



Webengineering

2017-06-26

Dr. Michael Lesniak

mlesniak@micromata.de

Roadmap 2017-06-27

- Solution to exercises

- Code

- Outlook

- 2017-07-03

- 2017-07-10

- 2017-07-17

- Lecture project

Solution to exercises

Authentication

- Add a new component for authentification
 - email
 - password
- On successful submit
 - set global axios header
 - what happens on reload?
 - how can you fix this? Hint: react-cookies
- Implement Logout mechanism
- Show a simple warning message if login was not successful

Routing

- Idea
 - Single Page Application
 - Components on different endpoints in the browser
 - Solution: react-router
- install react-router (similar to axios)
- Move post_create component to a new route /post/new

Post details

- Show all data of a post on a page /post/{id}
 - author
 - comments
- Add functionality to add comments

Outlook

- 2017-07-03
 - Component communication
 - CSS with Bootstrap
- 2017-07-10
 - TBD
- 2017-07-17
 - TBD

July

	M	T	W	T	F	S	S
26	26	27	28	29	30	1	2
27	3	4	5	6	7	8	9
28	10	11	12	13	14	15	16
29	17	18	19	20	21	22	23
30	24	25	26	27	28	29	30
31	31	1	2	3	4	5	6

Code

Internationalizaton

- I18N
- Translate user-visible text
- Do not translate backend
- Just a showcase to show you how it's done ...



