# Change request log

# Team

Group 3

Susheel Pagadala - xxp155831

Mayuri Banubakode - mxb156530

# Change Request

CR-FEMR-SPMB-137 - flag birthdays as being accurate or a guess

***Description:***

# Researcher wants fEMR to flag whether or not the patient's birthdate is real or if fEMR tried to guess what it was

# -So that data is accurately stored with integrity.

# Notes:

# 1) Patients often do not know what day they were born and sometimes don't even know their age.

# 2) fEMR allows you to identify how old a patient is in 3 different ways - enter the actual birthdate, enter an age integer, or select a category (child,adult,elder,etc). Our database has 2 options of storing this data - a birthdate OR the category. If the user enters an age integer, the system will create a fake birthdate, but it won't flag the birthdate as fake for future reference.

# Concept Location

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We ran the system |  |
| 2 | We went to Triage screen by clicking on the Triage Menu | To find out the screen where birthday is captured |
| 3 | We identified that the ‘age’ section of the screen is capturing the birthday using the three alternate methods explained in the change request | To find out the location on the screen where birthday is captured |
| 4 | We have decided it would be nice if the end user notifies the system on the type of birthday (i.e. real or fake) by clicking on a radio button. We determined the location of the change required is triage index.scala.html | Determine the UI page to be changed for birthday flag |
| 5 | We inspected the code using chrome developer and realized that the submit button is posting to indexPost method on the TriageController | To dentify the controller where request is posted |
| 6. | Using IntelliJ opened up TriageController.java class, searched for indexPost method. Looked at the services it is calling and the objects it is creating to populate these service methods we noted down the following methods/classe need change  IndexViewModelForm. bindFromRequest()  SessionService. RetrieveCurrentUserSession()  PatientItem  PatientService.createPatient()  photoService.createPatientPhoto()  encounterService.createPatientEncounter()  vitalService.createPatientEncounterVitalItems()  encounterService.screenPatientForDiabetes() | Typically, the instance variables will provide a way to identify the sequence of flow |
| 7. | Next we looked at each of the above classed and eliminated the classes that do not use age fields in any of the setters  And finalized the following list of classes as the where we might have to make code changes     1. femr.ui.controllers package   public class TriageController extends Controller  private PatientItem populatePatientItem   1. femr.data package   public interface IDataModelMapper  IPatient createPatient   1. femr.data package   public class DataModelMapper implements IDataModelMapper  public IPatient createPatient   1. femr.data.models.core package   public interface IPatient  Integer getFakeBDFlag()  void setFakeBDFlag(Integer fakeBDFlag)   1. femr.common package   public class ItemModelMapper implements IItemModelMapper  public PatientItem createPatientItem   1. femr.common package   public interface IItemModelMapper  Method public PatientItem createPatientItem   1. femr.common.models package   public class PatientItem  private Integer fakeBDFlag;  public Integer getFakeBDFlag()  public void setFakeBDFlag(Integer fakeBDFlag)   1. femr.business.services.system package   public class PatientService implements IPatientService  public ServiceResponse<PatientItem> updateSex(int id, String sex)  public ServiceResponse<PatientItem> createPatient(PatientItem patient) | To filter classes/methods that are irrelevant |
| 8 | We logged on to MySQL and filtered Patient related tables | To identify where we need to add the flag |
|  | and realized that when age classification is selected from the UI, the birthday is stored as empty in the patients table.  This led us to conclude that the system stores age only when birthday in the format of (mm/dd/yyyy) or (mm/yy) is entered on the UI. Thus decided to have a new flag in patient’s table with the name FakeBDFlag | To figure out where should we add extra flag. |
|  |  |  |

**Time spent (in minutes):** 180

# Impact Analysis

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We reviewed edit triage screen by entering existing patient id and correspondingly went to indexGet method and indexPopulateGet mthods and identified that we need to make code changes to searchService needs to be modified for it work | To see if the code change in create flow would impact the edit flow. |
| 2 | We navigated through other (Medication/Pharmacy/research) screens in the application and realized that changes are local to create and edit screens of Triage screen | to see what other methods/classes might have affected |

**Time spent (in minutes):** 60

# Actualization

Using the table below, describe each step you followed when changing the code. Include as many details as possible, including why classes/methods were modified, added, removed, renamed, etc.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We updated the Patients table and added a new column FakeBDFlag with Integer Value type | To differentiate between real or fake birthday |
|  | Updated  femr.data.models.core.IPatient.java and  femr.data.models.mysql.Patient.java  By adding additional persistent column FakeBDFlag | To store the above flag in the database |
|  | Updated femr.data.IDataModelMapper.java and  femr.data.IDataModelMapper .DataModelMapper.java | To update CreatePatient() method |
|  | Updated package femr.common.IItemModelMapper.java and package femr.common.ItemModelMapper.java | PatientService is calling method on the  ItemModelMapper’ createPatientItem |
|  | femr. business.service.system.PatientService.java and updated CreatePatient’s method arguments by adding the FakeBDFlag | To update Business layer |
|  | femr. business.service.system.SearchService.java and updated CreatePatient’s method arguments by adding the FakeBDFlag | To update Business layer |
|  | Updated femr.ui.controller | To add the FakeBDFlag in the private variables as the class uses it |
|  | ui\views\triage\index.scala.html | To add radio buttons on the html |

