Contents

1. 2018

1.1 October

HW Week 1: Vision Tenniskönig (2018-10-12 14:26)

Hi guys,

We want to tell you about our idea, on which we will be working the next year.

Our plan is to implement a website for a local tennis club. The main goal of this site will be to motivate the kids to play more tennis.

The Project will be created as a website, with the use of the Vaadin-Framework.

There will be different user groups, to manage the website and the scores. Which means that the kids will be able to enter their playing times and their scores and the club officials will be able to administer the kids' entries.

They get points for playing tennis and then can compare on a high score.

The last years the project was carried out on paper but to make it more comfortable to use we will transform it into a website.

Feel free to follow us on our journey in developing.

Unknown (2018-10-12 14:39:24)
Hey Guys,

I really like your idea because I think it's important for kids to do some sports.
I am curious how you realize that idea and I'm looking forward to hear from you.

A Web App is a good idea because it's easy to use on all platforms, but do you plan to bring that stuff also as an app?

Hear from you,
Benni

Tenniskoenig (2018-10-12 15:42:18)
Hi Benni,

thank you for your input.
There will be no mobile app. But to simplify the use of our website on smartphone there will be a mobile view for mobile browsers.

Kind regards,
Your Tenniskönig-Team

Anonym (2018-10-12 15:24:56)
Hey guys,

we like the idea to implement a function on the website to display the playing times and scores of the children. That is very motivating. Maybe you have to think about the structure of your blogpost. There are only a few information about the structure and content of the website you are going to build. Also did you think about implementing a function to reserve the tennis courts on the website?

Greetings
Find your Farm / FyF Team

Nico@KeepTrack (2018-10-12 15:34:16)
Hey guys,

really good idea you have there, I would have really liked that back when i played tennis.
Will you also make it possible to organize personal tournaments or can I just add single games and the scores of it?
And will you also make a mobile app or will I need to just use the website on my mobile browser?
Keep up the good work.

Greetings, Nico@KeepTrack

Tenniskoenig (2018-10-12 15:48:14)
Hi Nico,

thank you for good ideas.
There will be no possibility to organize personal tournaments, the focus is on single games.
Also you will have to use our website on your mobile browser since there will be no mobile app. But to make this as simple as possible there will be an extra mobile view of our website.

Kind regards,
Your Tenniskönig-Team

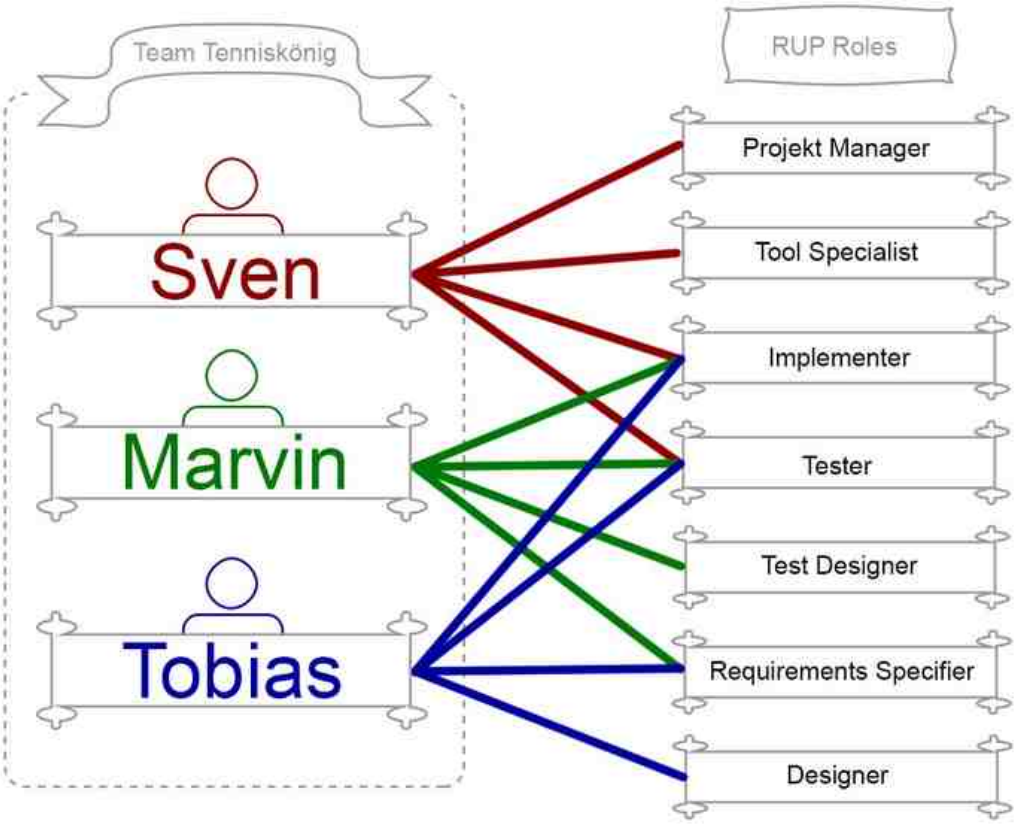
HW Week 2: Team/Roles/Technology (2018-10-14 18:55)

First lets talk about our team. By using the Raional Unified Process (RUP) terminology we assined roles to each teammember. In the graphic below you can see which member is responsible for which role.

If you want to learn more about the RUP Roles you can look them up here:

[1] https://www.ibm.com/developerworks/rational/library/apr05/crain/

[2]



Projekt Manager:

– plan, track, and manage risks for a single iteration

– make go/no-go decisions

Tool Specialist:

- Create guidlines for specific tools

Implementer:

– code a single set of classes or a single set of class operations

Tester:

- execute

specific test cases

Test Designer:

-

implement automated portions of the test design for the iteration

Requirements Specifier:

- Details a single set of requirement use cases

Designer:

– details the analysis and design for a single set of use cases

With the RUP terminology we splitted the process of developing in four phases:

1.

Inception - Idea of the project. Determine what is needed.

2.

Elaboration - Considere the architecture and required resources of our project.

3.

Construction - The project is developed and completed.

4.

Transition -Software is released. Small adjustments based on feedback

We gathered all technologies, we are planning to use:

.

IntelliJ (IDE)

.

Vaadin-Framework (frontend/backend)

.

Tomcat (Server)

.

GitHub (versioning)

.

Jira (management)

As our Project Management Tool, we will use Jira.

In Jira the issues will be classified in sprints (phases) and issues(workflow).

This link will lead you to our issue list:

[3] https://jiratinf16b4.it.dh-karlsruhe.de:8443/projects/TEN/issues/TEN-17?filte r=allopenissues

Stay tuned and see you next week.

Team Tenniskönig

1. https://www.ibm.com/developerworks/rational/library/apr05/crain/
2. https://3.bp.blogspot.com/-wbM3Ont7nf4/W8NN93qlTfI/AAAAAAAAAAM/86M9WcPf5MwOYAFqG_ixTSkdSMwQ111wwCLcBGAs/s1600/RUP%2BRoles.jpg
3. https://jiratinf16b4.it.dh-karlsruhe.de:8443/secure/RapidBoard.jspa?rapidView=19&projectKey=TEN&view=planning.nodetail

---

Anonym (2018-10-16 09:10:05)
Hey guys!

You clearly defined your team roles using the RUP. It would be very interesting if you could describe

the responsibilities of each team memeber a bit more detailted. Moreover I've never heard of the Vaadin-Framework so I'm very exited to read more about it in the future.
I also tried to have a look on your issue list, but authentication is reqired.

Best Regards,
Sven

Tenniskoenig (2018-10-18 10:39:40)
Hey Sven, thanks for your comment.
Our issue list is now open for public!
Also we described the responsibilitie more detailed.
Thanks for the feedback
Greeting Tenniskönig Team

MNZ-Team (2018-10-16 23:21:43)
Hey TENNISKÖNIG-Team,

we are the MNZ-Team from the TINF17B2 course.

The thought process used to determine the technologies you've provided is very interesting.

Great graphic you use to show role distribution, though a bit small.

We would very much appreciate if you could comment on our next Blog entry.

greetings MNZ-Team

Tenniskoenig (2018-10-17 10:51:13)
Hey MNZ-Team,
thanks for your comment.

If you want to see a bigger version of the graphic, just click on the small one :)

See you soon
Greeting Tenniskönig

Anonym (2018-10-17 14:57:54)
Hey guys,

great to see your new Blog entry. The structure is better now, so you have thought about it. This is very nice. We like that you split up the process of developing. But there are only a few informations. Maybe you can give more informations like little steps in the phases. For example in the construction phase. Are you going to test the Software before releasing it?

