

Application features

- Get personalized recommended preventative care and preventative screenings based on FHIR patient data.
- Create update and delete reminders for those items.
- Get notified via SMS notification when the reminder is due.
- Recurring reminders get automatically rescheduled based on the clinically recommended frequency.

Other Info

- This was an individual project
- Team members: Alex Kuznicki
- Github link to project: https://github.gatech.edu/akuznicki3/me_maintenance

Accessing the Deployed Application

App Link: <https://me-maintenance-staging.herokuapp.com/>

NOTE: If the app has not been recently used it may take a few seconds to start up.

Sample Users:

- user 1: (**fhir_patient_id: 1653603**)
 - username: daniel_moya
 - password: pizza
- user 2: (**fhir_patient_id: 1836961**)
 - username: alex_musterfrau
 - password: pizza
- user 3: (**fhir_patient_id: 1161192**)
 - username: maxima831_wilderman619
 - password: pizza

These are patients that exist in the <https://hapi.fhir.org> FHIR server. They have worked consistently for me over the last few weeks. I have noticed before that the FHIR Server has gone down for maintenance before. I hope that will not happen when you try to use the app.

I have also seen a previous test patient I was using get deleted from the server before. If you have trouble logging in, Verify that the patient exists in the FHIR server by making a GET Request for the patient using the patient ID I noted above. If that is the case, Please try to create a new patient using one of the above matched patient IDs that I have listed for the sample logins.

Creating A New User

You can create your own user if you want. Additionally to test the SMS reminder feature yourself and get SMS notifications you will need to do this so you can add your phone number.

NOTE: I have asked AWS support for a service limit increase on SNS text messaging. Currently AWS only allows my phone number so that is why the test patients use my phone number. I am not sure when AWS will resolve my case but hope they do by the time you try to create a test user.

Use the below request to create a user. Copy and paste in your terminal and execute.

```
curl --header "Content-Type: application/json" \
-v --request POST \
--data '{"username":"wxyz","password":"xyz",
"phone_number":"+xxxxxxxxxxx",
"fhir_server_address":"https://hapi.fhir.org/baseR4", "patient_id":"1234"
}' \
http://me-maintenance-api.herokuapp.com/api/user
```

- The phone number must be in E.164 format.
- All fields are required.
- The patient ID must match to an existing patient in the FHIR server address.
- The username must be unique.
- Everything is a string type.

A successful request will show something similar to the below response.

```
> POST /api/user HTTP/1.1
> Host: me-maintenance-api.herokuapp.com
> User-Agent: curl/7.71.1
> Accept: */*
> Content-Type: application/json
> Content-Length: 144
>
* upload completely sent off: 144 out of 144 bytes
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: Cowboy
< Connection: keep-alive
< Date: Sun, 12 Dec 2021 00:52:25 GMT
< Content-Length: 140
< Content-Type: text/plain; charset=utf-8
< Via: 1.1 vegur
<
{"username":"wxyz","password":"xyz","phone_number":"+xxxxxxxxxxx","fhir_server_address":"https://hapi.fhir.org/baseR4","patient_id":"1234"}
* Connection #0 to host me-maintenance-api.herokuapp.com left intact
```

A successful response still does not mean it will be a valid user. You will have to have followed the format that I've shown above exactly, With a corrected formal phone number and existing FHIR patient.

Once logged in

Creating Reminders

- Select one of the preventative care items from the presented list by clicking on the plus (+) button.
- Select the Reminders tab
- Click the "select date"

Updating Reminders

- Select a reminder
- navigate to the Reminders tab
- select the "Change Date"

Deleting Reminders

- Select a reminder
- navigate to the Reminders tab
- select the delete reminder button

Deploying the application from scratch

Although the application should be deployed to Heroku already here are the steps to deploy from scratch.

1. Clone the git repo
2. Run the terraform scripts under the `me_maintenance/infra` directory. This will require an AWS account. Configure the terraform scripts to use your AWS credentials.
 - First run the `remote-state` Configuration with a new bucket.
 - Then run the `rds` Configuration
 - Connect to the RDS database and run the SQL Scripts from here `me_maintenance/infra/rds/tables/`. Make sure to run them in the order that they are numbered.
 - Then run the `reminder` Configuration
3. Create yourself 2 new Heroku Projects that use container run times.
4. Update the `me_maintenance/push.sh` file With one of the Heroku Project that u created names in the CLI commands (replacing the ones for my project.) - Then run the script. This is for the web App deployment.
5. update `me_maintenance/backend/push-api.sh` file With the other Heroku Project name in the CLI commands (replacing the ones for my project.) This will be for the API. - Then run the script. Add environment variables `DB_HOST` (rds database hostname) and `DB_PW` (database password).
6. Update `me_maintenance/lib/config/app_config.dart`. Replace the API URL From mine to yours that you created in step 5.
7. Update `me_maintenance/backend/push-reminder.sh` With the S3 bucket that you use specified in the terraform. - Then run the script.

8. Launch the app at the heroku project name that you created and specified in step 4.