# Change request log

# Team

Team:6

Raghupathi Vasantavada(Raghu)

Devika Chakraborty

# Change Request

FEMR-186:

1. Change the name of the "Treatment Given" field to "Procedure/Counseling"
2. Change the name of this field in the tab\_fields table to reflect the new name.

# Concept Location



|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
|  | Concept location identification for label change (186.i) |  |
| 1 | We created a new patient in the Triage screen | We want to interact with the system and check which screen/tab has the treatment given/treatment fields |
| 2 | We went to the Medical tab and searched for the Patient ID | In order to find the treatment given field/text, we have to go through these steps. |
| 3 | We clicked on the treatment tab and there we found the “Treatment Given” text which we have to change to “Procedure/Counseling” | In order to find the treatment given field/text, we have to go through these steps. |
| 4 | To find out in which file this “Treatment Given” text field is there, we use IntelliJ IDE and searched for “Treatment Given” through Edit->Find in path | To make changes to the text in the relevant location, we need to find the location in the code base |
| 5 | We found the text “Treatment Given” in one file only “treatmentTab.scala.html | Assumption is we need to make changes only in this file to reflect the change from “Treatment Given” to “Procedure/Counseling” |
|  | Concept location identification for 186.II |  |
| 1 | We did a search (edit->find in path) for the word “Treatment” to resolve the second issue. The search returned multiple files where | To see in which classes and methods Treatment is used |
| 2 | We opened up the tab\_fields table directly in the database and found there is a treatment column there. We renamed the database column to Procedure as per the second problem | We thought changing it directly in the database table will automatically be reflected in the front end as the second problem statement says  “Change the name of this field in the tab\_fields table to reflect the new name” |
| 3 | However, while testing, we found that after restarting the application, the same field “treatment” is getting created in the tab\_fields table. Hence we understood we haven’t made changes in the correct location. We also realized that there must be some startup sql scripts which run and reconfigure tab\_fields table , probably in femr.util.startup | Did a trial and error approach to make the change to treatment column in tab\_fields table directly in database in order to get it reflected in the front end, but did not work. |
| 4 | We realized there might be some database mapping classes which are setting the value of the database columns. So we searched for the word “Treatment” again in all the files in the code | Trial and error approach did not work. So we are trying to find if we can approach it logically and check where the value of the columns is being set in the code |
| 5 | We went to data/daos/core/IMedicationRepository ,its implementation and looked at the method CreateNewMedication,  And then we looked at where this method is being used and its being used in the MedicationService.java  We went to MedicationService.Java and figured out that the field “treatment” is not set anywhere in that method | Part of the search to find in which method “treatment” is being set |
| 6 | The next method where we found “treatment” being used is DatabaseSeeder.java.We realized that “Treatment” is used in tabFieldsType.class. The method “seedDefaultTabFields” is setting the default field names and their values in the tab\_fields table in the database. So one location that we need to make code change is this DatabaseSeeder.java | The database fields are getting set in the DatabaseSeeder.java. So this is one place we confirmed we have to make code changes |
| 7 | We have to check if we need more code changes in service/UI layer because of the change in the DatabaseSeeder.java | As a natural propagation of the changes that we made, we need to check UI /service layers are consistent with this change. |
| 8 | Next, we looked in the Business.services.system.java, it’s finding the columns from the database, not setting them. So we don’t need to change anything in this method | To verify we don’t have to change anything on the service layer as a result of the change in DatabaseSeeder.java |
| 9 | Next is UI, we go to the MedicalController.java, we searched the place where the treatment tab is being inserted in to the view and looked in the method structureTreatmentFieldsForView and this method is also getting the fields from tab\_fields table and not setting any fields. Hence we don’t have to touch the controller. | To verify we don’t have to change anything on the UI layer as a result of the change in DatabaseSeeder.java |
| 10 | We looked through femr.ui.views.partials.medical.tabs namespace/package , and under that corresponding treatmentTab.scala.html.  We figured out that the column name “treatment” is retrieved from TreatmentTab.getTabFieldItemByName  Also, tabFieldItems are referenced by @item name. We need to change “treatment” textfield’s name | We confirmed that we needed to change the Treatment in the file treatmentTab.scala.html. |

