**Project:** DayaMed Web Application

**Technologies:** Angular Application

**Test Automation Technologies**:

**IDE:** Eclipse IDE 2021 – 12,

**UI Automation Framework:** Selenium 4.1.1,

**Language:** Java,

**Project Management framework:** Maven

**Assertions:** TestNG,

**Runner frameworks:** TestNG/~~Cucumber,~~

**Below are the Smoke Test Cases planned for Automation:**

1. Open “DayaMed” URL in the web browser.
2. Login with Valid admin Credentials.
3. Click on menu icon and click on the Providers and select the Providers

list.

1. Click Add Provider link, provide the required data then, click on the

submit button.

1. Click on menu icon and click on the Providers, select any Provider on

which click on the Edit link, perform some modifications, click on the

Submit button, Then, verify provider data.

1. Click Add Caregiver link, provide the required data then, click on

the submit Button, Then, verify added caregiver.

1. Click on menu icon and click on the Providers and select the Providers list   
                     7.1. Select any Provider, Click on the View Patients.   
                     7.2. Select any Patient and click on the edit patient link.   
                     7.3. Modify some data, click on the submit button.   
                    7.4. Select same patient, click on the edit button.

8. Verify patient details in the edit patient page.

                             8.1. Select any Provider, Click on the View Patients.   
                             8.2. Select any Patient, click on the "View Patient" button.

                             8.3. Verify Admin able to see all the prescriptions available for the patient or

not.

1. Select any prescription then, click on the "View Prescription" button.
2. Click on the Reports link.

                             10.1. Verify admin able to see the all the reports (Location Report ,

Adherence , Medication Report, Notifications)

1. Login with valid Pharmacist or Pharmacist Admin credentials.
2. Login with valid Pharmacist (or) Pharmacist Admin credentials.   
                    12.1. Click on the Add Patient button, provide required data.   
                    12.2. Click on the submit button.

                              12.3. Verify the newly added patient.

1. View Prescription Button

                            13.1. Select any patient, click on the "View Prescriptions" button.

                             13.2. Click on the Add Prescription button, provide required data, click

on the submit button.

                             13.3. Verify the prescription actions.

1. Modify Prescription data

                             14.1. Select any previously added Prescription then, click on the

"View Prescription" button.   
                             14.2. Modify the data and click on the submit button.

                             14.3. Verify the prescription actions.

1. Reports:

                             15.1. Select any patient, click on the Reports icon.   
                             15.2. Click on each report.

                             15.3. Verify the report(s)

1. Notifications

                            16.1. Select any patient, click on the "Notifications" icon.

                            16.2. Verify the Notifications screen.

1. Verify the zoom call

                            17.1. Login with valid Pharmacist (or) Pharmacist Admin credentials.   
                            17.2. Select any patient, click on the Call icon.   
                           17. 3. Model displayed with patient, and Patient caregiver list.   
                            17.4. Select and click on the Patient/Caregiver.

            Estimation Start Date: 1/28/2022

**Application**: DayaMed Mobile Application (Android/iOS)

**IDE**: Eclipse IDE 2021 – 12,

**UI Automation Framework:** APPIUM,

**Languages:** Java,

**Tools:** Appium Desktop, UI Automator2, XCUITEST, APPIUM Inspector,

**Project Management framework:** Maven,

**Assertions:** TestNG,

**Runner frameworks:** TestNG/~~Cucumber~~.

**UI Automation Smoke Test Cases: (Mobile)**

    1. Install and launch the “DayaMed mobile application”.

         1.1. Verify the app launch screen.

    2. Click on the Signup Screen

           2.1. Select the patient/caregiver.

           2.2. Click on the "Forgot/Set password?" link.

    3.  Provide the valid credentials as patient, click on the sign in button.

             3.1. Should navigate to home Page.

    4.  Click on the "Overall Adherence" icon, Screen should be navigated to the "Overall

Adherence" screen.

    5.  Click on the Back link in the "Overall Adherence" screen

                5.1. Screen should be renavigated to the "Schedule screen".

          6. Click on the Linked profiles icon

               6.1. Screen should be navigated to the "Linked Profiles" screen.

          7. Click on the any medicine card.

              7.1. Screen should be navigated to the respective medicine details screen.

          8. Click on the "As Needed" tab.

                8.1. As Needed" screen should be opened.

          9.  Click on the "More" link.

          10.  Click on the Profile link.

                  10.1. Screen should be navigated to the Profile screen.

          11. Click on the Events link.

                11.1. Screen should be navigated to the Events screen.

          12. Verify user able to consume the medicine from schedule screen or not

               12.1. By scanning barcode 2. Without scanning barcode 3. By scanning NFC

                       chip

          13.  Provide the valid caregiver credentials, click on the sign in button

          14. Click on the linked profiles/events.

                 14.1. Screen should be navigated to the respective screen.

          15. Click on the More tab, click on the profile tab.

                15.1. Click on the changed password, verify user able to change the password

                     and able to login with updated password or not.

