

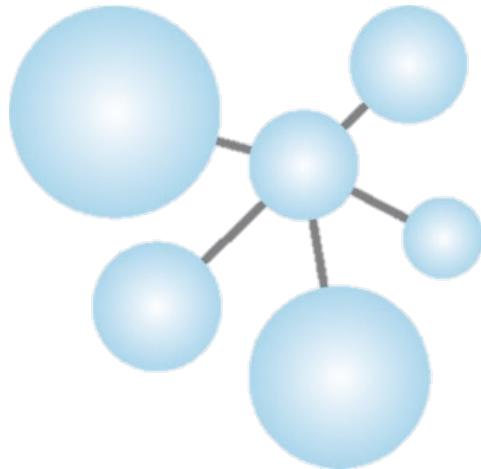
Documentation

Introduction Software Engineering

Group 9

Project

MOLN – Event Platform



Students

Lorenz Caliezi

Noah Schmid

Manuel Schüpbach

Adrian Jörg

Olivier Stähli

Assistant

Pascal André

10. December 2019

Table of Contents

Milestones.....	4
Milestone 1 (23-Oct-19).....	4
Milestone 2 (13-Nov-19).....	4
Milestone 3 (04-Dez-19)	5
Milestone 4 (11-Dez-19)	5
Manuals.....	6
Admin	6
Seller.....	9
Buyer.....	10
Challenges, Risks and Learning Outcomes	11
Requests.....	11
Backlog	12
Week 0-1.....	12
Week 2.....	12
Week 3.....	12
Week 4.....	12
Week 5.....	13
Week 6.....	13
Week 7	14
Week 8.....	14
Week 9.....	15
Week 10.....	15
SRS.....	16
Disclaimer	16
Introduction	16
Purpose	16
Stakeholders.....	16
System Overview.....	17
Use Cases	20
Use Case Sign Up with verified email	21
Use Case – Adding a product.....	22
Testing (Backend).....	24
UML/Comopdoc (Frontend).....	26
Comopdoc	26

Evolution of our hierarchy	31
.....	32
Usability Testing.....	34
Usability Testing 1	34
Takeaways	34
Conclusion.....	34
Usability Testing 2	35
Scenario	35
Documentation and Takeaways.....	35
Assistant 1	35
Assistant 2	36
Conclusion.....	37
Appreciation.....	38
Attachment.....	39
Design	39
Meeting Protocol.....	44
Disclaimer	45

Milestones

Milestone 1 (23-Oct-19)

The first milestone was prescribed by the assistants as follows:

- Get to know and learn the technologies
- ~~Collect and refine requirements for this milestone from your student assistants who are also your clients. Decide among your team who wants to play which roles. Ideally product owner, should talk to clients and get the requirements correct. He/she then conveys the requirements to the team.~~
- Start modeling your app first! Brainstorm within the team to identify important concepts of the domain and relationships between them. Use your knowledge of UML.
- For this milestone, you need to implement **authorization and authentication** features for the event management platform.

Implementation:

- We worked through the tutorials which were given by the assistants and started to build minimal applications by ourselves.
- We decided that Manuel and Noah will work on the backend and Lorenz and Olivier will work on the frontend. Adrian joined our team later and became part of the frontend team.
- We defined how we want to collaborate in an efficient manner. We defined GitHub, Slack, Trello and WhatsApp as project management and collaboration tools.
- We collected the requirements and started to refine them. Then the assistants adjusted the project so that we no longer have to work with the requirements. Therefore, this position became obsolete.
- We drafted the User Interface in Adobe Illustrator and brainstormed as a group.

Milestone 2 (13-Nov-19)

The second milestone was the main functionality of the platform. Within three weeks we implemented as much as we could.

Implementation:

- Admin panel where an admin can manage:
 - Categories
 - Users
 - Offers
- Adding products feature including an image and map picker
- Category carousel on homepage
- Profile page for users
- Product Page

- Subcategory page
- Filter function on subcategory page
- Routing between homepage, subcategories, and products

Milestone 3 (04-Dec-19)

For milestone three we decided to go one step further and implement more additional features that make our platform special and which improve the flow from registration to buying and selling. We also adjusted our platform thanks to the valuable outcome of the usability test.

Implementation:

- Chat function
- PayPal integration
- Styling
- New menu structure as a consequence of the usability testing
- More messages for success and error as consequence of the usability testing
- Notification system
- Styling
- Bug fixing

Milestone 4 (11-Dec-19)

We identified the presentation as a key element of this project. The aspect of demonstrating and selling a software project successfully seems like a critical skill to us that we want to learn. Therefore, we defined the presentation as a Milestone 4.

We invested plenty of time to create a good presentation. Our aim was to show how much time and effort we invested in this project and to show the main features of our platform. The main challenge was to stay within the time limit of 10 minutes.

In addition to preparing the presentation, we tried to improve our styling and spent some time to fix our bugs.

Manuals

Admin

One can become an administrator by contacting an existing administrator which can give one an admin-status in the admin panel under manage users, where you can set the field Admin from false to true.

Or one can set the admin flag in the database from false to true.

All the special capabilities that an admin has, are in the admin panel.

One can manage the offers:

- Either verify new offers that were uploaded
- Or let them revise the offers if something does not meet quality demands of the platform
- Or delete or reject a product for various reasons

One can manage users

- Delete other users
- Edit users

One can manage categories

- Delete/Edit/Add subcategories
- Delete/Edit/Add categories

OFFERS	USERS	CATEGORIES
	admin@admin.ch	
	olivier.staelli@students.unibe.ch	
	os@oxon.ch	
	noah.schmid77@gmail.com	
	lorenz.caliez@gmail.com	
	schuepbachmanel@gmail.com	
	schuepbachmanuel@gmail.com	
	adrian.joerg@students.unibe.ch	
	d2308843@urhen.com	
	d2308843@urhen.com	
		X DELETE
	tye9274@bcaco.com	
	ybi36939@bcaco.com	
	jan_matter@bluewin.ch	
	ir67705@eveav.com	

The admins are highlighted in blue and have an admin badge.

Press on a user to see the detailed mode. You can edit the profiles of non-admins and you are able to delete their profile.

Fast mode: Swipe left to delete users.

Figure 1: This is a screenshot of the admin panel where you can manage users

OFFERS

USERS

CATEGORIES

Entertainment

- DJ
- Actors
- Sports
- Art
- Speaker
- Musician

+SUBCATEGORY

Location

- Mansion
- Cabin
- Room
- Hall
- Hotel
- Eatery

+SUBCATEGORY

Services

- Photo
- Video
- EMT
- Market
- Security
- Staff

+SUBCATEGORY

Create a new Category

Add a Subcategory

Category Name

Category Slug

CREATE CATEGORY

Here you can delete or edit subcategories.

Press this button to add a subcategory in the according category.

Toggle this switch to add a subcategory instead of a category

Figure 2: This shows the page where an admin can manage the categories

OFFERS		USERS		CATEGORIES	
<input checked="" type="checkbox"/> Show verified <input checked="" type="checkbox"/> Show in Revise				Sort by ↑	
Name	Seller			Status	
Exakte Wissenschaften	Admin			X	▼
Lörus Schnapps	Lorenz Caliezi			X	▼
Olivier Staehli	✓			✓	▼
Delicious Barbecue	Olivier Staehli			✓	^
Delicious Barbecue		Description Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.			
		DELETE REVISE ACCEPT			
Healthy Salads	Olivier Staehli			✓	▼
Location for exclusive events	Olivier Staehli			✓	▼
Mansion with Pool	Olivier Staehli			✓	▼
Wonderful Villa	Olivier Stähli			✓	▼
Wiener Schnitzel	Olivier Staehli			✓	▼
Italian Food	Olivier Staehli			✓	▼
Urban Street Food	Olivier Staehli			✓	▼
High Quality Fondue Forks	Lorenz Caliezi			X	▼
Plate of multiple sushi rolls	Lorenz Caliezi			✓	▼
Chardonnay Year 2013	Max Mustermann			✓	▼

Filter and sort the offers you would like to see.

Fast mode: Swipe left to display the revise and delete button. Swipe right to display the verify button.

Click on an offer to display the details. As well as in the fast mode above you can delete, revise and accept.

Figure 3: This screenshot gives an impression how one can monitor the offers.

Seller

One can become a seller if he is registered on our platform (as a buyer) and completes the profile by entering the required data (name, country, address). If you click on the person icon on the top right, you can select Selling and you will be asked to add the missing data. Or you can go directly to your profile, complete the information and then go to Selling and add your product. If you have added a product you have to wait for max. 24 hours until an admin verifies your product. Only after verification of an admin will your product be visible to the public. If the product has been verified, you will get a notification. The notifications can be found on the top right of the page when you click on the bell icon.

If someone wants to order your product you will get a notification, or you can just check out the selling page under offers. You can decide whether you want to accept or reject the offer. If you want to answer any questions the buyer has or want to ask questions of the buyer, you can use the chat form.



Figure 4: This shows the process of becoming a seller and adding your first product in a nutshell.

Buyer

To become a seller, you need to register. Click on join, enter an email and a password. You will be asked to verify your email. After you have verified your email you are a buyer on our platform. You can log in and browse through the products. You can use the search bar or the categories and subcategories and filter the results by rating, date, and price as you desire.

When you have found a product you want to buy, you can order it by clicking on 'Order now' on the product details page. When you order, you will be asked to choose a date when you need the product and add some remarks.

After ordering you can no longer cancel the order. The seller can accept or reject your offer. You will get an according notification (top right (bell icon)) if the seller has reacted. If he accepted your offer you can pay it via your 'buying' page. If you have any further questions for the seller you can use the chat on the buying page.

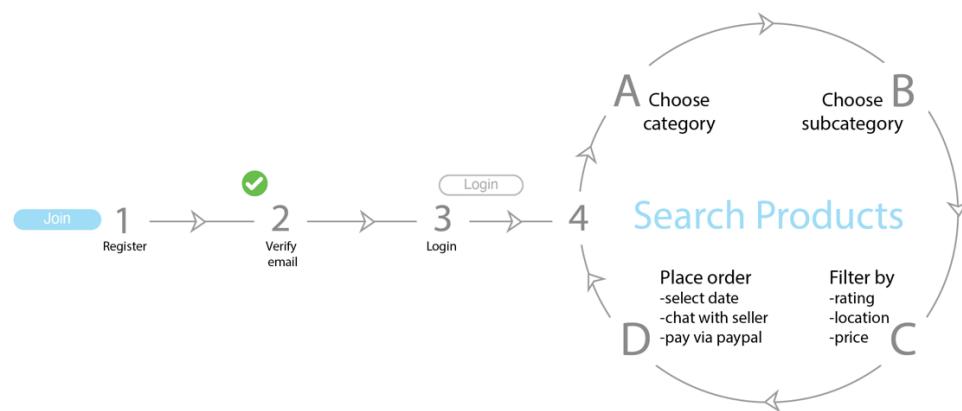


Figure 5: This is how the platform works for a user.

