CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Chvíla Ondřej** Personal ID number: **406912**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Graphics and Interaction**

Study program: **Open Informatics**

Branch of study: **Human-Computer Interaction**

II. Master's thesis details

Master's thesis title in English:

Web application for project planning of special effects and post-production company

Master's thesis title in Czech:

Webová aplikace k plánování projektů pro trikovou a postprodukční firmu (oponent je zadavatel)

Guidelines:

Design and implement HiFi prototype of the Web application (by using final ICT), which allows planning of tasks on particular projects for special effects and post-production company. The application should allow creation and later changes of the task pipeline for individual projects. Design and implement suitable visualizations, which will inform the user about the current project phase, persons responsible for particular tasks, and assigned workers. Focus on an appropriate solution of the overview-detail visualization problem. Consider also a suitable visualization of past status and an estimation of near future status of the project, which will help the user to understand better the project context and anticipate approaching problems of resource planning. Analyze suitable technologies and if they match with those used by the company, try to connect your application with company's relevant databases and application. The prototype user interface should focus on maximal flexibility and efficiency of its usage. Collect the user requirements using of qualitative user research. The whole development process (analysis, design, and implementation) should be iterative inline with the principles of User-Centered Design methodology.

Bibliography / sources:

Alan Cooper: About Face? The Essentials of Interaction Design, Wiley Publishing Inc., 2007
Bill Buxton: Sketching User Experience, Elsevier Inc., 2007
Colin Ware: Visual Thinking for Design, Elsevier Inc., 2008
Steve Krug: Don't Make Me Think, New Riders, 2014
Don Norman: The Design of Everyday Things, Basic Book, 2002
Jakob Nielsen: Designing Web Usability, New Riders, 1999

Name and workplace of master's thesis supervisor:

doc. Ing. Zdeněk Míkovec, Ph.D., Department of Computer Graphics and Interaction

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **31.10.2017** Deadline for master's thesis submission: _____

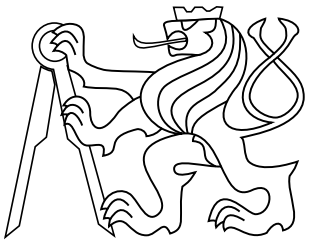
Assignment valid until:
by the end of winter semester 2018/2019

doc. Ing. Zdeněk Míkovec, Ph.D.	Head of department's signature	prof. Ing. Pavel Ripka, CSc.
Supervisor's signature		Dean's signature

master's thesis

Web application for project planning of special effects and post-production company

Bc. Ondřej Chvíla



May 2018

Advisor: doc. Ing. Zdeněk Míkovec, Ph.D.

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Computer Graphics and Interaction

Acknowledgement

I would like to express my gratitude to my advisor doc. Ing. Zdeněk Míkovec, Ph.D. for the useful comments and remarks he provided me during working of this master thesis. Furthermore, I would also like to thank all the people from UPP who participated in the user testing and those who willingly explained me the post-production process to detail. I would like to thank my family: my parents and to my sister for supporting me spiritually throughout writing this thesis and my life in general.

Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

Prague, May 22, 2018 _____

Abstract

The purpose of the thesis is to create a prototype of a web application that allows planning and visualization of tasks on particular projects for the trick and post-production company. Using a suitable visualization and user requirements that were collected using qualitative methods, a low-fidelity prototype was created, which was user tested. Based on the findings and recommendations, a high-fidelity prototype was developed using final technologies. Selection of suitable technologies has been analyzed. This prototype was then also user tested and the results of this test will be used for future development of the web application. Results of user testing showed that users need a complex system that allows great variability for individual projects.

Keywords

UCD; Prototyping; React; Post-production; VFX

Abstract

Diplomová práce si bere za cíl vytvořit prototyp webové aplikace, která umožňuje plánování a vizualizaci dílčích prací na jednotlivých projektech pro trikovou a post-produkční firmu. Pomocí vhodné vizualizace a uživatelských požadavků, které byly sesbírány pomocí kvalitativních metod byl vytvořen low-fidelity prototyp. Ten byl následně otestován přímo uživateli. Na základě nálezů a doporučení z testování byl vytvořen high-fidelity prototyp, který využíval cílové technologie. Za účelem výběru vhodných technologií byla provedena analýza. Tento prototyp byl posléze také testován s uživateli a výsledky tohoto testování budou použity pro budoucí vývoj webové aplikace. Z výsledků je zřejmé, že uživatelé potřebují komplexní systém, který umožňuje velkou variabilitu pro jednotlivé projekty.

Klíčová slova

UCD; Prototypování; React; Post-produkce; VFX

Contents

Abbreviations

List of abbreviations used in the thesis.

Hi-Fi	High Fidelity
Lo-Fi	Low Fidelity
VFX	Visual Effects
UCD	User-Centered Design
JS	JavaScript
NPM	Node Package Manager
DOM	Document Object Model
HTML	Hypertext Markup Language
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
SVG	Scalable Vector Graphics
API	Application Programming Interface
SQL	Structured Query Language
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator

1 Introduction

Post-production and visual effects in professional videos are today used in many industries, such as movies, TV commercials, and music videos. If there are more visual effects in the video, it is difficult to plan and track the entire production process, especially if the assignment of the project is constantly changing based on clients requirements. Universal Production Partners visual effects studio, which wants to expand a team and work on larger and more projects also faces this problem. The current situation in the UPP is that the supervisor oversees the entire process himself. If he supervises multiple projects, it is impossible to remember the whole process and respond well to changes. Team communication is through personal communication, an in-house telephone, or a notification application. Communication is not unified, and when the worker is reassigned to a particular task, it is necessary to reproduce all the comments that need to be implemented. The UPP analyzed the appropriate systems allowing planning and tracking projects that are on the market, but no system is suitable for them. Mainly for data security and accessibility, they do not want to be dependent on providers servers. Furthermore, systems are very complex, so their use for small projects can be inefficient. These systems would only be used for large projects and because of pricing method, it is disadvantageous for UPP to deploy the system. The UPP also uses a well-functioning notification and a booking application, and these systems cannot be connected to the already-completed ones.

1.1 Main objectives

The main objective of the diploma thesis is to come up with a new approach to plan and track projects for the post-production and visual effects company. This approach should allow to create and change individual tasks on projects, visualize their status and assign individual artists. This approach will be implemented into a web application prototype based on user requirements. The collection of the user requirements will be carried out using qualitative methods. The entire development process of the prototype implementation will be iteratively inline with the principles of user-center design methodology. It is necessary to analyze suitable technologies and visualizations. In order to find out if the design is flexible and efficient, the prototype will be evaluated on users.

2 User research

The outcome of any design effort must ultimately be judged by how successfully it meets the needs of both the product user and the organization that commissioned it. No matter how skillful and creative the designer is, if the designer does not have clear and detailed knowledge of the user's behavior and needs, the constraints of the problem, and the business or organizational goals that are driving design activities, she will have little chance to succeed.

The real insight into these topics can't be achieved by digging through the piles of numbers that come from a quantitative study like a market survey (though these can be critical for answering other kinds of questions). Rather, this kind of deep knowledge can only be achieved by qualitative research techniques. [1]

The information we want to find about users is their habits, experiences, and skills. A design that focuses on a user is called user-centered design.

2.1 User-centered design

User-centered design (hereinafter referred to as UCD) is a process and a set of techniques that can be used at creating new solutions for the world. These solutions include products, services, environments, organizations, and ways of interacting. The reason why we designate this process as "user-centered" is that it begins with people for whom solutions are proposed. The process of personalized design is based on the needs, dreams and behavior of the people we want to influence our solutions. We try to listen and understand what they want. We can designate this folder as an optimum of desirability. With this optics, we look at the world throughout the design process. As soon as we know the range of what is desirable, we begin to look at the solution with the optics of feasibility and viability. [2]

As mentioned in user research chapter there are two approaches to get user information that can be used in a complementary manner, but each provides a different kind of information. The first approach is using quantitative methods, the strength of this methods is in a large sample of people we are researching. The second approach is using qualitative methods that work with a small sample of the population, but it allows to explore problems more deeply.

2.1.1 Qualitative methods

Qualitative research methods by which they can study people in depth, understanding how they do their activities and the environmental factors that come into play. These

methods are very time consuming, so designers typically only examine small numbers of people, often numbering in the tens. [3]

The qualitative research attempts to interpret the views of subjects on the objects that the researcher takes their perspective. A detailed description of everyday situations is used. It is about understanding the actions and meanings in their social context. Qualitative research does not reduce the number of intentions or relationships between them, and their reduction is decided by the surveyed subjects themselves. Open and unstructured research plans are preferred. The analysis is based on a large amount of information on a small number of individuals. The interest in complex situations, the interaction between users and individual destinies prevails. The goal of qualitative research is to create a holistic image of the subject under study to capture how participants interpret the situation and capture these interpretations.

Advantages

- Explore the phenomena more deeply and more widely
- Explore phenomena in a relatively natural environment
- Allows formulating hypotheses
- Information about the participant also from his behavior

Disadvantages

- Time consuming
- The results are influenced by the researcher
- The results cannot be generalize

I use qualitative methods in collecting data because I want to explore the phenomena in depth and have the opportunity to remove the undesirable effects of all kinds. The population is small and known about 30 people. The population is easily available, but often very busy. Qualitative research is appropriate because collecting data during simple talks in many iterations does not bother busy workers and allow to collect lots of data. More about collecting data in the chapter 2.2.

2.2 Data collection

The data collected may come from many sources. There are also few techniques to use when collecting primary data. Below are some of the most common techniques for data collection.

- Observation
- Interviews
- Survey
- Focus Group
- Expert Estimation

I originally came out of the assignment from the company, it was not necessary to use some complex method for collecting data. However, I used a few of these methods in a semi-structured form. In the course of a year and a half, I work in an open office together with my colleagues for whom I design the system. I observe both the role of supervisor and artist in all-day situations. Due to this observation, I can offer kind of

an expert estimate. The most of the staff do not have time and so the longer interview with them is difficult, so I chose the form of a semi-structured interview in any possible situation, at lunch, during a coffee break or at a corporate party. During the testing period, I had the opportunity to speak with at least fifteen staff from the population. Using brief interviews I collected data in many iterations. I also had a meeting with three people from management and also my five colleagues, where I suggested this topic. A few minutes of conversation resembled a focus group, where each of the participants suggested what would be beneficial for the system to include. And I took the role of moderator.

Thanks to this form of semi-structured data collection methods, I was able to recognize the perceptions and needs of more than half of people in the population. It would not be possible without this form.

The result of this data collection was the specific functional requirements for the system. Their concretization was still taking place in the course of a consultation process after reading use cases, scenarios, and storyboards.

2.3 Functional requirements

Design methods employed by teams during the requirements definition phase provide the much-needed connection between the user and other models and the framework of the design. This phase employs scenario-based design methods with the important innovation of focusing scenarios on user tasks in the abstract, but first and foremost on meeting the goals and needs of specific users. [1]

2.3.1 Requirements prioritization

Prioritization is a crucial step towards making good decisions regarding product planning for single and multiple releases. Various aspects of functionality are considered, such as importance, risk, cost, etc. Prioritization decisions are made by stakeholders, including users, managers, developers, or their representatives. [4]

2.3.2 Requirements list

Formulation of the requirement typically has the following form.

```
1 <ID> <name> <priority> <description>
```

Priority

1.	Low	It is not necessary to implement this requirement
2.	Medium	It is desirable that this requirement will be implemented
3.	High	This requirement must be implemented in the system

1. System block a user access to the system if he is not logged in (priority: 1)
You'll need to sign in to use the system. The login information is only available to UPP staff who will use the system.

2. Create a project (priority: 3)
An authorized user can create a new project. When creating a project, the authorized user can assign the supervisor, the 3D lead and the 2D lead of the project. The created project can be edited by the authorized user in the process.

3. Create a sequence (priority: 2)
An authorized user can create a new sequence in a project. The project may contain more sequence i.e. more version, e.g. director's cut, commercial for different areas or commercial with different timing. The created sequence can be edited by the authorized user in the process.

4. Create a shot (priority: 3)
An authorized user can create a new shot in a sequence. The number of shots is predefined but can change over time. The sequence usually consists of several shots. The authorized user can assign the artist for the particular asset. The created shot can be edited by the authorized user in the process.

5. Create an asset (priority: 3)
An authorized user can create a new asset in a shot. An asset is an individual activity i. e. modeling, texturing, compositing. Combining these assets will create a shot. The authorized user can assign the artist for the particular asset. The created asset can be edited by the authorized user in the process.

6. Create links between assets (priority: 3)
An authorized user can create links between assets and create a pipeline for individual shots. The links may not be linear and the user will be able to set dependencies.

7. Comment on individual projects/sequence/shot (priority: 1)
Users can comment a project/sequence/shot. After adding a comment, you can see the time of the comment and who wrote it.

8. Comment on individual asset (priority: 3)
Users can comment an asset. After adding a comment, you can see the time of the comment and who wrote it. This feature will often be used. Using the function, the supervisor will comment on the progress of the work and the results of the individual asset.

8. Add new version of an asset (priority: 3)
Allows the user to add a new version of the asset. The version adds the path to the file, time of adding and the list of comments that have been implemented in the new version.

9. Approve an asset version (priority: 3)
The responsible user, i.e. supervisor, 3D lead, 2D lead, can approve the asset version. After approving the work is done on this asset.

10. Notify a responsible user (priority: 1)
The responsible user i.e. supervisor, 3D lead, 2D lead is notified of the newly added version of the asset for which he is responsible or if there is a pending asset to which no one is assigned.

11. Notify the assigned user (priority: 1)
The user i.e. artist is notified if someone commented or approved the asset to which he is assigned. Or is assigned to an asset that is the next one in the pipeline and the responsible user has approved the previous asset.

12. Filter in the system (priority: 1)
The system allows the user to filter data for project, sequencing, shots, and assets. Filter by name, artist/supervisor, and filter by type of asset.

13. Connect the system with UPP database (priority: 1)
The system uses the already created UPP database with resources - Booking.

14. Connect the system with UPP notification system (priority: 1)
Notifications of certain people will take place via UPP in-house notification system - Pusher.

2.4 Use cases

Use cases are a technique based on descriptions of functional requirements of the system, often of a transactional nature, focusing on low-level user action and accompanying system response. [1]

UC0 Login
For all the use cases described below, these validation is preceded.

Basic flow of events:
1. User opens application in web browser and enters login and password
2. User is validated

Alternative flows:

2. User is not validated
   2a. Application displays error message

UC1 Create a project

Use case allows a user to create a new project.

Basic flow of events:

1. User selects to create a new project
2. User fills the name of the projects, path to project and fills a number of sequences
3. System stores the data

Alternative flows:

2a. Name of project is invalid
   2a1. System alerts user and does not allow to create project
2b. Path of project is invalid
   2a1. System alerts user and does not allow to create project

UC2 Create a sequences

Use case allows a user to create sequences on the project

Basic flow of events:

1. User selects create new sequences
2. User fills the name of the sequences
3. System stores the data

Alternative flows:

2a. Names of sequences are invalid
   2a1. System alerts user and does not allow to create sequences

UC3 Create a shots

Use case allows a user to create shots on the project.

Basic flow of events:

1. User selects create new shots
2. User fills the number of the shots
3. System stores the data

UC4 Create an asset

Use case allows a user to create assets.

Basic flow of events:

1. User selects create new asset
2. User choose which asset wants to create
3. System stores the data

UC5 Create assets links
Use case allows a user to create assets links.

Basic flow of events:

1. User selects a shot where he wants to create links between the assets
2. User sets relation for all assets
3. System validates the relation
4. System stores the data

Alternative flows:

3a. Links between assets are not correct
    3a1. System alerts user

UC6 Comment on project/sequence/shots/asset
Use case allows user to comment project/sequence/shot/asset.

Basic flow of events:

1. User selects the project/sequence/shots/asset which he wants to comment
2. User fills the comment text and sends it
3. System stores the comment, including the time and the name of the commentator
4. System notify a assigned user (UC9)

UC7 Add new version of an asset
Use case allows a user to add a new version of the asset.

Basic flow of events:

1. User selects the asset which he wants to add the new version
2. User fills the name, path, and description of a new version
3. User select which comments have been implemented in the new version
4. System stores the name and path, including the time and the name of the commentator
5. System notify a responsible user (UC8)

Alternative flows:

2a. Name of new version is invalid
    2a1. System alerts user and does not allow to add new version
2b. Path of new version is invalid
    2b1. System alerts user and does not allow to add new version

UC10 Approve an asset version
Use case allows a user to approve a version of the asset

Basic flow of events:

1. User selects the asset which he wants to approve
2. User approves the version
3. System changes the status of the asset
4. System notify an assigned user (UC9)

Alternative flows:

2a. User disapproves the version
  2a1. System changes the status of the asset
  2a2. System notify a assigned user (UC9)

UC11 Filter projects/sequences/shots/assets

Use case allows user to filter projects/sequences/shots/assets. Filtering by artist, asset, date, and supervisor.

Basic flow of events:

1. User selects the projects/sequences/shots/assets where wants to filter
2. User select what he wants to filter
3. System shows the filtered elements

UC12 Assign artist to asset

Use case allows user assign responsible artist to asset.

Basic flow of events:

1. User selects the asset
2. System offer the list of artists
3. User assigns the responsible artist
4. System stores the data

UC13 Assign supervisor to project

Use case allows user assign responsible supervisor to project.

