**Pound Lock**

Use case definition

Read the user requirements and answer the following questions:

• What is the primary function of the system

Een sluis zorgt ervoor dat boten en andere watervoertuigen van waterstand kunnen wisselen tussen 2 delen van een rivier of kanaal.

• What are the secondary functions of the system

Stoplicht en nood functie

• Why is this system being built? What is it replacing and why?

Dit systeem zorgt ervoor dat boten van water niveau kunnen wisselen.

• What are the actors in this system?

Sluis-beheerder.

Construct the use case(s) for the primary function(s), taking in account:

• The role the external objects (actors) and system play in each scenario

• The interaction (flows) necessary to complete the scenarios in the use case(s)

• The sequence of events and data to realise the scenario in the use case(s)

• What variations on the scenario are possible?

|  |  |
| --- | --- |
| **Naam** | Toegang in/uit sluis |
| **Samenvatting** | De actor geeft een signaal aan de boot om de sluis in/uit te varen. |
| **Actor** | Sluis-beheerder |
| **Aanname** | Het waterpijl binnen de sluis is evenhoog als het waterpijl bij de boot en de sluis staat open aan de boot kant. |
| **Beschrijving** | Trigger) De actor drukt de **Allowed Sailing** in knop in. 1) Stoplichten aan boot kant gaan van rood naar groen  2) De stoplichten in de sluis gaan van groen naar rood licht.  Voor het uitvaren gebeurd dit process andersom |
| **Uitzondering** | Bij een alarm is er geen toegang tot de sluis. |
| **Resultaat** | De boot vaart in/uit de sluis |

|  |  |
| --- | --- |
| **Naam** | Waterpeil vananderen in sluis |
| **Samenvatting** | Het waterpeil wordt veranderdt in de sluis. |
| **Actor** | Sluis-beheerder |
| **Aanname** | Het waterpeil kan veranderd worden. |
| **Beschrijving** | Trigger) De actor drukt de **start** knop in.   1. De entry kant van de sluis gaat dicht. 2. De valves gaan open op het water erin of eruit laten. 3. De exit kant gaat open. |
| **Uitzondering** |  |
| **Resultaat** | Het waterpeil is veranderdt. |

|  |  |
| --- | --- |
| **Naam** | Alarm aflaten gaan |
| **Samenvatting** | De sluis gaat in alarm modes. |
| **Actor** | Sluis-beheerder |
| **Aanname** | - |
| **Beschrijving** | Trigger) De actor drukt de **alarm** knop in.  -) Als de deuren beide dicht zijn en valves open staan gaan de valves meteen dicht.  -) Als een van de deuren aan het openen of dicht aan het gaan is.  -) Start knop wordt geblokkeerd |
| **Uitzondering** | - |
| **Resultaat** | Sluis gaat in alarm mode. |

|  |  |
| --- | --- |
| **Naam** | System hervatten |
| **Samenvatting** | Het systeem wordt hervat na een alarm. |
| **Actor** | Sluis-beheerder |
| **Aanname** | Er moet een alarm aanwezig zijn. |
| **Beschrijving** | Trigger) De actor drukt de **alarm-recovery** knop in.  -) Sluis gaat terug naar normale (vorige) staat. |
| **Uitzondering** | - |
| **Resultaat** | Sluis gaat terug naar normale staat. |

Draw a use case diagram using the constructed use case(s).

Diagram

Description automatically generated

Working towards a class diagram

Please take the following steps:  
2. Association discovery: draw an object diagram which lists the objects and their  
associations, use sequence diagrams where appropriate to find associations  
3. Use the objects to define the classes you need, draw a first class diagram and define the  
responsibility of each class (usually defined in a table)  
4. Complete your first class diagram from step 3. You don’t need to use the appropriate  
relationships yet, however if you use inheritance you must indicate that correctly.  
5. Define the required operations for each class.  
6. Use sequence diagrams to verify the interactions between classes. Important: do not try to  
be complete in your sequence diagram: make a sequence diagram for a specific situation.  
E.g.: sequence diagram for trying to send a file in which the server does not accept the file.

Dit is het begin van mijn klasse diagram.

