CS427 Project Documentation

Yiran Tao, Kechen Lu, Chengyu Qian, Lingxiao Li,
Yansong Huang, Ke Liu, Run Zhang, Jiaqi Xing

December 7 2018

Contents

1 Overview

iTrust is an electronic medical records application that provides patients with a system to keep track of their medical history. The project is produced and maintained by Software Engineering department at North Carolina State University. The repo of iTrust can be found at https://github.com/ncsu-csc326/iTrust. It is based on the Java/JSP with Bootstrap and Apache Tomcat to build the whole system.

As our term project in CS427, we successfully implemented the following Use Case: UC 93 - Obstetrics Patient Initialization, UC 94 - Obstetrics Office Visit, UC 95 - Labor and Delivery Report, UC 96 - Childbirth Hospital Visit and a new Use Case we designed - Patient Prescription. The following sections will outline these use cases and our implementation details.

2 UC93

2.1 Description

This use case allows user to operate on the obstetrics table. A HCP can search for the patient by MID or patient name. And then he can select a patient to see the obstetric initialization records. Only HCPs with a specialization of "OB/GYN" can create a new obstetrics record. If the OB/GYN HCP selects to add a new record, he should enters information required for the initial obstetrics patient.

2.2 Architecture & Design

2.2.1 Back-End

UC 93 has created obstetric care table in the database, which could store different data as stated in the requirements, the detail is shown in Table 1.

Also this use case has provided basic insert/update/delete/review interfaces in the back-end, which is used to transfer data to front-end. Moreover, we've implemented the fuzzy Search and exact Search interfaces for eligible users to query patient's obstetric data in the back-end with either patient's MID or just the name(first name and last name).

We followed the structure design like DAO factory, so we created a new bean called ObstetricRecordBean in the corresponding place with fully encapsulation for the class and provided required interfaces of the bean.

2.2.2 Front-end

We have a new menu link for viewing Obstetrics Records. It's under **Patient Info → Patient Prescription**. After opened it, it will be prompted to the users to choose a patient to view. Only the patients with the eligible flag set to true are allowed to be viewed. HCPs can change the patient's eligibility to be true by clicking the "Change" button, but only HCPs with the obstetrics care qualification can add new obstetrics record. Figure 1 shows the screenshots of when HCP opens the record for one patient, and Figure **??** shows the form to add a new obstetrics record for the patient.

Column	Type	Description	Nullable
ID	int	Primary Key	NO
PatientID	bigint	Foreign key to Patient	NO
CreatedDate	varchar	When this record was created	Yes
LMP	varchar	Last MP date	Yes
yearOfConception	int	Conception Year	NO
numberOfWeekPreg	varchar	Weeks of pregnancy	NO
numberOfLaborHour	double	Hours of labor	YES
weightGain	double	Weight gained during pregnancy	YES
deliveryType	enum	Types of delivery babies	NO
numberOfBaby	int	Number of born babies	YES

Table 1: Table for ObstericsRecord

Obstetrics Care Records

Record ID	Record Date	LMP	Year of Conception	Delivery Type	# of Birth	Action
4	12/02/2018	11/01/2018	2018	vaginal delivery	1	View Details
1	04/11/2018	03/21/2017	2019	vaginal delivery	0	View Details
5	04/11/2018	03/21/2017	2019	vaginal delivery	0	View Details
8	04/11/2018	03/21/2017	2019	vaginal delivery	0	View Details

Add a new Obstetrics Record

Figure 1: Obstetrics Records

3 UC94

3.1 Description

This use case has two main functionalities. First is to let doctors add and edit obstetrics records for obstetrics eligible patients. Second is to allow doctors to upload and download the ultrasound records for the obstetrics patients. To be a little more specific, in the obstetrics office visit part, an HCP can only view the obstetrics records for a patient, and an OB/GYN HCP can view, add and edit and delete the obstetrics visit records. This use case also allow the OB/GYN HCP to arrange the next obstetrics visit for a patient. The ultrasound records part, which is a sub-functionality of the obstetrics office visit, allow an OB/GYN HCP to view and add ultrasound records for a patient, they can also upload and download the record file.

