joinsportsblog.wordpress.com/blog

Contents

1 2016 **5**

1.1 October . . . 6

Blog (2016-10-12 10:06) . . . 7

Home (2016-10-12 10:06) . . . 8

Our idea (2016-10-12 10:06) . . . 9

About us (2016-10-12 10:06) . . . 11

Week 2: Technology choice (2016-10-14 21:23) . . . 16

Week 2: Team roles (2016-10-19 12:40) . . . 18

Week 3: Software Requirements Specification (SRS) (2016-10-23 16:31) . . . 19

Links (2016-10-23 16:49) . . . 20

1.2 November . . . 21

Week 4: Use Case Specifications (2016-11-01 23:15) . . . 22

Week 5: Jira Scrumboard (2016-11-08 09:32) . . . 23

Week 6: Gherkin feature files (2016-11-16 12:32) . . . 24

Week 7: Class Diagram (UML) (2016-11-23 10:04) . . . 25

Week 8: MVP Architecture (2016-11-29 09:44) . . . 27

1.3 December . . . 28

Week 9: Microsoft Project (2016-12-06 15:25) . . . 29

Midterm Handin (2016-12-30 12:00) . . . 30

2 2017 **33**

2.1 April . . . 34

Risk Management & new Scope (2017-04-25 16:59) . . . 35

2.2 May . . . 36

Function Point Calculation (2017-05-05 13:37) . . . 37

Testing (2017-05-09 16:31) . . . 42

Refactoring (2017-05-15 17:21) . . . 44

Pattern (2017-05-23 08:38) . . . 45

Metrics (2017-05-30 08:44) . . . 47

2.3 June . . . 49

Installation (2017-06-12 18:33) . . . 50

1. 2016

1.1 October

Blog (2016-10-12 10:06)

Dies ist nur ein kurzer Auszug deines Blogs.

Home (2016-10-12 10:06)

Hello out there,

we are four students starting a software engineering project at DHBW Karlsruhe. Our project is about an Android application for people which are interested in sport - especially soccer ;) . With our app it will be possible to find places to join sport events and get togehter. There will also be a competive mode for existing teams to compete against others. All the necessary organisation for events will be possible by using this app!

We will develop an Android application, but maybe we will provide a web-based version too :).

Our idea (2016-10-12 10:06)

List of feature ideas:

Our project is about an Android application for people which are interested in sport - especially soccer ;) . With our app it will be possible to find places to join sport events and get togehter. There will also be a competive mode for existing teams to compete against others. All the necessary organisation for events will be possible by using this app!

- Find matches nearby your location.
- Use of Google Maps for visualize all possible offers near your region.
- Connect people with the same interests in sports.
- Use a calendar to manage the game dates and places.
- Chat function to arrange and organize the meetings.
- Creation of teams: you can join your teams or create your own!
- Competitive mode: for ambitious teams to compete against each other and rising up within a ranking. Against weak opponents => less points / against strong opponents => more points).
- View of game results after the events (both captains should agree with the final score).
- Mutual evaluation from each team after a match (regarding fairness, completion of the game, sports niveau, ...).

We hope that we could arouse your interest in our project. And we would like to get some feedback from you!

Dies ist nur eine Kurzfassung der Seite &nbsp;„Kontakt“.

BillChill (2016-10-26 09:42:35)
Hi JoinSports, cool idea, will there be a iOS version too? best regards

joinsportsblog (2016-10-26 10:45:40)
Thank you for your comment. Mainly we will concentrate on the Android version in this project. But if there is enough time we may develop it. Greetings, your JoinsSports - Team

AnKaLu Team (2016-10-26 10:51:59)
Hello JoinSports, i like your idea, especially the Competitive Mode. It connects people on a sporting level which is very important nowadays! Keep on the your work, looking forward to hear more from you! Best regards, AnKaLu-Team

gtagroup (2016-10-26 10:54:50)
Hi JoinSpots! Your project sounds really interesting! Do you plan to add a function which allows the user getting notified about nearby activities? We are looking forward to see your first further results.

joinsportsblog (2016-10-26 11:36:49)
Yes. This function is one of our plans too.

About us (2016-10-12 10:06)

Hello dear reader,

here we want you to tell who we are and which roles we fulfill in our team according to the [1]RUP terminology.

Timo Rautenberg

I'm an information technology student at Baden-Wuerttemberg Cooperative State University Karlsruhe (DHBW). I generally like all kind of technology where you can use computers to automatisize things to save time and make your life easier. I like programming in Java/C++, but I'm always interested in new programming languages. Moreover I enjoy programming with microcontrollers like Arduino Uno. Maybe the project idea has emerged of my addiction to football, because I know the problem very well to find enough people to start a match on free weekends ;-).

Team roles:

- Project Manager
- Implementer
- Tool Specialist
- Graphic Artist

Alexander Bierenstiel

I'm an Information Technology student and enjoying dealing with computer technology, especially writing programs with C++. Unfortunately I don't posses white-taped broken glasses or wear pocket protectors, but I'm feeling nerdy enough in my mind to justify my study.

Team roles:

- Implementer
- Test Designer/Tester
- Tech Writer
- Editor

Dominik Schmitt

Hello, I’m a student at the Baden-Wuerttemberg Cooperative State University (DHBW). I mainly develop web sites and their backend structure. I also love to deal with hardware like the Raspberry Pi or the Odroid to automate tasks. I often use Python, PHP, C++ to fullfil my wishes. But I like to learn new

interesting things like developing an app with Java.

Team roles:

- Implementer
- Test Designer/Tester

Computer engineering student at Universitat Rovira i Virgili (Tarragona). This semester studying at Duale Hochschule Baden-Württemberg. Owner of 3 cats and not a great German speaker.

Team roles:

- Implementer
- Graphic Artist

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

Dies ist nur eine Kurzfassung der Seite &nbsp;„Über“.

