# X.XX Construction X iteration – User defined spreadsheet

## X.1 Purpose

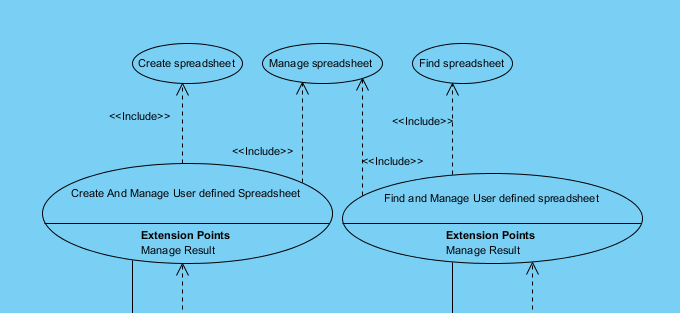
This iteration is divided into several use cases 2 “head” and 3 “sub”, see figure x 1 - UC Diagram.  
The user want to create “spreadsheet” where he can define the size himself and alternative give the headers name. It should be possible to add data in the spreadsheet and save the data to later use. It should also be possible to alter in the data and save it again.   
There is 2 alternatives either we could let the user create a spreadsheet in MS or OpenOffice and upload it to the system. Or we could create our own tables and present them integrated in the system. The problem is to save the table, since we don’t know what size and what kind of data we should store, it would be quit a job to create tables in a database and structure them. So we will try to store them as serializable object on the disc. There are 2 flows in this. Flow 1 define and create the spreadsheet, fill in data and save it. Flow 2 open existing spreadsheet alter data and save it again or delete spreadsheet. This will produce the following use cases, Create and Manage user defined spreadsheet (head use case) this include Create spreadsheet and Manage spreadsheet (sub use case) and Find and Manage user defined spreadsheet (head use case) this include Find spreadsheet and Manage spreadsheet (sub use case). You should also note that the 2 head use cases have a  
extension that is Manage Result, there is a part part of the manage spreadsheet flow, see SD shown in X.7.1.

Figure x – Use case diagram for Spreadsheet

## X.2 Model the domain

We have not added any conceptual classes on the domain model, since the spreadsheet is pure technical issues. They will be presented at the domain model as Result of type media. It have been quite a challenge to figure how this will work.   
At first we had thought the spreadsheet as a Result and therefore it was on the domain model, but one thing was drawing another was coding, it would not “fit” together. The end was that the data inside the user defined spreadsheet was the result, but not the shell spreadsheet.

## X.3 Detailed Use case

### X.3.1 Write use case text

### Create And Manage User defined Spreadsheet - Use Case

### Use Case Details

|  |  |  |
| --- | --- | --- |
| **Name:** | | Create And Manage User defined Spreadsheet |
| **Rank:** | | Medium |
|  | |  |
|  | |  |
|  | |  |
| **Stereotype:** | | UseCase |
| **Abstract:** | | false |
| **Documentation:** | |  |
| Flow of Events | | | |
|  | 1. Need for creating user defined spreadsheet | | |
|  | 2. Image0.png [User](#Vai64dSAUNn.8jPa) start  Image1.png [Create And Manage User defined Spreadsheet](#hGF64dSAUNn.8j0r:RT1u4TSAUNn..VdH) | | |
|  | 3. Image0.png [User](#Vai64dSAUNn.8jPa) start  Image1.png [Create spreadsheet](#hGF64dSAUNn.8j0r:G30b1TSAUNn.pYKD) | | |
|  | 4. System present spreadsheet | | |
|  | 5. Image0.png [User](#Vai64dSAUNn.8jPa) start  Image1.png [Manage spreadsheet](#hGF64dSAUNn.8j0r:BQhO2TSFS_gmRJx9) | | |
|  | 6. Image0.png [User](#Vai64dSAUNn.8jPa) end  Image1.png [Create And Manage User defined Spreadsheet](#hGF64dSAUNn.8j0r:RT1u4TSAUNn..VdH) | | |