Basic flow of events:

1. User selects the project
2. System offers the list of supervisors
3. User assigns the responsible supervisor
4. System stores the data

UC14 Assign 3D/2D lead to project

Use case allows user assign the responsible 3D/2D lead to the project.

Basic flow of events:

1. User selects the project
2. System offer the list of 3D/2D leads
3. User assigns the responsible 3D/2D lead
4. System stores the data

2.4.1 Use case diagram

Shows the behavior of the system viewed as the user. The purpose of the diagram is to describe the functionality of the system, what role users interact with and the system and what they expect. The diagram tells what the system can do, but does not say how it will do. See the Figure 1.

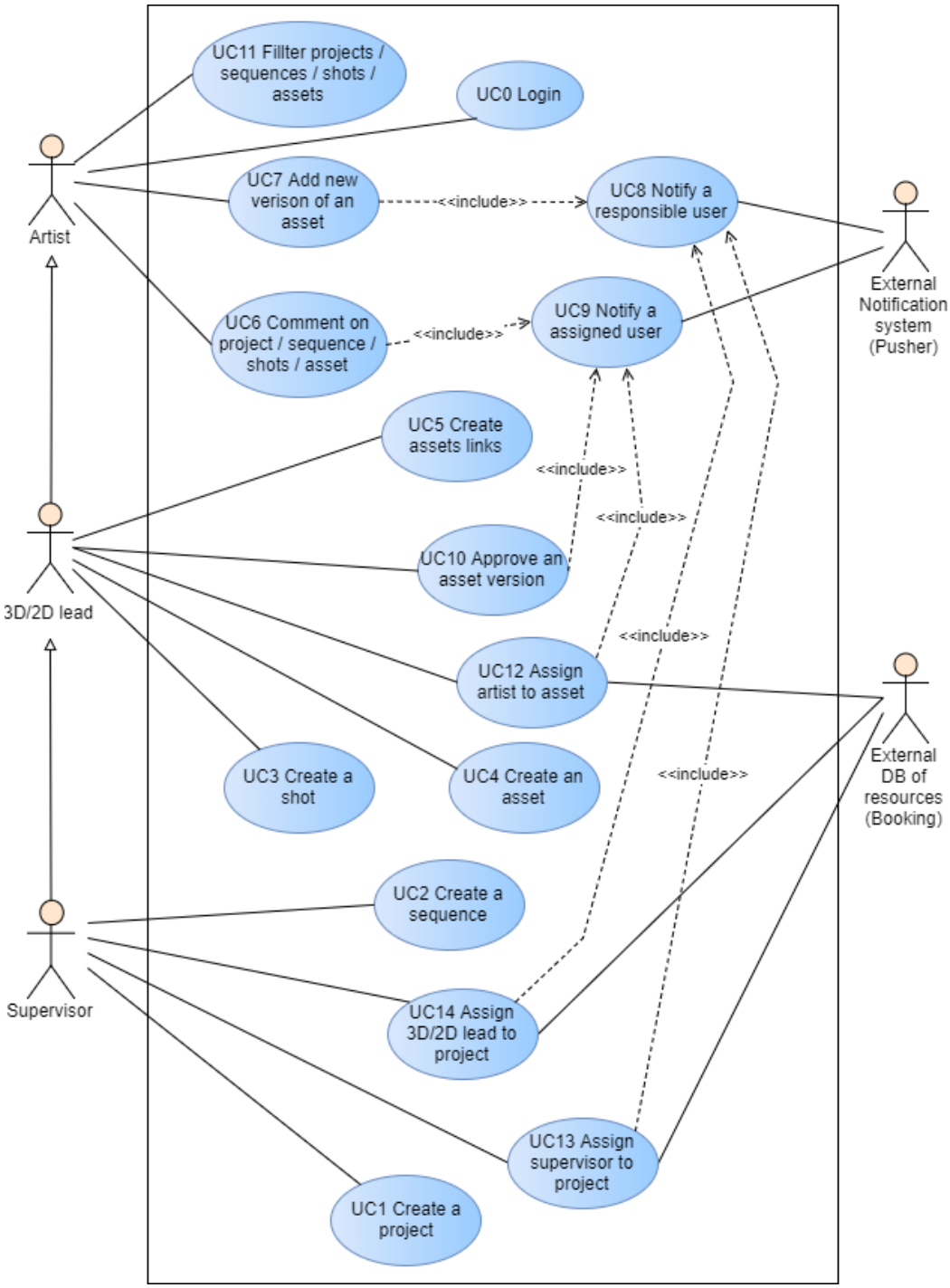


Figure 1 Use case diagram

2.5 Scenarios

Narrative, or storytelling, is one of the oldest human activities. Much has been written about the power of narrative to communicate ideas. However, a narrative is also one of our most powerful creative methods. Imagining a story about a person using our product leverages our creativity to a greater power than when we just imagine a better form factor or configuration of screen elements. Further, because of the intrinsically social aspect of narrative, it is a very effective and compelling way to share good ideas with team members and stakeholders. Ultimately, experiences designed around narrative tend to be more comprehensible and engaging for users because they are structured around a story. [1]

See the list of the main scenarios below.

Scenario 1
The scenario uses these use cases: UC0, UC1, UC2, UC3, UC4, UC5, UC12, (UC9) UC13, and (UC8).

It is working day afternoon and Kristina from production department arranged that UPP will work on the new commercial. The commercial will be fully CGI. The UPP received materials, storyboards, timings of every shot and few references. Kristina opens the system and logs in. She writes the project name and creates the new project. The system provides her possibility to assign the supervisor of the current project. If she does so the system notifies the supervisor.

Kristina or the chosen supervisor can set pipeline i.e. to determine the individual parts of the pipeline and connections between specific parts i.e. sequence and dependence of work. The system alerts her if the pipeline is not logical or incorrect. If Kristina or supervisor wants to set worker on specific part e.g. modeling, the system offers her list of workers/artists who are appropriate for the job. After completion of the work the system automatically notifies the workers who can start working without the necessary previous work in the pipeline.

Scenario 2
The scenario uses these use cases: UC0, UC12, UC6, (UC9), UC7, and (UC8).

Supervisor in the system sets Francesco the young artist on particular job e.g. texturing. In the meantime, Francesco is finishing another job. When suddenly gets notification from the system that he was assigned to a new job. He accepts the job. The system changes status that he accepted the job.

After completing the previous job Francesco opens the system. The system shows him projects where is he assigned and particulars jobs there. Francesco checks what he should do and informs the system that he just starts working on the job. In the system, he can see path to current references and path to materials i.e. the path to model that he will be texturing.

When the job is done, Francesco informs the system. The system notifies the super-

visor. Supervisor opens system and checks if the job is done correctly.

If everything is correct. The supervisor informs the system about it. Alternatively, if the job is done incorrectly he writes a comment into the system. The system stores all comments and notifies Francesco what he should remake it. When the changes are done Francesco informs the system again about it. The system notifies the supervisor again. If supervisor finally approves the changes the system sends a notification to another artist in the sequence of particular work.

Scenario 3

The scenario uses these use cases: UC6, (UC9).

Kristina from production department showed client existing progress. The client didn't like the result so far and decided to change existing model. Kristina told this situation to the supervisor of the project. The supervisor writes client's comments to the system. The system automatically informs worker, who did the model i.e. who did the final version of the model. When the worker receives the message he is allowed to accept or decline the work. If he declines, the system informs the supervisor to delegate work to another worker. If he accepts the system informs the supervisor, that the work is set.

Scenario 4

The scenario uses these use cases: UC11.

The client changes mind and wants to remake the whole model, although series of work was done with this model. The supervisor opens the system and filters the sequence of specific work which the model influenced. The system shows him the pipeline which has to be remade. The supervisor immediately knows how many works need to be remade to inform the client about the time delay.

2.6 Storyboards

By using a sequence of low-fidelity sketches accompanied by the narrative of the key path scenario, you can richly portray how a proposed design solution helps people achieve their goals. This technique of storyboarding is borrowed from filmmaking and cartooning, where a similar process is used to plan and evaluate ideas without having to deal with the cost and work of shooting the actual film. Each interaction between the user and the product can be portrayed on one or more frames or slides. Advancing through them provides a reality check for the coherence and flow of interactions. [1]

Storyboards were created using the Pixton web application, which focuses not only on storyboards but also on comics and graphic novels. The application can create fully possible characters to dynamic panels, props, and speech bubbles, every aspect of a comic can be controlled in an intuitive click-n-drag motion. [5]

Storyboards describe the situation from the scenarios in section 2.5. Although Scenario 1 and Scenario 3 describe a different situation, interactions between the user and

the system are similar. Therefore, there is no need to create Storyboard describing Scenario 3. The Storyboard for Scenario 4 will also not be created because of a simple user interaction with the system.

Storyboard 1

Storyboard describes the situation in Scenario 1.

The first of the storyboard shows the situation when a new project is being created. They show the situation when, at the beginning of a project, the main actor is production manager and using the system shifts responsibility to the supervisor and subsequently to the artist. Thanks to the storyboards we can see the continuity of iterations and their communication via the system.

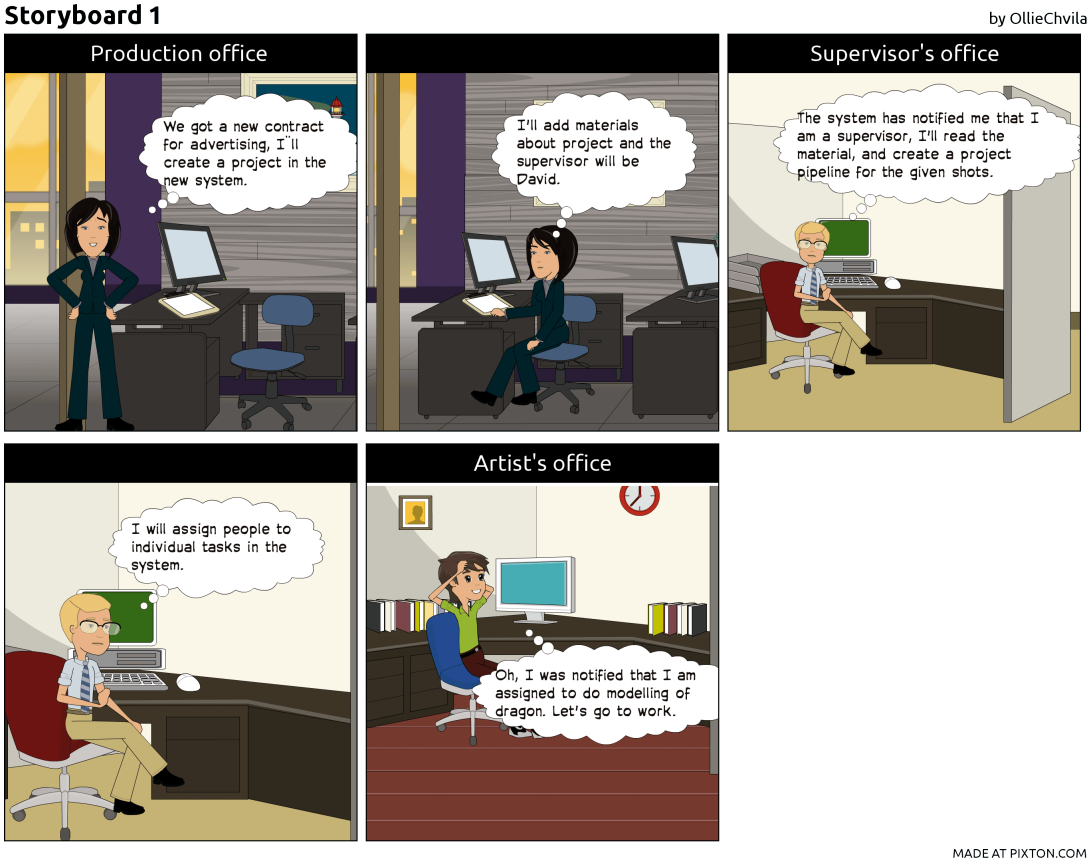


Figure 2 Storyboard 1

Storyboard 2

Storyboard describes the situation in Scenario 2.

The second storyboard shows the communication between the supervisor and the artist and how the individual actors are informed about the status of the process. This is a communication where the supervisor assigns a task to the artist and after completing this work, the supervisor is notified to approve the work as done.

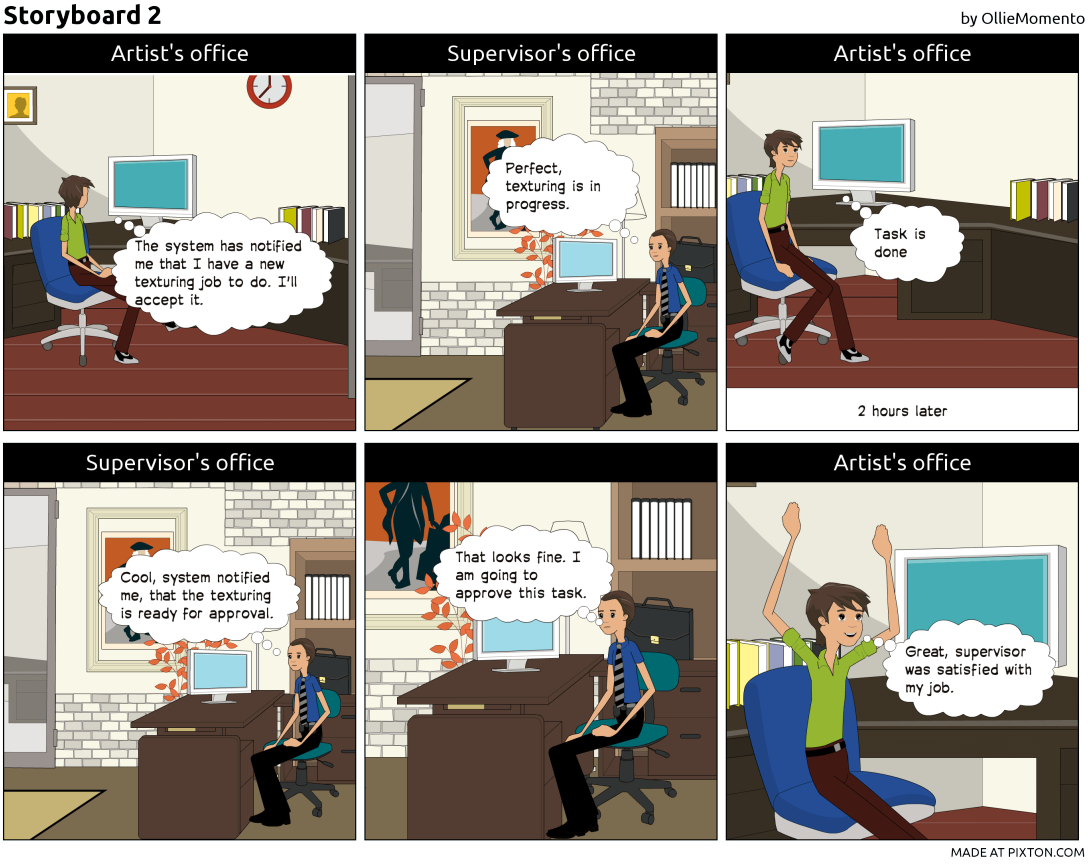


Figure 3 Storyboard 2

3 Visualization

In this chapter, we will focus on visualizing data using two methods. Then, we will discuss how to visualize the regions of interest in the graph and, lastly, the state visualization for the tasks on particular projects.

3.1 Data visualization methods

There is a large amount of data visualization methods. These methods have many advantages such as interactivity, the usability of interface features etc. In this diploma thesis, we will focus on only two conventional visualization methods, namely table, and graph (diagram).

3.1.1 Table

The table is the simple, easy to understand, and easy to interpret data representation technique. The table is a structured format, organized by rows and columns that convey relationships, is one of the most common definitions. Rows often represent variables and columns represent records with the set of values. [6] In our case, the table would look like the row is a sequence, shot, asset, or asset version. In the columns are their individual parameters. For example for an asset would be their type, name, status and who is assigned to work.

This method is implemented in all software that deals with the same topic. See the list of software in the inspiration library in the chapter 4.1. See the Figure 4 as an example of a table.

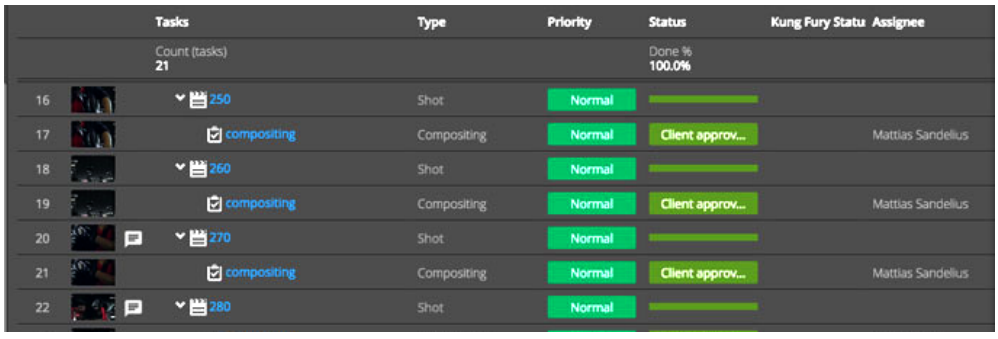


Figure 4 Table from ftrack App (ftrack 2013)[1]

[1] ftrack 2013. `https://www.ftrack.com/wp-content/uploads/2017/10/ftrack-interface.jpg` (visited on 03/28/2018).

