

Work Done

The main focus of this iteration was to, for all intents and purposes, complete any work on the application that did not pertain to transitioning to the cloud. This included refactoring existing code to cut down on redundant data storage and inefficient class structures, handling user input errors, and polishing the GUI to provide a more intuitive user experience. We also focused on completing our remaining use cases, namely displaying a list of previous visits for a particular patient and preparing aggregated data for inclusion in the Monthly Midwives Report.

Team 1, Cameron, was assigned the Monthly Report use case, which involved adapting the code written last iteration to display the aggregated data to the user, rather than just exporting the results to a file on disk. More specifically, Cam designed a new tab view which populates the user's requested monthly report and displays it in a clear and concise fashion, using the back-end aggregation methods that were developed in Iteration 3.

Cam also contributed to the refactoring of the domain classes, specifically the Pregnancy object and its related service and methods. Previously, the "getActivePregnancy()" method returned a list of all pregnancies related to a patient, with the most recent pregnancy at the front of the list. This was refactored to only return the most recent active Pregnancy object. He also removed the duplicate storage of Hemoglobin fields in the Pregnancy and Visit objects; that information now resides only in a Pregnancy object.

Team 2, Dejan and Josh, were assigned the List Visits use case, the last major feature to be added to the application. At a high level, this required a major overhaul to the GUI and logic on the back-end to support this new functionality. This work also included adding granular error checking and verbose messaging to the Visit input tab.

Dejan and Josh were also tasked with refactoring the application's Controller structure into a known design pattern. Early in the iteration we identified two patterns, Facade and Mediator, which could fit the bill. With this in mind Team 2 implemented a Facade/Front controller which reduced coupling and the complexity of the underlying subsystem. They also folded small patient subclasses, such as shots and tests, into the PatientInfoModel class.

Accomplishments

We were quite happy with our accomplishments during this iteration, as we successfully completed each of the goals we set for ourselves within the time that was initially estimated. It's my feeling that the iteration went smoothly and posed no serious risks, as we identified and mitigated large, time consuming risks in earlier iterations. We feel quite prepared to focus on the transition to Google Cloud and Postgres.

Risk Assessment

As mentioned in the Accomplishments section, most if not all of the major risks were mitigated in earlier iterations, leaving us to focus on bringing the project to completion. We ensured productivity and the understanding of design patterns through early and constant communication within the group, meeting frequently via conference call to plan, brainstorm, and glean the differences between the Facade and the Mediator patterns.

One major risk that we identified during this iteration is that we felt the functionality provided by the Consulting Register was lacking, possibly resulting in a lot of unexpected work to implement our own full patient search. As it stands, we have a package that allows us to search the Consulting Register, but no common UI or ability to add, edit, or delete patients. Further, there is no unique, forward-facing, non-nullable field in the Patient class, forcing each team to implement a list of possible search results rather than being able to immediately pull a particular patient. We feel that we exhausted our mitigation options for this risk; the Consulting Register was reluctant to provide any added functionality, a UI, or a new, unique identifying field in the Patient object after multiple requests and suggestions from different teams. We feel that at this point in the project, completely redesigning how we access patients and building a new GUI is impractical, and frankly the responsibility of the Consulting Register. We will continue pulling patients from the Consulting Register by NHIS number, unless we are advised by the stakeholder to proceed differently.