Lecture project

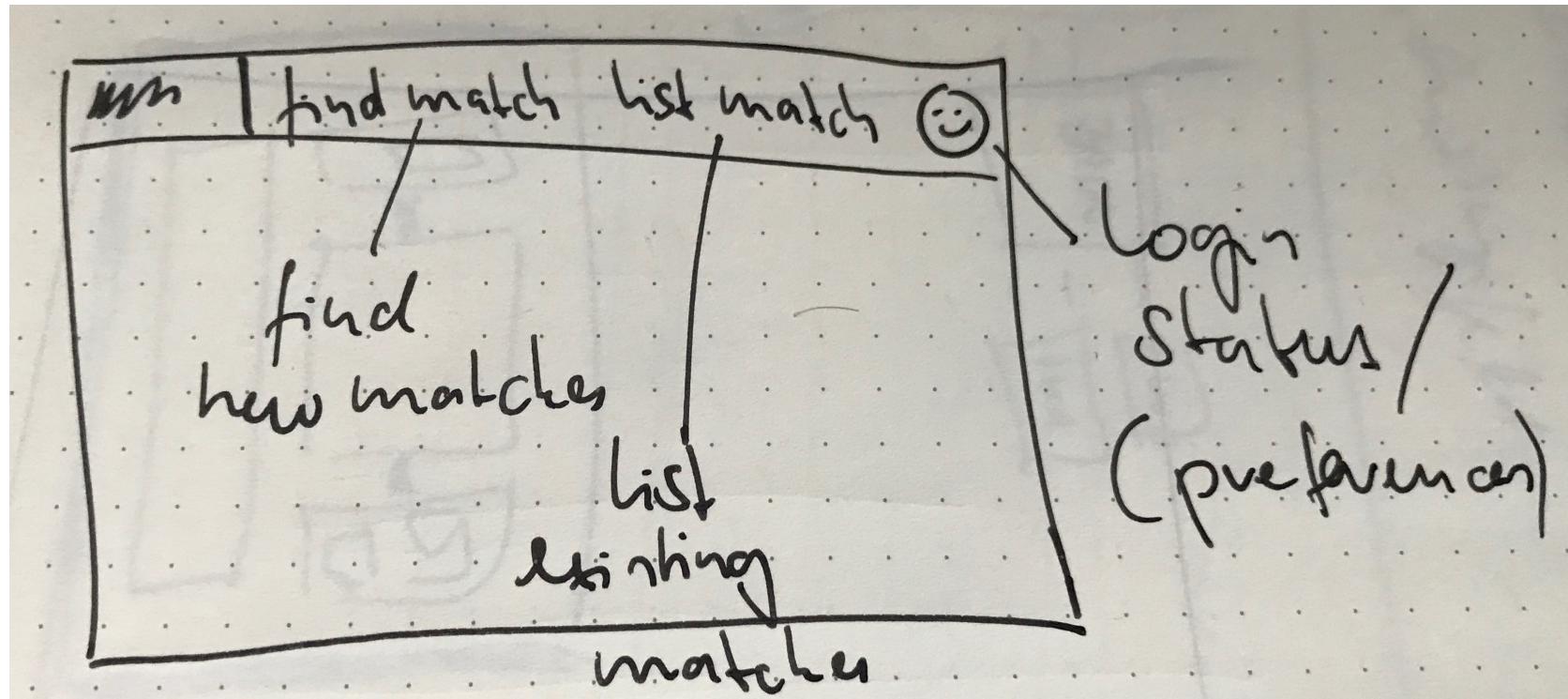
Idea >> fratcher

- **friend matcher**
- Find friends with similar interests, humor, etc...
- A fully functional web application which is deployed online
- Similar to a well-known dating app but without those sketchy pictures

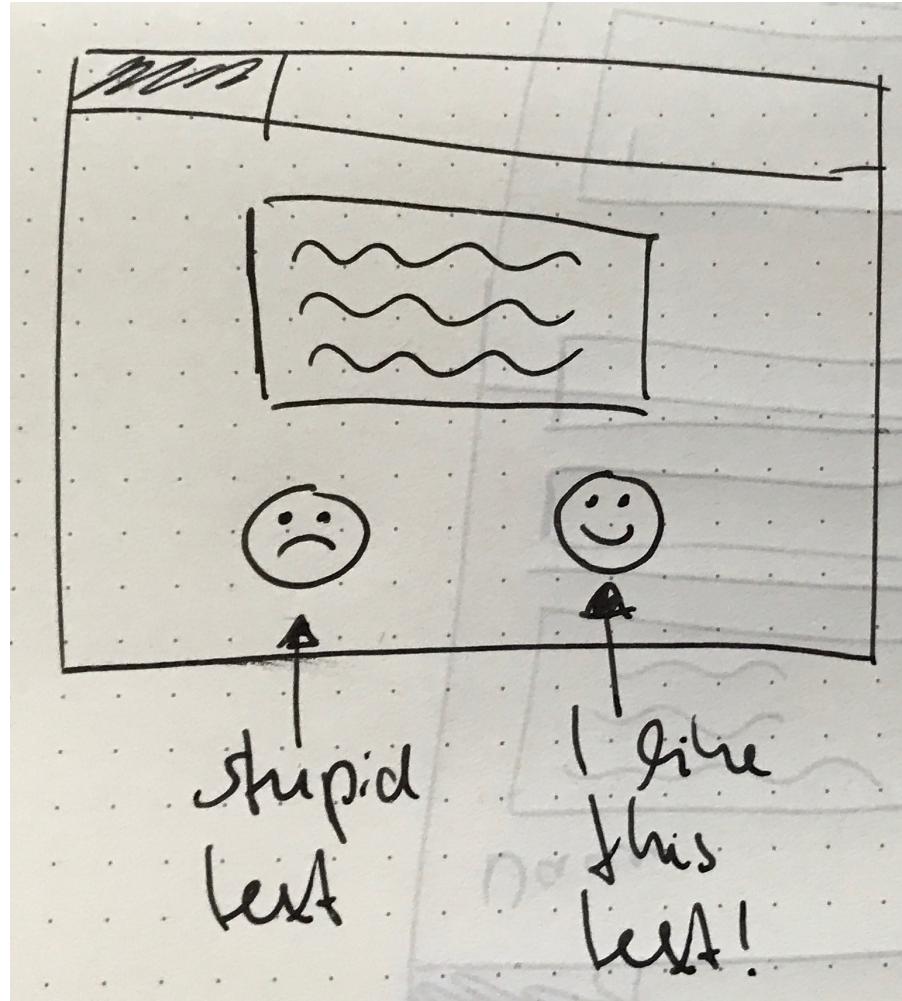
- Balances
 - technical complexity and trivial copying of lecture code
 - easy and more complex functional requirements
 - presentable frontend and complex UI