3.1.2 Graph

Generally, a graph is an abstract data type used to represent relations among a given set of data entities. The data can be represented by the nodes of a graph, with the edges representing the relations. Graphs are used in numerous applications within the field of information visualization, such as state-transition diagrams, and social networks. The size and complexity of graphs easily reach dimensions at which the task of exploring and navigating gets crucial. Moreover, additional requirements have to be met in order to provide proper visualizations. [7]

There are many types of graph, for this case, it is appropriate to use the oriented graph, where the arrow direction visualizes the sequence of works. It may also be a cyclical graph and even a bipartite one. More about the limitations on the individual graph in the subsection 6.6.1. Interaction with a graph, i.e. the addition of vertices and edges and positioning vertices in space. Setting the automatic layout would confuse the user, and in addition, in our case, we do not expect the graph to have a large number of vertices. The graph could have back and parallel edges, it is necessary to layout at least these edges to avoid overlapping.

Data visualization with graph can be found in many software used in postproduction, so users are accustomed to this visualization. Softwares like Houdini, Nuke, Flare. See the Figure 5 an example of a graph.

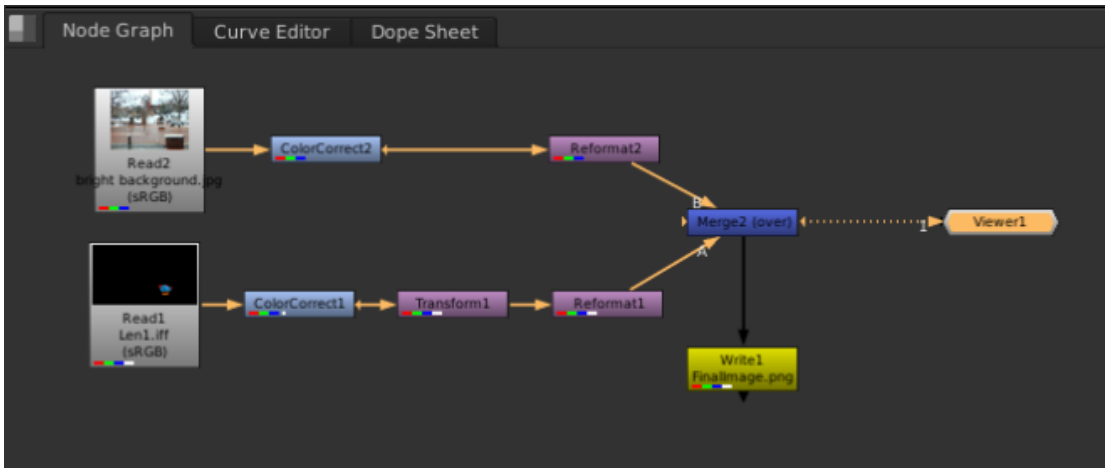


Figure 5 Graph from Nuke (Radhika Dhaipule 2016)[2]

3.2 Region of interest in a graph

Displaying all the information contained in the graph (assuming it can fit on a screen) shows the global graph structure, but has the drawback that details are typically too small to be seen. Alternatively, zooming into a part of the graph and panning to other parts show local details but loses the overall structure of the graph. Another option is

[2]Radhika Dhaipule 2016. `https://radhikadhaipule.files.wordpress.com/2015/09/nodegraph.png?w=656` (visited on 03/17/2018).

to use two or more graph views - one view will capture the entire graph and the other zoomed portion - so we gain the advantage that the user sees both the details and the overall structure of the graph. The problem, however, is that we force the user to a greater cognitive load in order to connect the two views. [8]

3.2.1 Zoom and pan

Zoom and pan are traditional tools in visualization. They are quite indispensable when large graph structures are explored. Zoom is particularly well suited for graphs because the graphics used to display them is usually fairly simple (lines and simple geometric forms). This means that zoom can, in most cases, be performed by simply adjusting screen transformations and redraw the screen's contents from an internal representation, rather than zooming into the pixel image. In other words, no aliasing problems occur.

Zooming can take on two forms. Geometric zooming simply provides a blow up of the graph content. Semantic zooming means that the information content changes and more details are shown when approaching a particular area of the graph. The technical difficulty, in this case, is not with the zooming operation itself, but rather with assigning an appropriate level of detail, i.e., a sort of clustering, to subgraphs. [6]

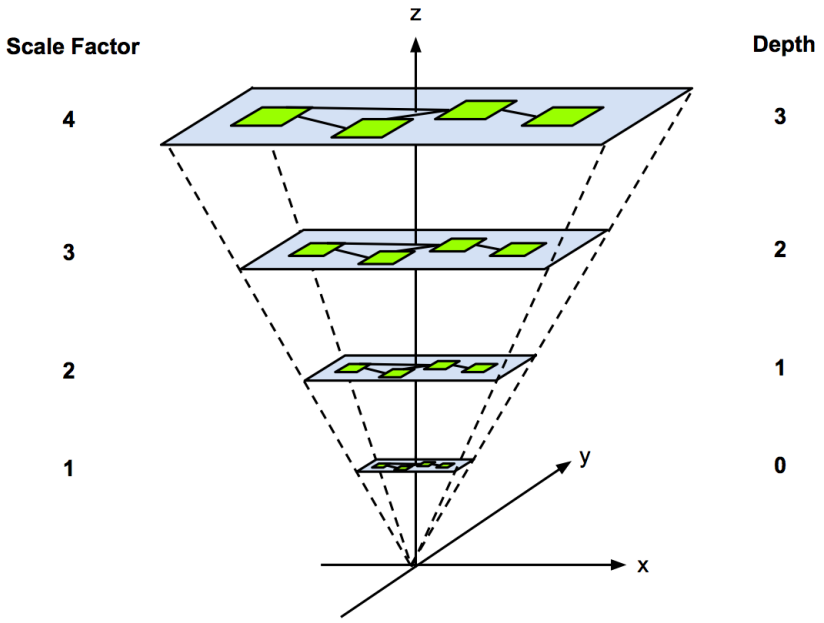


Figure 6 Space-scale diagram (Joel Oughton 2011)[3]

Furnas and Bederson introduce the concept of space–scale diagrams. See the Figure 6. The basic idea is to define an abstract space "by creating many copies of the original 2D picture, one at each possible magnification, and stacking them up to form an inverted pyramid". [6]

[3] Joel Oughton 2011. `http://joeloughton.com/blog/wp-content/uploads/2011/10/nav2-0.png` (visited on 03/17/2018).

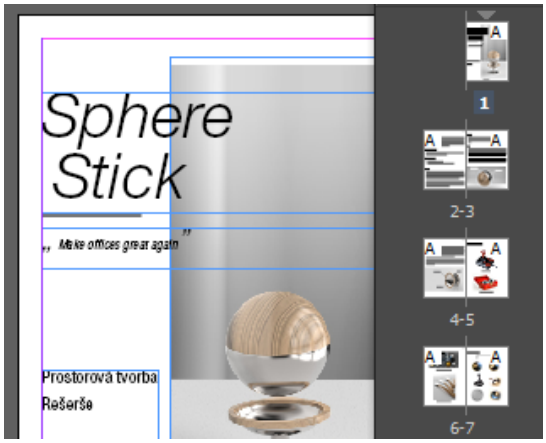
3.2.2 Overview-detail

This technique is characterized by the simultaneous display of overview views and detail. Both of these views are located in separate presentation areas. Thanks to the physical division of the image into two views, the user can work with each view separately. However, actions were taken in one view usually immediately reflect the view of the other and vice versa.

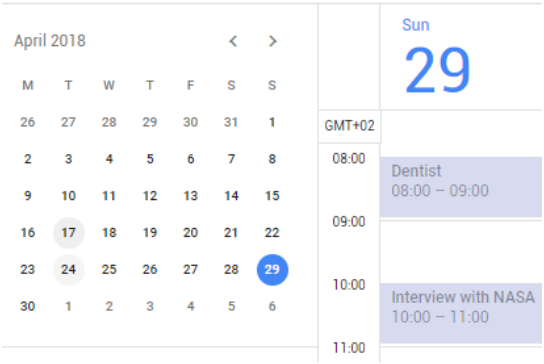
This interactive technique conserves overview to prevent disorientation in the detail view but endures visual discontinuity between overview and detail view. Both views compete for screen space and user have a larger cognitive load to connect these two views. [9]

Many forms of overview-detail interfaces exist, both in the standard desktop environment and in research systems, with important features including the ratio of scales in the detail and overview regions, the relative size and positioning of the views, mechanisms for navigation control, and the coupling between overview and detail displays. [10]

An example can be a form of overview-detail where a detailed view is just an enlarged part of the overview using the geometric zoom. See Figure 7a. Or another example is a calendar where the overview is the list of days of the month. Depending on the selected day, the detail is the event list for that day. See Figure 7b.



a) Adobe InDesign CC 2017



b) Google Calendar 2018

Figure 7 Overview-detail examples

3.3 States visulalization

In VFX and postproduction, it is necessary to consider that project states are constantly changing in every phase of creation. The visualization should, therefore, be such that the user has a constant overview of the status of the project. The project can have multiple sequences that consist of shots, each shot is created by sequencing assets, and each asset can have several versions. At each stage of the hierarchy, it is, therefore, necessary to visualize the states. See the Figure 8, the example of state visualization.

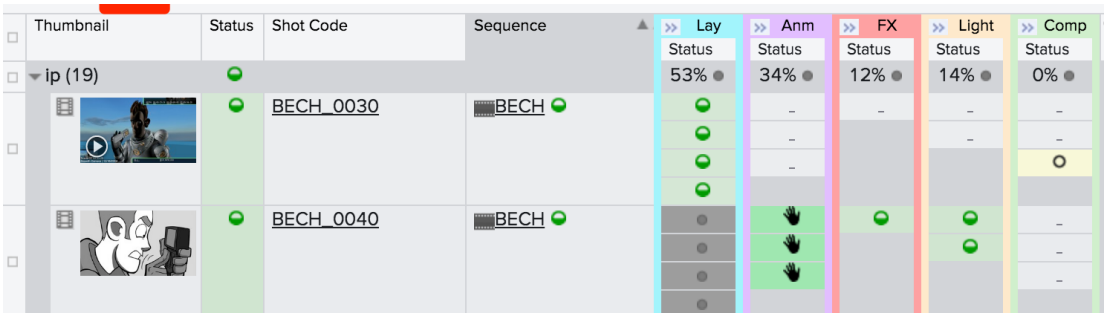


Figure 8 Shot states visualization (Shutgun Software 2014)[4]

Visualization of a status of project, sequence, shot and asset i.e. not started, in progress or done. For asset even if the version is approved or rejected by the supervisor or if the approval is still waiting. Another state to visualize is if the supervisor is assigned and who works on each element. It would be helpful to integrate this state into a graph using some visualization channel (if it is displayed as a graph) so as to offer a textual description of the states.

States for individual assets can also help in assigning sources. The user can then predict that if an asset impedes the process of the project, it is necessary to assign more resources to it. Otherwise, if the asset is done, it is possible to visualize the release of the used resources. It is also advisable to visualize the status of the individual asset waiting for approval so that the user can immediately see who is on the turn to complete the job.

[4]Shutgun Software 2014. `https://support.shotgunsoftware.com/hc/en-us/article_attachments/114094098973/02_shots_list_view.png` (visited on 03/17/2018).

4 Design of user interface

After creating the functional requirements, storyboards and scenarios of the system, we can proceed to the design of the user interface. However, this cannot be understood as only designing the graphic layout and appearance, but also take care of the basic features of user experience design, usability especially.

4.1 Design methodology

Design of user interface was divided into small parts. Individual parts were created using expert analysis, but also frequently consulted with the UPP management and last but not least, inspired by similar systems on the market. The few systems dealing with a similar topic exist on the market, see the list below.

Inspiration Library:

- Shotty App [1]
- Flix (The Foundry) [2]
- Shotgun Software [3]
- Ftrack App [4]

4.2 Low fidelity prototype

The low fidelity (hereinafter referred to as Lo-Fi) prototype was created in the Balsamiq Mockup program. Balsamiq Mockup is a rapid wireframing tool. It reproduces the experience of sketching on a whiteboard, but using a computer. It also provides interaction such as a transition between screens of prototype using active elements. [11]

Given that 3D lead and 2D lead have similar rights as the supervisor, these two roles will only be replaced by the supervisor role. Two prototypes were created to maintain clarity. The UI is similar to supervisor and artist, the difference is only in the user's rights.

Homepage screen

The homepage screen will be displayed to the user after login. The difference between the supervisor's and the artist's homepage is that the supervisor has assigned projects

[1] Shotty App. `http://shottyapp.com/` (visited on 01/18/2018).
[2] Flix. `https://www.foundry.com/products/flix/` (visited on 01/18/2018).
[3] Shotgun Software. `https://www.shotgunsoftware.com/` (visited on 01/18/2018).
[4] Ftrack App. `https://www.ftrack.com/` (visited on 01/18/2018).

under the supervision and artist has the assigned assets on the main page for quicker access. On the artist's homepage is not a button for creating a new project. See the Figure 9.

a) Supervisor **b)** Artist

Figure 9 Homepage screen

Project, sequence and shot screen

The user interface for the projects, sequence and shot screen is similar. On the left side of the screen is the information panel about the selected project, sequence or shot. In this panel, you can switch to tabs, information, comments or a list of who works in this part of work. On the right, is space for the list of lower elements and also options to view chosen element using the node diagram. See the Figure 10.

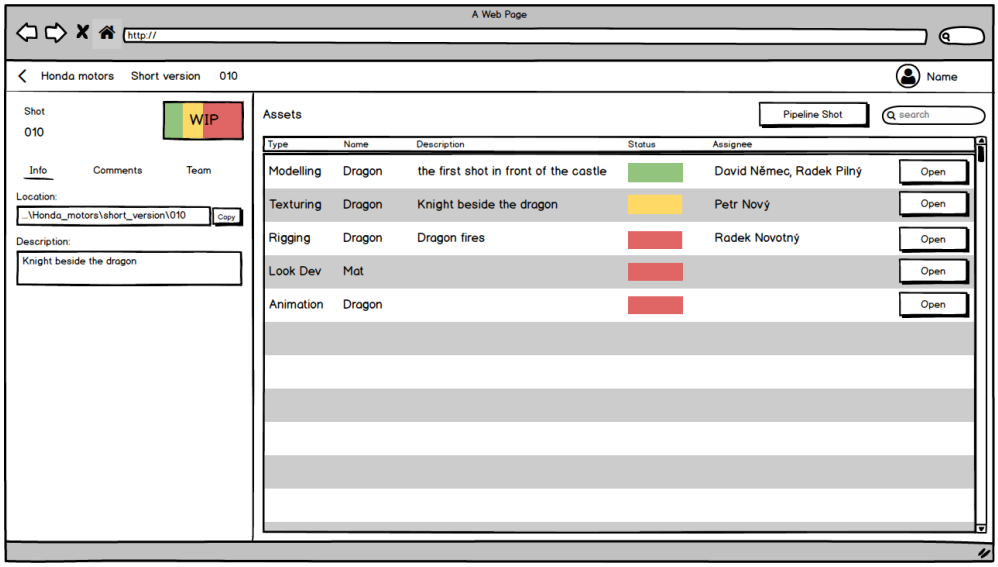


Figure 10 Shot screen

Sequence, shot pipeline screen

The screen covers the scenario 4 and use case UC12.

The different approach to data visualization is visualized on this page. This approach is using the node diagram and the context is expressed here by the oriented edge. On the left side of the screen is the information panel about the selected asset. See the Figure 11.

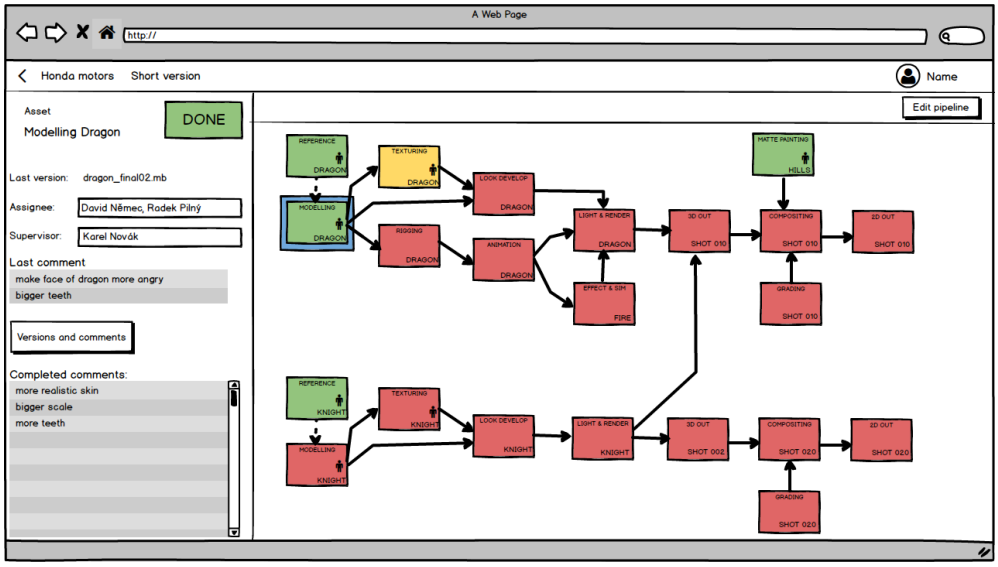


Figure 11 Sequence pipeline screen

Edit sequence, shot pipeline screen

The screen covers the use cases UC5, UC4 and scenario 1.

