Analyse Assignment 2

Requirements

27-09-2015

Steven Schenk, Robert Kraaijeveld, Cees-Jan Nolen, William Visser, Robin Bakker.

Inhoudsopgave

**Exercise A. 3**

User Stories 3

Functional Requirements5

Nonfunctional Requirements7

Use case and use case descriptions 9

**Exercise C. 20**

Pro’s 20

Con’s 21

# Exercise A.

## User stories:

### B1: The system will be able to read and save CityGis-data at high speeds.

1 As a User, I want to be able to import CityGis-data in CSV format into the system, so that data can be analyzed. (**M**)

2 As a User, I want the system to be as fast as possible, so I don’t have to wait too long. **(M)**

3 As a User, I want to have a simple graphical interface for importing Data, so I can easily store my data. (**M)**

4 As a user, I want to be able to see a progress bar, so I can see how the importing process is coming along. **(S)**

5 As a User, I want the system to inform me of any possible errors in the importing process, so I know what went wrong. (**S**)

### B2: The systems’ architecture will enable the creation and the saving of parallely aggregated reports

1 As a User, I want to be able to download reports about the collected data and about more specific parts of that data, so I don’t have to go through all the data manually. (**M**)

2 As a User, I want the system to automatically aggregate useful information about the collected data, so I don’t have to select what I want in the report by hand. (**M**)

4 As a User, I want to be able to download these generated reports in pdf, so I can easily distribute these reports through to my colleagues. (**S**)

5 As a User, I want to be able to generate and download multiple reports at the same time, so I do not have to wait for each one of them to finish before the other. (**M**)

6 As a User, I want to be able to see progress bar of the generating/downloading process, so I know when I can safely quit the program. (**S**)

### B4: The system will provide a web-interface in which vehicle maintenance-data can be made into reports and downloaded.

1 As a User, I want the web-interface to be easily understood and navigated, so I do not need to be an expert-data analyst in order to use it. (**M**) *(See also Business Requirement B3)*

2 As a User, I want the system to automatically aggregate useful information about the collected data, so I don’t have to select what I want in the report by hand. (**M**) *(See also Business Requirement B2)*

3 As a User, I want to be able to download reports about the collected data, so I don’t have to go through all the data manually. (**M**) *(See also Business Requirement B2)*

5 As a User, I want the system to also represent the vehicle-maintenance data in an easy-to-understand visual way, so I can easily see the trend of the data. (**S**)

### B5: The system will provide a web-dashboard on which some important vehicle-maintenance data points will be shown to the vehicle-administrator.

1 As a User, I want any alerts coming from the web-dashboard to be easily visible (from across all the web-interfaces), so I can address these issues quickly. (**M**)

2 As a User, I want alerts coming in the web-dashboard to contain basic information about the problem at hand, so I can quickly see what the issue is. (**M**)

4 As a User, I want the system to allow me to quickly share the information in the alerts, so I can alert my superiors. (**S**)

### B6: The system will provide a web-portal on which CityGis-meta data will be up for sale to 3rd parties.

1 As a User, I want to buy data from CityGis, so I can use it for my personal purpose. **(M)**

2 As a User, I want to be able to pay via PayPal, so I can easily transfer the money. **(M)**

3 As a User, I want to be able to select which data I want to buy, so I only have to buy the data that I need. **(S)**

# Functional requirements

## Business requirement 1: The system will be able to read and save CityGis-data at high speeds.

### Importing CityGis files

The system will be able to import CityGis files according to the assumptions provided in chapter 2.6 of this document. *Priority: High*

### Importing GUI

The system will have a graphical user interface in which the user will be able to import data-files, according to the Interface requirements of chapter 4 of this document. *Priority: High*

### Importing-process error-reporting

The system will inform display any kind of errors that interrupt the import process to the user. *Priority: Medium*

## Business requirement 2: The systems’ architecture will enable the creation and the saving of parallel aggregated reports

### Selecting data

The system will enable the user to select about which data-parts he wants to create a report. *Priority: High*

### Report aggregation

The system will automatically aggregate information from the chosen data-parts and enter these into the report when it is generated. *Priority: High*

### Downloading of aggregated reports

The system will enable the downloading of reports about the data-parts to the PC of the user. *Priority: High*

### Report Parallel downloading

The system will enable the generation and downloading of multiple reports at the same time. *Priority: High*

## Business requirement 3: The system will provide a web-interface that will display CityGis-software and hardware performance statistics for use in product development

### Performance statistics web-interface

The system will provide a simple web-interface on which hardware- and software performance statistics will be displayed. *Priority: High*

### Performance Measurements

The web interface will display aggregated measurements and averages of the software- and hardware-data. (which measurements the system will used will be determined at a later phase together with the Product Owner) *Priority: High*

### Performance-Report downloading

The web interface will allow the generating and downloading of reports about the performance statistics. *Priority: High (See also Business requirement B2)*

## Business requirement 4: The system will provide a web-interface in which vehicle maintenance-data can be made into reports and downloaded.

