

# PKSU - Props 2.0

Louise Knudsen, Helena Bach, David Pedersen

25. marts 2014

## 1 Problem definition

*The Royal Danish Theatre's props department uses a locally installed (??) database-system to search through all the props and productions corresponding OPSTILLINGS- OG KØRE-LISTER. The system was developed more than 20 years ago by their own Claus Nepper Fakkenberg. Claus alone holds the responsibility for maintaining the system, as he is the only one who understands the source-code, which will soon be a problem as he is now retiring. Due to the age of the system, some functionalities are no longer needed, and there are new requirements that are not implemented, most importantly a need to use the system outside of the office.*

The problem can be described with the following points:

- The current database system was developed in 1989, and lacks a lot of wanted functionalities.
- The developer, Claus, is the only one who can modify and support the system.
- Claus is retiring.

To analyze the problems defined above and develop a system definition we use the FACTOR Criterion.

### 1.1 The FACTOR Criterion

**F:** (functionality) Keeping track of the props and performances, and help in running these. As well as support the administration.

**A:** (application domain) The Royal Danish Theater

**C:** (conditions) The system should be usable by multiple user at a time, wherever there are access to the internet.

**T:** (technology) The system will be developed on standard laptops, and should be usable on standard PC's and tablets and supported by every OS.

**O:** (object) Props, pictures and employees in the props department.

**R:** (responsibility) Searching and administrative tool.

## 1.2 System definition

A web-based database-system of Det Kongelige Teaters props department. The system should primarily be a searching tool used to prepare and run performances and search for props in current and previous productions by the assistant stage managers, and secondly be used for budget monitoring, search for supplier information as well as keeping track of all performance information, by the chairman of the props department. The system should be usable for people with greatly variable computer experience.

## 1.3 Requirements

We have had meetings with different people and have some sort of idea of the requirements but only april 9th these will be formalised more specifically.

Acceptance criteria - A minimum of requirements that our system should meet.

## 1.4 Constraints

- Login, they handle that somehow...
- Should be able to talk with a media database that they have. We make sure they can talk to each other.

## 1.5 Application domain

The whole theatre.

## 1.6 Solution domain

The props department.

Cumulus, their media database. Only a limited part of this will be interfacing with our application though.

Most likely some sort of mirror of their data.

## 1.7 Deliverables

- PHP code
- MySQL database
- Instructions in the form of meetings of papers explaining the system.
- Probably also some support during the initial start up phase.

## 2 Initial Software Project Management Plan

### 2.1 Overview of the project

#### 2.1.1 Project summary

##### Work product

- MySQL database
- PHP code for the web-interface.
- Assignments describing the work process

##### Schedule

On April 9th we are scheduled to meet with Martin and Mikkel from the theatre, to finalize the requirements elicitation and sign the project agreement. We will thereafter be able to put together a more detailed time-schedule.

On June 23rd we have our final deadline, and will hopefully deliver the final product to the theatre.

##### Participants

Developers: David Pedersen, Helena Bach, Louise Knudsen

Project managers: David Pedersen, Helena Bach, Louise Knudsen

Client: The Royal Danish Theatre.

##### Tasks

noget med opgaver WORK BREAKDOWN STRUCTURE

#### 2.1.2 Evolution of the plan

All changes to the project management plan must be agreed to by all participants before they are implemented. All changes should be documented in the, to the developers, shared log, and in time become a part of the final SPMP.

### 2.2 References

All artifacts will conform to the theatres standards.

noget med PHP og MySQL??

### 2.3 Definitions

The Royal Danish Theatre uses a lot of different terms in their work, and some of these should also be used in the future database system. The terms we have been presented to so far are listed with a short description below:

Every production has a unique 8-digit number, containing of 4 random digits followed by a dash and premiere-year for this particular production. (There can be multiple productions of, for instance the same play, which then have different premiere-years)

If a production is *SKILT*, it means that it has been discarded and the props are back in the storage room.