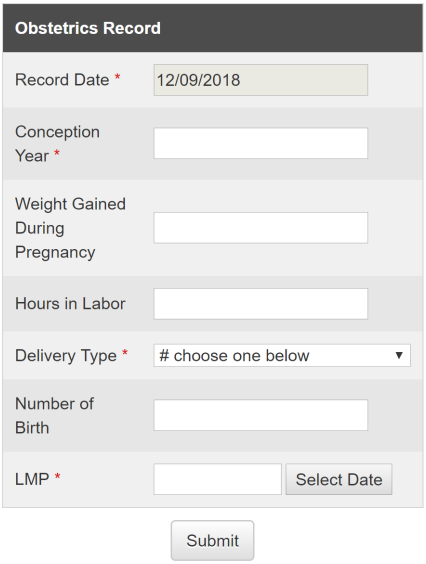

Figure 2: Obstetrics Records

3.2 Architecture & Deisign

3.2.1 Back-end

Column	Type	Description	Required
ID	Long	Primary Key	Yes
patientID	Long	Foreign key to Patient	Yes
createdDate	LocalDateTime	Created Date of the record	Yes
locationID	String	Location of the hospital	Yes
apptTypeID	Long	Appointment type of the record	Yes
notes	Text	Notes from the doctor	No
sendBill	Boolean	Whether bill is sent	Yes
LMP	LocalDateTime	Date of last menstrual period	Yes
weightInPounds	double	Weight in pound	Yes
bloodPressure	String	Blood pressure	Yes
FHR	Int	Fetal heart rate	Yes
numberOfBaby	Int	Number of baby	No
lowLyingPlacenta	Boolean	Whether a low lying placenta	Yes

Table 2: Database for UC94

The workflow is shown in figure 3, an OB/GYN HCP can view/add/edit/delete the obstetrics office visit records, and schedule the next obstetrics office visit, he/she can also add an ultrasound record or upload/download a file for ultrasound record under his/her office visit record by our use

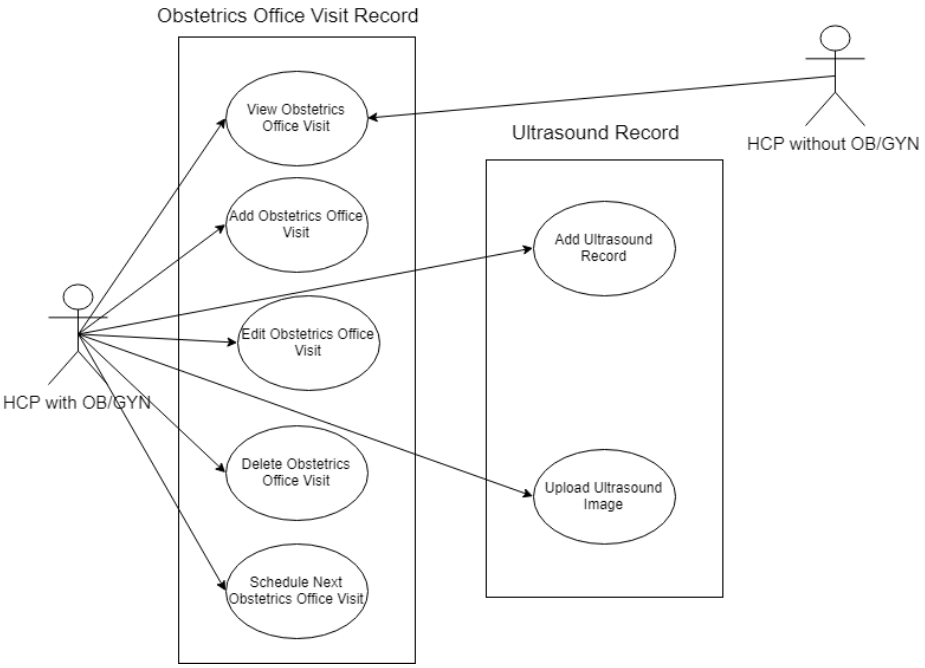

Figure 3: Workflow for UC94

case design. While an HCP can only view the obstetrics office visit records. We created a obstetrics office table shown in table 2, which could store different data as required for the obstetrics office visit.

Also this use case has provided basic add/update/delete interfaces in the back-end, which can be used to transfer data to front-end. For searching for a patient, we've implemented the fuzzy Search and exact Search interfaces for eligible users to query patient's obstetric data in the back-end with either patient's MID or just the name (first name and last name).