Lorenz (2016-10-19 11:35:17)
very clear and to the point. good work

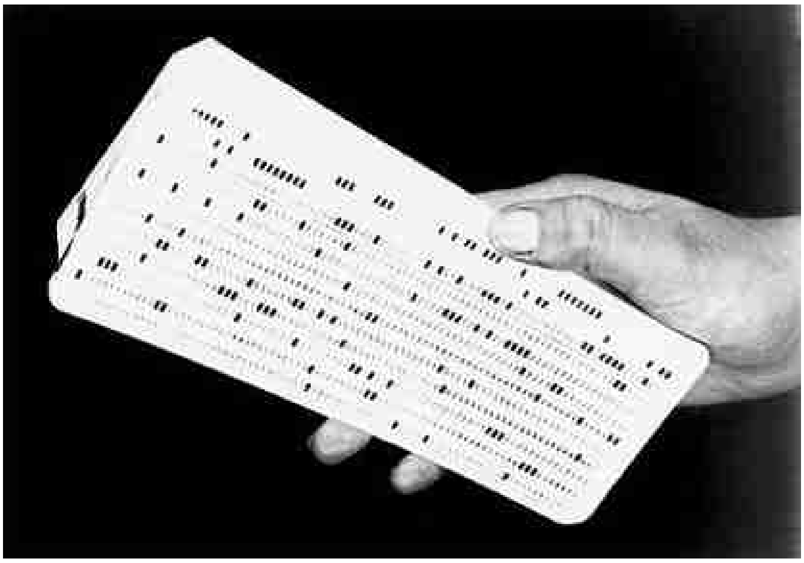
Possseidon (2016-10-19 11:48:02)
Your "About Us"-Page is very nicely structured. Dominik’s picture is looking a bit weird though... :D

Week 2: Technology choice (2016-10-14 21:23)

Hello there,

as it is no longer possible to forge software with our bare hands, as to the good old puch card times, we have to use tools to help us out. We want to present you the tools with which we create our sports app.

[1]



IBM punch card

Programming Language and IDE:
The programming language is the most vital part of a software project. So the programming language has to be chosen carefully.
As we want to create an Android app, we will use Java as programming language and produce code with the help of Android Studio as chosen IDE.

Subversion Tool:
No software project shall ever be touched without full source code controll! As our subversion tool we bet on the trustworthy GitHub plattform. Have at look at our repository: [2]https://github.com/Kaltther/Software-Project

Greetings, your JoinSports-Team

1. `http://www.computerhistory.org/atchm/wp-content/uploads/2015/01/IBM_Punch_Card.png`
2. `https://github.com/Kaltther/Software-Project`

---

gtagroup (2016-10-18 22:12:41)
Hey Sounds like you thougt much about which IDE and programming language to choose - and in our opinion you made a good decision! ;) (absolutely altruistic...) Good idea to use git!

joinsportsblog (2016-10-18 23:30:48)
Thanks for your opinion!

Tobias (2016-10-19 11:44:03)
Hey there Team Joinsports! Using Java as the language for your app is a solid choice. There is a large community of Android / Java programmers that will surely prove useful along the way. Android Studio is also great. Do you know already which Versions of Android you will support (e.g. Android 6.0 Marshmallow?)

SquadIT (2016-10-19 11:44:10)
Hi there, Since your technology choice is pretty obvious for android development i wonder if you're going to build a backend. Do you have any plans for that?

Midterm Handin – JoinSports (2016-12-30 13:55:56)
[...] technology choice [...]

Week 2: Team roles (2016-10-19 12:40)

Hello folks,

if you like to know the diffrent roles which each team member has, take a look in the [1]About Us section. There we discribe the diffrent task of the members and give a short personal introduction.

Greetings, your JoinSports-Team

1. `https://joinsportsblog.wordpress.com/about-us/`

Waldemar (2016-10-26 09:48:27)
Hi JoinSports, the choice of the teamroles seems to be solid for me. I have not found where you broke down your system in managable subsystems, maybe you want to add this information. regards Waldemar

Week 3: Software Requirements Specification (SRS) (2016-10-23 16:31)

Hello folks,

today we want to introduce you to our Software Requirements Specification or short SRS.

In the SRS are all requirements listed, which the software has to fulfill. It is most likely that these requirements will change during the development process. However the SRS will always hold the latest changes, so that software is build according to the specifications of the SRS.

If you want to say so, the SRS is the ruleset for all developers, on how the software has to look like and how it shall be developed.

To find our SRS, follow this link:
[1]https://drive.google.com/open?id=1xFv4Tk2eArceTPSyFU8FZWvju10H5MxulQs5 hfiriww

If you are interessted in our Use Case-Diagramm, follow this link:
[2]https://drive.google.com/open?id=0B9TxrfClR7EIenFXN2JmZmxqVXc

Have a great day, your JoinSports-Team

1. `https://drive.google.com/open?id=1xFv4Tk2eArceTPSyFU8FZWvju10H5MxulQs5hfiriww`
2. `https://drive.google.com/open?id=0B9TxrfClR7EIenFXN2JmZmxqVXc`

BillChill (2016-10-26 09:22:43)
Hey JoinSports-Team, your Use-Case-Diagramm looks promising. Im looking forward to see more from this project. best regards

profdh (2017-01-23 17:42:07)
not helpful

Possseidon (2016-10-26 10:37:35)
You should prioritize some of your points in the use-case-Diagram. You might not be able to fullfill all of them and if you just have a little bit of everything it probably won’t be any good. Also you shouldn’t use arrows and just straight lines without end. But I think you should know this by now. ;)

TEAM Ortus (2016-10-26 10:42:45)
Hi, the Use-Case has still arrows in it. Also the SRS needs some fixes like the headings. best regards, team ortus

Tim Saupp (2016-10-26 10:52:53)
Hi Team JoinSportse, your SRS is not completed please fill out all of the blue default text boxes from the template you’ve been using. Update your UC diagram (don’t use arrows!).

Midterm Handin – JoinSports (2016-12-30 13:55:59)
[...] SRS blog post [...]

Links (2016-10-23 16:49)

Our Github-Repo: [1]https://github.com/joinsports

[1] https://github.com/joinsports

1.2 November

Week 4: Use Case Specifications (2016-11-01 23:15)

Hello folks,

we worked out the first two use cases. The use case specification includes the document itself plus the linked mockup and activity diagram. Have a look for yourself:

1. [1]Create Team
2. [2]Suggest Team Result

Greetings, your JoinSports-Team

1. `https://drive.google.com/open?id=1MpWuQxur6B7U2V-hw1L3mlNFVZxt_IjOV612mEs8xBk`
2. `https://drive.google.com/open?id=1IvY0C70jJfbZq_R0n0R6IFElYvnEcD62yCSvH9-z73Y`

---

energycheetah (2016-11-02 09:57:33)
Hi, I like your use cases, but I have a suggestion. Maybe you could still put the pictures in the document itself, so that we don't have to jump from link to link. Greetings, Cheetah

SquadIT (2016-11-02 10:07:10)
Hi Joinsports! Your Use Case "create team" should be marked as part of a CRUD. Also your second Use Case resolution is very low that i can't really read anything. The screenshots and activity diagrams are all there. In your SRS are both links for both Use Cases. Greetings, SquadIT

gtagroup (2016-11-02 12:25:21)
Hi! We agree with previous comments. Another question about the UML Diagramm "Create Team": if the user has entered a name that already exists, does the session end? It would be more logical if you make a mistake on the display and jump back to "Input Team Name".