**Time spent (in minutes):** 115

# 

# Impact Analysis

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We made a list of methods that are used in DatabaseSeeder.java | To track the classes that could be impacted by the change. |
| 2 | We also checked all places where “treatment” is used i.e. in service layer, UI and Database layer. | To track the classes that could be impacted by the change. |
| 3 | We also looked at the service layer MedicationService.java to make sure to find if there is any impact of the code change in DatabaseSeeder.java | To track the classes that could be impacted by the change. |
| 4 | On the UI layer, we checked treatmentTab.scala.html to check if there is any impact. | To track the classes that could be impacted by the change. |

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

# 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 changed the text “Treatment Given” in “treatmentTab.scala.html to Procedure/Counseling | Since this is the only file with the text treatment given and it’s a UI change, we figured no other change is needed for this one |
| 2 | We looked at the tab\_fields table in the database femrdb  We deleted the treatment column using the following SQL statement  Delete from femr.tab\_fields where id =37; | The columns in the tab\_fileds table will get generated automatically and hence we need to delete unwanted columns. |
| 3 | We went to the DatabaseSeeder.java  and refactored “Treatment” to “Procedure” | The DatabaseSeeder.java is responsible for creating and retrieving the columns from the tab\_fileds table |
| 4 | Then we went to the UI ,TreatmentTab.scala.html and refactored “Treatment” label to Procedure/Counseling | As found in the context location, the label needs to be changed |
| 6 | On the same html page we refactored, in the TreatmentTab.getTabFieldItembyName, we refactored “Treatment” to procedure” | As found in the context location and to reflect the changes made to the treatment column to be propagated to and from the tab\_fileds table, treatment field name should change per the change request |
| 6 | We tested manually | To make sure everything works. |

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

# 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   1. Input: Restart the application (after resetting the server) 2. Create a new patient using Triage screen 3. Go to Medical tab, input patient ID 4. In the next screen, navigate to Treatment tab   Expected output: The Treatment given text should have been changed to Procedure/Counseling | We need to check while creating a new patient and entering the information that treatment given text is replaced by procedure/Counseling. This was the first requirement for defect#186  The test passed. |
| 2 | Test case defined:  Inputs:   1. Restart the application (after resetting the server) 2. Create a new patient using Triage screen 3. Go to Medical tab, input patient ID 4. In the next screen, navigate to Treatment tab 5. Fill in the necessary details and his ‘submit patient’ button     Expected output: The treatment tab should show “procedure” and the DB table tab\_fields should not contain the column ‘treatment’ | This test case is to check that database columns are not reset to default by application restart  The test passed |
| 3 | Test case defined:  Input:   1. Create a new patient using Triage screen 2. Go to Medical tab, input patient ID 3. In the next screen, navigate to Treatment tab 4. Fill in all the details in the treatment tab and hit submit patient button   Output:  the record should be successfully submitted and the message ‘record saved successfully’ should appear and verify the same in femrdb.tab\_fields table to verify the ”procedure” column exists and treatment column does not exist | System should allow saving of the treatment tab details after renaming/refactoring form fields.  The test passed |