We used the structure design like DAO factory for the ultrasound part, so we created a new bean called UltrasoundRecordBean in the corresponding place with fully encapsulation for the class and provided required interfaces of the bean. In addition, we use similar structure design like OfficeVisit based on datasource for ObstetricsOfficeVisit. We created a public class for elements of ObstetricsOfficeVisit and make full implement with the interfaces of the class including sql codes.

For the unit tests part for the back-end, the following files are mainly the tests for the back-end: In package officeVisit.obstetricsOfficeVisit under the test folder, there are test cases for the file we constructed in the corresponding files in the main folder, they are the tests for our own data when connecting the data base. And for the ultrasound part, as we create bean and dao, we also have tests for the bean called ultrasoundbeantest in the bean folder under test folder and the action tests used in dao in different test files under action folder.

3.2.2 Front-end

We have a new menu link for viewing Obstetrics Office Visits. It's under **Office Visit** → **Obtetrics Office Visit**. Like UC93, after opened it, it will be prompted to the users to choose a patient to view. Only the patients with the eligible flag set to true are allowed to be viewed. HCPs can change the patient's eligibility to be true by clicking the "Change" button, but only HCPs with the obstetrics care qualification can add new obstetrics record. Figure 4 shows the

Column	Type	Description	Required
ID	Long	Primary Key	Yes
patientID	Long	Foreign key to Patient	Yes
visitID	Long	Foreign key to Obtetrics Office Visit	Yes
createdDate	LocalDateTime	Created Date	Yes
URL	String	URL of ultrasound image	Yes
CRL	Double	Crown rump length	Yes
BPD	Double	Biparietal diameter	Yes
HC	Double	Head circumference	Yes
FL	Double	Femur length	Yes
OFD	Double	Occipitofrontal diameter	Yes
AC	Double	Abdominal circumference	Yes
HL	Double	Humerus length	Yes
EFW	Double	Estimated fetal weight	Yes

Table 3

Column	Type	Description	Required
ID	Long	Primary Key	Yes
recordID	long	Foreign key to ultrasound record	Yes
contents	Blob	Binary Data of ultrasound	Yes
fileName	String	Name of ultrasound image file	Yes

Table 4

screenshot when HCP opens the visit records for one patient, and Figure 5 shows the form to edit an existing visit, and the ultrasound records.

For using our use case, we can take our http test as an example, we have a file called ObstetricsOfficeVisitHttpTest in package edu.ncsu.csc.itrust.selenium; we test two functionalities. One is the obstetrics office visit, here are the overflows. We first log in as an OB HCP, click the Document Obstetrics Visit button under the Office Visit tab. Then we can search for a patient, just view his record, click on the Add Obstetrics Visit button to add a new obstetrics record for the current patient. Next, we type in the data in the Obstetrics Visit form, and try to submit the form, if it succeed, we can get a tip on the page that says New Obstetrcis Visit added. We also try to edit the obstetrics visit record that already existed, by clicking on the edit button, we can edit the former record, if the edition is successful, we can get a sentence in our page that says Edit Finished. The other functionality is ultrasound, we also have a test for it. Similar to the first http test, we go to the edit button for an obstetrics office record and try to add a new ultrasound record, if the submission is successful, we will have a note on the page saying Ultrasound record added.

Obstetrics Visits

Visit ID	Date of the Office Visit	LMP	Number of Weeks Pregnant	Weight in Pounds	Blood Pressure	Fetal Heart Rate	# of Pregnancy	Send Bill	Low Lying Placenta Observed?	Notes	
2	01/20/2015	01/20/2015	00-0	10.0	71/101	100	1	No	No	good	Edit
129	12/04/2018	11/14/2018	02-6	10.0	10/200	100	1	Yes	Yes	10	Edit

Add Obstetrics Visit

Figure 4: Obstetrics Visits

Obstetrics Visit

Record Date *	01/20/2015
FHR *	100
Blood Pressure*	71/101
Weight in Pounds*	10.0
Notes	good
Send Bill*	# choose one below
Low Lying Placenta Observed*	# choose one below
Number of Baby	1
LMP *	01/20/2015

Submit Delete

Ultrasound Records

ID	Visit ID	Record Date	CRL	BPD	HC	FL	OFD	AC	HL	EFW	Action	
2	2	01/21/2015	2.2	30.1	2.1	100.2	3.1	3.6	24.3	19.1	Upload	
52	129	12/05/2018	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	Upload	Download
51	129	12/04/2018	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	Upload	Download

Figure 5: Obstetrics Records

4 UC95: Labor and Delivery Report

4.1 Description

UC 95 allows users to generate labor and delivery report for patients with eligibility for obstetric care. This use case will at first provide search interface to let users obtain desired patients with either the MID or the first name and last name as the input. Also this use case will provide alert functions to notify users whether the chosen patient is valid and prompt to ask the user to input again in the search box.