This screen shows how the interaction could be done when editing a pipeline of a shot or a sequence. On the left is the asset-building panel and on the right side panel, where you can change the task parameters if the asset is selected. See the Figure 12.

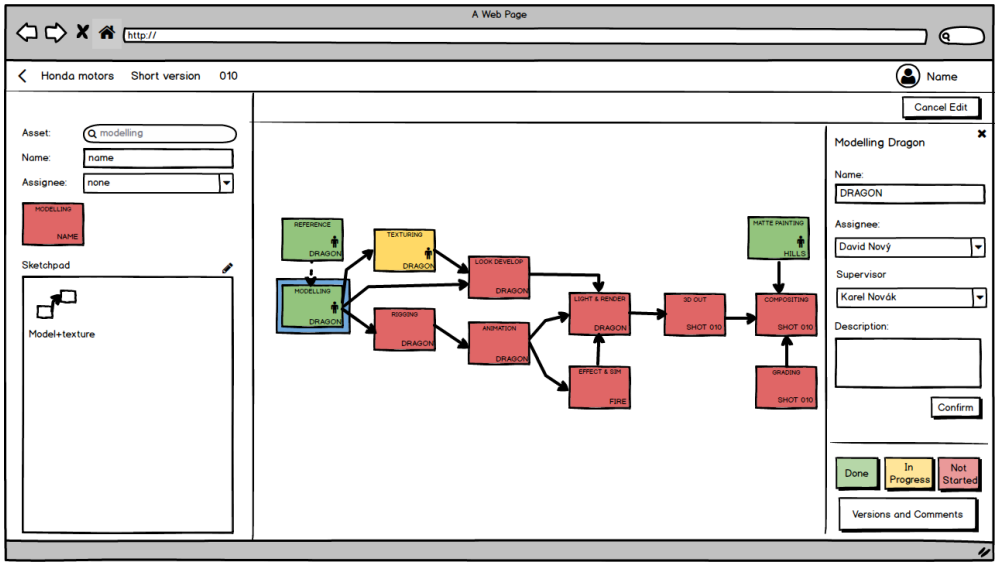


Figure 12 Shot pipeline screen in edit mode

Asset screen

The screen covers the use cases UC10, UC6, (UC9) and scenario 2, 3.

On this screen, we are at the lowest level in the project hierarchy, i.e. asset. On the left side, we find information about the asset. In the middle, we find individual versions

of the asset, sorted according to how they were added in time. On the right side of the screen, there are comments that have not been implemented and comments that have been implemented into a specific version. See the Figure 13.

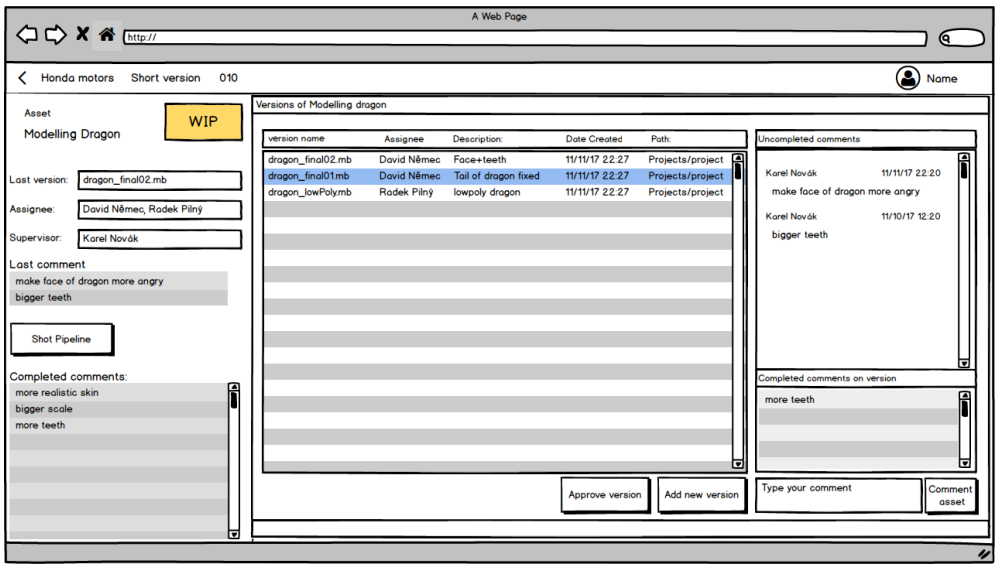


Figure 13 Asset screen

Adding new version of asset screen

The screen covers the use cases UC7, (UC8) and scenario 2.

In this screen, the user can add the version of the asset, by first choosing the file to add the path to, so he can write the description and mainly choose which of the comments he has implemented into the version. See the Figure 14.

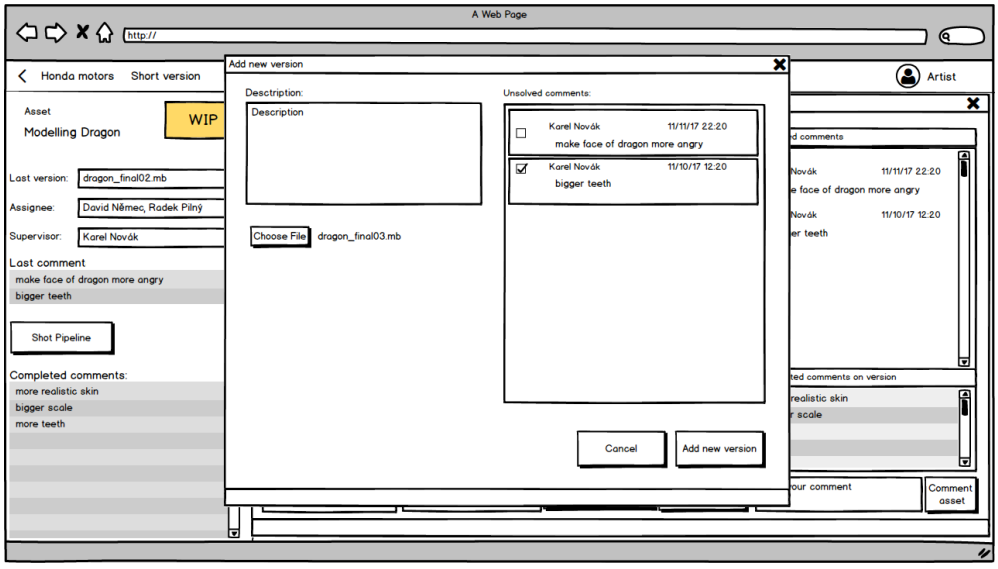


Figure 14 Adding new version of the asset screen

4.3 Prototype user testing

Usability testing is a collection of techniques used to measure characteristics of a user's interaction with a product, usually with the goal of assessing the usability of that product. Typically, usability testing is focused on measuring how well users can complete specific, standardized tasks, as well as what problems they encounter in doing so. Results often reveal areas where users have problems understanding and utilizing the product, as well as places where users are more likely to be successful. [1]

To test the usability of the proposed prototype, testing was performed on end users of the system. Important for testing was that users were able to navigate the system without explanation.

4.3.1 Test environment

Testing was run on a laptop where Balsamiq Mockups 3 was installed. [11] Where a clickable prototype was launched in full screen presentation mode. The laptop did not belong to the participant and a mouse was attached to it. The laptop screen has a 13.3-inch screen, although monitors in the environment where the system will be used, are at least 24 inches and usually they have two monitors. The OBS studio image and sound capture software has been installed on the notebook and each of the participants agreed to this recording.

4.3.2 Testing process

At the start of testing, participants were introduced to basic principles. It's the system, who is tested and not them, so they do not have a bad feeling when some of the tasks are not solved correctly. That there is no need to hurry to solve the task and that it is desirable to think aloud. They were told that if they thought they had finished the task to say it. After each task, they also returned to the main prototype page.

Before the testing itself began, a few questions were asked in pre-test. Demographics about the participant, such as gender and age. They were also asked about their position in the company and the software they use. In addition, they were asked the question and the statement to which they had to comment. The question and statement depended on their position in the company. A complete list of questions and answers can be found in the individual testing of each participant in the Appendix B.

In total, 5 participants were tested. Three from the position of supervisor and two from the position of the artist. Each group was tested on a different but very similar prototype (it differs only from the rights to some activities) and they had a list of tasks. The group of supervisors, in addition, had tasks for activities that the artist is not entitled to. To eliminate the ability to learn for each task, the order of tasks for each participant was scrambled. The list of tasks for each group you could find in the Appendix A.

After testing, there was a post-test that included additional questions about the

prototype. A list of complete questions and answers can also be found in the Appendix B.

4.3.3 Findings and recommendations

List of usability findings found during testing. Some were found in testing itself, some in additional questions after testing. See the Table 1.

Priority:

1. Cosmetic problem
2. Less usability problem, remove with lower priority
3. More serious problem of usability, important to remove
4. Critical usability problem, need to be removed

Tasks under the numerical designation can be found in Appendix A.

Table 1 Findings and recommendations from Lo-Fi prototype testing

Task number	Problem	Recommendation	Priority
S3	Open button in the list.	Make clickable row	3
S7, A4	Four panels with different types of comments are not clear	Make one or two panels of comments, but use another visualization channel, such as color	4
S7, A4	Naming the comments panel. Completed, uncompleted, completed on version	Change the name to done, pending, and implemented.	1
S8	Participants did not know whether the changes had already been made.	Add confirm button for editing	4
S8	The participant did not know what the button “Cancel Edit” is doing. If it saves the changes or cancels the changes.	Rename the button to “Cancel Editing Mode”	2
S4	Button order for “Cancel” and “Create”	Switch order or delete “Cancel” button and let just exit button in the window bar.	1
S1, A2	Little visibility of tabs	Make tabs more visible and change it to know it’s clickable	1
A1	Little visibility of sequence and shot pipeline button	Make it more visible, or place it on the main window bar.	2

Testing also showed that the participants did not understand the prototype hierarchy. This is primarily a two-way approach using a graph and a table approach. Table approach also forces users to remember in which sequence the shots and the assets are.

Table access also confuses a user because it duplicates data, so it is possible to get to the same place via several paths.

For this reason, the table approach will not be implemented in subsequent phases. However, the table approach allows you to see a large amount of data, it might be a problem if we were to visualize huge amounts of data using only graph approach, we would have to focus on filtering, grouping, and semantic zooming. Because of the difficulty, therefore, the prototype will first be made for a medium amount of data. From later testing, we will see if this estimate was correct.

5 Technology analysis

JavaScript (JS) is essentially the most used scripting or programming language worldwide that allows web developers to implement complex things on web pages. Choosing a JavaScript framework for our project will be crucial. JavaScript frameworks are developing at an extremely fast pace, meaning that today we have frequently updated versions of Angular, ReactJS and another quite new framework Vue.js.

5.1 Properties of individual framworks

In order to decide which framework to use, it is good to ask the following questions.

- How mature are the frameworks?
- Are the frameworks likely to be around for a while?
- How extensive and helpful are their corresponding communities?
- How easy is it to use the frameworks for our project?
- What does the learning curve look like for each framework?

One way to answer the questions is to look at the npm trend graph. They show the number of downloads for the given npm packages. It can be seen from the Figure 15 that the strongest user base has ReactJS. Angular and Vue.js are similarly downloaded frameworks.

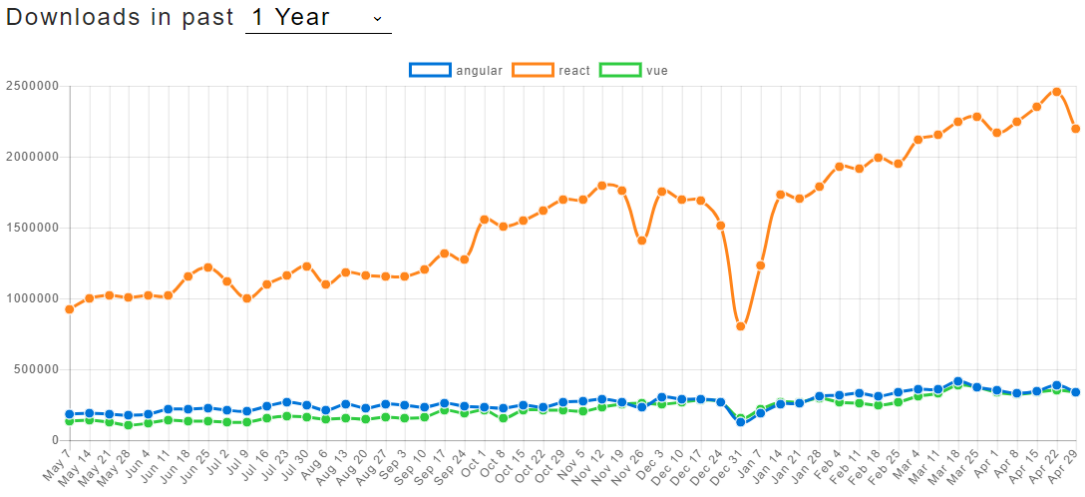


Figure 15 The npm download numbers for the given npm packages in the last 1 years

ReactJS

Advantages

- Easy to learn because of simplicity in term of syntax. No need to deeply learn TypeScript like in Angular.
- High level of flexibility and maximum of responsiveness.
- Virtual DOM (document object model) that allows arranging documents in HTML, XHTML, or XML formats into a tree from.
- Combined with ES6/7.
- Dozens of packages.

Disadvantages

- Lack of official documentation because of fast development.

Angular

Advantages

- Detailed documentation.
- MVVM (Model-View-ViewModel) that allows developers to work separately on the same app section using the same set of data.

Disavantages

- Complex syntax.
- Migration issues which can appear while moving from the older version to the latest ones.

Vue.js

Advantages

- Detailed documentation.
- Awesome integration. Can be used for building STA (single-page applications) and also difficult web applications.
- Large scaling.
- A tiny size that keeps it fast and flexible that allows reaching much better performance.

Disavantages

- Small user base.

From the above analysis, ReactJS framework has been chosen as the most promising for the project. Vue is suitable for use in a single-person team, but a small user base is a big obstacle and, moreover, React offers a lot of tutorials online, which makes it an easy to understand. Angular, on the other hand, is rather for large applications with many developers in a team.

5.2 ReactJS

The framework was developed and is maintained by Facebook. It was developed in 2013 to eliminate the problems with building user interfaces. Nowadays ReactJS is used by world-famous companies like Facebook, Instagram, Netflix, New York Times, Yahoo, Khan Academy, Whatsapp, Codecademy, Dropbox, Airbnb, and Microsoft. ReactJS is an open-source javascript library that allows dynamic rendering. Views are rendered as HTML tags and inserted into the page. The views themselves are created as a component hierarchy in which subcomponents cannot affect parent components. The data flow is strictly set from top to bottom. [12]

5.2.1 Main features

Declarative

ReactJS creates very interactive and dynamic user interface for websites. Create simple views for each state in your application, and React will update and render just the right components when the data changes. [13]

Virtual DOM

In ReactJS, for every DOM object, there is a corresponding "virtual DOM object." A virtual DOM object has the same properties as a real DOM object, but virtual DOM could directly change what's on the screen.

JSX

JSX is a markup syntax, very similar to HTML. It is Javascript and XML compilation. JSX makes writing ReactJS components, the building blocks of ReactJS user interface, easier by making the syntax developers use for generating these strings of HTML they will inject into the web page. [13]

5.3 Graph drawing library

The main feature the prototype needs is a creating and interacting with a graph. For this reason, it is best to use an ideal graphics library that allows it. The following section lists the most commonly used libraries for working with graphs and their properties.

JointJS

JointJS can be used to create either static diagrams or fully interactive diagramming tools. It offers customizable links, their arrowhead and labels, also zoom in and out method. JointJS also has a paid version Rappid, which offers many examples. For the unpaid version, unfortunately, there is not so much. [14]

MxGraph

MxGraph is an interactive JavaScript HTML 5 diagramming library. MxGraph is a fully client-side library that uses SVG and HTML for rendering models. The biggest advantage of MxGraph is that is developed since 2005 so there are many examples and also uses no third-party software, it requires no plugins. This library is for instance used in Draw.io. The elements in MxGraph are at first glance very obsolete, so it is necessary to change the visual to all elements, but the great advantage is that this change is possible. Other libraries do not have to allow it. [14]

GoJS

GoJS is a feature-rich JavaScript library for implementing interactive diagrams across modern browsers and platforms. GoJS makes constructing diagrams of complex nodes, links, and groups easy with customizable templates and layouts. They also provide over 150 interactive samples and API is well documented. Although the GoJS library is the most promising for our project in terms of features. It is paid for commercial purposes. The price is $995 per license for less than 250 users. [14]

The MxGraph library has been selected for the project because it is fully customizable, so there will be no problem with the library being used for future functionality enhancements. It is also free for commercial purposes.

5.4 Database

In order for the prototype to store changed states and behave like a functional web application, it is necessary to deploy the database. There are two kinds of data structure - relational (SQL) and non-relational (NoSQL).