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

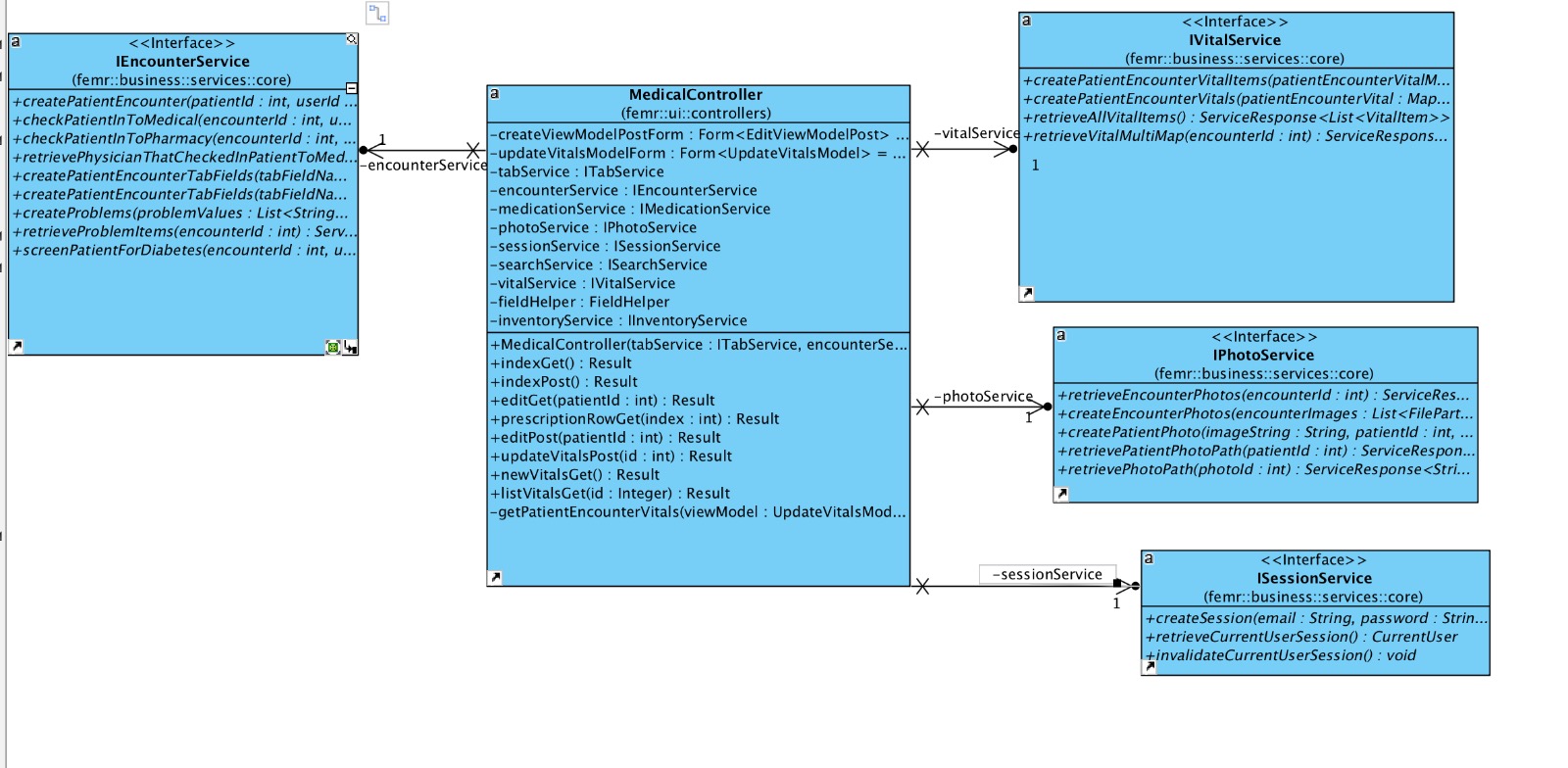
# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 115 |
| Impact Analysis | 30 |
| Prefactoring | 0 |
| Actualization | 120 |
| Postfactoring | 0 |
| Verification | 40 |
| Total | 275 |

# Reverse engineering

**Medical Class Diagram**



# Conclusion

For the change (ii) concept location was moderately challenging because it took time for us to identify how the data is being populated in the tab\_Fields table and how it is retrieved in the application. The impact analysis, actualization was moderately tough even though the architecture is relatively easy to understand.

The layers of the code are well defined. We performed manual verification that the fix is working.

Tables changed/Updated:

Tab\_fields

Column deleted: Treatment

Java Classes and methods changed:

femr.util.startup.DatabaseSeeder.java

femr.ui.views.partials.medical.tabs.treatmentTab.scala.html