App Feature: Intro to the CRM App

Relevel
by Unacademy

Topics to be Covered

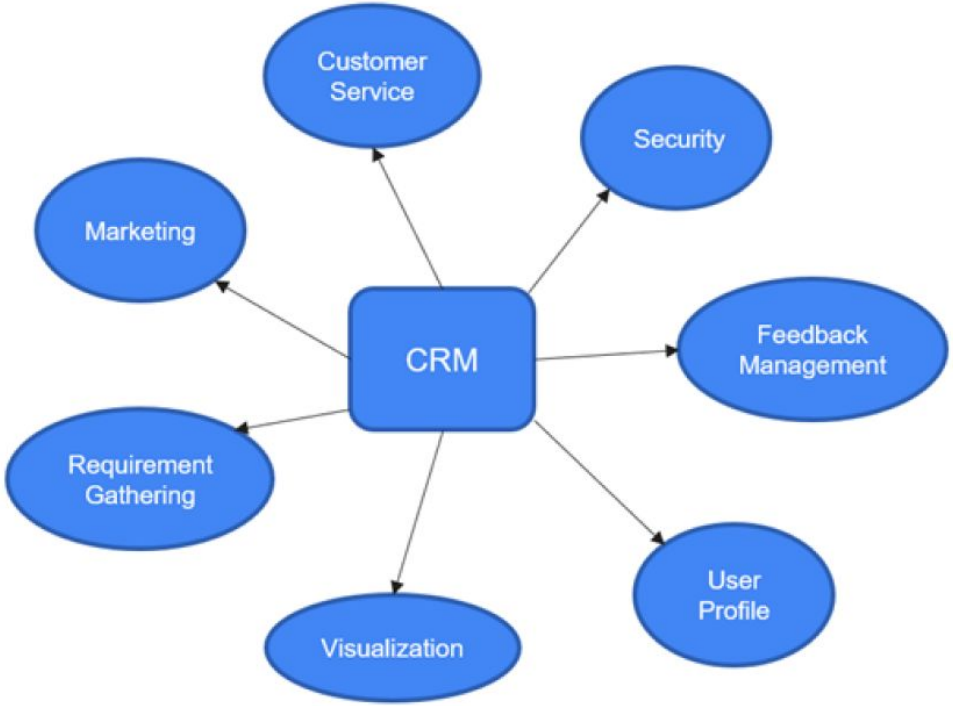
- Requirement Gathering
- Requirement Analysis
- Taking decisions on tech stack to be used for the CRM app

What is CRM?

CRM i.e. Customer Relationship Management refers to creating, developing, and maintaining a relationship with customers. This can be achieved in multiple ways -

- Marketing of assets
- Providing Service to the customers
- Delivery with profit and so on

1. **Customer Service** - This includes providing service to the customers which can help them to keep track of their activities in the application. This also includes providing an interface that is more convenient to them.
2. **Marketing** - This involves the implementation of efficient strategies to sell the product after recognizing the potential customers.
3. **Requirement gathering** - This involves the expectations of customers from the application.
4. **Visualization** -> This involves representing data in form of visuals like charts, tables, etc to do the analysis on the strategy involved
5. **User profile** - This involves maintaining and updating user data which can be further used to communicate with them to provide better service
6. **Feedback Management** - This includes gathering feedback and complaints from customers so that in the future, we can provide them with better service and solution
7. **Security** - This involves securing and maintaining user data and integrity


Customer Service
Security
Marketing
CRM
Feedback Management
Requirement Gathering
User Profile
Visualization

Use of CRM

- Identify potential customers
- Increase the number of customers
- Maintain profitable relationships with customers
- Provide insights into the customers

Problem Statement

In our upcoming sessions, we will focus on developing a CRM application feature-wise. Our main problem statement is - “Create a CRM application that can be leveraged to accept the customer complaints and provide the complete life cycle management of the issues raised by the customers.”

We need to create a system which would handle the complaints logged by the customer.
The complaints will be logged as a ticket and they will be fixed by the engineer. We need to manage this complete life-cycle using our application.
Admin can keep track of the complete life cycle of the issue from logging of complaints as a ticket, fixing and updating the ticket.

