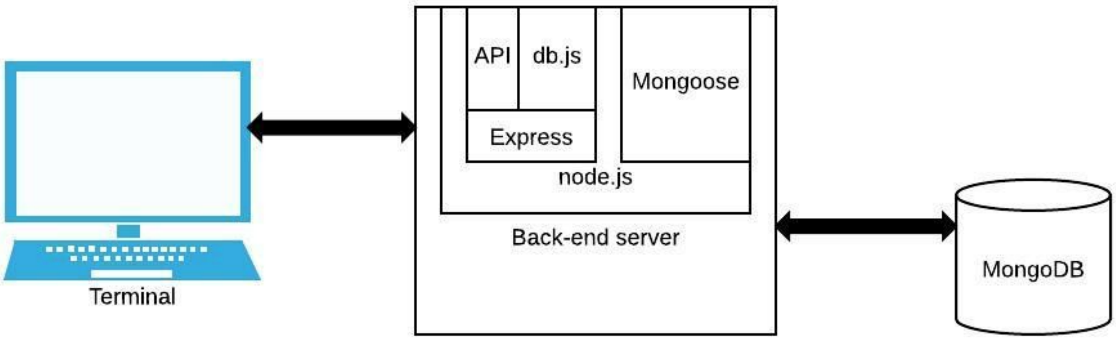
Service Layer Design

For

Project Daily Report

Student: Parisa Nikoukar

ProjectDailyReport app will be developed by Express.js framework and MongoDB as the database, the following image gives the whole idea about this application.



One of the most popular use-cases for Node.js is to write RESTful APIs using it.

RESTful API is the main focus for describing service endpoints in this application. The following HTTP verbs will be used for RESTful services for this context:

- POST: This method is used to create new reports in the app
- GET: This method is used to retrieve the list of reports based on their reporter or project name

Identify all service endpoints required for your MVP includes HTTP verbs, query parameters, path parameters, messages structures, and error codes

1.Endpoint: http://www.projectdailyreport.com/createDailyReport

HTTP verb: POST => By using this HTTP verb a new report is added to the list of daily reports **Path parameter:** /createDailyReport **Path query:** /createDailyReport?action = create	
Name	**Description**

body * required	{
	reporter_id: 100
	project_id: 50,
	completedTask: “Corrosion evaluation of columns”,
	tasksInProgress: “Approach Eastern side columns and evaluate above water portion”,
	Obstacles: “thunder storm’,
	estimatedProjectedDate: ‘12/29/2018’,
	nextTask: ‘Bridge surface structure’
	}

Parameter content type:	application\json
Responses	**Description**
201	The request has been fulfilled and the new daily report being created
400	The request cannot be fulfilled due to missing data

2. Endpoint: http://www.projectdailyreport.com
This Endpoint is going to be the root end point

HTTP verb: GET => By using this HTTP verb, the daily reports are searched by their reporter name **Path parameter:** https://www.projectdailyreport.com => homepage **Path query:** /?reporter_id = 100 => finds all the daily reports related to reporter with reporter_id of 100	
Parameters	
Name	*Description*
reporter_id*required integer($int64) Parameter content type:	the id of the reporter to return application\json
Responses	**Description**

200	Successful operation
	
	Example:
	{
	reporter_id: 100
	project_id: 50,
	completedTask: “Corrosion evaluation of columns”,
	tasksInProgress: “Approach Eastern side columns and
	evaluate above water portion”,
	Obstacles: “thunder storm’,
	estimatedProjectedDate: ‘12/29/2018’,
	nextTask: ‘Bridge surface structure’
	}

404	No report was found
400	Invalid ID supplied

HTTP verb: GET => By using this HTTP verb, the daily reports are searched by their project name
Path parameter: http://www.projectdailyreport.com => homepage

Path query: /?project_id = 50 => finds all the daily reports related to reporter with project_id of 50

Parameters

Name	*Description*
project_id*required integer($int64)	the id of the project to return
Parameter content type:	application\json

Responses	**Description**

200	Successful operation
	
	Example:
	{
	reporter_id: 100
	project_id: 50,
	completedTask: “Corrosion evaluation of columns”,
	tasksInProgress: “Approach Eastern side columns and evaluate above water portion’,
	Obstacles: “thunder storm’,
	estimatedProjectedDate: ‘12/29/2018’,
	nextTask: ‘Bridge surface structure’
	}

404	No report was found
400	Invalid ID supplied

3. **Stretch feature:** Endpoint: http://www.projectdailyreport.com/newUser

HTTP verb: POST => By using this HTTP verb a new user is added to the list of users

Path parameter: /newUser

Path query: /newUser?action = create

Name	**Description**

body * required	{
	reporter_id: 105
	FirstName: ‘Martha’
	LastName: ‘Anderson’
	Department:’Sales’
	Email:’MarthaAnderson@example.com
	}

Parameter content type:	application\json

Responses	**Description**
201	The request has been fulfilled and the new user being created
400	The request cannot be fulfilled due to missing data

4. **Stretch feature:** Endpoint: http://www.projectdailyreport.com/newProject

HTTP verb: POST => By using this HTTP verb a new project is added to the list of projects **Path parameter:** /newProject **Path query:** /newProject?action = create	
Name	**Description**

body * required	{
	project_id: 110
	ProjectName: ‘Solar system analysis’
	ProjectManager: ‘Gary Hunt’
	StartDate:’11/29/18’
	EndDate:’12/20/18’
	}

Parameter content type:	application\json
Responses	**Description**
201	The request has been fulfilled and the new project being created
400	The request cannot be fulfilled due to missing data

Describes the purpose of each endpoint based on the goals of a user persona as it relates to their data.

