Enabling SMART Apps in OpenMRS

⚠ This module has been released. Please see Smart Container Module for more information

OpenMRS mentor (primary)	Hui Xiao
SMART mentor	Josh Mandel
Assigned to	Unknown User (ajanthan)

Background

In today's health informatics ecosystem, there is no standard way to write an "App" that interacts with patient-level health data. Just as OpenMRS allows extensibility through modules, every EMR provides its own distinct extensibility interfaces. The result is a heterogeneous landscape where it's very difficult to deploy a single app across multiple systems.

The SMART Project provides a common API to access patient records, fueling apps with rich, unambiguous data. SMART is a 4-year, $15M project funded by the Office of the National Coordinator for Health Information Technology. SMART's goal is to lower the barriers to creating apps for health, so that application developers can focus on their own great ideas, not the minutia of systems integration. Developers write a SMART app once and deploy it on any compatible Electronic Medical Record or Personal Health Record system.

Abstract

The goal of this project is to create a module in OpenMRS that is a SMART Container. This container will enable SMART apps to be installed in OpenMRS, with two major benefits:

1. OpenMRS leverages a growing community of SMART developers and applications
2. SMART applications extend their reach to OpenMRS installations

Design

Objectives

1. Create a module in OpenMRS that has a page / set of pages to display SMART apps
2. With this module, turn OpenMRS into a "SMART Container":
   a. Expose basic data types (patient demographics, medications, and problems) as SMART RDF
   b. Implement SMART Connect API calls to communicate with SMART apps
   c. Provide a simple user interface to launch and display SMART Apps

Extra Credit

1. Develop methods for access to users
2. Develop methods for access to allergies
3. Develop methods for access to labs
4. Develop scripts to automate installation of SMART app in OpenMRS

Design

- Questions to ponder
  - How to install a SMART app?
  - Is it worth implementing the REST api instead of the direct javascript connect POST methods?
  - How to require that concepts be mapped to the vocabularies SMART uses (SNOMED for problems, RxNorm for medications)

Resources

- http://www.smartplatforms.org/
- http://dev.smartplatforms.org/Main_Page
- http://dev.smartplatforms.org/Developers_Documentation:_SMART_Container_Javascript_Libraries

Implementation Details

UI mockup

The module will contain three sets of ui.