Use Case Details

|  |  |
| --- | --- |
| **Level:** | User |
| **Complexity:** | High |
| **Use Case Status:** | Name Only |
| **Implementation Status:** | Scheduled |
| **Preconditions:** | User have data and know how many rows and columns there should be used |
| **Post-conditions:** | Spreadsheet with data or emptyspreadsheet is saved |
| **Author:** | clausbeck |
| **Assumptions:** | Frequency of occurrence: 100 times pr. month |

### Find and Manage User defined spreadsheet - Use Case

### Use Case Details

|  |  |  |
| --- | --- | --- |
| **Name:** | | Find and Manage User defined spreadsheet |
| **Rank:** | | Unspecified |
|  | |  |
|  | |  |
|  | |  |
| **Stereotype:** | | UseCase |
| **Abstract:** | | False |
| **Documentation:** | |  |
| Flow of Events | | | |
|  | 1. Image0.png [User](#Vai64dSAUNn.8jPa) start  Image1.png [Find and Manage User defined spreadsheet](#hGF64dSAUNn.8j0r:arab1TSAUNn.pYYT) | | |
|  | 2. Image0.png [User](#Vai64dSAUNn.8jPa) start  Image1.png [Find spreadsheet](#hGF64dSAUNn.8j0r:06Sb1TSAUNn.pYRc) | | |
|  | 3. System present spreadsheet | | |
|  | 4. Image0.png [User](#Vai64dSAUNn.8jPa) start  Image1.png [Manage spreadsheet](#hGF64dSAUNn.8j0r:BQhO2TSFS_gmRJx9) | | |
|  | 5. Image0.png [User](#Vai64dSAUNn.8jPa) end  Image1.png [Find and Manage User defined spreadsheet](#hGF64dSAUNn.8j0r:arab1TSAUNn.pYYT) | | |

### Manage spreadsheet - Use Case

### Use Case Details

|  |  |
| --- | --- |
| **Name:** | Manage spreadsheet |
| **Rank:** | Medium |
|  |  |
|  |  |
|  |  |
| **Stereotype:** | UseCase |
| **Abstract:** | false |
| **Documentation:** |  |

Flow of Events

|  |  |
| --- | --- |
| Flow of Events | |
|  | 1. Image0.png [User](#Vai64dSAUNn.8jPa) starts manage spreadsheet |
|  | 2.if Spreadsheet is empty |
|  | 2.1.   Image0.png [User](#Vai64dSAUNn.8jPa) enters data |
|  | 3.else if Spreadsheet is filled |
|  | 3.1.   Image0.png [User](#Vai64dSAUNn.8jPa) modifies data |
|  | 4.else if Image0.png [User](#Vai64dSAUNn.8jPa) want to save emtpy spreadsheet |
|  | 4.1. Go to step 5 |
|  | end if |
|  | 5. System save spreadsheetdata as Result |
|  | 6. Image0.png [User](#Vai64dSAUNn.8jPa) ends manage spreadsheet |

Use Case Details Details

|  |  |
| --- | --- |
| **Level:** | User |
| **Complexity:** | Medium |
| **Use Case Status:** | Base |
| **Implementation Status:** | Started |
| **Preconditions:** | SpreadsheetCatalog exist. Spreadsheet is selected or present |
| **Post-conditions:** | Spreadsheet is modified or filled or empty and saved |
| **Author:** | clausbeck |
| **Assumptions:** | Frequency 300 times pr. year |

### Create spreadsheet - Use Case

### Use Case Details

|  |  |  |
| --- | --- | --- |
| **Name:** | | Create spreadsheet |
| **Rank:** | | Unspecified |
|  | |  |
|  | |  |
|  | |  |
| **Stereotype:** | | UseCase |
| **Abstract:** | | false |
| **Documentation:** | |  |
| Flow of Events | | | |
|  | 1. Image0.png [User](#Vai64dSAUNn.8jPa) starts  Image1.png [Create spreadsheet](#hGF64dSAUNn.8j0r:G30b1TSAUNn.pYKD) | | |
|  | 2. Image0.png [User](#Vai64dSAUNn.8jPa) enters spreadsheet size and column names (rows, columns, columnNames) | | |
|  | 3. System present spreadsheet | | |
|  | 4. Image0.png [User](#Vai64dSAUNn.8jPa) ends  Image1.png [Create spreadsheet](#hGF64dSAUNn.8j0r:G30b1TSAUNn.pYKD) | | |