Challenges, Risks and Learning Outcomes

We have documented our challenges, risks and learning outcomes in our Biweekly reports.
Please go to the folder /Documentation/biweeklyReports.

Requests

Please find the documentation of our requests on GitHub (request.md).

Backlog

Week 0-1

Frontend

- Getting to know Angular and Ionic

Backend

- Getting to know Express and Mongoose

Week 2

Frontend

- Implemented the very basic structure of the project.
- Designed the header.
- Added a bunch of smaller stuff like bootstrap, the favicon etc.

Backend

- Implemented login
- Implemented sign-up
- Implemented email verification
- Get specific user by id
- Update user information
- Check authentication of a user

Week 3

Frontend

- 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)

Backend

- Added ‘forgot password’ option and finished email verification.

Week 4

Frontend

- 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

Backend

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

Week 5**Frontend**

- Implement admin panel (delete offer)
- Refine category carousel on homepage
- Add routing between homepage, subcategories and products
- Re-implement code for the navigation-bar
- Restructure and rename basic design of the project:
 - top-header → header
 - middle-header → home-banner
 - bottom-header → navigation-bar
- Styling

Backend

- Changed the category system from a nested system to one based on materialized paths. Thanks to this adjustment, we don't have to save categories multiple times. Editing and deleting categories get much simpler and faster.
- We did a lot of cleaning up as we learned new ways of implementation.
- Rewrote all tests using chai as request and assertion framework. Implemented the before/after Hooks of the tests.
- Made the buildup and teardown function used by said hooks

Week 6**Frontend**

- Implement admin panel (manage user)
- Add profile page for users
- Filter function in subcategory page
- Dropdown menus for subcategories
- Add Compodoc documentation
- Reach a 25% documentation coverage on our code (frontend)
- Add functionality such that user can add products
- Styling
- Add infographics to homepage

Backend

- Added support for uploading and storing images on the backend.
- Optimized performance of many controllers.
- Implemented authorization features, so that only admins can edit every value. Users can now just edit their own profile and their products.
- Images are getting deleted as soon as they're not needed anymore.
- Started implementing the notification and order system, which will be needed when implementing placing orders. The notification system will be needed to notify sellers that their product was bought, or whether it was accepted.
- Comments/ratings can be added to products now.
- Wrote tests covering user, product, and category actions. The user is fully tested, but product and category have some untested requests.
- Shown below is the build, testing and cleanup of a testcase.
- In the real test there are more tests being made on the returned user. But they are omitted in this example. Note that the blue Arrows represent a request of which the response is being tested.

Week 7**Frontend**

- Add add-categories to the admin panel
- Edit product on your profile
- Bug-fixing and styling

Backend

We ran into a problem with the file system we've implemented so far. Our hosting provider frequently deletes all newly created files on the server.

That means if you upload a product/category/user picture, it won't be accessible anymore after a few hours. To fix this, we had to come up with a new system to save our files. At first, we tried implementing google drive support using Google's API. This got way too complex and unhandy for our situation. So, we finally looked for another hosting provider, where we implemented the file system using PHP. Now, if you upload anything to the backend, an HTTP request to the file storage service will be made, where the file gets stored.

Week 8**Frontend**

- Add revise function to manage offer page in the admin panel
- Styled the carousel on the frontpage
- Add google maps to the add product page

- Filter products on the product page
- Add icons to the subcategories
- Reorganized categories and subcategories
- Start to implement notification system
- User testing at Switzerlan (see attachment)

Backend

Support for creating and handling orders was added to the backend functionality. If a user places an order, the seller can decide whether he wants to accept or deny the order request. Also, a lot of minor adjustments and bug fixes had to be made.

We implemented the notification system on the backend. This system will be used to notify sellers about their product status and potential customers. Most important tests for notification system, user, product and categories have been implemented.

Week 9

Frontend

- Bug fixes
- Styling
 - New header
 - Product details page
 - New Infographics
 - Subcategory page
- Implemented new menu structure
- Implemented ordering system
- Order details page
- Code documentation
- Presentation and demonstration

Backend

- Bug fixes

Week 10

Frontend

- Bug fixes
- Styling
- Presentation and demonstration

Backend

- Bug fixes

SRS

Disclaimer

After the initial requirements collection this project has been changed from a 'client-to-developer-team' project to a project, where we - as the developer team - decide what features we want to implement, how we implement them and in what order we do that, we have no given requirements.

In the figure bellow, you see our use cases for the system, with all features that we wanted to have implemented and realized. Note that they aren't specific at all, as we decided, that the person in charge of a feature decides how the best implementation is done.

If more than one person was designing, developing or implementing a new feature, the feature was discussed bilateral and then the requirements were written into a new Trello card. To manage our workflow and debating over new or changing old requirements we discussed them in our (at least) weekly meetings and transferred the take-aways to our Trello board. As we learned the frameworks and web developing practices on-the-fly, we ended up with the idea that such dynamic requirements (that we just agreed upon with a use case and some basic form of how to implement it) were the most efficient way to get work done.

While our main communication was done via Slack, our requirements collection/feature-set-planning was done via two Trello boards, one of which was for the backend (<https://trello.com/b/41OknUVk/backend>) and one for the frontend (<https://trello.com/b/jnSnH8oi/frontend>). On these Trello boards, you will find a card called completed stack, where you find our implemented features, bugfixes and so on.

With the change of project, we decided against a detailed SRS log for the reasons mentioned above. Nevertheless, we have written a short SRS log, so that we could gather first experiences in this area as well and give those interested an insight into our project.

Introduction

This chapter shows the plan for the development of the platform MOLN. MOLN is an event platform where buyers can register to buy or book products, services, infrastructure, etc. These offers come from buyers who can also register on the platform.

Purpose

Especially in our materialistically oriented society, people experience fewer lasting experiences. At the same time, however, the significance of experiences is increasing strongly. That is why we want to provide an event platform on which people can easily put together an event. With just a few mouseclicks you can organize a location, a catering, a band and much more.

Stakeholders

We have three different kinds of users:

Customers

Potential Customers: The ones that have registered on our platform.

Actual Customers: The ones that have purchased at least one product.

The target audience involves everyone. From the 5-year-old who wants a bouncy castle and a clown to the 100-year-old who would like to book an orchestra.

Sellers

Potential Sellers: The ones that have completed their profile information to become a seller.

Actual Sellers: The ones that offer at least one product on our platform

The target audience involves people and organizations which offer a product, service or a location that is related to events.

Admin

People who manage the platform and have admin access.

Platform owner

The most important factor for the platform owner is the business case:

1. With a commission of a few percents (<5%) on each purchase that is made (deducted from the part belonging to the seller) the platform could become profitable
2. By advertising, you can generate a constant revenue flow as soon as you have a stable customer frequency.

System Overview

The system is going to be a hybrid application. One that runs on mobile and on desktop. A team of five computer science students in the third semester will work on this project. The subject is awarded 5 ECTS, therefore a total of 750 hours should be invested (lectures included). We expect to commit more hours.

The frameworks we use are Angular and Ionic for the frontend and Express and Mongoose for the backend.

Functionality

Declaration: An admin has the functionality of a seller. A seller has the functionality of a customer and a customer has the functionality of a visitor.

- Visitors should be able
 - To browse through the categories and subcategories
 - To see details about products
 - To search for products
 - To filter a selection of products
 - Price
 - Rating
 - To read ratings about the products
 - To register
- Customers should be able
 - To login

- To reset their password
 - To buy products
 - To rate products
 - To communicate with the seller
- Sellers should be able
 - To upload a product
 - To see who has ordered their product
 - To communicate with the customer
- Admin should be able
 - To verify products
 - To delete products
 - To manage other users
 - To add new categories and subcategories

Features List

1. Sign Up
2. Log in / Log out
3. Forgot Password / Reset Password
4. Resend verification Email
5. Verify Email Address
6. Search for Products
7. Carousel the categories
8. Side menu
9. Filter the products
10. Add/Edit Profile Information
11. Delete a user
12. Add a product
13. Take a picture with the camera
14. Auto Locate on the map / Pick location on the map
15. Order a product
16. Pick dates
17. Accept / Reject Offers
18. Chat between buyer and seller
19. Manage Users
 - a. Edit user information
 - b. Delete users
20. Manage Offers
 - a. Verify/Decline Products
 - b. Delete Products
21. Manage Categories
 - a. Add new (Sub-)Categories
 - b. Edit (Sub-)Categories
 - c. Delete (Sub-)Categories

