

Dejan Tisma

Iteration 2 Recap

Work Done:

The first programming team was Cam and Joe, and Josh was the solo programmer for this iteration. First we discussed current architecture and workflow, after we all were on the same page, we continued to discuss what makes sense for a new user story or functionality to the system. We certainly agreed Swing would be implemented, and also agreed that we should focus on features pertaining strictly to antenatal care. This would be medical information like gestation, parity, fundal height, etc. and basic medical information like height, weight, blood pressure and more (since we didn't have this at the end of iteration 1). This medical information needed to be incorporated into the current patient model (which turned out to be the consulting register model) as well as the visit model. We also had to deal with the problem of patients having multiple pregnancies, we needed that stored somewhere.

Cam and Joe at a high level worked on the 'Add visit' use case. Which turned into Pregnancy (model, dao, service), building on the old BaseVisit model for a Visit model which also had dao, service, and a controller. When a patient is searched, a new visit object is created. If there's previous information from a visit (under the same pregnancy), the visit UI is pre-populated with those numbers. After the visit form is filled in, that visit object is updated and saved to the dao.

Josh at a high level worked on the 'Display patient' use case. Halfway through the iteration the Consulting Register pushed out a package pertaining to a Patient model and a search feature. Josh implemented our medical information (model, dao, service) to tie to the CR patient via NHIS id, this was demonstrated through the home tab search feature (which also gave our application an entry point). Similar behavior as above where the UI is populated and ultimately can manipulated the dao.

Accomplishments, Issues, and a look ahead:

There's certainly a lot of accomplishments since this iteration added a lot to the system. The biggest accomplishment was getting a Swing application working with lots of functionality! This was a big risk for us since none of us used Swing before-hand. The prototype certainly speaks for itself in terms of what was accomplished, especially compared to the CLI of iteration-1. Current issues are mostly just design / architectural problems. A shared 'MainMenu' resource as opposed to multiple views is one issue. This was the root of a merge conflict, although it was simple to append new code, it likely isn't best practice. Personally, I think that the open-closed principle applies here "software entities should be open for extension, but closed for modification," so that's a problem we face. At a high level the solution seems obvious, but we need to implement it in code. During this iteration (especially since our iteration wasn't perfectly aligned to material in class) we deemed it fine, but it's something we plan to address.

In terms of looking ahead we also plan to add more medical information. Things like Sickling, TT, PMTCT, etc. will likely require complex classes and perhaps collaborations from a team such as Immunization. The reason we opted for things like height, weight, blood pressure, etc. is because they were primitive types and easy to implement while we were using Swing. This goes back to that idea of

coverage, where we can easily add new fields and input if we have similar functionality already. We have already discussed on making the patient window read only (can't edit medical information) instead it's updated dynamically via data from visits.

Risk Assessment:

We mitigated conflicts via architecture and workflow during the first meeting. A big risk mitigated was Swing. This speaks for itself, but it was a brand new framework for all of us and even a different way to code for some (action/event based programming). A new risk that appeared was connecting the patient from the Consulting Register to work with our current system. Alongside this came issues with compiling and running code on different machines and environments. These are issues dealing with project settings/preferences and local java (JDK and JRE) versions.