ReHAP: Rehabilitation Healthcare Analytics Platform

Project 11 Final Report, EN.600.446

David West
dwest25@jhu.edu, drwest15@gmail.com
Mentors: Dr. Krishnaj Gourab, Michael Cohen, Gorkem Sevinc, John Adamovich

Abstract

Rehabilitation therapy is experiencing trends in increasing demand that leave are driving up the cost of services and decreasing patient care. Therapy teams in both hospital and home-care settings struggle to meet this growing demand, and hospitals would greatly benefit from more efficient operations. ReHAP is a web-based decision support system for patient rehabilitation teams. The software employs algorithms that help physical and occupational therapists to optimize care efficiency. In a proof-of-concept study using a MATLAB prototype of the algorithm, ReHAP's saved on average 20 minutes per therapist per day, saved therapy coordinators 150 minutes per day, and decreased the number of high-priority patients not receiving care. This project implemented this approach to be deployed in clinical settings as part of a pilot program beginning in the summer of 2016.

Problem and Significance

Owing to an aging American population, the healthcare expenditure for providing rehabilitation therapy service (RTS-physical and occupational therapy) is projected to increase by $7.5 billion for patients in the hospital, and $32.3 billion for patients outside the hospital over the next decade.[1] Due to a shift towards bundled payments, decreasing the amount of RTS delivered to patients in a high-cost setting (i.e. in the acute hospital) will translate to a net saving for hospitals and payers. At the same time, increasing the number patients receiving RTS in the post-acute care setting (i.e. in an inpatient rehabilitation unit, or in the outpatient clinics) is a net revenue generator for most healthcare systems. These financial considerations also have to be balanced with the need for providing appropriate RTS at each setting, in order to maintain optimal clinical[2,3] and operational[4] outcomes.

To address the increasing patients population, hospitals or other RTS providers could hire more therapists, but increase costs further. Alternatively, a hospital could attempt to reduce costs by decreasing therapy for patients, but thus resulting in poor outcomes, including fuction loss or post-acute care readmission. The ideal strategy to address these problems would be to make operations more efficient. Currently, day-to-day RTS operations are performed by therapy managers, often using raw EMR data, Excel, and pen and paper (or whiteboard). ReHAP seeks to automate and optimize a laborious and error-prone process.

Technology and Approach

Environment

The ReHAP application is built in Ruby on Rails. Because security and compliance are critical requirements of the system, the application was built on a secure, HIPAA compliant server. ReHAP will be tested in other hospital environments, and is configured to be deployed on other secure servers. All patient information is stored in a Postgresql database configured within the rails environment.