Week 5: Jira Scrumboard (2016-11-08 09:32)

Hello out there,

we created a scrumboard in Jira for our project. To view the board follow this [1]link .

We also created and started a sprint for the hello world exercise.

Regards

Team JoinSports

1. `http://193.196.7.27:8080/secure/RapidBoard.jspa?rapidView=101`

---

ankalublog (2016-11-09 12:52:03)
Hey there, It looks like you understood the useage of jira. The Hello world sprint looks really detailed. I'm looking forward to future work! Best regards AnKaLu

gtagroup (2016-11-09 14:47:49)
Hey JoinSports, Your scrumboard looks pretty good, especially because it's well structured. GJ! But don't forget to label your tasks according to RUP :) Maybe calculate some time to discuss / write review etc. in your next sprint? Greetings GTA Project <3

Week 6: Gherkin feature files (2016-11-16 12:32)

Hello out there,

we recently added feature files in our Use-case documentation on Github. You can find the feature files on the following two links:

UC _CreateTeam:
https://github.com/JoinSports/Documentation/blob/master/UC/UC _CreateTeam.md

UC _SuggestMatchResult:
https://github.com/JoinSports/Documentation/blob/master/UC/UC _SuggestMatchResult.md

Greetings, your JoinSports-Team

The Mathinator (2016-11-21 11:57:19)
Hi JoinSports-Team, the feature files are correct and enough detailed. Hope it will work. Keep Going! Regards. Mathinator

gtagroup (2016-11-23 12:06:56)
Hi there, your feature files seem to be complete and you thought about enough scenarios. Well done! lovely greetings <3 GTA-Group

Midterm Handin – JoinSports (2016-12-30 13:56:08)
[...] feature files [...]

Week 7: Class Diagram (UML) (2016-11-23 10:04)

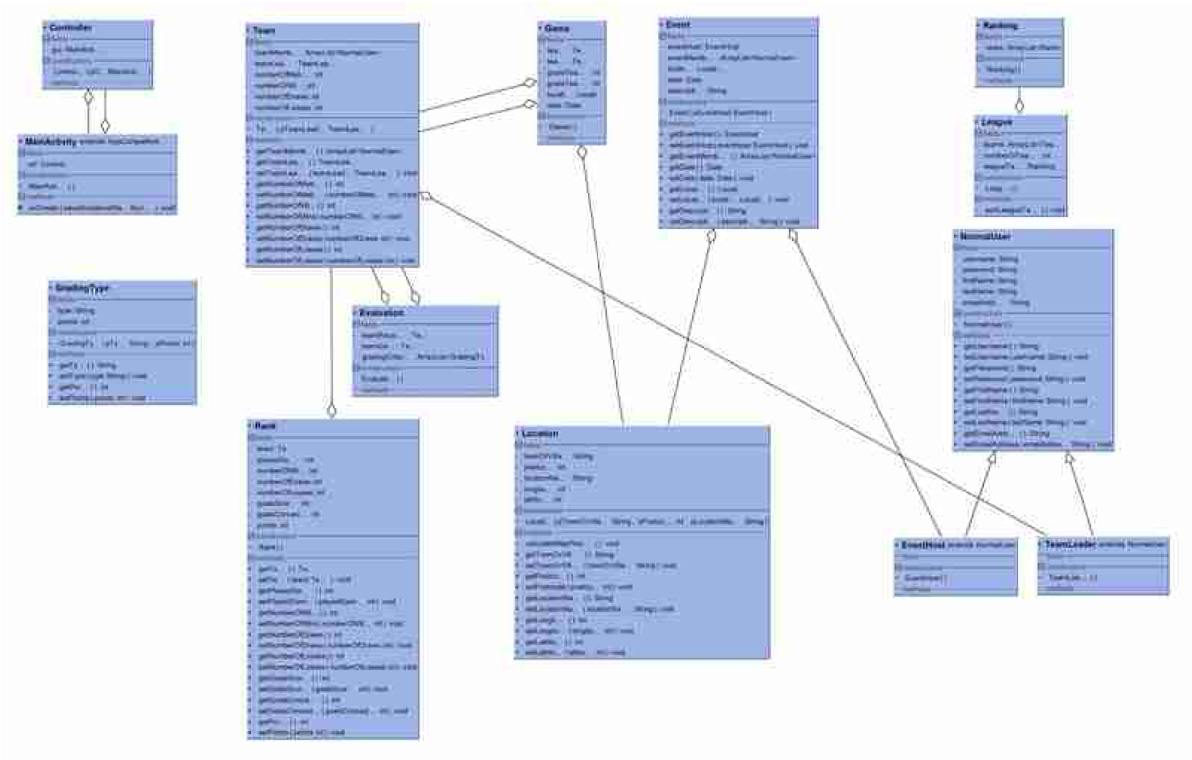
Hello folks,

during the last week we built a code structure of our Android application. We implemented the first important classes for our project in Android Studio.

For generate and visualize the different classes we use the plugin simpleUMLCE, which can be downloaded from the jetbrains website following this [1]link.

But now, here you can see the generated UML-Diagramm from the used simpleUMLCE plugin.

[2]



Grettings, your JoinSports-Team

1. `https://plugins.jetbrains.com/plugin/243`
2. `https://joinsportsblog.files.wordpress.com/2016/11/classdiagram_uml_cut.jpeg`

---

ankalublog (2016-11-23 11:31:14)
Hey there, your Class diagramm looks really detailled. I like that you worked with the simpleUMLCE plugin. It lookes really good with your work and brings structure to it. Best regards, AnKaLu

squaditblog (2016-11-23 12:05:37)
Hi there, As Ankalu already stated, your class diagram is very detailled. Is it intended, that your arrows are shaped so weirdly? Why are there multiple arrows from one object to another object? Isn't one enough? The Class "GradingType" isn't connected with anything, is that intended? The rest looks pretty complete. Good job. Greetings SquadIT

joinsportsblog (2016-11-23 12:30:38)
Hi, thank you for your opinion. But the diamond form is the typical UML syntax to express relations between classes. It defines an "aggregation". If there are more than one aggregation arrow it means that in every class there is created an object from the other class. The reason why the class "Grading Type" isn't connected with another class: In reality there are some more relations (like between "Grading Type" & "Evaluation") but the objects are organized in an ArrayList. That's why the simpleUMLCE plugin can't detect the relation. Maybe if we would add the class ArrayList to the UML diagram there would be a relation arrow between "ArrayList" & "GradingType". (You can get some further information here: https://de.wikipedia.org/wiki/Klassendiagramm) Greetings, your JoinSports-Team

Midterm Handin – JoinSports (2016-12-30 13:56:11)
[...] class diagram [...]