### Maintenance-data Web interface

The system will provide web-interface on which maintenance data statistics will be displayed. *Priority: High (See also Business requirement B3)*

### Maintenance-data Aggregation

The system will automatically aggregate information from the data-parts and enter these into the report when it is generated. *Priority: High (See also Business requirement B2)*

### Maintenance-data Downloading

The system will enable the downloading of reports about the data-parts. *Priority: High*

*(See also Business requirement B2)*

### Maintenance-data Visual representation

The web interface will represent the measurements and averages in a visual way. *Priority: Medium*

*(See also Business requirement B3)*

## Business requirement 5: The system will provide a web-dashboard on which some important vehicle-maintenance data points will be shown to the vehicle administrator.

### Alert visibility

The system will make sure that any alerts coming from the web-dashboard will be visible across al other web-interfaces. *Priority: High*

### Alert information

Any alerts coming in on the web-dashboard will contain information about the problem at hand. *Priority: High*

### Alert Forwarding

The web-dashboard will allow the user to forward the alert to his/her superiors.

*Priority: Medium*

## Business requirement 6: The system will provide a web-portal on which CityGis-meta data will be up for sale to 3rd parties.

### Meta-data buying

The system will allow third-party buyers to buy CityGis meta data. *Priority: High*

### Meta-data payment client

The system will allow third-party buyers to pay for their meta data using an external payment client. *Priority: Medium*

### Meta-data Specification

The system will allow the users to specific pieces of meta-data that they want to buy. *Priority: High*

# Non-functional requirements

## Business requirement 1: The system will be able to read and save CityGis-data at high speeds.

### Importing Error reporting

The system should not report trivial errors, only those that can interrupt or otherwise damage the importing process.

### Importing progress bar

The system will have a progress bar which will inform the user of the progress the importing process has made. *Priority: Medium*

### Importing Speed

//The system save imported files as fast as possible.

## Business requirement 2: The systems’ architecture will enable the creation and the saving of parallel aggregated reports.

### Report Limits

The system will have a limit of 5 as to the amount of reports that can be created at the same time, so the system doesn’t crash unexpectedly.

### Report Format

The system should generate the reports in .pdf format.

## Business requirement 3: The system will provide a web-interface that will display CityGis software- and hardware performance statistics, for use in product development.

### Performance report downloading Limits

The system will have a limit of 5 as to the amount of reports that can be created at the same time, so the system doesn’t crash unexpectedly.

### Performance reports downloading GUI

The system will have a progress bar which will inform the user of the progress the generating process has made.

## Business requirement 4: The system will provide a web-interface in which vehicle maintenance-data can be made into reports and downloaded.

### Maintenance data limits

The system will have a limit of 5 as to the amount of reports that can be created at the same time, so the system doesn’t crash unexpectedly.

## Business requirement 5: The system will provide a web-dashboard on which some important vehicle-maintenance data points will be shown to the vehicle-administrator.

*None.*

## Business requirement 6: The system will provide a web-portal on which CityGis-meta data will be up for sale to 3rd parties.

### Designation

The system should only put designated meta-data up for sale.

# Use cases and use case descriptions

## BR1

## 1 Brief Description

This use case describes how a user can import City-gis data to the system for analyzing   
in this usecase userstories 1-5 of business requirement 1 are put together.

## 2 Actors

## User

## System

## 3 Preconditions

There is an active network connection from the user to the system.   
   
The user has city-gis data provided in .csv-format.