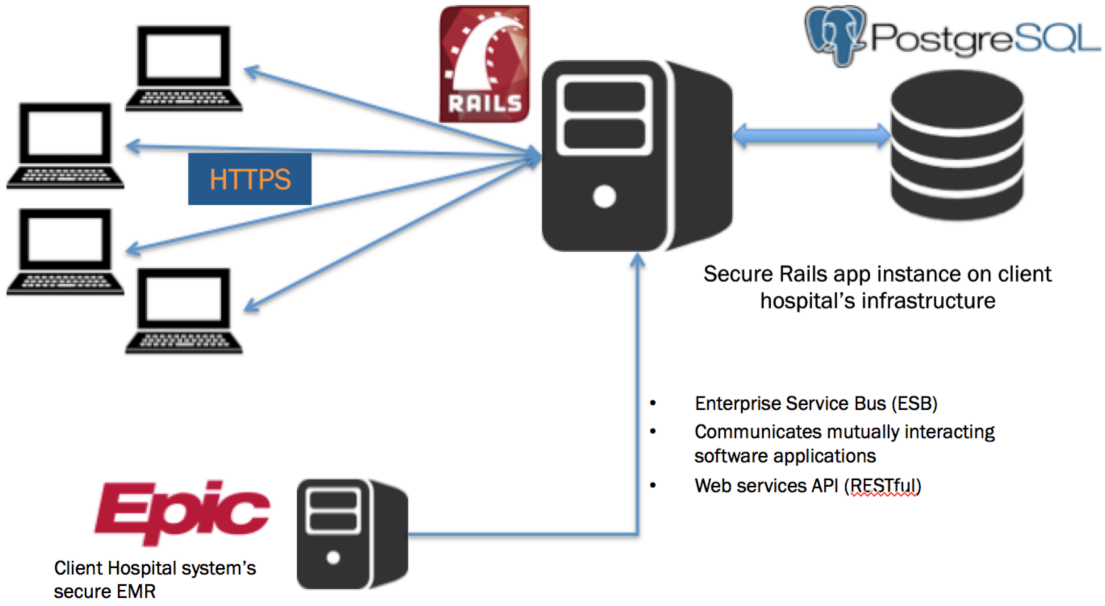


Figure: High-level overview of application architecture

Data Capture

The initial intent of the project was to pull data from directly from Epic. One dependency was the cooperation of the Web Services team at JHMI to set up an API for the application to use via an enterprise service bus (ESB) architecture. Slow-downs on the administrative end prevented this. However, the system was built with this in mind. Thus, it is easy to modify for ESB integration. While an optimal application would automate the data-pull process, the interim solution is to upload csv snapshots of the EMR. Stored in the database are 125 "dummy" patient records that were uploaded in csv form. New data can be easily uploaded, and the application automatically parses and enters into the database according to the schema.

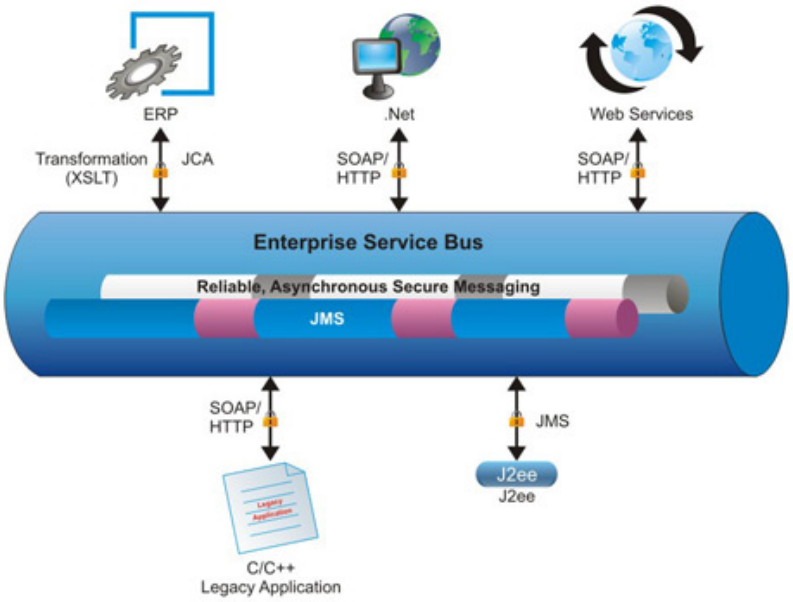


Figure: Enterprise Service Bus. Application is configured to use this (with approval and testing)

