Deliverable week 3.

This file contains the first versions of our use cases and the user stories, furthermore will this file contain a list of who did what.

First you will find the user story for an actor, then the use cases for that same actor.

User Story	Repairing a truck		
Code	US-BS-1.1		
Package	Business Support		
File	US-BS.docx		
Story	A truck breaks down and a new part has to be delivered to repair this truck. The department of business support is responsible for this.		
Refined by	UC 1.1		
Version	1.0	Herm Lecluse	

User Story	Saving Files		
Code	US-BS-1.2		
Package	Business Support		
File	US-BS.docx		
Story	When an order is completed and all the information is collected, those data should be stored in a proper way		
Refined by	UC 1.2		
Version	1.0	Herm Lecluse	

Use case	Order new Truck part		
Code	UC-BS-1.1		
Package	Business Support		
File	UC-BS-1.1.docx		
Actor	Staff employee		
Description	A truck has a broken part which has to be replaced as soon as possible		
Requirements	- The truck must be at the HQ - Access to the system & internet - Knowledge of the problem		
Scenario	1. Message from another department comes in via mail 2. System shows information about what part(s) is/are broken. 3. Employee orders a new truck part via the internet. 4. Employee adds to the system that a part is being delivered 5. System returns a status message: “Part is on its way”. 6. Employee logs out		
Exceptions	**3.1 Part is not available at the moment, reparation will be delayed** 3.1.1 Employee adds to an field in the system :”part was out of stock”. Use case ends here.		
Extensions	The truck mechanic is delivered the required parts. So he can attach them		
Result	The truck has been repaired and can be used again for new transports		
Version	1.0	Herm Lecluse	

Use case	Archive incoming paper work		
Code	UC-BS-1.2		
Package	Business Support		
File	UC-BS.docx		
Actor	Staff employee		
Description	Information of an certain order/invoice/etc. should be stored		
Requirements	- Access to the system & internet		
Scenario	1. The employee gets an email which contains information about the deliveries of a certain period. 2. Employee logs into the system 3. Employee enters the information to the system 4. System will store this in a database 5. Employee saves the changes 6. Employee logs off.		
Exceptions	**4.1 Database gives errors about inconsistent data.** 4.1.1 Employee checks his input for mistakes.		
Extensions			
Result	The information is stored in a good and efficient way.		
Version	1.0	Herm Lecluse	

Use case	Arranging a new truck		
Code	UC-BS-1.3		
Package	Business Support		
File	UC-BS.docx		
Actor	Staff employee		
Description	When a truck can't be repaired anymore		
Requirements	- Access to the system & internet		
Scenario	1. The employee gets an email which contains information about the deliveries of a certain period. 2. Employee logs into the system 3. Employee enters the information to the system 4. System will store this in a database 5. Employee saves the changes 6. Employee logs off.		
Exceptions	**4.1 Database gives errors about inconsistent data.** 4.1.1 employee checks his input for mistakes.		
Extensions			
Result	The information is stored in a good and efficient way.		
Version	1.0	Herm Lecluse	

User Story	CEO is prompted	
Code	US-ceo-2.0	
Package	CEO	
File	US-ceo-2.0.docx	
Story	CEO will be prompted in case of delays or issues in order to be up to date of any issues and then can act accordingly.	
Refined by	US-ceo-2.1 (prompt for issues)	
Version	1.1	Bas de Weerd

User Story	CEO requests overviews	
Code	US-ceo-1.0	
Package	CEO	
File	US-ceo-1.0.docx	
Story	CEO can request an overview of driving schedules, employees, orders, financial situation, issues or customer relations at any given time. This will give the CEO an ability to see how his business is doing whenever he wants.	
Refined by	UC-ceo-1.1 (request financial report) UC-ceo-1.2 (request issues overview)	
Version	1.1	Bas de Weerd

Use case	Request issues overview	
Code	US-ceo-1.2	
Package	CEO	
File	UC-ceo-1.2.docx	
Actor	Chief Executive Officer (CEO)	
Description	CEO requests an overview of all issues	
Requirements	- Data is up to date - Working PC and working software connected to the database through internet connection - User is logged in	
Scenario	1. User goes to the issues section 2. User views a complete overview of all issues with dates and descriptions 3. Users selects issue(s) he wants to address and selects them for further details 4. CEO addresses the issue	
Exceptions	Not applicable	
Result	User is able to view details of issues	
Version	1.1	Bas de Weerd