- http://www.projectdailyreport.com/createDailyReport

This endpoint is for the personas of the application who want to create new daily reports. For example, a persona like Emily use this endpoint to create her daily report and submit it which is done by POST http verb, her report will be added to the list of daily reports. If she does not enter data some of the required fields, the saving operation will be unsuccessful and there will be response of 400 that there is a missing data and creating the daily report should be repeated.

- http://www.projectdailyreport.com/

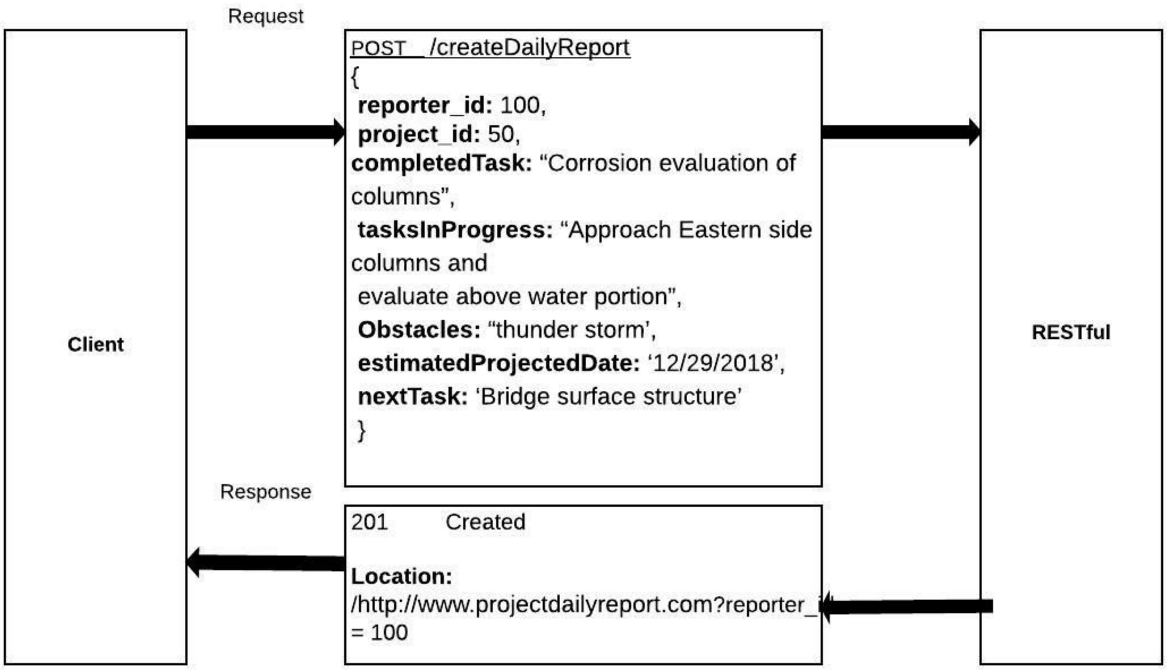
This root endpoint is for the personas who want to search daily report by the reporter name or project name. For instance, a persona called George who is a project manager wants to search all the daily reports submitted by one of the project team members like John or he wants to search all the daily reports based on the project name like ‘Summit Bridge Maintenance’, all of these procedures are done by GET http verb which filters the reports based on the reporter by reporter_id or based on the project name by project_id.

-Stretch feature: http://www.projectdailyreport.com/newUser
This endpoint is for the personas who are managers and would like to enter new user names in the application themselves. For instance a persona called Ben is the vice president of a solar system company and would like to enter new people who are added to their team, for doing this he goes to newUser page and fills in the form for the new user, all these operations are done by POST http verb which the a new user will be added to the list of users. If she does not enter data some of the required fields, the saving operation will be unsuccessful and there will be response of 400 that there is a missing data and creating the new user should be repeated.

-Stretch feature: http://www.projectdailyreport.com/newProject
This endpoint is for the personas who are managers and would like to enter new projects information in the application themselves. For instance a persona called Ben is the vice president of a solar system company and would like to enter new projects in their company, for doing this he goes to newProject page and fills in the form for the new project, all these operations are done by POST http verb which the a new project will be added to the list of projects. If she does not enter data some of the required fields, the saving operation will be unsuccessful and there will be response of 400 that there is a missing data and creating the new project should be repeated.

Provide example requests and responses, including erroneous requests:

1. Request - Response related to successful creating of a daily report

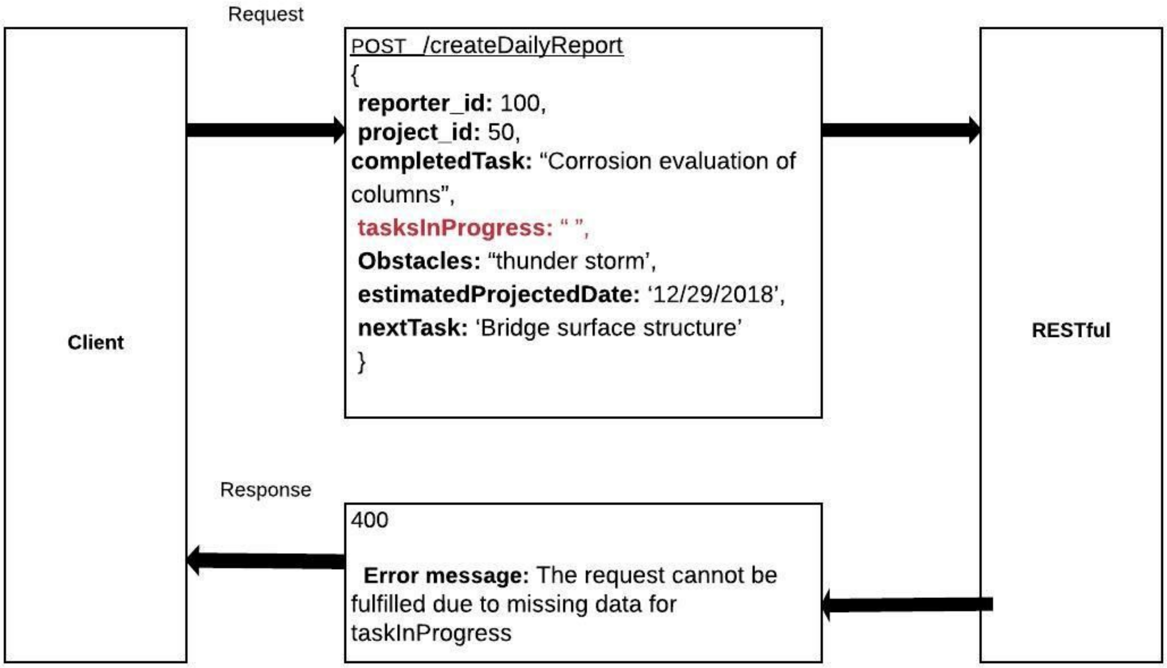


```
POST   /createDailyReport
{
 reporter_id: 100,
 project_id: 50,
completedTask: “Corrosion evaluation of
columns”,
 tasksInProgress: “Approach Eastern side
columns and
 evaluate above water portion”,
 Obstacles: “thunder storm’,
 estimatedProjectedDate: ‘12/29/2018’,
 nextTask: ‘Bridge surface structure’
 }
```

```
201        Created

Location:
/http://www.projectdailyreport.com?reporter_i
= 100
```

2. Request-response related to unsuccessful creation of a daily report:

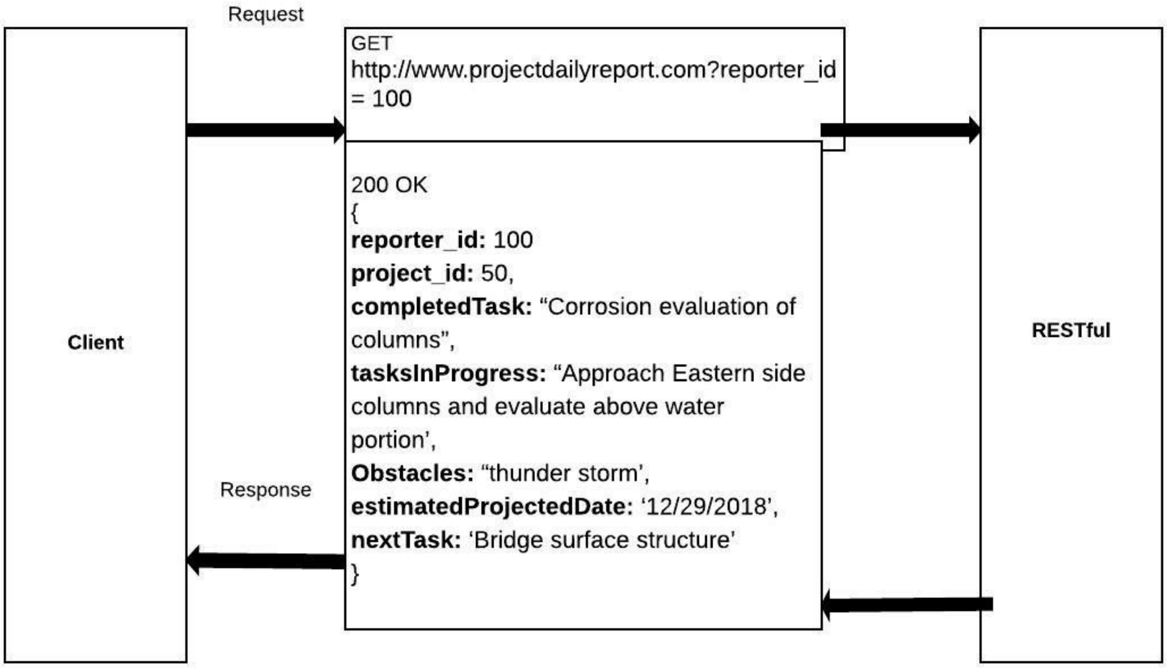


POST /createDailyReport
{
reporter_id: 100,
project_id: 50,
completedTask: “Corrosion evaluation of columns”,
tasksInProgress: “ ”,
Obstacles: “thunder storm’,
estimatedProjectedDate: ‘12/29/2018’,
nextTask: ‘Bridge surface structure’
}

400

Error message: The request cannot be fulfilled due to missing data for taskInProgress

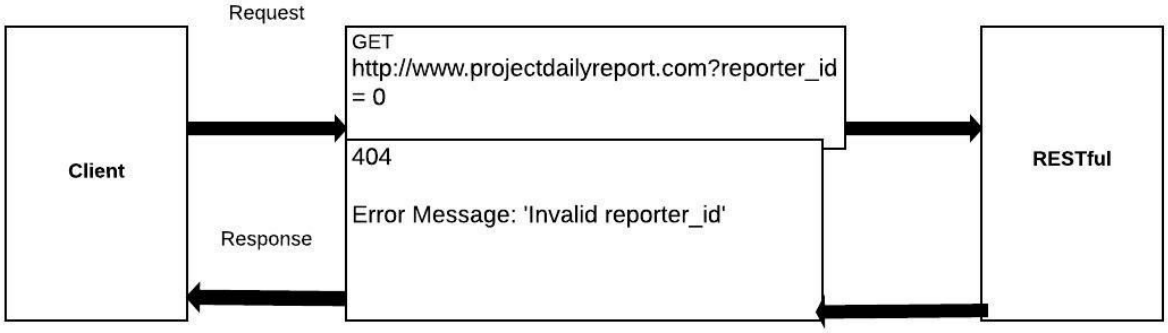
3. Request-response related to finding daily reports by reporter name:



GET
http://www.projectdailyreport.com?reporter_id
= 100

200 OK
{
reporter_id: 100
project_id: 50,
completedTask: “Corrosion evaluation of columns”,
tasksInProgress: “Approach Eastern side columns and evaluate above water portion’,
Obstacles: “thunder storm’,
estimatedProjectedDate: ‘12/29/2018’,
nextTask: ‘Bridge surface structure’
}

4. Error message related to invalid reporter_id for searching daily reports:

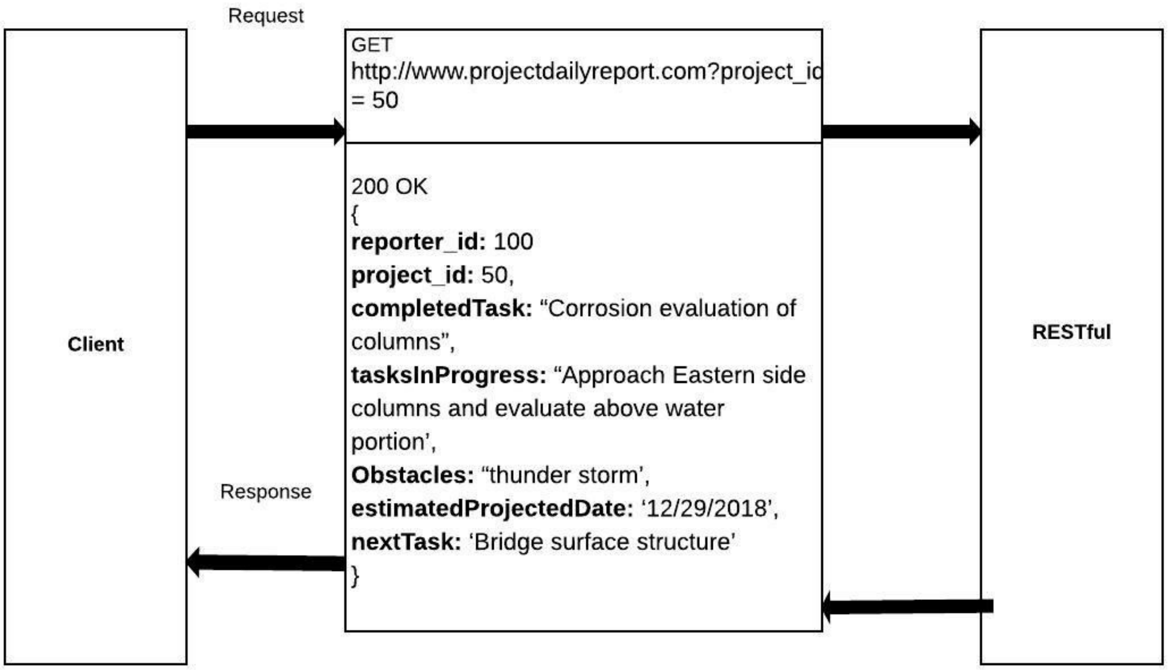


GET
http://www.projectdailyreport.com?reporter_id
= 0

404

Error Message: 'Invalid reporter_id'

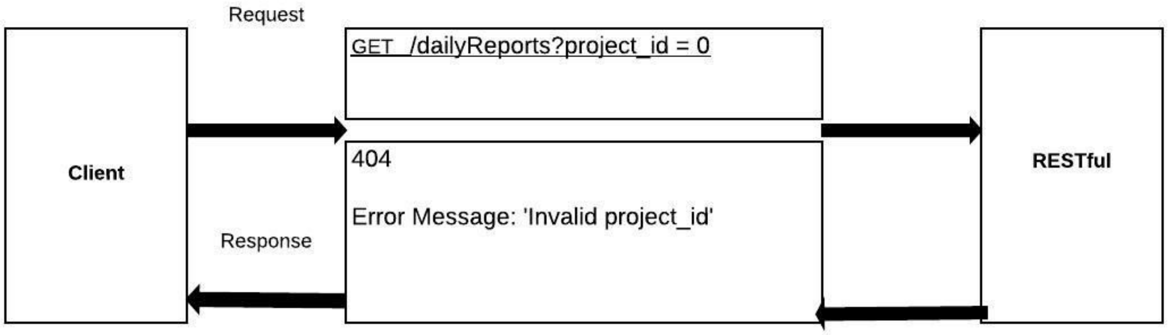
5. Request-response related to finding daily reports by project name:



```
GET
http://www.projectdailyreport.com?project_id
= 50
```

```
200 OK
{
reporter_id: 100
project_id: 50,
completedTask: “Corrosion evaluation of
columns”,
tasksInProgress: “Approach Eastern side
columns and evaluate above water
portion’,
Obstacles: “thunder storm’,
estimatedProjectedDate: ‘12/29/2018’,
nextTask: ‘Bridge surface structure’
}
```