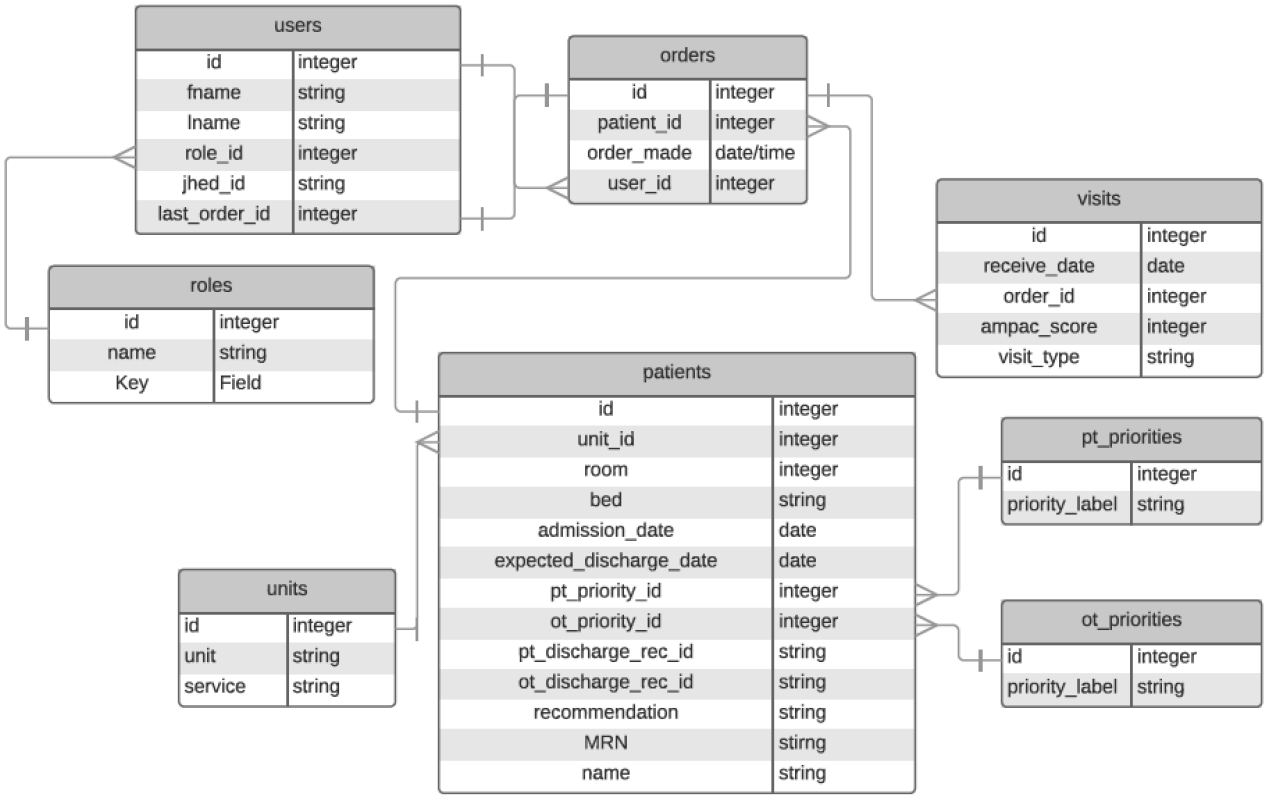
Schema

Designing the database schema was a lengthy and iterative process that involved heavy amounts feedback between the technical side (David, Michael) and the clinical side (Krishnaj, Ruben, therapy team). The schema is well documented in the project repository, but it's worth noting a few key things in summary. In rails, each table is a "model", and the database is abstracted and interacted with in Ruby as opposed to SQL. Data input is centered around "patients", with parsing done in the patients model (patients.rb has a method *.parse_data*() that moves raw csv or EMR records into the database). Each patient record has a unique MRN number, and will be admitted to a stay in acute or post-acute care one or more times. Each admittance is an "**order**". So in sql, the relationship is a "has many" relationship – a **patient** has many orders. We define an order by a "last admission date". When we see a new admitted date for a patient, we create a new **order**. Therapists see

MRN
Exp Disch Date
PT received date
OT received date
OT AMPAC RAW SCORE
Room/Bed
PT AMPAC RAW SCORE
Patient name
Unit
PT Priority
Last Admission Date
OT Priority
OT
Admission date
PT
OT Freq
PT Freq

Table: Fields pulled from each EMR record

patients many times, in their stay. An **order** *has many* **visits** (from their assigned therapist). This patient has many orders has many visits paradigm is key to understanding the ReHAP database.



