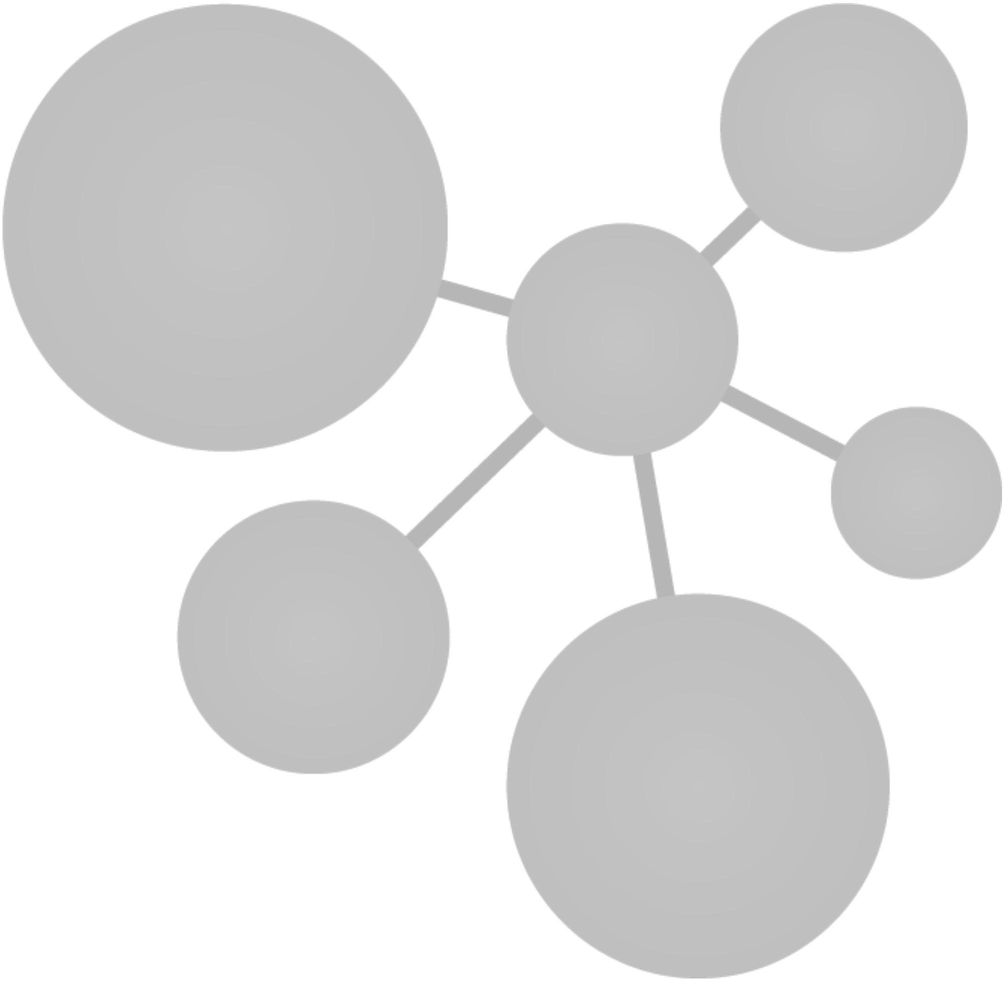
Biweekly Report II

Introduction Software Engineering



Group 9

Project Name:
MOLN – Event Platform

Students:
Lorenz Caliezi
Noah Schmid
Manuel Schüpbach
Olivier Stähli

Assistant:
Pascal Andre

Date: *30. October 2019*

Table of Contents

Frontend

Design

Mobile

After some research we decided to develop our software mobile-first.

Definition Mobile First: Mobile first, […] is a design strategy. While it may use a mobile responsive framework, it considers mobile users' needs first and foremost. Instead of creating a desktop website and then forcing it to fit in a mobile box, you create a website that considers the majority of users (on mobile) first.
(Source: *https://www.searchenginewatch.com/2019/03/05/mobile-first-design-for-2019/*)

The two main reasons were:

- More traffic is on mobile than on desktop
- We had the impression that the Ionic design framework aims for mobile as well

As a first step we made a design sketch how it should look like on the mobile. Of course, we took the style from the desktop version we described in the Biweekly Report 1.

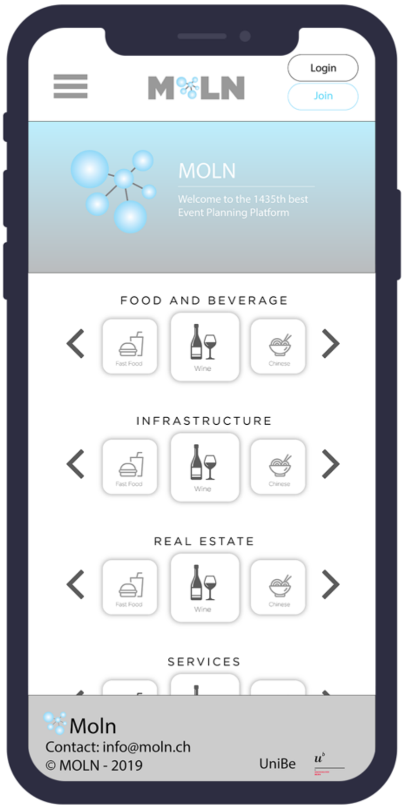


Figure 1: This is the Illustrator design raft for our mobile application.