6. Error message related to invalid project_id for searching daily reports:

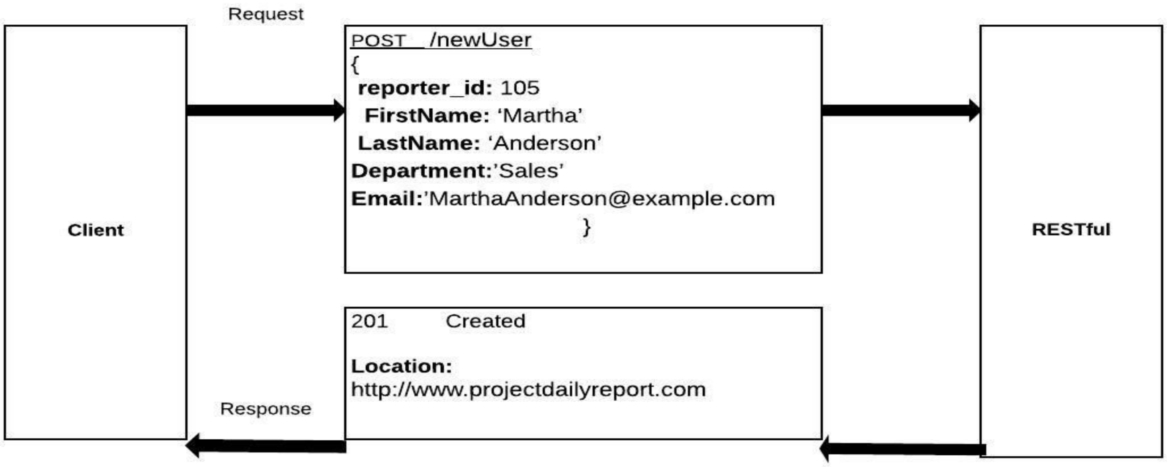


```
GET /dailyReports?project_id = 0
```

```
404

Error Message: 'Invalid project_id'
```

7. Stretch feature: Request - Response related to successful creating of a new user

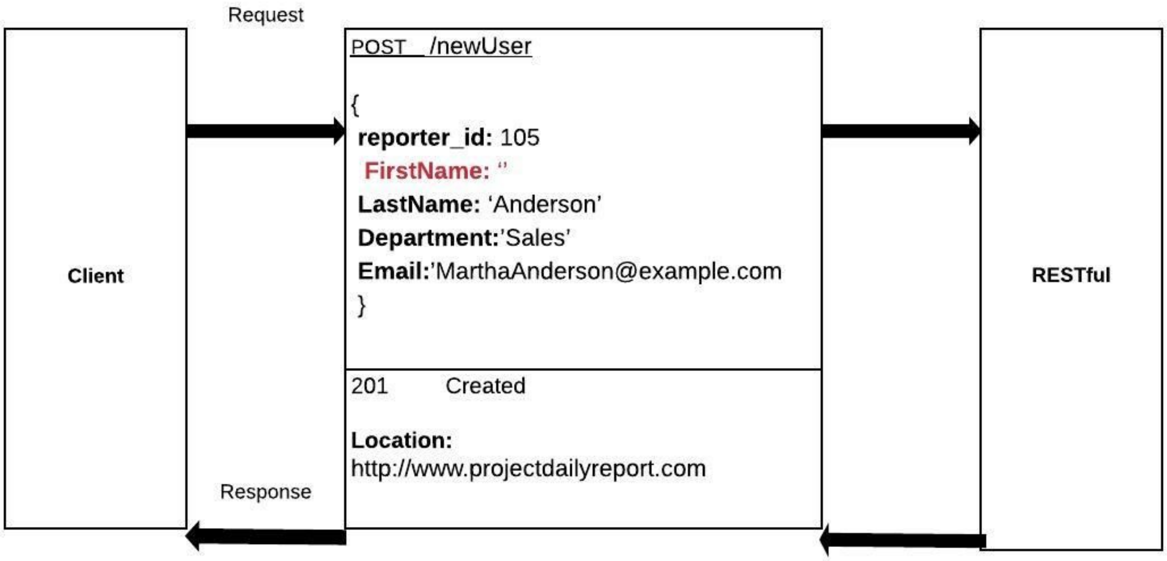


POST /newUser
{
reporter_id: 105
FirstName: ‘Martha’
LastName: ‘Anderson’
Department:’Sales’
Email:’MarthaAnderson@example.com
}

201 Created

Location:
http://www.projectdailyreport.com

8. Stretch feature: Request-response related to unsuccessful creation of a user:



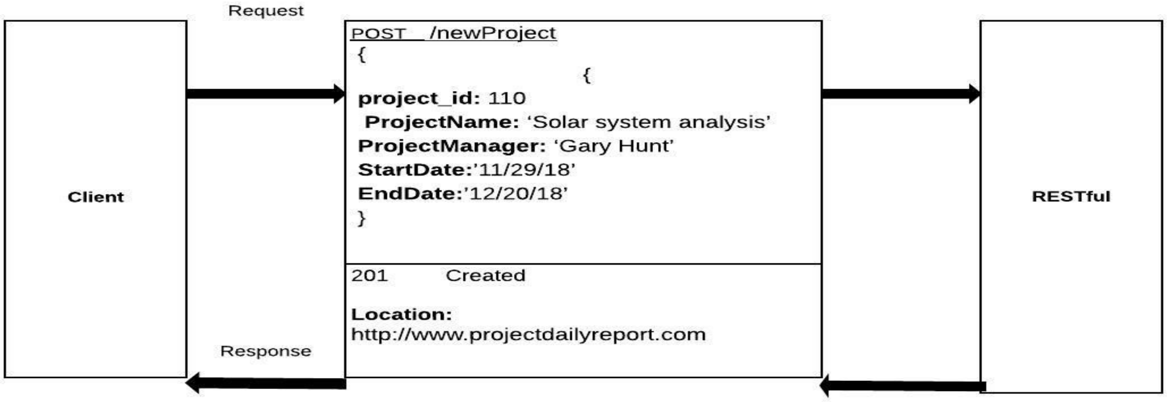
POST /newUser

{
reporter_id: 105
FirstName: ‘’
LastName: ‘Anderson’
Department:’Sales’
Email:’MarthaAnderson@example.com
}