Overview of algorithm

ReHAP software employs prioritization algorithms that order and sort patients in a therapy case workload according to multiple factors. In summary, the algorithms primarily consider:

- Unit – where the patient is in a hospital
- Lag Time – when a patient was last seen
- AMPAC Score – the performance of a patient

Lag Time

Perhaps the most critical factor that ReHAP considers is lag time. The lag time is defined as:

(today's date) – (last PT/OT date)

AMPAC Scores

The algorithm is highly dependent on ability to measure patient performance. The AMPAC score, or Activity Measure for Post Acute Care, has become widely implemented system in major hospitals in the US in recent years. Developed by researchers at Boston University, the system seeks to standardize mobility assessment in post-acute settings, though is widely used in acute settings as well. At Johns Hopkins and other hospitals, AMPAC Scores are tracked in electronic medical records. Thus, ReHAP algorithms leverage AMPAC scores as the sole factor for patient performance.

Categorization

Patients are sorted into categories based first on units (location), then sub-categories based on other factors, most importantly lag-time. For example, patients with a lag time approaching a threshold (usually 3 days) and over a threshold are sorted into sub-categories. Patients over the threshold are flagged (indicated in red in the app). Within those sub-categories, patients are ordered by mobility. Moreover, the algorithm considers patients with high mobility and low lag-time as being over-seen or ready for discharge. Furthermore, some patients with low mobility and high lag-time will be flagged. A detailed enumeration of the algorithm logic, categorization, flagging, and prioritization, along with clinical implication is shown below.

