

Introduction

Refactor the existing appointment scheduling module and bring it out as a new repo written in React. This would act like the version 2 of appointment scheduling.

The refactoring will help us move away from a monolith BahmniApps code to a more modular code, using the latest frameworks. This will ease the maintenance of code and also act like a POC for moving other modules out from BahmniApps.

Path for refactoring

We would need to do the following to achieve the final goal:

1. **Bundle common code:** Identify & bundle the code used commonly between appointment scheduling and other modules of Bahmni-Apps as a single js file. To do this we need to copy the code in the common folder (/ui/app/common), in the current code and generate a common js file using build tools like webpack. We will add the common code as a dependency to the new appointment module, which will then not be altered. Any enhancements required should be made directly in the new code base.
2. **Create a new react project:** Create a new project with the required react dependencies. In this project we also need to include the angular js dependencies, which will be required until we migrate the entire module to react. Also include the required test dependencies, for testing the code written in react/angular. (Note- we can use angular-cli/ create-react-app to provide out of box project for angular or react)
3. **Move the existing appointment code to new project:** To move the existing, we need to use react+angular js adapter like React2Angular, which will allow angular js app to run side by side to react. We can move the code slowly from angular js to react, until we have migrated the entire module. To know the things to be moved refer the story list below.

The order in which we will move the code from angular to react will be to migrate all the directives in a controller and then migrate the controller.

(Note- Once we have the majority of the code on react and less on angular we can start using angular2React package, which will convert angular code to react components)

- 4. Write the build & deploy steps for this new project:** Using build tools like webpack, write the various steps required to build & deploy the new project.
- 5. Add the required proxy pass setting in ssl.conf:** Make entries similar to implementer-interface in ssl-conf for the new repo.
- 6. Write the steps in bahmni-playbook to deploy this new repo.**
- 7. Create a pipeline to build and deploy the common code and the new appointment repo.**
- 8. Translation using react-intl:** Before we move the actual app completely to react we will need to move the translations to react, using libraries like react-intl.

Notes:

It would be a good idea to use react along with redux, redux would be used to maintain the state of the entire application, hence making it easier to maintain. We can wrap all the individual components to redux and pass the required values.

Stories:

Serial No.	Card	Estimates (days)
1	Identify and bundle the common code as a js file	4
2	Create a new repo with react and angular js dependencies, also include the other dependencies required by the current appointment scheduling module	3
3	Create an adaptor layer to create a bridge between react+ redux and angular	2

4	Write webpack configuration to bundle the new repo	
5	Include a react library like react-intl to help in translation of new react components which will be created	3
6	Create react text input component with auto search and drop down (as per the new UI suggestions)	3
7	Create a react component with just drop down	2
8	Write a common layer to make API calls like get, post, put, delete. This layer will pass all the required headers etc.	2
9	Move patient search api to react and use the auto suggest text input throughout appointment scheduling	2
10	Create a react component to select multiple values from dropdown and show, use this for providers throughout appointment scheduling	5
11	Move the add new appointment button to react	2
12	Move the save appointment button to react	2
13	Move the search service box to react	3
14	Move the tree structure of speciality and service to react	5
15	Move api to fetch location, service, service type to react	3
16	Create a date picker/calendar in react, as per the enhanced ui	5
17	Create a time picker in react, as per the enhanced ui	3

18	Create week day selector component in react	3
19	Move the week picker to new tech stack, as per the new mock	4
20	Move the current grid for showing scheduled appointment to new tech stack	20
21	Move the routing to React	5
22	Redesign the add and edit screen as per the new ui	7
23	Move the appointment list view to new tech stack	10
24	Move the serach filter to react	4
25	Move the today button and date switcher to react	4
26	Move the print list view to new tech stack	8
27	Move the Admin console of Appt scheduling to new tech stack	14
28	Write a gulp/yarn task to help build and deploy the new repo	5
29	Migrate the color picker directive to a component	5
30	Write the playbook steps to deploy it with Bahmni Installer	4
31	Create a pipeline to build and deploy this new code	4
	Total	146