Use Case Details

|  |  |
| --- | --- |
| **Level:** | User |
| **Complexity:** | Low |
| **Use Case Status:** | Complete |
| **Implementation Status:** | Complete |
| **Preconditions:** | User are logged in. User knows size of spreadsheet. If needed user knows name on columns |
| **Post-conditions:** | Spreadsheet is created and present for user |
| **Author:** | clausbeck |
| **Assumptions:** | Frequency: Approximately 400 pr year |

### Find spreadsheet - Use Case

### Use Case Details

|  |  |  |
| --- | --- | --- |
| **Name:** | | Find spreadsheet |
| **Rank:** | | Unspecified |
| **Justification:** | |  |
| **Leaf:** | | false |
| **Root:** | | false |
| **Stereotype:** | | UseCase |
| **Abstract:** | | false |
| Flow of Events | | | |
|  | 1. User starts  Image1.png [Find spreadsheet](#hGF64dSAUNn.8j0r:06Sb1TSAUNn.pYRc) | | |
|  | 2. Image0.png [User](#Vai64dSAUNn.8jPa) select a workstation | | |
|  | 3. Image0.png [User](#Vai64dSAUNn.8jPa) enters vehicleID | | |
|  | 4. System present spreadsheets attached to workstationID and vehicleID | | |
|  | 5. Image0.png [User](#Vai64dSAUNn.8jPa) select correct spreadsheet | | |

Use Case Details

|  |  |
| --- | --- |
| **Level:** | User |
| **Complexity:** | Medium |
| **Use Case Status:** | Initial |
| **Implementation Status:** | Scheduled |
| **Preconditions:** | User is logged on. Spreadsheet catalog exist. |
| **Post-conditions:** | Spreadsheet is found. |
| **Author:** | clausbeck |
| **Assumptions:** | Frequency: Aprroximately 400 timer pr. year |

### X.3.2 System Sequence Diagram – SSD

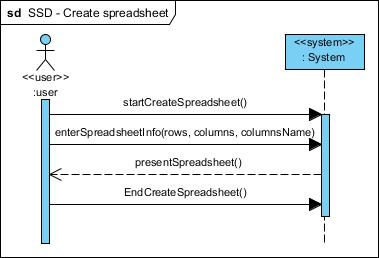
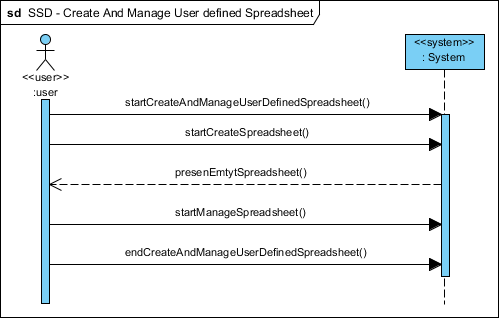


Figure x - Start for creating a spreadsheet, is taken to Figure x 3 there you enter rows, columns and alternative names for column, afterwards you will return to here and continue to the Figure x 4 Manage Spreadsheet where you will enter or edit data, and save (one flow) the spreadsheet data as a result. And return to here.

Figure x - Enter the basic information for creating a user defined spreadsheet.

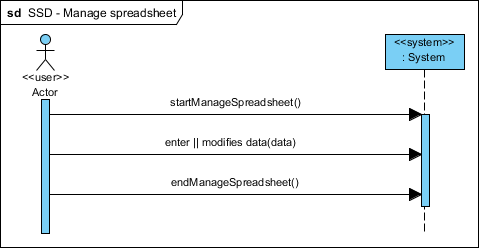


Figure x – Enter or modify data in the spreadsheet. Or delete data or complet spreadsheet, and return either to Figure x 2Create and Manage or Figure 5 Find and Manage use case.