If a production is *I REPERTOIRE*, it is being played in the current season and the props are in use.

If the production is *I CONTAINER* or similar, the production is not yet *SKILT* nor is it currently being played, and the props are therefore occupied but not in use.

## **2.4 Project Organization**

### **2.4.1 External interfaces**

The props database and appertaining web-interface will be managed and developed by David Pedersen, Helena Bach and Louise Knudsen.

The photographs of the props and productions, along with the login system and server connection will be handled by the XX, Martin Thaarup Larsen.

All managers will be in continuous contact with the chairman of the props department, Mikkel Rasmus Theut, as well as Martin Thaarup Larsen.

### **2.4.2 Internal structure**

Skal vi have den her med? det samme som partisipants?

### **2.4.3 Roles and responsibilities**

Noget med opdeling af arbejdet. noget med skill matrix: David 1 i web, Helena og Louise 1 i DB?? Alle vil være indover alle opgaver, men David har ansvaret for web, Helena og Louise for DB?????

## **2.5 Managerial process plans**

### **2.5.1 Start-up plan**

The total developments time is estimated to be 15 week, counting from 13/03/2014, where first meeting with the client were held.

Noget med roller i løbet af processen.

All necessary hardware are available, as well are the software since MySQL and PHP both are open-source.

### **2.5.2 Work plan**

WEEK 1-5:

- Initial meeting with the chairman of the props department.
- More technically oriented meeting with the XX and the chairman of the props department.

- Meeting with head of the furniture subdivision. Prepare the requirements elicitation (kravs spek.???) and the Project Agreement.
- Meeting with he XX and the chairman of the props department - signing of the requirements elicitation (kravs spek.???) and the Project Agreement, and project kick-off!

WEEK 6-7:

- Focus on making the SQL-script.

WEEK 8-11:

- Finish SQL.
- Start design of the web-application.
- Begin PHP-coding and testing.

WEEK 12-15:

- Finish PHP-coding and testing.
- Final documentation.

### **2.5.3 Control plan**

Noget i forhold til Roles/responsibilities, med at David har ansvaret for Web, og Helena og Louise har for DB. Vi har fælles ansvar for dokumentation. Dette sker med intern kontakt mindst 2 gange ugenligt.

### **2.5.4 Risk management plan**

The risk factors and the tracking mechanisms are as follows.

As one of the goals of the project is to achieve more functionality than the existing system, there should in the testing-process be compared results between the two.

There should in the designing-process be a great amount of communication with the client to ensure an as user-friendly interface as possible, as the system mainly will be used by people with limited computer-experience.

There is a slim chance of hardware failure, in which case each developer is responsible for there own computer, and the client holds the responsibility for the server.

Each developer is responsible for continuous testing of their assigned subsystem(s) and jointly cross-testing between these.

### **2.5.5 Closeout plan**

noget ????

## **2.6 Technical process plans**

noget med kap 15...

### **2.6.1 Process model**

### **2.6.2 Methods, tools, and techniques**

### **2.6.3 Infrastructure**

### **2.6.4 Product acceptance plan**

## **2.7 Supporting process plans**

### **2.7.1 Configuration management plan**

Git and GitHub will be used in all aspects of the project to ensure version control.

### **2.7.2 Verification and validation plan**

noget med test ift. 6'eren.

### **2.7.3 Documentation plan**

måske noget med loggen, og PHP-Doc?

### **2.7.4 Quality assurance plan**

noget???

### **2.7.5 Reviews and audits**

??

#### **2.7.6 Problem resolution plan**

#### **2.7.7 Subcontractor management plan**

#### **2.7.8 Process improvement plan**

### **2.8 Additional plans**

#### **2.8.1 Presentation**

In connection with the delivery of the final product, the system and its functionalities will be presented to the chairman of the props department.

As one of the purposes of developing the new system, is to make it possible for the XX to keep the system up to date himself, we will in addition to the presentation, be going through code with the XX (Martin).