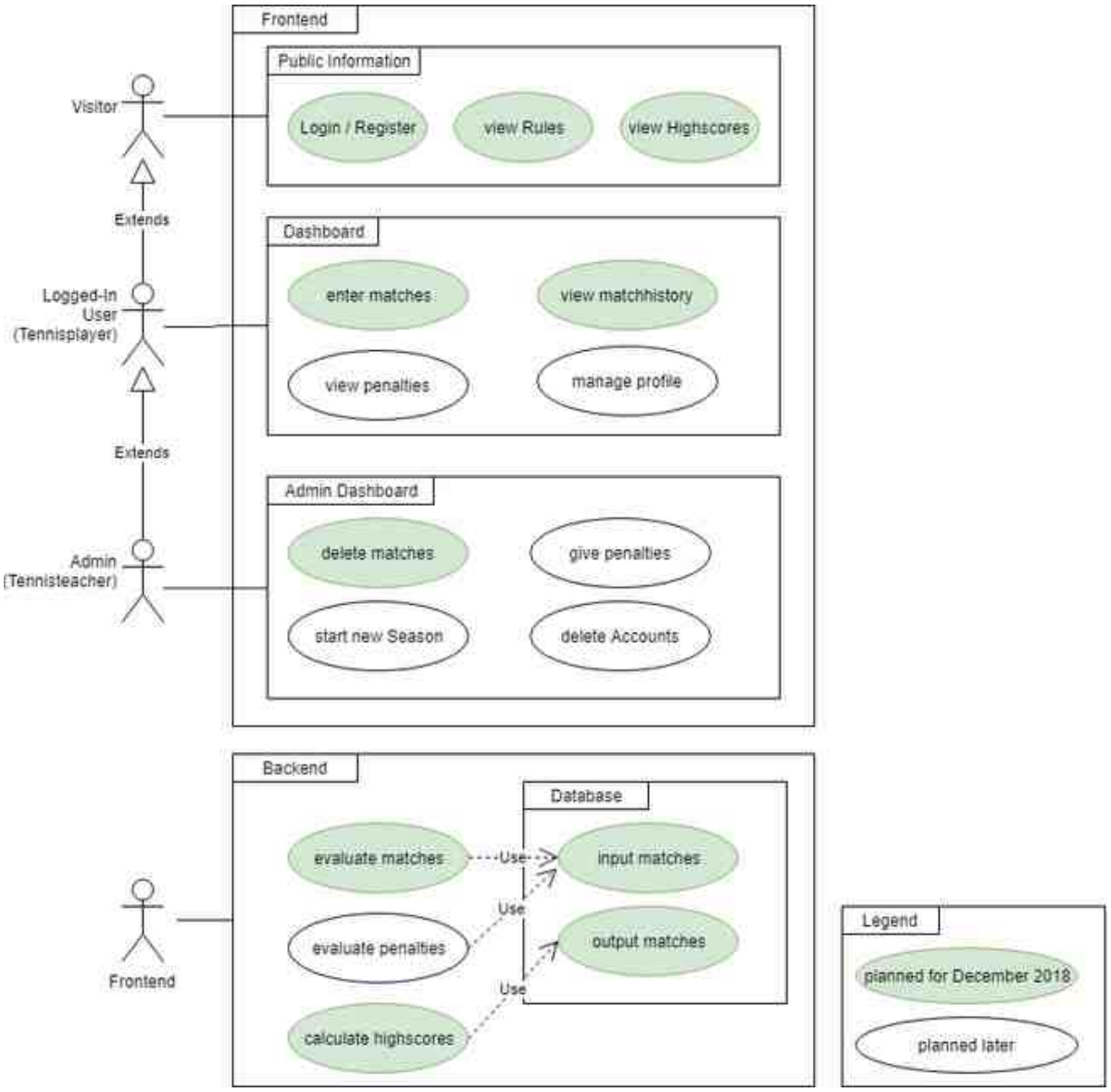
Keep up the good work, we are really looking forward to your results in the next weeks.

Best regards
Simon, FyF-Team

HW Week 3: Software Requirements Specification (2018-10-21 19:05)

Hey guys,

this week we worked on our Software Requirements Specification (SRS) document and created an Overall Use Case Diagram (OUCD).



You can always find the newest version oft these documents at the following places:

- [1]Software Requirements Specification
- [2]Use Case Diagram

1. https://github.com/tenniskonig/Dokumentation/wiki
2. https://github.com/tenniskonig/Dokumentation/blob/master/Use_Case_Diagram.png

Anonym (2018-10-22 10:06:23)
Howdy boys,

your Use-Case-Diagram is very compact yet clear.
With the SRS it became clearer what you are planning and how you want the app to work.
We are looking forward to learn more about your app in the progress.

Best Regards,
Luis@DHBWieWarsEssen

Anonym (2018-10-22 10:27:07)
Hey guys,

I like your OUCD. It's clear structured and easy to understand. But I have one question. Has only the admin permission to delete profiles. And how can the users delete their accounts? Do you have to write an E-Mail to the admin or is der just a Button to click?

Keep up the good work!

Greetings
Simon, FyF-Team

Nico@KeepTrack (2018-10-22 10:30:39)
Hey guys,

good work on your SRS and UCD this week.
Your UCD is well structured and gives a good overview over your use cases. We also liked your use of colours, so we can see, what can we expect from your project in the next months.
Furthermore, your SRS shows the functionality of your website very well, so in your futere drafts of the SRS maybe you could focus more on the technologies you are going to use and how they interact.
Again, good job this week, keep up the good work.

Greetings,
Nico@KeepTrack

HW Week 4: Use Cases (2018-10-24 13:36)

Hi guys,

this week we created our first two use cases with their particular mockups and activity diagrams.

You can find them here:

- [1]Use case "Calculate Highscores"
- [2]Use case "Create Match"

1. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-Calculate-highscore
2. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-Create-match

---

Anonym (2018-10-24 17:13:20)
Hey guys,

your blogpost this week is queit nice to read. What is about the point 3 and 5 of the calculate highscore UC?
Was it forgotten or just not defined yet? Also with the third point of the UC of creating a match.
Nevertheless, good work and keep it up!

Greetings,

FyF-Team

Felix Hausberger (2018-10-24 22:51:10)
Dear Team Tenniskönig,

I just had a look through your UC "Create Match". I really appreciate, how you differentiate between the user view and application view. But is this really intended in a use case diagram? As far as I know, you should only display the flow of actions made by the user.
But if so, then you should add another branch to the flow chart, that checks, that input is entered correctly. I'm asking myself, what would happen, if the input provided by the user uncharacteristic or tennis (like 11:1 instead of 6:0, or a unrealistic high amount of points like 1.000.000 if the kids try to cheat on the system to get honored in the end ;)). To which screen will the user be directed in this case or will he just receive a message? Therefore you should add another mock-up page.

But in the end, you already did a quite good job! Could you maybe add to the blog post, which use case diagrams and specifications are coming soon? I cannot wait to get insights in the other use cases, that's why I'm aksing ;)

Your sincerely
Felix from Team dashup

Anonym (2018-10-25 11:23:01)
Dear Team Tenniskönig,

All in all I think that your blogpost meets this weeks blog post requirements quite well. But like Felix has mentioned before, adding a branch that checks whether an input made to your app is sensible makes sense in the context of a highscore tracker. In addition to the already mentioned criticism, your "calculate highscores" use case description seems to be missing the preconditions part. If these issues are fixed your blog post will be really solid.

I am very curious and exited how your project is going to take shape in the coming weeks and months.

Your sincerely
Falko@CodeCrunch

1.2 November

HW Week 5: Feature Files (2018-11-03 19:29)

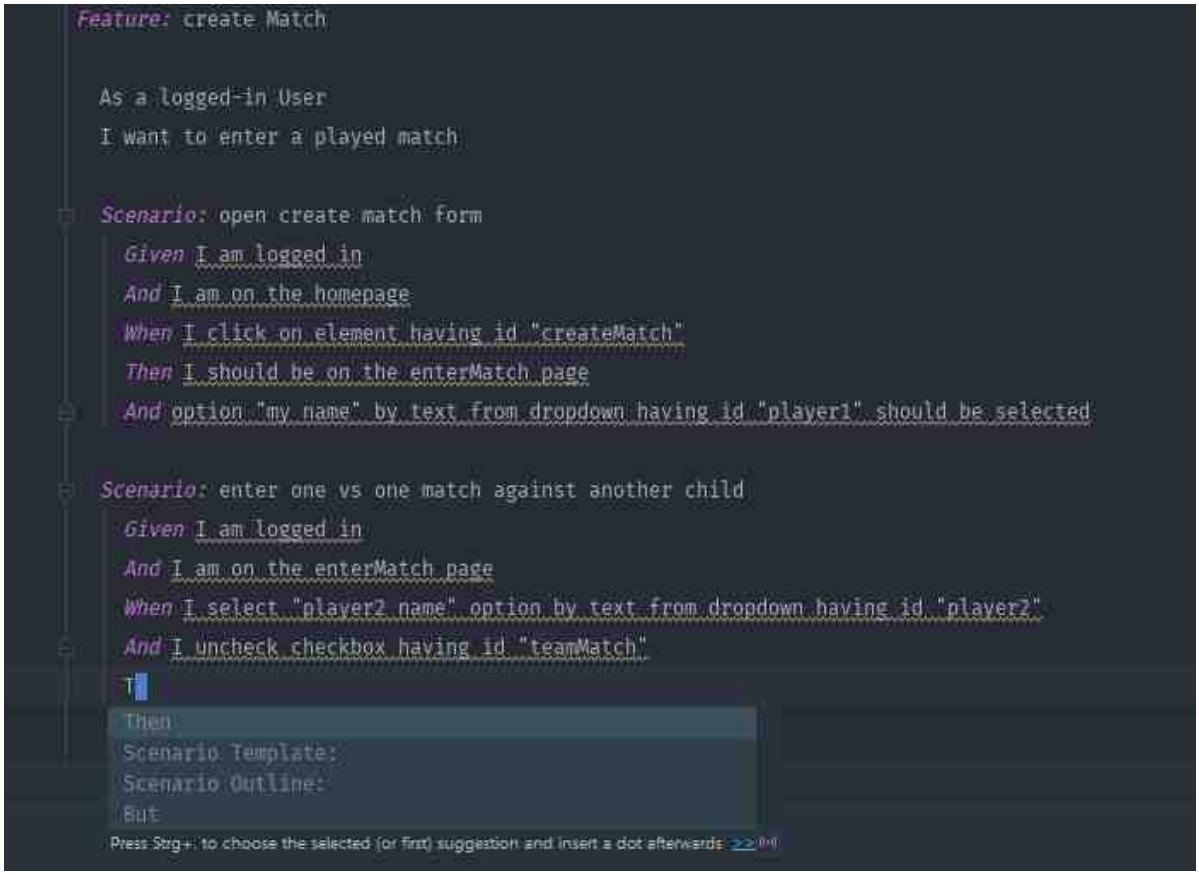
Hi there,

this week we implemented the first two feature files for our use cases.

You can find them here:

- [1]Use case "Calculate Highscores"
- [2]Use case "Create Match"

We also integrated the Gherkin Plug-In into our IDE:



1. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-Calculate-highscore
2. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-Create-match

---

Anonym (2018-11-06 00:18:12)
Hey Tenniskönig Team,

your feature files, exspecially for the "Create Match" Use Case are very detailed.
The "When" statements for your submit scenarios contain expressions like valid or invalid data. I think an examples table would be useful here. You can reference to the content via a variable in your statements and tests with diverse data will be possible.

I like the assignment of IDs for UI elements. It certainly will be useful for testing.

Best Regards
Sven (dashup Team)

PerfectTimeCrew (2018-11-07 00:34:10)
Hi guys,

we are impressed by your good work this week. Well done!

Your feature files are very detailed and because of the easy written sentences also understandable for non-programmers.

In your Use-Case Specification: Will the Use-Case Specification automatically be updated in case you change something in the feature files?

Keep up the great work. We are looking forward to your next steps.

