# INDEX

|  |
| --- |
| Problem Statement |
| Proposed Wireframe |
| Database Structure |
| Cloud Deployment Strategy |
| Frontend Screen List |
| Frontend Screenshot Reference |

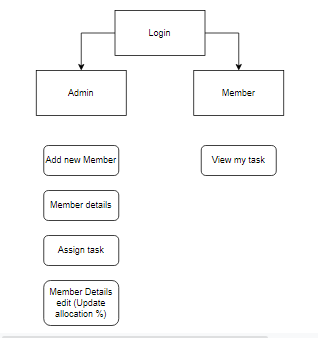
## PROBLEM STATEMENT

Project Management Tracker Application is microservice based Cloud Native Application. The Main purpose of Project Management Tracker is to provide the ability to add team members with skill set, assign a task and allocate them to the project:

* Put a new member to the team.
* Assign the task to each member.
* Update the allocation percentage.
* Enlisting of assigned task for each member
* Enlisting of all members from a team along with complete profile details.

The scope includes developing the application using tool chain mentioned below.

## PROPOSED APPLICATON WIREFRAME



# Database Structure

Table Name: memberdetails

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Member\_ID | Allocation\_percentage | Description | Member\_Name | Project\_end\_date | Project\_name | Project\_start\_date | Skillset | Total\_Exp |

Table Name: taskdetails

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Member\_ID | Deliverables | Member\_Name | Task\_End\_Date | Task\_Name | Task\_Start\_Date |

## REST API ENDPOINTS EXPOSED

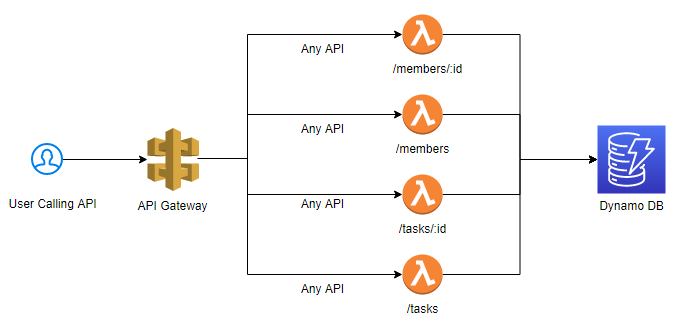
Backend Base URL : https://4yvcvf6lv2.execute-api.us-east-1.amazonaws.com

|  |  |  |
| --- | --- | --- |
| Operation | Endpoint | Type |
| Fetches details of all the team members | /members | Get |
| Fetch details of a team member | /members/:id | Get |
| Add Member | /members | Post |
| Assigns a task to each member of the team | /tasks | Post |
| View the assigned task | /tasks/:id | Get |
| Updates the allocation percentage | /members | Put |

## Angular Features Implemented

* HTTP Interceptor for API error handling and jwt token
* Common Services for calling APIs
* Components for defining business logic
* HTML, CSS for template definition and styling
* Angular Material for tables
* Reactive form for form control and validation
* Auth guards
* Basic routing

Cloud Wireframe



# Cloud Deployment Strategy:

## UI:

### AWS Services Required:

S3

### Steps:

* Provision S3 Bucket
* Enable static website hosting on S3 Bucket properties
* Make the items public
* Change policy to allow get item from outside access
* Build the application
* Upload /dist folder in the S3 bucket

## Backend:

### AWS Services Required:

* S3
* Lambda
* IAM
* API Gateway

### Steps:

* Install Serverless
* Run serverless at desired location
* Select correct template
* Make changes to handler function and serverless.yaml file
* Run serverless deploy to deploy the application
* Create relevant tables in Dynamo DB
* Update IAM role of generated lambda function to allow access to Dynamo DB

# Frontend Screen List

**Login**

**Manager Homepage**

**Add new member :** Add new member to the project

Constraints:

* Only if the number of experiences Only API to be developed is greater than 4, the member can be part of this project
* Member should possess at least 3 skillsets

Validations:

* All the fields are mandatory
* Project end date should be greater than project start date
* Allocation should be provided as percentage

**Member Details:** Show list of all members

Considerations:

* Fetch all the complete profile details of the team members
* Sort the list in the descending order of number of experiences

**Assign tasks:** Assign task to member

Constraints:

* If task end date is greater than Project End date, then custom exception should be thrown
* Task End date should be greater than task start date

**Member Details Edit:** Edit allocation percentage

Considerations:

* While fetching the task details the API must return the project details also (Project Start Date, Project End date, Allocation percentage)

**User Homepage**

**View my tasks:** View task details by Id

Constraints:

* If project end date is lesser than the current date, then the allocation percentage must be updated as 0.
* If the project end date is greater than the current date, then the allocation percentage must be 100%

# Frontend Screenshot for reference

