Antenatal Care Iteration 2 Report

**Team Leader: Kyle**

**Team 1: Aaron Harvey, Naomi Heuer**

**This team was focused on satisfying the use case: “CHO searches for past antenatal exams of a patient.” This involved modifying the GUI to include a process for accessing previous pregnancy records (and their associated subsequent visit records). In addition to the changes to the GUI, we also needed to integrate changes to the controller in order to retrieve objects and particular data points in order to populate the fields in our pregnancy record form when retrieving existing records.**

**Team 2: Nicholas Curlew, Ross McLain**

**This team was focused on satisfying the use case: “CHO conducts a follow-up exam and records new information into the register.” This involved creating a class, controller, service and DAO for subsequent visits. In addition, this involved creating GUI elements to help create these subsequent visit objects.**

**Use Case List**

\* CHO conducts first antenatal exam for patient and records information into the register.

\* CHO conducts a follow-up exam and records new information into the register.

\* CHO searches for past antenatal exams of a patient.

\* CHO updates information for an exam.

\* CHO requests a Monthly Midwifes Report.

\* System alerts CHO when the next exam should be scheduled.

\* System alerts CHO when values are not within healthy ranges.

\* System alerts CHO when a patient has missed an appointment.

\* System prompts CHO to deliver critical information.

**Iteration 2 Use Cases:**

**CHO conducts a follow-up exam and records new information into the register**

1. CHO opens Consulting Register, enters the patient search information and finds account.
2. Once found, the CHO opens the Antenatal Care Register.
3. Consulting Register panel above Antenatal panel populated with data.
4. CHO conducts exam and fills out new Subsequent Visit information.
5. System alerts the CHO if any fields are missing.
6. CHO adds missing information if necessary.
7. System alerts the CHO that the Subsequent Visit is complete and saved to storage.
8. CHO views complete Subsequent Care form and reviews Pregnancy Record information and looks for any optional fields not filled out.
9. CHO screens patient and fills out any new fields discovered to be valid (for example, Venereal Disease Lab Results, once the patient has had the test after the initial antenatal visit).
10. Once applicable fields are filled out, CHO submits info.
11. System alerts the CHO that the Pregnancy Record form has been updated.

**CHO searches for past antenatal exams of a patient.**

1. CHO opens Consulting Register, enters patient search information and finds account.
2. Once found, the CHO opens the Antenatal Care Register.
3. Consulting Register panel above Antenatal panel populated with data.
4. Display Subsequent Visits below Pregnancy Record.

**Tasks:**

\* Ensure Input Validation

- Ensure input types are valid

- Produce Error message popups to inform user of invalid entries

\* Expand program to include interface for searching for an existing patient's Antenatal Care information

- Expand GUI to include interface for search.

- Connect new GUI interface to controller for database search.

- Pull data from database and populate fields on screen.

- Allow for user to update currently-empty fields from this view.

\* Expand AntenatalVisit View to include option for adding subsequent visit exam.

- Create AntenatalSubsequentVisit object.

- Add data structure of AntenatalSubsequentVisit objects to AntenatalVisit object.

- Build window with Subsequent Visit fields in GUI.

- Collect and store information from Subsequent Visit window.

**Accomplishments**

* Created a working entry/retrieval system in our software.
* Reached our goal of retrieving an existing Pregnancy Record and creating the framework for Subsequent Visits.
* Updated our documentation and wiki for use in coming iterations
* Expanded out GUI to include the Subsequent Visits and retrieved data from previous records.
* Team collaborated well, meeting frequently and helping keep each other caught up with code and design changes.

Issues

* Did not leave enough time at the end of the iteration for exhaustive testing
* Was unable to complete the retrieval of subsequent visits.
* Unable to ensure complete data validation upon entry – will continue next iteration.
* Closer to using actual person object while designing, but not completely.
* More refactoring needed in GUI controls and display.

Risks

* UI design could prove challenging.
* Shared Person data structure.
* Will need to interact with other sections of the application.
* Error Checking (partially mitigated)
* Learning Swing (mitigated)
* Integrating PostgreSQL could be difficult
* Ensuring subvisits cannot be entered unless an initial visit has been recorded.