Then it will generate a web page which shows all the predefined information about the patient, including the past pregnancy records, estimated delivery date, obstetrics office visit information and other related medical data about this patient.

4.2 Architecture & Deisign

For the front end part, UC 95 add a new entry under the "Other" section for user to generate report. By choosing this new entry, users will enter a search page which allows user to query desired patients according to their MID or names(first and last).

Then the search result is shown below the search box and every record has a "View" button for user to click on. However, for patients with invalid eligibility for obstetric care, once the user select them, a dialog will pop out and suggest user to search again, then the web page will be redirect to the previous search page.

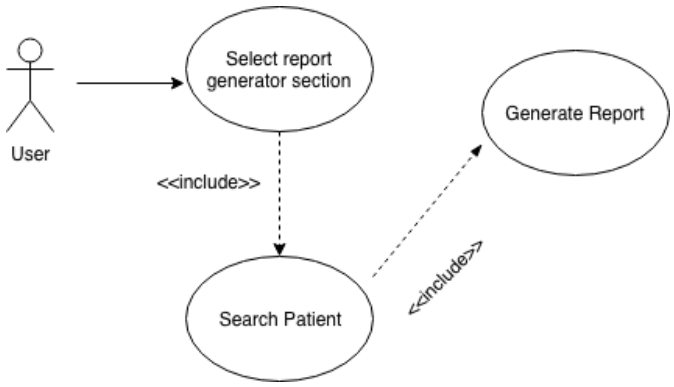

Figure 6: UC95 Use Case Diagram

After a valid patient is chosen, the web page will be directed to the report page which will provide all the predefined data obtained from the back-end. We strictly follows the MVC framework and previous DAO factory design, so at the front-end, by calling the newly added function, which returns a Bean Object, in the SearchAction.java file, we successfully construct the connection between the back-end and font-end.

At last for the report web page, we implemented multiple tables, each table will be able to present a type of information as defined in the project requirement, and the style of the Html codes still follow the previous project design.

For the backend, I firstly create the Record Bean to include all the information, Information for each past pregnancy,Obstetrics Office Visit Information,Pregnancy complication warning flags all referred in UC95.

All the information are created in previous user case, so I only need to call previous search function to get those information. For example, for the Obstetrics Office Visit Information, I only need to cal the search function in UC94 and given the specific patient ID or name, return the Obstetrics Office Visit Information. So, I modify searchUserAction class and add two new function, one input is for given patient ID and one input is for given name. For each function, I includes all the needed search function implemented before to return the Record Bean. For the information in Pregnancy complication warning flags, there are some information about the pre-existing conditions, such as Diabetes, Cancers and STDs. I need to add those illness' icdCode to specific database.

For test part, I modify the searchUserActionTest class and to test those two new search functions and specifically, I add different pre-existing conditions as test data.

5 UC96: Childbirth Hospital Visit

5.1 Brief Description

UC96 is dedicated to provide a user-friendly interface for HCP to check patients' scheduled hospital visit or emergency room visit whose goal is to give a birth. It will display the history of patients' obstetric care which could function as a reference for HCP to decide the delivery type. HCP could modify the patient's preferred childbirth method and the dosage of drugs during labor and delivery. After the real delivery, now HCP could record all the basic information of all babies like gender, name. Later on, any HCP could still edit or delete all the information in this use case.

5.2 Architecture & Design

5.2.1 Architecture

In this use case 96, we implemented a new interface residing in the office visit scenario, and providing HCP general entry to see and create hospital visit for childbirth patients. Within the component Office Visit and MVC pattern, this use case also builds ChildHospitalVisit DAO model and relevant actions as well as the visit record bean for data interaction between front-end and back-end. According to the requirements and team discussions for the use case scenario, we had a series of use case items and its dependencies described in the Figure 7 below. Plus, the childbirth hospital visit schedules depend on the UC94 we demonstrates before.