Use case	Request financial report	
Code	UC-ceo-1.1	
Package	CEO	
File	UC-ceo-1.1.docx	
Actor	Chief Executive Officer (CEO)	
Description	CEO requests a financial report	
Requirements	- Data is up to date - Working PC and working software connected to the database though internet connection - User is logged in	
Scenario	1. User goes to the finance section 2. User selects time period 3. User submits 4. User receives financial report	
Exceptions	**2.1 User selects invalid time period.** 1. System responds with corresponding error message. 2. Use case proceeds at step 2.	
Result	User has a financial report of a certain time period	
Version	1.1	Bas de Weerd

Use case	Prompt for issues	
Code	UC-ceo-2.1	
Package	CEO	
File	UC-ceo-2.1.docx	
Actor	Chief Executive Officer (CEO)	
Description	CEO will be prompted in case there are any issues or delays.	
Requirements	- Other user submits top priority issue - Working PC and working software connected to the database through internet connection - User is logged in	
Scenario	1. User gets a pop up containing short information about issue 2. User can view issues section for more details 3. User closes pop up 4. User acts accordingly with a solution to the problem	
Exceptions	**1.1 User is away from keyboard and thus cannot respond.** 1. User sees message when he/she returns	
Result	If necessary a solution is applied, CEO is aware of issue instantly	
Version	1.1	Bas de Weerd

User Story	Mailing and then viewing an order		
Code	US-cu-1.0		
Package	customer		
File	US-cu-1.0.docx		
Story	The user thinks up what he wants to order and writes this in an email along with his personal information. The user will email their order to the orders and invoices department of the company. The order will be processed there. The user will receive an order code. The user will enter this order code in the system. Now the user can see the status of their order.		
Refined by	UC-cu-1.1 (Checking Order)		
	UC-cu-1.3 (Mailing Order)		
Version	1.0	Jurian Janssen	

Use case	Checking Order status		
Code	UC-cu-1.1		
Package	customer		
File	UC-cu-1.1.docx		
Actor	Customer		
Description	Checking the status of a placed order by a customer		
Requirements	- Working PC and working Software - Unoccupied and working phone.		
Scenario	1. User goes to the application 2. User clicks the “Check order” button 3. User fills in the order code in the text field 4. User clicks “Confirm” 5. Order details will be displayed to the user		
Exceptions	**Order number is incorrect** 1. Customer will be asked to fill in their number again 2. If correct the customer will now see their order		
Result	The user can view the status of their order		
Version	1.0	Author	Jurian Janssen

Use case	Faxing an order		
Code	UC-cu-1.2		
Package	customer		
File	UC-cu-1.2.docx		
Actor	Customer, Orders and Invoices		
Description	Faxing an order to the orders and invoices department		
Requirements	- Working PC and working Software - A fax.		
Scenario	1. User creates a fax of their soon to be order 2. User sends their order to the orders and invoices department by fax 3. Order is processed via fax 4. Order is received by Orders and Invoices 5. Orders and invoices will add the order (see creating order use case)		
Exceptions	**Fax document is unclear** 1. Orders and invoices will contact the customer and asks them to send it again.		
Result	Order will be processed by Orders and Invoices		
Version	1.0	Author	Jurian Janssen

Use case	Mailing an order		
Code	UC-cu-1.3		
Package	customer		
File	UC-cu-1.3.docx		
Actor	Customer, Orders and Invoices		
Description	Mailing an order to the orders and invoices department		
Requirements	- Working PC and working Software - An email client.		
Scenario	1. User creates a mail of their soon to be order 2. User sends their order to the orders and invoices department by mail 3. Order is processed via mail 4. Order is received by Orders and Invoices 5. Orders and invoices will add the order (see creating order user case)		
Exceptions	**Mail document is unclear** 3. Orders and invoices will contact the customer and asks them to send it again.		
Result	Order will be processed by Orders and Invoices		
Version	1.0	Author	Jurian Janssen