one for listing available Apps and allow user to launch it within a HTML iframe by clicking the icon of the app.This UI will be placed as a tab in the patient dashboard.
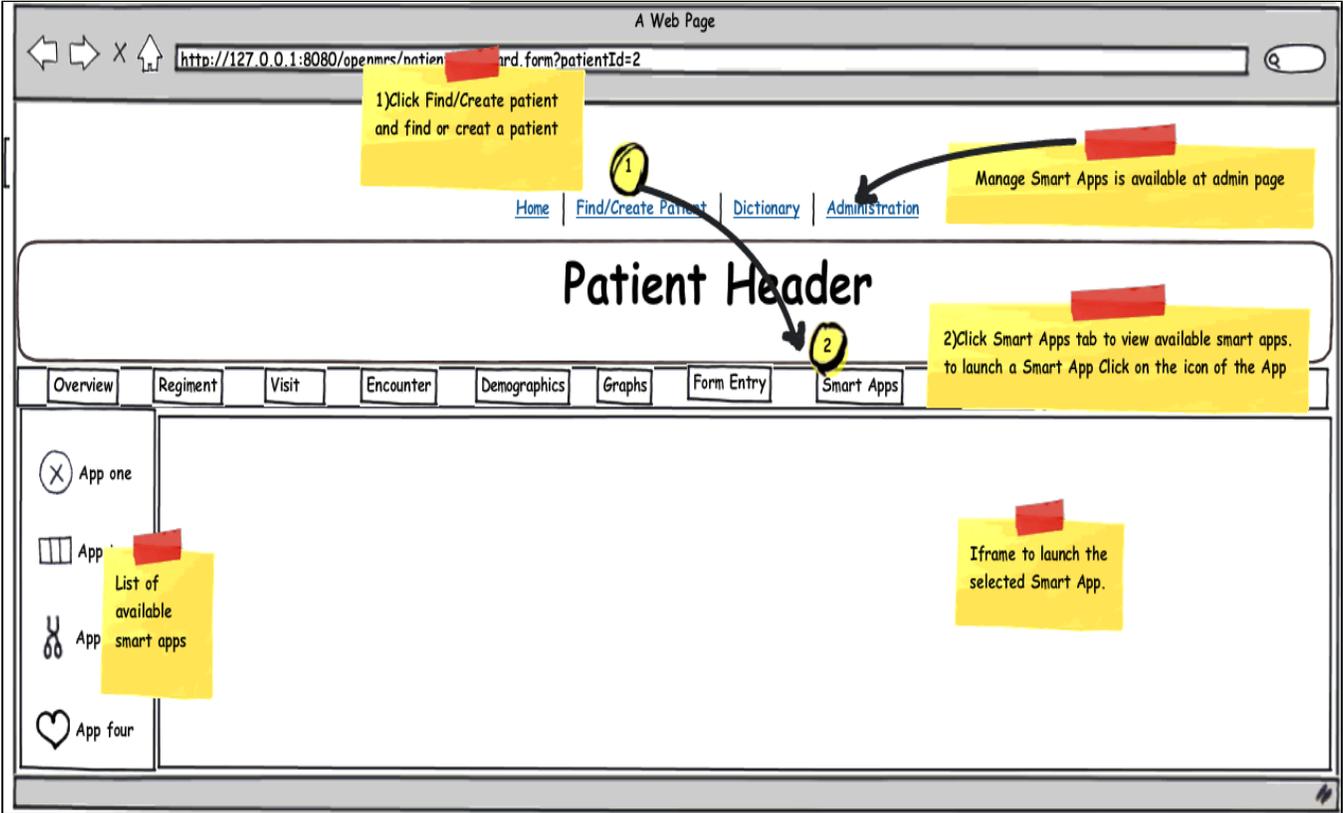


