**3.       Sprint 2**

3.1.    Sprint planning

After the second sprint meeting, we split the huge user story to smaller stories. And after that we calculated the amount of time we will have this sprint and added the most important user stories to the sprint first. We could exactly add all to this sprint if we have estimated the time right.

**Backlog**



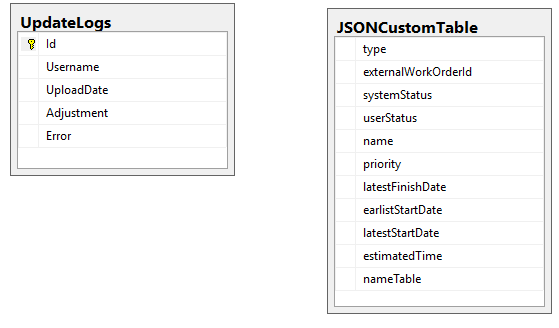
**Figure x.x.** *Backlog of sprint 2*

This is the completed backlog items in sprint 2. In this one, we still have a huge user story, the first one but it is smaller than before, but I should still have been break to smaller stories.

3.3.    GUI

!!!!!!!!!!!!!!!!111

3.4.    Data model



**Figure x.x** **Database table update*.***

So as you can see from last time, the “UpdateLogs “there is added to more columns, now it have Error and ID, the reason for Error is added, is because then that row can easily be highlighted, so you can see there the errors. ID is added. So the log can add the log class one by one, so you can see that it load the data.

The “JSONCustomTable” was added, so the user could add the customized format for how they want the json file to look like. So now they can made their own customization and add it to SQL database.

3.5.    Implementation

In this sprint Is added :

* Multithreading
* Connection pool
* Facade pattern to BLL layer
* Singleton to model
* Convert csv
* Customize how the json file should look like
* Start, stop and pause threading
* Made a log for what each user does in the program
* Moderator able to add user to the program

3.5.1. Code examples



**Figure x.x Thread**

So for the conversionProgress method , it start with create a CompletableFutere, it is use because if one or more thread are trying to complete or cancel a CompletableFutere, only one of the them will succeeds.

runAsync means that it will return a new CompletableFutere then it is done, so it will continuous to run to there is no more files in the list. But if we run something their will change the view in this method without Platform.runLater, it will freeze.

In runAsync we have the method convertfile(File), that is their the file their convert the files to json files.

So that why we use Platform.runLater, is a JavaFX application thread. It will run this Thread at some point, it will add the event to an event queue so it know what to do, then this thread run.

So in this Platform.runLater, it will remove the file from the model, so you can see how many files you still need to convert on the view. There is also a progress bar so you can see how far you are with convention files, the progress bar Is calculated, by the amount of files converted divided by the amount of files their need to be converted.

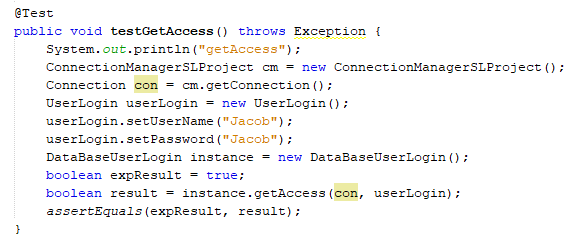
3.5.2. Design Patterns

In this sprint was implemented Connection pool, the reason for that is, it will keep the connection to the database for a specific amount of time, so we do not have to connect to it each time we will get some information or save. That will optimize the program a lot.

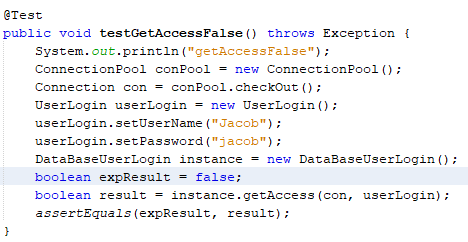
In the models, there is used singleton so, there will only be created one model, so there will be no null pointer exception if you try to create a new model. Because you only can create one instance of this class.

3.5.3. Unit test

In this application, there is not must to use unit test on. So the unit test was done on some of the logic from the database.



This was one of the test, there was done, ass you can see, we need to create a connection to the database so we can get the data form the SQL database and send a userLogin with the right password and username to the DataBasesUserLogin, and the logic in that class will return true if that match the userLogin in the SQL database table.



The second test in this class in the opposite, their in BE Class UserLogin send the wrong password to the current username, so we expect it will be false, assertEquals will fail the test if expResult and result not is the same.

3.6.    Sprint Review

This was the second sprit review meeting with the client, and he told us that.

* Buttums should be easier to see that they are bottom.
* Missing array in the JSON file .
* Naming of the button should be more precise.
* Add information so it is easy to fix the bug for the customer.
* See the users.

3.7.    Sprint Retrospective

|  |
| --- |
| **Good**   * We did what we want to do. * Made the requirements for this project. |
| **Could have been better**   * Better working together * More meetings. |
| **Improvement**   * Write better notes of what we did. * Discus how it should look like in program so we do not waste to must time |