Use case	Ordering by phone		
Code	UC-cu-1.4		
Package	customer		
File	UC-cu-1.4.docx		
Actor	Customer, Orders and Invoices		
Description	Phoning an order to the orders and invoices department		
Requirements	- Working PC and working Software - A phone.		
Scenario	1. User writes down what he wants to order 2. User calls and tells their order to the orders and invoices department 3. Order is processed via phone 4. Order is received by Orders and Invoices 5. Orders and invoices will add the order (see creating order usercase)		
Exceptions			
Result	Order will be processed by Orders and Invoices		
Version	1.0	Author	Jurian Janssen

User Story	Staff of the order and invoice department takes orders and creates reciepts		
Code	US-oi-1.0		
Package	order-invoice		
File	US-oi-1.0.docx		
Story	Staff of the order and invoice department takes orders by phone or fax and writes invoices.		
Refined by	UC-oi-1.1 (Take order by phone)		
	UC-oi-1.2 (Take order by fax)		
	UC-oi-1.3 (Print an invoice)		
	UC-oi-1.4 (Take order by email)		
Version	1.1	Schahab Kaiumi	

Use case	Take order by phone		
Code	UC-oi-1.1		
Package	order-invoice		
File	UC-oi-1.1.docx		
Actor	Order and invoice department		
Description	The staff of the order and invoice department takes an incoming phone order.		
Requirements	- Working PC and working Software connected to the database - Unoccupied and working phone.		
Scenario	1. The staff accepts the incomming phone call. 2. System gets all all information. (name of the company, adress, contact information and sales tax ID number, how many tons, where and when it has to be load and unload, what kind of liquid) 3. System checks if the filled information is complete. 4. System checks if the customer has payed all his invoices yet. 5. System adds the current date and a new invoice number will be inserted automatically. 6. System gives an internal number to each task and subtask that has to be done by the truck driver. (load/unload) 7. System checks the availability of trucks and drivers. 8. The staff accepts the order. 9. End of the conversation.		
Exceptions	**3.1 The typed information is incomplete.** 1. The missing information has to be asked by the staff. 2. The missing information has to be filled in. 3. Use case proceeds at step 4.		
Exceptions	**4.1 The customer has not payed his debts yet.** 1. Use case ends here. 2. A report about the customers call will be send to the finance department.		
Result	All needed information are collected and the job can be done. Now they can be used for an invoice that has to be created next.		
Version	1.1	Schahab Kaiumi	

Use case	Take order by fax		
Code	UC-oi-1.2		
Package	order-invoice		
File	UC-oi-1.2.docx		
Actor	Order and invoice department		
Description	The staff of the order and invoice department takes incoming fax order.		
Requirements	- Working PC and working Software connected to the database. - Unoccupied and working fax device.		
Scenario	1. A fax arrives and gets printed out. 2. System gets all information typed in. (name of the company, adress, contact information and sales tax ID number, how many tons, where and when it has to be load and unload, what kind of liquid) 3. System checks if the filled information is complete. 4. System checks if the customer has payed all his invoices yet. 5. System adds the current date and a new invoice number will be inserted automatically. 6. System gives an internal number to each task and subtask that has to be done by the truck driver. (load/unload) 7. System checks the availability of trucks and drivers. 8. System sends feedback to the customer and confirms the order.		
Exceptions	**3.1 The typed information is incomplete.** 1. The customer has to be called by the staff to get those missing information. 2. The staff gets the need information. 3. Use case proceeds at step 4.		
Exceptions	**4.1 The customer has not payed his debts yet.** 1. Use case ends here. 2. A report about the customers call will be send to the finance department.		
Result	All needed information are collected and the job can be done. Now they can be used for an invoice that has to be created next.		
Version	1.1	Schahab Kaiumi	

Use case	Print an invoice		
Code	UC-oi-1.3		
Package	order-invoice		
File	UC-oi-1.3.docx		
Actor	Order and invoice department		
Description	The staff of the order and invoice department calculates the price and creates an ivoice.		
Requirements	- Working PC - Working Software connected to the database. - Working Printer		
Scenario	1. System accepts the order and saves it in the internal DBS. 2. System determines the deadline for the payment automatically. (14 days) 3. System prints the invoice.		
Exceptions	**2.1 The determined day of payment is a nonbusiness day** 1. The System automatically takes the next possible date. 2. Use case proceeds at step 3.		
Exceptions	**3.1 The printer does not work.** 1. The System reminds the staff of the invoice department to print daily. 2. If the print process was successful, then this alert will not appear no more. 3. This use case ends here.		
Result	All needed information about the order and the invoice are saved in the databse and the invoice is printed out.		
Version	1.1	Schahab Kaiumi	