- 22. Pay orders
- 23. Review system
- 24. Notification system

Use Cases

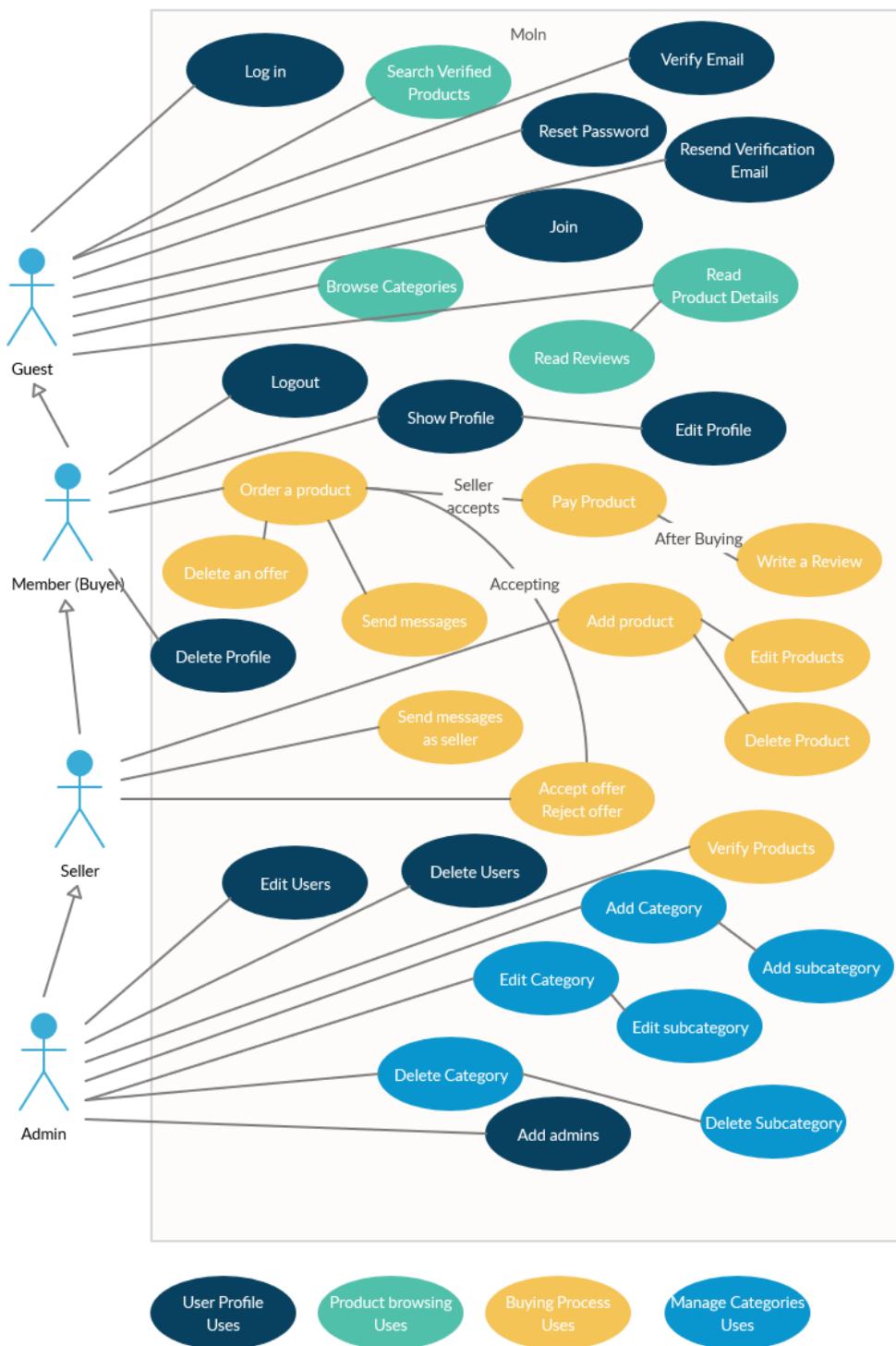


Figure 6: This Use Case diagram gives an overview over all use cases.

Use Case Sign Up with verified email

Name: Sign Up with verified email.

Description: A new user wants to sign up on the website. He registers and

Precondition: User has an internet connection, is on the website and has an email address

Event Flow:

1. User presses on the sign-up button.
2. User enters his email address
3. User enters a password
4. User can now press the submit button
5. User presses the submit button
6. The user data is sent to the backend
7. The backend saves the new user
8. The backend sends an email to with a verification token
9. The user changes to the email inbox
10. The user presses the button
11. The website sends an email verification status update to the backend
12. The user is now signed up

Post Condition: The newly created user is stored in the database and the user can now log in with his credentials.

Alternate Event Flow:

1. User presses on the sign-up Button.
2. User enters his email address
3. User enters a password
4. User can now press the submit button
5. User presses the submit button
6. The user data is sent to the backend
7. The backend saves the new user
8. The backend sends an email to with a verification token
9. The user doesn't receive the email
10. The user presses the resend button
11. The backend sends an email to with a verification token
12. The user changes to the email inbox
13. The user presses the button
14. The website sends an email verification status update to the backend
15. The user is now signed up

Post Condition: The newly created user is stored in the database and the user can now log in with his credentials.

Use Case – Adding a product

Name: Adding a product

Description: A user wants to add a product. He is not a seller yet.

Precondition: The user is logged in on the website but has not filled in all the required information on the profile page.

Event Flow:

1. The user opens the side menu
2. The user presses «Add Product» button
3. The user is missing some information
4. An error toast is shown
5. The user is redirected to the profile page after clicking OK on the error toast
6. User fills in the missing information
7. The user fills in the information
 - a. The user fills in the title
 - b. The user picks an image
 - i. The user takes a new picture with the camera OR
 - ii. The user uploads a file from the filesystem
 - c. The user fills in the price
 - d. The user picks a location
 - i. The user uses «Auto locate» OR
 - ii. The user picks a location on the map
 - e. The location field gets filled in automatically
 - f. The User selects a category
 - g. The Selection options on the subcategory selection get updated to only show the subcategories of the selected category
 - h. The user enters a description
8. The user has filled in all the correct information
9. The user presses the Add Product button
10. The new product data is sent to the backend
11. The backend sends the response if the Product has been added
12. A toast is displayed with information about the product verification process

Post Condition: The newly created Product is saved in the database and can be seen by admins and the seller (on the page «Selling»).

Alternate Event Flow:

1. The user presses on the profile icon
2. The user presses on the selling button
3. The user is missing some information
4. The user can directly fill in the information
5. The user fills in the information and submits
6. The user can now press on «Add product»
7. The user fills in the information
 - a. The user fills in the title
 - b. The user picks an image

- i. The user takes a new picture with the camera OR
- ii. The user uploads a file from the filesystem
- c. The user fills in the price
- d. The user picks a location
 - i. The user uses «Auto locate» OR
 - ii. The user picks a location on the map
- e. The location field gets filled in automatically
- f. The User selects a category
- g. The Selection options on the subcategory selection get updated to only show the subcategories of the selected category
- h. The user enters a description
- 8. The user has filled in all the correct information
- 9. The user presses the Add Product button
- 10. The new product data is sent to the backend
- 11. The backend sends the response if the Product has been added
- 12. A toast is displayed with information about the product verification process

Post Condition: The newly created Product is saved in the database and can be seen by admins and the seller (on the page «Selling»).

Alternate Event Flow:

- 1. The user presses on the profile Icon
- 2. The user presses on the selling button
- 3. The user is missing some information
- 4. The user can directly fill in the information
- 5. The user fills in the information and submits
- 6. The user can now press on «Add product»
- 7. The user fills in the information
 - a. The user fills in the title
 - b. The user picks an image
 - i. The user takes a new picture with the camera OR
 - ii. The user uploads a file from the filesystem
- c. The user fills in the price
- d. The user picks a location
 - i. The user uses «Auto locate» OR
 - ii. The user picks a location on the map
- e. The location field gets filled in automatically
- f. The User selects a category
- g. The Selection options on the subcategory selection get updated to only show the subcategories of the selected category
- h. The user enters a description
- 8. The user has not filled in all the required information

Post Condition: An error toast is displayed that the form is incomplete. The user can now try to fill in the form correctly.

Testing (Backend)

We implemented 70 Tests to make sure our backend endpoints always work properly. For that, we used mocha as a basic test framework and chai for assertions. To make the requests to our backend server we needed chai-HTTP.

The problem was that the tests should not depend on each other. But oftentimes we needed registered users, admins, categories or products to test certain functions. To do that, we wrote stateless classes that make a needed instance and can also delete them again. That helped the development of the tests a lot and made the code more structured. For each top-level test, the environment is set up so that all needed instances are available. The individual test cases could then make mutations on these instances. After all the test cases are run, whether successful or not, the database is reset to its original state. That means we can test on the live server but cannot alter the state of the database.

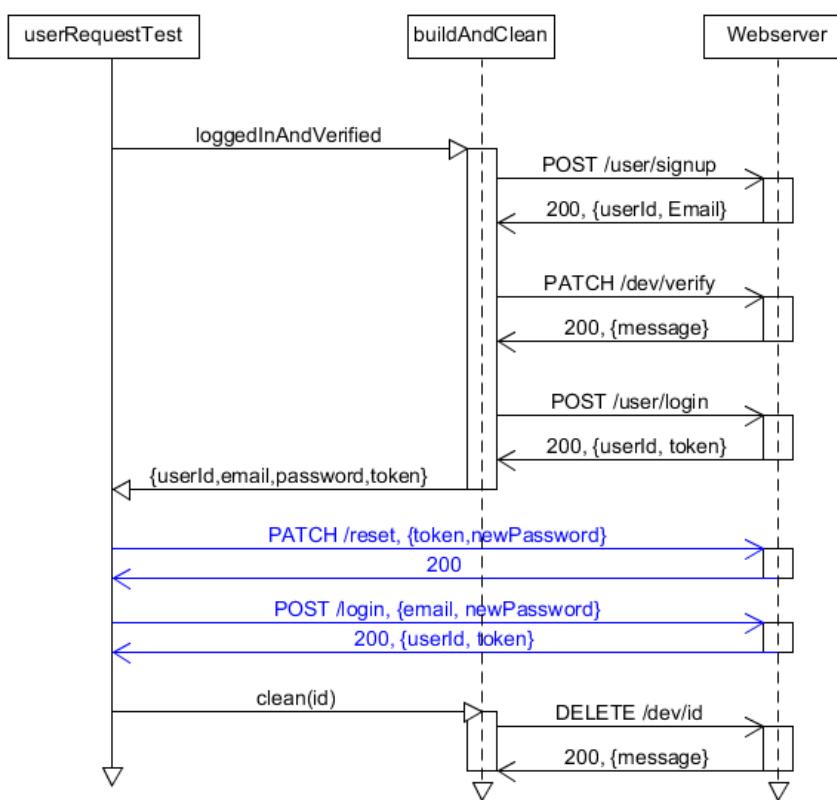


Figure 7: This example shows how the reset password request is being tested. Note that all requests utilized by `buildAndClean` are being tested before being used here.

Below you can see all our endpoints for the backend server. To the left, we can start on the Base URL (themoln.herokuapp.com). As we move to the right through the arrows, we add a path (e.g '/user') to the URL. Then the function is displayed that is called if we get a request for that path. Note that some requests can only be made as an admin or if you are logged in.