Admin Frontend

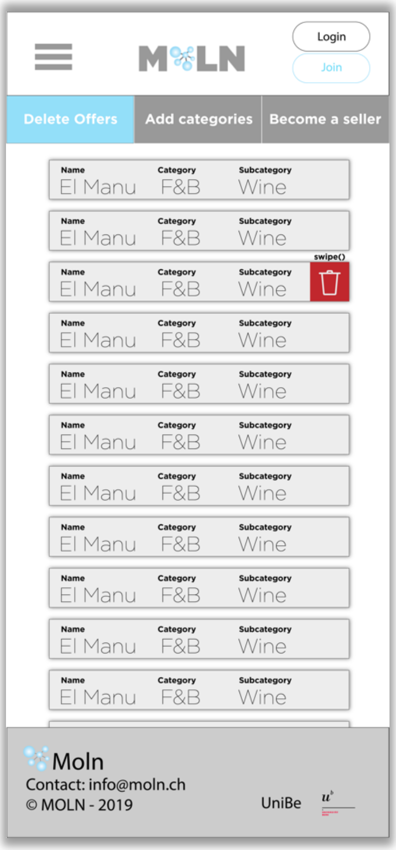


Figure 2: The admin should be able to delete offers that do not meet our guidelines and requirements.

Figure 3: The admin should be able to add subcategories. (on hold, due to heavy workload)

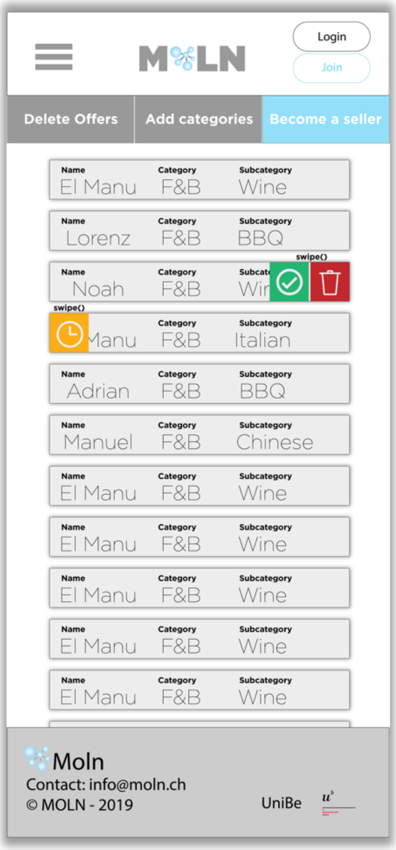

M LN
Login
Join
Delete Offers
Add categories
Become a seller
Name El Manu Category F&B Subcategory Wine
Name Lorenz Category F&B Subcategory BBQ
swipe()
Name Noah Category F&B Subcategory Wine
swipe()
Name El Manu Category F&B Subcategory Italian
Name Adrian Category F&B Subcategory BBQ
Name Manuel Category F&B Subcategory Chinese
Name El Manu Category F&B Subcategory Wine
Moln
Contact: info@moln.ch
© MOLN - 2019
UniBe

Figure 4: The admin can reject and approve sellers or can give feedback, such that the seller can edit and enhance.

Implementation (Status on 30 October)

Week 1:

- We learned how to do proper styling, according to the best practices from Maximilian Schwarzmüller and implemented it accordingly
  - Defined own ionic colors
  - Styled ionic components
  - Adjusted advanced ionic theming
- Designed a draft of how the mobile version should look like
- Decided to develop mobile first and started with its implementation
- Implemented a new hierarchy (pages, components, modules)