Actors

Customer

I should be able to register myself

I should be able to login myself for registering/viewing complains

I should be able to raise an issue

I should be able to check the latest status of the issues I raised

I should be able to modify the issue raised

I should be able to check the complete history of the issues raised

I should be able to close my ticket myself

Actors

Engineer

I should be able to accept an issue

I should be able to update an issue

I should be able to close an issue

I should be able to see the complete list of issues assigned to me

I should be able to search for an issue

I should be able to filter the issues assigned to me based on the creation date

Actors

Admin

I should be able to see all the customer's non-PII details

I should be able to see all the Engineers

I should be able to see all the tickets details

I should be able to see all the active tickets

I should be able to filter the tickets based on status | date, etc

I should be able to re-assign a ticket to another Engineer

I should be able to add a new Engineer

I should be able to remove an Engineer

Tech Stack

Node.js -

We are going to use the Node.js environment for our application.

JavaScript used to only work inside the browsers. For running JS code, browsers use the JavaScript engine.

NodeJS is a JavaScript runtime environment that allows us to run JS code outside of a web browser. To achieve this, NodeJS uses Chrome's V8 Engine which is open source(free to use) and very performant. A Node.js app runs in a single process, without creating a new thread for every request.

NodeJS has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

Express Framework -

Express is a Node.js web application framework that provides multiple features to develop web and mobile applications.

Features –

1) Middleware between requests
2) Routing Concept
3) Dynamic Operations
4) Asynchronous Operations

Tech Stack

MongoDB -

We will use MongoDB as our database. MongoDB is a document-oriented database that is non-structured. We can have a dynamic schema that can be scaled easily. It stores data in the form of documents that are in JSON format that are scalable and easier to maintain without keeping some specific constant structure.

Mongoose -

We will use Mongoose as our object modeling tool. It is used to do object mapping between Nodejs and MongoDB. We can use it as an interface to the database which can be used to create, update, delete and query the records.

Application

In our next sessions, we would be building our application feature wise. Let's have a look on all those features

Let’s have a look at different features of a CRM application one by one. We are going to build these features from scratch in our upcoming session

Feature 1 : User Authentication Authorization APIs

Three kinds of users :

Customer
Engineer
Admin

- Engineer/Admin registration will be supported through API, but it needs to be approved by the ADMIN
- Customer registration will be supported through API with no approval needed from the ADMIN
- API to support the ADMIN login. Login API call should return the access token, which will be used to make all the other calls
- API to support the CUSTOMER login. Login API call should return the access token, which will be used to make all the other calls
- API to support the ENGINEER login. Login API call should return the access token, which will be used to make all the other calls.
- Login API will succeed only if the ENGINEER registration request has been approved by the ADMIN. Proper error message in the case ADMIN has yet not approved/rejected the registration request

SignUp request

POST /crm/api/v1/auth/signup

Sample request body :

```
{
        "name": "Vishwa",
        "userId": "Vish07",
        "email" : "abc@xyz.com",
        "password" : "Welcome1",
        "userType" : "ENGINEER"
}
```

Sample response body :

```
{
    "name": "Vishwa",
    "userId": "Vish07",
    "email": "abc@xyz.com",
    "userTypes": "ENGINEER",
    "userStatus": "PENDING",
    "createdAt": "2022-02-20T04:47:43.842Z",
    "updatedAt": "2022-02-20T04:47:43.842Z"
}
```

Login Request

POST /crm/api/v1/auth/signin

Sample request body :


```
{
        "userId": "Vish01",
        "password" : "Welcome1"

}
```


Sample response body :


```
{
    "name": "Vishwa",
    "userId": "Vish01",
    "email": "abc@xyz1.com",
    "userTypes": "CUSTOMER",
    "userStatus": "APPROVED",
    "accessToken":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IlZpc2gwMSIsImlhdCI6MTY0NTMzMjg3NiwiZXhwIjoxNjQ1NDE5Mjc2fQ.21lRt9VIL-suvP7Z_lamH1PcchOB1TJOhZPSpX9kqt8"
}
```

Feature 2 : Authenticating and Authorizing the User APIs

- API for getting the list of all users
- API for the getting the user based on User ID
- API for updating the user type and status
- Authenticating and Authorizing above APIs, so that only authenticated ADMIN will be allowed to perform the above operations
- ENGINEER/ADMIN user should be able to login successfully after the approval from ADMIN user