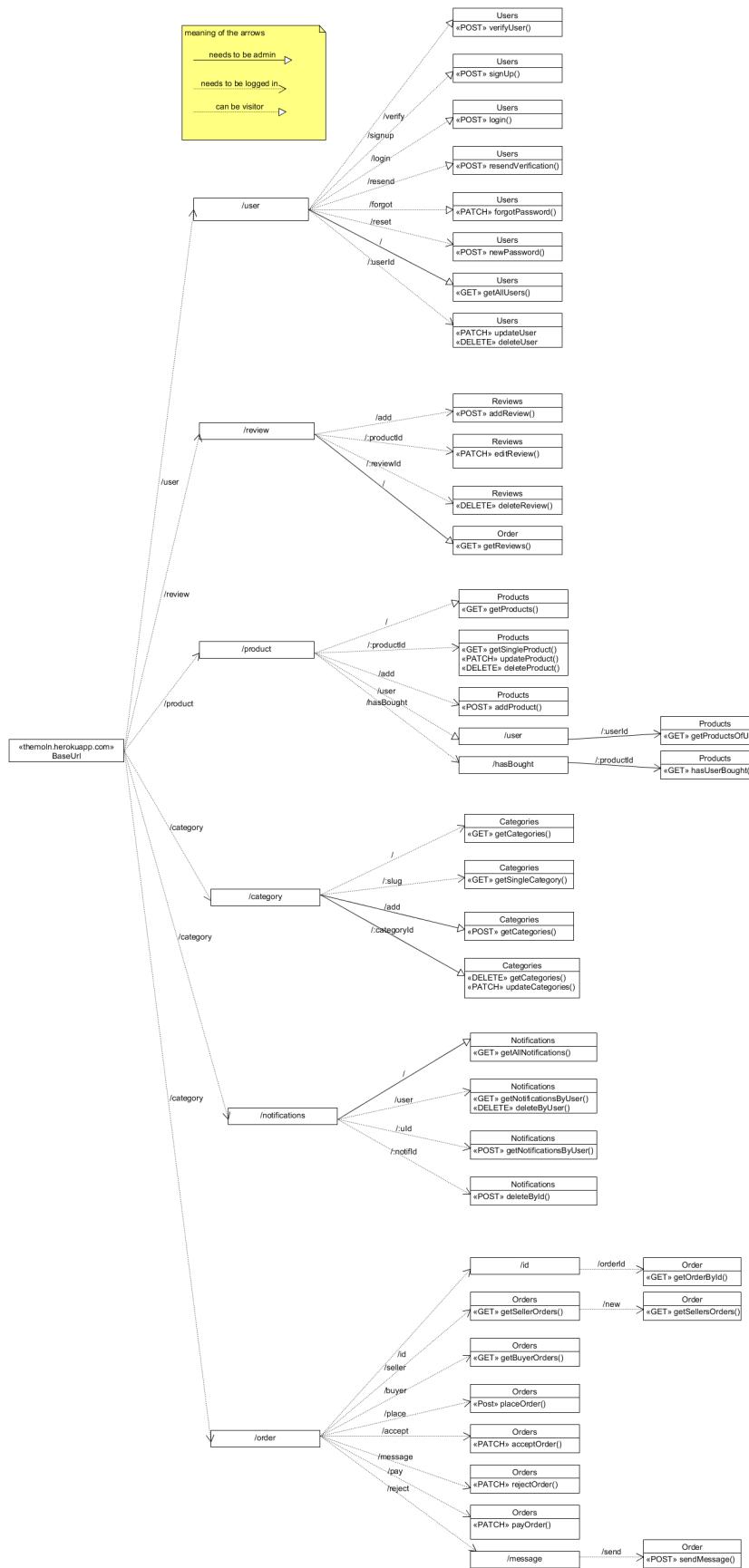
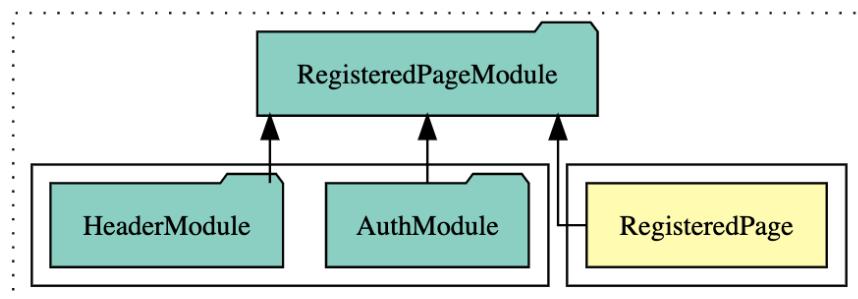


Figure 8: This illustration shows the different endpoints of the backend server. Note the different arrows which are specified in the figure.

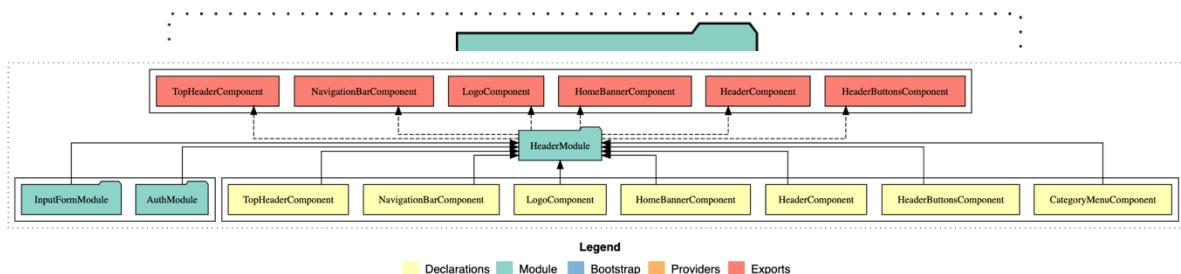
UML/Compodoc (Frontend)

Compodoc

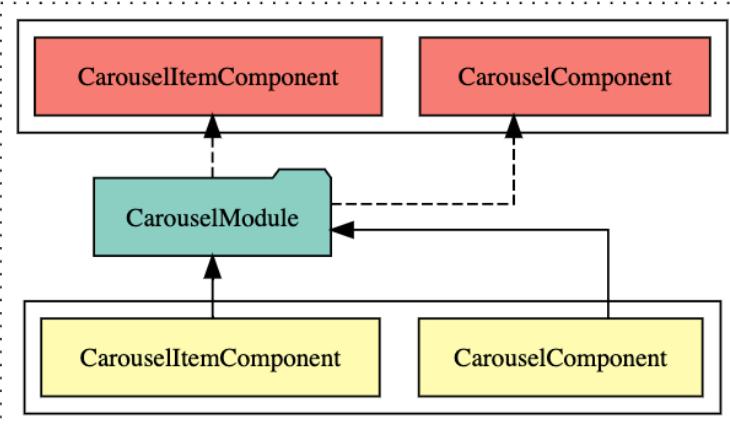
We used Compodoc as a documentation tool. Compodoc allows generating automated visualizations. Please run `npm run compodoc` to see all the visualizations. We provide some insights about our modules bellow. The overviews are unfortunately too large in size to display it in this document.



Legend

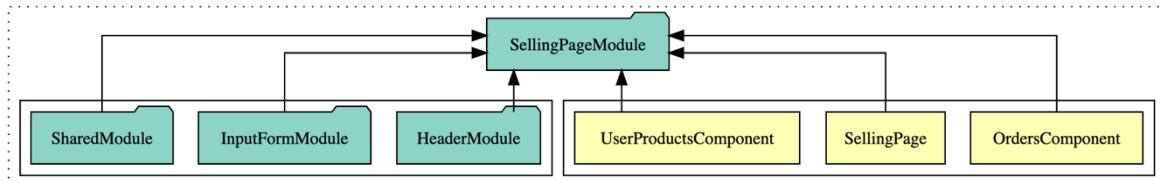


Legend



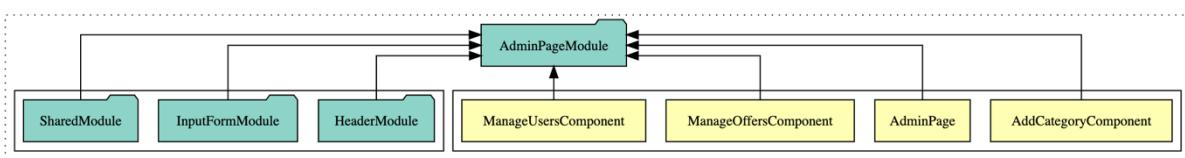
Legend





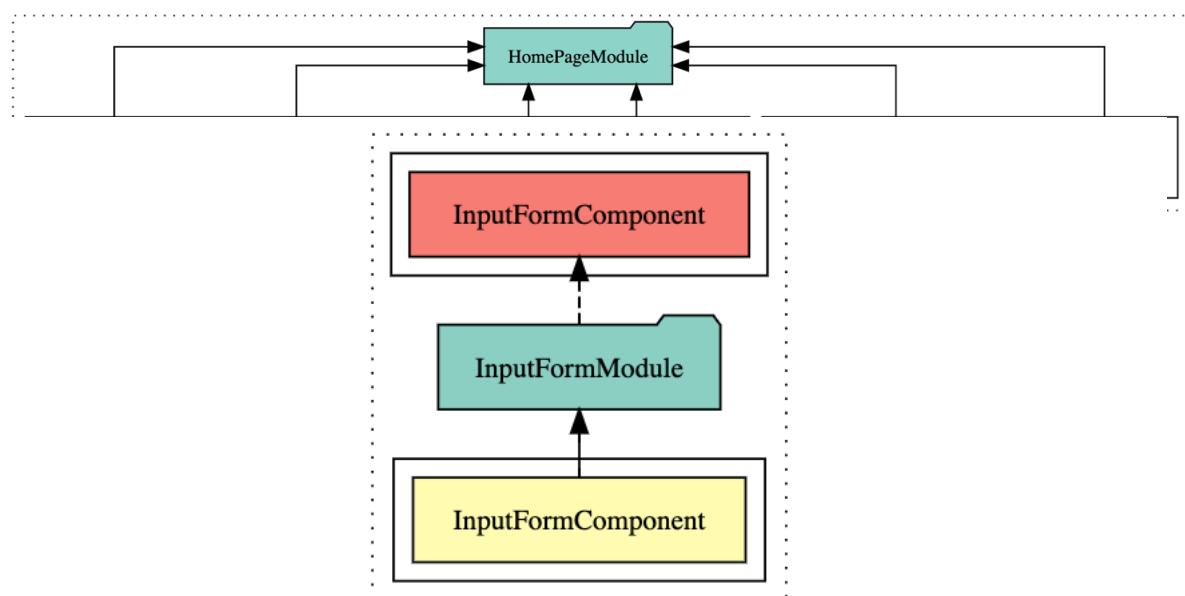
Legend

- Yellow square: Declarations
- Green square: Module
- Blue square: Bootstrap
- Orange square: Providers
- Red square: Exports



