Documentation & Project Diary

Innovation Lab 1
Year 2021

Project: **WebApp for social Impulse analysis**

Team: **Group 14**

1. General Information

Project name: WebApp for social Impulse analysis

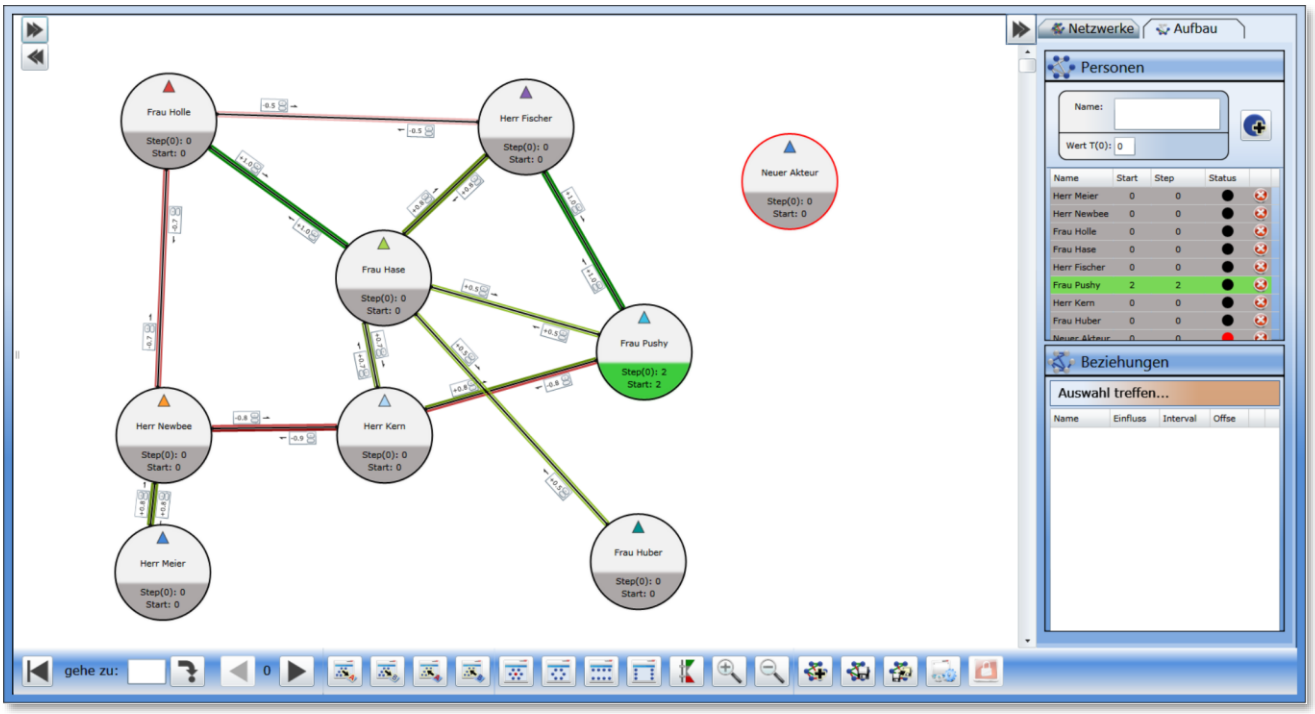
Supervisor: Florian Eckkrammer

Innovation Lab 1, winter term 2021

Project team:
Brandenburg Jonas, if20b243@technikum-wien.at
Dohnalek Raphael, if20b075@technikum-wien.at
Duskanich Markus, if20b078@technikum-wien.at
Duskanich Michael, if20b077@technikum-wien.at, project manager
Meindl Tobias, if20b125@technikum-wien.at

Management Summary of the Project

This project concerns itself with the topic of social impulse analysis. Social impulse analysis is all about recording, visualizing, and evaluating social connections within a group.
A software developed in 2010 already provides a basis for this project:



Name:
Wert T(0): 0

Name	Start	Step	Status	
Herr Meier	0	0	●	
Herr Newbee	0	0	●	
Frau Holle	0	0	●	
Frau Hase	0	0	●	
Herr Fischer	0	0	●	
Frau Pushy	2	2	●	
Herr Kern	0	0	●	
Frau Huber	0	0	●	
Neuer Akteur	0	0	🔴	

Auswahl treffen...

Name	Einfluss	Interval	Offse

Figure 1 The 2010 developed A-SIA interface

As it turned out, the software, *A-SIA*, became outdated, leading to the main task: Redesigning the frontend with a modern framework, as well as outsourcing resource heavy computation to a server-side application.

Framework Conditions and Project Environment

All the following Framework decisions are subject to change after next week's requirement specification.

This project requires a full stack development. As the project might be expanded in the future, it must be well documented.

Frontend:

- Angular
- Jasmin / Karma
- Compodoc
- SCSS

As there already exists prior knowledge with the MEAN stack, Angular was chosen as a frontend Framework.
Angular natively includes Jasmin and Karma, being the obvious choice for unit testing.
For automatic code documentation, Compodoc was chosen – a popular open source, Angular-friendly tool.