Week 8: MVP Architecture (2016-11-29 09:44)

Hello out there,

this week we filled in the SAD document, which you can find [1]here.
Android doesn't use the typical MVC standard. The View part is implemented by XML-files, which have to be loaded by Activity-Classes. This classes are a mixture of View and Controller. That's why our Architecture is more like a MVP (Model View Presenter).

Greetings, the JoinSports-Team

1. `https://github.com/JoinSports/Documentation/blob/master/Architecture/Architecure.md`

---

energycheetah (2016-11-29 10:41:36)
Hey, I like your SAD, but in Section 3 I would add a picture from your MVC tool. For example this is for rails: https://shijitht.files.wordpress.com/201 0/09/mvc.png Cheers

BillChill (2016-11-29 12:09:47)
Hi JOINSPORTS, you prepared your SAD well, but I think your mvc-overview is a bit overloaded. Best regards BillChill

Midterm Handin – JoinSports (2016-12-30 13:56:14)
[...] MVP (blog post) [...]

1.3 December

Week 9: Microsoft Project (2016-12-06 15:25)

Hello visiters,

recently we started our long term planning with Microsoft Project. This tool helps you to present the work that you have done in an overview.

You can find our long term plan made with MSProject aswell as the gantt chart in our github repository under the folder MSProject. ([1]Click here)

Greetings, the JoinSports-Team

1. `https://github.com/JoinSports/Documentation/blob/master/MSProject/Software%20Engineering%20Longterm%20Plan.pdf`

---

squaditblog (2016-12-07 10:50:21)
Hi there :) Your long term plan looks really nice. It seems as if you got all necessary information in it. It would be nice to see a picture of your Gantt chart (The PDF export of MS Project looks always horrible :'(...). Keep up the good work, Greetings from SquadIT

ankalublog (2016-12-07 11:07:06)
Hey there, Your plan at the beginning of the document looks good, but you can't really see anything from the actual Gantt Chart in the end. It would be nice if you could fix that, so that it's possible to see your working process visualised with the critical path. Best regards AnKaLu

Midterm Handin – JoinSports (2016-12-30 13:56:18)
[...] msproject long term planning (blog post) [...]

Midterm Handin (2016-12-30 12:00)

Hi,

today we created a complete overview of all parts we have done for our project so far.

Below you can find all links in a list beginning from week 1 to week 9 and our midterm presentation (slides and demo).

week 1:

- Setting up the blog ;)
- [1]vision

week 2:

- [2]team roles / about us
- [3]GitHub documentation
- [4]GitHub code APP
- [5]GitHub code PHPConnector
- [6]technology choice

week 3:

- [7]SRS blog post
- [8]SRS document (direct link)

week 4:

- UC:

1. [9]Create team
2. Delete team
3. Create user
4. Login user
5. Delete user
6. Update user
7. [10]Suggest match result

week 5:

- [11]Scrumboard (Jira)
- [12]Jira sprint 1 (Hello World)
- [13]Jira sprint 2 (Midterm sprint)

week 6:

- [14]feature files
- robotium
- cucumber

week 7:

- [15]class diagram

week 8:

- [16]MVP (blog post)

- [17]SAD

week 9:

- [18]msproject long term planning (blog post)
- [19]msproject long term planning (documents)

midterm:

- [20]presentation slides
- working demo

1. `https://joinsportsblog.wordpress.com/our-idea/`
2. `https://joinsportsblog.wordpress.com/about-us/`
3. `https://github.com/JoinSports/Documentation`
4. `https://github.com/JoinSports/AndroidApp`
5. `https://github.com/JoinSports/PHPConnector`
6. `https://joinsportsblog.wordpress.com/2016/10/14/technology-choice/`
7. `https://joinsportsblog.wordpress.com/2016/10/23/software-requirements-specification/`
8. `https://drive.google.com/open?id=1xFv4Tk2eArceTPSyFU8FZWvju10H5MxulQs5hfiriww`
9. `https://github.com/JoinSports/Documentation/blob/master/UC/UC_CreateTeam.md`
10. `https://github.com/JoinSports/Documentation/blob/master/UC/UC_SuggestMatchResult.md`
11. `http://193.196.7.27:8080/secure/RapidBoard.jspa?rapidView=101`
12. `http://193.196.7.27:8080/secure/RapidBoard.jspa?rapidView=101&projectKey=JOIN&view=reporting&chart=burndownChart&sprint=275&estimate=field_timeoriginalestimate`
13. `http://193.196.7.27:8080/secure/RapidBoard.jspa?rapidView=101&projectKey=JOIN&view=reporting&chart=burndownChart&sprint=363&estimate=field_timeoriginalestimate`
14. `https://joinsportsblog.wordpress.com/2016/11/16/feature-files/`
15. `https://joinsportsblog.wordpress.com/2016/11/23/class-diagram-uml/`
16. `https://joinsportsblog.wordpress.com/2016/11/29/mvp-architecture/`
17. `https://github.com/JoinSports/Documentation/blob/master/Architecture/Architecure.md`
18. `https://joinsportsblog.wordpress.com/2016/12/06/week-9-microsoft-project/`
19. `https://github.com/JoinSports/Documentation/tree/master/MSProject`
20. `https://github.com/JoinSports/Documentation/blob/master/Midterm%20Presentation/Midterm%20Presentation%2014.12.16.pptx`

2. 2017

2.1 April

Risk Management & new Scope (2017-04-25 16:59)

Hey there,

we are back after the Midterm and we did a lot of work. We redefined our Scope for the new semester and did some risk managment. Below you can find the links to the new scope and the risk managment table. We prepared our time spent per UseCase for the next week.

Scope: [1]click me

Risk managment: [2]click me

Time spent: [3]click me

1. `https://drive.google.com/open?id=0B9TxrfClR7EIc25CZFpTQTY2Q28`
2. `https://docs.google.com/spreadsheets/d/1Ol4IDfD7qA2mtBJi1oEv4PtsS9QAoELNkzqZYnkeOJA/edit?usp=sharing`
3. `https://docs.google.com/spreadsheets/d/1ry6spo5wNM3ulQopz1wHvXCId5N6SyQ6MX0vt9oaxos/edit?usp=sharing`

---

gtagroup (2017-04-27 13:51:16)
Hey there, your risk management seems very good, it's even more detailed than expected by the criteria :) The timetable looks alright, too. All in all you did a good job. Lovely greetings <3

Sim91 (2017-05-02 10:37:29)
Hey Team Joinsports, your Risk-Management looks complete and nicely structured. It is also very understandable for outsiders, like ourselves. Your timetable makes it very clear, that you have used up a lot of time for testing, which shows, how important testing is, for such a project. Concluding, I can only say: "Nice job!" Greetings, TextVenturer

2.2 May

Function Point Calculation (2017-05-05 13:37)

Hi everybody,

today we want to show you the function point estimation of our Use Cases (UC).