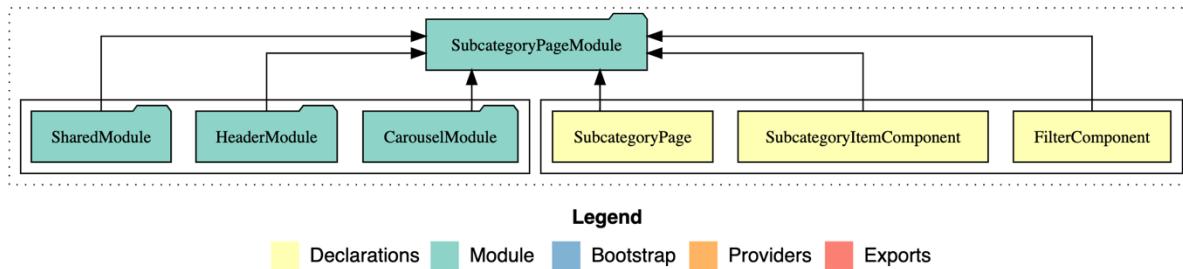
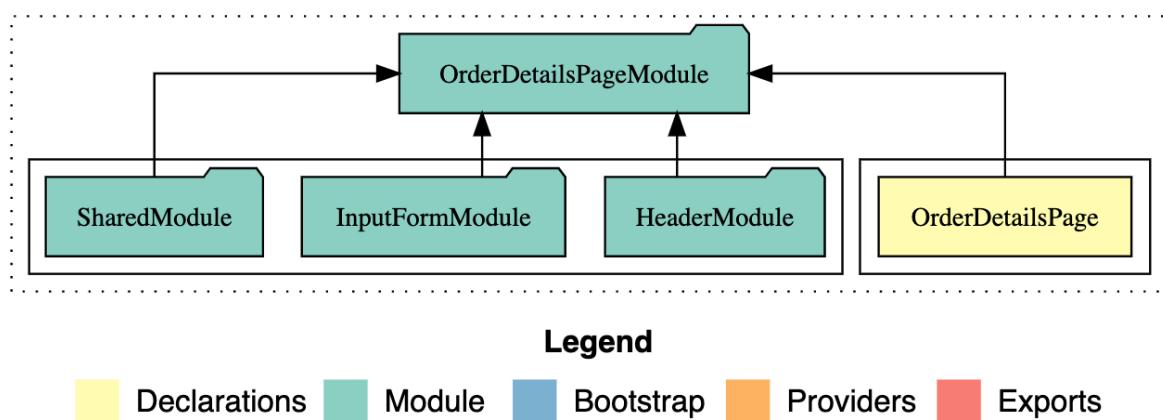
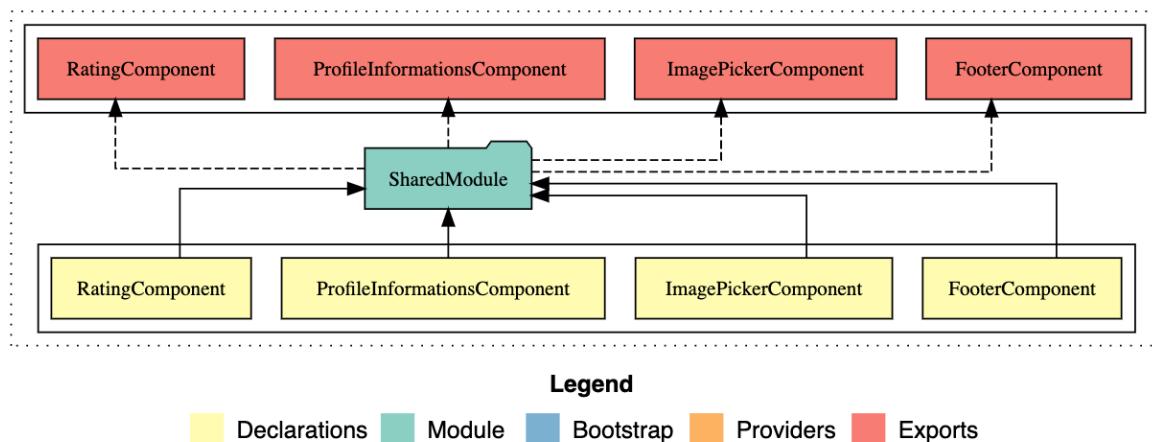
Legend

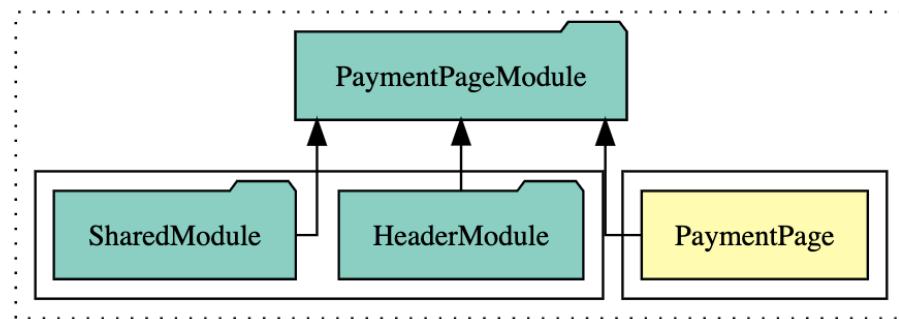
- Yellow square: Declarations
- Green square: Module
- Blue square: Bootstrap
- Orange square: Providers
- Red square: Exports



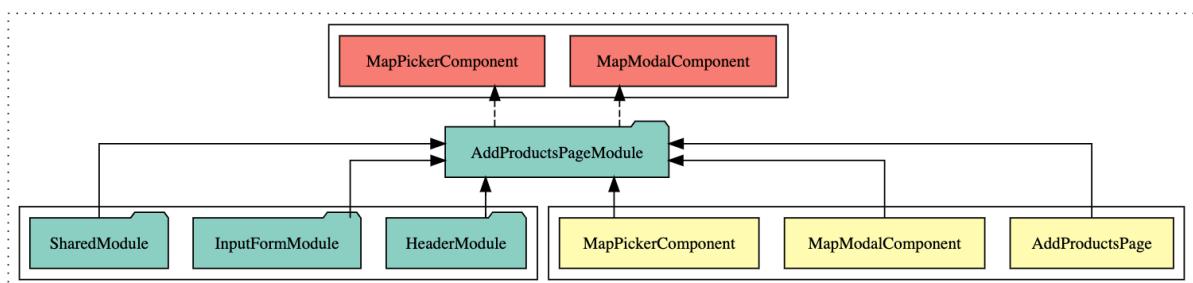
Legend

Yellow square: Declarations Green square: Module Blue square: Bootstrap Orange square: Providers Red square: Exports

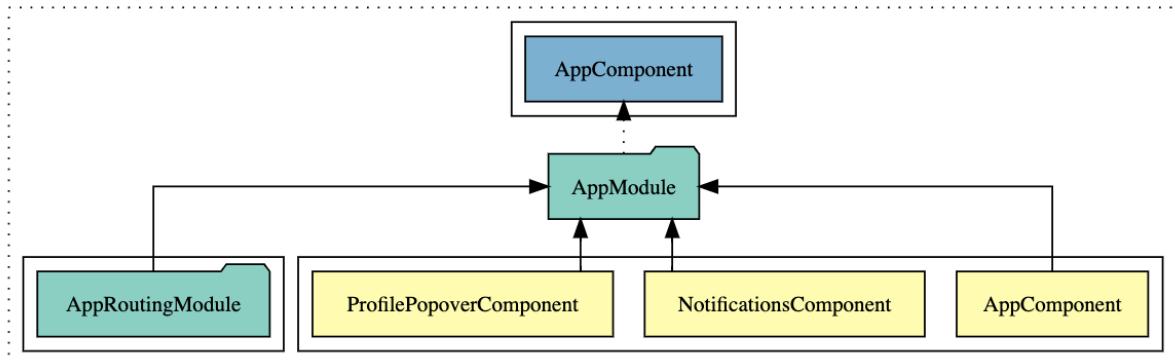


**Legend**

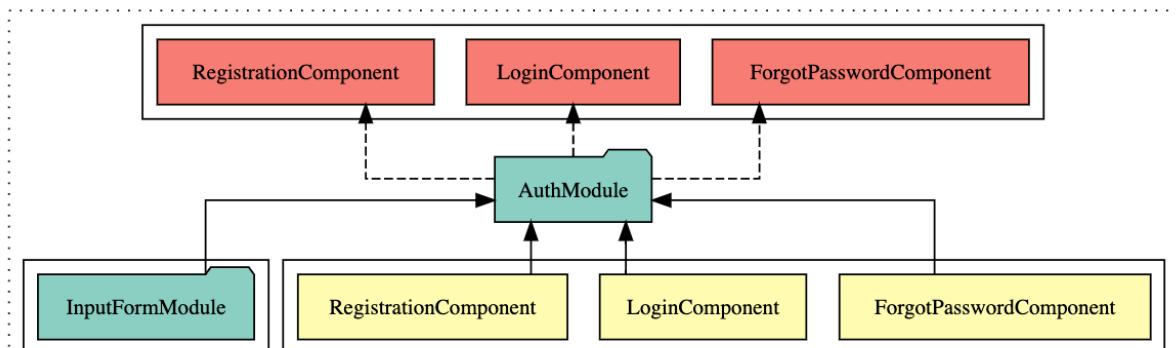
■ Declarations	■ Module	■ Bootstrap	■ Providers	■ Exports
---	--	---	--	---