201 Created

Location:
http://www.projectdailyreport.com

9. Stretch feature:Request - Response related to successful creating of a new project

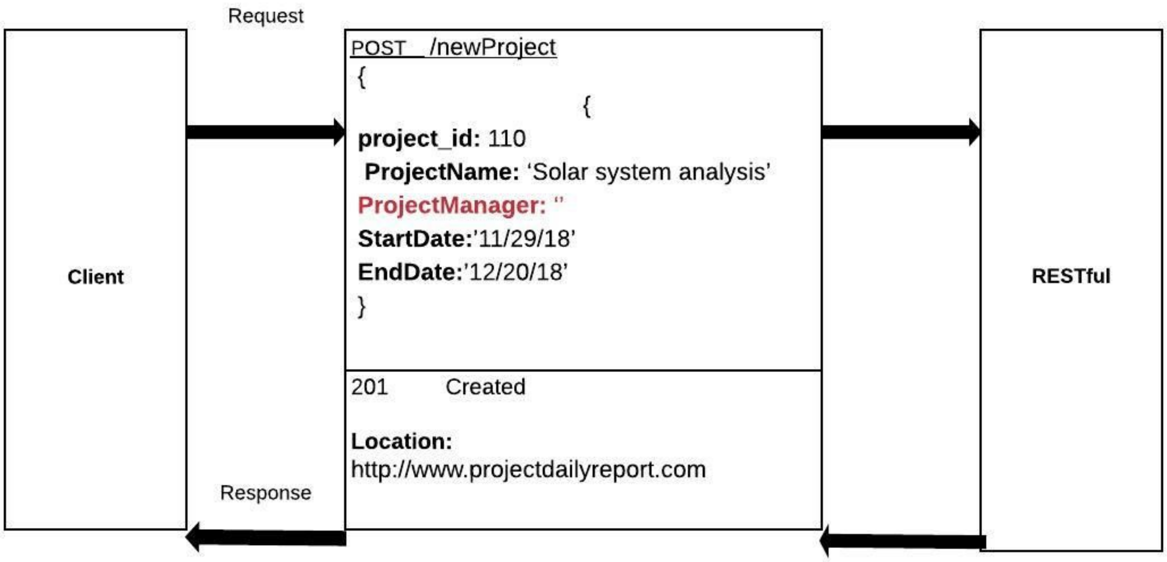


POST /newProject
```
{
                              {
project_id: 110
 ProjectName: ‘Solar system analysis’
ProjectManager: ‘Gary Hunt’
StartDate:’11/29/18’
EndDate:’12/20/18’
}
```

201   Created

Location: http://www.projectdailyreport.com

10. Stretch feature:Request - Response related to unsuccessful creating of a new project



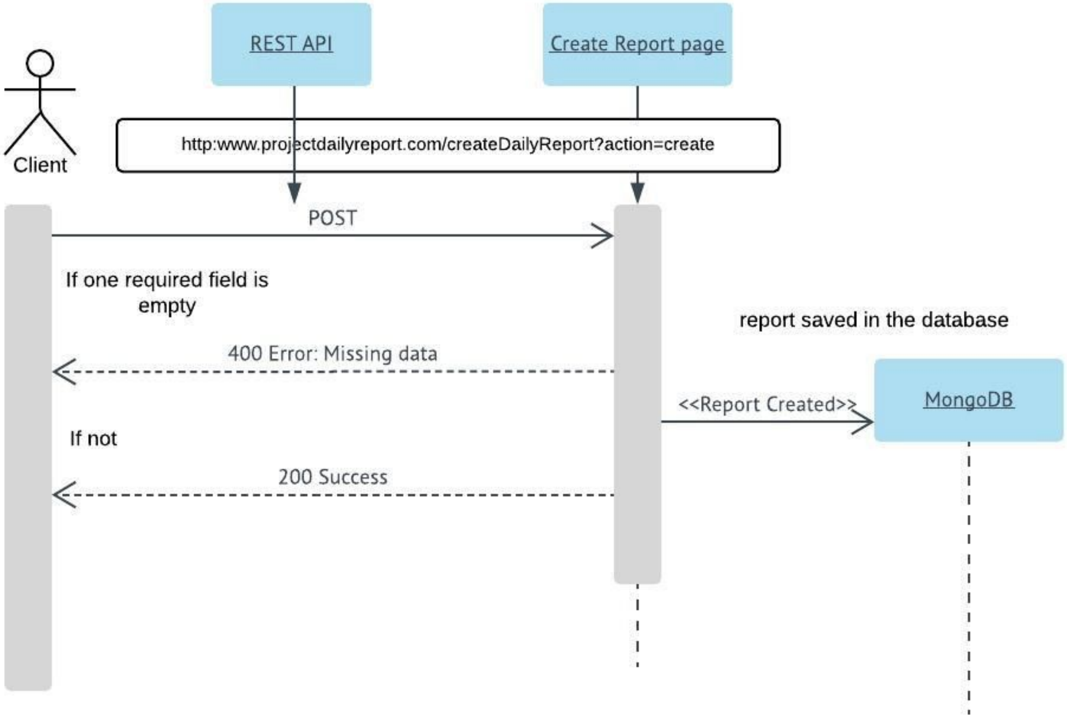
POST /newProject
```
{
                              {
project_id: 110
 ProjectName: ‘Solar system analysis’
ProjectManager: ‘’
StartDate:’11/29/18’
EndDate:’12/20/18’
}
```