SQL database

SQL databases use structured query language (SQL) for defining and manipulating data. It is more rigid, structured way to store data. It is not possible to store different information in the same table. A relational database consists of tables with columns and rows. Each row represents a different record, and each column stores a specific type of information. [15]

NoSQL database

NoSQL database has a dynamic schema for unstructured data. It stores JSON-like field-value pairs of documents. Using collection similar documents can be stored which is analogous to an SQL table. [15]

It is not necessary for the data to be stored in a structured form, and when adding more functionality, it is necessary for the database to have a dynamic schema. NoSQL databases are also used by UPP and will, therefore, be used for this project.

6 Implementation

In this chapter, the reader will be familiar with how the crucial functions of the high-fidelity web application prototype have been implemented. Which packages were used in the implementation, how the prototype and database were connected and how navigation was solved. Additionally, the reader will be familiar with the implementation details for interacting with the graph.

6.1 Development environment

The main application development equipment is an integrated development environment (IDE) or a simple editor. For this project was used the source code editor by Microsoft - Visual Studio Code. It includes support for debugging, syntax highlighting, intelligent code completion, and code refactoring. It also provides the option to install support plug-ins for development over the ReactJS framework.

Using the command prompt, a localhost was launched on port 3000, and then a web browser was launched on that URL. At localhost on port 3001, a server was launched that offered data using rest API.

6.2 Package manager

To manage libraries and dependencies in the prototype was chosen NPM package manager for Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages called the npm registry. The best way to manage locally installed packages is to use the package.json file. The NPM provides documentation on which packages the project is dependent on and also the version of the package. It also facilitates sharing among developers by making the project reproducible.

6.2.1 Packages

More than 25 libraries were used to create the prototype. The complete list of libraries and specific versions see in the source codes in package.json file. In the list below are the main used libraries and their short description of the function.

axios[1] Axios is a promise-based HTTP client that works both in the browser and in a Node.js environment. It intercepts and transforms request and response data.

foreman[2] Foreman allows to run both the server with API and the prototype at the same time.

express[3] Express is a Node.js web application server framework. It provides small, robust tooling for HTTP servers, web applications, or public HTTP APIs.

history[4] History easily manages session history anywhere JS runs. It provides a minimal API that lets you manage the history stack, navigate, confirm navigation, and persist state between sessions.

material-UI[5] Material-UI is a set of React components that implement Google's Material Design specification.

mongodb[6] The official MongoDB driver for Node.js. Provides a high-level API on top of mongodb-core that is meant for end users.

mongoose[7] Mongoose is a MongoDB object modeling tool designed to work in an asynchronous environment.

mxgraph[8] MxGraph is a fully client side JavaScript diagramming library that uses SVG and HTML for rendering.

nodemon[9] Nodemon will watch the files in the directory in which nodemon was started, and if any files change, nodemon will automatically restart your node application.

react-router[10] React-router synchronizes the user interface with the URL of the web browser and provides an API that can route the URL to different parts of the application.

[1] axios. `https://www.npmjs.com/package/axios` (visited on 05/02/2018).
[2] foreman. `https://www.npmjs.com/package/foreman` (visited on 05/02/2018).
[3] express. `https://www.npmjs.com/package/express` (visited on 05/02/2018).
[4] history. `https://www.npmjs.com/package/history` (visited on 05/02/2018).
[5] material-UI. `https://www.npmjs.com/package/material-UI` (visited on 05/02/2018).
[6] mongodb.`https://www.npmjs.com/package/mongodb` (visited on 05/02/2018).
[7] mongoose. `https://www.npmjs.com/package/mongoose` (visited on 05/02/2018).
[8] mxgraph. `https://www.npmjs.com/package/mxgraph` (visited on 05/02/2018).
[9] nodemon. `https://www.npmjs.com/package/nodemon` (visited on 05/02/2018).
[10] react-router. `https://www.npmjs.com/package/react-router` (visited on 05/02/2018).

6.3 ReactJS implementation

ReactJS is an efficient and flexible framework for user interfaces as mentioned in section 5.2, the problem is when installing and configuring tools such as Webpack or Babel, which can be a complex process. That's why Facebook created a create-react-app[11] node module to generate and a boilerplate version of a ReactJS application. To create a project, you just have to install the create-react-app globally and use it to create a project.

```
1 npm install -g create-react-app
2 create-react-app uppweb
```

6.3.1 Component hierarchy

When the project is initialized, it is important to devote great preparation to the start of work. It is necessary to build a component tree from which a final UI application will be created because there is only one-way data flow. I took the design of the pages, where I have marked certain components using colored squares, this method is also recommended in the ReactJS documentation [16]. When deciding what should be own component, I follow the technique called single responsibility principle. That is, a component should ideally do only one thing. If it does more it should be decomposed into smaller subcomponents.

To create the hierarchy components, I used the Lo-Fi prototype from section 4.2 as an inspiration, because some components have been changed due to test results, so the technique will be shown directly on the Hi-Fi prototype.

Homepage

In the hierarchy, the highest component is the Homepage that covers the entire screen area. The Homepage component has direct descendants - Header. (The component is the same for all pages.) Another is the Toolbar. (There is a button to add a new project and a search bar that filters projects.) Another component is ProjectList, which has the descendants of all projects. See the Figure 16.

Shotpage

The next page is Shotpage. (Projectpage has the same layout.) The main component that covers the screen area is Shotpage. The direct descendants are Header, Breadcrumbs (Graphic displaying navigation in the prototype), Graph and LeftPane. LeftPane is parent component that has a direct descendant named NameAndStatus, which specifies the name and status of the shot. TabsBar is also a direct descendant of LeftPane, where tabs with Info, Comments, and Team are located. See the Figure 17.

[11] create-react-app . `https://github.com/facebook/create-react-app` (visited on 05/02/2018).

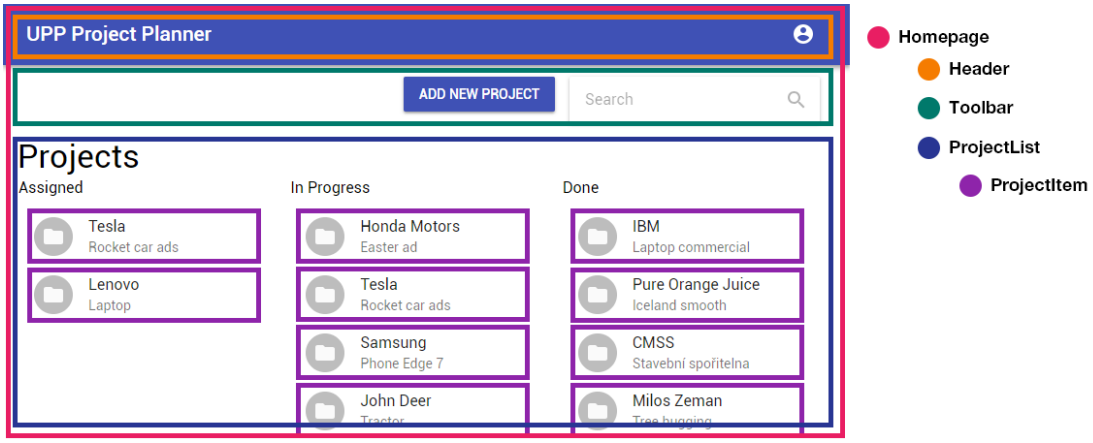


Figure 16 Homepage hierarchy of components

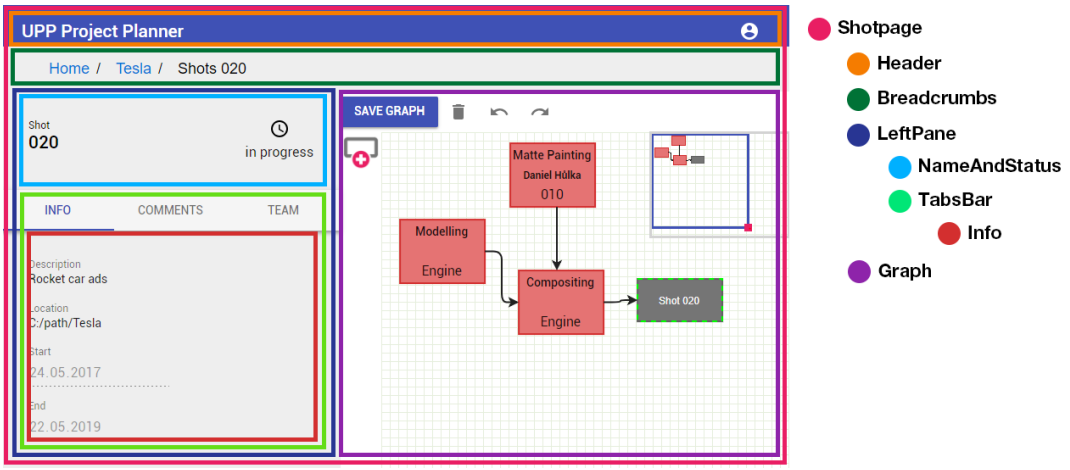


Figure 17 Shotpage hierarchy of components

6.3.2 Component implementation

There are two ways to implement a React component that is very similar. One approach is using the *React.createClass* method or using a syntax sugar for better use with ES6 modules by *extends React.Component*. The only required method in a component is a render method that is responsible for parsing the HTML in JavaScript. Below is a simplified example of creating a component.

```
1  import React, { Component } from 'react';
2  class Homepage extends Component {
3
4      render() {
5          return (
6              <div className="Homepage">
7                  <Header />
8              </div>
9          );
10     }
11 }
12 export default Homepage;
```

The example shows that the render method has div where is the Header component, the components can be chained to create the necessary hierarchy. However, this method does not allow the creation of dynamic applications. In order to pass the data around the system, we will need parameters called *props*. These *props* can be passed from the parent component to the child and can not be changed. We need *states* to update the data. In general, you should initialize *state* in the constructor and then call *setState* if you want to change it. In order to allow the parent component to be updated based on the state change in the child component, it can be used one of the *props* properties. The *prop* may not only contain the object but also the method. So it's possible to call the parent method from the child component. The example below shows that a project, people, asset and also the updateAsset method are passed to the RightPane component, where the bind method determines the context.

```
updateAsset(asset){
        this.setState({
            asset: asset
        })
    }

<RightPane
    project={this.state.project}
    people={this.state.people}
    asset={this.state.asset}
    updateAsset={this.updateAsset.bind(this)}
/>
```

6.3.3 Component lifecycle

Each component has several lifecycle methods that you can override to run code at particular times in the process. There is a selection of the most used methods.

Mounting

These methods are called when an instance of a component is being created and inserted into the DOM.

- constructor()
- render()
- componentWillMount()
- componentDidMount()

Updating

An update can be caused by changes to props or state. These methods are called when a component is being re-rendered.

- componentWillReceiveProps()
- shouldComponentUpdate()
- componentWillUpdate()
- render()
- componentDidUpdate()

Unmounting

This method is called when a component is being removed from the DOM.

- componentWillUnmount()

These methods are also used in the prototype implementation. For example, when it is necessary to first render the div on which is appended graph. Below is a simplified preview of how HTML is first rendered, and then, using *componentDidMount()*, calls the *loadGraph()* method to access the already-created divs using reference.

```
1  class Graph extends Component {
2
3      constructor(props) {
4          super(props);
5          this.state = {
6             //... removed for brevity
7          };
8      }
9
10     loadGraph() {
11             //... removed for brevity
12             var sidebar = ReactDOM.findDOMNode(this.refs.graphSidebar);
13             var toolbar = ReactDOM.findDOMNode(this.refs.graphToolbar);
14             //... removed for brevity
15     }
16
17     componentDidMount() {
18         this.loadGraph()
19     }
20
21     render() {
22         return (
23             <div className="graph" ref="divGraph" id="divGraph">
24                 <div className="graph-toolbar" ref="graphToolbar"
25                 id="graphToolbar" />
26
27                 <div className="graph-sbcont">
28                     <div className="graph-sidebar" ref="graphSidebar"
29                      id="graphSidebar" />
30
31                     <div className="graph-container" ref="graphContainer
    "
32                      id="graphContainer" />
33                 </div>
34                 <div id="outlineContainer" ref="outlineContainer" />
35             </div>
36         );
37     }
38 }
39
40 export default Graph;
```

6.4 Database and API

I have agreed with UPP management that connecting the database used by UPP and the prototype will only occur if we will agree to continue with the development. For the development of the prototype, it will be useful to use a custom database with mock data. The database technology uses UPP are the same as in the prototype, so future connecting will not be problematic.

As already mentioned in the section 6.1, it is necessary to create an API that will offer data on the port 3001 according to the specified URL. Below is an abridged version of how to initialize this API.

```
1  var express = require('express');
2  var mongoose = require('mongoose');
3
4  var app = express();
5  var router = express.Router();
6
7  var port = process.env.API_PORT || 3001;
8
9  router.get('/', function(req, res) {
10   res.json({ message: 'API Initialized!'});
11 });
12
13 //Use router configuration - localhost:3001/api
14 app.use('/api', router);
15
16 //starts the server and listens for requests
17 app.listen(port, function() {
18   console.log(`api running on port ${port}`);
19 });
```

Database service for this project will be used mLab, which is a fully managed cloud database service that hosts the MongoDB databases. MLab offers 0.5GB free space and so it is a good service to use. In order to connect the API with the created database, this code is sufficient. (It is necessary to create an account on a mLab webpage.[12])

```
1  var mongoDB = 'mongodb://OllieMomento:a123456@ds211289.mlab.com:11289/
       upp';
2  mongoose.connect(mongoDB, { useMongoClient: true })
3
4  var db = mongoose.connection;
5  db.on('error', console.error.bind(console, 'MongoDB connection error:'))
       ;
```

When APIs and databases are connected, it is possible to implement functions for creating, updating, and deleting the records. Two databases were created for the prototype. One database contained all projects and the other contained a list of people. In order to add a record, it is necessary to create a schema that will show what our database records look like. See the schemas in the source code in model directory.

[12] mLab. `https://mlab.com/` (visited on 05/02/2018).

6.5 Prototype navigation

The react-router library was used to navigate in the prototype, as mentioned in subsection 6.2.1. The main components are Router and Route. Where Router is the highest component of the whole react-router hierarchy. Router contains child's Routes, which indicates which component to render based on the URL. The Router component can have only one child, so if you want to have more Routes, you have to put them all in a div. To have a prototype organized, it's good to place the Router and Routes components in the main prototype file. Each component needs a path which is the URL and then a component that will be rendered when navigating to that path. See the code below that is used in the prototype.

```
1  class App extends Component {
2    render() {
3      return (
4          <Router history={history}>
5            <div>
6              <Route exact path="/"
7                     render={() => <HomepageS history={history} />} />
8
9              <Route exact path="/projects/:id"
10                    component={ProjectPage} />
11
12             <Route exact path="/projects/:id/shots/:shot"
13                    render={() => <ShotPage project={this.props.project
     }/>} />
14
15             <Route exact path="/projects/:id/asset/:asset"
16                    render={() => <AssetPage project={this.props.project
     }/>} />
17           </div>
18         </Router>
19     )
20   }
21 }
22 export default App;
```

In the code, we see the other features of the react-router library offers. Before the path, we see prop *exact* that indicates that the route should only match when the pathname matches the route's path exactly. As you can see, there are two ways to determine which component will be rendered. *Component* or *render* can be used. The advantage of using *render* is that using the arrow function it is possible to pass the props. Sometimes you need to keep the variables that are in the URL. For example, if there is a page with a particular project, the id of this project have to be in the URL. The other feature of react-router can do it. For example using *path = "/projects/:id"*, where after */projects/* can be any string. This string will then be stored in *match.params.id.*

If all routes are implemented, prototype needs a way to navigate between pages. The React-router offers a Link component that, after the click, updates the URL, and the rendered content will change without reloading the page.

```
1  class ProjectItem extends Component {
2      render() {
3          return (
```

```
4               <Link to={{ pathname: `/projects/${this.props.project._id}`,
5                state: {project:this.props.project}}}>
6                   <ListItem>
7                       //... removed for brevity
8                   </ListItem>
9               </Link>
10          );
11      }
12  }
13  export default ProjectItem;
```

6.6 Graph

As already mentioned in the subsection 6.2.1, the library mxGraph was used to create the graph. MxGraph is divided into several packages. One of the main packages is the *editor* that provides the classes required to implement a graph editor. The main class in the packages is mxEditor. Other packages are view and model that implement the graph component, represented by mxGraph. It refers to a mxGraphModel which contains mxCells and catches the state of the cells in the mxGraphView. The library also provides event handling and key bindings. Another of the packages is the IO (input-output) that implements a generic mxObjectCodec for turning JavaScript objects into XML. The main class is mxCodec.

To understand the interaction in the graph, it's important to know how the graph and the interacting elements are placed on the page. See the Figure 18.