**Legend**

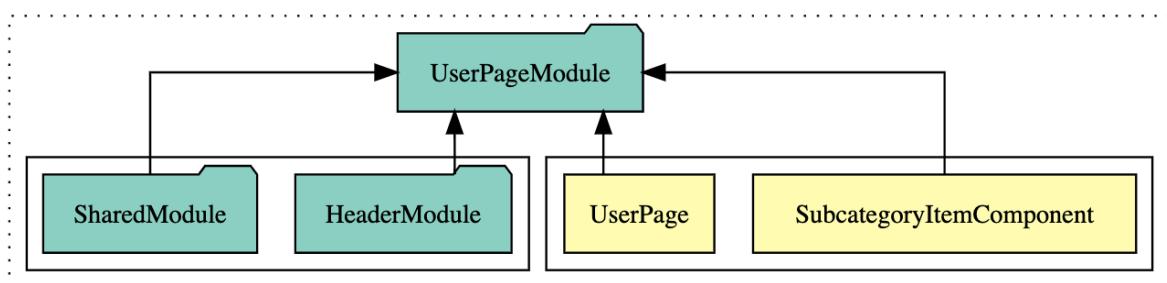
■ Declarations	■ Module	■ Bootstrap	■ Providers	■ Exports
---	--	---	--	---

**Legend**

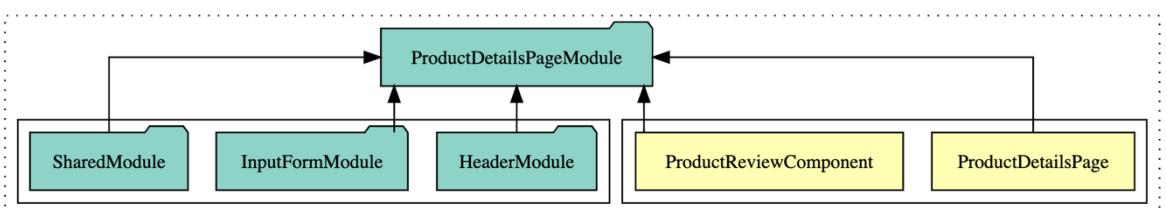
■ Declarations	■ Module	■ Bootstrap	■ Providers	■ Exports
---	--	---	--	---

**Legend**

 Declarations	 Module	 Bootstrap	 Providers	 Exports
---	---	--	--	---

**Legend**

 Declarations	 Module	 Bootstrap	 Providers	 Exports
---	---	--	--	---

**Legend**

 Declarations	 Module	 Bootstrap	 Providers	 Exports
---	---	--	--	---

Evolution of our hierarchy

The following three graphics show how our hierarchy evolved over time.

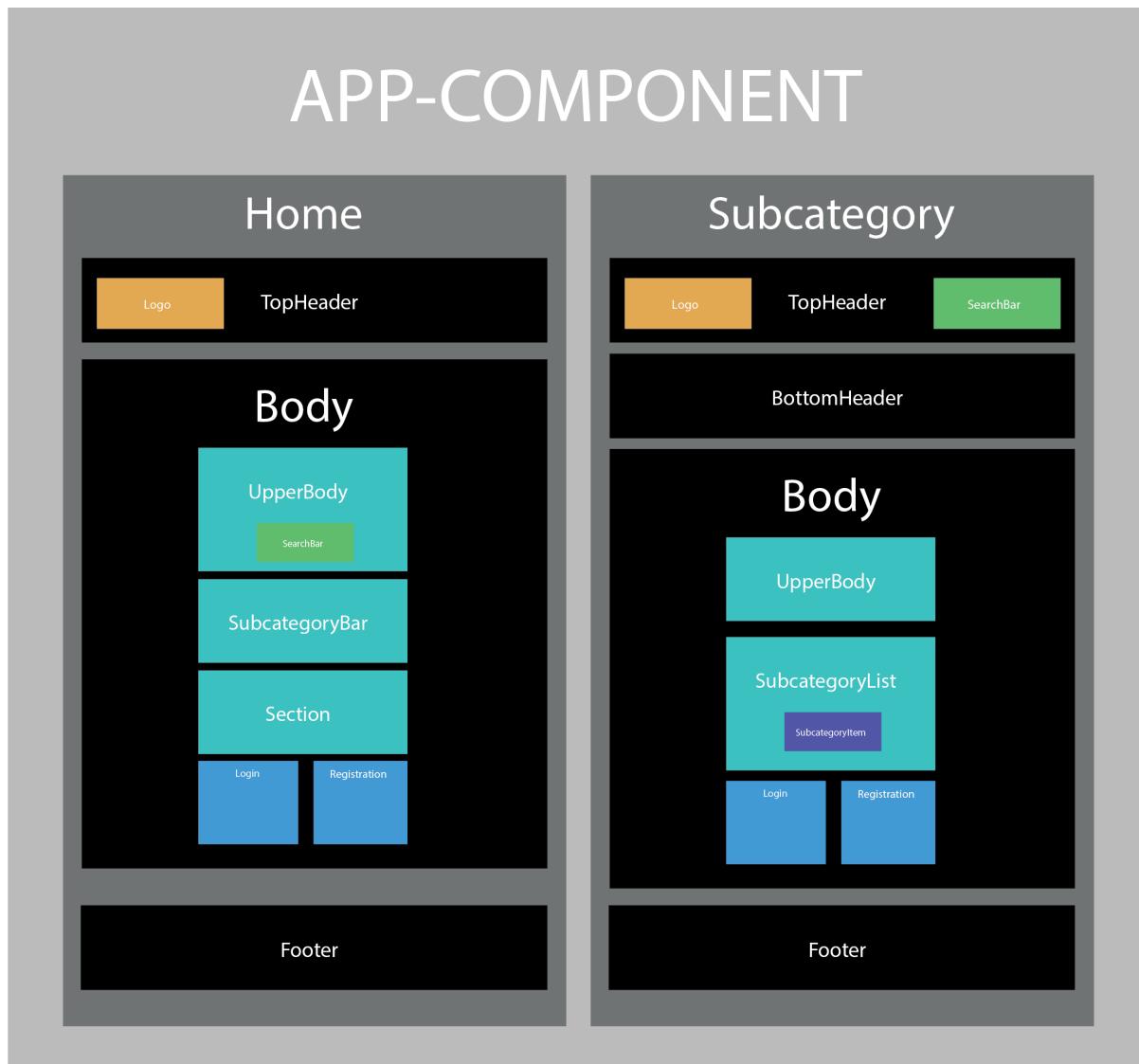


Figure 9: This is the first draft of the component hierarchy we have made for the first biweekly report (16-October-19). However, this was only a plan and we did not actually implement this. We pivoted approximately one week after.

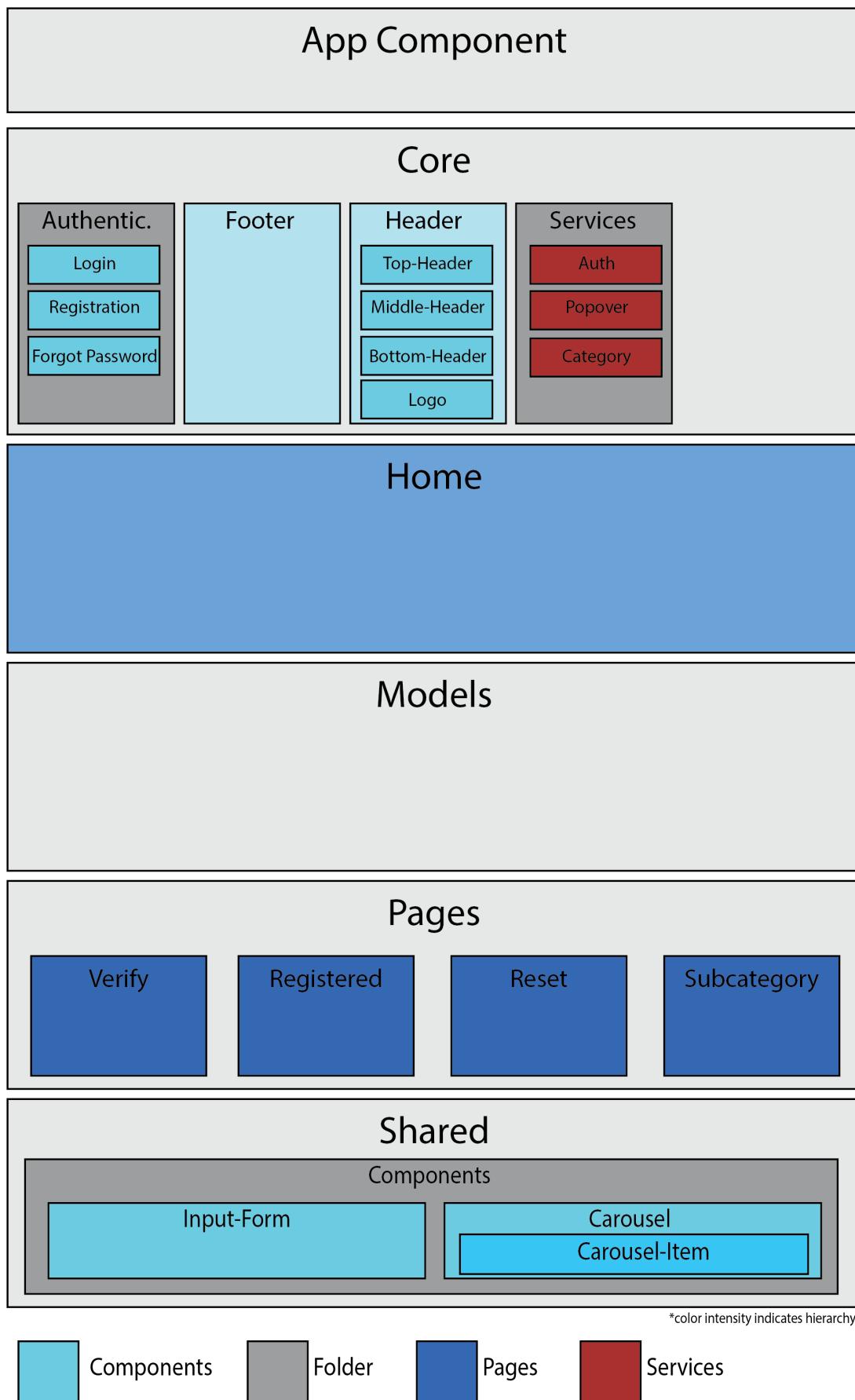


Figure 10: This is the hierarchy on 30. October. This was the actual status of implementation one week after we have submitted Milestone 1.

