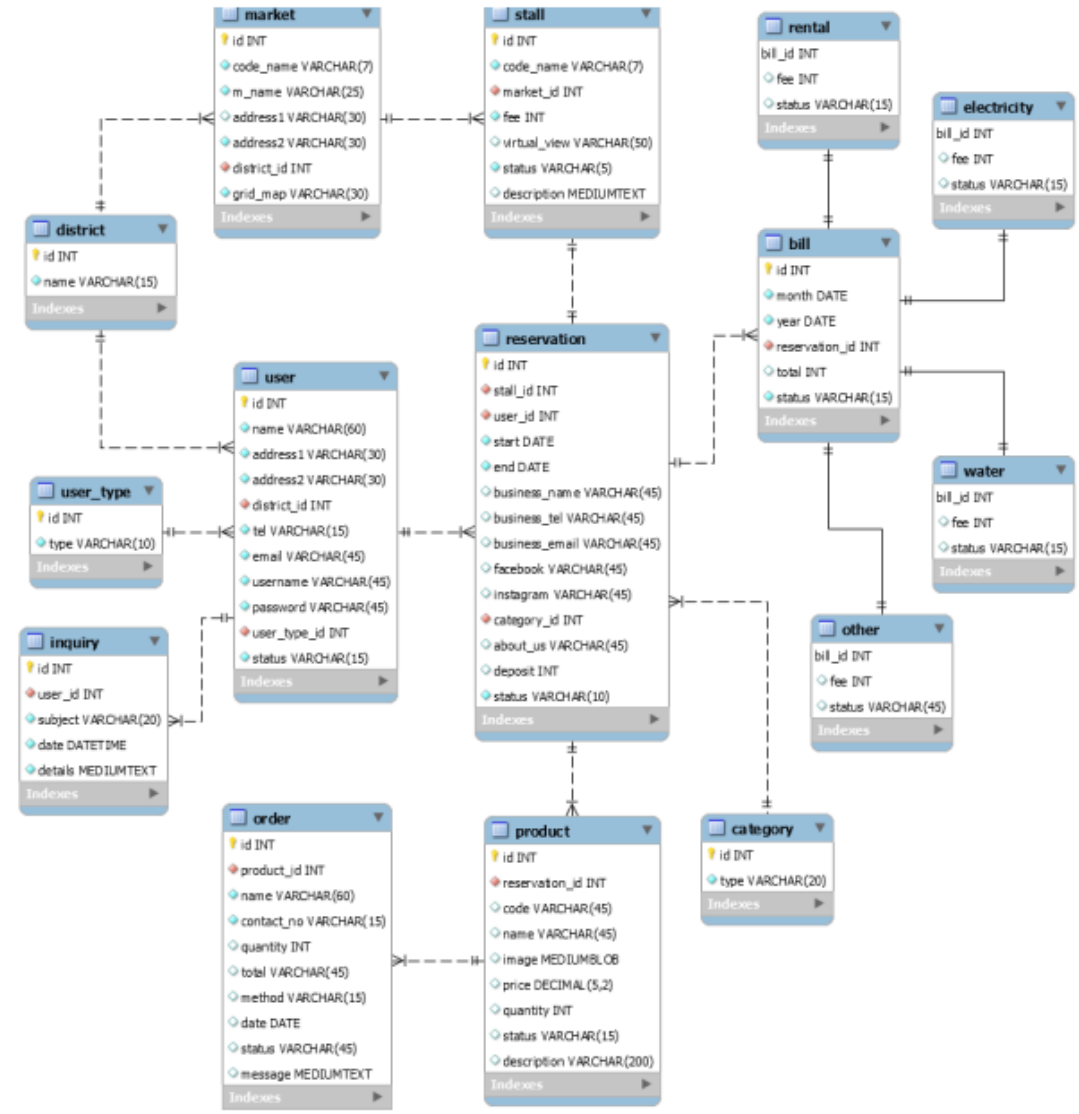


Program File Structurer



Database Design Schema

Database Schema:



Technologies

- Reactjs-Frontend Language
- Nodejs- Backend Language
- Mysql - Database
- Azure VM Ubuntu
- Nginx Server

Testing

Ten Uses cases were used for the testing phase for our system.

- TC1 - **UC-1: AuthVendorandManagement**
- TC2 - **UC-2: ViewMarketMap**
- TC3 - **UC-3: GetMarketRoute**
- TC4 - **UC-4: TourMarket**
- TC5 - **UC-5: TourStall**
- TC6 - **UC-7: BuyItem**
- TC7 - **UC-11: GenerateReport**
- TC8 - **UC-15: ApproveReservedStall**
- TC9 - **UC-18: GenerateInvoices**
- TC10 - **UC-19: ViewInquires**

Testing Framework:

- Jest

Example Demo of the testing Framework:

Let's get started by writing a test for a hypothetical function that adds two numbers. First, create a `sum.js` file:

```
function sum(a, b) {
  return a + b;
}
module.exports = sum;
```

Then, create a file named `sum.test.js`. This will contain our actual test:

```
const sum = require('./sum');

test('adds 1 + 2 to equal 3', () => {
  expect(sum(1, 2)).toBe(3);
});
```

Add the following section to your `package.json`:

```
{
  "scripts": {
    "test": "jest"
  }
}
```

Finally, run `yarn test` or `npm run test` and Jest will print this message:

```
PASS  ./sum.test.js
✓ adds 1 + 2 to equal 3 (5ms)
```

You just successfully wrote your first test using Jest!