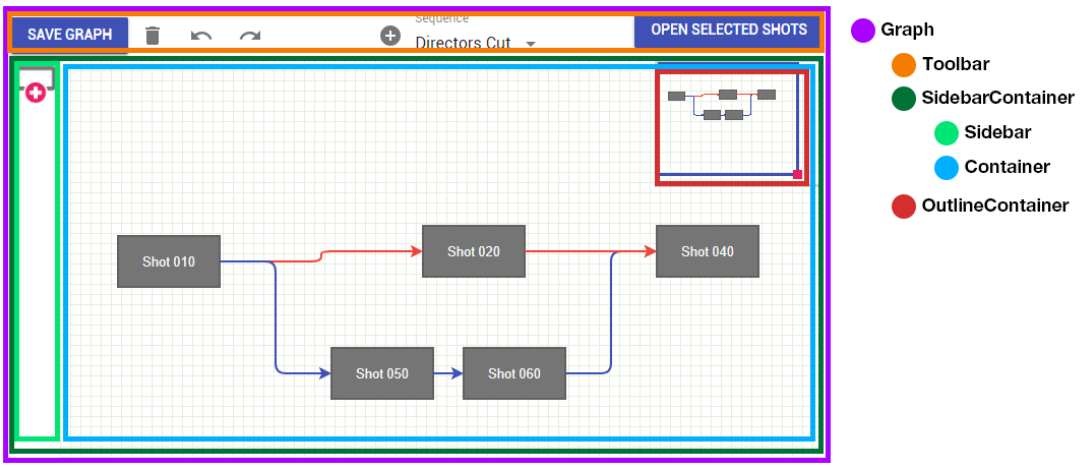


Figure 18 Layout of graph and elements for interaction

6.6.1 Interaction

This chapter describes the main interactions with the graph.

Creating vertex

The main feature of the graph is the creation of new shots and assets. Creating is done using drag and drop. The user clicks on the icon from the sidebar to create a new shot or asset and drags to the container where wants the elements to be placed. The creation of new vertices (shot or asset) is done in the mxGraph library using the *insertVertex* method, which needs to have a parent, id, value, x, y, width, height and style parameters.

When creating a new asset, a popup menu is displayed after the asset is placed, from which the user chooses a type of asset to insert. There is also a bar to filter these assets. This feature is implemented to speed up work with graph creation.

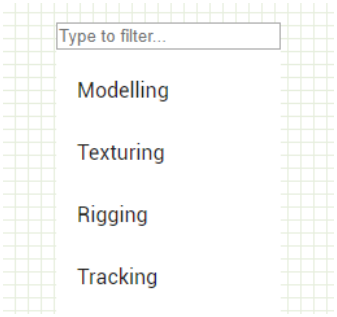


Figure 19 Pop-up menu

The pop-up menu is realized that when a drag and drop event is handled, the *popUpMenu()* method is called, which will open the window in the place of drop. Each list item has an *onClick()* method that assigns the selected asset and hides the popup menu. Below you can see a snippet of code.

```
1  //popUpSelect.js
2  <ListItem key={index} button onClick={() => this.props.setAsset(asset)}>
3      <ListItemText primary={asset} />
4  </ListItem>
5
6  //graph.js
7  popupMenu(evt, x, y) {
8          this.setState({ activePopUp: true })
9          this.setState({
10             clientCoord: {
11                 x: x,
12                 y: y
13             }
14         })
15
16         var clientX = evt.clientX
17         var clientY = evt.clientY
18
19         var popUp = document.getElementById("popUpMenu")
20         popUp.style.top = clientY + "px"
21         popUp.style.left = clientX + "px"
22 }
```

Creating edge

Creating an edge between nodes indicates either sequence identification (project page) or assets relationship (shot page). The implementation is the same for both cases. To create an edge between two nodes, use the *insertEdge* method from the mxGraph library. The method has parent, id, value, source, target and style parameters.

A hover icon in the middle of the element (shot or asset) is used to create an edge. Clicking on the icon and dragging to the next element creates an edge. If it is illogical to link these two elements, or if those elements are already linked, the user is notified and the edge is not added. In order to add this implementation logic to edge addition, it's possible to change the behavior of the *insertEdge* method by using *mxConnectionHandler.prototype.insertEdge*. See the snippet code from shot page below.

```
mxConnectionHandler.prototype.insertEdge = (parent, id, value, source, target, style) => {

                //... removed for brevity
                var flag = true;
                edges.every(edge => {

                    let cell = edge.firstChild
                    cell.source = cell.getAttribute("source")
                    cell.target = cell.getAttribute("target")

                    if (cell.source == source.id || cell.target == target.id) {
                        alert("Cannot connect")
                        flag = false
                        return false
                    }
                    return true
                })

                if (flag) {
                    var doc = mxUtils.createXmlDocument();
                    var edge = doc.createElement('Sequence')
                    edge.setAttribute('seq', color);
                    graph.insertEdge(parent, id, edge, source, target, 'strokeColor=' + color)
                }
            }
}
```

Editing vertex

In the prototype, asset editing is implemented using hover icons. When hovering on an asset, four icons appear at the corners of the asset. See the Figure 20. In the top right corner is the icon for deleting the asset, in the top left corner the icon to add or change assignee on the given asset. In the lower left corner, there is an icon to change the name of the asset, and a copy of the asset is implemented in the lower right corner.

A snippet of the code below that assigns a worker to an asset.

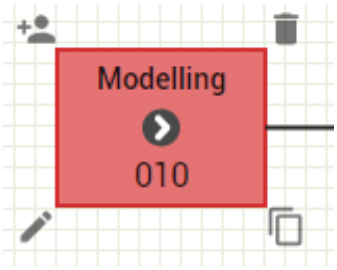


Figure 20 Asset hover icons

```
1  //hoverIcons.js
2  var img = mxUtils.createImage(AddAssignee);
3      img.setAttribute('title', 'Add Assignee');
4      img.style.position = 'absolute';
5      img.style.cursor = 'pointer';
6      img.style.width = '20px';
7      img.style.height = '20px';
8      img.style.left = (state.x - 20) + 'px';
9      img.style.top = (state.y - 20) + 'px';
10
11     mxEvent.addGestureListeners(img,
12         mxUtils.bind(this, function (evt) {
13             // Disables dragging the image
14             mxEvent.consume(evt);
15         })
16     );
17
18     mxEvent.addListener(img, 'click',
19         mxUtils.bind(this, function (evt) {
20             this.openWindow(state.cell)
21         })
22     );
23
24     state.view.graph.container.appendChild(img);
25     this.images.push(img);
```

Other interactions

The graph for the project page and for the shot page has implemented buttons for basic interaction with the graph. These buttons are located in the toolbar. Button for deleting selected elements (vertexes and also edges) and then the undo and redo button. To quickly interact with the graph, it is necessary to implement keyboard shortcuts. See the Table 2.

Table 2 Key bindings

Key combination	Funkcionality
Delete	Delete selected elements
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + X	Cut selected elements
Ctrl + C	Copy selected elenetes
Ctrl + V	Paste
+	Zoom in
-	Zoom out

Panning is implemented as click and hold the right mouse button and drag the graph to a new place. Use the right mouse button and drag to rubberband selection. Although it is implemented in the mxGraph library, it did not work. Therefore, it was re-implemented.

6.6.2 Edge layout

In some cases, it is possible to have more edges between the two vertices, so that these edges do not overlap, it is necessary to choose the correct layouting and allow the user to change the trajectory of the edge. The most appropriate is the use of a parallel layout, where the edges are orthogonal with a slight rounding. See the Figure 21.

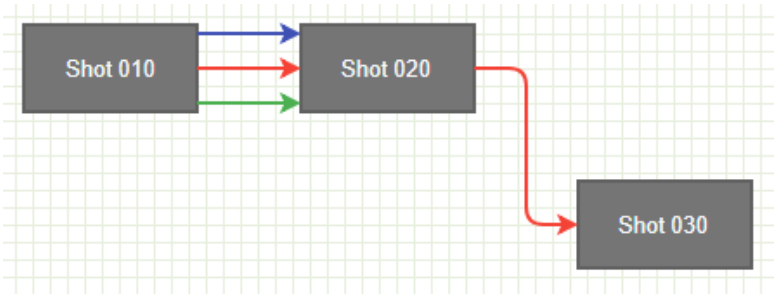


Figure 21 Edge layout

The mxGraph library supports this layouting. See below the implementation.

```
var layout = new mxParallelEdgeLayout(graph);
        var layoutMgr = new mxLayoutManager(graph);
        layoutMgr.getLayout = function (cell) {
            if (cell.getChildCount() > 0) {
                return layout;
            }
        };
        layoutMgr.executeLayout(layout, graph.getDefaultParent())
        layout.execute(graph.getDefaultParent());
```

7 User testing of Hi-Fi prototype

User testing of high fidelity (hereinafter referred to as Hi-Fi) prototype from the chapter 6 gives us a great opportunity to find out what we could not test with the Lo-Fi prototype. Test the interaction in the graph, which is the main function of the whole prototype. But not only that but also test the whole workflow, specific UI components, graphic elements, page hierarchy and legibility. The Hi-Fi prototype looks like software ready to use to users. So the participants behave realistically as if they were interacting with real software, with a sketchy prototype they may have unclear expectations about what is supposed to work and what is not.

7.1 Recruiting participants

Testing in a company like UPP is very challenging for recruiting participants. All employees are busy and the workload is changing from hour to hour. Therefore, it is not possible to arrange with them in advance time for testing. Despite the fact that during the testing period the UPP had a lot of unfinished projects, I managed to recruit a total of 5 participants, out of which 3 as supervisor and 2 as an artist, only 2 of them were tested in the previous iteration.

7.2 Test environment

The test environment was set up similar to the Lo-Fi prototype testing in section 4.3.1 i.e. the test was run on a 13.3-inch laptop with an attached mouse that didn't belong to the user. The localhost server, which served as a rest API for an application runs on another localhost port on the laptop. So the user tested the prototype in a browser window. The browser was Chrome version 65. The screen and sound capture provides OBS studio software and participants agreed with recording.

The testing took place in a quiet corner of an open office where the participant was not distracted by the surroundings, even though they had to be on standby if somebody was looking for them. While the participants were solving individual tasks, I sat next to them and observed their behavior while writing notes.

7.3 Testing process

As in the previous testing, each participant was familiar with the basic testing principles. In order for a participant not to have bad feelings in testing, it is important to emphasize

that he is not tested himself, but the system. It is not important to speed up the task but to understand the behavior using the technique called think aloud. If the participant believes that the task is over, it is important to say this and return to the homepage before each next task.

Prior to the test, the participant was asked for information on company position, age, and gender. See the in the subsection 7.4.1. The participant was acquainted within ten seconds with a prototype with basic functionality and clarification of the terminology used. Different terminology in previous iterations has led to many misunderstandings. In addition, the participant was aware that the system of registration is not implemented in the prototype and for the purposes of testing is the name of the participant Karel Novák. In order for a participant to see what events he did.

The test itself took approximately 20 minutes for the artist to complete four tasks, and for the supervisor 45 minutes to solve eight tasks. This is primarily because the supervisor can also represent the work of an artist. The testing tasks can be found in the subsection 7.4.2 and 7.4.3. After testing, the participant answered three closed questions on the Likert scale, and the four questions were open. In the open questions was also space for a possible discussion about future of the prototype. This was primarily done by the supervisors. See the post-test answers in the subsection 7.4.4.

7.4 Session guide

7.4.1 Pre-test questions

Gender:

Position:

Age:

7.4.2 Testing tasks - Supervisor

S1. Find out how many sequences do have a project called Google Inc.

S2. Create a new project called *Lenovo*, choose any Start and End date, don't assign supervisor and set description to *Laptop*, don't choose an image file.

After creating a project assign yourself - *Karel Novák* as a new supervisor.

S3. In the project *Lenovo* create two sequences of shots - Normal and Director's Cut.

First sequence (Normal) consists of four shots. Second sequence (Directors Cut) consists of five shots.

Shot 1,3 and 4 are the same as in normal sequence but instead of shot 2 place different shots 5 and 6. See the Figure 22. Names of shots are not important!

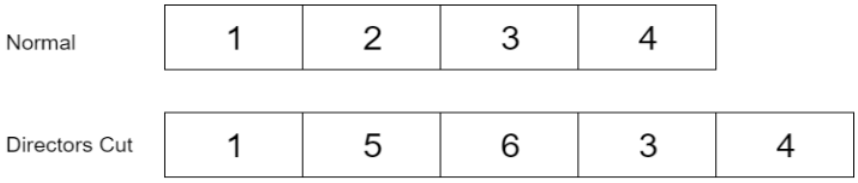


Figure 22 Shots in Normal and Directors Cut sequences

S4. Go to *Tesla* project you are assigned in. Place the same asset *Matte painting* to *Compositing Car* and *Composition Engine* in Shot 010 and Shot 020. Assign artist *Daniel Hůlka* to the *Matte painting* you have created.

S5. You are assigned to *Modelling Edge 7* in Project *Samsung* in Shot 010. The supervisor of the project commented your work, that you need to make the model with sharper edges and with a bigger main button. You created the new version of model considering the supervisor's comment - bigger main button. Add the new version with a description: *main button* and upload the file: *edge7.mb*

S6. You are a supervisor of project *Tesla*. In Shot 010 approve the last version of *Simulation Dust*.

S7. Find out what artists are working on *Garnier Fructis* project.

S8. Find out what comments have been implemented in *Matte painting Hills* asset in the *version2.psd* project *Yamaha*, Shot 010.

7.4.3 Testing tasks - Artist

A1. Find out how many sequences do have a project called Google Inc.

A2. You are assigned to *Modelling Edge 7* in Project *Samsung* in Shot 010. The supervisor of the project commented your work, that you need to make the model with sharper edges and with a bigger main button. You created the new version of model considering the supervisor's comment - bigger main button. Add the new version with a description: *main button* and upload the file: *edge7.mb*

A3. Find out what artists are working on *Garnier Fructis* project.

A4. Find out what comments have been implemented in *Matte painting Hills* asset in the *version2.psd* project *Yamaha*, Shot 010.

7.4.4 Post-test questions

For closed questions, the participant selected the answer from the likert scale.

1. Strongly Agree
2. Agree
3. Neutral
4. Disagree
5. Strongly Disagree

Closed questions

Using the app was comfortable. (ergonomics)
Using the app was flexible. (It will allow everything I need)
Using the app was efficient.

Opened questions

1. How would you rate the complexity of tasks?
2. Did you have any trouble with tasks?
3. What's your overall feeling of using?
4. Was the use of the prototype clear?

7.5 Test results

In the next section are the results that were found from single-participant testing. In the testing tasks, the results are marked with a question number from the subsection 7.4.3 for an artist role and in the subsection 7.4.2 for a supervisor.

7.5.1 Participant 1

Pre-test questions

Gender: *Male*
Position: *Artist*
Age: *27*

Testing tasks

A1. Participant opened Google Inc. project and using the select menu came to the number of sequences.

A2. He could not find a Samsung project for a while and did not even use the search bar. When he found the project, went to shot 010, but he thought he was wrong because he did not know he could open also the asset. He went back to the homepage and went on the same path to get back. When he came back to the page with Shot 010, he discovered that it is possible to open the asset by double-clicking. When adding a new version, he chose a comment that he implemented - sharper edges because he was confused by the name "edge7.mb" he did not know it is the name of the type of Samsung phone.

A3. This task did not make a problem for the participant. He knew intuitively that this information would be in Team tab.

A4. Since the participant already knew the hierarchy of the application, he went directly to the Yamaha project, Shot 010 and then to the Matte painting Hills asset. He clicked on the desired version. The system showed him which comments were implemented with this version. However, the system did not emphasize to the participant that the selected version is active. Despite this bug, the user has reached the goal.

Post-test questions

Closed questions

Using the app was comfortable. (ergonomics)
Strongly agree

Using the app was flexible. (It will allow everything I need)
Strongly agree

Using the app was efficient.
Agree

Opened questions

How would you rate the complexity of tasks?
Simple.

Did you have any trouble with tasks?
Improve visualization of changes or better confirmation.

What's your overall feeling of using?
This work is often changing and these changes need to be adjusted right now, so I think it's easier to tell it personally or call him instead of putting it into the app.

Was the use of the prototype clear?
I just did not know that the asset could be opened.

7.5.2 Participant 2

Pre-test questions

Gender: *Male*
Position: *Supervisor*
Age: *31*

Testing tasks

S1. Participant found in Google Inc. how many of the sequences are here, but he said he was not used to such an editing visualization and could not imagine the situation if there were more sequences. He would appreciate sequence filtering.

S2. The participant had no problem creating the project and wanted to know how to assign more supervisors.

S3. When creating sequences, it chose a similar color to the existing sequence and wanted to change it back - which is not possible in the prototype. After completing the task, he played for a while with a layout, and he liked the way he wanted to interact. He also asked how the situation would be treated if one shot would be the same as the other, only a few seconds longer.

S4. He used the search bar to find a project. First, he only opened Shot 010 and created an asset Matte Painting, which he assigned to the asset compositing Car. After saving the graph, he did not know how to assign it also for the Shot 020.

After a moment of searching, he noticed the "Open selected shots" button that he used to open both shots and assign it to the second asset of the compositing Engine.

S5. The participant opened the right asset, but to solve the problems, he used the possibility of the comments from the supervisor erased. That was not the right option. I wrote the comments again and the participant has already found the right way to solve the task.

S6. He fulfilled the task without any problems.

S7. The participant found the solution intuitively.

S8. The participant uses the overview window in the top right corner to navigate the graph and after finding the right asset he found the requested comments.

Post-test questions

Closed questions

Using the app was comfortable. (ergonomics)
Neutral

Using the app was flexible. (It will allow everything I need)
Disagree

Using the app was efficient.
Agree

Opened questions

How would you rate the complexity of tasks?
Simple.

Did you have any trouble with tasks?
No.

What's your overall feeling of using?
I was surprised, with visualization of the sequences, I have never seen this visualized. I would appreciate the possibility to see all assets on the project and filter the sequences.

Was the use of the prototype clear?
At first I did not understand how to work with sequences.

