CS 427 Software Engineering I
Fall 2018
Final Project Documentation

T804

Group member: Roger Ho, Che-Lin Huang, Bennett Ip, Jae Lee, Shuhan Li, Yidan Li, Dongjun Seung, Fei Teng

Teaching Assistant: Tiantian Fang

Professor: Tao Xie

Table of Content

1. Introduction

This is a final project for the course CS 427 - Software Engineering that focuses on implementing the principles in the field of Software Engineering. Throughout the semester, we worked as a team of 8 on an existing open source Electronic Medical Record - iTrust. We were tasked with total number of 4 use cases to add to the existing codebase by the instructors as though they are our customers. We adopted the method of Extreme Programming, Test Driven Development and Pair programming within the Agile Development methodology.

In order to effectively carry out the development of the system, we split the group into four subgroups with 2 members on each subgroup to focus on implementing one of the four use cases. After each use cases have been correctly implemented and tested, we then merge all use cases together and rerun various tests to form the final working system. We created a shared wiki page to document and track all of our meeting, goals and progress. We also use the Slack application as a means of communication between members of the team to ensure all members could react and communicate efficiently to each other in time.

This documentation also follows our initial development method as we will break down the overall project into four parts, each corresponding to a certain use case while providing a detailed explanation for each individual use cases.

2. Use case 93

i) Description:

This use case revolves around the HCP and the obstetric information of a patient. The use case allows all HCP's to look at the information of all patients who are eligible for obstetric care within the system.

A patient's eligible for obstetric care would have the information as followed:

- The obstetric initializations of the patient:
    - Date of the first day of the patient's last menstrual period (LMP).
    - The estimated due date (EDD).
    - The number of weeks pregnant of the patient from the day when the record is initialized.
- Prior pregnancies:
    - Year of conception.
    - The number of weeks pregnant (how many weeks and days the pregnancy lasted).
    - The number of hours in labor.
    - Weight gain during pregnancy.
    - Delivery type:
        - Vaginal delivery.
        - Vaginal delivery.
        - Vacuum assist.
        - Vaginal delivery forceps assist.
        - Cesarean section.
        - Miscarriage.
    - Whether the pregnancy was a multiple, and how many.

The system also only allows HCPs' who have the specialty of an "OB/GYN" to have the authority to set any certain patient eligible for obstetric care and to create obstetric initialization records, along with adding pregnancy information for certain patients who are eligible for obstetric care through the system.

ii) Architecture & Design:

For use case 93, we created two tables in MySQL, called pregnancy and obstetricsInitRecord in the sql/createTables.sql file and created corresponding beans: PregnancyBean and ObstetricsInitRecordBean. The PregnancyBean contains a patient MID, year of conception, weeks of pregnancy, hours in labor, weight gain, delivery type, numbers of pregnancy. The ObstetricsInitRecordBean contains a patient MID, LMP, EDD, weeks of pregnant and record created time. We also created BeanLoader, DAO, and BeanValidator classes along with Bean to be able to load the Bean object in the jsp file later we used to generate the pregnancy information and initial obstetric record. The newly created tables were also added in sql/data/deleteFromAllTables.sql and sql/dropTables.sql to address the change made in

sql/createTables.sql. New enum values were added in the src/main/edu/ncsu/csc/itrust/model/old/validate/ValidationFormat.java to validate the format of the PregnancyBean and ObstetricsInitRecordBean object.

iii) Appendix:
All tests for this use case are under the directory called "UC_93" that is under the package path of "test/java/edu.ncsu.csc.itrust/unit/T804TestcaseCollections." User's can click on the package and run the unit test or also with coverage to see the results of the test.

To run the manual test on this use case, follow the instructions below:

After choosing a certain HCP user to log into iTrust, click on the "Patient Info" tag on the drop-down menu located on the left side of the screen. Within the drop-down menu of "Patient Info" there are only two tags related to this use case: "Patient Information" and "Patient Obstetric Care History."

The "Patient Information" is used to search a certain patient through a patient's MID or name, and displays whether or not a patient is eligible for obstetric care and provides a button a the functionality of enabling each patient to become eligible for obstetric care based on the system user's specialty.