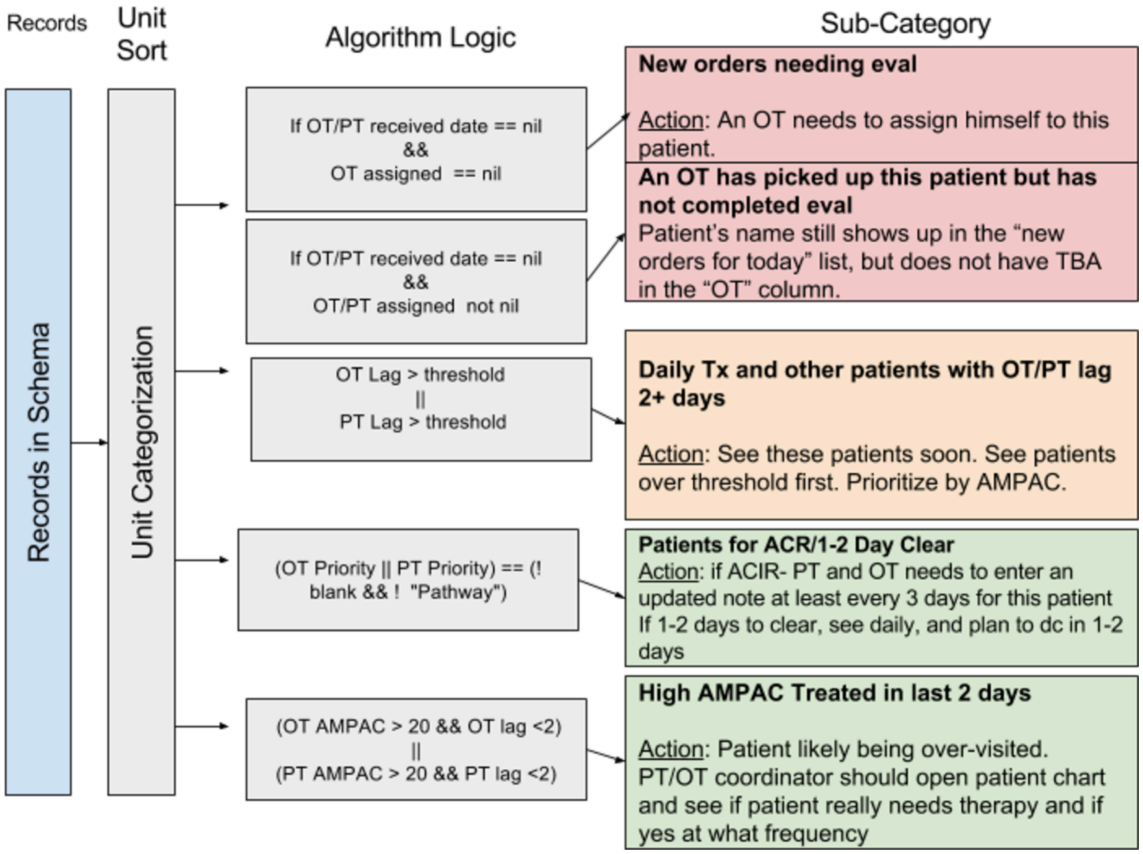


Figure: Algorithm Diagram

Interface Design and Results

User experience and interface design was a critical component of this project. The goal is to be used in clinical settings, and thus there is a motivation for usability. I spent time shadowing therapy teams and working directly with a therapist to understand their workflow. Because the application is web-based, it can be accessed anywhere, which is important as the therapists spend their day on their feet. Designing for the web requires different considerations that the matlab prototype. ReHAP adhere's to the Material Design framework popularized by Google.[5] ReHAP uses a plugin Material Design Lite, similar to bootstrap, to streamline the html and css. There are three main views in the current status of ReHAP. The main view is the therapist dashboard. This dashboard includes a list of all hospital units and a summary of information (caseload). Users can click on a unit and see all the information and the patient categories (generated by the

algorithm) for that unit. The initial prototype and mockups did not have this pre-filtering. It simply listed all the patient level information by hospital units. The downside to this is that it required quite a bit of scrolling. However some early usability design feedback from therapists suggested that some preferred the scrolling. To fix this, both solutions were built, with the ability to perform A/B testing in the future. The floating action button at the bottom of the unit view (/dashboard/therapist) takes the user to a unit-aggregated caseload view.