Your PerfectTimeCrew

HW Week 6: Class-Diagram (2018-11-12 12:52)

Hey guys,

This week we created our Class-Diagram.
It's the first version and therefore it's not that extensive.

You can find it [1]here.

Best regards,

Team Tenniskönig

1. https://github.com/tenniskonig/Dokumentation/blob/master/ClassDiagram.svg

---

CodeCrunch (2018-11-14 13:47:26)
Hello Tenniskönig,

Your UML diagram seems to meet all criteria to me.
I'm a little confused why some classes are translucent while others are not,
but other than that, great work!

Greetings Janis@CodeCrunch

Anonym (2018-11-14 16:10:57)
Hello Tenniskönig,

the optik and structure of the class diagram looks not bad and I think you know what you're going to develop. The only think I must criticize is the sharpness of your image. I can't read it very well, it is to small. It would be great if you could fix that.

Otherwise good work and keep the good work up.

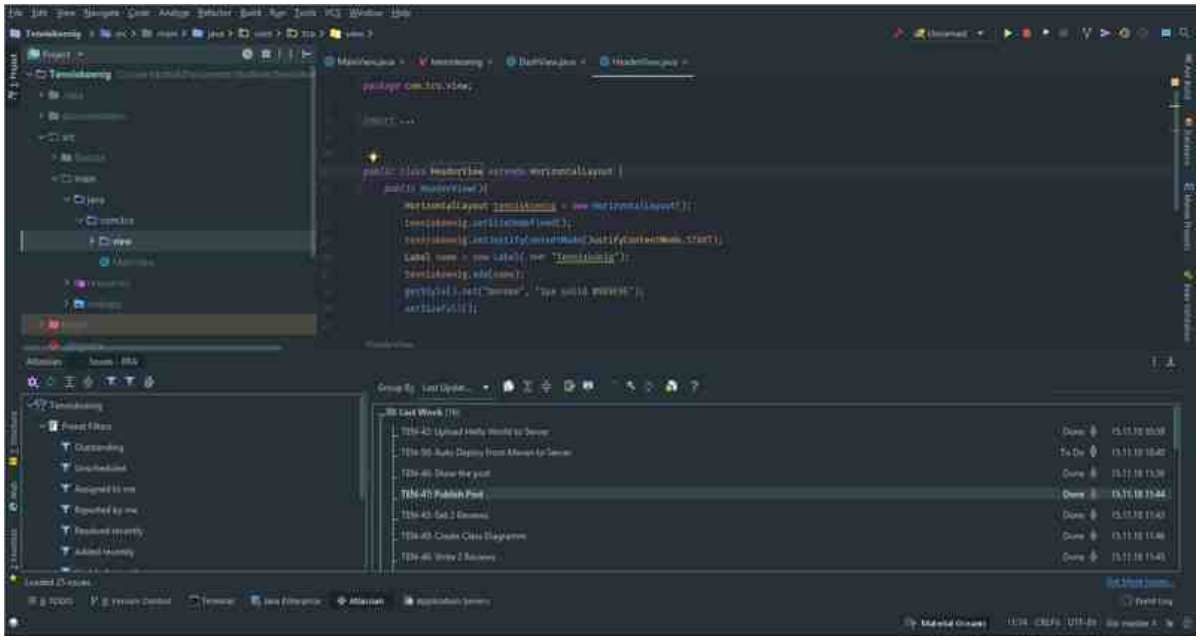
Kind regards,

FyF-Team

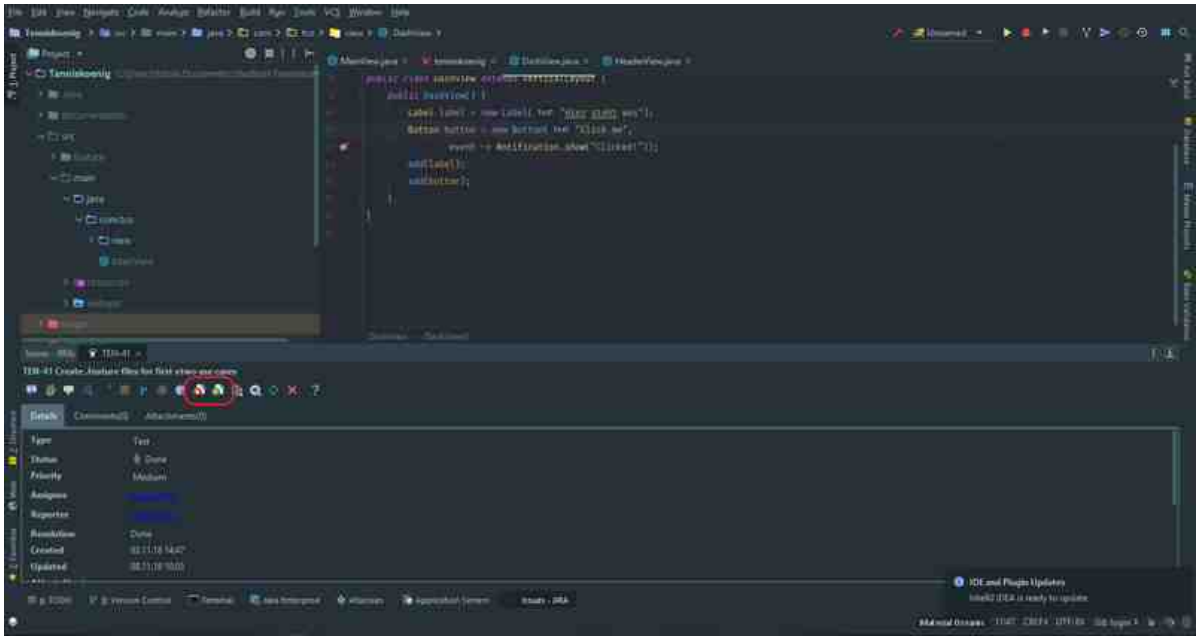
HW Week 7: Scrum (2018-11-19 15:11)

Hey guys,
this week it was all about Scrum so we have some pretty diagrams to show you.

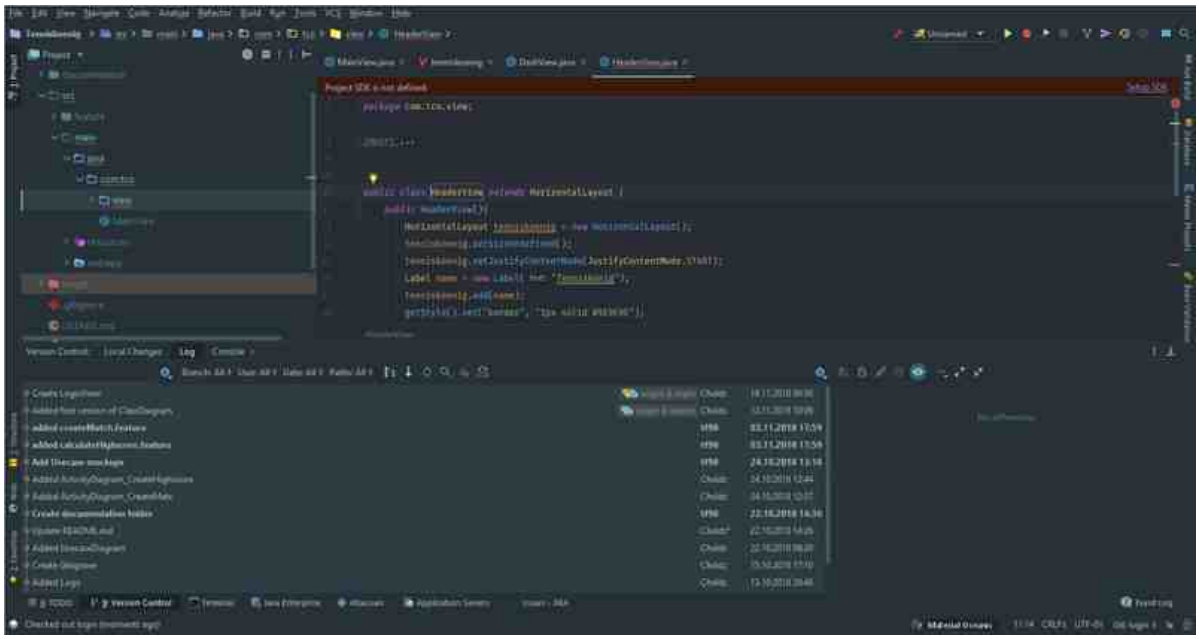
So lets start with our Project Managing tools, which we integrated in our IDE IntelliJ.



Jira integrated in IntelliJWith this we are able to track our time without leaving the IDE, which makes things much easier.

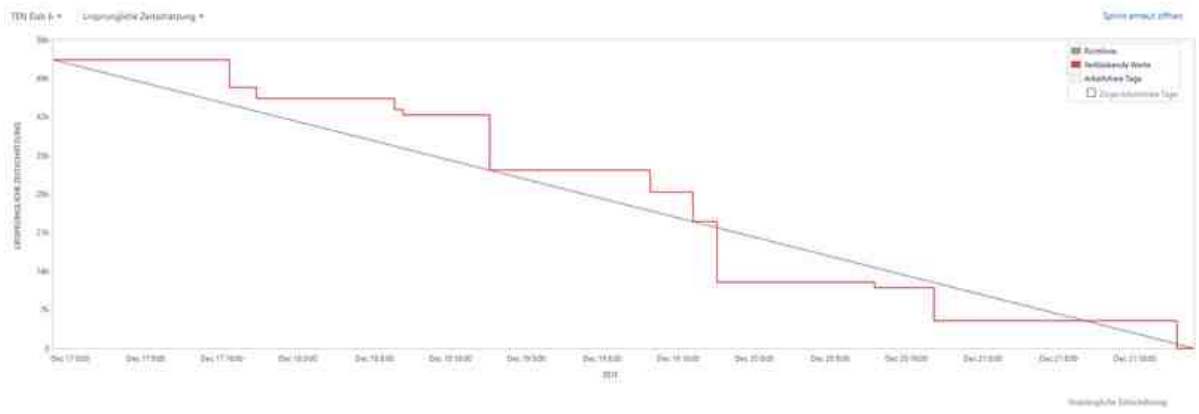


Time Tracking in IntelliJ



Git integrated in IntelliJWith Git in our IDE we can switch the branch or do a simple commit inside the IDE.