The "Patient Obstetric Care History" is used to search a certain patient through a patient's MID or name, and display the user's Obstetric initialization records and their past pregnancy information. Only an HCP with the specialty of an "OB/GYN" be able to initialize a patient's obstetric record along with their pregnancy information.

1. To test as a non-OB/GYN specialty HCP, choose to log in as a user named "OptHCP0" or "OphHCP0."

   After typing the name or MID of a patient within the search bar of the "Patient Information" webpage, only the list of the patients MID, First name and Last name would be displayed. The column showing whether each patient is eligible for obstetric care or not along with the button to set a patient to be eligible for obstetric care would not be displayed.

   After typing the name or MID of a patient within the search bar of the "Patient Obstetric Care History" webpage, a list of the patients that are eligible for obstetric care would be displayed with their information containing their MID, First name and Last. The MID button of each patient eligible for obstetric care could then be clicked on to further show the webpage that contains a list of all initialization records of the patient and a list of all prior pregnancy information of that certain patient.

2. To test a OB/GYN specialty HCP, choose to log in as the user named “ObHCP0.”

   In the “Patient Information” webpage, the list of the patients MID, First name and Last name would be displayed, along with another column called “Obstetric Status,” where it would show the status of whether each patient is eligible for obstetric care or not. These two status are presented as “Set Obstetric” in a Blue color font indicating the patient is “not” yet eligible for obstetric care. Another status would be presented as “Cancel Obstetric” in a Red color font indicating the patient “is” eligible for obstetric care and would show up after being searched within the “Patient Obstetric Care History” webpage’s search bar. All the words under the “Obstetric Status” are presented as a button which could be clicked on to change and toggle a patient’s obstetric care eligibility status. When the button with the words of “Set Obstetric” in a Blue color font is clicked on, the corresponding patient would be set as a patient eligible for obstetric care with their status being updated within the database and their names would show up after being searched for within the search bar of the “Patient Obstetric Care History” webpage, whereas when the button with the words of “Cancel Obstetric” in a Red color font is clicked on, the patient would no longer be eligible for obstetric care, and their information would not appear when typed into the search bar of the Patient Obstetric Care History” webpage.

   Furthermore, there would also be a button on the top named “Initialize Obstetric Record.” in the Obstetric Record history. After being clicked on, it would display two table forms, the top one being “Obstetric Initialize Information” and the bottom on being “Pregnancy Information.”
   No fields within these two tables could be empty, or else after clicking the submitting button named “Edit Patient Record,” users would be led to a page notifying a Null Pointer Exception and prompted to try all over again. For the “Obstetric Initialize Information” table, after clicking on the “Select Date” button to choose a certain date of a patient’s LMP in the drop-down calendar, click on the tag in red below the date named “Update EDD” to have the system calculate and fill in the other three fields of “EDD”, “Weeks Pregnant” and “Record Created Date.” After all needed information is filled in, click on the “Edit Patient Record” submit button, the system would save all information into the database and redirect the user back to the obstetric history record page after 2 seconds, and show the newly added information within the two lists. If a patient already has their prior pregnancy information saved within the iTrust system, the fields of the Pregnancy form would be prepopulated by the system when accessed. If there are any replicated information of pregnancy or initialization record data, the system would ignore them when these replicated data forms are being submitted so that each record would remain unique.

3. Use case 94

i) Description:
With the implementation of this use case, an HCP can document or edit an obstetrics office visit for a current obstetrics patient. An ultrasound may also be performed during an office visit. At the end, the next visit will be automatically scheduled for the patient.

ii) Architecture & Design:
Obstetrics office visit is a new set of data that does not exist in the original database. Therefore, we created a new table "officevisitrecord" to store the data needed for obstetrics office visits. This includes:

- Number of weeks pregnant as of the office visit date (generated from last menstrual period)
- Weight in pounds
- Blood pressure
- Fetal heart rate (FHR)
- If the current pregnancy is a multiple, and how many
- Whether a low lying placenta was observed during the visit

Ultrasound record is another set of data that is needed to be stored. We created a new table "ultrasoundrecord" that stores the following information:

- Crown rump length (CRL)
- Biparietal diameter (BPD)
- Head circumference (HC)
- Femur length (FL)
- Occipitofrontal diameter (OFD)
- Abdominal circumference (AC)
- Humerus length (HL)
- Estimated fetal weight (EFW)

We also utilized new data and function implemented by UC93 to determine if a patient is an obstetric patient.

iii) Appendix:
To run the tests for the required implementations related to UC_94, there is a package under test/java/edu.ncsu.csc.itrust/unit/T804TestcaseCollections called UC_94. One can click on the package and run the unit test (or with coverage) to see the results of the test.