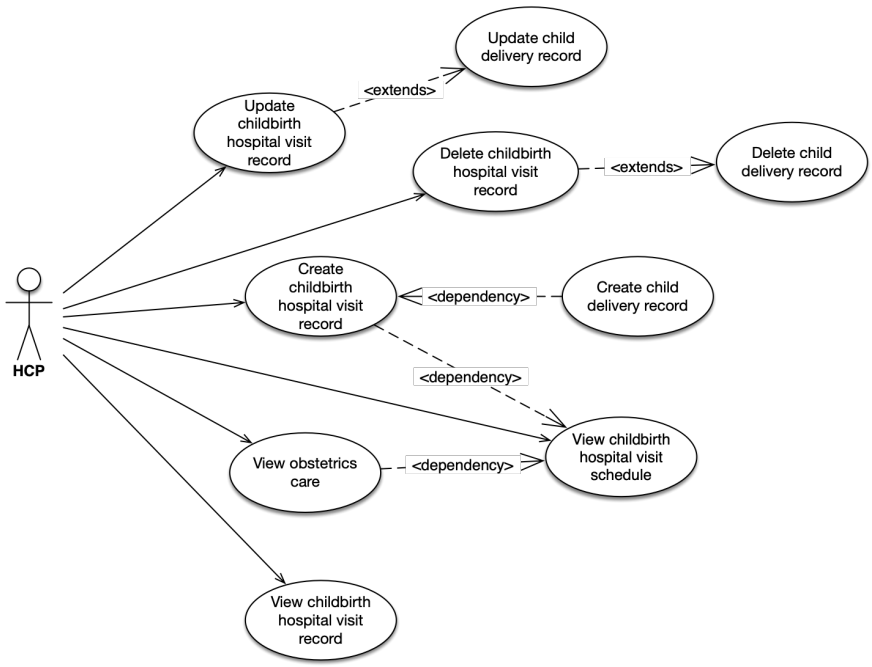

Figure 7: UC96 Use Case Diagram

5.2.2 Table Design

A few new tables have been added into database. It contains all the information we need to store. The ER diagram of the new tables is displayed below:

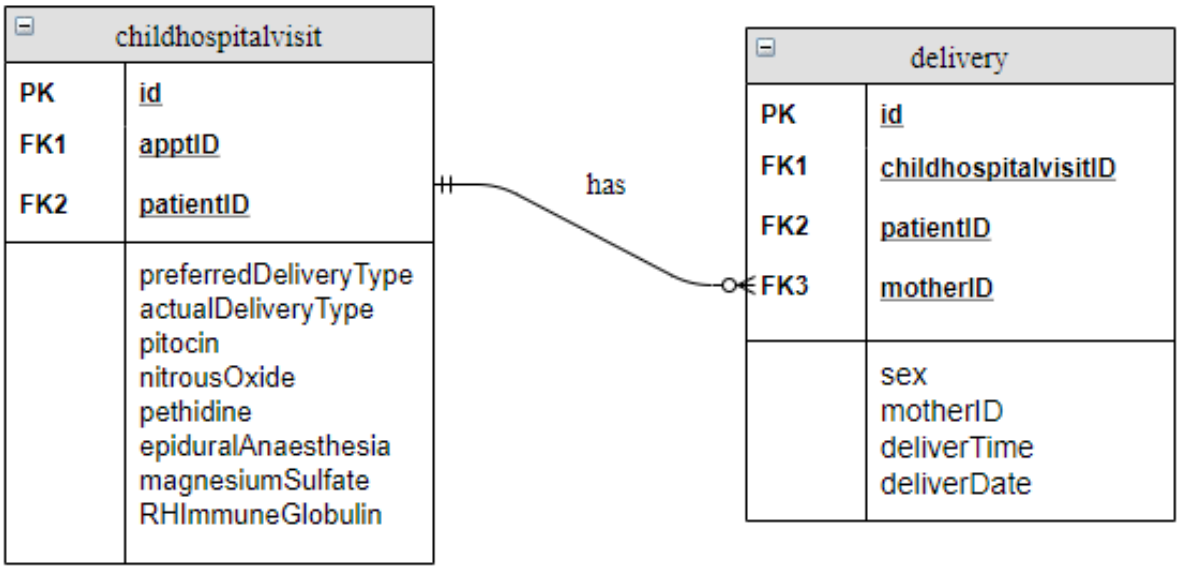

Figure 8: ER diagram of new tables

The 'childhospitalvisit' table stores all the information of the visit except for the the information about the delivery. We designed a new table called 'delivery' to help record all the delivery information. The 'delivery' table basically maintains the date and time of the delivery as we notice that patients may give birth to several babies. Sometimes parents want to record the exact time of their birth to decide their names or something like that. It contains the patientID of the mother and the baby which helps to get all babies of a mother.