Also we can show you our "best" Burn-Down-Diagram:



TEN Sprint 6

This is the link to the Sprint: [1]Jira Sprint Elab 6
Looks pretty clear :D

We had a look to show you a Gantt Chart, but as it turns out the plugin for what aren't for free. So we are still looking fo ra solution, but other teams who also work with Jira have the same problem. So we will notify you if we have a Gantt Chart.

So that was it from our side, we are looking forward to have your comment down below!!

Greetings Sven

1. https://jiratinf16b4.it.dh-karlsruhe.de:8443/secure/RapidBoard.jspa?rapidView=19&projectKey=TEN&view=reporting&chart=burndownChart&sprint=89&estimate=field_time

---

Anonym (2018-11-20 09:58:54)
Hey guys,

your tools are integrated very well into your IDE whih will make live easier.
One thing I want to mention is the time tracking. In your screenshot about the integration of Jira there is only the issue list vissible but I can't find any possibilty for time tracking in the picture.
Unfortunately the sprint you have linked is not publicly accessible. It would be nice if you could fix that.

Best Regards,
Sven (dashup)

Tenniskönig (2018-11-22 10:37:47)
Hi Sven,

thank you for your feedback. We have now added a picture which shows how you can track time directly in IntelliJ.

Unfortunately it seems like you can't make the burndown diagrams publicly accessible in Jira. We're sorry about that but we're trying to find a solution for that.

Greetings,
Tobias from Tenniskönig

Anonym (2018-11-21 11:55:52)
Hey Guys,

it's nice that you use Jira directly in your IDE. I think this simplified your work. Also using Git, in your IDE. You wrote about changing easily Branches. So do you develop on different Branches?
I'm glad to see your Gantt chart, so I hope you find a solution to solve the problem.

Keep up the good work!

Greetings,
Simon / FyF-Team

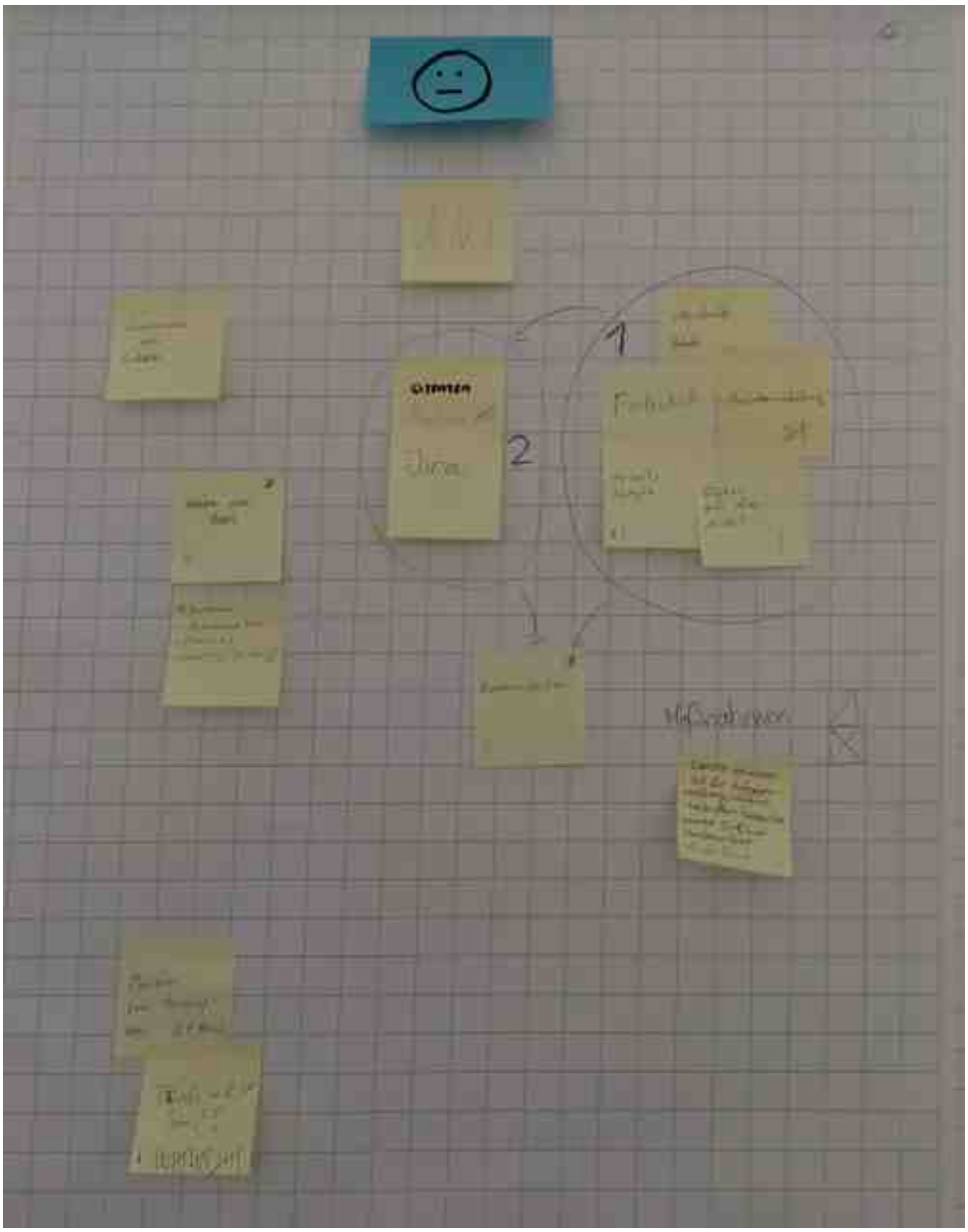
HW Week 8: Retrospective (2018-11-24 13:24)

Hey guys,

in this week we did our first retrospective! For that a agile trainer came to our course and showed us how to do it!

In the first session we looked at all positives side we experienced in our team and our work. We put them all together at a flip chart. Here you can have a look at our positive things.

Next we had a look at the negative sides. We also put them on a chart, so have a look!



At the end of the course each group presented their result to the class. We shared our positive and negative points and how they could do better in the future.
We got some new ideas to make our process faster and more efficient.

So we are look forward to reach our goal to an productive website faster.

See you next week
Tenniskönig

Unknown (2018-11-28 00:57:34)
Hey guys,
thanks for sharing your thoughts.
Your update is well structured but maybe you could share some of your ideas on how you want to improve your work you mentioned in your last paragraph.
That kind of documentation could help you to remember those plans in the maybe not so distant future.

Cheers

Kai@CodeCrunch

Jannik Möll (2018-11-28 09:41:40)
Hi there,

nice blog post! I like that you inculded pictures of the retrospective charts!
Some of the pictures are hard to read so it would be good to have the contents of the pictures in the post as a list.

You also mentioned you had "[...] some new ideas to make our process faster and more efficient." would you mind sharing what those ideas are?

Best regards
Jannik@DigiWill

Anonym (2018-11-28 09:50:24)
Hey guys,

your positive and also negative experiences concerning the project have been dealt in a very structured way. So you can get a good overview of the project process so far.
One point that should be mentioned is the improvement of aspects not yet running smoothly. This could be more detailed.

Best Regards
Sven (@dashup)

1.3 December

HW Week 9: MVC Tool (2018-12-04 21:14)

Hey guys,

this week we continued the work on our project and implemented Unit test.
Also we have created a Software and Architecture Document, which you can find in our Github Repo or just klick this link: [1]SAD

That's it for the week.
See you next one
Greeting Tenniskönig

1. https://github.com/tenniskonig/Dokumentation/wiki/Software-Architecture-Document

---

Unknown (2018-12-06 09:47:33)
Hey guys,

your blog post is nice to read this week. The sas dacument contain all important and useful points. It is interesing to see how your program will work, what you need to implement and this things.

Thanks for offering the document this week.

Keep up the good work.

Kind regards,

Mikka, FyF-Team

Anonym (2018-12-06 10:40:14)
Hello Tenniskönig,

your Software Architecture Document is very detailed, good work on that!
However it seems to be missing a class diagram that specifies how exactly you implemented the MVC architecture.

Cheers, Janis@CodeCrunch.

HW Week 10: Cucumber Running (2018-12-11 16:40)

Hey guys,
we are glad to present you that our Cucumbertest are finally working now. It was a hard way to get there but now everything is working.
So have a look in our video!

See you next time
Greetings Tenniskönig

HW Week 11: Midterm (2018-12-13 09:40)

Hello guys,

this week we present our newst developements

Here is our Handout:
[1]PDF Handout

Here you come to our git:
[2]Git-Repo

Our Projectmanagement:
[3]Jira

Best regards,

Marvin (Tenniskönig)

1. https://github.com/tenniskonig/Dokumentation/blob/master/Handout.pdf
2. https://github.com/tenniskonig
3. https://jiratinf16b4.it.dh-karlsruhe.de:8443/projects/TEN/summary

---

2. 2019

2.1 April

HW Week 12: Confirmation of Scope (2019-04-08 09:27)

Hey guys,

so after three months of break. We're finally back! We couldn't make any implementations in this time, but now we are starting fresh off! I can tell you we are looking forward to implement a lot of cool features.
So what are we planning to do in the next months? We discussed our current situation and the future task.