7.5.3 Participant 3

Pre-test questions

Gender: *Male*
Position: *Supervisor*
Age: *38*

Testing tasks

S1. The participant mentions that the term edits is rather used instead of the sequence. The participant opens the project but does not know where to find the number of sequences the project has. He looks for this information in the info tab. Then he realized that the sequences are visualized by a color of edges. He is dissatisfied with this visualization, and after being shown that the list he could see in the select menu is satisfied. After a moment of thinking, he comes to the conclusion that the sequence can be in the extreme case over a hundred and he would not want to count on it. Additionally, it is common for sequence names to be very long and therefore very unclear.

S2. The participant asks what is a start date and end date and that this information is unnecessary, that it is more important to have the start date and end date for the individual sequences. The participant chooses the supervisor and looks for the save button.

S3. The participant mentions that editing is often sent directly from the editor. The editing that they would create in UPP is done only with full CGI advertisements. Finally, the participant creates the first sequence and, when creating the second sequence, forgets to link the shot 030 and the shot 040. He would appreciate the possibility of assigning priority to each sequence.

S4. The participant assigned the matte painting asset to one asset compositing car, not knowing how to assign the same asset to the second asset compositing engine. After a while, I had to guide the participant and show him the open selected shots button. After the demonstration, the participant finished the task.

S5. The participant first searches for the Team tab on the shot page. Then he tries other options when he gets to the right page, he completes the task.

S6. He fulfilled the task without any problems.

S7. He fulfilled the task without any problems.

S8. The participant will find the requested comments. But he asks if these comments can be prioritized or visualized which ones are from the director, agency or recommendation from UPP.

Post-test questions

Closed questions

Using the app was comfortable. (ergonomics)
Agree

Using the app was flexible. (It will allow everything I need)
Strongly disagree

Using the app was efficient.
Strongly agree

Opened questions

How would you rate the complexity of tasks?
I did not find it complicated.

Did you have any trouble with tasks?
No.

What's your overall feeling of using?
I like the design is minimalist and it is user-friendly.

Was the use of the prototype clear?
Yes, just resolve a better zoom in the graph.

7.5.4 Participant 4

Pre-test questions

Gender: *Male*
Position: *Artist*
Age: *31*

Testing tasks

A1. The participant searched for information in the info tab at first. Then he clicked on the select menu and found the information. Asking why he did not know so long, he said he did not know where else to find the information. So he tried to click on everything.

A2. At the beginning, the participant did not know how to work with the graph and that it is possible to open a shot and also an asset by double-clicking. He was confused by the hover button in the center of the cell. The participant was pleased with how to choose which work was implemented using checkboxes. But he did not know that these comments were like a communication channel, or rather the assignment of partial works.

A3. The participant found this information without any problems.

A4. He first searched for this information in the info tab in his asset, but then he quickly found out where the information is. He was only confused that he did not know that the version was selected if it was already approved.

Post-test questions

Closed questions

Using the app was comfortable. (ergonomics)
Neutral

Using the app was flexible. (It will allow everything I need)
Neutral

Using the app was efficient.
Agree

Opened questions

How would you rate the complexity of tasks?
Simple.

Did you have any trouble with tasks?
No.

What's your overall feeling of using?
The system is easy to use and I like the simplicity of the interface, but it is inadequate in what I need. Comments are often visual and the text itself is not enough.

Was the use of the prototype clear?
I did not know why to click on the select menu in the sequences in the first task.

7.5.5 Participant 5

Pre-test questions

Gender: *Male*
Position: *Supervisor*
Age: *33*

Testing tasks

S1. The participant has figured out how many sequences it is by counting how many different color edges are in the graph.

S2. The participant finished the task without any problems.

S3. The participant pointed out that he has to create two sequences in the task, but one of them is already created and it confused him. For a while, he played with a layout and praised the graph interaction. When creating the second sequence, he forgets the last edge between the shot 030 and the shot 040.

S4. First he did not open the two shots at once but went only to shot 010. He realized that he had to assign both, so he came back and opened both shots using the open selected graph button.

S5. The participant proceeded systematically without any problems.

S6. The participant fulfilled the task without any problems.

S7. The participant fulfilled the task without any problems.

S8. The participant fulfilled the task without any problems.

Post-test questions

Closed questions

Using the app was comfortable. (ergonomics)
Agree

Using the app was flexible. (It will allow everything I need)
Strongly disagree

Using the app was efficient.
Agree

Opened questions

How would you rate the complexity of tasks?
Simple.

Did you have any trouble with tasks?
No.

What's your overall feeling of using?
I like the application, the usage is clear. I'd like to be reminded that the graph has been changed and I have to save the graph. I have to remember in which shots the assets are.

Was the use of the prototype clear?
Yes, the application does what it should do. But it does not do what I want to do.

7.6 Findings and recommendations

The list of findings and their recommendations from the testing tasks can be found in the Table 3. Tasks are labeled with the same designation as in subsection 7.4.2 and 7.4.3. For the list of findings and their recommendations from the post-test questions, see the Table 4 and the question number corresponds to the number of opened questions in subsection 7.4.4. Recommendations for problems will be implemented in the next iteration in future development. Each problem found has a priority that matches the following list.

Priority:

1. Cosmetic problem
2. Less usability problem, remove with lower priority
3. More serious problem of usability, important to remove
4. Critical usability problem, need to be removed

Other questions the participants had answered after the test were three more closed questions where they had a choice of answers using the Likert scale. These questions asked about comfort, flexibility, and efficiency. See in the subsection 7.4.4. These responses were visualized in the Figure 23.

Table 3 Findings and recommendations from testing tasks

Task number	Problem	Recommendation	Priority
A2	Lack of using the search bar	Change the background color of the search bar to primary	1
A2	Not knowing that asset cell is double clickable	Change the position of hover icon to adding a new edge	1
S8, A4	Approved version in asset page does not show hover and select action	After hover on approved version change the background color and also when the version is selected	3
S1, A1	Visualisation of sequences with large quantities (more than 50) of sequences (often with long names)	Add a list of sequences visualized by a table and add filtering method to see only selected ones	3
S2	Assigning more supervisors	Add possibility to add more supervisors with select menu and chips	2
S3	Editing sequence color and name	Create editing window for sequences	4
S4	Opening more than one shots	Change the position of "open selected shots" button instead of "save graph" button	1
S2	Sequence creating and editing (often comes from editor)	Add function to import sequences from other software	2
S2	Assigning the last shot to the second sequence	When leaving the page, ask the user if they have not forgotten to assign the last shot.	2
S8, A4	Prioritization of comments on a asset page	Possibility to assign the label to comment director's, agency's, client's and supervisor's recommendation.	3

Table 4 Findings and recommendations from post-test questions

Question number	Problem	Recommendation	Priority
2	Visualization of the last changed state	Add a footer bar with last changed state	2
3	Do not have to remember in which shot is which asset.	Add a page with all assets on the project and allow their filtering.	3
3	Visualize the graph more clearly if there are many cells	Implement semantic zoom and grouping option	3
3	Comments are often visual and the text itself is not enough	Implement an option to add an image to a comment.	2
3	Alert users if the graph has changed to save it.	Highlight the save graph button if any changes are made.	1

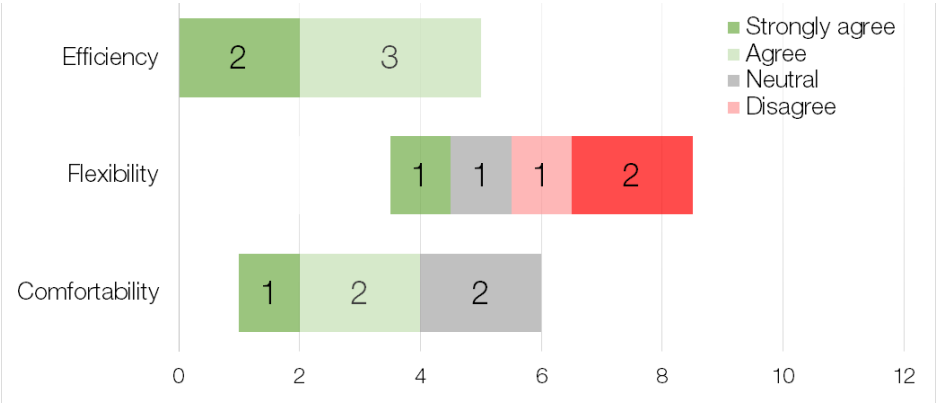


Figure 23 Participants feeling with the prototype

It is clear from the Figure 23 that the prototype was very efficient and comfortable. On the contrary, the prototype has little flexibility, so it does not do what the participants want. As already mentioned in the tables above. Participants would welcome many more choices and features. This is primarily a preview of all assets in the project and enhanced filtering of sequences and assets. They also would like to have the priorities and labels for comments. Additionally, supervisors would welcome a more robust design just for extreme cases that did not happen in the company or only occurred a few times.

They were particularly grateful for the simplicity, clarity, and intuitiveness of the prototype. They also liked the interaction with the graph when some of the features they know from the software they use daily were implemented. The graph allowed them to precisely align the cells with the others. They also appreciated the creation of an asset where, after drag and drop, the system will offer them a list of the types they can use.

8 Conclusion

I chose to design and implement a prototype web application as the main goal for my diploma thesis. This prototype allows planning of task on particular projects for visual effects and post-production company. In order to collect user requirements, I used the qualitative methods of user research directly on the staff of the biggest trick studio in Central Europe UPP.

I was aware that in a one-person team when I'm doing all the development process myself, a complex system can not be created. Even when it comes to collecting requirements when the understanding of the whole post-production process is very challenging, I assumed that the system will become complex. I analyzed the appropriate visualizations that deal with this issue and created a low-fidelity prototype that visualizes the post-production process using both tables and graphs. People from UPP are used to work with graphs. From user testing, it turned out that this double approach is confusing, and so only graph visualization was realized.

I analyzed the appropriate technologies for creating a high-fidelity prototype and found that choosing ReactJS is a suitable framework. When implementing the prototype, there was no situation where I would encounter a framework problem that would restrict or delay me in the development. The mxGraph graph creation library also seems to be a good choice, although I had to implement a lot of functionality on my own. After completing the high-fidelity prototype, I conducted user testing with UPP employees again. The test results showed that the participants were performing efficiently and the using of the prototype was comfortable. On the contrary, they have come to the conclusion that the system does not allow functionalities that would completely replace the existing system. However, data visualization using the graph appeared to be appropriate and the participants quickly became accustomed to it.

8.1 Future development

In order to use the system and completely replace the existing solution, it is necessary to add missing functionality. It is especially the possibility of interacting with a graph with a large number of vertices (more than 50). Add an image to comment and prioritize these comments. Additionally, it is necessary to allow editing of sequences. Another important feature is the possibility to list all assets in a project so the user does not have to know in which shot asset is. In order for the system to be complete, it is necessary to implement an authentication and authorization system and connect the system with UPP database and notification application.

Bibliography

[1] Alan Cooper, Robert Reimann, and Dave Cronin. *About Face 3: The Essentials of Interaction Design*. New York, NY, USA: John Wiley & Sons, Inc., 2007. ISBN: 9780470084113.

[2] *HCD. Design zaměřený na člověka : soubor nástrojů*. Brno: Flow, 2013. ISBN: 978–80–905480–1–5.

[3] Donald A. Norman. *The Design of Everyday Things*. New York, NY, USA: Basic Books, Inc., 2002. ISBN: 9780465067107.

[4] Aybüke Aurum and Claes Wohlin. *Engineering and Managing Software Requirements*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN: 3540250433.

[5] Pixton Comics Inc. *Pixton Comic Maker*. URL: `http://www.pixton.com/` (visited on 01/10/2018).

[6] Ivan Herman, Guy Melançon, and M. Scott Marshall. "Graph Visualization and Navigation in Information Visualization: A Survey". In: *IEEE Transactions on Visualization and Computer Graphics* 6.1 (Jan. 2000), pp. 24–43. ISSN: 1077-2626. DOI: `10.1109/2945.841119`. URL: `http://dx.doi.org/10.1109/2945.841119`.

[7] Raga'ad M. Tarawaneh, Patric Keller, and Achim Ebert. "A General Introduction To Graph Visualization Techniques". In: *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011*. Ed. by Christoph Garth, Ariane Middel, and Hans Hagen. Vol. 27. OpenAccess Series in Informatics (OASIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, pp. 151–164. ISBN: 978-3-939897-46-0. DOI: `10.4230/OASIcs.VLUDS.2011.151`. URL: `http://drops.dagstuhl.de/opus/volltexte/2012/3748`.

[8] Manojit Sarkar and Marc H. Brown. "Graphical Fisheye Views of Graphs". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '92. Monterey, California, USA: ACM, 1992, pp. 83–91. ISBN: 0-89791-513-5. DOI: `10.1145/142750.142763`. URL: `http://doi.acm.org/10.1145/142750.142763`.

[9] Muzammil Khan and Sarwar Shah Khan. "Article: Data and Information Visualization Methods, and Interactive Mechanisms: A Survey". In: *International Journal of Computer Applications* 34.1 (Nov. 2011). Full text available, pp. 1–14.

[10] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. "A Review of Overview+Detail, Zooming, and Focus+Context Interfaces". In: *ACM Comput. Surv.* 41.1 (Jan. 2009), 2:1–2:31. ISSN: 0360-0300. DOI: `10.1145/1456650.1456652`. URL: `http://doi.acm.org/10.1145/1456650.1456652`.

[11] *Balsamiq*. URL: `https://balsamiq.com/` (visited on 01/10/2018).

[12] Samer Buna. *Yes, React is taking over front-end development. The question is why.* Mar. 2017. URL: https://medium.freecodecamp.org/yes-react-is-taking-over-front-end-development-the-question-is-why-40837af8ab76 (visited on 04/14/2018).

[13] *7 Top ReactJS Features Which Makes It Best For Development.* Feb. 2018. URL: https://www.angularminds.com/blog/article/7-top-reactjs-features-which-makes-it-best-for-development.html (visited on 04/14/2018).

[14] *10 JavaScript libraries to draw your own diagrams (2018 edition).* Apr. 2018. URL: https://modeling-languages.com/javascript-drawing-libraries-diagrams/ (visited on 04/14/2018).

[15] Craig Buckler. *SQL vs NoSQL: The Differences.* May 2017. URL: https://www.sitepoint.com/sql-vs-nosql-differences/ (visited on 04/14/2018).

[16] *Thinking in React - React.* URL: https://reactjs.org/docs/thinking-in-react.html (visited on 04/17/2018).

Appendix A

User testing of Lo-Fi prototype

A.1 Session guide - Supervisor

Pre-test questions:

Gender:
Position:
Age:
Software:

Has it ever happened to you that some of your workers worked on a different version than they should? (how often?)

I am often informed about the project phase of my project.
Strongly disagree | Disagree | Neutral | Agree | Strongly agree

Testing tasks

S1. Find out who is already assigned to project *Honda Motors*.

S2. Find out local path to *Shot 010* for Project *Honda Motors* and sequence *short version*.

S3. Look at the pipeline for the whole sequence and change assignee only to David Němec for asset *Modelling Dragon* (Project *Honda Motors* and sequence *short version*).

S4. Create new project called *Skoda Auto* and new sequence *Directors Cut* with 3 shots.

S5. Find out the local path to *dragon_final02.mb* and approve this version (Project *Honda Motors*, sequence *short version*, *Shot 010*, asset *Modelling Dragon*).

S6. Find out who is assigned to asset *Modelling Dragon*. (Project *Honda motors*, sequence *short version*, *Shot 010*).

S7. What comment(s) has been included in the version *dragon_final01.mb* (Project *Honda Motors*, sequence *short version*, *Shot 010*, asset *Modelling Dragon*).

S8. Edit pipeline of *Shot 010* for sequence *short version* and project *Honda Motors*. Change there status for *Modelling Dragon* from Done to In Progress.

S9. Add a new version of asset *Modelling Dragon* (Project *Honda Motors*, sequence *short version*, *Shot 010*) in the new version you solved the bigger teeth.

Post-test questions

How would you rate the complexity of tasks?

Did you have any trouble with tasks?

What's your overall feeling of using?

Did you find the use of a prototype clear?

A.2 Session guide - Artist

Pre-test questions:

Gender:
Position:
Age:
Software:

Has it ever happened to you that you worked on a different version than you should? (how often?)

Testing tasks

A1. Find out who is the supervisor of the project *Honda Motors*.

A2. Find out local path to *Shot 010* for Project *Honda Motors* and sequence *short version*.

A3. Look at the pipeline for the whole sequence *short version* for Project *Honda Motors*.

A4. Add a new version of asset you are working on *Modelling Dragon*. In the new version you solved the bigger teeth. (Project *Honda motors*, sequence *short version*, *Shot 010*).

A5. What comment(s) has been included in the version *dragon_final01.mb* (Project *Honda Motors*, sequence *short version*, *Shot 010*, asset *Modelling Dragon*).

Post-test questions

How would you rate the complexity of tasks?

Did you have any trouble with tasks?

What's your overall feeling of using?

Did you find the use of a prototype clear?

Appendix B

Test results - Lo-Fi prototype

B.1 Participant 1

Pre-test questions:

Gender: *Male*
Position: *30*
Age: *Supervisor*
Software: *Maya, Houdini, Softimage, Muke, Photoshop*

Has it ever happened to you that some of your workers worked on a different version than they should? (how often?)
Once a week.