5.2.3 Back-end Design

For the back-end part, it follows the overall architecture of the project-MVC. Two beans called "ChildHospitalVisitBean" and "DeliveryBean" were created. Corresponding DAO named "ChildHospitalVisitDAO" and "DeliveryDAO" were also created to manipulate the database. "ChildHospitalVisitDAO" provides interfaces for users' actions like add, search, check, edit and delete. "DeliveryDAO" provides interfaces for users' actions like add, search, edit and delete. We use patientID and apptID to search the specific Child Hospital Visit record. MotherID is needed to return all the delivery record of a mother. PatientID is used to help HCP delete the birth record of a baby. Also, we will validate the correctness of data and ask the users to reenter the data if it contains error.

5.2.4 Front-end Design

As for front-end design in UC96, we illustrate our process flow for users HCPs in the Figure 9 below. Since the front-end leverages multiple features from the current iTrust project, the JSP files could be easily built on the original structure of that. We have two JSP files for implementing this use case. First, when HCPs run into the Childbirth Hospital Visit page, current visit schedules display and HCP could choose initialize a record or enter this record either. Later if having initialized the record, HCP can update the certain information on this visit, specifying the drugs or preferred delivery type, as well as viewing the obstetrics records of this patient. When delivery completed, HCPs could record the delivery time/date, type and children information while creating a new patient in patient table as well. HCPs are able to delete all of this record and return to uninitialized state of this visit.

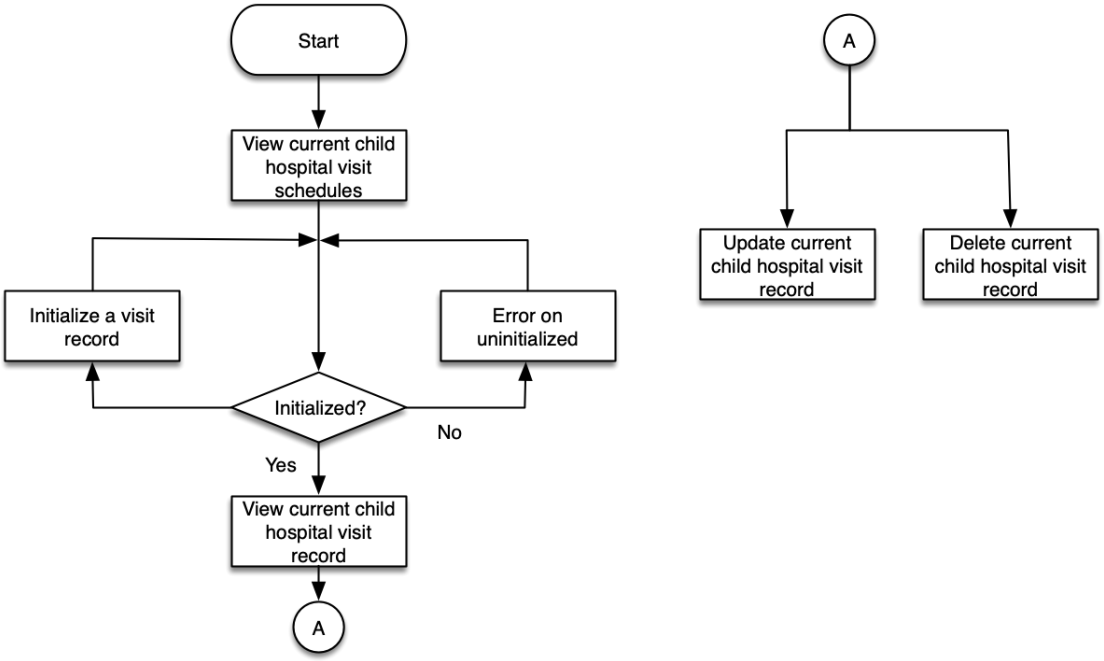

Figure 9: UC96 Flow Diagram

6 UC99: Patient Prescription

6.1 Description

UC99 is our new use case proposal. This use case is dedicated to provide HCPs and patients with a way to use a functionality of prescriptions. Specifically, this Use Case allows doctors (HCP) to view, add, edit and delete prescriptions for patients, and patients can also view the prescriptions from their doctors.