Customer Testing Components:

Customer Class

```
const { getProductById } = require("./products");

class Customer {

    //UC-2: ViewMarketMap
    viewMarketMap(name){

        const market =  ['San Ignacio','Belmopan','Belize', 'Orange Walk']// list of markets



        if(market.includes(name)){

            return true
        }
        return false
    }
    //UC-3: GetMarketRoute
    getMarketRoute(clat,clong,dlat,dlong){

        if(clat == '17.1573248' && clong == '-89.0830848' && dlat == '17.1569824,' && dlong == '-89.0730774'){

            return true
        }

        return false
    }

    // UC-4: TourMarket
    tourMarket(name){

        //Avaliable Markets
        const market =  ['San Ignacio','Belmopan','Belize', 'Orange Walk']

        if(market.includes(name)){

            return true
        }
        return false

    }

    //UC-5: TourStall - function verify that stall exist for the tour stall
    tourStall(name){

        const stall =  ['C10','P15','VF30', 'F3']// stall list avaliable

        if(stall.includes(name)){

            return true
        }
        return false
    }

    //UC-7: BuyItem
    buyItem(productId, stallId, amount){

        //This Function will get us all the details of the product including availability
         const product = getProductById(productId);

        //Return fail if no product is found
         if(!product) { return false }

        //Checks if the product can be found in that stall
        const isItemAvailable = product.availableAt.includes(stallId);

        //return fail if not in the stall desired
        if(!isItemAvailable) { return false}

        //Finally return success
        return true;

    }
}

 module.exports=Customer
```

- The customer class has five public functions
- getProductById is import to be used on the buyitem function

TC2 - UC-2: ViewMarketMap

```
//UC-2: ViewMarketMap
viewMarketMap(name){

    const market =  ['San Ignacio','Belmopan','Belize', 'Orange Walk']// list of market

    if(market.includes(name)){//verify market name pass in is valid using market lists

        return true
    }

    return false
}
```

ViewMarketMap

- The function accepts string parameter
- Inside the function an array of market name is stated
- An if statement is use for the validation of the string pass in
- Return true if the string name pass in exist in the market names array
- Return false if the string pass in doesn't exist

TC3 - UC-3: GetMarketRoute

```
//UC-3: GetMarketRoute
getMarketRoute(clat,clong,dlat,dlong){

    if(clat == '17.1573248' && clong == '-89.0830848' && dlat == '17.1569824,' && dlong == '-89.0730774'){//validate current location and destination location coordinates

        return true
    }

    return false
}
```

GetMarketRoute

- The function accepts four string parameters(current location and destination location coordinates)
- An if statement ise use for the validations of the parameters pass in
- Return true if the parameters pass in match with coordinates in the function
- Return false if the parameters pass in doesn't match with coordinates in the function

TC4 - UC-4: TourMarket

```
// UC-4: TourMarket
tourMarket(name){

    //Avaliable Markets
    const market =  ['San Ignacio','Belmopan','Belize', 'Orange Walk']

    if(market.includes(name)){//validate market name with avaliable market list

        return true
    }

    return false

}
```

TourMarket

- The function accepts a string parameter
- Inside the function market name array is initialized
- An if statement is used to validate the string pass in
- Return true if the parameter pass in matches with any names of the array
- Return false if the parameter pass in doesn’t matches with any names of the array

TC5 - UC-5: TourStall

```
//UC-5: TourStall - function verify that stall exist for the tour stall
tourStall(name){

    const stall =  ['C10','P15','VF30', 'F3']// stall list avaliable

    if(stall.includes(name)){

        return true
    }
    return false
}
```

TourStall

- The function accepts a string parameter
- Inside the function stall array is initialized
- An if statement is used to validate the string pass in
- Return true if the parameter pass in matches with any names of the array
- Return false if the parameter pass in doesn’t matches with any names of the array

TC6 - UC-7: BuyItem

```
//UC-7: BuyItem
buyItem(productId, stallId, amount){

    //This Function will get us all the details of the product including availability
     const product = getProductById(productId);

    //Return fail if no product is found
     if(!product) { return false }

    //Checks if the product can be found in that stall
    const isItemAvailable = product.availableAt.includes(stallId);

    //return fail if not in the stall desired
    if(!isItemAvailable) { return false}

    //Finally return success
    return true;

}
```

BuyItem

- The function accepts three string as parameters
- getProductById is call and productId is pass in to get the specific product
- The first if statement validate if the product exist
- isItemAvailable is call and stalled is pass in to get verify if the product is available
- The second if statement validate if the item is available
- Return true if both if statement return true