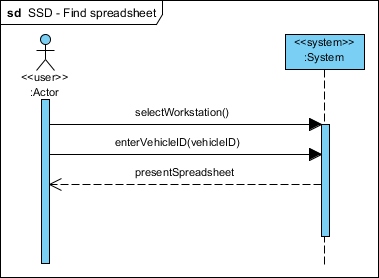
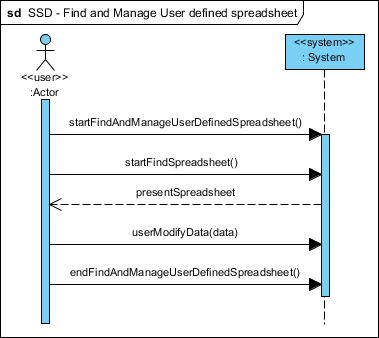


Figure x – Searchcriteria is set as vehicleID and are found if exist.

Figure x – Start for finding an existing spreadsheet, is taken to Figure x 6 there searchcriterias are set, afterwards you will return to here and continue to the Figure x 4 Manage Spreadsheet where you will enter or edit data, and save (one flow) the spreadsheet data as a result. And return to here.

### X.3.3 Operation Contracts

Since we have decided that spreadsheet is not a part of the system as itself but just a technical issue it will not create an object, the data from the spreadsheet saved as a Result will create an object, but then the operation contract will be made in the Result use case

## x.4 Architectural analysis

Spreadsheet have place in the model layer as Spreadsheet and SpreadsheetCatalog

## x.5 Use case analysis

## x.6 GUI Design

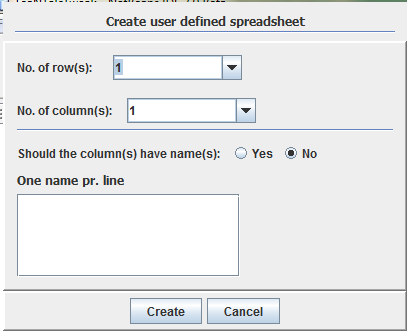
As rest of the system it will be presented in Nimbus style if possible. The spreadsheet itself is just a JTable there will be integrated in the Result GUI. There is one GUI for creating the spreadsheet.  


Figure x – View of Create spreadsheet GUI. It is possible to defined how many row and columns there should be. The combobox is filled with 1-1000, and is auto decorated (meaning that you starts enter the number you want and it will be presented). There is a choice of giving the column names and enter the name(s) in the text area, one pr. line.

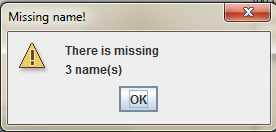
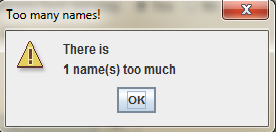
If you have chosen to give the column names, and you enter to many or not enough one of these two error messages will be presented.  


Figure x – One name to much is entered in the Create GUI

Figure x – There is missing 3 names in the Create GUI

## X.7 Use case Design

### X.7.1 Sequence Diagram (Dynamic View)

See next page (A3 fold out)

