UNIVERSITETET I AGDER

MM200 – Application Development

REPORT FINAL PROJECT

2017

Arnaud Soler
Jin-Ho Park
Henning Holms

Table of Contents

I – Introduction

We all have used a presentation tool in our life. There are several different types to choose between. Some powerful and complex such as Keynote, PowerPoint, Prezi, but there are also tool less complicated, such as Haiku Deck which are more focused on the task of presenting.

In this project our goal was to make a presentation tool for the web, where users are able to to create their own accounts, create a presentations, edit contents, save, import and export. They are also supposed to be able to personalize the content of their presentations.

II – Features

a) User account

One of the main functionality of this application was the ability to create their own user account, where they should be able to store and retrieve their files. They should be able to do this with the classical sign-up procedure, by choosing a username, enter their full name, email, make a password and upload an avatar.

After the creation of the account, all data will be stored in database and the users will be redirected to their dashboard, which displays their data and all the presentations that they made, further being able to create a new one. The user will have the option to delete their account as well.

b) Presentation and layout

This part was about creating a new presentation. We used our experiences from other presentation applications to discuss the way of how we should do about the interface. We wanted a main page with navigation bars, editing buttons and miniature slides distributed in

a clean and user-friendly way. And in center, the current slide the user is working on. This would have different options of style editing, adding colours and images.

When the user finished the presentation editing, they should be able to save the presentation, by storing it in the database. The user should also be able to import and export his presentation in a specific format such as .txt.

c) Different tools and functionallities

There are almost unlimited features that can be applied to a presentation app. We had many ideas, and after throwing out a bunch of features, but we decided to start with the most necessary features. The user must be able to create text boxes and edit inside contents. Here they can change size of the frame and move it to desired position. The user should have options to change font size, font colour, and text contents. The ability to add images to current slide was also one of the top priority.

Another essential feature was to separate the content of the presentation into multiple slides and further navigate through, with both clicking and arrow keys. We also wanted functionality to give the same background or so called theme, to the whole collection of slides, by selecting an image file. Being able to store the whole presentation information was a feature we wanted.

We also realized that the ability to delete different elements was essential. After editing is finished, the user should be able present it in full screen mode, and return to editing mode by pressing Esc key.

III – Challenges and Solutions

Client Side :

As we started working on the presentation part we spent a little while deciding whether to use the html canvas element, iframe or just a div. Henning had some experience with creating canvas animations, he spent some time trying to make this work, but as soon as trying to implement text editing features, it became clear that this was not the way to go.

Further we had seen some examples of how iframe might be a solution. As we could create separated documents for each slide, that could be stored without too much trouble. But this also brought with it some challenges of connect the coordinates. We ended up choosing the div element as editor frame.

After that, we need to save the box information and we knew that it is very important to discuss carefully about data structure of slide information but it was too late to discuss about it. The reason why it is very important is that because we will have to store image files either on server or database. we assumed that the best solution is storing only text information on database and save image files on server. Regardless of that importance, we ended up using database for storing all the presentation information and instead put one presentation information in one string, we divided them into several strings like one slide for one string. Retrieving part was an easy part because it is totally depends on how we store the presentation information. Next, we worked on making a function for making different slides but here we had a problem that we should decide how to connect left-slide(small, navigating) and middle-slide(big, center). We decided to connect them with the order of presentation. For example, the first left-slide will be connected to first middle-slide.

After that, we moved to presentation mode part. Even though we initiate the div element(text boxes, image files) with percentage position value, if we change(drag, resize) the position and replace them, their value will be changed from percentage to pixel value. So before we maximize the middle-slide in full-screen mode(presentation mode), we had to replace the pixel value with percentage value. This solution worked fine, but there was a huge challenge when we tried to implement our members system to this function.

Because one of our member used grid system and it has trouble with full screen mode and could not solve it perfectly. Except that, there are a lot of small and big issues with working together, task dividing, and time issues etc.

Security :

One potential solution was to encode the text in base64 and decode it during the importation, but even if it's encoded in base64 a good hacker knows how to encode malicious code in base64 and then import the file, so it wouldn't change anything.
The problem here was to create a file with another extension than .txt because it's insecure, for instance a malicious user could try to put some dangerous code in the txt file and import it to the application. We didn't find a proper solution to avoid this insecurity.

a) Database

According to the conceptual data model, the database contains 3 tables:

- Account
- Presentation
- Slide

We found that it was enough and we didn't need more table. Otherwise it would make the database more difficult to use.

