

Demo Day Preparation

Dirk Riehle, Univ. Erlangen

AMOS B07

Licensed under [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

Documentation

Documentation has been initialized and is being maintained

The AMOS Demo Day

The demo day is the final day of the course

- The demo day is organized as a fair (“Messe”)
- Student teams show the results of their project
- Audience are industry partners and fellow students

Demo Day Process

Opening (20 min.)

- General introduction
- 1 slide, 1 min. introduction by each team (one speaker)

Demos (100 min.)

- After the introduction, we split up into the different projects
- One presentation booth (in-person) / demo room (online) for each project
- At least one person from each project needs to be ready to demo
- Demo day participants (including students without booth duty) roam around

Closing (5 min.)

- We come together in the main room to say goodbye

Demo Day Schedule

Time	Duration	Responsible	Title	Room
10:15	10 min	Riehle	Introduction	Main room
10:25	10 min	Teams	One slide summary	Main room
10:35	20 min	Teams	Demo	Demo rooms
10:55	20 min	Teams	Demo	Demo rooms
11:15	20 min	Teams	Demo	Demo rooms
11:35	20 min	Teams	Demo	Demo rooms
11:55	20 min	Teams	Demo	Demo rooms
12:15	5 min	Riehle	Conclusions	Main room

In-Person Presentation Booth

You (students) are given a “presentation booth”

- The booth is a table plus pinboards for posters
- Please pin your posters to the pinboards and be ready to discuss them

You will demo your work in this booth

- Demo your project, usually using your laptop
- Explain your work using the posters
- Support the demo with a slide deck
- Have the demo day video ready as a backup

Online Demo Room

You (students) create a Zoom room from the project demos

- Please create the Zoom room and share with the teachers
- We will point demo day participants to your demo rooms

You will demo your work in this Zoom room

- Demo your project using your laptop
- Explain your work using a slide deck; should contain
 - One slide on product management
 - One slide on software development
 - A team photo slide (can be screenshot)
- Have the demo day video ready as a backup

Use of Corporate Identities

Please use your university logo

Please use your team logo

Please use your industry partner logo, but ask first

Example Posters 1 / 2

**FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG**

**JOWBANA**



Personalfragebogen 2.0

Personnel Questionnaire Automation



Personalfragebogen 2.0

Benutzername
[Benutzername eingeben]

Passwort
[Passwort eingeben]

Remember me! ☐ (Passwort speichern)

© 2015 ARSO GROUP 4



Employees - Overview

Benutzername: [Benutzername eingeben]

Passwort: [Passwort eingeben]

Remember me! ☐ (Passwort speichern)

© 2015 ARSO GROUP 4





Personalfragebogen 2.0* is a personal data management software solution, supporting companies of any size in hiring new employees more efficiently.

The product improves the hiring process by automating the collection of personal data during the hiring procedure, and provides aid in managing the collected data.

Demo Day
Friedrich-Alexander Universität
Erlangen
Wednesday, 15 July 2015 10.15-11.45



*Personalfragebogen 2.0 is a joint project between FAU's OSR Group and DATEV eG.

**FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG**

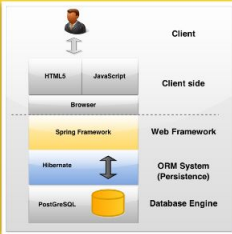
**JOWBANA**

Personalfragebogen 2.0

Personnel Questionnaire Automation



Software Architecture



```
graph TD
    Client[Client] <--> Browser[Browser]
    subgraph Client_side [Client side]
        Browser
    end
    subgraph Web_Framework [Web Framework]
        Spring[Spring Framework]
    end
    subgraph ORM_System [ORM System (Persistence)]
        Hibernate[Hibernate]
    end
    subgraph Database_Engine [Database Engine]
        PostgreSQL[(PostgreSQL)]
    end
    Browser <--> Spring
    Spring <--> Hibernate
    Hibernate <--> PostgreSQL
```

Technology

Name	Function
Spring Framework (4.1.6)	Java based Web Framework
Java SE (7u79)	Fundamental Platform
HTML5	Client-side core technology
Selenium (2.45.0)	UI Testing/Integration Testing
JUnit (4.12)	Java Unit Testing Framework
Hibernate ORM (4.3.9)	ORM System for persistence
PostgreSQL (9.4.1)	Database Management System
Tomcat 7.0.61	For local deployment

*Personalfragebogen 2.0 is a joint project between FAU's OSR Group and DATEV eG.