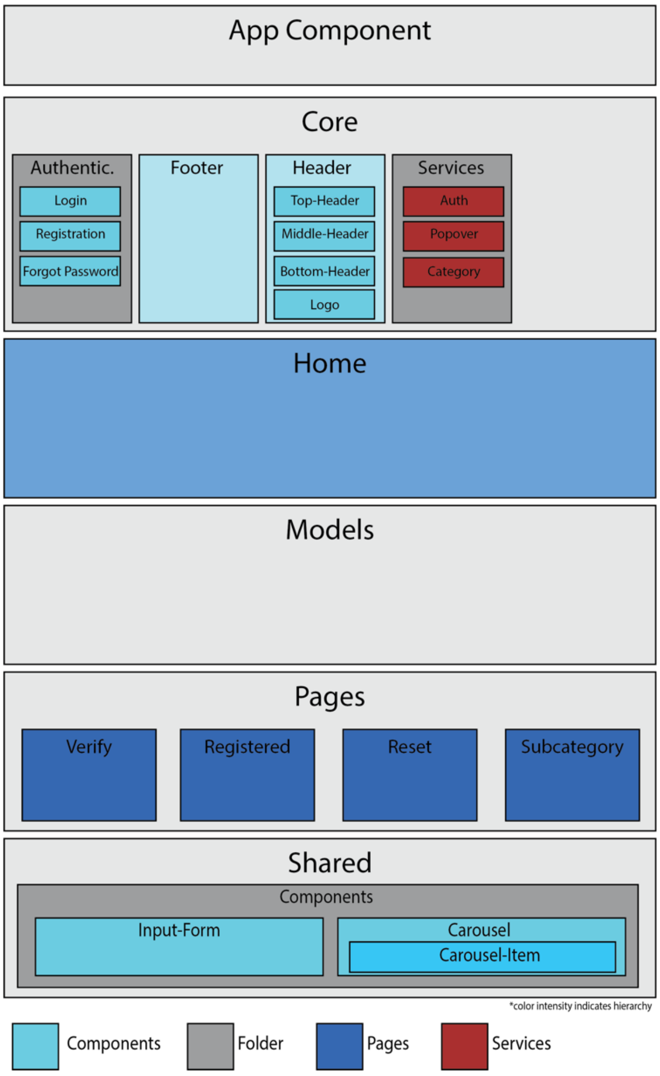


Figure 5: This is the new folder structure of our frontend.

Week 2:

- Added 'forgot password' function from Milestone 1
- Implemented logic for the category-carousel on our homepage (See figure 1)
- Developed a concept and draft of how the admin panel should look like
- Added subcategory page and started to style it
- Added content sections to the homepage according to our draft in the Biweekly report 1

Major Challenges

Solved

- Implemented a new hierarchy (pages, components, modules)
- How to develop the project further to support both desktop and mobile
  - Develop mobile first and support desktop as good as possible

Pending

- Clean-up Code – Coding according to best practices (*because it's never solved*)

- Open pop-over from anywhere without using duplicated code
- How to create a hierarchy/ division for different styling sheets.
- Apply more popular methods like Scrum, Kanban etc. to improve our workflow

Learning Outcome

We learned:

- that the Nielsen Norman Group defines the standard on how a UI should look like. Therefore, we applied some of these industry standards to our project.
- how to do proper styling
- why mobile first is a forward-thinking approach
- how to improve project hierarchy (components, modules, pages)
    - https://itnext.io/choosing-a-highly-scalable-folder-structure-in-angular-d987de65ec7

Backend

Implementation (Status on 30 October)

Week 1:

- Added “forgot password” option and finished email verification.

Week 2:

- Added category system to create/update/delete nested categories and add products to those categories.
- Added better tests and better testing structure.

Major Challenges

Coming up with a good and agile design for nested categories using a non-relational database was a minor challenge.

It was challenging to find a good method to reset the password. The solution was that the user can send our backend a request containing an email. We check if that email exists as an account in our database. If so, we send a temporary (1h) link to this email, enabling them to reset the password. However, if the email address was not present in our database, we send them an email, informing them that they can register under a certain link.

The problem with writing tests was, to understand the framework. What made this task even harder, was that some functions are called asynchronously, leading to inconsistent results. Furthermore, are some features of the backend (like the email verification) untestable. The reason is that we cannot assert that our test email address receives the verification email. To verify a test account and to make other actions on our DB easier, we made some functions (under url:/dev).

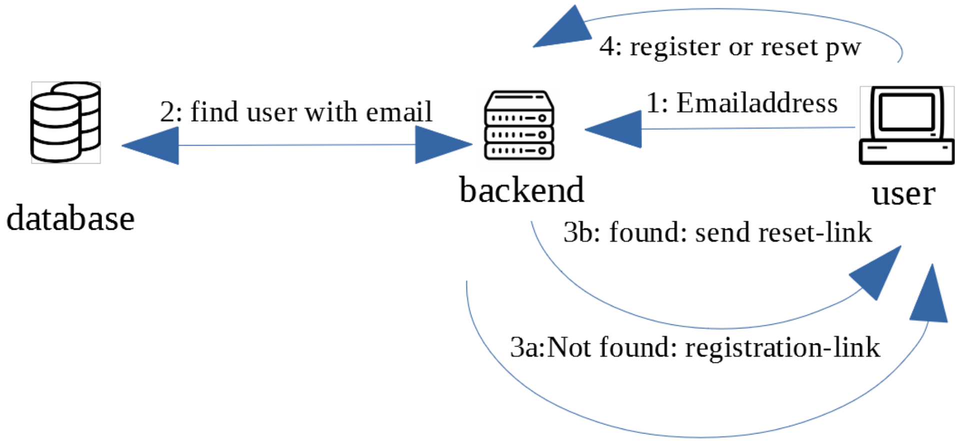


Figure 6: This image shows the process of resetting the password.

Learning Outcome

- Better understanding of MongoDB designs and working with nested documents. Learned how to better organize request handling using promises.
- Understanding of Mocha and Chai framework and how they play together. Additionally, we got a better understanding for the password resetting process