I am often informed about the project phase of my project.
Agree, but not in detail.

Testing tasks

Look at the pipeline for the whole sequence and change assignee only to David Němec for asset Modelling Dragon (Project Honda Motors and sequence short version).
Wanted to click on the whole row to open, not on the button "open" on the project page. He found it, changed the assignee and he was looking for a button to confirm it.

Find out the local path to dragon_final02.mb and approve this version (Project Honda Motors, sequence short version, Shot 010, asset Modelling Dragon).
He used the search box to find the asset. In the asset page, he didn't know what version was approved.

Find out a local path to Shot 010 for Project Honda Motors and sequence short version.
He found the path but explained that in reality, there is no such path to the shot.

What comment(s) has been included in the version dragon_final01.mb (Project Honda Motors, sequence short version, Shot 010, asset Modelling Dragon).
In the system, the participant was oriented well and he found the page with comments and version, but he didn't understand the placement of comments, and what the comment panels mean.

Edit pipeline of Shot 010 for sequence short version and project Honda Motors. Change there status for Modelling Dragon from Done to In Progress.
He found the pipeline really well. He changed the status and he was looking for the confirmation button. He was confused with the button called "cancel edit".

Create a new project called Skoda Auto and new sequence Directors Cut with 3 shots.
This task went well, I just noticed that he originally wanted to click the cancel button instead of the create button.

Add a new version of asset Modelling Dragon (Project Honda Motors, sequence short version, Shot 010) in the new version you solved the bigger teeth.
There were no problems or hesitation in this task.

Find out who is assigned to asset Modelling Dragon. (Project Honda motors, sequence short version, Shot 010).
Participant went to pipeline instantly. (After an additional inquiry, he explained that he had gone there because he had been there once and noticed it before).

Find out who is already assigned to project Honda Motors.
He found it without problems.

Post-test questions:

How would you rate the complexity of tasks?
Easy

Did you have any trouble with tasks?
I did not understand the comments, I missed the overall view of the project and confirming edits.

What's your overall feeling of using?
I missed the overall view, so I thought it was just a clicking.

Did you find the use of a prototype clear?
For the first time, I was confused, but I quickly became accustomed, I would just mention the comments to the asset.

B.2 Participant 2

Pre-test questions:

Gender: *Male*
Position: *40*
Age: *Supervisor*
Software: *Maya*

Has it ever happened to you that some of your workers worked on a different version than they should? (how often?) *Not so often, once out of three projects.*

I am often informed about the project phase of my project.
Strongly agree

Testing tasks

Find out the local path to dragon_final02.mb and approve this version (Project Honda Motors, sequence short version, Shot 010, asset Modelling Dragon).
He first went to shot "030" because he was confused by the description. When he realized that, he was looking for a path to a file dragon_final02.mb in assets.

Find out who is already assigned to project Honda Motors.
Didnt notice the team tab. When he found it on the page, he thought it wasn't clickable.

Edit pipeline of Shot 010 for sequence short version and project Honda Motors. Change there status for Modelling Dragon from Done to In Progress.
He searched for a status change button in not edit mode.

Find out a local path to Shot 010 for Project Honda Motors and sequence short version.
Done this task without problems.

Find out who is assigned to asset Modelling Dragon. (Project Honda motors, sequence short version, Shot 010).
Done this task without problems.

Look at the pipeline for the whole sequence and change assignee only to David Němec for asset Modelling Dragon (Project Honda Motors and sequence short version).
He changed assignee in editing mode. He remembered it from the previous task.

Add a new version of asset Modelling Dragon (Project Honda Motors, sequence short version, Shot 010) in the new version you solved the bigger teeth.
Done this task without problems.

What comment(s) has been included in the version dragon_final01.mb (Project Honda Motors, sequence short version, Shot 010, asset Modelling Dragon).
He did not understand the comments panels, he thought he found a comment, but it was a comment for the whole asset.

Create a new project called Skoda Auto and new sequence Directors Cut with 3 shots.
Done this task without problems

Post-test questions:

How would you rate the complexity of tasks?
Simple

Did you have any trouble with tasks?
I didn't understand the interface of the comments.

What's your overall feeling of using?
I didn't think that I was effective.

Did you find the use of a prototype clear?
Yes, except for comments.

B.3 Participant 3

Pre-test questions:

Gender: *Male*
Position: *40*
Age: *Artist*
Software: *Nuke, Flare, Photoshop*

Has it ever happened to you that you worked on a different version than you should? (how often?)
Yes, twice for the project.

Testing tasks

What comment(s) has been included in the version dragon_final01.mb (Project Honda Motors, sequence short version, Shot 01", asset Modelling Dragon)
He clicked the line and waited for the opening, found the requested comment, but was nervous about having duplicate comments on the page.

Find out a local path to Shot 010 for Project Honda Motors and sequence short version. *Done this task without problems.*

Find out who is the supervisor of the project Honda Motors.
Done this task without problems.

Add a new version of asset you are working on Modelling Dragon. In the new version, you solved the bigger teeth. (Project Honda motors, sequence short version, Shot 010).
He immediately understood what the checkboxes do. He completed the task without problems.

Look at the pipeline for the whole sequence short version for Project Honda Motors.
He looked at the pipeline of shot "010" and thought he had accomplished the task.

Post-test questions:

How would you rate the complexity of tasks?
Simple

Did you have any trouble with tasks?
I didn't know the interface at first. After completing a few tasks, he was already orientated.

What's your overall feeling of using?
The overall feeling is good.

Did you find the use of a prototype clear?
I did not orientate myself in the interface.

B.4 Participant 4

Pre-test questions:

Gender: *Male*
Position: *32*
Age: *Artist*
Software: *Photoshop*

Has it ever happened to you that you worked on a different version than you should? (how often?)
No, my work as a matte painter is not dependent on previous work.

Testing tasks

Find out a local path to Shot 010 for Project Honda Motors and sequence short version. *He automatically went to the search box.*

Find out who is the supervisor of the project Honda Motors.
Done this task without problems.

What comment(s) has been included in the version dragon_final01.mb (Project Honda Motors, sequence short version, Shot 01", asset Modelling Dragon)
First he was looking in the project's comments. After the alert, he has already found the comment. He read the headlines in the comments panels and understood what the comments meant.

Look at the pipeline for the whole sequence short version for Project Honda Motors.
He did not see the button for pipeline sequence. During the questioning he said he was looking for him on the left side of the bar.

Add a new version of asset you are working on Modelling Dragon. In the new version, you solved the bigger teeth. (Project Honda motors, sequence short version, Shot 010).
Done this task without problems.

Post-test questions:

How would you rate the complexity of tasks?
Easy, no big problems.

Did you have any trouble with tasks?
I did not understand why the assets are on the homepage.

What's your overall feeling of using?
Good, I was easy to navigate in the system.

Did you find the use of a prototype clear?
Yes.

B.5 Participant 5

Pre-test questions:

Gender: *Male*
Position: *33*
Age: *Supervisor*
Software: *Maya, Houdini*

Has it ever happened to you that some of your workers worked on a different version than they should? (how often?) *On every larger project.*

I am often informed about the project phase of my project.
Strongly agree

Testing tasks

Find out who is already assigned to project Honda Motors.
Went to the project and back again. After a moment he returned and the completed task.

Find out a local path to Shot 010 for Project Honda Motors and sequence short version.
He found the local path, but he mentioned that this route is completely useless because it does not exist in reality.

Look at the pipeline for the whole sequence and change assignee only to David Němec for asset Modelling Dragon (Project Honda Motors and sequence short version).
The participant did not go into the sequence pipeline but into the shot pipeline, but he changed the assignee correctly.

Create a new project called Skoda Auto and new sequence Directors Cut with 3 shots.
Done this task without problems.

Find out the local path to dragon_final02.mb and approve this version (Project Honda Motors, sequence short version, Shot 010, asset Modelling Dragon).
Done this task without problems.

Find out who is assigned to asset Modelling Dragon. (Project Honda motors, sequence short version, Shot 010).
He went to the shot pipeline to see the assignee. He remembered it from his previous task.

What comment(s) has been included in the version dragon_final01.mb (Project

Honda Motors, sequence short version, shot "010", asset "Modelling Dragon")
Done this task without problems. He mentioned that naming is not appropriate. He recommended naming: pending, done, implemented.

Edit pipeline of Shot 010 for sequence short version and project Honda Motors. Change there status for Modelling Dragon from Done to In Progress.
At first attempt, he tried to find it in not editing mode. Upon entering the editing mode, he changed the status but was looking for a button to confirm the changes.

Add a new version of asset Modelling Dragon (Project Honda Motors, sequence short version, Shot 010) in the new version you solved the bigger teeth.
Done this task without problems.

Post-test questions:

How would you rate the complexity of tasks?
Confused tasks, but simple.

Did you have any trouble with tasks?
Not really.

What's your overall feeling of using?
Pretty good, sometimes needs to click a lot.

Did you find the use of a prototype clear?
Relatively yes.

Appendix C

Screenshots

The following images show the appearance of the user interface on the main four screens.

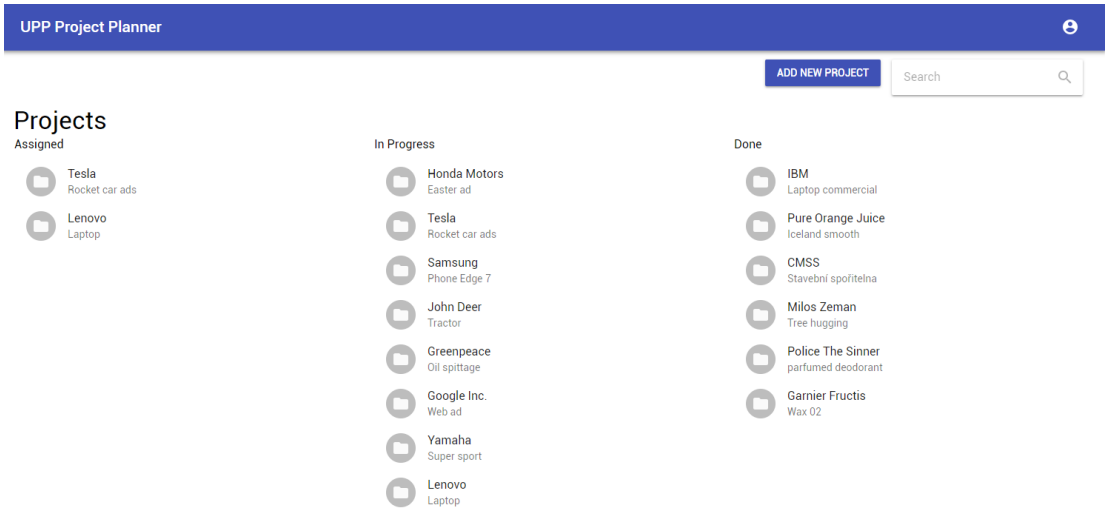


Figure 24 Homepage screen

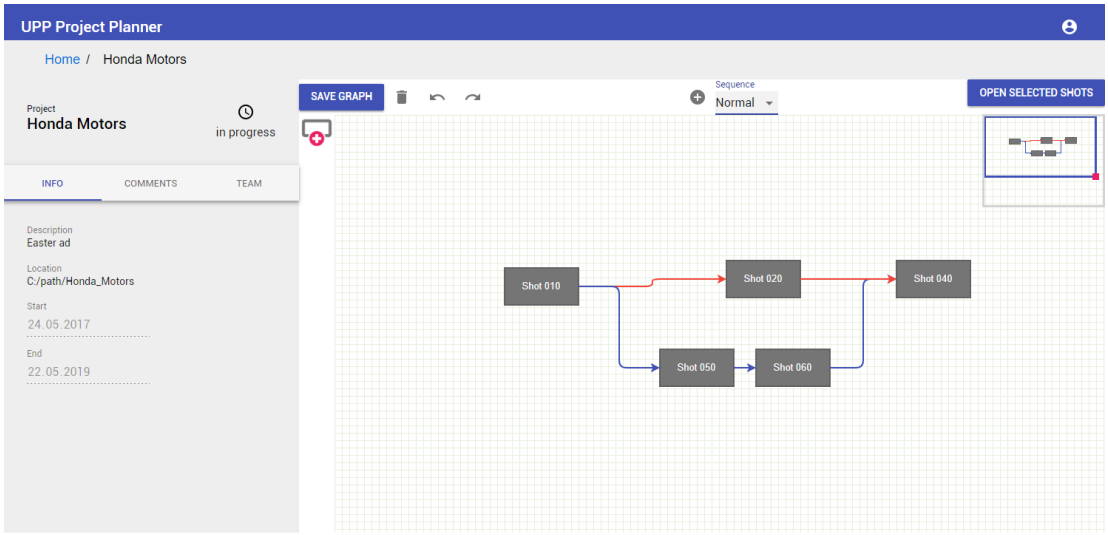


Figure 25 Projectpage screen

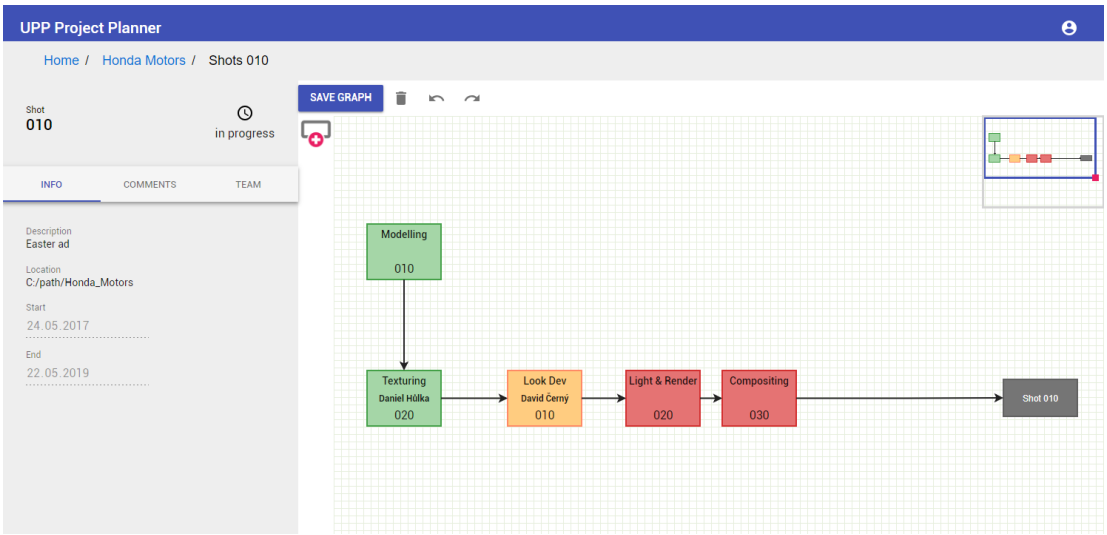


Figure 26 Shotpage screen

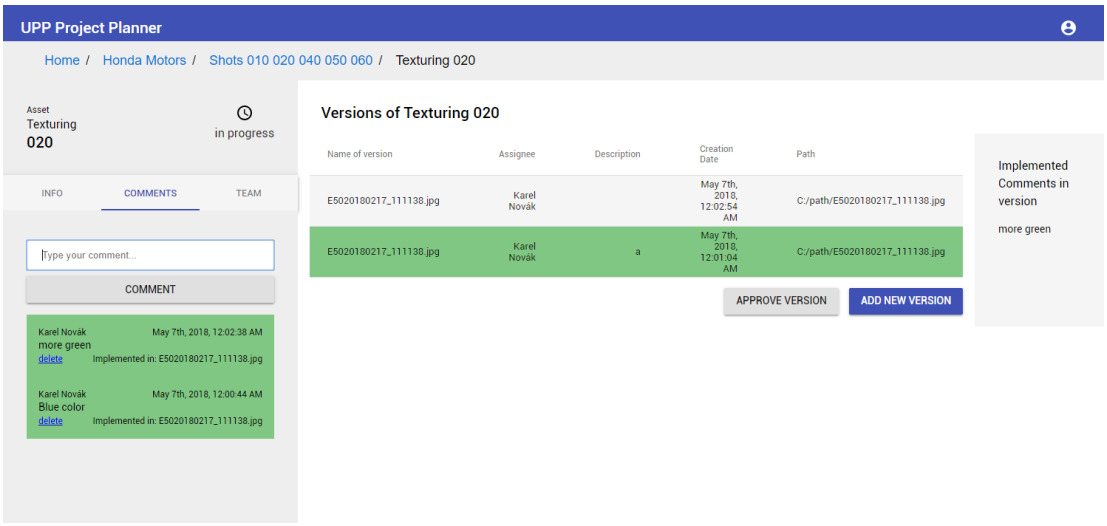


Figure 27 Assetpage screen

Appendix D

Contents of enclosed ZIP file

```
/
├── Storyboards/
│   ├── sb1.png
│   └── sb2.png
├── Lo-Fi prototypes/
│   ├── artist_lofi.bmpr
│   └── supervisor_lofi.bmpr
├── usecase_diagram.png
├── uppweb/ ..... Source code of Hi-Fi prototype
│   ├── model/
│   ├── public/
│   ├── src/
│   ├── package.json
│   ├── package-lock.json
│   ├── Procfile
│   └── README.md
├── MT-chvilond-latex-2018/ ..... Master thesis in LaTeXversion
└── MT-chvilond-2018.pdf ..... Master thesis in PDF version
```