Get all users

GET /crm/api/v1/users/
Headers :
Content-Type:application/json
x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Sample response body :


```
[ {
        "name": "Vishwa",
        "userId": "admin",
        "email": "kankvish@gmail.com",
        "userTypes": "ADMIN",
        "userStatus": "APPROVED"
    },
```

```
{
    "name": "Mohan",
    "userId": "Mohan01",
    "email": "abc@mohan.com",
    "userTypes": "CUSTOMER",
    "userStatus": "APPROVED"
}]
```

Get user based on the userId

GET /crm/api/v1/users/{userId}

Headers :
Content-Type:application/json
x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Sample response body :

```
[
   {
      "name": "Vishwa",
      "userId": "Vish01",
      "email": "abc@xyz1.com",
      "userTypes": "ENGINEER",
      "userStatus": "APPROVED"
   }
]
```

Update the user information- Type and Status

PUT localhost:8080/crm/api/v1/users/Vish01

Headers :

Content-Type:application/json

x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Sample request body :

```
{
        "name": "Vishwa",
        "userId": "Vish01",
        "email": "abc@xyz1.com",
        "userTypes": "ENGINEER",
        "userStatus": "APPROVED"
}
```

Sample response body :

```
{
    "message": "User record has been updated successfully"
}
```

Feature 3 : Ticket Creation

- API for the authenticated user to raise a request
- API for the authenticated user to update an existing request
- API for an authenticated user to check the status of the request
- API for an authenticated user to check the list of the all the requests raised so far
- API for the authenticated user to raise a request
- Registered Engineer if any should be assigned the ticket automatically

Create a new ticket

POST /crm/api/v1/tickets/

Headers :

Content-Type:application/json

x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Sample request body :

```
{
    "title": "Not able to use the device",
    "description" : "Device is not turning on with power"
}
```

Sample response body :


```
{
    "title": "Not able to use the device",
    "ticketPriority": 4,
    "description": "Device is not turning on with power",
    "status": "OPEN",
    "reporter": "Vish01",
    "assignee": "Vish07",
    "id": "6215d8288d78a94e0a5a0610",
    "createdAt": "2022-02-23T06:46:00.414Z",
    "updatedAt": "2022-02-23T06:46:00.414Z"
}
```


Get all tickets

GET /crm/api/v1/tickets/

Headers :

Content-Type:application/json

x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Sample response body :


```
[ { "title": "Not Able to update",
        "ticketPriority": 4,
        "description": "Update functionality is not working for my device",
        "status": "OPEN",
        "reporter": "Vish01",
        "assignee": "Vish07",
        "id": "6215c1e85e0d5a53afcd4e68",
        "createdAt": "2022-02-23T05:11:04.841Z",
        "updatedAt": "2022-02-23T05:11:04.841Z"
    },
    { "title": "Not able to use the device",
        "ticketPriority": 4,
        "description": "Device is not turning on with power yessss",
        "status": "CLOSED",
        "reporter": "Vish01",
        "assignee": "Vish07",
        "id": "6215ceb86ba4fcbac9433282",
        "createdAt": "2022-02-23T06:05:44.352Z",
        "updatedAt": "2022-02-23T06:05:44.353Z"
    },
```

```
  { "title": "Not able to use the device",
    "ticketPriority": 4,
    "description": "Device is not turning on with power",
    "status": "OPEN",
    "reporter": "Vish01",
    "assignee": "Vish07",
    "id": "6215d8288d78a94e0a5a0610",
    "createdAt": "2022-02-23T06:46:00.414Z",
    "updatedAt": "2022-02-23T06:46:00.414Z"
  }
]
```


