1. **Change request log**
2. **Team**

Kendra Castaneda(Documentator) and Cyndi Streun(Coder)

1. **Change Request**

Change order [FEMR-137](https://teamfemr.atlassian.net/browse/FEMR-137) it is requested that there is a flag for the patient if the patient that was created was assigned a birthday by the system or if it is their actual birthday in order to ensure that the data is 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.

1. **Concept Location**

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We logged into FEMR with the credentials provided |  |
| 2 | We then clicked to “triage” to find the create a new patient screen | We recalled that in the Triage tab when we created our user Bob, since we did not give him a birthday the system gave him one.  Select the birth date as the area to be changed. |
| 3 | We looked in the database for the table Patients, there was not a birthdate column here there was just an age column | We were looking for where the patient information would be stored, it seemed the logical thing to do |
| 4 | We queried the database for the patient that we created earlier (Bob), we looked at the results and the age column was populated for Bob. | We wanted to see what was populated when there was no age or birthdate was provided to see what was populated on the table for their age. |
| 5 | Created another patient with a birthdate and queried to see the age column populated with the age as of the date we provided | Wanted to see the effect of just adding a birthdate |
| 6 | Created a patient with just their age and then queried the database and saw that the age that we provided was what was put into the database. | We wanted to verify that the age that we provided was also put into the database. |
| 7 | We used the IDE search functionality to search the code to look for where “Age” could be found in the code. And we found Triage page java code | We wanted to find where in the code age could be found. |
| 8 | In Triage java we found the calculation birthdayfromage function in patientitem.java | we marked this section as to be modified. |

**Time spent (in minutes):** 35

**Impact Analysis**

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 0 | Review database table patients | Add column to table for persistece |
| 1 | We kept searching around and started analyze dependencies in intellij | Trying to track all the areas that are affected by the change |
| 2 | We searched within git hub project with word add column, and decided we’d start with patientitem.java | Searching around |
| 3 | We looked at patientitem.populatepatientitem() and saw that it calls viewmodelpost.getage() | We wanted to make a note of things involved with the class. – Marked this class to be modified |
| 4 | Next we looked at the Patient.java found that this class would need to be modified | We marked this class to be modified |
| 5 | We searched for PatientItem | We were covering impacted sections |
| 6 | Since the birthdate is found in the triage page we marked the triagecontroller.java for change | We knew where the change was in the webpage so we included the triage page controller |
| 7 | We went to the Indexviewmodelpost.java class and looked to see if it would be needed to be changed | Marking this java class for a change |
| 8 | Then looked at the related Index.scala.html to include our change to the screen | Marking this file for a change |