Next UI is to allow user to select apps to display in the patient dashboard from the available installed apps.This UI will be placed under the My Profile->manage smart app.
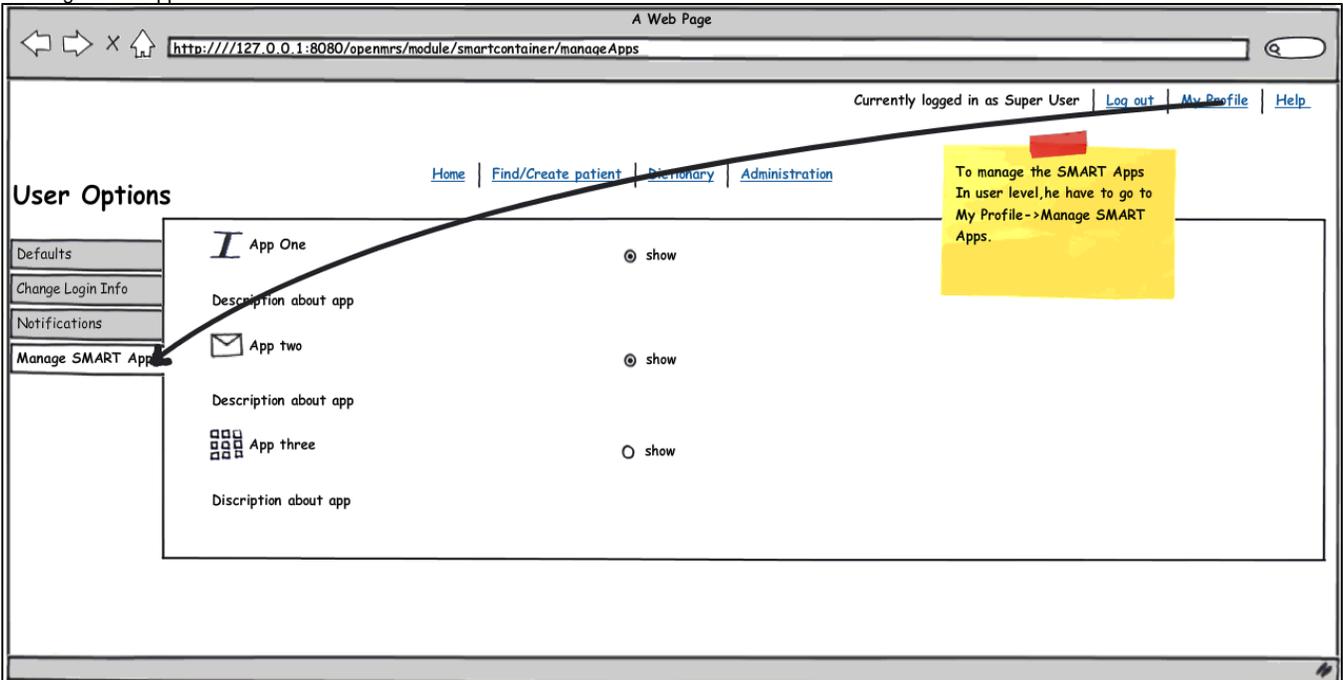
The folowing UI allow administer to install new apps or remove apps that are already installed.The appropriate place to display this UI is Administration->manage smartapps.
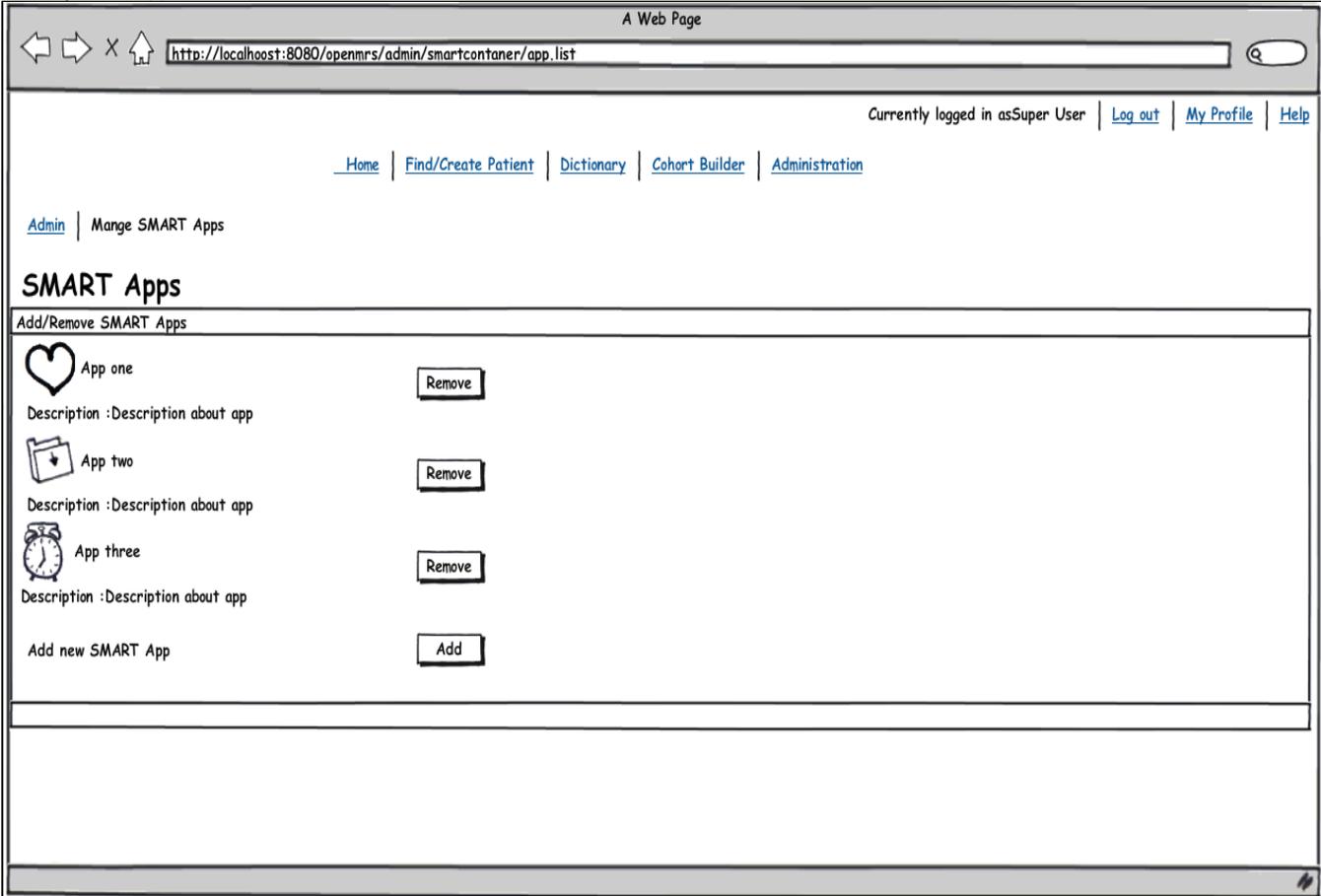

