

An architectural description of the functionality:

The 'Ethiopia Pill Reminder(ICAPPR)' module is built on the top of the Motech IVR, Platforms event, Service modules, and uses IVR service provider 'Verboice' to interact with the patients. It listens to the published Motech Events and invokes required methods from the Motech IVR service to handle occurred incidents.

ICAPPR modules listens to the following events:

1. Send campaign messages
2. Pill Reminder, side_effects, and adherence survey call
3. Enrollment in the program
4. Update, stop service
5. Appointment call
6. PIN Authentication

Following are the main classes/components of ICAPPR module:

Listeners: The ICAPPR module uses the MotechSchedulerService to schedule the Survey, Pill reminder calls and send the campaign messages. When such events are triggered, messages from the event queue are published. Listeners classes listen to the published messages. These classes invoke various service method such CallIntiationService, MotechService, IVR service to handle such requests.

-CommCareStubFormListener:

- 1.The health workers register the patients for any of mentioned programs using the CommonCareHQ app called Enrich Ethiopia.
2. The CommonCare health API sends the HTTP forward stub form request to the Motech-ICAPPR server. As a result. event with the subject 'FORM_STUB_EVENT' is triggered. Class 'MotechEvent' already configured to handle required subjects. It also sends the event data to the Motech Scheduler when the event gets fired. Motech scheduler schedule the event for the future.
3. CommonCareStubListener listens to the published event with the subject Send_Stub_Form. The method handleStubForm(MotechEvent event) is annotated with custom @MotechListeners to listen to this event.
4. The annotated method check for the null values of formID and if no exception occurs, it calls to the Motech's CommonCareService.retrieveForm(formID) method to retrieve the particular instance of a form from the database.
5. handleForm() recovers the form as the JSON payload. It also checks the namespace for the retrieved form and calls appropriate handler method depending on the type of form, i.e., registrationForm, updateForm, and stopForm.
6. If it's the Registration Form, the request is further handled by ICAPPR's RegistrationFormHandler class. It creates the PillReminderRegistartion instance and retrieves the information such as preferred language, phone number, pref_call_time and set the PillReminderRegistartion object attributes. It then calls PillReminderRegistrar.register() method.
7. PillReminderRegistrar model the Patient data such phone number, PIN, preferred language and save it to the Motech-server for later use. It also enrolls the patient for the Pill Reminder Service.

-SendCampaignListener:

1. the SendCampaignMessageListener. sendCampaignMessage() get invoked when scheduler publishes SEND_Message event.

2. The above method calls `MRSPatientAdapter.getPatientbyID()` to get the saved entity from the Motech-server. It retrieves the patient phone number, language preferences, patient id and encapsulates in the request object.
3. The request object is passed as a parameter to the `initiateCall(request)` of the `CallInitiationService`.
4. `CallInitiationService` takes care of initiating the call. The `requestType` is evaluated to get Call flow id from the `PillReminderSetting`.
5. The instance of `CallRequest` is created by passing the channel name, phone number and timeout time as constructor arguments. The patient motechID, request type, flow_id, callback_URL, etc. are stored as payload data. IVR Service uses this data to maintain the flow session for the further use.
6. Following this, the mentioned listener class call to the Motech `IVRService.intiateCall(callRequest)` method to start the call.

-InitiateCallListener:

1. The methods in this class listen to the events such `SIDE_EFFECT`, `Adherence_assessment`, `appointment`, and `pill reminder`.
2. Instantiated request object encapsulates the required data such patient id, phone number, preferred language and call the `initiateCall()` method of the `CallInitiationService` to start the call.

-`CallInteractionListener` listens to `Yes_Side_effect` and `No_side effect` responses, maintains the correct order of the events depending on the patient answers.

-`EndOfCallRetryListener` handles the `End_OF_Call` event. It identifies the reasons for the call end and depending on that schedule the retry call using the `MotechSchedulerService`.

-`EndProgrammListener` listens to the event with `Stop_Call` and invokes the necessary service method to handle the event.

Controllers: The controller classes handles the HTTP requests from the Verboice.

-`VerboiceInteractionController` handles HTTP requests from the Verboice and publishes event messages during the Side Effect survey call.

-`PinAuthenticationController` handles the PIN authentication request from the Verboice.

-`SchedulerController` handles schedule/unschedule HTTP requests from the Verboice and invokes the various method in `MotechSchedulerService`.