To run the manual test on this use case, follow the instructions below:

1. Log in as an OB/GYN HCP.

- An HCP without the OB/GYN specialization cannot create or edit an obstetrics office visit.

2. Click “Office Visit” -> “Document Office Visit”.
3. Enters a MID or name of an obstetric patient.
    - If an invalid patient name or medical identification number is entered, a warning message (“Patient MID or Name does NOT exist! Please renter a valid patient MID or Name”) will be displayed.
4. Click on the id to choose the patient.
    - If the patient chosen is not a current obstetrics patient a warning message (“Not a current obstetrics patient! Please try again!”) and a return button will be displayed.
5. Fill out the office visit information and submit.
    - If any information is missing a warning message (“Please input…”) will be displayed.
6. Click “Office Visit” -> “View My Office Visit”, and click “Edit” on a office visit record.
7. Change some information and click “Change”. A message (“Success: Office Visit Record changed”) will be displayed.
8. Click “Office Visit” -> “View My Office Visit” again, and click “Add” (ultrasound record) on a office visit record.
9. Click “Upload UltraSound Record” to upload ultrasound image, and fill out other information. A message (“Success: Ultrasound Record added”) will be displayed.
10. Click “Appointments” -> “View My Appointments”, and click “Add to my calendar” -> “Google Calendar” to open the Google Calendar page for the appointment chosen.

4. Use case 95

i) Description:
With the implementation of this use case, any HCP may search for the patient by MID or patient name to view a generated report containing all relevant information about the pregnancy. All HCPs can generate this report in case of emergency.

ii) Architecture & Design:
For this use case, we had to gather numerous information from all over the place to generate the required report: UC_93, UC_94, UC_96, and sql/data/icdcode.sql.

From UC_93, UC_94, and UC_96, we gathered the following information.

- Obstetrics Office Visit Information, most recent visit first
    - Weeks pregnant at office visit - pregnancy table (from UC_93)
    - Weight at an office visit. - pregnancy table (from UC_93)
    - Whether the pregnancy was a multiple, and how many - pregnancy table (from UC_93)
    - If a low lying placenta was observed during the visit - (from UC_94)
    - Blood pressure at office visit - (from UC_94)
    - Fetal heart rate (FHR) at office visit - (from UC_94)
    - Estimated Delivery Date(EDD); - (from UC_94)
    - Complications
- Pregnancy complication warning flags
    - High Blood Pressure (pre-eclampsia) - (from UC_94)
- High blood pressure is 140/90. If either value, systolic or diastolic, is at or above those ranges, it is considered a complication

  (i.e., 140/80 is high, 130/90 is high, 139/89 is not high) - (from UC_94)
    - Advanced Maternal Age - patient table
    - This means the patient is 35 or older at the time of delivery (or expected delivery)
    - Low-lying placenta - (from UC_94 Office Visit record)

      Abnormal fetal heart rate - (from UC_94)

- Normal fetal heart rate range: [120, 160] inclusive
- Multiples in current pregnancy - (from UC_94)
- Atypical weight change - (from UC_94)
- Typical weight gain range (in pounds): [15, 35] inclusive

Some pre-existing conditions and pregnancy complication warning flags, including diabetes, chronic illness (autoimmune disorders), cancers, STDs, Hyperemesis gravidarum, Hypothyroidism, High genetic potential for miscarriage were not available from the given iTrust codebase. To address these, we added new ICD codes that we found online in the sql/data/icdcode.sql file.