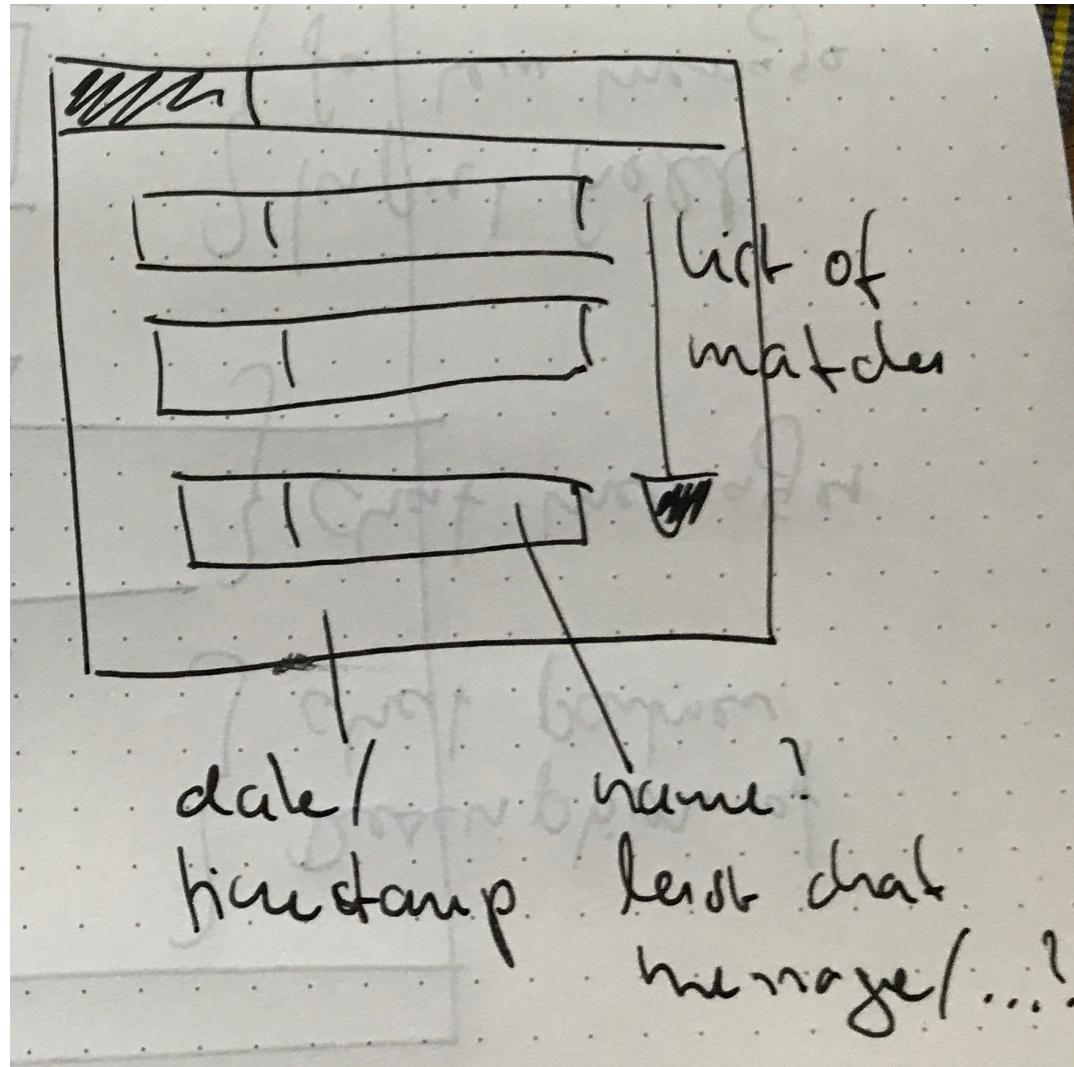
Design idea >> menu



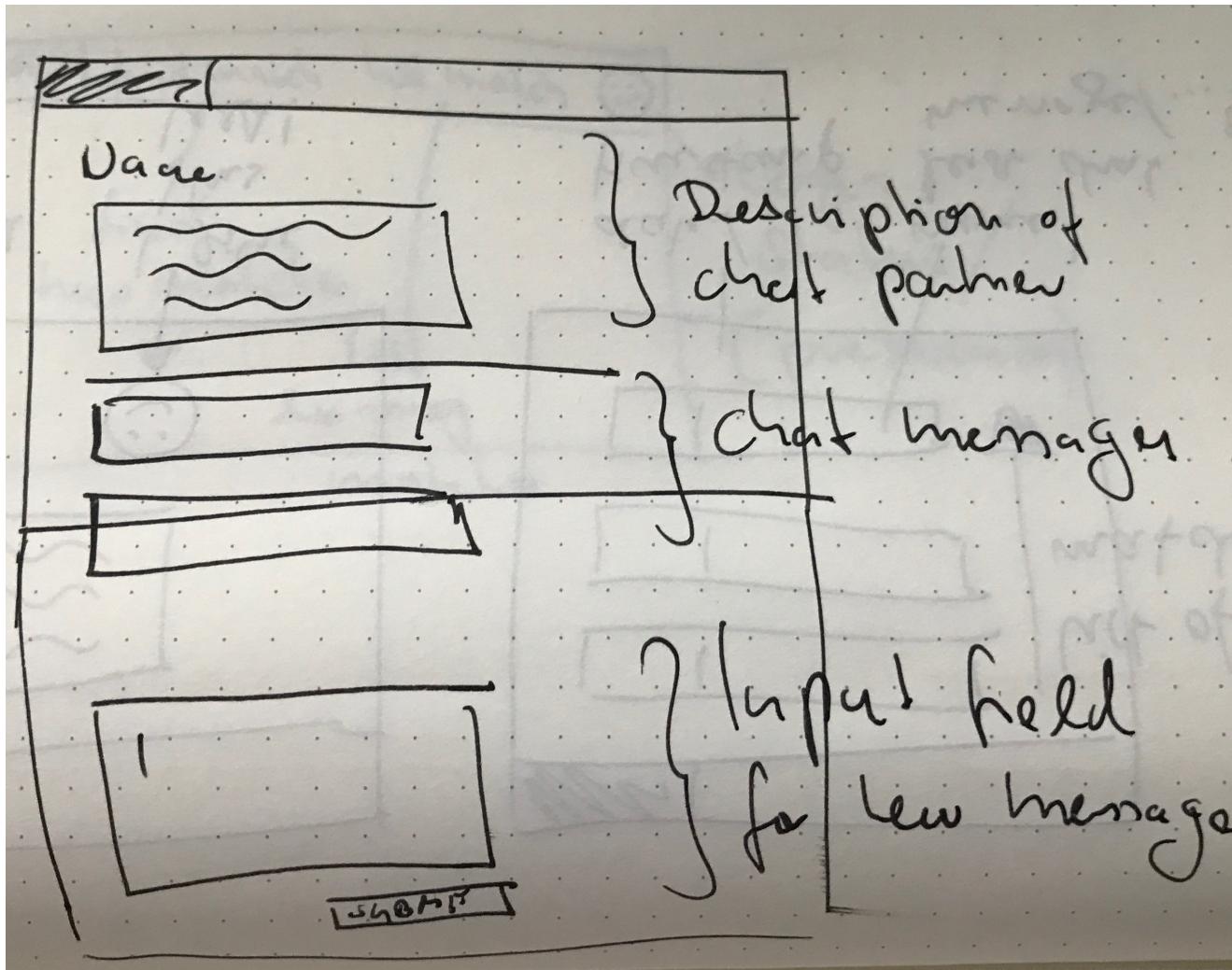
Design idea >> match making



Design idea >> match listing



Design idea >> chatting



Functional components

Match making

- View single text of a user
- Like or dislike his text

Match viewing

- View a list of all matches
- Optional: CRUD operations

Match communication

- Leave mutual messages
- Does not have to be in real-time

User management

- Sign-In
- Sign-Up
- Optional: User preferences

Technical requirements

- Logging of all functional operations
- Deployed and working on Heroku

Goals and non-goals

- working web application
- implementation of the functional requirements
- I18N support – but one language configuration suffices
- readability
- well-structured application
 - backend
 - frontend

- mobile application
- mobile responsiveness
- scalability to hundreds or thousands of users
- the most beautiful application in the world ...

- Add optional features after the requirements are met
