# Change request log

# Team

Team 2: Michelle Jones (Coding) and Rangy Samuel (Documentation)

# Change Request

FEMR-137 <https://teamfemr.atlassian.net/browse/FEMR-137>

# Concept Location

|  |  |  |
| --- | --- | --- |
| **Step #** | **Description** | **Rationale** |
| **1** | We ran the system | Can’t fix what we can’t run. |
| **2** | Since this has to deal with birthdays, executed the following line in the cmd prompt to try and find anything birthday related "grep –ri birthday ." | Unsure where the age is calculated. The new patient for allows you to enter an approximate date, age range or just guess the age.  In this case, unless they entered an exact date, it is an estimation that needs to be reflected in the database |
| **3** | Examine public/js/triage/triage.js | Grep returned this file and looks like java script is called by the HTML code when user created in triage. One patient is created in triage. |
| **4** | DataModelMapper.java - this is where the patient is being created into an object. | The birthdate is already passed in when we create this object so we should find where that is called when it is created. |
| **5** | DataModelMapper.java implements an interface. Looks like we will have to update the interface IDataModelMapper.java with the changes needed to IPatient | Need to update the IPatient class with the added field, but first we have to see if any other classes call the patient class. |
| **6** | In cmd, executed "find . -name IPatient\* -print" | DataModelMapper.java creates a patient from the IPatient class. Need to see where that class is in order to see how they use date |
| **7** | IPatient is an interface. Next step is to see which class implements this interface by doing another grep... "grep –r "implements IPatient ." | Patient.java implements IPatient. This file will have to be updated with the new column |
| **8** | Examined file app/femr/business/services/system/PatientService.java | This file has the object IPatient. It will need to be updated with our additional column. |
| **9** | Examined file  app/femr/business/services/system/SearchService.java | This file has the object IPatient. It will need to be updated with our additional column. |
| **10** | Examined file app/femr/common/IItemModelMapper.java | This file is an interface that also has mehtod for PatientItem. Has to be updated with the additional column |

**Time spent (in minutes):** 1 hr

# Impact Analysis

Use the table below to describe each step you followed when performing impact analysis for this change request. Include as many details as possible, including why classes are visited or why they are discarded from the ones that have to change.

Do not take the impact analysis of your changes lightly. Remember that any small change in the code could lead to large changes in the behavior of the system. Follow the process on impact analysis covered in the class. Describe in details how you followed this process in the change request log. Provide details on how and why you finished the impact analysis process.

|  |  |  |
| --- | --- | --- |
| **Step #** | **Description** | **Rationale** |
| **1** | The list of classes that would need updating because they are either an interface or they have the IPatient/Patient object within them grew substantly to 11 classes. | Going to have to update the code in order to implement this change, but there is a lot of variables to consider for each file. Multiplying that by 11 is dangerous. |
| **2** | We haven't found a specific class that executes MySQL statements, but we know we need to alter the table. The only location in the code where we were able to find and sql was under conf/evoluions.default | Not sure why the scripts are so ingeniously called 1.sql, 2.sql, …, 98.sql but the auto incrementing tells us that something is iterating through this on startup. Going to add a 99.sql to see what happens!  \*99.sql will alter the patient table and add the column\*\* |

**Time spent (in minutes):** 5

# 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** | Created file 99.sql  This updates the patients table with a new column isApproximateAge.  The column is not required  Yes – Not a guess  No – A guess  Null - Null | Have to update the table in the database with this column. Figured it doesn't have to be required so that way we don't have to create a new set of logic to update the current patient list with a guess... |
| **2** | Updated the definition of the patient object to include the String isApproximateAge with getter and setter...   * App/femr/data/models/core/IPatient.java * App/femr/data/models/mysql/Patient.java | The PatientClass now takes into account whether the birthdate was a guess or is an approximation for new users! |
| **3** | Updated the following classes to have the newly updated Patient.java class declaration. This was adding 1 line every time a new Patient object was created   * App/femr/business/services/system/PatientService.java * App/femr/business/services/system/SearchService.java * App/femr/common/IItemModelMapper.java * App/femr/common/ItemModelMapper.java. * App/femr/common/models/PatientItem.java * App/femr/data/DataModelMapper.java * App/femr/data/IDataModelMapper.java * test/mock/femr/data/MockDataModelMapper.java | All the classes that use the Patient object are now updated to include the string for isApproximateAge. |
| **4** | Updated Public/js/triage/triage.js to set the value for isApproximateAge depending on whether the age was input from an age group, age guess or approximate age | This is where the value is being calculated based on the input from the user. This value will be used when creating a new patient in the Patient.java class through the DataModelMapper class. |