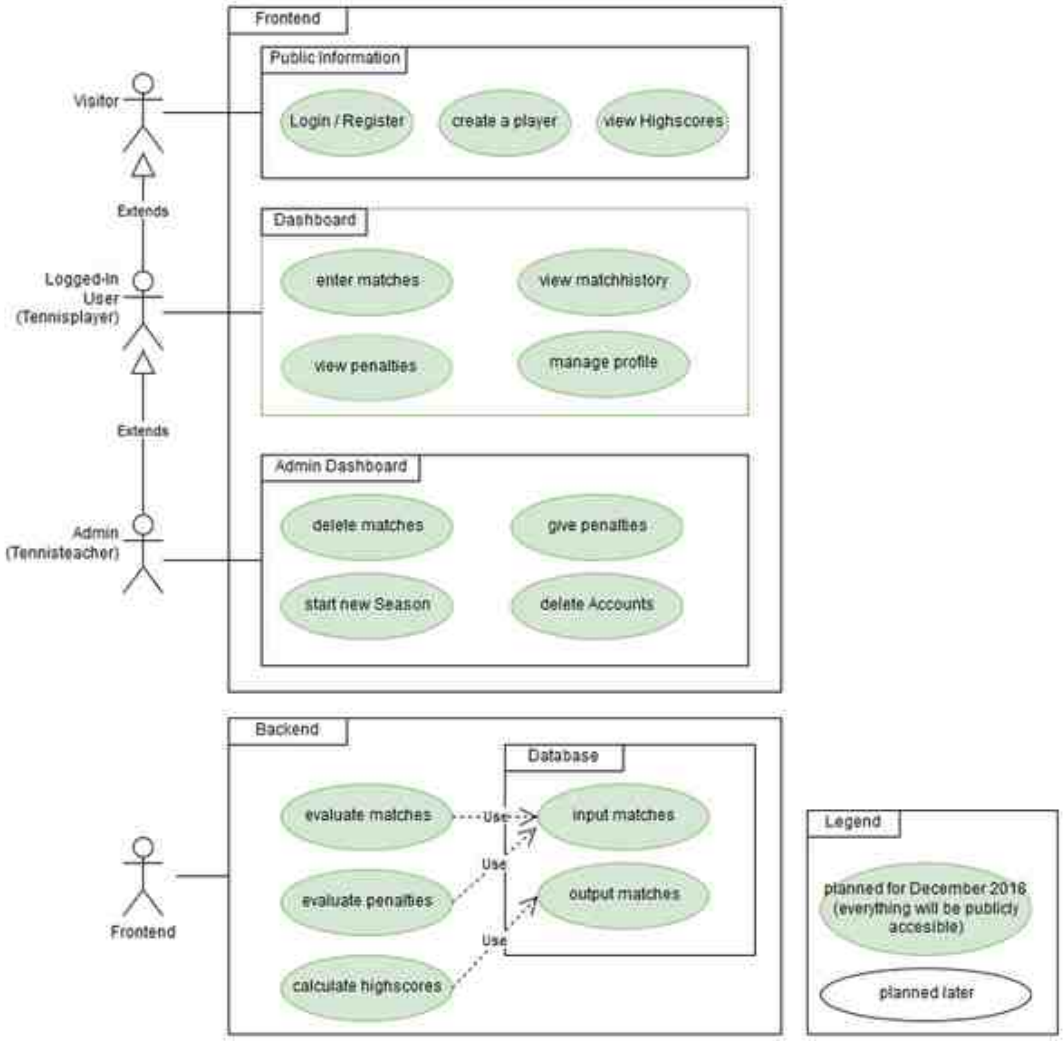
- User Login
- Administration features
- implementing penalties

Since we encountered a lot of problems with integrating Spring and Vaadin we decided to switch out our frontend technology. Instead of Vaadin we are now using Angular as frontend framework. As Angular is more independent we also reorganized our git repositories. You can now find them if you follow these links:

- [1]Frontend
- [2]Backend
- [3]Documentation

We've also updatet our UseCase-Diagram. As you can see we have activated all UseCases. We are looking forward to implement all planned features.

[4]



So stay patient and be excited what is coming in the next weeks.

Your Tenniskönig-Team

1. `https://github.com/tenniskonig/frontend`
2. `https://github.com/tenniskonig/backend`
3. `https://github.com/tenniskonig/Dokumentation`
4. `https://raw.githubusercontent.com/Marv1104/Tenniskoenig/master/documentation/Use_Case_Diagram_2.png`

Thimo (2019-04-15 09:28:39)
Greetings Friends,

we are happy that you stayed motivated over the 3 months of not working on your project. I really hope that the changes you made with switching to Angular will help you with implementing your frontend.

Excited and patient with best regards
Thimo - DHBWieWarsEssen

Robin DigiWill (2019-06-18 11:57:50)
Hey guys,
wow what a change. Hope you made the right choice by switching to angular. Looking forward to see the project finished.
Best Regards
Robin - DigiWill

HW Week 13: time and risk management (2019-04-15 10:00)

Hey guys,

We have created now for the old five UC an own Wikisite in our Repository. So have a look in that to:

[1]UC - Wiki

Here is our timeprotocol:

[2]Zeitverfolgungsbericht

Also we discussed the risks that may come up in our project. You can find an excelsheet with the summary of our results here:

[3]Risk Management

Greetings Tenniskönig

1. https://github.com/tenniskonig/Dokumentation/wiki
2. https://jiratinf16b4.it.dh-karlsruhe.de:8443/secure/ConfigureReport.jspa?versionId=-1&sortingOrder=least&completedFilter=all&subtaskInclusion=onlySelected&selec
3. https://drive.google.com/file/d/1CHuObFTy4rdCw1jg-kfYpkci0yYtBL92/view?usp=sharing

---

Jonas (2019-04-16 13:02:14)
Hey,

I really like your detailed usecase wiki pages. The include every important information (diagram, mockup, cucumber file) for each usecase.

Your time tracking table is very detailed as well, but it looks like it's missing a time estimated/spent by usecase (and not by ticket).

Your Jonas from turnie.re

Anonym (2019-04-17 09:19:59)
Hey guys,

your risk management looks good. Seems like you have thought about all problems that can occur. Also your time tracking is very clear. There are all information about estimation and the time you spent for every use case.

So keep up the good work.

Greetings
Simon / FyF

HW14: Function Points (2019-04-23 11:24)

Hello everyone,

this week we worked on our function point (FP) calculation. With this calculation we are able to estimate the time we will need to implement future usecases (UC).

To calculate the FPs for the UCs from semester 3 we used the [1]Tiny Tools FP Calculator. This tools gives you the following to tables to calculate your FPs:

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	1	○	◉	○
Number of User Outputs	1	○	○	◉
Number of User Inquiries	0	◉	○	○
Number of Files	5	○	◉	○
Number of External Interfaces	0	◉	○	○

Complexity Adjustment Table | FP Calculation

Domain Characteristic Table

Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE No influence 0	1	2	3	4	Essential 5
1	Does the system require reliable backup and recovery?	○	○	◉	○	○	○
2	Are data communications required?	○	○	○	○	○	◉
3	Are there distributed processing functions?	○	○	◉	○	○	○
4	Is performance critical?	○	◉	○	○	○	○
5	Will the system run in an existing, heavily utilized operational environment?	○	◉	○	○	○	○
6	Does the system require on-line data entry?	○	○	○	○	◉	○
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	◉	○	○	○	○	○
8	Are the master files updated on-line?	○	○	◉	○	○	○
9	Are the inputs, outputs, files or inquiries complex?	○	○	○	◉	○	○
10	Is the internal processing complex?	○	○	○	○	◉	○
11	Is the code to be designed reusable?	○	◉	○	○	○	○
12	Are conversion and installation included in the design?	◉	○	○	○	○	○
13	Is the system designed for multiple installations in different organizations?	◉	○	○	○	○	○
14	Is the application designed to facilitate change and ease of use by the user?	○	○	○	◉	○	○

Domain Characteristic Table | FP Calculation

Complexity Adjustment Table With this tool we calculated the FPs for our previous UCs. Also we calculated the total time spent per UC and the time needed per FP. The following picture shows our result.

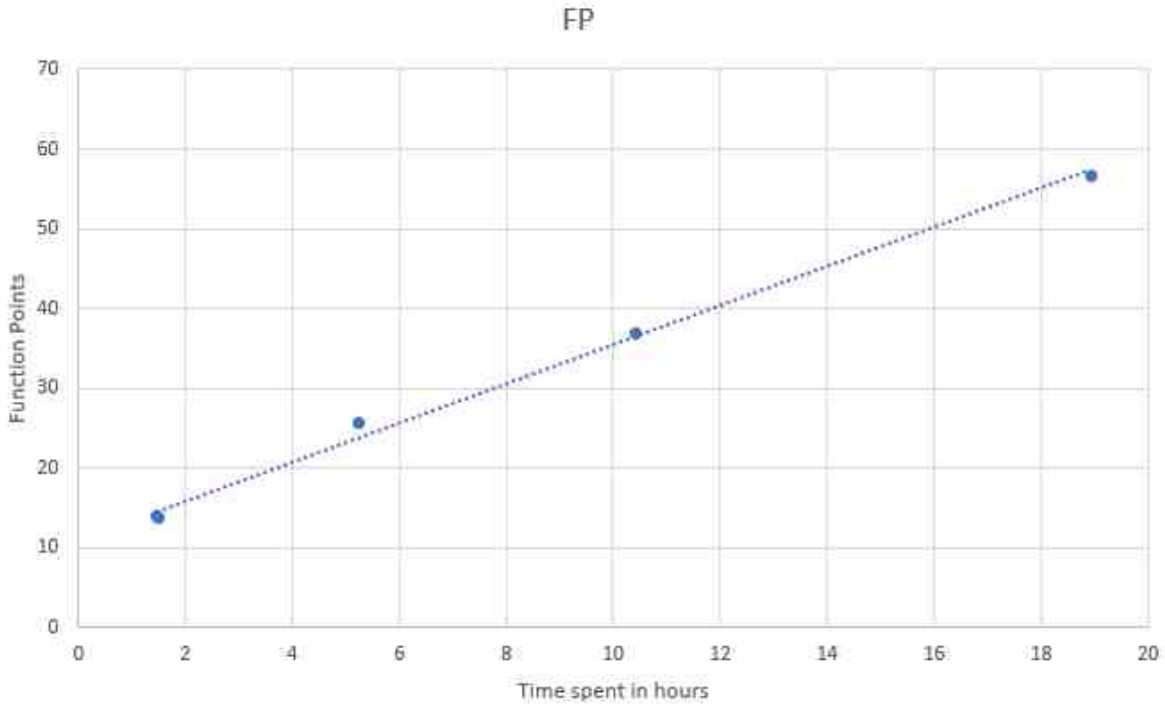
Time spent on use cases

UC	FP	Time spent in hours	Time / FP
Evaluate Matches	56,73	18,93333	0,33374464
Enter Match	36,96	10,41667	0,28183622
View Matchhistory	25,73	5,25000	0,20404197
View Penalties	13,77	1,50000	0,10893246
Highscore	13,94	1,46667	0,10521282

FP calculation for previous UCs

This data made it possible to draw a graph whitch shows the correlation between time spent and FP. The determined formula for calculating estimated time from FPs is therefore ***f(x) = 0,4073 x -***

4,4718 where f(x) is time and x is FP.



We then also calculated the FPs for our future UCs for the fourth semester and calculated the estimated time for them.

