Path

--module-path

"C:/Program Files/Java/JavaFx/javafx-sdk-17.0.1/lib"

--add-modules

"javafx.controls,javafx.fxml"

**30/02/2022**

Activities

-review with supervisor

-domain model(simplified)

-task sub division with expected timings

Requirements for the sprint:

1. As a librarian, I want to be able to add multimedia items to the system, so library users can borrow them.

6. As a librarian, I want to be able to remove multimedia items from the system, so multimedia items will not belong to the system anymore.

Tasks:

-Use case Diagram 5min

-Test description 1h

-brief description 45 min

-table Main flow

-table At flow 2H

-diagrams 2h

-class Diagram 5h

-ScenemBuidel 1h

-DB EER 2h

-global 1h

-mvvm 2h

-java doc 30 min

-db script 1h

-connect 8h

-TDD model 4h

-Test junit 4h

**20/04/2022**

Retrospective

-Improve communication

-Define task clearly

-Improve github

-Use asana when interacticting with tasks

-Book rooms

-More daily meetings

-Don’t refactor before a spring meeting

Activities

-presentation of the artefacts to the supervisor

-rectify asana task view

-work together on class diagram task

Tasks not done previous sprint:

-class diagram(update class diagram) 30 min

-Gui scene builder 15 min

-db connection (put and remove info of book and magazine in to the database server side) 8 h

-db connection (put and remove info of book and magazine in to the database GUi side) 8 h

-junit 1h

-add date to magazine 1h

Requirements for the sprint:

1. As a librarian, I want to be able to add multimedia items to the system, so library users can borrow them. (sprint 1)

2. As a library manager, I want to be able to add librarians to the system, so they can start using it once they are hired. (sprint 1)

5. As a library manager, I want to be able to remove librarians from the system, so they cannot access it anymore if they are fired or gone.

6. As a librarian, I want to be able to remove multimedia items from the system, so multimedia items will not belong to the system anymore.

New tasks:

-Use case diagram 5 min

-Use case brief description 1h

-Test case (add remove librarian) 2h

- use case table Main flow 1h 30 min

-use case table Alternative flow 1h 30 min

-system sequence diagram 1h(req use case table main flow)(optional)

- sequence diagrams (optional)

- activity diagrams (optional)

-class Diagram 6h (update of the last class diagram)

-ScenemBuidel 1h

-DB EER 1h

-logical model db 1h

-global relational diagram db 1h

-general mvvm (connect GUI with mode, add librarian manage and login class to model) 4h

-javaDoc( create every time a class/method is created)

-create table in database (sql) 1h

-connect db to server as a manager(be able to retrieve and put manager librarian info in to the database) 5h

-cleint/server implemtation as a manager(create client for manager) 2h

-Test junit 5h

**04/05/2022**

Retrospective

-Divide junit in more part

-Create more branches

-Push only working versions

-Organize more daily scrum meetings

-Add and remove stuff should be 2 different tasks

Requirements for the sprint:

1. As a librarian, I want to be able to add multimedia items to the system, so library users can borrow them. (sprint 1)

2. As a library manager, I want to be able to add librarians to the system, so they can start using it once they are hired. (sprint 1)

5. As a library manager, I want to be able to remove librarians from the system, so they cannot access it anymore if they are fired or gone. (sprint 2)

6. As a librarian, I want to be able to remove multimedia items from the system, so multimedia items will not belong to the system anymore. (Sprint 2)

3. As a librarian, I want to add a library user to the system, so the librarians can lend multimedia items to them.

Tasks not done previous sprint:

\* sprint 1

-[DB Connection Magazine (put and remove info of magazine in to the database GUI side) 8h](https://app.asana.com/0/1201982972614892/1202153155021844)  
-[JUnit model 2h sprint 2](https://app.asana.com/0/1201982972614892/1202222749766375)  
-[JUnit viewModel 2h sprint 2](https://app.asana.com/0/1201982972614892/1202222749766377)  
-[JUnit Server/Client 1h sprint 2](https://app.asana.com/0/1201982972614892/1202222749766379)

