

## **Antenatal-Care-Team**

**Team Leader: Hamza Zmou**

Iteration #4

### **Overview of the iteration**

For iteration # 4 the focus was on refactoring the classes and creating the midwife monthly report. Integration was supposed to be done during this iteration, but due to the amount of issues that was found in consulting register base code, this will be done in the next and final iteration. A pdf file listing all the issues found in the consulting register code base is under the documentation.

The following is a list of major issues that have been identified for this iteration:

- Refactoring the controller and splitting out controller by view.
- Refactoring the service by making an abstract Service.
- Refactoring class names.
- Change the patient wrapper to a list of pregnancies, by making pregnancies a list.
- Create a midwife monthly report that generates a file that outputs the report as specified by the documentation.

### **Team 1: Dan and Zack- Work completed**

Dan and Zack worked on refactoring the view, by correcting misspellings and catching the nulls. Moving commonly duplicated code blocks and showing how the new visit button on the GUI is supposed to be used. Dan worked on refactoring class names by changing the Antenatal Model class to Pregnancy class and renaming the RegisterView pregnancy related name. The new view classes are now: PregnancySelectionView.java and PregnancyView.java. This team also built a workaround system that allows basic selection of a patient to use with the application. Most of the work done by this team was done in a paired programming style.

Time estimates: 7hrs of paired programming on the issue. Estimate time was 6hrs.

### **Team 2: Jordan and Riley – Work Completed**

Jordan and Riley worked on refactoring the controller and service classes. The controller was split to handle the two service classes PregnancySelectionService and PregnancyService. The controller logic becomes a handler for the two service classes. This team also worked on refactoring the service by making it abstract with a patientWrapper interface and removed Antenatal Service. Jordan and Riley worked also on making the patientWrapper a list of pregnancies. Unit testing for the new service classes was also done by this team. Most of the issues tackled by this team were done in a pair programming style.

Time spend on issues by this team:

#55 Estimated time: 8 hours - Actual time 11 hours

#54 Estimated time: 2 hours - Actual time: 2 hours

#51 No time estimate - Actual time: 15 minutes

#50 Estimated time: 2 hours - Actual time: 1 hours

### **The midwife Monthly Report – work completed**

Zack and Hamza worked together on building the Midwife Monthly report following the same specifications that are defined by the documentation. This class under service generate a report text file that output the midwife monthly report just like specified by the documentation. It retrieves all the patient Wrappers with at least 1 pregnancy in the past month, aggregate data from the past month of pregnancies and generates a text file that contains the report. A button for generating the report was also added to the GUI by Zack.

Task done in a paired programming style

Time estimates: 2hrs working session to build the class. (Task done in a paired programming style)

### **Fixed bugs in this iteration**

The following bugs were fixed in this iteration:

- Parity in midwife Monthly report prints null. Issue resolved by Zack PR#68
- Entering data for a single patient wiped all other patient data. Issue was resolved by Zack PR #64
- When a pregnancy is selected right after adding a visit the tabbed visits are duplicated and it's impossible to proceed. Issue was resolved by Zack and Dan

### **Design patterns used in this iteration**

- The overall design pattern for our project is model view controller (MVC)

- The service is designed as a façade structural design pattern with an interface that hides the complexity of the other service classes that implement it.

### **Upcoming work and conclusion**

We have an optional feature that we want to add to the next iteration which is to implement notifications for new shots (issue # 49). Unit testing for the midwife monthly report and the data integration with other teams are also going to be done in the final iteration. Work done in this iteration consisted mostly of refactoring of the view and service classes to better prepare for cloud migration. Midwife monthly report was also implemented and integrated to the GUI during this iteration