6.2 Architecture & Design

6.2.1 Back-end

The main workflow can be described in the use case diagram in Figure 10.

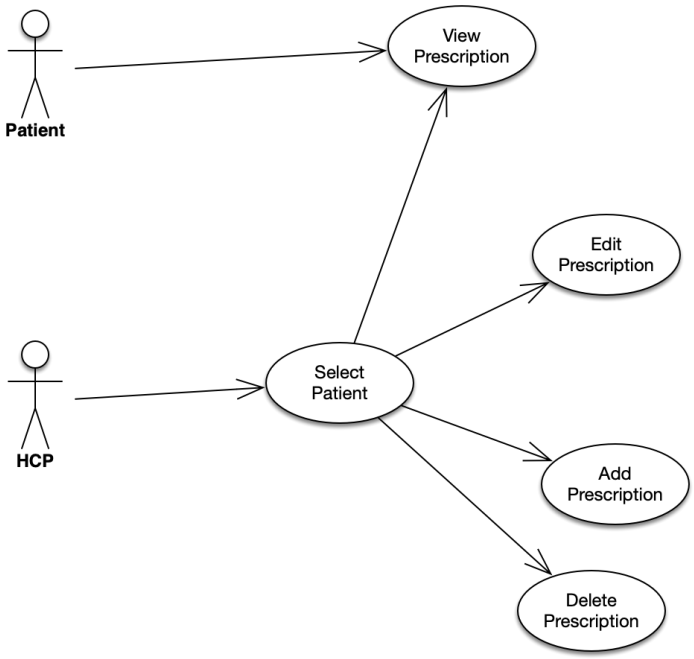

Figure 10: Workflow for UC99

The HCP can view/add/edit/delete the data. When adding a new data, a created date field will be automatically populated, and the HCP can enter data for the name, the dosage, and the notes on how to take the medicine. When the HCP saves the new prescription or edit the existing one, the new prescription will use the current patient id in the patientID field. The patient can only view the prescription with the same MID as its user only.

We have a new table to describe the patient prescription and it's listed in table 5.

Column	**Type**	**Description**	**Required**
ID	Long	Primary Key	Yes
PatientID	Long	Foreign key to Patient	Yes
CreatedDate	DateTime	When this record was created	Yes
Name	Text	Name of the prescription	Yes
Dosage	Double	Dosage of the prescription	Yes
Notes	Text	Notes from the doctor	No

Table 5: Table for PatientPrescription

6.2.2 Front-end

When an HCP opens the page, he or she can view the patient's prescription under **Patient Info** → **Patient Prescription** and patient can view under **View** → **View my Prescription Report**.

When opening the **Patient Prescription** menu, the HCP will need to select a patient for viewing the prescriptions. After opened it, he or she will be able to check the current prescriptions for the selected patient. He can click the "Edit" button to modify the prescription. Inside the "Edit" page, he can delete this data. He can also add new prescription for this patient.

For the patient, after he or she opens the **View my Prescription Report** link, a list of current prescriptions will be displayed. He or she will not be able to modify these prescriptions.

7 Appendix

We run the JUnit tests of all the use cases as below.
UC93:
ObstetricsValidatorTest1.java
InputValidatorTest1.java
SearchUsersActionTest.java
ObstetricsPatientBeanTest.java
ObstetricsRecordsLoggerTest.java
UC94:
SearchUsersActionTest.java
ObstetricsOfficeVisitMySQLTest1.java
ObstetricsOfficeVisitMySQLTest2.java
ObstetricsOfficeVisitTest.java
ObstetricsOfficeVisitValidatorTest.java
UltrasoundRecordBeanTest.java
AddUltrasoundRecordTest.java
EditUltrasoundRecordTest.java
ObstetricsOfficeVisitLoggerTest.java
UC95:
SearchUsersActionTest.java
UC96:
AddDeliveryActionTest.java
CheckAddChildHospitalVisitTest.java
SearchChildHospitalVisitActionTest.java
SearchDeliveryActionTest.java
ChildHospitalVisitTest.java
UC99:
PatientPrescriptionValidatorTest.java
PatientPrescriptionBeanTest.java
AddPatientPrescriptionTest.java
EditPatientPrescriptionTest.java
PrescriptionLoggerTest.java
And we run all the HtmlUnit tests in the /iTrust/src/test/java/edu/ncsu/csc/itrust/selenium.