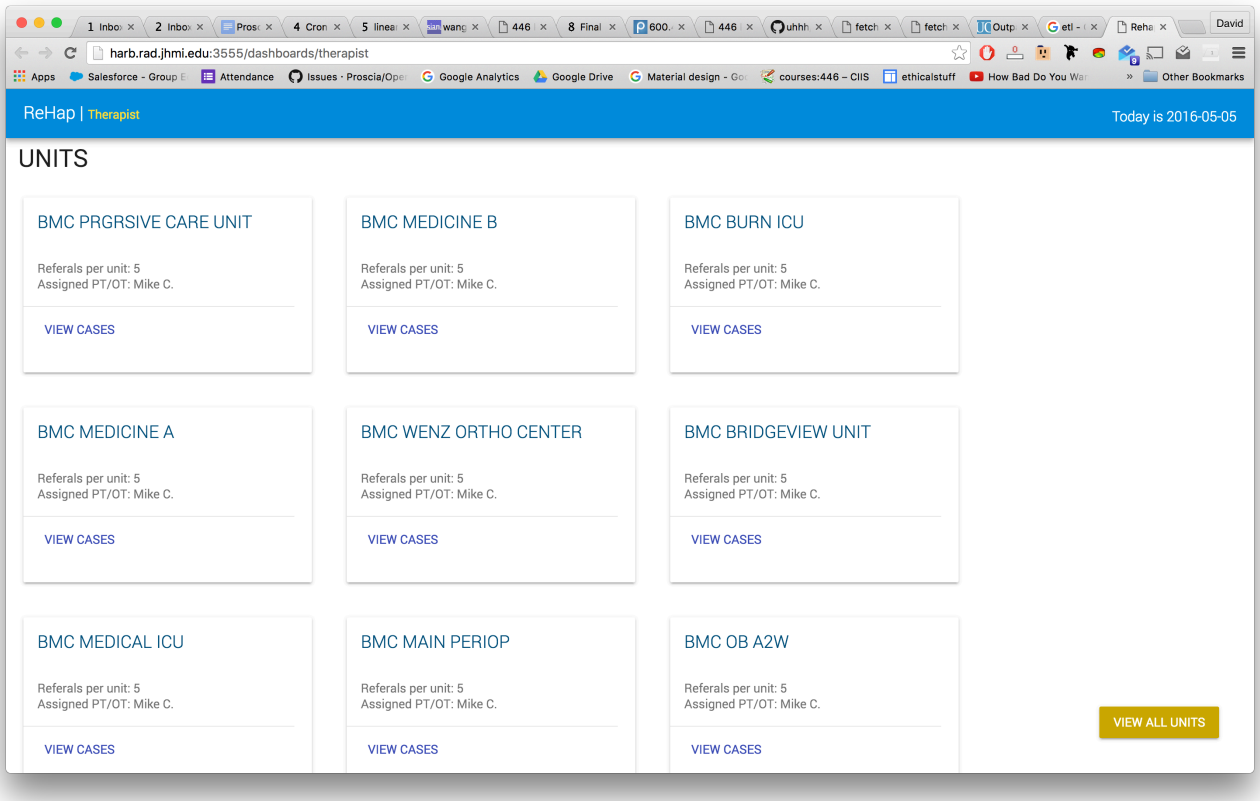


Figure: /dashboard/therapist view. Summarizes hospital units and allows users to filter. Clicking a unit brings the user to a single unit view at /dashboard/therapist/unit/:unit_id where :unit_id is pulled from the database.

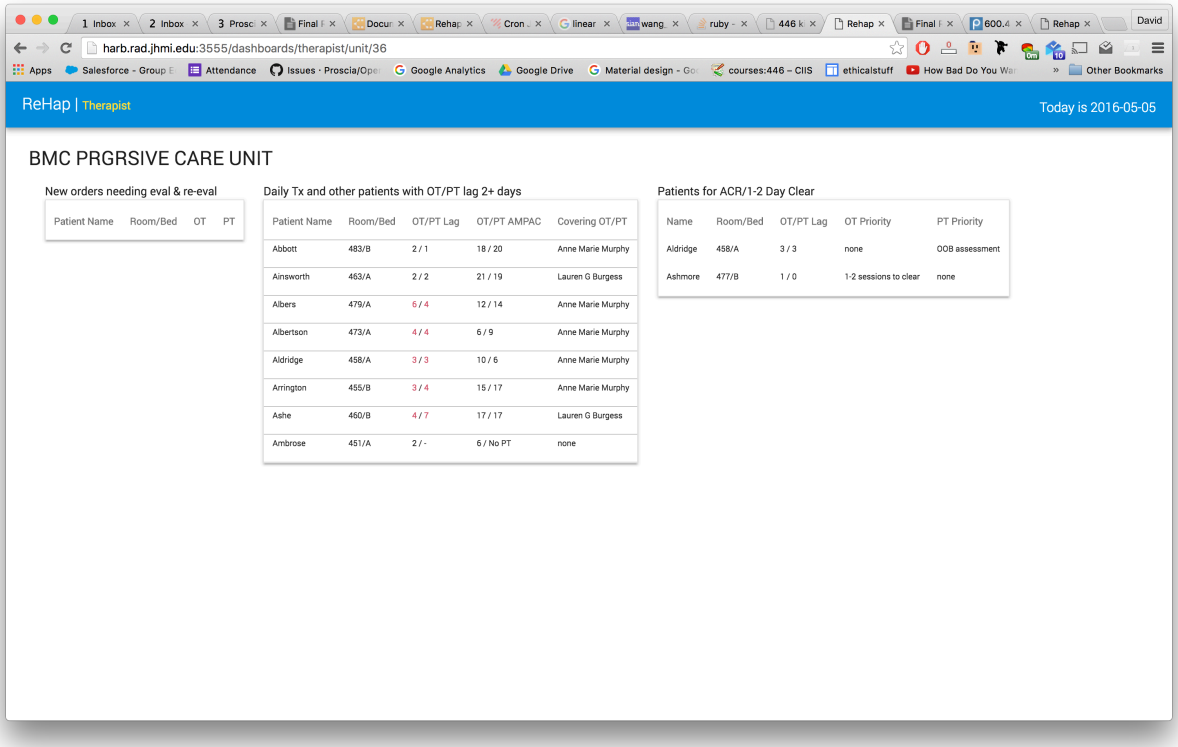


Figure: /dashboard/therapist/unit/:unit_id. Shows patient information for a single unit. Note the three tables representing sub-categories of patients. Also note the red flagged cases with lag times over the threshold. A future implementation will have dynamic thresholding.