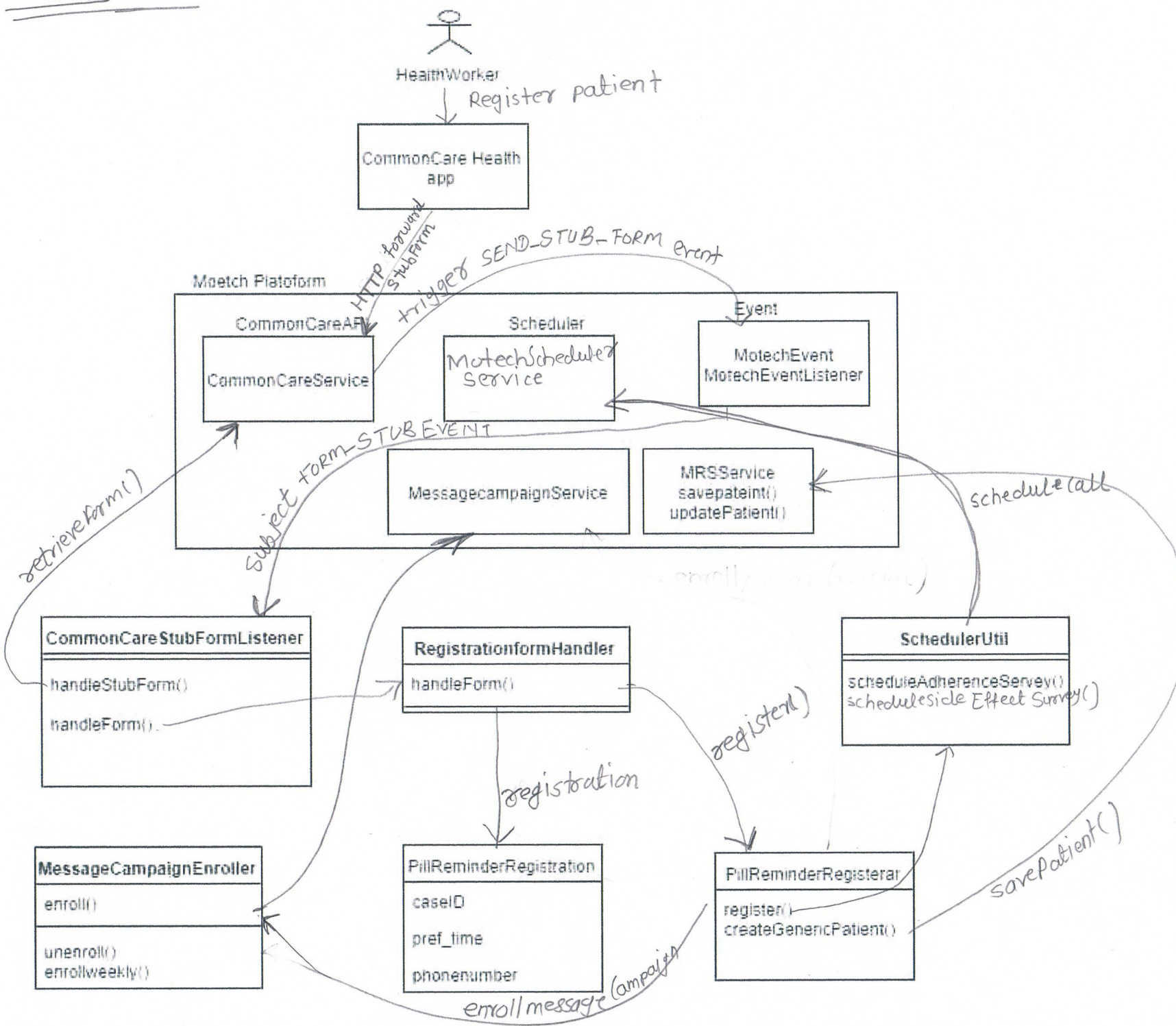
Handler classes:

-`RegistrationFormHandler` class handles the new registrations events.

-`UpdateFormHandler` handles the `PillReminderUpdate` event. This component calls various methods in `MessageCampaignEnroller` class to handle unsubscribe, update requests for the events such as `Send Campaign messages`, `Pill Reminder`, and `Appointment calls`.

-`StopFormHandler` handles the `Stop service request`.

Enrollment and call initiation sequence diagrams are shown as below:



Call initiation, interaction, end of call listeners