#### **2.8.2 Support**

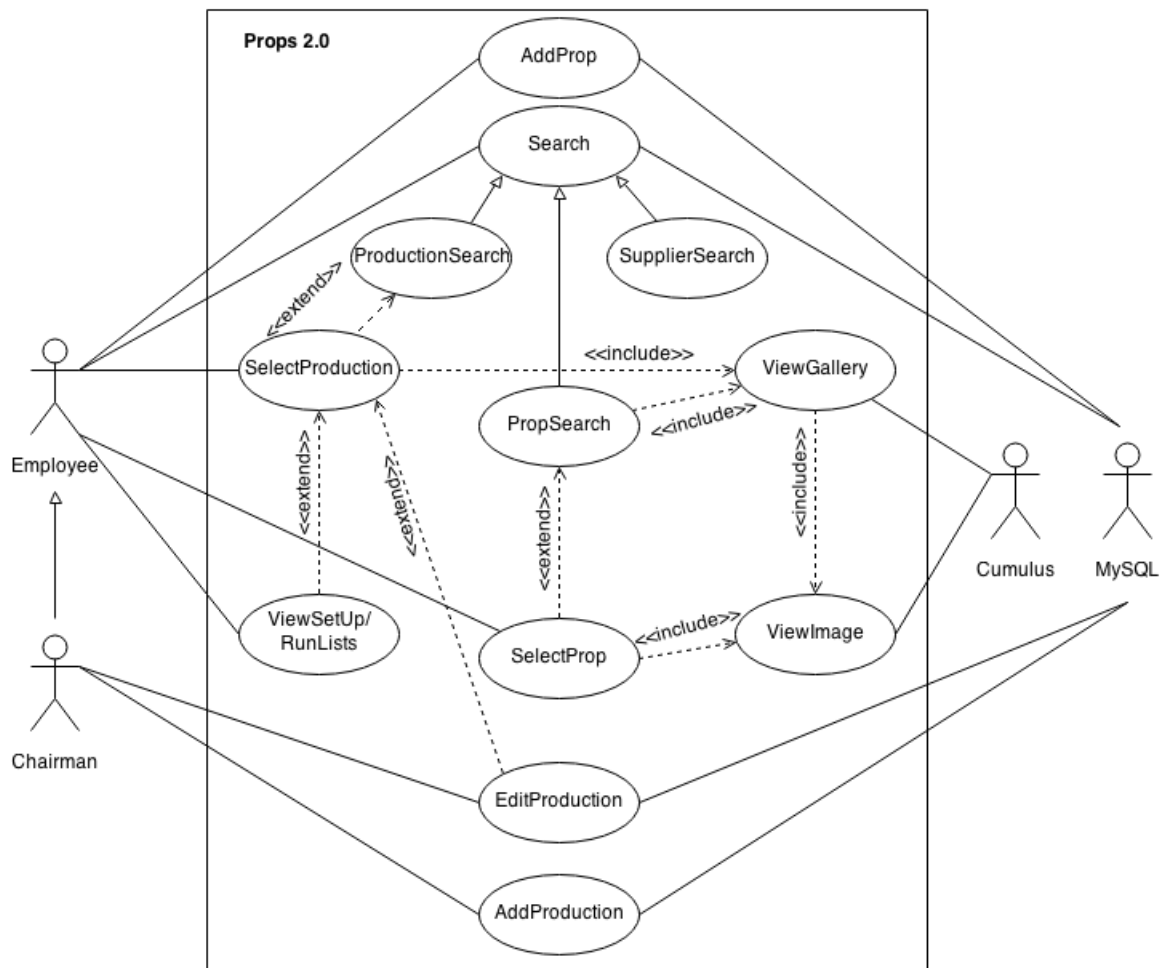
During the initialisation phase support will be offered free of charge.