Get all tickets fitered by status

GET /crm/api/v1/tickets?status=OPEN

Headers :
Content-Type:application/json
x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Response :


```
[
    {
        "title": "Not Able to update",
        "ticketPriority": 4,
        "description": "Update functionality is not working for my device",
        "status": "OPEN",
        "reporter": "Vish01",
        "assignee": "Vish07",
        "id": "6215c1e85e0d5a53afcd4e68",
        "createdAt": "2022-02-23T05:11:04.841Z",
        "updatedAt": "2022-02-23T05:11:04.841Z"
    },
```

```
    {
        "title": "Not able to use the device",
        "ticketPriority": 4,
        "description": "Device is not turning on with power",
        "status": "OPEN",
        "reporter": "Vish01",
        "assignee": "Vish07",
        "id": "6215d8288d78a94e0a5a0610",
        "createdAt": "2022-02-23T06:46:00.414Z",
        "updatedAt": "2022-02-23T06:46:00.414Z"
    }
]
```

Feature 4 : Ticket Manipulations and ADMIN capabilities

- API for authenticated Engineer to update the ticket
- Updated ticket should be visible to the customers immediately
- API for authenticated Engineer to search for the ticket
- API for authenticated Engineer to be able to accept a ticket
- API for authenticated Engineer to be able to see the complete list of tickets assigned to him/her
- API for the authenticated ADMIN to get the list of all the customers
- API for the authenticated ADMIN to get the list of all the issues
- API for the authenticated ADMIN to get the list of all the issues after applying certain filters
- API for the authenticated ADMIN to get the list of all the active issues
- API for the authenticated ADMIN to get the list of all the ENGINEER registration requests
- API for the authenticated ADMIN to Accept/reject the ENGINEER registration requests

Update the given ticket (Only the creator of the ticket should be able to update it)

PUT /crm/api/v1/tickets/6215c1e85e0d5a53afcd4e68

Headers :

Content-Type:application/json

x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Request body :

```
{
        "title": "Not Able to update",
        "ticketPriority": 4,
        "description": "Update functionality is not working for my device",
        "status": "CLOSED"

}
```

Response body :


```
{
    "title": "Not Able to update",
    "ticketPriority": 4,
    "description": "Update functionality is not working for my device",
    "status": "CLOSED",
    "reporter": "Vish01",
    "assignee": "Vish07",
    "id": "6215c1e85e0d5a53afcd4e68",
    "createdAt": "2022-02-23T05:11:04.841Z",
    "updatedAt": "2022-02-23T05:11:04.841Z"
}
```

Get the ticket based on the ticket id

GET /crm/api/v1/tickets/{id}

Headers :

Content-Type:application/json

x-access-token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImFkbWluIiwiaWF0IjoxNjQ1NTA4NDY0LCJleHAiOjE2NDU1OTQ4NjR9.PgKiGRN_J8aDGwrBLOGhWUKArcfegDd76dEgGtV6Qh0

Response :


```
{
        "title": "Not Able to update",
        "ticketPriority": 4,
        "description": "Update functionality is not working for my device",
        "status": "OPEN",
        "reporter": "Vish01",
        "assignee": "Vish07",
        "id": "6215c1e85e0d5a53afcd4e68",
        "createdAt": "2022-02-23T05:11:04.841Z",
        "updatedAt": "2022-02-23T05:11:04.841Z"
}
```

Feature 5 : Creation of the Notification Service

- API for raising the notification request
- API to get the result of the notification request
- Scheduled job to regularly check for any new request and then send email notifications
- to every one listed

Feature 6 : Integrating the Notification Service with CRM application

CRM APP should call Notification service, every time a new ticket is created or an existing ticket is updated.

Practice Code

- Mention advantages of CRM application along with the use case
- Mention few real-life examples of CRM
- Write a GET API to fetch users list present in database

MCQ

1. Which command can be used to install express?

A. npm install express [Correct Answer]
B. Install express
C. npm express
D. None

2. Which module can be categorized as part of CRM?

A. Feedback Management
B. Marketing
C. Requirement Gathering
D. All of the above [Correct Answer]

3. Which functionality can be included as part of CRM application?

A. Get User list
B. Update User profile
C. Authenticate User
D. All of the above [Correct Answer]

4. What are the roles of Engineers in CRM Applications?

A. Create an issue

B. Update an issue

C. Fix an issue

D. All of the above [Correct Answer]

5. Which of these is not an advantage of CRM?

A. CRM can identify potential customers

B. CRM can increase the number of customers

C. CRM maintain a profitable relationship with customers

D. CRM requires huge cost management [Correct Answer]

Thank You!