**Time spent (in minutes):** 240

# Validation

Using the table below, describe any validation activity (e.g., testing, code inspections, etc.) you performed for this change request. Include the description of each test case, the result (pass/fail) and its rationale.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Test case defined: Valid data - verify Traige screen showing the newly added radio buttons for identifying the if age is real or fake  Inputs: Navigate to Traige screen  Expected output: The screen should display birthday type | This is the expected behavior.  The test passed. |
| 2 | Test case defined: Valid data – Choose birthday flag as real, enter the birthday in mm/dd/yyyy format  Inputs: Choose birthday flag as real, enter the birthday in mm/dd/yyyy format  Expected output: ‘Patient data saved successfully message’ should show up | This is the expected behavior.  The test passed. |
| 3 | Test case defined: Valid data – Choose birthday flag as real, enter the birthday in months and years format  Inputs: Choose birthday flag as fake, enter the birthday in in months and years format  Expected output: ‘Patient data saved successfully message’ should show up | This is the expected behavior.  The test passed. |
| 4 | Test case defined: Valid data – Edit Triage screen by entering valid patientID created via step2 above  Inputs: Edit Triage screen by entering valid patientID created via step2 above and submit  Expected output: Patient data should show up along with birthday type ‘real’ pre-selected | This is the expected behavior.  The test passed. |
| 5. | Test case defined: Valid data – Edit Triage screen by entering valid patientID created via step3 above  Inputs: Edit Triage screen by entering valid patientID created via step2 above and submit  Expected output: Patient data should show up along with birthday type ‘fake pre-selected | This is the expected behavior.  The test passed. |