Estimation Start Date: Estimation End Date: 3/18/2022

**DayaMed API performance testing:**

**IDE**: Visual Studio Code

**Tool Used:** LOCUST,

**Languages:** Python,

**WEB APIs planned:**

1. **Login API – “**/medicare/api/login?lang=en"
2. **Patient Adherence Report –** /medicare/api/rest/adherence/intervaledreport?pharmacyId=1&lang=en
3. **Get over all patient adherence –** /medicare/api/rest/adherence/patient/analytics/89579?lang=en
4. **Get patient details by id–** /medicare/api/rest/patients/46243?lang=en
5. **Add/Update prescription–** /medicare/api/rest/patients/89579/prescriptions/add?lang=en
6. **Get the list of prescriptions in For review tab screen–** /medicare/api/rest/patients/89579/prescriptions?prescriptionAdherenceRequired=true&lang=en
7. **Get user details–** /medicare/api/rest/providers/byuser/1174?lang=en
8. **Get the calendar notifications by patientid–** /medicare/api/rest/utility/patient/calendernotifications/byuserid/1265?lang=en
9. **Get the list of patients in patient list screen–** /medicare/api/rest/w/patients?lang=en
10. **Get the list of patients in RX refill screen–**/medicare/api/rest/w/pharmacist/patients?limit=10&offset=0&sortBy=lastActiveTime&sortOrder=desc&query=and&lang=en

**Script:**

from locust import HttpLocust, TaskSet, task, between

import logging, sys

import json

from credentials import \*

from caregivercredentials import \*

class LoginWithUniqueUsersSteps(TaskSet):

    username = "NOT\_FOUND"

    password = "NOT\_FOUND"

    caregiverusername = "NOT\_FOUND"

    caregiverpassword = "NOT\_FOUND"

    responsedict = {}

    responsedict1 = {}

    responsedict2 = {}

    def on\_start(self):

            if len(USER\_CREDENTIALS) > 0:

                self.username, self.password = USER\_CREDENTIALS.pop()

            if len(CAREGIVER\_CREDENTIALS) > 0:

                self.caregiverusername, self.caregiverpassword = CAREGIVER\_CREDENTIALS.pop()

    @task(1)

    def addprescription(self):

        response = self.client.post("/medicare/api/login?lang=en",

                                    json={"languagePreference": "en", "password": self.password, "typeofdevice": "web",

                                          "username": self.username}, verify=False, catch\_response=True)

        str\_Login\_response = json.dumps(response.json())

        self.responsedict = json.loads(str\_Login\_response)

        print(self.responsedict['jwtToken'])

        #print(self.responsedict['userDetails']['id'])

        self.client.post("/medicare/api/rest/w/patients?lang=en",  json={"limit":10,"providerId":[],"offset":0,"sortBy":"lastName","sortOrder":"asc","query":""}, headers={"authorization":self.responsedict['jwtToken']})

        self.client.post("/medicare/api/rest/w/patients?lang=en",  json={"limit":10,"providerId":[],"offset":1,"sortBy":"lastName","sortOrder":"asc","query":""}, headers={"authorization":self.responsedict['jwtToken']})

        self.client.post("/medicare/api/rest/w/patients?lang=en",  json={"limit":10,"providerId":[],"offset":2,"sortBy":"lastName","sortOrder":"asc","query":""}, headers={"authorization":self.responsedict['jwtToken']})

        response1 = self.client.post("/medicare/api/rest/w/patients?lang=en",  json={"limit":10,"providerId":[],"offset":0,"sortBy":"lastName","sortOrder":"asc","query":""}, headers={"authorization":self.responsedict['jwtToken']})

        str1 = json.dumps(response1.json())

       # print("Line46 "+str1)

        self.responsedict1 = json.loads(str1)

        print(self.responsedict1['patients'][0]['id'])

        self.client.post("/medicare/api/rest/patients/"+str(self.responsedict1['patients'][0]['id'])+"/prescriptions/add?lang=en", headers={"authorization":self.responsedict['jwtToken']}, json ={"canPatientModify": "true","comment": "","commodityInfoList": [],"deviceInfoList": [],"diagnosisList": [{"id": "35", "name": "test3"}],"0": {"id": "35", "name": "test3"},"diseaseInfoList": [{"id": "null", "disease": {"id": "5", "name": "null", "type": "null", "description": "null", "category": "null"}}],"0": {"id": "null", "disease": {"id": "5", "name": "null", "type": "null", "description": "null", "category": "null"}}, "dosageDevices": [],"dosageInfoList": [{"id": "null"}],"0": {"id": "null"},"expectedAdherence": "null","id": "null","patientId": "null","solution": "dcs"}, verify=False)