Function points (FP) are used for estimating the time that will be spent on certain UCs.
They are calculated in reference on External Inputs, External Ouputs, External Inquiries, Internal Logic Files and External Interface File.

Description of these characteristics:

External Inputs:
How many Input Fields?

External Ouputs:
How many Outputs will be generated for the user to see?
–> number of reports generate, not the number of fields of data on each report

External Inquiries:
How many times the user has to interact with the program without entering any data in?
(buttons, menus or anything the user has to interact with without inputting data)

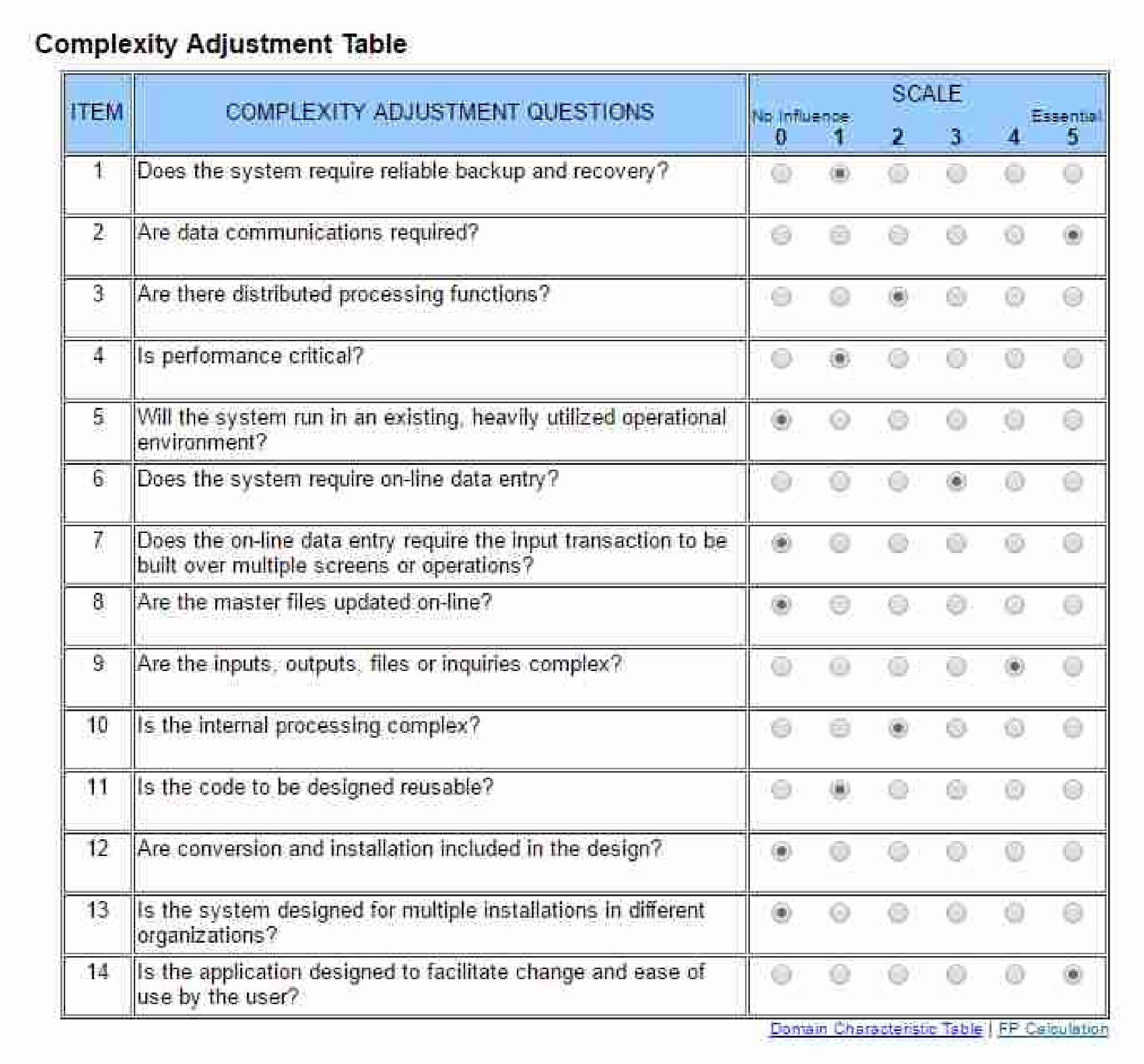
Internal Logic Files:
How many files that the software you plan to develop will have to access?
–> only includes any data files that your software will have to interact with
–> does not include header files, library files or any files that your compiler will need to compile your software

External Interface File:
How many other systems or software that your software will have to interface with?
(a piece of database software, a web-based server or any other object that your software will have to interface with)

Through Data Element Type, Record Element Type and File Type Reference we identified the complexity which can be LOW, AVERAGE or HIGH. In our UCs these parameters are always "LOW", because of the simplicity of the UCs.
To estimate how long it will take to implement a Use Case in future, function points of UC that have been already implemented are put in relation to the time we spent on it.

For the calculation of our function points values we used the tool [1]"Tiny Tools".

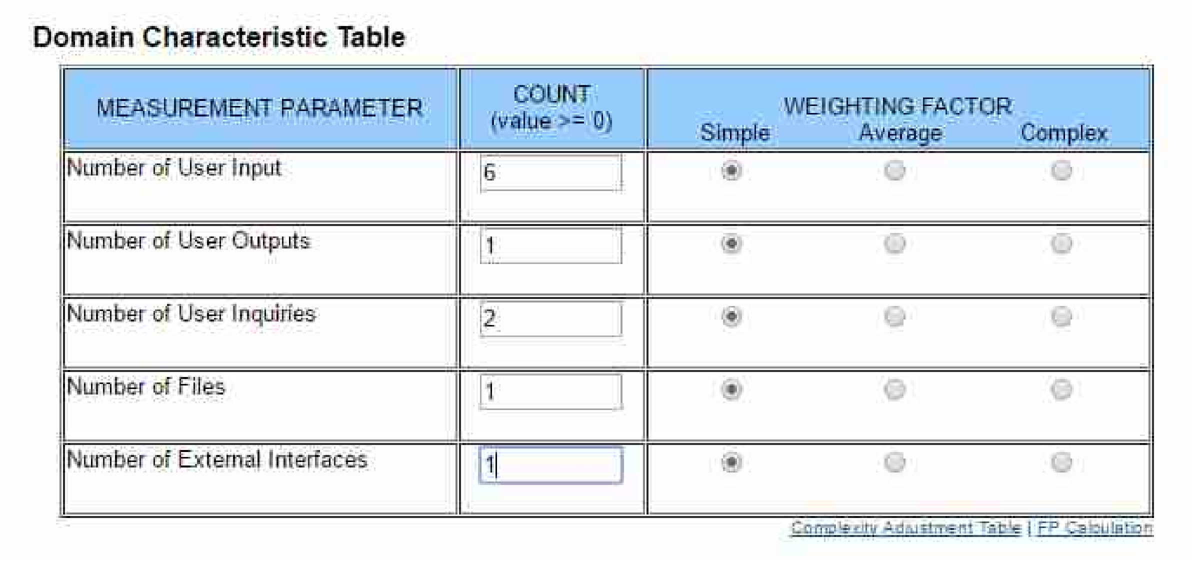
For the calculation of FP with TINY TOOL you need to fill in some extra information about the whole application called "Complexity Adjustment Questions".
This settings are valid for all the UCs and must only defined once.



Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	No Influence		SCALE			Essential
		0	1	2	3	4	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?	○	○	○	○	○	●

To calculate the FP values, you have to fill in the "Domain Characteristic Table" for each of the several UCs. (Example: UC - CreateEvent)

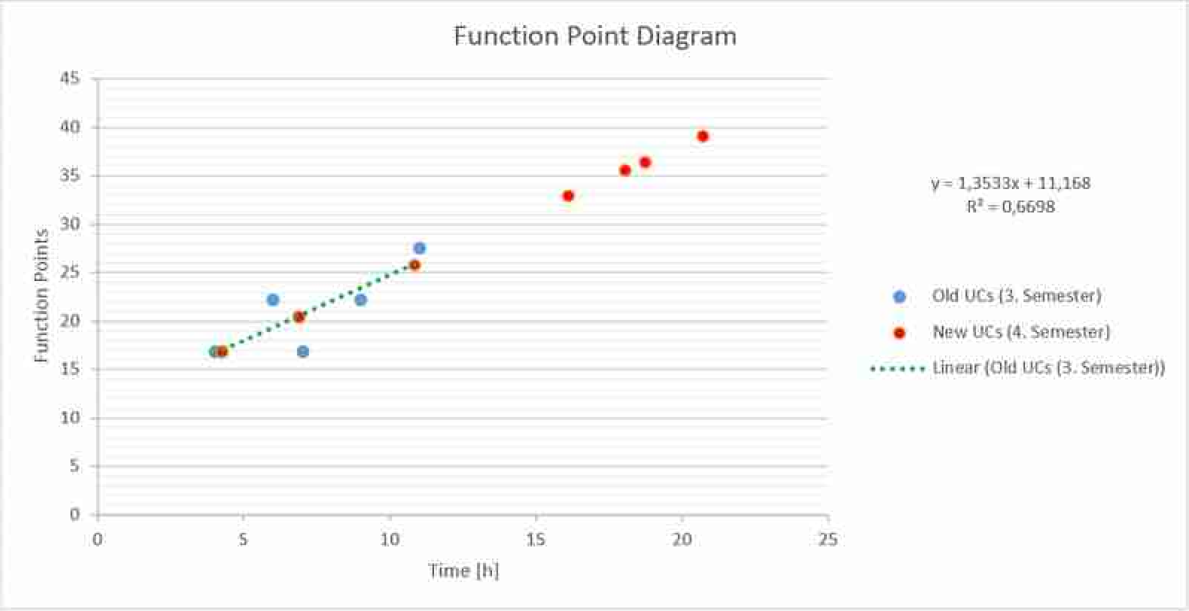


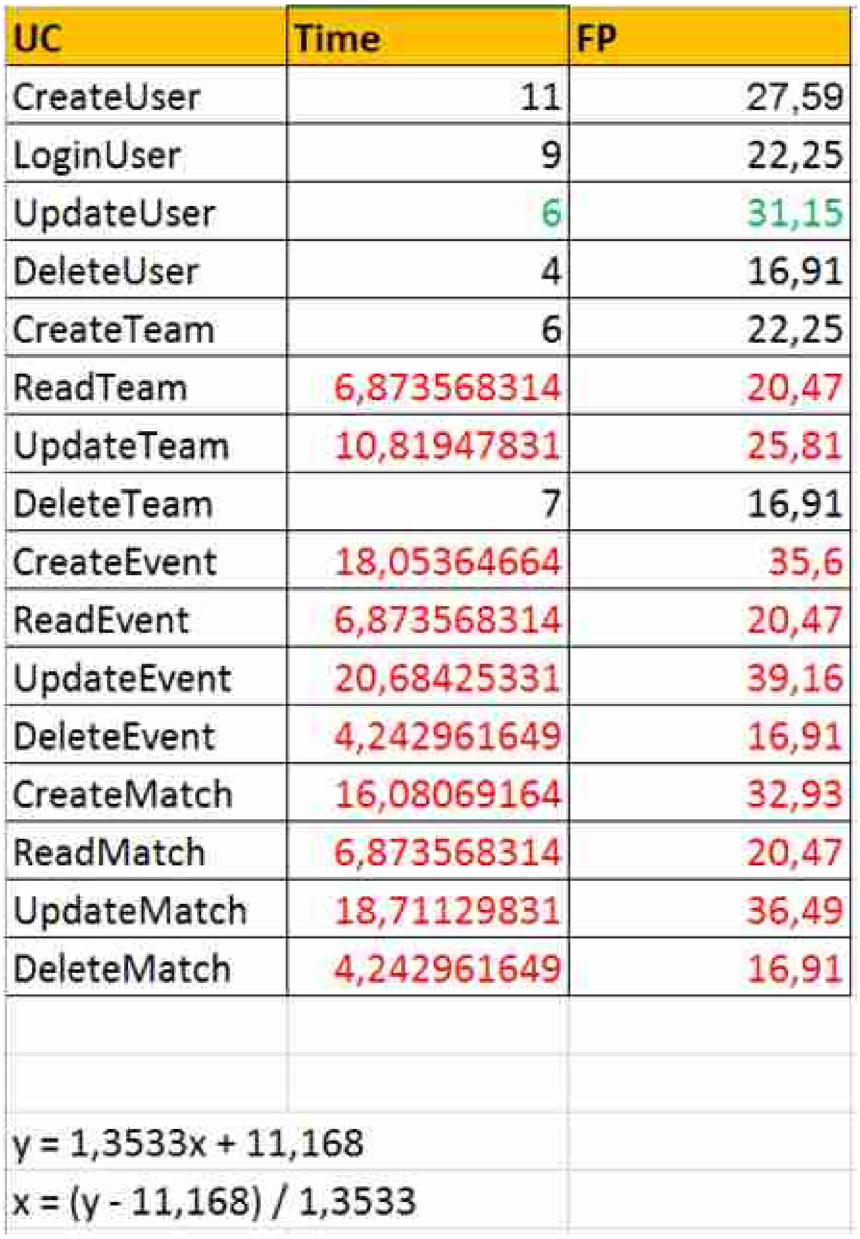
Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR		
		Simple	Average	Complex
Number of User Input	6	●	○	○
Number of User Outputs	1	●	○	○
Number of User Inquiries	2	●	○	○
Number of Files	1	●	○	○
Number of External Interfaces	1	●	○	○

On this [2]link you can find the whole table of our FP which contains our already existing UCs and the new UCs we want to implement within this semester.