Time estimation for future use cases

	y = 0,4073 x - 4,4718
UC	FP	Estimated Time in hours
Login / Register	55,18	18,00301
delete matches	29,58	7,57613
give penalties	20,16	3,73937

Time estimation for future UCs

You can find an excel sheet with all our calculations [2]here.

The detailed FP calculation can be found here:

[3]FP Calculation - View Penalties

[4]FP Calculation - Enter Match

[5]FP Calculation - Evaluate Matches

[6]FP Calculation - Highscore

[7]FP Calculation - View Matchhistory

[8]FP Calculation - Give Penalties

[9]FP Calculation - Delete Matches

[10]FP Calculation - Login/Register

Since we don't have enough time to implement all our usecases we decided not to implement the following:

- manage profile
- delete matches
- start new season
- delete accounts

Greetings,

your Tenniskönig team

1. http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP_Calc.html
2. https://drive.google.com/file/d/1GsKlrBXBNfxS61WlV4ixzLQNyJx-C-N2/view?usp=sharing
3. https://drive.google.com/file/d/13_im4shy8X6IzHUykKm3xOjXvKgQYs51/view?usp=sharing
4. https://drive.google.com/open?id=11zPtic7Xl06tbQblxgBVnzMetTSUUFzd
5. https://drive.google.com/open?id=1YozCRztISGAsnpTWKOBwAlAaqtbXjkTl
6. https://drive.google.com/open?id=1jQrvMy_XGuMgNc37I2CawBrdTQP8ACTT
7. https://drive.google.com/open?id=1vHGsrA-876QeedgG1PAXa56G-f-D1bGK
8. https://drive.google.com/open?id=13_im4shy8X6IzHUykKm3xOjXvKgQYs51
9. https://drive.google.com/open?id=1iyFP3GuBQBg8lHbpsb48D2WyBRg8KFMk
10. https://drive.google.com/open?id=1vHGsrA-876QeedgG1PAXa56G-f-D1bGK

Unknown (2019-04-23 23:18:50)
Dieser Kommentar wurde vom Autor entfernt.

Falko (2019-04-23 23:21:41)
Dieser Kommentar wurde vom Autor entfernt.

Falko (2019-04-23 23:24:25)
Greetings Tenniskönig Team,

from a formal point of view I think this Blogpost is almost flawless. You provided the intention of this blog post, mentioned which tools you used to do the function point calculation and provided the fp calculation for every UC you made to come up with the h/fp graph.
As of this moment though I am missing an interpretation of the h/fp graph, which has been created as a result of this blogpost. An example of what could be covered in such a graph interpretation, would be an explaination on why the data points in the coordinate system you provided correlate almost perfectly. From a mathematical stand point a correlation of such kind is a special phenomenon and must be at least appreciated in some form because correlations like this don't occur often in the real world and usually yield great insights for the plotted data. One of those insight which I will give away right now for some inspiration is that such a correlation usually suggest that future values added to this data series are probably highly predictable meaning that your calculated estimated time should be very accurate.
Lastly I would like to ask you why you decided to select the y-axis of the coordinate system of your graph to represent time since in this case the time dimension is the inpendant variable and independant variables

usually are represented on the x-axis rather than the y-axis.
Other than that I found your blog post quite good. Keep up the good work!

Best Regards,
Falko@CodeCrunch

Tenniskönig (2019-04-24 08:54:42)
Hi Falko,
thank you for your feedback.

We too were really surprised about this correlation. We hope that this will also be the case for the next semester.

Thank you for the tip with our diagram we fixed that now.

Greetings
Tenniskönig Team

Jonas (2019-04-24 09:09:18)
Hi,

good job on your blog post. As you already fixed the stuff Falko mentioned earlier, I don't really see any problems with your post.
It contains all relevant information and a good explanation of the given task.

Your Jonas from turnie.re

Stiven@KeepTrack (2019-04-24 09:21:17)
Hi Tenniskönig-Team,

I was really impressed by the detail and effort that went into your planning, especially in the excel sheet you provided. It gives quite a good overview of your project management approach and this weeks homework.

Your blogpost was well structured and as far as I'm concerned there is nothing more to add.

Keep up the great work, I'm curious to see where your project is headed.

Kind regards,
Stiven@KeepTrack

2.2 May

HW Week 16: Refactoring (2019-05-02 14:39)

Hey guys,

this week we did a little refactoring practise.
So have a look at all our Repositories.

We worked with IntelliJ, which showed us many refactorings.
But working with SonarLint was better, as it has more detailed descriptions and examples on how to fix this.

Sven: [1]https://github.com/Chsldz/DHBWRefactoring
Marvin: [2]https://github.com/Marv1104/Fowler-Refactoring
Tobias: [3]https://github.com/tf98/Fowler-Refactoring/tree/master

1. https://github.com/Chsldz/DHBWRefactoring
2. https://github.com/Marv1104/Fowler-Refactoring
3. https://github.com/tf98/Fowler-Refactoring/tree/master

Anonym (2019-05-03 08:31:47)
Hey there,

overall a nice refactoring example project.
In Chsldz repo I haven´t found JUnit test code. Furthermore you can also replace the switch statement with polymorphism.
Have you thought about getting a grade for your code style from here: https://app.codacy.com/projects ?

Kind regards
Nils@CommonPlayground

dashup (2019-05-05 17:08:10)
Hey Team Tenniskönig,

first of all, good work! Don't forget to present some refactoring features, that your IDE provides you. Just give some screenshots to be compliant with the grading criteria. As Nils already mentioned, codacy lets you check your refacoring progress, you should really consider using it, it takes approximately 2 minutes to set it up. It even gives you some additional refactoring advices.

@Marv1104. Maybe you should not test, whether the statement string remains the same, but rather if the result of getCharge() and getFrequentRenterPoints() is the same as before. You could even add a getTotalCharge() and a getTotalFrequentRenterPoints() test. This way, you will find out more quickly, which part of your program fails, as you have extracted your test logic into 4 different tests. So use your knowledge about refactoring and refactor your test ;)

@Chsldz. Don't forget to add the tests. If you plan to continue refactoring (which I hope, because there are some things missing from chapter one), please provide the name of the code smell in the commit message.

HW Week 15: Testing (2019-05-07 10:03)

[1]

Hey guys,
this week is all about testing.

We are using:

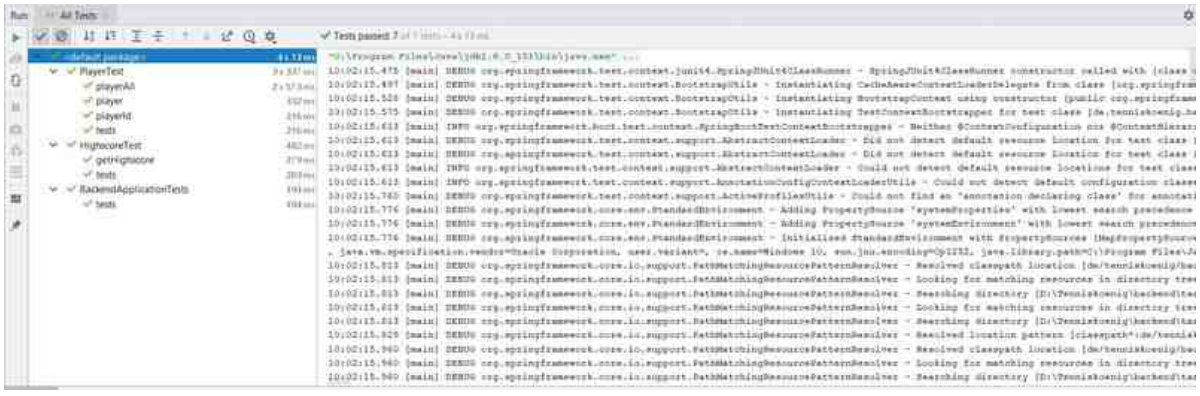
- Unit test, which run automatically in Travis
- Feature files, testing the functionality of the webpage
- User test, which are giving us the real feedback of the users

All our Test are based on our Testplan we have created. Please have a look here: [2]TestPlan
This document contains all the information about our handling of tests and scope regarding tests.

All our testcode can be found here: [3]https://github.com/tenniskonig/bac - kend/tree/master/src/test/java/de/tenniskoenig/backend

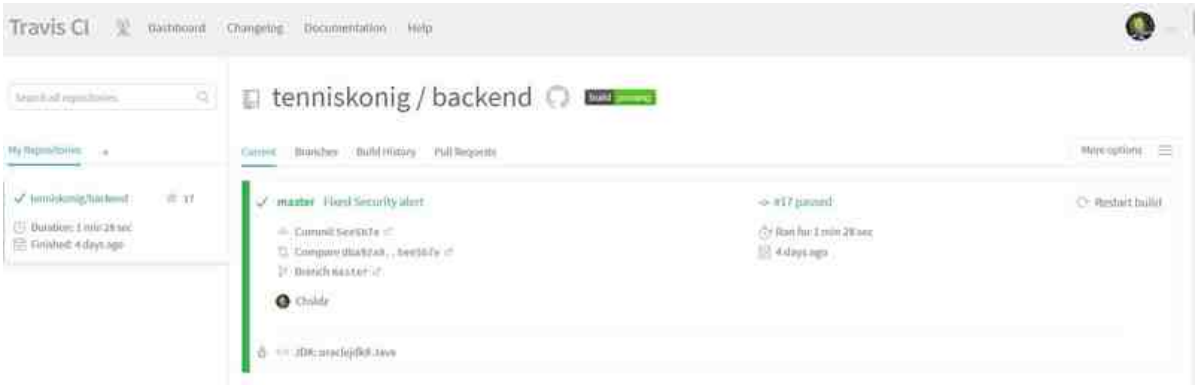
We've implemented the test dependencies here: [4]https://github.com/tenniskonig/bac-kend/blob/master/pom.xml

Out tests running in the IDE:



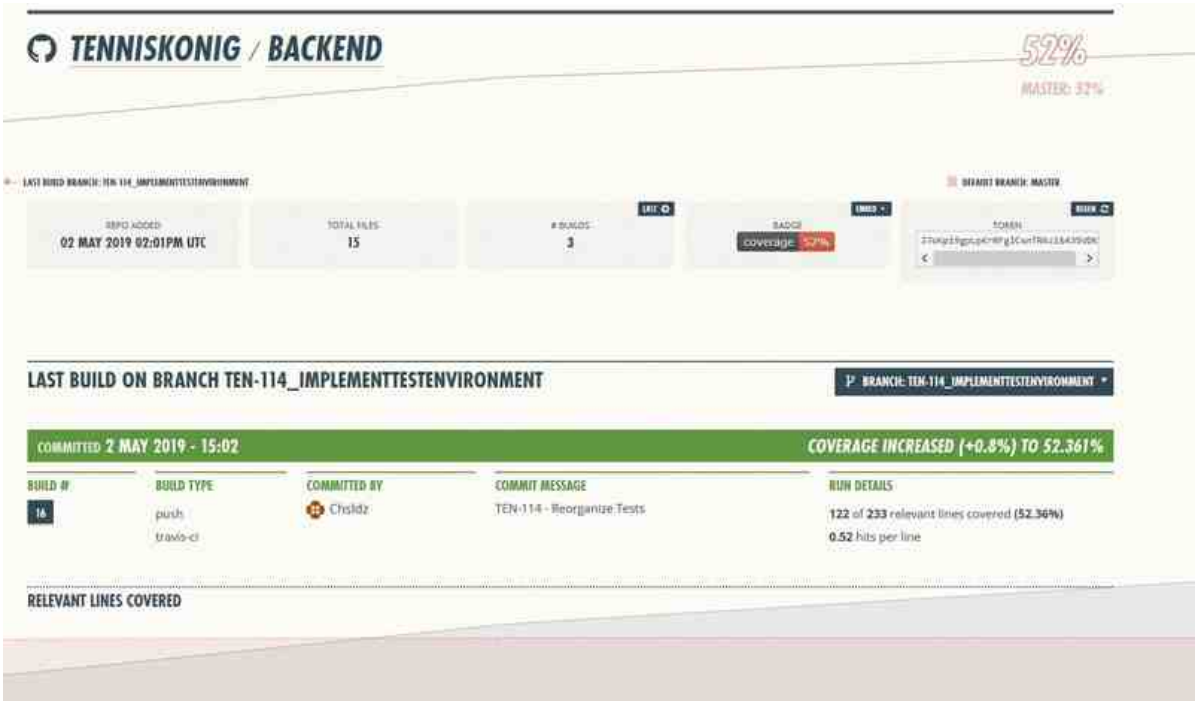
Travis CI:

More tests will be merged into the master branch soon.
With Travis the software will be build and tested. We can see the log output for debugging purpose.
Also we can deploy our Software if the build passed all tests.



Coveralls:

Coveralls recieves the result from Jacoco after running in our CI and will display them nicely.



As we are using Coveralls for viewing the coverage. You can see our live coverage here:
[5]
As most test aren't merged to the master yet the number doesnt look great, but it will continue to grow for more coverage.

See you next week
Greeting Tenniskönig

1. `https://www.blogger.com/`
2. `https://github.com/tenniskonig/Dokumentation/wiki/Test-Plan`
3. `https://github.com/tenniskonig/backend/tree/master/src/test/java/de/tenniskoenig/backend`
4. `https://github.com/tenniskonig/backend/blob/master/pom.xml`
5. `https://coveralls.io/github/tenniskonig/backend?branch=master`

Unknown (2019-05-07 20:13:46)
Hey Tenniskönig,

Nice to see you starting out with 52 % of test coverage. But looking into your tests, It looks like these are just integration tests. Therefore it would probably be helpful for you to write some unit tests as well. Afterall they help you the most, as they reduce the hassle when you change something and it does not work. If you test all methods thoroughly it will be way easier to catch the mistake you made.
Another thing I noticed is that you just test, if the status code of the response is 200, which is obviously good, but not the whole story. It would be better if you were also checking if the object that was within the response is like the one you are expecting. As you have done the hard part of getting the tests to run, it will be pretty easy to just check a few extra things. Given that you probably know, that it works at this point in time, you could also just run the test within debugger and break right before the assert line. Then you can check what variables are valid at that point and you can incorporate them into the test. After that if you change the code all these tests will fail if it also changes its behaviour.

I hope i explained it well enough. If not, feel free to ask me.

Greetings,
Daniel - turnie.re

Rouven@KeepTrack (2019-05-08 09:36:58)
Hey guys,

great work on your tests! I really like that you already have everything up and running in your pipeline regarding automated tests and test coverage.
It also seems like you managed to plug different tools together perfectly, so you even have a nice overview in the end!
In contrast to Daniels post, I think integration tests are way more useful than unit tests, but that might be down to personal preference!

Keep up the good work!

Rouven@KeepTrack

Anonym (2019-05-08 10:01:19)
I did not say, that integration tests were not useful, but in my opinion it is important to have both and also to have integration tests that actually test stuff instead of just checking for the http status code.
Daniel - turnie.re

Rouven@KeepTrack (2019-05-08 12:11:18)
Dieser Kommentar wurde vom Autor entfernt.

Rouven@KeepTrack (2019-05-08 12:14:20)
I think we are generally on the same page here. It’s just these sentences: "Therefore it would probably be helpful for you to write some unit tests as well. Afterall they help you the most, as they reduce the hassle when you change something and it does not work.", that bother me a little, because I personally think integration tests are more useful than unit tests. Especially when written by the person who wrote the code itself, unit tests are often not as useful as they seem at first in my opinion.
You should have both of course, though!
Also, the point you’re making regarding the response check for just the status code is very valid, I’m definetly agreeing with you on that.

Rouven@KeepTrack

HW Week 17: Retrospective (2019-05-08 11:20)

Hello there,

today we did our second Retro.

First we described what we learned since the last one.

- creative Greetings

- Eclipse is bad, IntelliJ is better

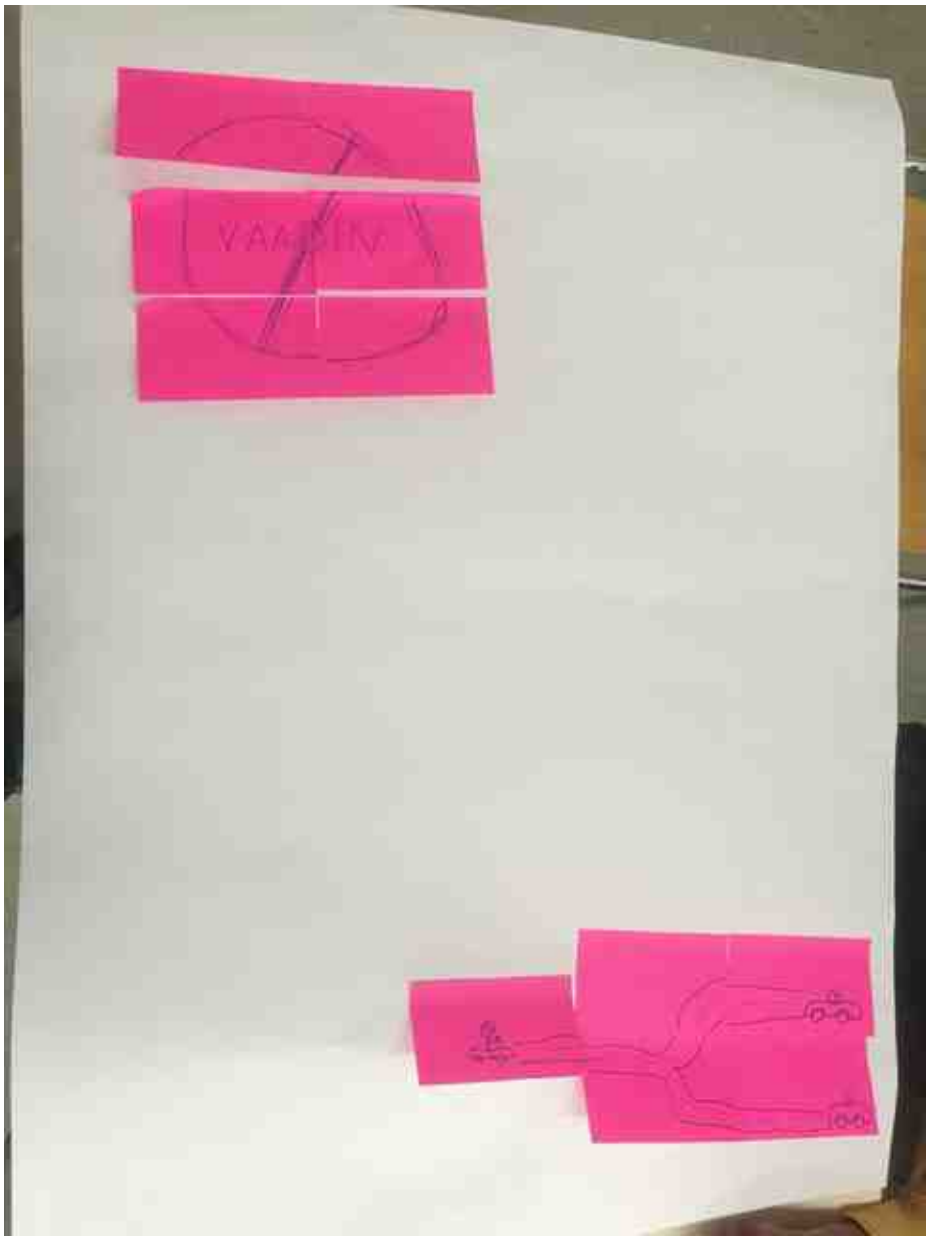
"My English is gooder now :) " - Marvin

- better communication

- clear defined distribution of labour

- usage of Jira

Then we draw a picture which depicts our project:



And the last point was figuring out what we can do better:

The only point we could think of is to write our blogposts and comments earlier.

Pictures of all our thoughts will follow as soon as we get them!

Greetings

Tenniskönig Team

DHBWieWarsEssen (2019-05-13 09:52:05)
How’s it hanging dudes,

as we worked together this day, I really have to say you guys do a very good job and you gave us a lot of input of improving our future work.
You made good suggestions about teamwork and workflow.
Keep your motivation and good work going.

Best Regards

Thimo@DHBWieWarsEssen

HW Week 18: Design Patterns (2019-05-28 22:13)

Hello pupils,
This weeks task was to look through different design patterns and implementing one of them in our Project. This served as a source to inform ourselves about design patterns [1]Factory-Pattern.

For our application we chose the Factory Method Pattern. The idea of the Factory Method Pattern is to disguise the construction of a model with another class, the factory.