## **3 Initial software architecture**

## **4 Project Agreement definition**

As mentioned in previous sections our final requirement elicitation and project agreement will be formalized on April 9th, at our meeting with the chairman of the props department and the XX. To prepare for the meeting and make an outline for this final definition we have made the following UML use case diagram (Se Appendix X for use cases) and functional/nonfunctional requirements overview.

## 4.1 UML use case diagram



## 4.2 Nonfunctional Requirements

- The system should be web-based and usable on standard PC's and tablets, and supported by every OS.
- The webapplication should be user friendly and easy to navigate. The most commonly used functionalities, namely the run- and set-up-lists should not be more than 3 clicks away.
- It should be possible for the XX (Martin) to maintain and support, and possibly expand the system after product-delivery. Therefore the system will be developed using MySQL and PHP, which is familiar to the XX.
- The system should be able to communicate with the existing media-database Cumulus.
- noget med firewall login og halløj.



- The interface should be in Danish.

### 4.3 Functional Requirements

- It should be possible to do cross search for props, productions and suppliers.
- It should be possible to add, edit and delete props, productions and suppliers.
- noget med Kongeplanen
- It should be possible to print run- and set-up-lists.
- noget med drag/drop props til run- and set-up-lists.

### 4.4 Use cases

#### 1. Tilføj Rek

<i>Use case name</i>	AddProp
<i>Participating actors</i>	Initiated by <b>Employee</b> Communicates with MySQL
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The <b>Employee</b> activates the "Add prop"function.</li> <li>2. <b>Props 2.0</b> responds by presenting a form to the <b>Employee</b>.</li> <li>3. The <b>Employee</b> fills out the form with the relevant prop-info, and submits it.</li> <li>4. <b>Props 2.0</b> receives the form, and the MySQL database gets updated. <b>Props 2.0</b> displays a confirmation of the update.</li> </ol>
<i>Entry condition</i>	The <b>Employee</b> must be logged into the Theatre's Wi-Fi
<i>Exit condition</i>	The <b>Employee</b> has received a confirmation OR An explanation indicating why the transaction could not be processed.
<i>Quality requirements</i>	Not yet defined

## 2. Tilføj forestilling

<i>Use case name</i>	<b>AddProduction</b>
<i>Participating actors</i>	Initiated by <b>Chairman</b> Communicates with MySQL
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The <b>Chairman</b> activates the "Add production"function.</li> <li>2. <b>Props 2.0</b> responds by presenting a form to the <b>Chairman</b>.</li> <li>3. The <b>Chairman</b> fills out the form with the relevant production-info, and submits it.</li> <li>4. <b>Props 2.0</b> receives the form, and the MySQL database gets updated. <b>Props 2.0</b> displays a confirmation of the update.</li> </ol>
<i>Entry condition</i>	The <b>Chairman</b> must be logged into the Theatre's Wi-Fi
<i>Exit condition</i>	The <b>Chairman</b> has received a confirmation OR An explanation indicating why the transaction could not be processed.
<i>Quality requirements</i>	Not yet defined

## 3. Søg

<i>Use case name</i>	<b>Search</b>
<i>Participating actors</i>	Initiated by <b>Employee</b> Communicates with MySQL
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The <b>Employee</b> activates the "Search"function.</li> <li>2. <b>Props 2.0</b> responds by presenting a form to the <b>Employee</b>.</li> <li>3. The <b>Employee</b> fills out the form with the search criteria, and submits it.</li> <li>4. <b>Props 2.0</b> receives the form, the MySQL database returns the matching data and <b>Props 2.0</b> displays it</li> </ol>
<i>Entry condition</i>	The <b>Employee</b> must be logged into the Theatre's Wi-Fi
<i>Exit condition</i>	The <b>Employee</b> has received a list of matching data OR A "No match"message
<i>Quality requirements</i>	Not yet defined

We have not yet fully specified the search criteria with client. We know that separate search-functions for productions, props and suppliers are wanted, but until we have the detailed criteria, we will not include these as specific use cases. 4. Vælg rek