We created a new table called PreExistingConditionRecord in the sql/createTables.sql file and created a bean called PreExistingConditionRecordBean that contains a patient MID and patient's pre-existing conditions and pregnancy complications that were previously not available from other use cases. We also created PreExistingConditionRecordBeanLoader, PreExistingConditionRecordDAO, and PreExistingConditionRecordBeanValidator classes along with PreExistingConditionRecordBean to be able to load the PreExistingConditionRecordBean object in the jsp file later we used to generate the overall report. The newly created table was also added in sql/data/deleteFromAllTables.sql and sql/dropTables.sql to address the change made in sql/createTables.sql. New enum values were added in the src/main/edu/ncsu/csc/itrust/model/old/validate/ValidationFormat.java to validate the format of the PreExistingConditionRecordBean object.

After all the backend portion of UC_95 was set up, we worked on the jsp file (WebRoot/auth/hcp-uap/viewReport.js) that uses all the relevant bean objects from UC_93, UC_94, UC_96, and the above PreExistingConditionRecordBean object. The jsp file correctly generated a report containing all relevant information about patient's pregnancy with a proper logging statement when an HCP tried to view a patient's overall report.

iii) Appendix:

To run the tests for the required implementations related to UC_95, there is a package under test/java/edu.ncsu.csc.itrust/unit/T804TestcaseCollections called UC_95. One can click on the package and run or run with the coverage to see the code coverage as well. We also created a dummy table called "preExistingConditionRecord1.sql" in sql/data folder to test implementations in PreExistingConditionRecordDAO.

To run the manual test on this use case, follow the instructions below:

After choosing a certain HCP user to log into iTrust, such as "ObHCP0", log into iTrust. Now, click on the menu called "Patient Info" on the left and click on the tab called

“My Report Requests.” An HCP will see a table that says “Report Requests” with an id representing each report request, name of the patient, requested date, and viewed date, and the status notifying whether the HCP has viewed the patient’s info, and the button that says “View,” which will allow the HCP to view all relevant pregnancy information of the patient. Click “Add a new Report Request” to generate a new report for a patient. The HCP will then be prompted to select a patient by name or MID. Put in any number or name that the HCP would like to choose. If the HCP types an invalid medical invalid identification number or name or a medical identification number or name of a patient not in the system, the HCP is prompted to try again by showing an error message, “Patient MID or Name does NOT exist! Please re-enter a valid patient MID or Name.” Suppose the HCP selected a person with MID “1.” The HCP is redirected to the page that shows the table of Report Requests. The system shows that “Report Request Accepted” on the top of the page. Click on the “View” button on the right side of the table to view the patient’s pregnancy information. If the website says “Selected patient does not have an obstetrics record,” then the HCP should click on the “Patient Info” menu again on the left side and click on the “Patient Information” tab. Click on that button and choose a patient with MID “1” to set up an obstetrics record. Click on the button that says “Set Obstetric” on the Obstetric Status to change the status to “Cancel Obstetric.” Now, the HCP can set up an obstetric record for the patient. Click on the “Patient Info” menu and click on the “Patient Obstetrics Care History” tab. Choose the patient with MID “1” and cick on the MID to be redirected to a page where the HCP can initialize Obstetric Record. Click on the “Initialize Obstetric Record” button and fill out the necessary information. After done filling out the information, click “Edit Patient Recor.” Go back to the “Patient Info” menu and click on the “My Report Requests.” Find the previous patient in the table and click on the action “View” again. The HCP can now view all information about the patient’s pregnancy. To see that the patient is ineligible to do the action as described above, logout from iTrust and log back in with any patient, such as “Patient 1.” The patient does not have an option to request a report unlike HCPs.

5. Use case 96

i) Description:

Use case 96 is mainly circled around the main theme of childbirth hospital visit. A patient's childbirth hospital visit is scheduled apriori during an office visit.

- Schedule an appointment and select childbirth appointment type with specific data and time, HCP could choose to add comment if needed.

Then under the childbirth hospital visit tab, there are four functionalities.

