John Hofacker
COS420 - Childhood Immunization
Iteration 2 Report
March 8, 2020

Work done by each team

Team 1: Calen Cyr, Alexander Millet

Use Case

Team 1 was assigned the use case of a patient returning to the immunization registry. What this means is when a patient who has already been to the Immunization Registry comes in for a subsequent visit. Their previous form must be taken from permanent storage, updated, and then stored again.

Refactored Code

At the end of the last iteration the classes used to store files in permanent storage were mostly stub code and didn't actually do what they were supposed to, team 1 went and changed it to provide function. Team 1 also refactored the fields for our immunization registry class to match more directly with the form and reduce overlap with the consulting registries patient class.

Team2: Marie Hartung, Fazil Shaikh

Use Case

The use case that was assigned to team 2 was about when a patient shows up for the first time to the immunization registry. This use case details when a Patient who is already registered with the consulting registry comes in, but has not yet been to the Immunization Registry. A new form must be created, filled out and stored in permanent storage.

Refactored Code

When iteration 2 started there were some issues with imports or classes referencing other classes that didn’t exist or were deemed unneeded, team 2 sorted out these errors. There was also controller logic mixed in with various services especially at the top menu. Team 2 sorted out these misplaced functions and reworked the classes to fit into a more logical scheme.

Significant accomplishments for iteration 2:

In iteration 1 we focused mainly on the idea of a “happy path” the most common use case of the software assuming there are no special cases such as data entry error. At the start of this iteration we realized that the happy path we were moving towards actually involved two different use cases. Those were the two use cases the teams worked on this iteration, the first time immunization registry patient and the returning immunization registry patient.

Team 2 was able to complete the use case for a patient's first visit to the immunization registry. Team 1 made significant progress in their use case but certain entries for what immunizations the patient got throw errors, so these problems will need to be polished out next iteration.

As well as these two use cases a good deal of code was refactored in order to streamline the current function of the program and make it easier to scale up going forward. New diagrams were also created to reflect the changes in the system sequence.

Milestones and time estimates / actual time spent

Milestone	Time Estimate	Time put in	Completeness
Completion of Immunization Registry Class	2 work hours	2 work hours	100%
Successful storage and retrieval of Immunization Registry Class from .json	2 work hours	1 work hour	100%
Patient First Visit to Immunization Registry	12 work hours	9 work hours	100%
Patient returning to Immunization Registry	10 work hours	10 work hours	75%
Consulting Registry Integration	3 hours	2 hours	50%

Short Range Plan

The highest priority thing going into iteration 3 is to complete the integration of the Consulting Registries provided package. As it stands right now the jar file is imported into the project and functions when methods are called on it. However our current model of a patient needs more fields to actually take this information and store it.

Long Range Plan

In the grand scheme of the project what needs to be done from here on out is to stop integrating the less than ideal use cases. As the program stands right now it is able to provide function for both of the use cases that we identified as the happy path and most common use of our service. Next we need to identify what these other use cases are and assign priorities to them.

Risk Assessment

- Identification of more use cases
- Monthly Returns form
- Swing UI integration
- Full consulting registry integration

Issues and Code that may need Refactoring:

Right now a lot of our service assumes that the user enters valid data, in the future we will have to go in and do more error checking. Use case for returning patients also needs to be completed and looked back over since the coding of it was somewhat rushed near the end of the iteration.