In our case factories consist of static methods which are called to create a new model object, fill it with data and finally return it. The following diagram shows where this design pattern lies within our application.

Class-diagram

This example commit shows how the new Pattern was integrated into our Code. [2]Factory pattern Commit
Using a UserFactory a user object is created, filled with data and finally returned. This change at that point might not seem very useful but it helps to ensure that we work Object-Oriented as we create an Object before we use it's data instead of just taking it from somewhere else.

In the future this might help making our code more clear and understandable.

While the Factory Method Pattern works for our application, it's questionable whether it's reasonable to use it as right now our Models lack the complexity a Factory would disguise. Right now all the factories are doing is calling the setter methods of our models and thereby don't provide any additional value.
Greetings Team Tenniskönig

1. https://www.oodesign.com/factory-pattern.html
2. https://github.com/tenniskonig/backend/commit/a1fe014be834b61df7e2a69c303eec05a9c22f34

Turniere (2019-05-28 22:16:38)
Hey guys,
looks like you did solid work this week.
I would do some improvements to get the best grade.
Please link directly to the commit and not to the repo.
A Classdiagram which show the added Factory would also be nice.
Keep going
Greeting Jonas - Turniere

Tenniskoenig (2019-05-29 08:34:14)
Hey Jonas,
thanks for your reply. I think we fixed all problems.
Keep watching your grammar.
Greetings Sven

EventLAB (2019-05-29 10:14:17)
Hello Tenniskönig,
scince you have fixed, the things that Jonas had mentioned. I think this is a good blogpost and i have nothing to complain.
Yours EventLab

HW19: Metrics (2019-05-29 10:07)

Hello there,

this week we had a closer look at some metrics to improve code quality on our project. Metrics are used to numerically evaluate the code. It helps to find critical code sections, examine them more precisely and eliminate them hopefully.

For this purpose, we used [1]sonarcloud.io. Sonar's dashboard gives an overview of the current code situation. You can find our dashboard [2]here.

For more detailed infos we used the IntelliJ Plugin Metrics Reloaded. Metrics of different categories can be automatically generated. We decided to use the [3]Chidamber & Kemerer object-oriented metrics suite, which unites various metrics.

Two of these are Weighted Methods per Class (WMC) and Response for a Class (RFC). WMC counts the number of methods in each class. This value is relevant because you want to avoid too many methods. This makes it difficult to maintain and reuse the code. Somewhat more complex is RFC. The methods are also counted here. In addition to this value, the number of methods that are called within the methods of the class is also added. A high RFC value usually means that the class is more prone to errors. The complexity of the class is high. This can lead to difficulties in understanding and make both testing and debugging more difficult.

Class metrics

class	CBO	DIT	LCOM	NOC	RFC	WMC
de.tenniskoenig.backend.controller.UserController	4	1	1	0	25	7
de.tenniskoenig.backend.domain.Game	2	1	10	0	24	24
de.tenniskoenig.backend.config.SecurityConfig	0	2	4	0	21	6
de.tenniskoenig.backend.BackendApplicationTests	2	1	2	2	19	5
de.tenniskoenig.backend.config.AuthorizationServerC	0	2	2	0	16	2
de.tenniskoenig.backend.domain.User	6	1	7	0	16	15
de.tenniskoenig.backend.config.DatasourceConfig	0	1	3	0	14	3
de.tenniskoenig.backend.factory.UserFactory	2	1	1	0	13	7
de.tenniskoenig.backend.controller.GameController	3	1	1	0	12	4

As you can see all numbers are fairly low. The highest values are inside our models and controllers thats because there are a lot of getters and setters and we are using the factory pattern. So that keeps the RFC and WMC up.

For these reasons we decided not to refactor our code.

1. http://sonarcloud.io/
2. https://sonarcloud.io/dashboard?id=tenniskonig_backend
3. https://www.aivosto.com/project/help/pm-oo-ck.html

Anonym (2019-05-29 10:18:47)
Wunderschönen guten Tag,

your metrics look really good. Those values are more than acceptable. There really is not much to add. Keep it up!

Best Regards,
Luis@DHBWieWarsEssen

Unknown (2019-05-29 10:25:18)
Hey guys,

I'm very interested in your new post. I think with SonarCloud you used a good tool for metrics analysis. I think it's ok that you not refactor the code because the RFC and WMC values are not too high.

Keep up the good work

2.3 June

HW Week 20: Installation (2019-06-12 10:58)

Finally, the day we all have been waiting for is here!!!

So hey guys, to today blog post. After now nearly one year of hard work we are ready to publish Tenniskönig.
And you can host your own server! I will show you how easy it is.

To do so you will only need to run docker on your system. [1]https://www.docker.com/

Clone our [2]installation-repository. In there you find the docker-compose.yaml, which you need to run our software.

You install and run Tenniskönig with the command "*docker-compose up*".
You can now follow the installation process, which will take some time.
But when it's finished starting up, you can reach the web application [3]here
Hope you all get it working and have fun!

Greeting Team Tenniskönig

1. `https://www.docker.com/`
2. `https://github.com/tenniskonig/Installation/tree/master`
3. `http://localhost/`

Unknown (2019-06-18 19:25:08)
Hey Tenniskönig,

I tried installing your application, which worked perfectly for me! Docker makes it so easy to download, I literally had to run one command. Very good job!

Daniel - turnie.re

HW Week 15.2: The third test (2019-06-18 19:05)

Hey everybody,

we already described the third test in this Blogpost:
So have a look there. [1]http://tenniskoenig.blogspot.com/2019/05 /hw-week-15-testing.html

All the best
Tenniskönig

1. http://tenniskoenig.blogspot.com/2019/05/hw-week-15-testing.html

---

HW21: Final (2019-06-19 08:00)

After working over half a year we're finally ready to present you all the things we have done in this time.

Requirements:

- [1]Use-Case-Diagram
- [2]UC: Calculate Highscore
- [3]UC: Create Match
- [4]UC: View Matchhistory
- [5]UC: View Penalties
- [6]UC: View Rules

Software Requirements Specification:

- [7]SRS

Testing:

- Feature Files
- [8]View Rules
- [9]View Highscores
- [10]Enter Match

[11]Testlog

Functional test

- [12]Cucumber Running

[13]Testcoverage

[14]Testplan

Project management

- Burndown chart:
- [15]TEN Sprint 6
- [16]TEN Sprint 9
- [17]Hours per Teammember
- [18]Hours per Usecase
- [19]Jira Pie Chart (you have to be logged in to see anything!)
- [20]Function Point Calculation

Ability to execute

- [21]Code (Zip Archive with all code inside)
- [22]Installation

Quality

- [23]Architecture
- [24]SAD

Configuration Management / Environmental Setup

- [25]Metrics
- [26]Risk Management
- Automated Testing
- [27]Cucumber
- [28]Travis

Used Technologies (see our [29]SRS)

[30]Patterns

Other stuff:

- Final presentation [31] Powerpoint
- Blog as [32]PDF File

Thank you for your company and all the comments you let on our post. Maybe we will see us in the future for further development on Tenniskönig.

Best Regards

Tenniskönig Team

1. https://github.com/tenniskonig/Dokumentation/blob/master/Use_Case_Diagram2.png
2. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-Calculate-highscore
3. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-Create-match
4. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-View-Matchhistory
5. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-View-Penalties
6. https://github.com/tenniskonig/Dokumentation/wiki/UC-Specification:-View-Rules
7. https://github.com/tenniskonig/Dokumentation/wiki
8. https://github.com/tenniskonig/frontend/blob/master/e2e/src/features/viewRules.feature
9. https://github.com/tenniskonig/frontend/blob/master/e2e/src/features/viewHighscores.feature
10. https://github.com/tenniskonig/frontend/blob/TEN-67_Test-UC-Match-Entry/e2e/src/features/createMatch.feature
11. https://github.com/tenniskonig/Dokumentation/blob/master/Test_Log_Frontend.png
12. https://tenniskoenig.blogspot.com/2018/12/hw-week-10-cucumber-running.html
13. https://coveralls.io/github/tenniskonig/backend?branch=master
14. https://github.com/tenniskonig/Dokumentation/wiki/Test-Plan

15. https://jiratinf16b4.it.dh-karlsruhe.de:8443/secure/RapidBoard.jspa?rapidView=19&projectKey=TEN&view=reporting&chart=burndownChart&sprint=89&estimate=field_time
16. https://jiratinf16b4.it.dh-karlsruhe.de:8443/secure/RapidBoard.jspa?rapidView=19&projectKey=TEN&view=reporting&chart=burndownChart&sprint=96&estimate=field_time
17. https://github.com/tenniskonig/Dokumentation/blob/master/Zeit_pro_User_Diagramm.png
18. https://github.com/tenniskonig/Dokumentation/blob/master/Zeit_pro_UC_Diagramm.png
19. https://jiratinf16b4.it.dh-karlsruhe.de:8443/secure/ConfigureReport.jspa?projectOrFilterId=project-10201&statistictype=issuetype&selectedProjectId=10201&reportK
20. https://tenniskoenig.blogspot.com/2019/04/hw14-function-points.html
21. https://github.com/tenniskonig/Dokumentation/blob/master/Tenniskoenig.zip
22. https://tenniskoenig.blogspot.com/2019/06/week-20-installation.html
23. https://tenniskoenig.blogspot.com/2018/12/hw-week-9-mvc-tool.html
24. https://github.com/tenniskonig/Dokumentation/wiki/Software-Architecture-Document
25. https://tenniskoenig.blogspot.com/2019/05/hw19-metrics.html
26. https://tenniskoenig.blogspot.com/2019/04/hw-week-13-time-and-risk-management.html
27. https://tenniskoenig.blogspot.com/2018/12/hw-week-10-cucumber-running.html
28. https://tenniskoenig.blogspot.com/2019/05/hw-week-15-testing.html
29. https://github.com/tenniskonig/Dokumentation/wiki
30. https://tenniskoenig.blogspot.com/2019/05/hw-week-18-design-patterns.html
31. https://github.com/tenniskonig/Dokumentation/blob/master/Tenniskoenig.pptx
32. https://github.com/tenniskonig/Dokumentation/blob/master/Tenniskoenig-Blog.pdf

BlogBook v1.1,
LaTeX 2$_\varepsilon$ & GNU/Linux.
https://www.blogbooker.com

Edited: June 19, 2019