- HCP could search for a user's name or MID. A list of patient will show up upon search. After click on the MID of a patient, HCP will be redirected to another page which will show all previous appointment scheduled by the patient.
- There are six types of drugs that a patient can use when giving birth.
    - Pitcoin
    - Nitrous oxide
    - Pethidine
    - Epidural anaesthesia
    - Magnesium sulfate
    - RH immune globulin

  HCP will be able to specify which of these drugs to use at birth time. HCP can choose multiple drugs with no limitation. All recorded drugs will show up when HCP visits this page again.
- There are five ways of giving birth which are
    - vaginal delivery
    - vaginal delivery vacuum assist
    - vaginal delivery forceps assist
    - caeserean section
    - miscarriage

  HCP can specify which delivery method to use during an appointment. HCP will also be able to view this information when the patient comes in at birth time.
- When patient finishes giving birth, HCP can search for the patient's MID. After clicking on it, HCP will be redirected to another page where HCP will be required to input the child's name, birth time, birth date, and parent's delivery method. (In this case, delivery method will display the preferred delivery method during a previous appointment. Otherwise, delivery method will display nothing.) After successful input, HCP can continue to view the information of the new born child. New born child's father or mother information will be captured based on search in the first place.

ii) Architecture & Design:

For use case 96's four main themes,

- Appointment type needs to be modified to take in childbirth appointment. In order to do this, sql table needs to add one more option “childbirth” associated with relative time duration and cost. Then under childbirth hospital visit tab, upon successful search of a user. Using functions from patientDAO, appointmentDAO, we choose to display all related appointment information ordered by date.
- In order to specify what drugs to use, by looking into the patient’s table. We create one column for every drug that will be possibly used. When a new patient is created, this field will be initalized null. But if HCP chooses to specify which drug to use. “1” will be used to indicate usage of a drug and “0” will be used to indicate non usage of a drug. Because there are changes made to the patient table, paientBean also needs to add relative getter and setter functions. PatientBeanLoader also needs to take in additional parameters to work with patinetBean and patientDAO. Finally, in patientDAO, we need to modify it so that it makes the modified mysql command.
- For preferred delivery method, there is also modification needed to be made in the patient’s table. Upon successful search for patient, we create a new column called “preferedMethod” which includes five delivery methods (vaginal delivery, vaginal delivery vacuum assist, vaginal delivery forceps assist, caesarean section, miscarriage). Then, patientBean also needs to add relative getter and setter functions to take in this command. PatientBeanLoader and PatientDAO also needs to be modified so that mysql database can be updated properly when HCP decides to specify the prefered birth method for a patient.
- For create a child’s birth record, upon a successful search of user, we manage to format a form which allows the HCP to input relative information including name, birth date and time, and parent’s prefered delivery method in case anything needs to be changed. To do this, we use the same function as add a patient to the system. But with additional parameters like birth time, because patient’s table in mysql database does not have this column before. Then based on the searched user’s gender, we set the child’s father or mother information accordingly. We also added form validator to check if the information inputted are correct. Upon correct input information, we will create a new patient profile in the system. HCP will be able to continue to make changes to the child’s profile in patient information page.
- Overall, users’ request from the webpage will be processed and information will be gathered in four JSP file for each individual functionality: WebRoot/auth/hcp-uap/childbirthRecord.jsp, childDelivery.jsp, drugUsedRecord.jsp and /preferBirthMethod.jsp. After the information is validated, the next step is to update the users’ information. Get a PatientBean object and modify the patient to data in SQL by using the PatientDAO and through PatientBeanLoader class. Finally, update the webpage to illustrate the corresponding changes.

iii) Appendix:

All tests for this use case are under the directory called “UC_96” that is under the package path of “test/java/edu.ncsu.csc.itrust/unit/T804TestcaseCollections.” Users can click on the package and run the unit test or also with coverage to see the results of the test.