<i>Use case name</i>	<b>SelectProp</b>
<i>Participating actors</i>	Initiated by <b>Employee</b>
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The <b>Employee</b> selects a prop from the results of a premade search</li> <li>2. <b>Props 2.0</b> responds by presenting a more detailed description of the prop to the <b>Employee</b>.</li> </ol>
<i>Entry condition</i>	This use case extends the <b>PropSearch</b> use case
<i>Exit condition</i>	The <b>Employee</b> is looking at the prop description
<i>Quality requirements</i>	This use case includes the <b>ViewImage</b> use case

## 5. Vælg forestilling:

<i>Use case name</i>	SelectProduction
<i>Participating actors</i>	Initiated by Employee
<i>Flow of events</i>	1. The Employee selects a production from the results of a premade search 2. Props 2.0 responds by presenting a more detailed description of the production to the Employee.
<i>Entry condition</i>	This use case extends the ProductionSearch use case.
<i>Exit condition</i>	The Employee is looking at the production information.
<i>Quality requirements</i>	This use case includes the ViewGallery use case.

## 6. Køre/opstil

<i>Use case name</i>	ViewSetUp/RunLists
<i>Participating actors</i>	Initiated by Employee
<i>Flow of events</i>	1. The Employee selects the list from the results of a production search 2. Props 2.0 responds by presenting the list to the Employee.
<i>Entry condition</i>	This use case extends the SelectProduction use case
<i>Exit condition</i>	The Employee is looking at the list
<i>Quality requirements</i>	Not yet defined

## 7. Redigér/slet

<i>Use case name</i>	EditProduction
<i>Participating actors</i>	Initiated by Chairman Communitates with MySQL
<i>Flow of events</i>	1. The Chairman activates the "Edit production"function 2. Props 2.0 responds by presenting a form to the Chairman 3. The Chairman fills out the form and submit the changes OR The Chairman selects the "Delete"option 4. Props 2.0 receives the form, and the MySQL database gets updated. Props 2.0 displays a update- OR delete-confirmation
<i>Entry condition</i>	This use case extends the SelectProduction use case
<i>Exit condition</i>	The Chairman has received a confirmation OR An explanation indicating why the transaction could not be processed.
<i>Quality requirements</i>	Not yet defined

8. Se Galleri:

<i>Use case name</i>	<b>ViewGallery</b>
<i>Participating actors</i>	Initiated by <b>Employee</b> Communicates with <b>Cumulus</b>
<i>Flow of events</i>	1. The <b>SelectProduction</b> or <b>PropSearch</b> use case gets evoked 2. <b>Props 2.0</b> requests <b>Cumulus</b> for the image data 3. <b>Cumulus</b> returns the matching images 4. <b>Props 2.0</b> responds by presenting the images as a gallery
<i>Entry condition</i>	The <b>Employee</b> must initiate the <b>SelectProduction</b> or <b>PropSearch</b> use case to initiate this use case
<i>Exit condition</i>	The <b>Employee</b> is looking at the gallery.
<i>Quality requirements</i>	This use case includes the <b>ViewImage</b> use case.

9. Se billede:

<i>Use case name</i>	<b>ViewImage</b>
<i>Participating actors</i>	Initiated by <b>Employee</b> Communicates with <b>Cumulus</b>
<i>Flow of events</i>	1. The <b>ViewGallery</b> or <b>SelectProp</b> use case gets evoked 2. <b>Props 2.0</b> requests <b>Cumulus</b> for the image 3. <b>Cumulus</b> returns the matching image 4. <b>Props 2.0</b> responds by presenting the image to the <b>Employee</b>
<i>Entry condition</i>	The <b>Employee</b> must initiate the <b>ViewGallery</b> or <b>SelectProp</b> use case to initiate this use case
<i>Exit condition</i>	The <b>Employee</b> is looking at the image.
<i>Quality requirements</i>	Not yet defined