Example Posters 2 / 2



CroudTrip!

sponsored by Elektrot

The CroudTrip! application wants to revolutionize the car-ride-sharing market with its easy, user-friendly and highly automated way of organizing shared Trips!

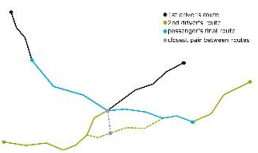
The Product



- Offer and join shared Trips at short-notice!
- For drivers: Easily find passengers on the way you are going anyway ... and earn money with it!
- For passengers: Reach your destination comfortably!
- We will automatically match you to the best offer in real-time!
- Simply check-in and check-out of your Trips using NFC on your device!
- No direct Trips? No problem - Join a SuperTrip! with multiple drivers!

The Concept

SuperTrip!



- Car driver's route
- Driver reports route
- Passenger's route
- Connection point

Multiple Passengers





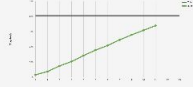
- Driver's original route
- Modified route by first passenger
- Modified route by second passenger

- Combine multiple offered routes to serve passengers even if there is no direct connection available
- Find routes which can pick up a passenger from his start position or drive to his final destination
- Subdivide those routes, compute the closest pair of those waypoints and use it as a "connection point"
- If the distance of the closest pair is too large, start a recursive matching process with these two waypoints
- Match multiple passengers with one driver who will pick them up and bring them to their destinations in an optimal order
- Optimal order is constrained by given internal order of each waypoint pair, because each passenger has to be picked up before the driver reaches his destination location
- Compute optimal order by solving the Travelling Salesman Problem via Brute Force (max. 4 passengers)

The Interactions


Driver	Server	Passenger
Driver posts an offer	1. New Trips are stored in the server's database to be matched against incoming passengers' queries.	2. Passenger sends a query
Driver receives pending requests	3. The server starts a matching process with available offers. Additional constraints such as the maximum waiting time for passengers or diversion from the driver will be taken into account.	3. Passenger receives Trips
Driver accepts passenger	4. Even Trips with multiple drivers are possible.	4. Passenger joins the Trip
Passenger status updates	5. If no offered trip matches a passenger's query, the search will run in the background and notify the passenger about new available Trips using Google Cloud Messaging (GCM).	5. Driver status updates
Driver status updates	6. The status of the driver and the passenger is changed and displayed in real-time.	6. Driver status updates
	7. Each route is computed using Google Directions API. Those routes are passed to the client to be visualized for both drivers and passengers. Each passenger that should be picked up is shown as a single marker on the driver's route.	

The Process




- Total # of story points: 370
- Development Speed: 30.9
- 13.23% of total effort used for bugfixing
- Slack as main communication tool
- Integrations for Travis CI, Github and Crashlytics
- Total # of Commits: 727
- Lines of Java Code: 15362
- Lines of Comments: 3938


The Team





The Sponsor



The University





Twenty Minutes of Fame

At	What happens / to do
0min.	Demo room opens, participants stream in
1min.	Demo starts
10min.	Demo finishes, discussion starts
19min.	60 second countdown to room closing starts
20min.	Room closes, everyone is pulled back into the main room

Demo Preparation

Dos

- Have a clean user interface
- Use domain terms and examples
 - In the user interface (labels, titles)
 - In the stories you tell
- Have a story to tell, for example,
 - A day in the life of ...
 - A workflow example
- Make the demo data reentrant
 - You will have to start over repeatedly
 - You want to start at the same point

Don'ts

- Use “test” or “help” as labels
- Not follow the advice on the left

Demo Execution

Have two people ready to demo

- One explains what is going on (talks to people)
- One demos the software in line with story

Thank you! Any questions?

dirk.riehle@fau.de – <https://oss.cs.fau.de>

dirk@riehle.org – <https://dirkriehle.com> – [@dirkriehle](#)

Legal Notices

License

- Licensed under the [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/) license

Copyright

- © Copyright 2023 Dirk Riehle, some rights reserved