To run the manual test on this use case, follow the instructions below:

After choosing a certain HCP user to log into iTrust, click on the “Childbirth Hospital visit” tag on the drop-down menu located on the left side of the screen. Within the drop-down menu, there are four tags related to this use case: “Childbirth Record at ER”, “Preferred Childbirth Method”, “Drug Used Record” and “Delivery Time”.
Click on the “Childbirth Hospital visit” to view all the appointment associated with a certain patient. There might be childbirth appointment scheduled. If not, schedule a appoint under the “Appointments” tag on the menu.
A HCP can view, add or modify the patient’s preferred childbirth method under the tag “Preferred Childbirth Method”. The current selection will be shown as “Currently preferred childbirth method”. To modify, select one of the options below and click submit. If information update successfully, the notice will indicate “Information Successfully Updated.”
To view or record what drugs have been used during the childbirth, click on the “Drug Used Record” tag. the previously applied drug will be checked if there is any. To update the drugs, click whatever drugs have been used and click on “Update”. A new line should inform that it has been updated.
To create a new record for the newborn baby, click on the “Delivery Time” tag. Fill in the form with names, gender, birth date, time, and delivery method. If there are preferred delivery method set previously, it will be the default option. Click on “Confirm” to proceed, and a new patient should be created will MID and Temporary Password been specified.

6. Use case 99

i) Description:

With the implementation of this use case, any HCP may search for the patient by MID or name. The history of the patient's travel information is displayed as one of the items for viewing patient (left tab). Any HCP has the option to select an existing record to view. HCPs can create a new travel history and add a new record. Patients can only view their records.

ii) Architecture & Design:

For this use case, we created a new table called TravelHistories in the sql/createTables.sql file and created a bean called TravelHistoryBean that contains a patient MID, the start date of the travel, end date of the travel, and the traveled cities. We also created TravelHistoryBeanLoader, TravelHistoryDAO, and TravelHistoryBeanValidator classes along with TravelHistoryBean to be able to load the TravelHistoryBean object in the jsp file later we used to generate the history of a patient's travel information. The newly created table was also added in sql/data/deleteFromAllTables.sql and sql/dropTables.sql to address the change made in sql/createTables.sql. New enum values were added in the src/main/edu/ncsu/csc/itrust/model/old/validate/ValidationFormat.java to validate the format of the TravelHistoryBean object.

After all the backend portion of UC_99 was set up, we worked on multiple files for the frontend side: WebRoot/auth/hcp/menu.xhtml, WebRoot/auth/hcp/addTravelHistory.jsp, WebRoot/auth/hcp/addTravelHistoryToDB.jsp, and WebRoot/auth/hcp/editTravelHistory.jsp. We added a new tab by modifying the WebRoot/auth/hcp/menu.xhtml file called "Travel History." Clicking on this tab directs the HCP to a patient's travel history page where he or she can edit/add travel histories. The main action of modifying patient's data regarding travel history occurs in WebRoot/auth/hcp/editTravelHistory.jsp. WebRoot/auth/hcp/addTravelHistory.jsp and WebRoot/auth/hcp/addTravelHistoryToDB.jsp files are used to add a new travel history information in the patient's database.

iii) Appendix:

To run the tests for the required implementations related to UC_90, there is a package under test/java/edu.ncsu.csc.itrust/unit/T804TestcaseCollections called UC_99. One can click on the package and run or run with the coverage to see the code coverage as well. We also created a dummy table called "TravelHistory1.sql" in sql/data folder to test implementations in TravelHistoryDAO.

To run the manual test on this use case, follow the instructions below:

After choosing a certain HCP user to log into iTrust, such as “ObHCP0”, log into iTrust. Now, click on the menu called “Patient Info” on the left and click on the tab called “Travel History.” The HCP will then be prompted to select a patient by name or MID. Put in any number or name that the HCP would like to choose. If the HCP types an invalid medical invalid identification number or name or a medical identification number or name of a patient not in the system, the HCP is prompted to try again by showing an error message, “Patient MID or Name does NOT exist! Please re-enter a valid patient MID or Name.” Suppose the HCP selected a person with MID “1.” The HCP is redirected to the page that shows the travel history for the selected patient. The HCP can select a button called “Add Travel History” to add another travel history record for the given patient. Click on the button and fill out the required information by following the format guideline as suggested. After doing so, click on the button that says “Add Travel History” and the webpage will notify you saying “Successfully Added.” Either click on “Add More Travel History” to add more records or click on “Back to Patient Travel History” to go back to see the updated patient’s travel history information. To see that the patient is ineligible to do the action as described above, logout from iTrust and log back in with any patient, such as “Patient 1.” The patient does not have an option to view anyone’s travel history and therefore cannot add any record for travel history information.