201   Created

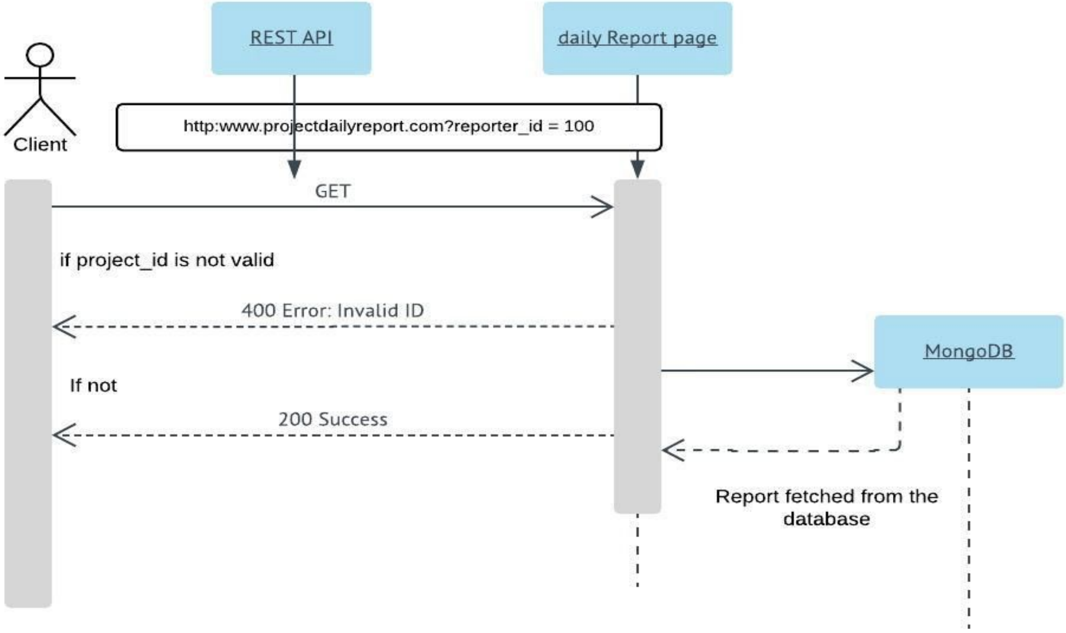
Location: http://www.projectdailyreport.com

Diagrams which user interface pages will interact with which service endpoints:

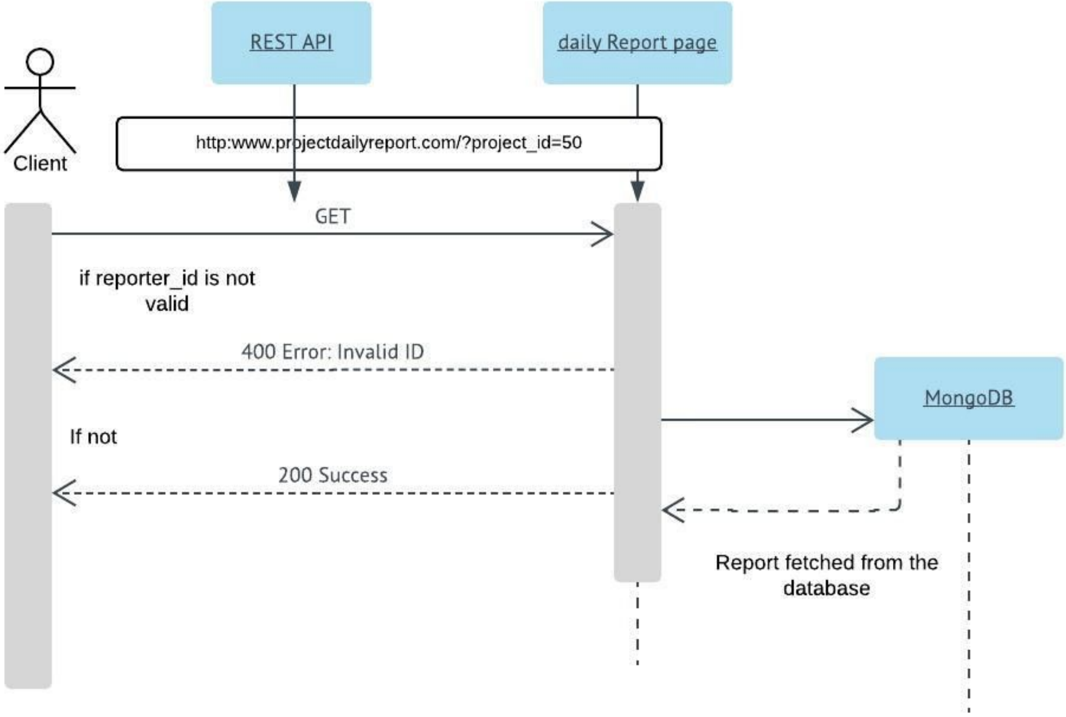
1. Interaction with the interface for creating a new daily report:


REST API
Create Report page
Client
http:www.projectdailyreport.com/createDailyReport?action=create
POST
If one required field is empty
report saved in the database
400 Error: Missing data
<<Report Created>>
MongoDB
If not
200 Success