While administrator wants to add new SMART Apps he have to click on add button in above UI.This will give a popup window that allow administrator to up load a SMART App manifest or give a link to a manifest file.
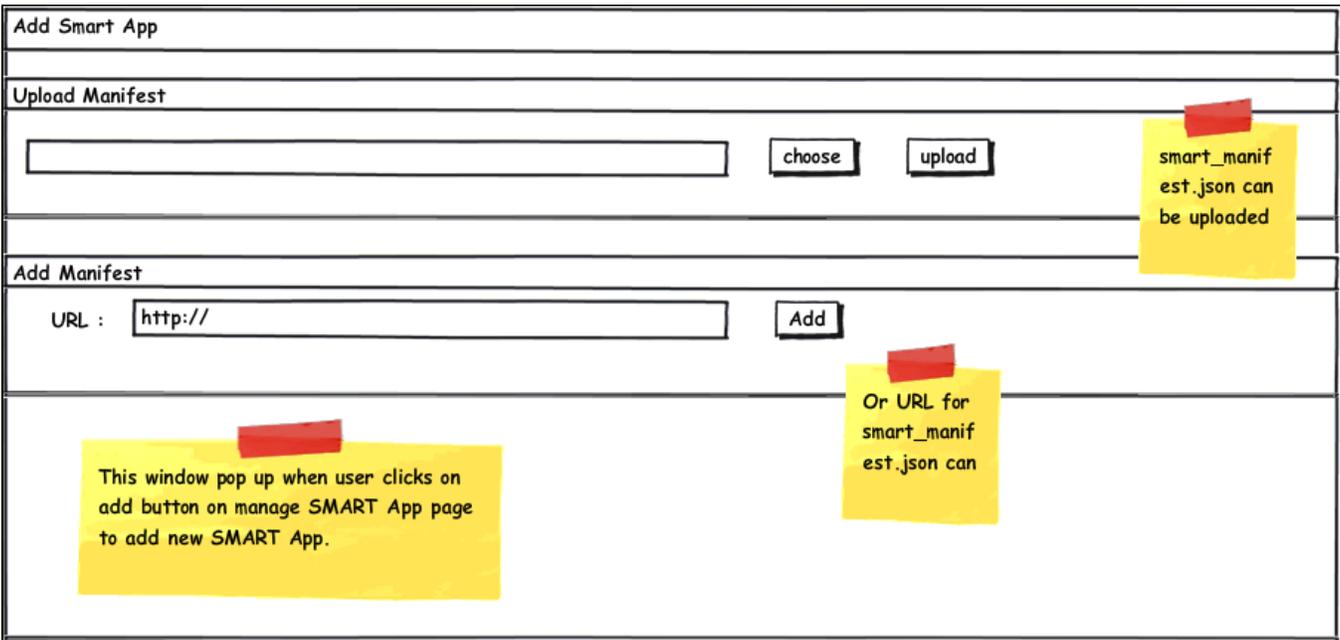
Database Table Design