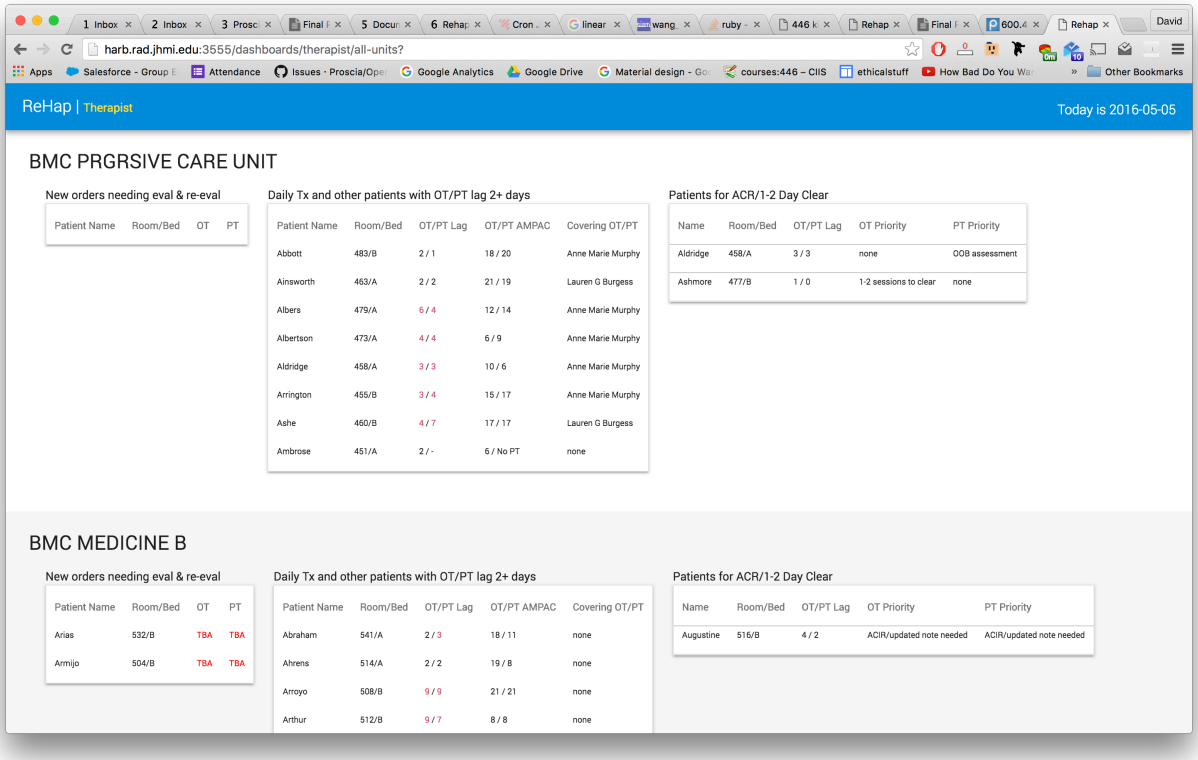


Figure: /dashboard/therapist/all-units view. Shows the units in aggregate in line with the original prototype design.