Backend:

- C# (with DocFx) or NextJs
- Database TBD

There is still some speculation, as to which backend language will provide the most value. The two main choices are C#, and NextJs. This will be determined after further research.
With social impulse analysis being an application suitable for graph visualization, a graph-based database might pose an obvious solution. This, however, will also be determined after completing the requirement meeting.

Semester-Roadmap

TBD after detailed requirement analysis until next meeting.

Collaboration & Tooling

- ALM - Azure DevOps
- Communication - Zoom
- VC - GitHub (Private, hosted by Michael Duskanich)
- Mockups - Figma

2. Brief Description of the Project

- What is the theoretical background of this app?
    - A-SIA was a software for modeling teams and social networks. The tool supported the recording of the influencing structures, which could either be entered manually or be automatically generated from answered questionnaires. The relationship patterns collected in this way were graphically displayed and support the processing of social and group dynamic phenomena in teams. Furthermore, it simulated *What-If* scenarios, visualizing group dynamic effects.

- Why does it have to be reworked?
    - A-SIA was developed on *Microsoft Silverlight*, now unsuitable for modern app development. Moreover, a lot of computation is done client-side. A visual refactoring, as well as a redistribution of server-side calculation is required.

- Who is the target audience / customer?
    - A-SIA was used by a multitude of different companies. This project, however, does not necessarily need to be deployment ready with multiple pricing options. The focus lies on a well-documented and futureproof web application.

- What are the main features of the application?
    - *Visualizing a sociogram:*
      A sociogram consists of actors and relations. Those will have to be created manually (and possibly generated via an auto generated form) within a user interface, which is displayed as an undirected graph. Trust, mistrust, irritation, insistency and *echodampening* are metrics that can be helpful for a more descriptive display. Different arrangements and filtering options should be available.
    - *Simulation:*
      The prominent premise of this application is, that it can simulate social dynamics. For example, an actor can be selected to propose an idea. Then, the user of the software can step through the simulation.
      The algorithm for this simulation will be provided.
    - *Backend:*
      Data (sociograms) can be stored and loaded from a database. Furthermore, the computation will be done server-side, reducing the load on the client.

- What are non-targets of this project?
    - Improving the simulation algorithm.
    - A mobile first approach
    - Keeping the calculation client-side

3. Specification of the Solution

< Once the order has been clarified (pre-project phase), you start the project implementation. Create a specification of your solution parallel to the implementation of your project across the sprints!

Before each sprint, at least those details must be specified that you will implement in the next sprint. Use techniques such as writing epics & user stories and build a product backlog (use the course content from the course Agile Project Management).

For the specification, generally use visualization techniques that fit the task at hand. For example, in addition to the mockups and user stories, database diagrams, class diagrams, or sequence diagrams (representation of temporal processes) can also be useful.

Usually, you go from rough to detail. The structure of this section can be as follows:

- *System environment: Describe the delimitation of the solution to be implemented (system boundaries)*
- *Features (functional requirements): All required solution properties - in the case of software usually the features or a description of these as user stories or similar)*
    - *Create screen mockups of all essential UI views!*
- *Interfaces: All relevant interfaces of your solution.*
- *Quality characteristics, technical requirements (non-functional requirements): performance, scalability, availability, usability, information on architecture and expandability, etc.*
- *Other "not clear at first glance" but essential solution features!*

Agree with your supervisor how the specification should be structured!

Ask whenever you feel that there may be a misunderstanding, different expectations, or if you did not fully understand a requirement! >

4. Delivery

< In this section you describe the scope of delivery of your solution and everything you need to pass it on to a customer or another software team (in practice this is often referred to as "hand-over to operations" when the solution enters the operational phase).

- *Final solution or solution components including source code*
- *System architecture and data storage*
- *List of any required licenses and information about copyrights (e.g. if third-party software / frameworks or similar were used).*
- *Any hardware specifications*
- *Description of how to install your solution including a list of all components to be installed, installation procedures, migration of databases, etc.*

The content of this section is mostly project-specific. Agree with your supervisor what exactly this section should contain! >

5. Our Project Diary

< This section should be a kind of diary in which you record “what happened in our team in the project”. Use photos from your meetings, take photos of any reflections from whiteboards. Take screenshots.

Describe in short text sections which problems there were, which challenges were solved, what was "cool" in the project, etc.

ATTENTION: Create this section continuously (!) Parallel to the project and not only at the end on the last evening before the project is submitted! This enables your supervisor to understand why something worked particularly well or not so well, why there was great progress or delays, etc.

In practice, such a diary is used as the basis for a project retrospective and team feedback rounds.

Tip: Meet each other at the end of the semester and let your project "pass in review" over a good project closing meal: This is a good opportunity to discuss what you have experienced again and for the future or what you have learned in the next semester and take the Innovation Lab with you! >