        self.client.get("/medicare/api/rest/providers/byuser/"+str(self.responsedict['userDetails']['id'])+"?lang=en", headers={"authorization": self.responsedict['jwtToken']})

        self.client.get("/medicare/api/rest/patients/"+str(self.responsedict1['patients'][0]['id'])+"?lang=en", headers={"authorization":self.responsedict['jwtToken']})

        self.client.get("/medicare/api/rest/patients/"+str(self.responsedict1['patients'][1]['id'])+"?lang=en", headers={"authorization":self.responsedict['jwtToken']})

        response2 = self.client.get("/medicare/api/rest/patients/"+str(self.responsedict1['patients'][0]['id'])+"?lang=en", headers={"authorization":self.responsedict['jwtToken']})

        str2 = json.dumps(response2.json())

        #print("Line 53 "+str2)

        response3 = self.client.get("/medicare/api/rest/patients/"+str(self.responsedict1['patients'][1]['id'])+"?lang=en", headers={"authorization":self.responsedict['jwtToken']})

        str3 = json.dumps(response2.json())

        #print("Line 54 "+str2)

        self.client.get("/medicare/api/rest/adherence/patient/analytics/"+str(self.responsedict1['patients'][0]['id'])+"?lang=en", headers={"authorization":self.responsedict['jwtToken']})

        self.client.post("/medicare/api/rest/utility/patient/calendernotifications/byuserid/"+str(self.responsedict1['patients'][0]['userDetails']['id'])+"?lang=en", json={"start":"2021-10-31","end":"2021-12-05"}, headers={"authorization":self.responsedict['jwtToken']})

        self.client.post("/medicare/api/rest/utility/patient/calendernotifications/byuserid/"+str(self.responsedict1['patients'][1]['userDetails']['id'])+"?lang=en", json={"start":"2021-10-31","end":"2021-12-05"}, headers={"authorization":self.responsedict['jwtToken']})

        response4 = self.client.post("/medicare/api/rest/utility/patient/calendernotifications/byuserid/"+str(self.responsedict1['patients'][0]['userDetails']['id'])+"?lang=en",  json={"start":"2021-10-31","end":"2021-12-05"}, headers={"authorization":self.responsedict['jwtToken']})

        str4 = json.dumps(response4.json())

        #print("Line 63 "+str4)

        self.client.get("/medicare/api/rest/w/pharmacist/patients?limit=10&offset=0&sortBy=lastActiveTime&sortOrder=desc&query=and&lang=en", headers={"authorization":self.responsedict['jwtToken']})

        response6 = self.client.get("/medicare/api/rest/w/pharmacist/patients?limit=10&offset=0&sortBy=lastActiveTime&sortOrder=desc&query=and&lang=en", headers={"authorization":self.responsedict['jwtToken']})

        str6 = json.dumps(response6.json())

        print("Line 68 "+str6)

        self.client.get("/medicare/api/rest/patients/"+str(self.responsedict1['patients'][0]['id'])+"/prescriptions?prescriptionAdherenceRequired=true&lang=en", headers={"authorization":self.responsedict['jwtToken']})

        response5 = self.client.get("/medicare/api/rest/patients/"+str(self.responsedict1['patients'][0]['id'])+"/prescriptions?prescriptionAdherenceRequired=true&lang=en", headers={"authorization":self.responsedict['jwtToken']})

        str5 = json.dumps(response5.json())

        #print("Line 71 "+str5)

        self.client.get("/medicare/api/rest/adherence/intervaledreport?pharmacyId=1&lang=en", headers={"authorization":self.responsedict['jwtToken']})

class LoginWithUniqueUsersTest(HttpLocust):

    task\_set = LoginWithUniqueUsersSteps

    wait\_time = between(10, 20)

**Reports for 4 users per second:**

Chart

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

**Status:** Done with main API’s. if any new APIs should be added we have to add.

**Mobile APIs planned:**

1. Validate Token - **api/validate/mobiletoken**
2. Login - **dayamed-web/api/user/login**
3. Get User Details - **/api/rest/shared/userdetails/byuserid/1272**
4. Active Prescriptions - **/api/rest/prescriptions/activate**
5. Get All Active Prescriptions - **/api/rest/prescriptions/patient/active**
6. Start Prescriptions - **/api/rest/utility/startprescriptionscheduling/multiple**
7. Get Current day med cards **- /api/rest/shared/m/adherence/patient/83899**
8. Calendar Month Adherence **- /api/rest/shared/m/adherence/patient/83899**
9. Fetch Medication Images **- /api/rest/fetchimage/12939016008**
10. Get Analytics **- /api/rest/adherence/patient/analytics/83899**
11. Get PRN Prescriptions **- /api/rest/PRNprescription/83899**
12. Get Pending Prescriptions **- /api/rest/pendingprescriptions/83899**

**Status: In progress, can be completed by 25/02/2022**