Based on these data table we created a Function Point Diagram. After excluding the outliner UC points, we calculated a regression line.
Using this regression line we we were able to estimate the working time of the new UCs. So it is easier for us to plan time duration of prospective UCs.





UC	Time	FP
CreateUser	11	27,59
LoginUser	9	22,25
UpdateUser	6	31,15
DeleteUser	4	16,91
CreateTeam	6	22,25
ReadTeam	6,873568314	20,47
UpdateTeam	10,81947831	25,81
DeleteTeam	7	16,91
CreateEvent	18,05364664	35,6
ReadEvent	6,873568314	20,47
UpdateEvent	20,68425331	39,16
DeleteEvent	4,242961649	16,91
CreateMatch	16,08069164	32,93
ReadMatch	6,873568314	20,47
UpdateMatch	18,71129831	36,49
DeleteMatch	4,242961649	16,91

y = 1,3533x + 11,168

x = (y - 11,168) / 1,3533

(Some UCs have the same FP values, that is why you can not see all the new UCs in the diagram, because the points are overlapped.)

We hope you liked it and we are looking forward to some feedback.

Greetings,
the JoinsSports-Team

1. http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/harvey/FP_Calc.html
2. https://docs.google.com/spreadsheets/d/1bN4NsM-U_6mVvRkdjg4H5C8xJPyxioUnHC0WsJpidxQ/edit#gid=0

billchill (2017-05-08 22:33:56)
Hi joinsports, you described your implementing time of your left usecases detailed and replicable. I like your well created chart. Best regards BillChill

TheMidnightExpress (2017-06-10 14:05:28)
Hi Joinsports, you described everything very well. I liked the link which show all the usecases in detail. You have a lot though. Good work! Regards TheMidnightExpress

Testing (2017-05-09 16:31)

Hi,

this week we introduced testing to our app. We will use three different kinds of testing.

- JUnit
- Espresso
- user testing

We will use Coveralls with Travis CI to automate our coverage reports.

This week we wrote only one simply test. The main part was setting up the whole part with Travis CI and Coveralls (thrust me it’s a mess).

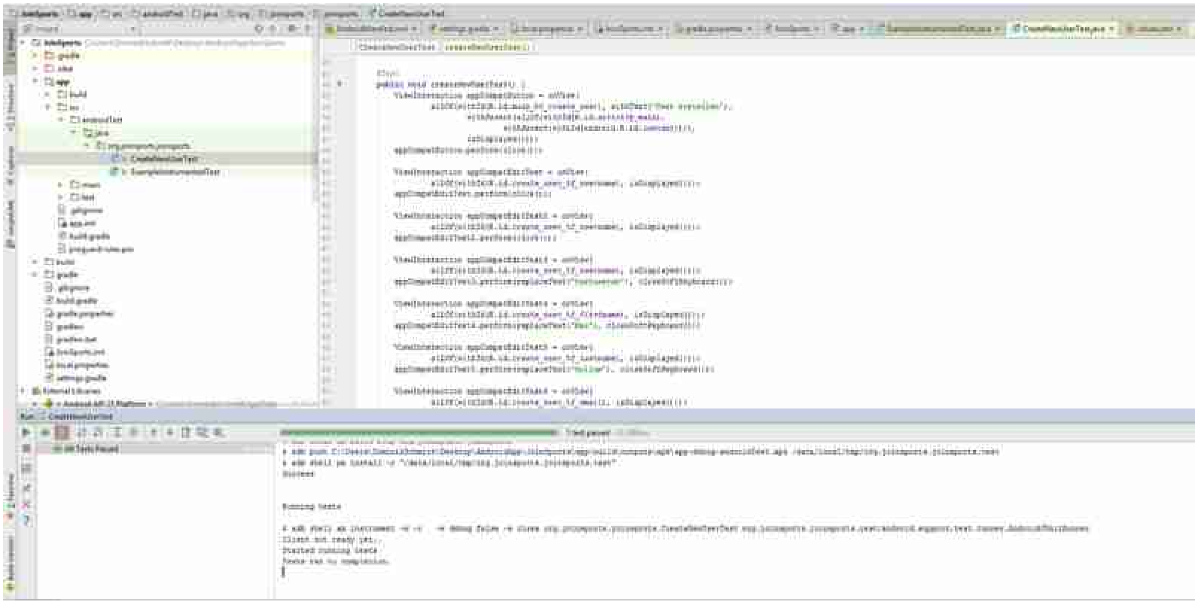
Here is the link to the first running test file: [1]click me

Here is our modified build.gradle (please note we did some work on the travis.yml, too):

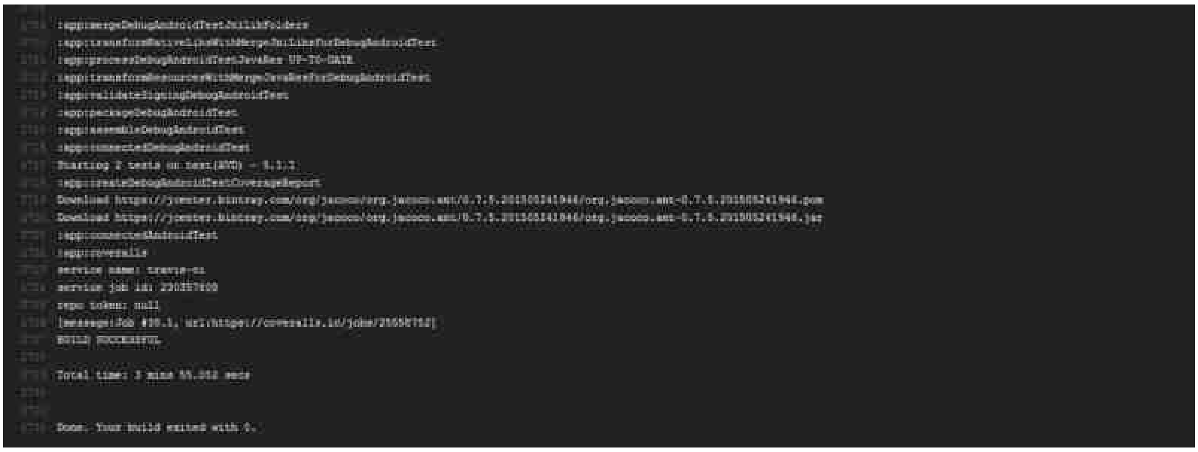
[2]Build.gradle

[3]Travis.yml

The running test looks like this in Android Studio [4](large version)