**Time spent (in minutes):** 40

1. **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 searched within git hub project with word add column, and decided we’d start with patientitem.java | We needed to make sure we started at the first instance where the change was needed |
| 2 | We looked at patientitem.populatepatientitem() and saw that it calls viewmodelpost.getage() | We wanted to make a note of things involved with the class. |
| 3 | We create a sql script to add a new column to the database in order to add our new flag | We need to be able to add a new column for the new flag |
| 4 | We used the alter command to modify the table patients to add the column isBirthCalculated | The change that we made |
| 5 | Next we modified the Patient.java to include :  femr.data.models.mysql  private Boolean isBirthCalculated;  @Column(name=”isBirthCalculated”,referencedColumnName=”isBirthCalculated”) | We did this in order to include the new column in the java code so that it would be used. |
| 6 | We then searched for birth and modified the PatientItem.java to include:  Private Boolean isBirthCalculated; | To make sure we handle the impacted classes |
| 7 | We searched for PatientItem | We were covering impacted sections |
| 8 | In order to test our change we set breakpoints on the triagecontroller.java patientitem and ran it, we were able to see the new database column in the patient list | We were testing to verify that the change was behaving as we expected |
| 9 | We went to the Indexviewmodelpost.java class and made a change to add a declaration get and set. | Next we needed to figure out how to display this change. |
| 10 | Then made a change to Index.scala.html to include:  @input(“Is Birth Calc?”,”isBirthCalculated”, after birthdate, we ran our changes and experienced a compile error | Adding to include this change in the html file |
| 11 | Researched the issue that we experienced and the reason was that getisBirthCalculated is not a member of femr.common.models.PatientItem | Reasearched why our code would not compile to resolve it. |
| 12 | Next we modified the ItemModelMapper.java class in order to add isBirthCalculated | Tried to fix the issue we experienced |
| 13 | Modified the IItemModelMapper.java class in order to add to create isBirthCalculate | Also part of our issue |
| 14 | Compiled in debug mode and received the same error | Trying to see if the error we had was fixed |
| 15 | We modified common/models/patientitem.java to add isBirthCalculated  We tried to compile again and got an error with patientservice | Continuing with trying to fix out compile issue |
| 16 | We used the dependency map of patientService and looked at every class on the list to find out what we may have missed  We then found that triage.js to set is birth calculated when it uses calculateBirthdayFromAge | We were trying to find what we missed |
| 17 | Looked for our change on the screen and it did not show up, then changed the database and it was not saved there either | Looking to see that our change was implemented |
| 18 | We set breakpoints however, the issue is in the java script and it did not go into debug mode | Trying to figure out the issue |
| 19 | Changed the var in triage.js to setting of: patientinfo.isbirthcalculated | We looked at ways of how to solve the issue we were experiencing |
| 20 | Updated database and set isBirthCalculated for new patient and went to search and display was not working | See if display works but set of field is broken |
| 21 | We used the debug mode to figure out what was wrong and did not use this change as it broke the display, reverted the changes | Since it did not fixed the issue we decided to revert |
| 22 | The submit was fixed and the display is ok but setting birthdate calculated does not work. | Trying to resolve the issue |
| 23 | We started to use the debug mode again to try to figure out where the problem is, we found out the issue is in the triage.js  Boolean is hard to display, added a nested if but failed, used the tostring function | We need to figure out where the problem is and how to change it. |
| 24 | The change that was made worked but it did not write to the database |  |
| 25 | We found create new patient in triageController and added the setter for is birth calculated |  |
| 26 | Stepped through very carefully and missed on set in datamapper | Trying to find if there was anything else we missed |
| 27 | The change finally wrote into the database |  |
| 28 | Add sql script to  intellij by copying 98.sql and editing by hand |  |
| 29 | Tested the change and it worked | We tested everything was working as before and that the change we made worked as well. |
| 30 | We committed our changes with git. | Just in case we need to revert our changes. |

**Time spent (in minutes):** 130

1. **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: would create a patient with age to check that the new flag would be set  Inputs: just the age  Expected output: the age should be on the database and the flag should also be set | This is the desired state of after the change  The test passed. |
| 2 | Test case defined: would create a patient with a birthdate to check that the new flag would not be set  Inputs: just the birthdate  Expected output: the age should be on the database and the flag should not be set | This is the desired state of after the change  The test passed. |

**Time spent (in minutes):** 30

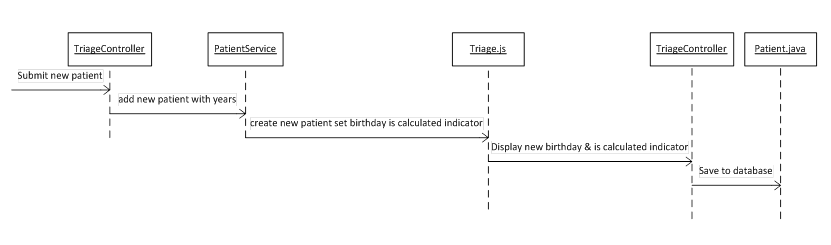
1. **Timing**

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 35 |
| Impact Analysis | 40 |
| Actualization | 130 |
| Verification | 30 |
| Total | 235 |

1. **Reverse engineering**

Create a UML sequence diagram (or more if needed) corresponding to the main object interactions affected by your change.





1. **Conclusions**

For this change request, we started out smoothly but there were many complications as we had issues with compiling our code. We searched through the code to figure out what we could have missed and continued to make changes. We had to try many things until we were finally able to complete the change request. Previous knowledge of the application helped us with the searching through the code; however, this change order was more complicated thus making it harder for us to make the change. It seemed that we were hung up on the java script longer than necessary and in hind sight might have chosen that database field to be character rather than Boolean if it makes the display easier.

Classes and methods changed:

* femr.business.services.system/PatientService.java
* femr.business.services.system/SearchService.java
* femr.common/IItemModelMapper.java
* femr.common/ItemModelMapper.java
* femr.data/DataModelMapper.java
* femr.data.daos.system/MedicationRepository.java
* femr.data.models.core/iPatient.java
* femr.data.models.mysql/Patient.java
* femr.ui.controllers/TriageController.java
* femr.ui.models.history/ ViewModel.java
* femr.ui.models.triage/IndexViewModelPost.java
* femr.ui.models.triage/ index.scala.html