## 4 Basic Flow of Events

1. The use case begins when the user selects the data on his/her PC.

2. The user uploads the selected data to the system/server

3. The system shows a progress bar while uploading the data

4. The system shows a confirmation when the upload has finished

## 5 Alternative Flows

## Error

If in step 3 of the basic flow the system gives an error, then

1. The use case ends with a failure condition. And the user will know what has gone wrong.

## User closes program

If in step 1, 2 or 3 of the basic flow the user closes the program, then

1. The use case ends. And the user won’t get noticed since the program has closed.   
The system will delete the file that was being uploaded.

## 6 Key Scenarios

## Uploading Data to the server/system

## 7 Post-conditions

## Successful Completion

The user has received a confirmation that his/her data has been uploaded

## Failure Condition

The data hasn’t been transferred to the system/server

## 8 Special Requirements

None

## BR2

## 1 Brief Description

This use case describes how a user can download reports of city-gis data   
in this usecase userstories 1, 3-6 of business requirement 2 are put together.

## 2 Actors

## User

## System

## 3 Preconditions

There is an active network connection from the user to the system.

## 4 Basic Flow of Events

1. The use case begins when the user requests a report of the data

2. The system returns a report

3. The browser of the user starts downloading the report

4. The system shows a progress bar of the downloading.

5. The user receives a confirmation when the download is completed.

## 5 Alternative Flows

## User closes program

If in step 3 of the basic flow the user closes the program, then The use case ends with a failure condition. The download will be aborted.

## User requests multiple files

If in step 1 of the use case the user askes for multiple reports, then the System will return multiple reports. Step 4 of the use case: There will be multiple progress bars to show the downloading files

## 6 Key Scenarios

## 6.1 Downloading reports from the server/system

## 7 Post-conditions

## Successful Completion

The user has received a confirmation that his/her report has been downloaded

## Failure Condition

The report hasn’t been downloaded from the server

## 8 Special Requirements

None

## BR2

## 1 Brief Description

This use case describes how a the system processes data to make a report   
in this usecase userstory 2 of business requirement 2 are put together.

## 2 Actors

## User

## System

## 3 Preconditions

There is an active server for the processing of the system

## 4 Basic Flow of Events

1. The use case begins when there is a trigger of a user that wants to download a file

2. The system searches for and selects an available server

3. The server starts processing to make a report

4. Once the processing is done the server gives a confirmation to the system

5. The system stores the file and makes it ready to download.

## 5 Alternative Flows

## All servers are busy

If in step 2 of the basic flow the system can’t find an available server, then the use case ends with a failure condition. The system returns an error with error-information.

## 6 Key Scenarios

## Processing reports on the server

## 7 Post-conditions

## Successful Completion

The system stores the report

## Failure Condition

The report hasn’t been made since there was an error

## 8 Special Requirements

None

## BR4

## 1 Brief Description

This use case describes how a user can download report about vehicle maintenance   
in this usecase userstories 1-5 of business requirement 4 are put together.

## 2 Actors

## User

## System

## 3 Preconditions

There is an active connection between the user and the server.

## 4 Basic Flow of Events

1. The use case begins when the user requests a report concerning vehicle-maintenance.

2. The system aggregates the requested data

3. The server starts processing to make a report

4. The system completes the processing

5. The system makes the report available for download for the user.

6. The browser of the user starts downloading the report

7. The system shows a progress bar to see the download progress

8. When the download is completed the user gets an confirmation

## 5 Alternative Flows

## The user quits the program

If in step 6 of the basic flow the user quits the program, then the use case ends with a failure condition. The system stops the download and disconnects the connection to the user.

## 6 Key Scenarios

## Downloading the report about vehicle maintenance

## 7 Post-conditions

## Successful Completion

The user gets the report

## Failure Condition

The report hasn’t been downloaded since there was an error

## 8 Special Requirements

None

## BR5

## 1 Brief Description

This use case describes how a vehicle office gets an alert when the car needs repair   
In this usecase userstory 1, 2 and 4 of business requirement 5 are put together.

## 2 Actors

## 2.1 User

## 2.2 System

## 3 Preconditions

There is an active internet connection between the system and the user

## 4 Basic Flow of Events

1. The use case begins when the system creates a new alert in the vehicle maintenance dashboard

2. The user receives an alert icon in each web interface

3. The user can see basic information about the alert

4. The user forwards the alert to his superiors or co-workers

## 5 Alternative Flows

4 The user decides that the alert is not important enough to forward and deletes it.

## 6 Key Scenarios

## 6.1 Alerting the user

## 7 Post-conditions

## 7.1 Successful Completion

The system alerts the user

## 7.2 Failure Condition

None

## 8 Special Requirements

None

## BR6

## 1 Brief Description

This use case describes how a user can buy city-gis meta-data   
in this usecase userstory 1- 3 of business requirement 6 are put together.