Management Summary

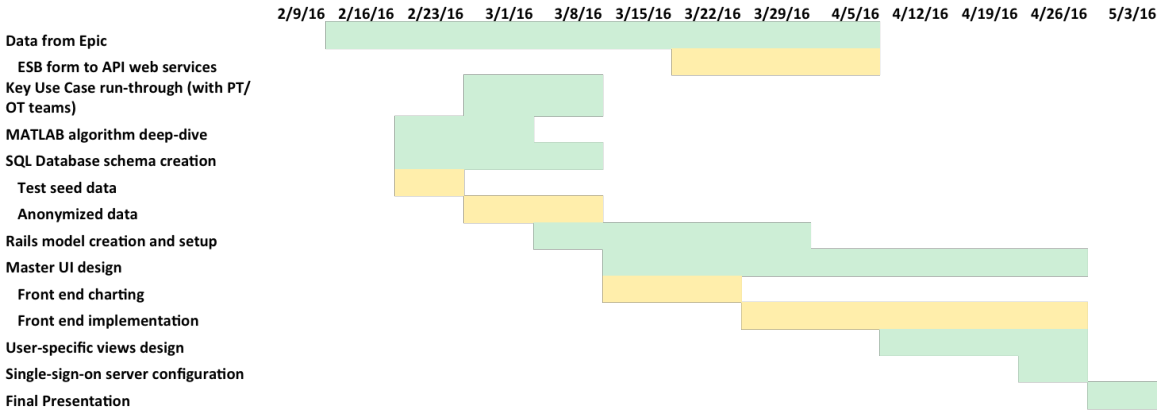
ReHAP was a one-man technical team, with the help of some fantastic and very supportive mentors. Originally, I thought I would be joining a team with another developer, and set my deliverables with reasonable ambition. This other developer, not part of the class, left ReHAP, thus significantly altering my schedule. Below is the original deliverables set (though note, it is not quite phrased in the form of deliverables as much as to-dos, a lesson learned from Dr. Taylor during one of my presentation). The completed deliverables are bolded.

- Minimum
  - **Create and populate database**
  - **Implement Rails app framework**
  - **PT/OT view and permissions**
  - **Shadow-informed front-end mockups for all users**
  - **Prioritization algorithm implemented in Rails app**
- Expected
  - Manager view + permissions
  - ESB set up and integrated with Rails app
  - Refreshing every 5 mins
  - **Single-sign-on server configuration**
- Maximum
  - Physician view + premissions
  - Deployed at JHBMC
  - **Tested by PT/OT teams at JHBMC**
  - **Instances deployed or configured to be deployed at non-JH facilities**

The last two are not quite complete, but the app is at the point where they *can* be. The rest of the ReHAP team from TIC has done a great job setting up pilot institution to begin using the system. Two of the missed expected deliverables are due to Web Services to provision API integrations for us. We are still waiting on them, but the system is prepared to accept that information with slight modifications. Missing the manager view was simply a development resource constraint stemming from the aforementioned teammate

issue. When this other rails developer left, I was unsure the minimum would be achievable, and so I'm personally very satisfied with the progress made – it is incredibly fulfilling that my work will be able to be used in clinical settings.

Interestingly, the Gantt chart was followed incredibly closely with the exception of the ESB integration. All other goals were accomplished very close to the timeline planned. I used JIRA to manage sprint planning and the Git (via Bitbucket).



With the pilot users coming on and impending commercialization milestones, there was a strong incentive to move fast on preparing the application – I felt that was my duty to the team. Documentation suffered a bit (include that which was required for the class), but was overall to the point where a new developer could reasonably pick up where I left off. The code was commented in many areas, with ample TODO's and suggestions for future developers. I will spend the next few weeks all the documentation is all tied up. I plan on spending my free time in the summer after graduation continuing this project. I loved the team at BMC therapy and TIC!

Appendix

1. Physical Therapists: Occupational Outlook Handbook: US Bureau of of Labor Statistics
2. Occupational Therapists: Occupational Outlook Handbook: US Bureau of of Labor Statistics
3. Sleiman, I., R. Rozzini, P. Barbisoni, A. Morandi, A. Ricci, A. Giordano, and M. Trabucchi. "Functional Trajectories During Hospitalization: A Prognostic Sign for Elderly Patients." *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences* 64A.6 (2009): 659-63. Web.
4. Peiris, Casey L., Nicholas F. Taylor, and Nora Shields. "Extra Physical Therapy Reduces Patient Length of Stay and Improves Functional Outcomes and Quality of Life in People With Acute or Subacute Conditions: A Systematic Review." *Archives of Physical Medicine and Rehabilitation* 92.9 (2011): 1490-500. Web.
5. https://www.google.com/design/spec/material-design/introduction.html