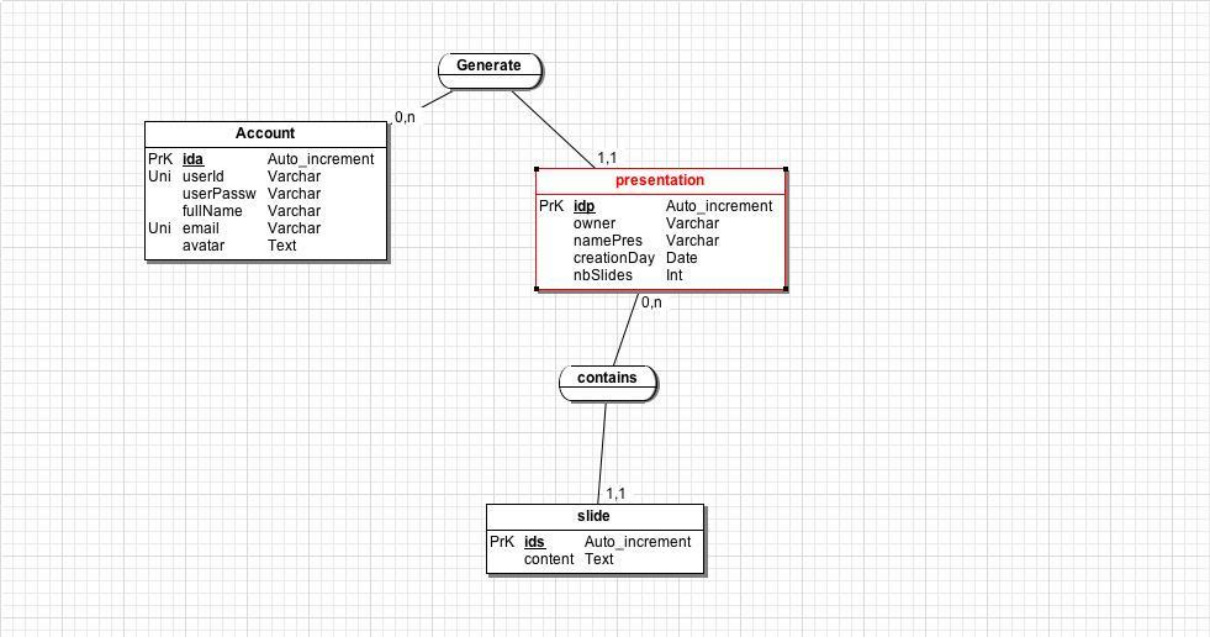

Figure 1 Conceptual Database Model

b) Regular expression to protect against vulnerabilities

We used regular expressions on all the input box, to permanently avoid SQL injection attack[1] and XSS[2] vulnerability. Indeed because of the input box, anyone who has malicious intentions could try to do something with the database, the user login webpage and user creation webpage are directly connected to the database and it's a potential vulnerability.

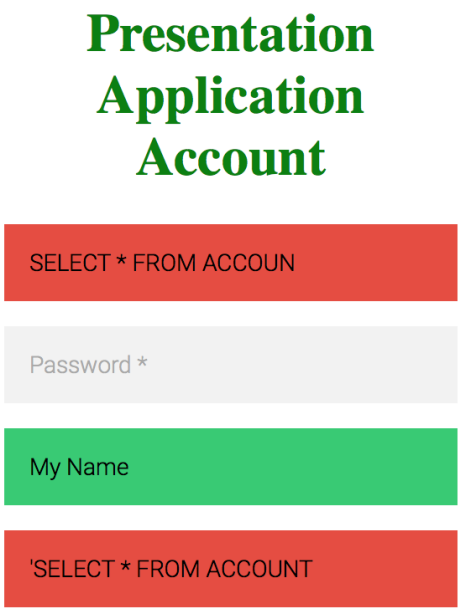

Figure 2 Input box regex result

When the input is wrong the input-box becomes red, otherwise it becomes green or white if it's empty.

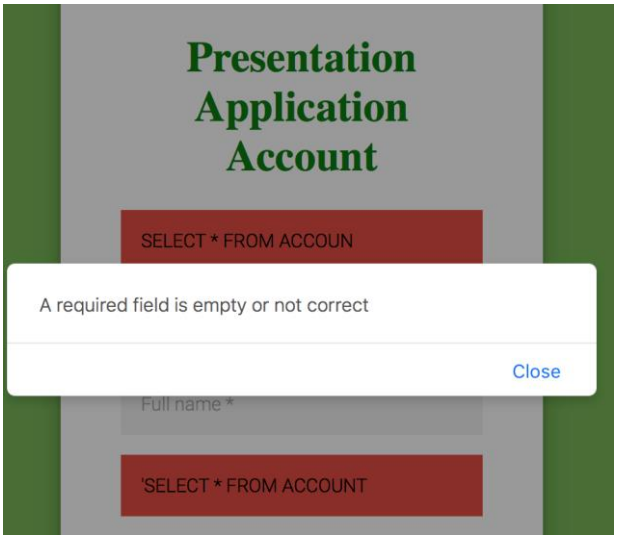

Figure 3 On click button create user account

If there is one or more red field an error message is displayed and the user account is not created. Besides regex, we also use a character limitation constraint. By this we limit the maximum of character input in the formulary. Concerning the input file, which is a really vulnerable input by XSS attack, we used a regular expression to force the user to load only files with jpg, jpeg, png or gif extension name.

IV – Discussion

The project in its entirety was clearly solvable, but we encountered several challenges of which directions to take, as there are always advantages and disadvantages for the different solutions. One of the challenges was probably how to split the workload into different parts, where each one got a task to do, independently from the others. As to of us we have limited experience of how to create an app, this was maybe the biggest. Another was concerning code error and developer issues. We obviously had to search lot of documentation on the web to fix our bugs and errors, but also to get the knowledges that we needed.

V – Conclusion

The biggest teaching of achieving the project was how to be methodical and organised. We though the project have been really interesting, learned how important it is with good communication between each team members. How to plan the tasks that we had to do, and to share our code and understand it between all of us. The project has been done in his totality: user account management and access control, creation and organization of presentations, slide creation, presenter notes, presentation mode, import/export of presentation to a text file. We wish we had a couple days extra, as we felt the dynamics us in between greatly improved until the end. There was a couple more features we wished to add, but I think we are all content with what have achieved.

VI – References

1 - https://en.wikipedia.org/wiki/SQL_injection
2 - https://en.wikipedia.org/wiki/Cross-site_scripting