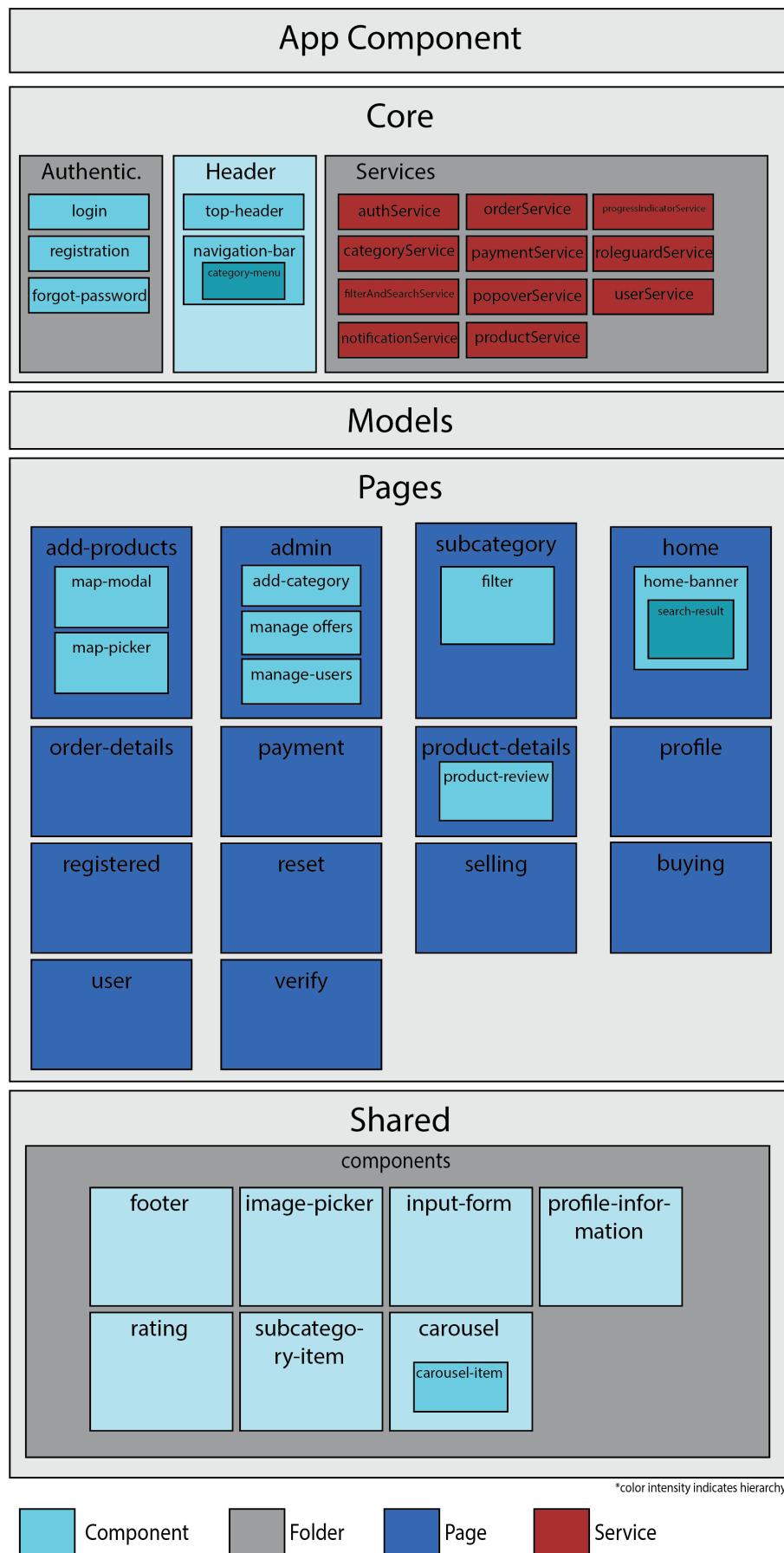


Figure 11: This is our final hierarchy (08-Dec-19). In the period from 30. October to 08. December we have kept the basic structure we have implemented and added many pages and services and moved some components to the shared folder.

Usability Testing

To check how user-friendly our platform is, we carried out two usability tests. The first one was conducted by Lorenz at Switzerlan. There he could already get important feedback that we were able to implement before the second test with the assistants. We used the first test as our preparation for the usability test with the assistant because we learn best when we actually try it.

The tests with the assistants provided further valuable feedback, which is why we decided to revise our menu structure again to make it more user-friendly.

Usability Testing 1

At Switzerlan 2019 we conducted a small user test to prepare for the usability testing with the ESE assistants and to figure out which areas of our website we need to adjust. The user testing was done with different people all aged between 20 and 25. They were given a set of tasks (see below). We measured the time it took them to execute the tasks. We helped them as little as needed, which led to a cleaner result.

Scenario:

Task 1: Forgot password (just enter the given email, I will tell you your password)

Task 2: Log in with your new password

Task 3: Change one field of your profile (e.g. name)

Task 4: Add a product (under userTesting / userTestingProducts)

Task 5: Change the description of your product

Task 6: Verify your newly created product as admin

Task 7: Delete a random user, that is not an admin

Task 8: Add a new subcategory under userTesting

Task 9: Browse through the page to find your product (category: userTesting, subCategory: userTestingProducts)

Task 10: Delete your product, that you created earlier (as a user not as the admin)

Task 11: Log out

Takeaways

- The website is generally easy to operate (most of the task were executed flawlessly)
- The admin panel and profile page still need some improvements
- The users do not understand that there is a swipe function on the admin panel when deleting a user
- Add an add-product button in profile
- Proper characterization of an admin user
- The testing also showed us, what we need to change for the usability testing this
- Wednesday.
 - Extend scenario to include registration
- Help even less to find out whether they can operate through all tasks by themselves.

Conclusion

Overall the testing gave some interesting insight into the acceptance of our website and also showed us where we have to invest more time and resources.

MOLN user testing											Notes
Nr	Time used for the different tasks										
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10	Total time in sec
1	14.17	2.37	19.53	72.07	22.14	20.39	47.24	25.24	40.06	22.17	2.15
2	22.3	9.8	23.68	92.16	33.64	40.23	44.82	54.35	39.6	9.48	1.8
3	60	1.82	34.87	71.88	42.64	25.67	42.27	39.03	50.66	22.2	3.42
4	28.37	2.3	38.98	60.4	37.33	30.98	45.77	22.04	45.64	15.18	1.5
5	33.45	4.8	30.71	80.51	30.64	23.23	35.83	40.65	60.34	17.46	2
6	15.42	2.8	25.13	100.68	30.18	42.58	39.32	37.16	44.69	20.2	1.68
7	10.57	1.3	18.71	65.17	20.68	24.18	45.1	29.58	35.18	15.9	1.9
											268.27

Figure 12: In the first usability testing we timed each step. However, after evaluation we realized that this was not very beneficial for us. Therefore, we changed to screen-recording in the second usability testing.

Usability Testing 2

We used our knowledge gained from the first test to improve our scenario and our methods. We documented both tests with screen recording and two of us wrote a protocol of what the assistant was doing. After the tests, we reviewed it and summarized the findings in a table.

Scenario

Task 1: Create a user and verify the used email address

Task 2: You have now forgotten your password, get a new one.

Task 3: Log in with your new password

Task 4: Add a product

Task 5: Change the description of your product

Task 6: Verify your product, that you created earlier

Task 7: Delete a random user but not another admin

Task 8: Add a new subcategory and delete it immediately

Task 9: Browse through the page to find your product

Task 10: Go to foods and select a price range of 0-50CHF

Task 11: Delete your product, that you created earlier

Task 12: Log out

Documentation and Takeaways

Assistant 1

Findings	Solution
Popover does not close / no confirmation (Forgot password)	Bug fix
Does not find add product on mobile (Hamburger Menu)	Implemented a new menu structure to make it user-friendly
Does not understand link to become seller (Alert - click ok)	Changed process to become a seller
Pick map overwrites location input field	Changed order of the input fields
Does not understand that the product has been created	Improved toast – increased time in which it is displayed.

Does not understand which users are admins	Instead of a grey background admins have a blue background and an admin-badge
Uses input field instead of slider at price	Improved through styling
Held featured products for navigation	Removed featured products
Filter reset at category change	Bug fix
Searched for own products in sidebar and not in profile	Implemented a new menu structure to make it user-friendly
Profile does not open	Bug fix

Feedback:

- Very mature compared to others, looks great
- A button *[not specified which one]* looks like he's disabled *[we have found a grey submit button and changed it to our primary color]*

Assistant 2

Findings	Solution
Add product: Ignores toast, doesn't understand what's happening	Improved toast – increased time in which it is displayed.
Skips add photo	Changed order of the input fields
Does not see invalid prizes – no error message if price is wrong	Bug fix
Does not understand difference from admin and other users	Instead of a grey background admins have a blue background and an admin-badge
Does not understand swipe function	Because admins are advanced users of our platform, we do not explain this feature further. If someone does not understand the swipe function whatsoever one can click on the user and the same functions appear as when one swipes.
Has added category, not subcategory	Improved user-experience by adjustments though styling
Forgets to add icon	-

Searched for his product in Add Products	Implemented a new menu structure to make it user-friendly
Hasn't noticed drop-down change at category change	Improved through styling

Feedback:

- Some ambiguities
- Feedback unclear [*we improved our toasts (clearer messages, displayed longer)*]
- Quite mature
- Separate my products from profiles [*we implemented this recommendation in our new structure*]

Conclusion

The usability testing provided very insightful results for us. Because we could observe who someone who has never seen our platform before uses it. Since we worked for weeks on it, we lost sight of how important it is that we provide feedback to the user what is going on and that everything must be as easy to understand as possible. We concluded that we need to improve the feedback that is given to the user. For example, to display the toasts longer and to have more toasts to signalize what is going on. Another main takeaway is that we need to improve the structure of the menu. The assistants got lost sometimes when trying to find a menu item. After the evaluation we drafted a new menu structure and implemented it right away. We then showed it to friends and families and got very positive feedback.

Appreciation

We would like to thank our assistant Pascal André for supervising our project.

A special thanks to Prof. Dr. Oscar Nierstrasz and his team that they made it possible to do a project in college that's so practice-oriented.