-[JUnit Storage 1h sprint 2](https://app.asana.com/0/1201982972614892/1202222749766381)  
-[DB Implementation Book 2h](https://app.asana.com/0/1201982972614892/1202222749766383)

\*sprint 2

-[JUnit model 2h sprint 2](https://app.asana.com/0/1201982972614892/1202153155021840)  
-[JUnit viewModel 2h sprint 2](https://app.asana.com/0/1201982972614892/1202222710641480)  
-[JUnit Server/Client 1h sprint 2](https://app.asana.com/0/1201982972614892/1202222710641484)  
-[JUnit Storage 1h sprint 2](https://app.asana.com/0/1201982972614892/1202222710641486)  
-[DB Implementation](https://app.asana.com/0/1201982972614892/1202222749766385)

New tasks:

- class diagram update 4h

-SQL 1 h

-logical model 1h

-EEr 1h

-Junit view Model 3h

-junit model 1h

-Junit client/server 3h

-Junit storage 2h

-Db implementation 3h

-client server implementation 2h

-View / viewModel/mediator 3h

-Main flow table 1h

-Alt flow 1h

-test case table 3h

- Update Use Case diagram

**11/05/2022**

Fullfilement req:1,2,5,6

Activities:

-Presentation: magazine that works till the database

-Presentation: Book genre is not working

-Presentation: LIBRARIAN WORK WITH FAKE STORAGE

-Presentation of the test

We are having an irresolvable db problem(googled it didn’t find the answer)

After solving the db problem we should merge in to the main

We are able to add librarian to the db

Retrospective:

-Github usage is still subpar

Requirements for the sprint:

3. As a librarian, I want to add a library user to the system, so the librarians can lend multimedia items to them.

20. As a librarian, I want to be able to delete a specific library user, so they won’t be able to borrow any multimedia item anymore.

Plans for future:

We decide to create a new sprint with the old tasks of the not done tasks

We decide no more implementation after 25

Tasks not done previous sprint:

(all new tasks from sprint 3)

- class diagram update 4h

-SQL 1 h

-logical model 1h

-EEr 1h

-Junit view Model 3h

-junit model 1h

-Junit client/server 3h

-Junit storage 2h

-Db implementation 3h

-client server implementation 2h

-View / viewModel/mediator 3h

-Main flow table 1h

-Alt flow 1h

-test case table 3h

- Update Use Case diagram

New tasks:

none

**14/05/2022**

Retrospective

-We improved the use of github

-We improved the organization of work

-We should revise the junit testing

-Working in the same room seems to be the best strategy

-Sprint end 18/05/2022

Tasks not done previous sprint:

-View model library user is not done

-Implementation mvvm is not done

-Fix add/remove library user

Requirements for the sprint:

4. As a librarian, I want to lend a specific item of multimedia to a specific library user for 14 days, so they can borrow it.

New tasks:

-Brief description

-Main flow

-Alternative flow

-Test case table

-DB eer

-DB global relations

-Db logical model

-DB sql()

-Junit server/client

-Junit storage

-Junit model

-Junit view model

-Implementation model

-Implemetation client/server

-Implementation mvvm

-Implementation DAO

-Class diagram

-Scene builder

**19/05/2022**

Fulfilled req: 4, 3, 20

Retrospective

-Fix old stuff before starting with the new

-Design UI together

-Be more careful with the Exeptions

-(don’t send sql exeptions to the view)

-Improve the hour expectation of the tasks

Tasks not done previous sprint:

-Junit

-Fix double bindings

-Update class diagram

New requirements

21. As a librarian, I want to return a specific item of multimedia to a library user borrowed for 14 days, so other library users can borrow the item.

New tasks:

-Brief description 1h

-Main flow 2h

-Alternative flow 2h

-Test case table 3h

-Junit server/client 2h

-Junit storage 3h

-Junit view model 6h

-Implemetation client/server 3h

-Implementation mvvm 8h

-Implementation DAO 3h

-Class diagram 6h

-Scene builder 1h

**24/05/2022**

Fulfilled req:21

Retrospective

-Sprint good

-Write a quick message if the daily meeting is not possible

Task not done previous sprint:

Junitn view model return Item

Lunit view model loan book