- Might give bonus point, decided on a case-by-case basis

Technical matters and artifacts

- When I start the application in development mode, the database must be filled
 - with test users in different matching states
 - some chats
- Technology stack
 - Backend with Java, Maven and Spring-Boot
 - Frontend with React
- Submission until 2017-08-28 23:59 to mlesniak@micromata.de
 - archive with all code
 - including .git for commit history
 - should build with maven
 - link to deployed Heroku instance



necessary for
fair comparison

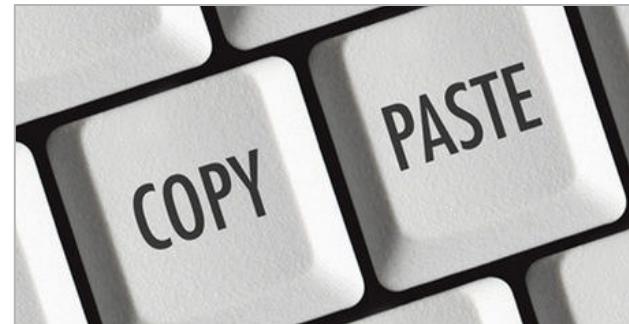
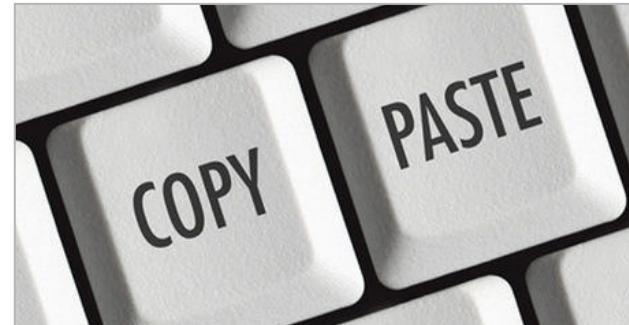
Artifcat >> README.md

- Build and deployment instructions, ...
- Should contain a list of
 - all relevant design decisions you made with their respective advantages and disadvantages
 - features you are especially proud of
 - anything you want to tell me :-)

A good README.md might raise your mark
since I can see motivations behind your decisions

Copy and paste?

- Please do not copy and paste code from your fellow students
- I will [probably ;-)] catch you
- If I will, both students will fail this class – no discussions, no exceptions
- BUT
 - you can fully use all code that is in the lecture repository
 - you can **discuss your ideas** with your fellow students
 - if you find code online, e.g. on StackOverflow, Tutorials, etc. add a reference as a comment



Tipps

- Step by step
 - Start small
 - Micro-commit (might be a personal preference)
- Use the next three lecture's exercises to
 - discuss design ideas
 - show me code you might have written

Organisation

- The lecture project's requirements will be constantly updated in a separate document.
- A changelog with clarified requirements will be posted on the GitHub forum.
- Please post ALL questions on the forum such that other students might profit, too.
- You can contact me by mail too. If your question is not private I might insist that you repost your question on the forum before I answer it.