Use case	Take order by email		
Code	UC-oi-1.4		
Package	order-invoice		
File	UC-oi-1.4.docx		
Actor	Order and invoice department		
Description	The staff of the order and invoice department takes an incoming email order.		
Requirements	- Working PC and working Software connected to the database.		
Scenario	1. The staff gets a new email notification. 2. System gets all all information typed in. (name of the company, adress, contact information and sales tax ID number, how many tons, where and when it has to be load and unload, what kind of liquid) 3. System checks if the filled information is complete. 4. System checks if the customer has payed all his invoices yet. 5. System adds the current date and a new invoice number will be inserted automatically. 6. System gives an internal number to each task and subtask that has to be done by the truck driver. (load/unload) 7. System checks the availability of trucks and drivers. 8. The staff accepts the order. 9. End of the conversation.		
Exceptions	**3.1 The typed information is incomplete.** 1. The missing information has to be asked by the staff. 2. The missing information has to be filled in. 3. Use case proceeds at step 4.		
Exceptions	**4.1 The customer has not payed his debts yet.** 1. Use case ends here. 2. A report about the customers call will be send to the finance department.		
Result	All needed information are collected and the job can be done. Now they can be used for an invoice that has to be created next.		
Version	1.1	Schahab Kaiumi	

User Story	(re-)Scheduling orders		
Code	US-PL-1.1		
Package	Planning		
File	US-BS.docx		
Story	The planner, Creates a schedule, get details from the customer, CEO and truck drivers, creates an individual plan for every day. Later he gives the information or the plan to the financial department and personal department.		
Refined by	UC 1.1		
Version	1.0	Herm Lecluse	

<table>
<tr><td>Use case</td><td colspan="3">Scheduling an order</td></tr>
<tr><td>Code</td><td colspan="3">UC-PL-1.1</td></tr>
<tr><td>Package</td><td colspan="3">Planning</td></tr>
<tr><td>File</td><td colspan="3">UC-PL.docx</td></tr>
<tr><td>Actor</td><td colspan="3">Planner</td></tr>
<tr><td>Description</td><td colspan="3">Planner creates an schedule for an order</td></tr>
<tr><td>Requirements</td><td colspan="3">Planner needs information about who is allowed to transport hazardous fluids
The order that has to be assigned is communicated to Transport department.</td></tr>
<tr><td>Scenario</td><td colspan="3">1. Order comes from order department
2. Planner logs in to system
3. Planner checks if the fluid is hazardous
4. Planner looks up in the system what trailers, trucks and drivers are ready to be scheduled.
5. Planner assigns trailer truck and driver to order
6. Planner saves these fields
7. System returns a message to the driver which contains his order
8. Planner logs out</td></tr>
<tr><td>Exceptions</td><td colspan="3"><b>3.1 order contains a hazardous fluid and the driver is not allowed to transport
this order</b>
3.1.1 System returns error message
3.1.2 Planner assigns order to a certified driver
3.1.3 Use case continuous at step 4.
<b>3.2 Driver is already booked</b>
3.2.1 System returns error message.
3.2.2 Planner assigns new driver to the order
3.2.3 Use case continuous at step 4.
<b>3.3 Order is already scheduled</b>
3.3.1 System returns error message
3.3.2 Planner realizes he already planned this order
3.3.3 Use case ends here.</td></tr>
<tr><td>Extensions</td><td colspan="3"></td></tr>
<tr><td>Result</td><td colspan="3">The order that came in from the Order and invoices department is scheduled for a driver.</td></tr>
<tr><td>Version</td><td>1.0</td><td>Herm Lecluse</td><td></td></tr>
</table>