**Time spent (in minutes):** 5

# 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** | Created 3 new patients, one with a estimated age, one with an age group and one with an approximate age. | Best way to test is to try all three at once. |
| **2** | Manually logged into MySQL and performed a selection based on last name (Made sure they all had the same last name) | Fastest way to see what is saved to the database |
| **3** | All users regardless of input method are having a string value of "NULL" for isApproximateAge... but the column is being populated! | Not a good sign, but the column is being populated. |

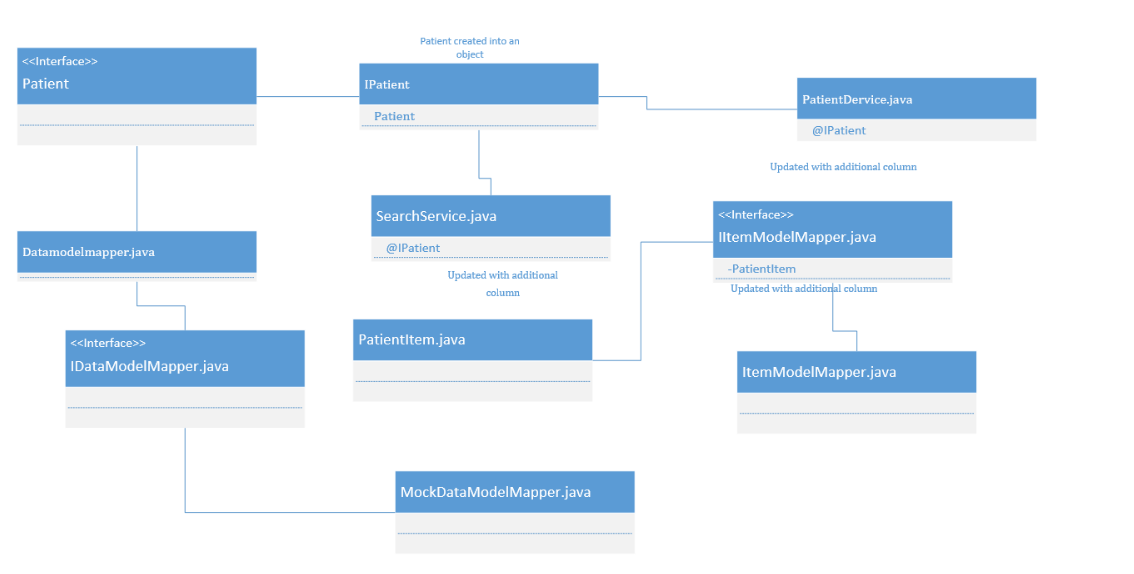
**Time spent (in minutes):** 5

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| **Phase Name** | **Time (in minutes)** |
| Concept location | 60 |
| Impact Analysis | 30 |
| Prefactoring | 0 |
| Actualization | 15 |
| Postfactoring | 0 |
| Verification | 5 |
| **Total** | 1 hr 50min |

# Reverse engineering



# Conclusions

This change request was the most challenging of the 3. It involved changing the structure of the database as well as values in the code. One major task was figuring out what to modify and make sure that everything was modified correctly. The change report lists that it only wants to see the data stored correctly, not that it wants the user to see any indication if the data was guessed, so we decided to not update the UI for that reason. That left us with only figuring out where in the sql to add the column, what the column should be (boolean was out because we saw 3 possible options of YES, NO, NULL). To make it easier we didn't create a way to update an original database with this new field and "guess" for the existing patients if it was a guess or not. Instead, we made sure the column could be null.

Classes and methods changed:

* App/femr/business/services/system/PatientService.java
* App/femr/business/services/system/SearchService.java
* App/femr/common/IItemModelMapper.java
* App/femr/common/ItemModelMapper.java.
* App/femr/common/models/PatientItem.java
* App/femr/data/DataModelMapper.java
* App/femr/data/IDataModelMapper.java
* App/femr/data/models/core/IPatient.java
* App/femr/data/models/mysql/Patient.java
* Conf/evolutions/default/99.sql
* Public/js/triage/triage.js
* test/mock/femr/data/MockDataModelMapper.java