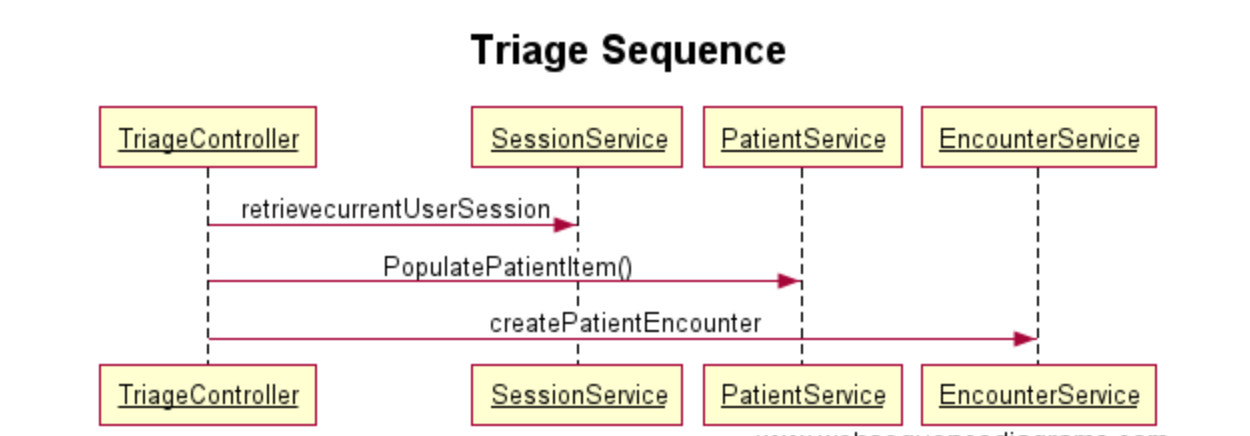
**Time spent (in minutes):** 120

# Timing

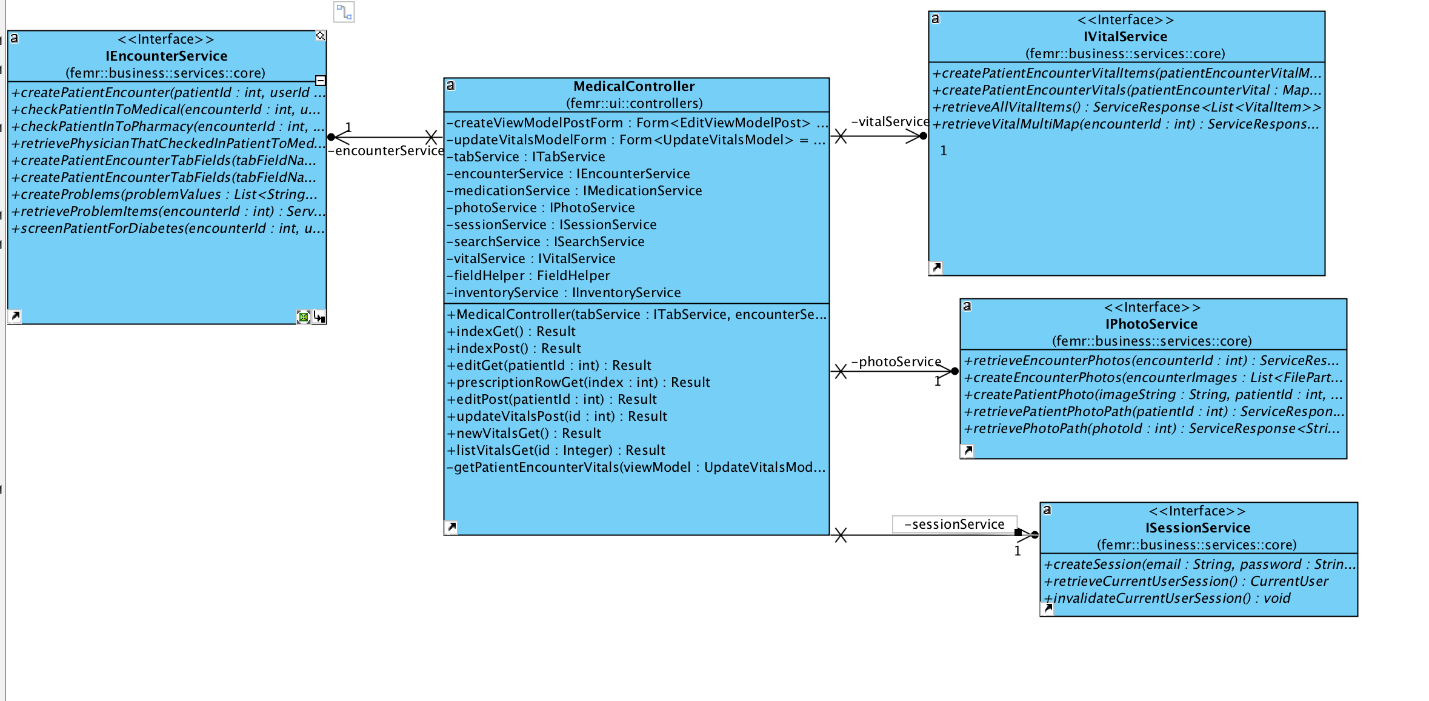
Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 180 |
| Impact Analysis | 60 |
| Actualization | 240 |
| Verification | 120 |
| Total | **600** |

# Reverse engineering



**Class Diagram**



# Conclusions

This was complex change as it required an understanding of overall frameworks implemented and also how the application flows. Spent time on triaging the required changes by tracing back each component at a time. Also form a functional usage perspective had to discuss a lot on UI implementation and corresponding updates to classes and eventually storing in DB. There could have been multiple ways to implement changes using check box, radio button, fixed buttons etc. picked a solution that provided faster implementation and intuitive to use.

Files, Classes and methods changed:

1.

package femr.business.services.system;

public class PatientService implements IPatientService

public ServiceResponse<PatientItem> updateSex(int id, String sex)

public ServiceResponse<PatientItem> createPatient(PatientItem patient)

2. package femr.business.services.system;

public class SearchService implements ISearchService

public ServiceResponse<PatientItem> retrievePatientItemByPatientId(int patientId)

public ServiceResponse<PatientItem> retrievePatientItemByEncounterId(int encounterId)

3. package femr.common.models;

public class PatientItem

private Integer fakeBDFlag;

public Integer getFakeBDFlag()

public void setFakeBDFlag(Integer fakeBDFlag)

4. package femr.common;

public interface IItemModelMapper

Method public PatientItem createPatientItem

5. package femr.common;

public class ItemModelMapper implements IItemModelMapper

public PatientItem createPatientItem

6. package femr.data.models.core

public interface IPatient

Integer getFakeBDFlag()

void setFakeBDFlag(Integer fakeBDFlag)

7 package femr.data

public class DataModelMapper implements IDataModelMapper

public IPatient createPatient

8 package femr.data

public interface IDataModelMapper

IPatient createPatient

9 package femr.ui.controllers

public class TriageController extends Controller

private PatientItem populatePatientItem

10 package femr.data.models.mysql;  
public class Patient implements IPatient

public Integer getFakeBDFlag()

public void setFakeBDFlag(Integer fakeBDFlag)

11 ui\views\triage\index.scala.html

Created new radio buttons for Birthday flag fake generated or real entered