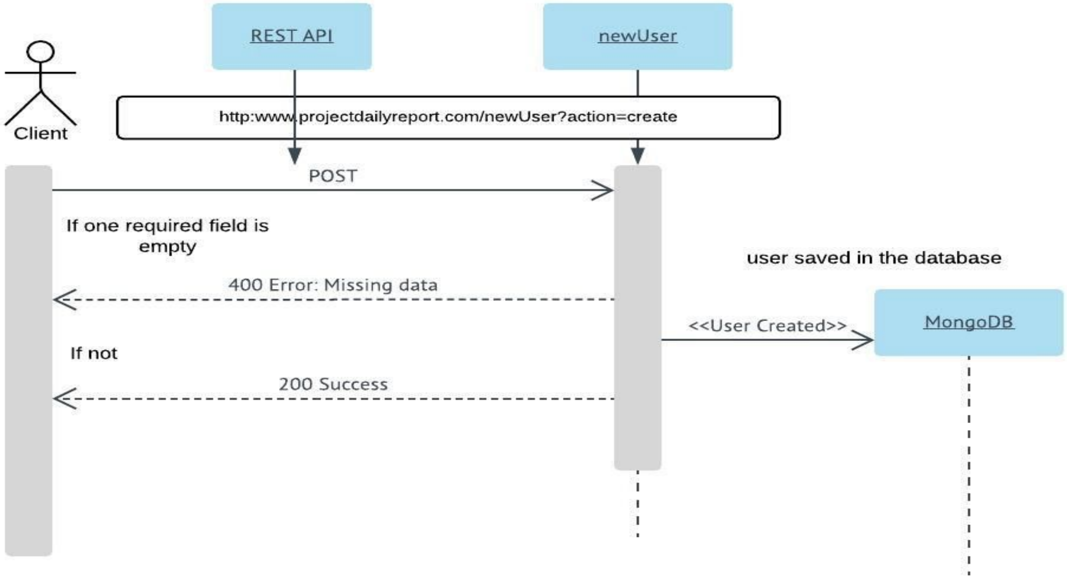
2. Interaction with the interface for searching reports by the reporter:


REST API
daily Report page
Client
http:www.projectdailyreport.com?reporter_id = 100
GET
if project_id is not valid
400 Error: Invalid ID
MongoDB
If not
200 Success
Report fetched from the database

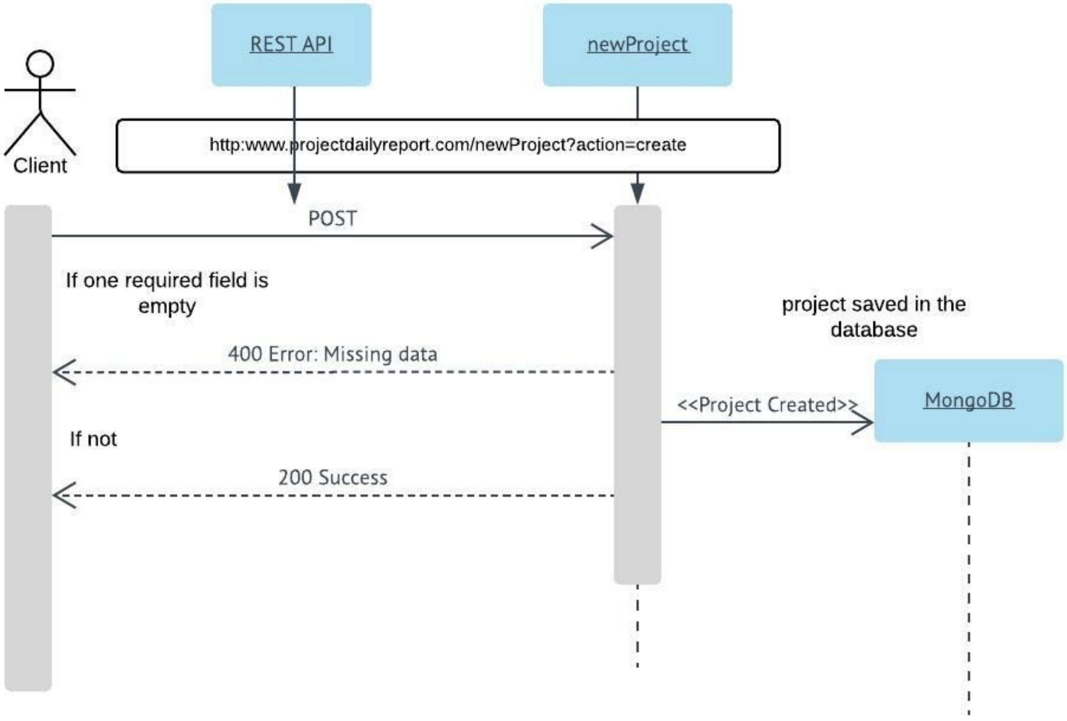
3. Interaction with the interface for searching reports by the reporter:


REST API
daily Report page
Client
http:www.projectdailyreport.com/?project_id=50
GET
if reporter_id is not valid
400 Error: Invalid ID
MongoDB
If not
200 Success
Report fetched from the database

4. Stretch feature: Interaction with the interface for creating a new user:


REST API
newUser
Client
http:www.projectdailyreport.com/newUser?action=create
POST
If one required field is empty
user saved in the database
400 Error: Missing data
<<User Created>>
MongoDB
If not
200 Success

5. Stretch feature: Interaction with the interface for creating a new project:


REST API
newProject
Client
http:www.projectdailyreport.com/newProject?action=create
POST
If one required field is empty
project saved in the database
400 Error: Missing data
<<Project Created>>
MongoDB
If not
200 Success