Authentication

TC1 - UC-1: AuthVendorandManagement

```
//UC-1: AuthVendorandManagement

const user = require('./user')// object with valuid users

function authUser(uname, pwd, utype){//the function authUser will verify if user credentials exist in the system - username, password, usetype(management,vendor)


    if(user.users.some(data => data.username === uname && data.password === pwd && data.uType === utype)){ //verify that parameter pass in exist
        return true
    }
    return false

}

module.exports=authUser
```

```
1   //object for users
2   const user = {
3       users:[
4
5           {
6
7               username: "ilopez",
8               password: "%34k",
9               uType: 'vendor'
0           },
1
2           {
3
4               username: "rRamos",
5               password: "%34k",
6               uType: 'management'
7           },
8
9       ]
0   }
1
2   module.exports=user
```

AuthVendorAndManagement:

- User object is declare
- The function accepts three string parameters
- An if statement validate the parameters pass in
- Return true if parameters matches with any data inside user object
- Return false if parameters doesn’t matches with any data inside user object

Vendor Component:

Vendor Class

```
//Class Vendor - Includes all functions related to Vendor
class Vendor{

    generateReport(reportType, authUser){// accept two parameters reportType and authUser

        const reports = [ 'Monthly Expense', 'Pending Invoices', 'Yearly Expenses']//List of type of reports

        if(reports.includes(reportType) && authUser == true){

            return true
        }

        return false


    }
}

module.exports =Vendor
```

- The Vendor class has one function

TC7 - UC-11: GenerateReport

```
generateReport(reportType, authUser){// accept two parameters reportType and authUser

    const reports = [ 'Monthly Expense', 'Pending Invoices', 'Yearly Expenses']//List of type of reports

    if(reports.includes(reportType) && authUser == true){

        return true
    }

    return false


}
```

Generate Report:

- The function accepts two parameters(string and boolean)
- Inside the function type of reports array is initialized
- An If statement is used to validate the parameter pass in with the array
- Return True if parameter matches or Return false if parameter does not matches

Management Component:

Management Class

```
// Management class - include all function relate to management include a
//const auth = require('./auth')
class Management{


    //Function for approveReserveStall - verify that the stall is avaliable, vendor has exist and also authenicate management user
    approveReserveStall(vendorId, stallId, authUser){

        const stall =  ['C10','P15','VF30', 'F3']// list of stall
        const vendor = ['v56', 'v34', 'v90', 'v32']// list of vendor account by their id

        if(vendor.includes(vendorId) && stall.includes(stallId) && authUser == true){

            return true
        }

            return false
    }

    //Function to generate Invoice - Verify that vendor and stall exist. Authenicate user management account
    generateInvoice(vendorId, stallId, authUser){

        const stall =  ['C10','P15','VF30', 'F3']// list of stall
        const vendor = ['v56', 'v34', 'v90', 'v32']// list of vendor accounts

        if(vendor.includes(vendorId) && stall.includes(stallId) && authUser == true){

            return true
        }

        return false

    }

    //Function to view inquires - authenicate user management account to view inquires
    viewInquires(authUser){

        if(authUser == true){

            return true
        }

        return false
    }
}

module.exports = Management
```

The Management Class has three function

TC8 - UC-15: ApproveReservedStall

```
//Function for approveReserveStall - verify that the stall is avaliable, vendor has exist and also authenicate management user
approveReserveStall(vendorId, stallId, authUser){

    const stall =  ['C10','P15','VF30', 'F3']// list of stall
    const vendor = ['v56', 'v34', 'v90', 'v32']// list of vendor account by their id

    if(vendor.includes(vendorId) && stall.includes(stallId) && authUser == true){

        return true
    }

        return false
}
```

ApproveReserveStall

- The function accepts three parameters(two string and one boolean)
- Inside the function two array are initialized one for stall and vendor
- An if statement is used to validate the parameters
- Return true if the parameter matches the criteria
- Return false if the parameter doesn't matches the criteria

TC9 - UC-18: GenerateInvoices

```
//Function to generate Invoice - Verify that vendor and stall exist. Authenicate user management account
generateInvoice(vendorId, stallId, authUser){

    const stall =  ['C10','P15','VF30', 'F3']// list of stall
    const vendor = ['v56', 'v34', 'v90', 'v32']// list of vendor accounts

    if(vendor.includes(vendorId) && stall.includes(stallId) && authUser == true){

        return true
    }

    return false

}
```

Generate Invoice

- The function accepts three parameters(two string and one boolean)
- Inside the function two array are initialized one for stall and vendor
- An if statement is used to validate the parameters
- Return true if the parameter matches the criteria
- Return false if the parameter doesn't matches the criteria

TC10 - UC-19: ViewInquires

```
//Function to view inquires - authenicate user management account to view inquires
viewInquires(authUser){

    if(authUser == true){

        return true
    }

    return false
}
```

View Inquires

- The function accepts one parameter(boolean)
- An if statement is used to validate the parameter pass in
- Return true if the parameter matches the criteria
- Return false if the parameter doesn’t matches the criteria

MMS

- *To be implemented for demo 2*

Frontend

Components:

Layout

Vendor

Customer

Management

Backend

Server.js

Routes

Controller

Models