The module has to manage installed application and user.The information regarding apps and user have to be modeled to save in to the database. OpenMRS already has user object(can be found here),so the details about this object is avoided.
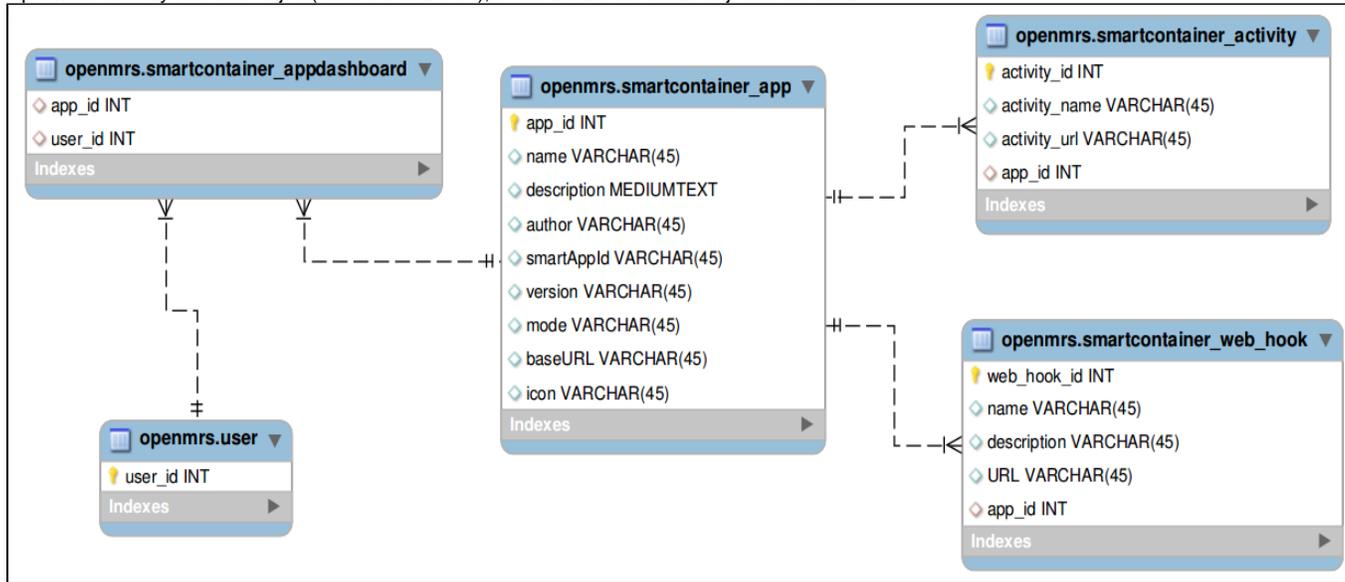
Planning

Project Mile Stones

- Implementation of data model
    - Creating required database tables --Done
    - Creating required basic classes(App,AppService....) --Done
- Implementation of UIs
    - Installing module will create
        - A tab in patient dashboard --Done
        - A horizontal tab in user option page --Done
        - A link in Admin page --Done
    - Creating main page at patient dashboard --Done
    - creating manage app page at user option page --Done
    - Creating manage app page and app upload window at admin page --Done

- Implementation of SMARTAPIService
    - Creating RDF Service --Done
    - Creating Concept Mapping Service --Done
    - Creating method to access patient demographics, drugs and problems --Done

(Above goals will be achieved before midterm evaluation)

- Enhancing SMART API --Exposed Lab Result and Vital Sign**
- Adding Extra methods to SMARTAPIService and Controllers --Done
- Adding Authentication --Instead of adding authentication the code was reorganized
- Documentation
    - Technical design documentation --Done
    - User documentation

Extra Credit(If time permit)

- Develop methods for access to users --Done
- Develop methods for access to allergies
- Develop methods for access to labs --Done

Detailed Time Line

	Time	Goal
1	May 23-May 30	Implementation of data model
2	June 1-June 15	Implementation of UIs

3	June 16-July7 July 10	Implementation of SMARTAPIService submit midterm evaluation
4	July 18-July 31	Enhancing SMART API
5	August 1-August 7	Adding Authentication
6	August 9- August 15	Documentation
7	August 15	Pencil down