Furthermore, we would like to thank Inpeek for the hint that we should use Compodoc as a documentation framework. This saved us many hours.

Attachment

Design

This is a selection of the design drafts we have made to create a brand identity. Other visualizations are included on our homepage, in the biweekly reports, and in this documentation.

Any questions?
We are happy to help!

Contact
Project Owner
noah.schmid@studnets.unibe.ch
Exakte Wissenschaften

Any questions?
We are happy to help!

Contact
Project Owner
noah.schmid@studnets.unibe.ch
Exakte Wissenschaften

© MONU - 2019



Barbeque (BBQ)

pieces of beef, fowl, fish, or the like, roasted over an open hearth, especially when basted in a barbecue sauce.



Our Top 5 Picks

Filter

★ Rating▼

📍 Location▼

\$ Price▼



Title

★★★★☆

Price

Location



Title

★★★★☆

Price

Location



Title

★★★★☆

Price

Location



Title

★★★★☆

Price

Location



Title

★★★★☆

Price

Location



Title

★★★★☆

Price

Location



Title

★★★★☆

Price

Location



Title

★★★★☆

Price

Location



Title

★★★★☆

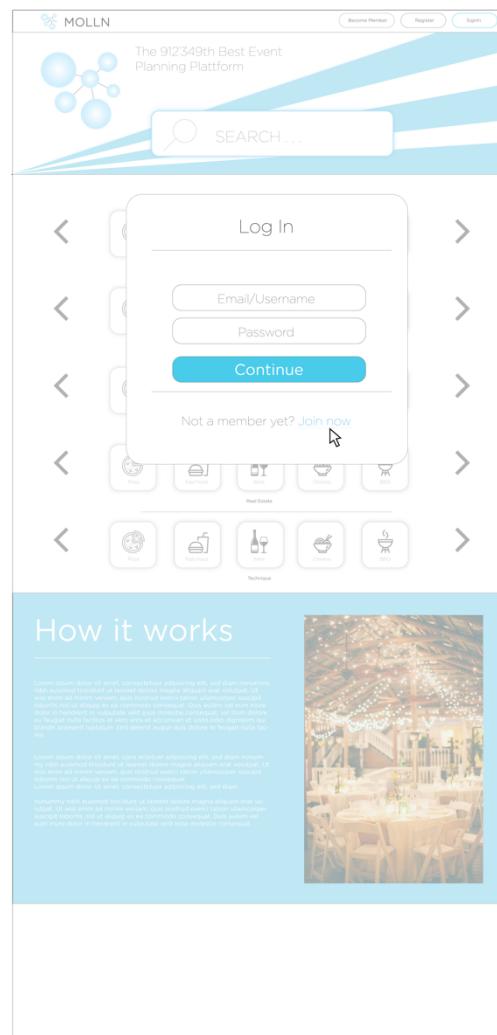
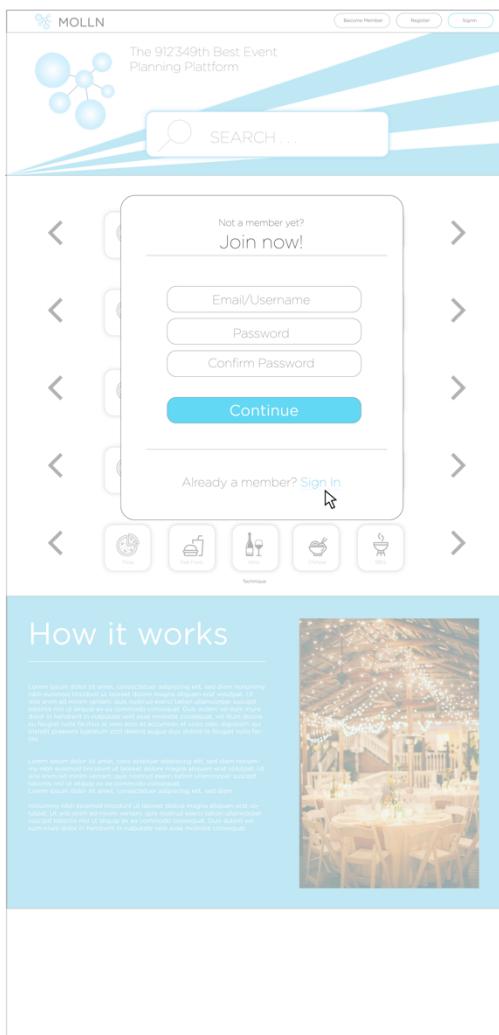
Price

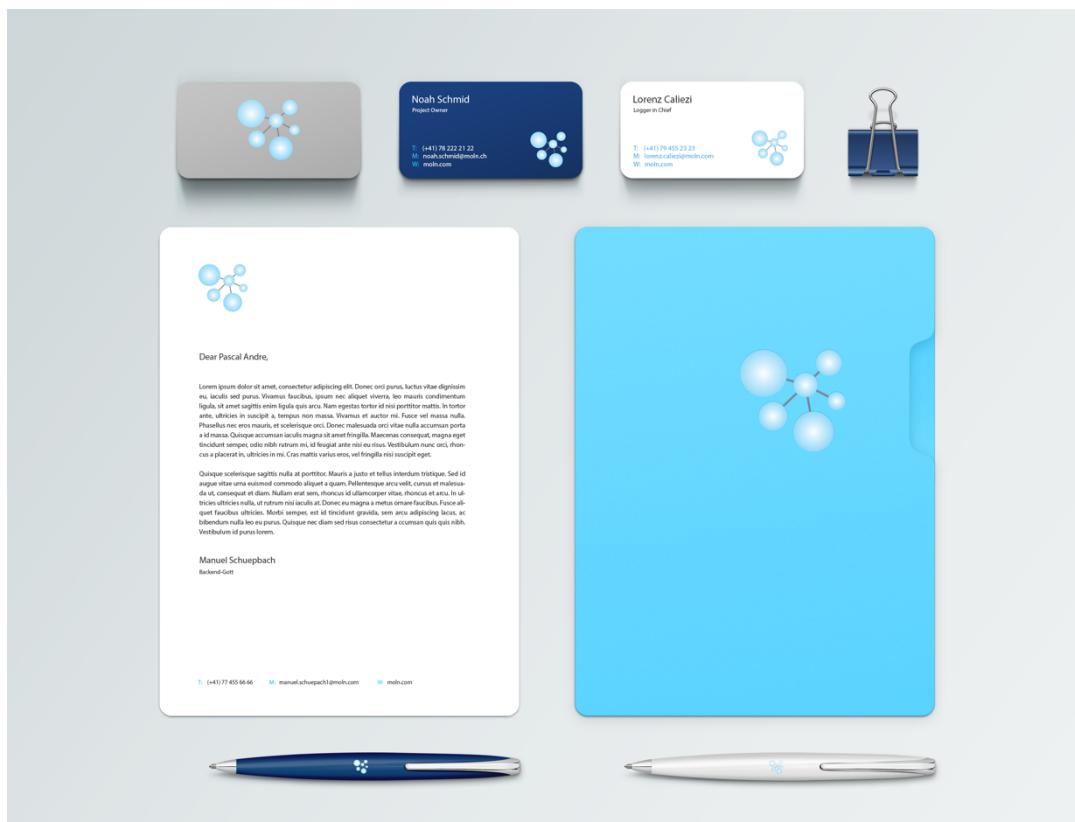
Location



The image displays the MOLN mobile application interface across four screens:

- Home Screen:** Shows the MOLN logo, a molecular icon, and a welcome message: "Welcome to the 1435th best Event Planning Platform". It features sections for FOOD AND BEVERAGE, INFRASTRUCTURE, REAL ESTATE, and SERVICES, each with icons for Fast Food, Wine, and Chinese.
- Food and Beverage Category Screen:** A detailed view of the F&B category, listing items like "El Manu" (Category: F&B, Subcategory: Wine). It includes a delete button and a "swipe()" action.
- Infrastructures Category Screen:** A detailed view of the Infrastructure category, listing items like "El Manu" (Category: F&B, Subcategory: Wine). It includes a delete button and a "swipe()" action.
- Real Estate Category Screen:** A detailed view of the Real Estate category, listing items like "El Manu" (Category: F&B, Subcategory: Wine). It includes a delete button and a "swipe()" action.
- Services Category Screen:** A detailed view of the Services category, listing items like "El Manu" (Category: F&B, Subcategory: Wine). It includes a delete button and a "swipe()" action.
- Seller Profile Screen:** Shows a summary of seller offers: "Delete Offers", "Add categories", and "Become a seller". Below is a list of offers, each with a delete icon and a "swipe()" action. The offers include "El Manu" (F&B, Wine), "Lorenz" (F&B, BBQ), "Noah" (F&B, Wine), "O Manu" (F&B, Italian), "Adrian" (F&B, BBQ), "Manuel" (F&B, Chinese), "El Manu" (F&B, Wine), and "El Manu" (F&B, Wine).
- Contact Information Screen:** Displays the MOLN logo, contact information ("Contact: info@mолн.ch"), copyright ("© MOLN - 2019"), and UniBe branding.





Meeting Protocol

Sitzung vom 13.11.2019

Milestone 3

Task	Wer	Priorität
1) Google Maps Location	Oli	2
2) Review System	Löru	1
3) Search	Adi	2
4) Notification System	Löru, Manu	1
5) Filter	Adi	2
6) Kamera auf Mobile	Oli	1
7) Zahlungen (PayPal) (optional)		3
8) Bestellungen via Website	Adi, Noah	3
9) Produkte bearbeiten	Löru	3
10) Chat (optional)		3
11) Logo Animation	Oli, Adi	2
12) Punkte der Vorlesung (optional)		

Presentation (10 mins)

- Intro
- Demo mit Features
- Jeder 2 Minuten? Aufteilung finden, wer redet

Project Deliverables

- Testing Frontend (Manu)
- Documentation Backend
- Backlog, SRS, Sprint Schedule
- Biweekly Report
- ReadMe

ToDo BusyWork

	ToDo
Manu	Reviews für bestehende Produkte erstellen
Oli	Icons für Kategorien finden
Löru	Subkategorien erstellen
Noah	Produkte erstellen
Adi	Frontend Dokumentation

Disclaimer

We did not involve all the protocols, design drafts and documents we created. If whoever it may concern would like to see further resources, please contact us and we are happy to provide more.