Use case	Change an schedule		
Code	UC-PL-1.2		
Package	Planning		
File	UC-PL.docx		
Actor	Planner		
Description	Planner needs to change something in an schedule he made earlier		
Requirements	Planner must have access to the system Planner has a reason to reschedule an order or change a schedule.		
Scenario	1. Planner logs in to system 2. Planner opens schedule for an order he previously planned 3. Planner applies changes 4. Planner saves the changes 5. System updates the schedule of the drivers (where needed). 6. Planner logs out.		
Exceptions	**4.1 The new truck/driver/trailer is already occupied** 4.1.1 System returns error message 4.1.2 Planner assigns another truck/driver/trailer to the order 4.1.3 Use case continuous at step 5.		
Extensions	Another change has to be scheduled repeat same use case		
Result	The schedule has been changed and the problems should be resolved. Truck drivers are informed as well.		
Version	1.0	Herm Lecluse	

User Story	Finding your schedule		
Code	US-td-1.0		
Package	Truckdriver		
File	US-td-1.0.docx		
Story	The user starts up their device and starts the application. They fill in their employee number and their password. If these are entered correctly the user will see the homescreen of the application. The user then clicks on the schedule tab. A new screen will appear with several options. the user selects "Personal schedule". His or her personal schedule will be shown on the screen.		
Refined by	UC-td-1.1 (Checking Schedule)		
	UC-td-1.3 (Logging in)		
Version	1.0	Jurian Janssen	

Use case	Checking schedule		
Code	UC-td-1.1		
Package	Truckdriver		
File	UC-td-1.1.docx		
Actor	Truck Driver		
Description	Checking the schedule for the day		
Requirements	- User must have a machine that runs the software - User must be connected to the database server - User must be logged in		
Scenario	1. User clicks the "Schedules" tab 2. User clicks on the "Personal Schedule" 3. The user views his personal schedules		
Exceptions			
Result	User can view his or her schedule for a certain timeperiod		
Version	1.0	Author	Jurian Janssen

Use case	Confirming Delivery		
Code	UC-td-1.2		
Package	Truckdriver, customer		
File	UC-td-1.2.docx		
Actor	Truck Driver		
Description	Confirming the delivery was made		
Requirements	- User must have a machine that runs the software - User must be connected to the database server - User must be logged in		
Scenario	1. User clicks on the "Deliveries" tab 2. User selects the delivery he wants to fill out 3. User fills out the different fields 4. User lets the customer sign in a digital field 5. User clicks on the "Save" button		
Exceptions			
Result	Forms are filled in and saved on the server. The order will be set as “Delivered”		
Version	1.0	Author	Jurian Janssen

<table>
<tr><td>Use case</td><td colspan="3">Logging in</td></tr>
<tr><td>Code</td><td colspan="3">UC-td-1.3</td></tr>
<tr><td>Package</td><td colspan="3">Truckdriver</td></tr>
<tr><td>File</td><td colspan="3">UC-td-1.3.docx</td></tr>
<tr><td>Actor</td><td colspan="3">Truck Driver, CEO, Planner, Business Support, Orders and Invoices</td></tr>
<tr><td>Description</td><td colspan="3">Logging in to the system in order to work for the company</td></tr>
<tr><td>Requirements</td><td colspan="3">- User must have a machine that runs the software
- User must be connected to the database server</td></tr>
<tr><td>Scenario</td><td colspan="3">1. User starts the software on his or her device
2. User fills in his or her employee-number
3. User fills in his or her password
4. User clicks the loginbutton
5. The user us redirected to the homepage</td></tr>
<tr><td>Exceptions</td><td colspan="3"><b>Incorrect password or employee number</b>
The user will be asked to re-enter their information</td></tr>
<tr><td>Result</td><td colspan="3">The user enters the system and is redirected to the homepage of the software.</td></tr>
<tr><td>Version</td><td>1.0</td><td>Author</td><td>Jurian Janssen</td></tr>
</table>

List of work done during week 3:

- Herm Lecluse: Second interview with CEO van der heijden, Use cases & stories (Business support and planning) and creation of this document.
- Bas de Weerd: Use cases & stories (CEO), fixed Nithilan Kanesamoorthy’s failed attempt to create a document of the interview with the planning expert. And the Use-case-model (alpha)
- Schahab Kaiumi: Use cases & stories (Orders&invoices), Use-Case-Model(Alpha).
- Jurian Janssen: Use cases & stories (Truck driver & customer), fixed some look and feel issues of some use cases.
- Nithilan Kanesamoorthy : Created not acceptable use cases and user story’s, have been re-done by someone else. Made a start on a Dictionary.
- Chris Backus: Didn’t attend, did not delivered what he agreed on delivering.