## 2 Actors

## 2.1 User

## 2.2 System

## 3 Preconditions

There is an active internet connection between the system and the user.

## 4 Basic Flow of Events

1. The use case begins when the user selects the meta-data he wants to buy

2. The user confirms his selection

3. The system provides the amount of money the user needs to pay.

4. The user pays with PayPal.

5. The system receives the purchase confirmation.

6. The system provides the data to the user for download.

7. The user downloads the data in his browser.

8. The system provides a progress bar while the data is downloading.

9. The user gets a confirmation when the download is completed.

## 5 Alternative Flows

## 5.1 The user declines payment

If in step 4 of the basic flow the user quits the program, then

1. The use case ends with a failure condition. The system shows an error message that the user didn’t pay and the system gives the option to retry the purchase.

The user quits the program while downloading If in step 7 of the basic flow the user quits the program, then The use case ends with a failure condition. The system stops the downloading and disconnects the connection to the user. But it keeps the payment information of the user, so he can later download again without going through the whole payment process again.

## 6 Key Scenarios

## 6.1 Selling data to the user

## 7 Post-conditions

## 7.1 Successful Completion

The Data is sold to the user, and the user receives the data as a download.

## 7.2 Failure Condition

The purchase is cancelled

## 8 Special Requirements

None

# Exercise C.

## Pro’s

### Project Group

Functional- and nonfunctional requirements   
For the project group functional and nonfunctional requirements usually come in handy. They are easy to read, it is just a simple sentence which can be easily translated into a work item for the project. The list of requirements can also be used in a contract with the customer to cover yourself legally. Using functional- and nonfunctional requirements created a nice agreement of which functionalities will be implemented into the system. When the system is built, you can easily verify if all requirements are done by using the list of requirements.

#### User stories, use cases and use case descriptions

Using this method gives the developer more detailed information about WHY the customer wants certain features built, not just what he wants built. Using User Stories gives the developers a much deeper insight into the actual needs and goals of the customer over Functional requirements. Another advantage is that when the stories are written together with the client/user, the developers have to make less assumptions which will result in less conflicts/problems with the customer.

### Client

Functional- and nonfunctional requirements   
Functional- and nonfunctional requirements can be easily formed. The client can say exactly what he wants. He can also give constraints and restrictions to some requirements by using nonfunctional requirements. These can be very useful. For example; if the client wants a specific system, he can write down exactly what kind of functionalities he wants in the system. And by using the nonfunctional requirements he can say that the system must work on Ubuntu server, instead of Windows Server.

#### user stories, use cases and use case descriptions

By using user stories etc., a client can express his desires for a system in normal language, without having any technical knowledge of his own. The true power of user stories lies in allowing the client to express why he wants to have certain features. Because of this ability, the client can make sure that the system that uses his user stories is built exactly for the things he wants it for, and nothing else.

## Con’s

### Project Group

functional- and nonfunctional requirements   
Functional and nonfunctional requirements can be very vague. If they are not properly formed, they can be very hard to understand, or it may be very easy to make a misunderstanding. As a result, the project group have to either make assumptions or ask the client for more detailed information. This may slow down the development process, which may result in additional costs/time.

#### user stories, use cases and use case descriptions .

Using this method, nonfunctional requirements cannot be specified. This may result in misinterpretations. In use case description you may not find on what operating system the application should work. If for example, the developers develop the system for Linux, but it should operate on Windows, this will be a big problem.

### Client

#### functional- and nonfunctional requirements.

Functional and nonfunctional requirements are easy to write down, but can be easily misunderstood by the developers. This may result in wrong functionalities in the system, which may lead to additional costs. Functional requirements also do not detail the clients’ wishes and motives in the same powerful way as user stories do.

#### User stories, use cases and use case descriptions

User stories are great add specifying a customers’ wish in a simple format, and illuminating his motives for the wish. What they are not so great at is providing exact information about system functionality: User stories are relatively short and are written from the perspective of the user, leaving them open to misinterpretations; the developer might see a userstory as talking about functionality X, even though the client really wants functionality Y. This ambiguity may lead to the wrong functionality being built, which will cost both time and money to fix.