### X.7.2 Design Class Diagram

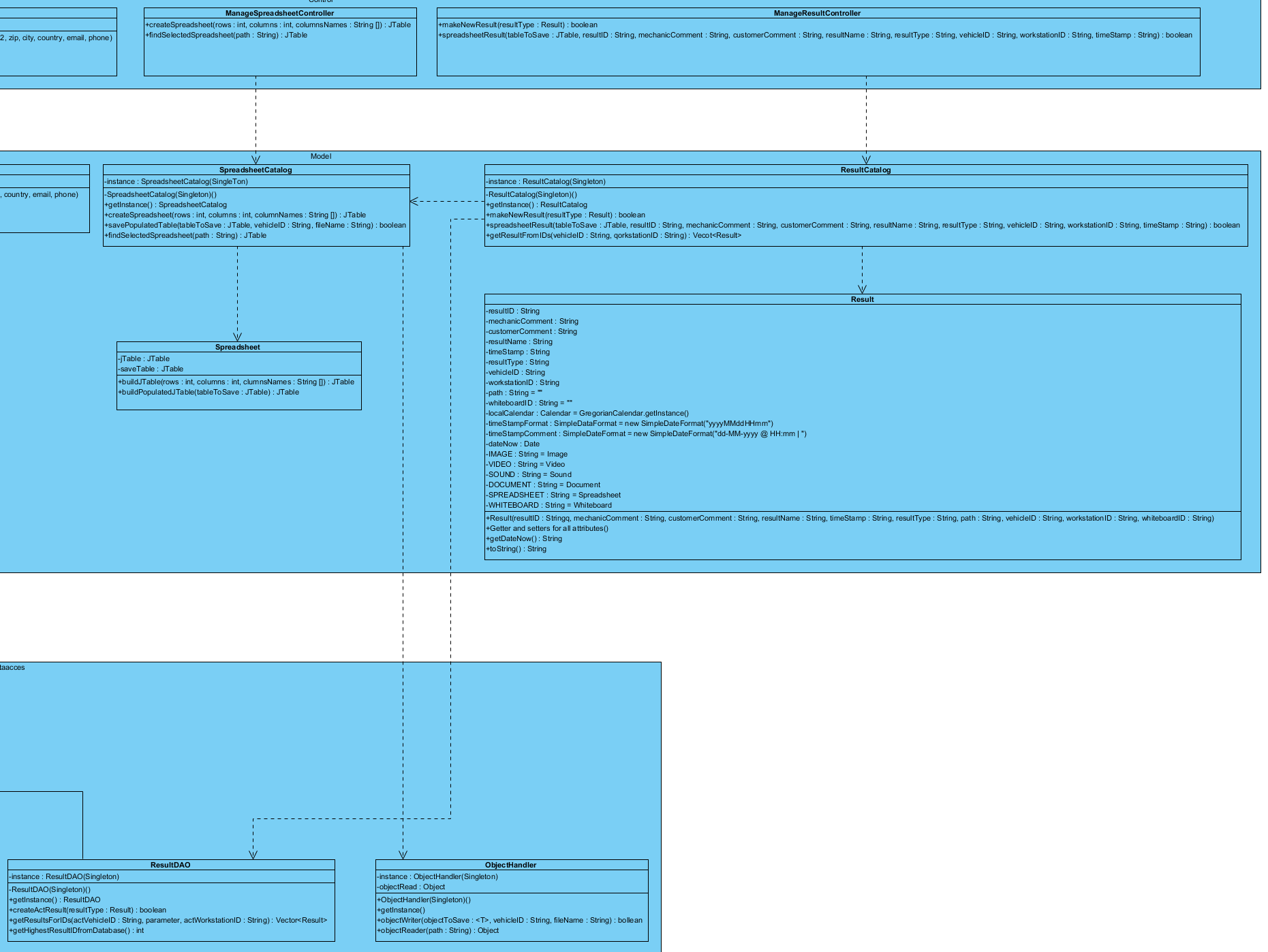
Here is the picture for how the different classes talk together. It gives a view over how we start Creating a spreadsheet and how we can get a Result out of it. As shown the SpreadsheetCatalog is connected with the ObjectHandler in the dataaccess layer, the ObjectHandler will write the Spreadsheet as a Serializable table to a given folder on the disc, this give a path which we will add in the Result. Therefore is ResultCatalog and SpreadsheetCatalog connected. ResultCatalog will talk with ResultDAO there will create a row in the ResultTable in the database. And this is the explanation on why Create and Manage spreadsheet have an extension to ManageResult in the use case diagram.   
X.7.3 Database Design

Figure x – View of the connections between Spreadsheet and Result, and how it is saved in a file and in database.

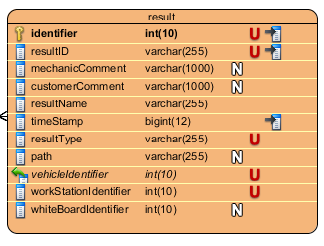
This table is taken to 3NF  


Figure x Result Table, holds all information about a result. If the result is upload(saved file on disc) there will be given a string in path attribute, to locate the file again. If the result is a whiteboard is it identified by a foreignkey from a WhiteboardTable (not design yet, an overview can be seen on the Domain Model). A result is always connected to a vehicle (and order comes automatically thrue the vehicle) and a workstation, it is only possible to see a result on the workstation it’s done.