And like this in Travis CI [5](large version)



```
:app:mergeDebugAndroidTestJniLibFolders
:app:transformNativeLibsWithMergeJniLibsForDebugAndroidTest
:app:processDebugAndroidTestJavaRes UP-TO-DATE
:app:transformResourcesWithMergeJavaResForDebugAndroidTest
:app:validateSigningDebugAndroidTest
:app:packageDebugAndroidTest
:app:assembleDebugAndroidTest
:app:connectedDebugAndroidTest
Starting 2 tests on test(AVD) - 5.1.1
:app:createDebugAndroidTestCoverageReport
Download https://jcenter.bintray.com/org/jacoco/org.jacoco.ant/0.7.5.201505241946/org.jacoco.ant-0.7.5.201505241946.pom
Download https://jcenter.bintray.com/org/jacoco/org.jacoco.ant/0.7.5.201505241946/org.jacoco.ant-0.7.5.201505241946.jar
:app:connectedAndroidTest
:app:coveralls
service name: travis-ci
service job id: 230357800
repo token: null
[message:Job #30.1, url:https://coveralls.io/jobs/25650752]
BUILD SUCCESSFUL

Total time: 3 mins 55.052 secs


Done. Your build exited with 0.
```

Finally we created out testing plan. You can find it [6]HERE

We hope you enjoyed this weeks blog post. See you next week ;)

1. `https://github.com/JoinSports/AndroidApp/blob/TestingTestCodeCoverage/JoinSports/app/src/androidTest/java/org/joinsports/joinsports/CreateNewUserTest.java`
2. `https://github.com/JoinSports/AndroidApp/blob/TestingTestCodeCoverage/JoinSports/app/build.gradle`
3. `https://github.com/JoinSports/AndroidApp/blob/TestingTestCodeCoverage/.travis.yml`
4. `https://raw.githubusercontent.com/JoinSports/Documentation/master/Testing/AndroidStudio.png`
5. `https://raw.githubusercontent.com/JoinSports/Documentation/master/Testing/TravisBuildTest.png`
6. `https://github.com/JoinSports/Documentation/raw/3201a4ff2d7e60235e0b0b6c8f94f29cde696b2f/Testing/Master%20Test%20Plan.pdf`

---

energycheetah (2017-05-16 08:56:33)
Hey guys, nice post ;) But the link of the testing plan doesn't seem to work. Cheers Cheetah

AnKaLu Team (2017-05-30 09:18:48)
Hey guys, we like the structure of your post. Equally the classification of your Human Resources looks well structured and well considered. We can see nothing negative in your post. Well done! Best regards AnKaLu-Team

TheMidnightExpress (2017-06-10 14:13:46)
Hi JoinSports, I like your post. It is well structured and into detail. We use Travis CI, too. I think it's also a good solution for your project. Good Job! Regards TheMidnightExpress

Refactoring (2017-05-15 17:21)

Hey,

this week we did some refactoring ourselfs.

Below you can find a list of the repositories:

- [1]Alexander
- [2]Dominik
- [3]Timo

Thats all for this week. As you maybe noticed our coverage changed since the last time. We're updating and creating tests right now.

See ya next week.

Team JoinSports

1. `https://github.com/abieren/SE_Refactoring`
2. `https://github.com/dominik-code/CodeSmells`
3. `https://github.com/Raute37/SoftwareEngineering_CodeRefactoring`

---

gtagroup (2017-05-16 10:23:39)
Hey joinsports, It seems you did a good job doing your homework! Although Dominik should have a few more commits ;)
lovely greetings <3

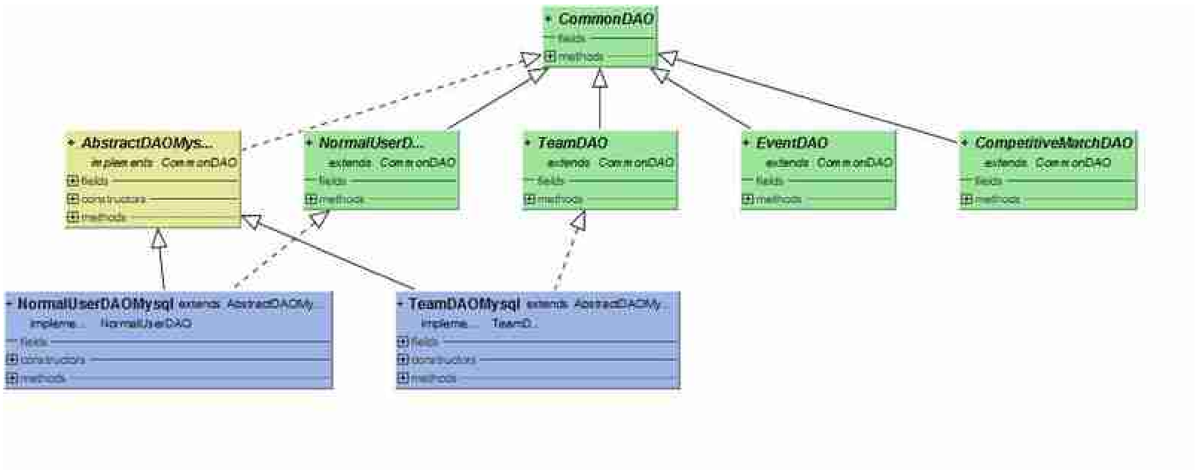
themidnightexpress (2017-05-16 13:32:02)
hey joinsports, looks solid. short and simple commit messages. good work. kisses, themidnightexpress

Pattern (2017-05-23 08:38)

Hey,

this week we introduced a pattern to our code.

We are using the default MVP of Android as base. Because we needed more options we created our own pattern to access the database. Below you can see a part of the class diagram. It clearly show where the way is leading to. We are using a DAO Pattern which means "direct access object". We introduced for each database object a new class which implements our DAO pattern.



As you can see the NormalUserDAOMysql uses the AbstractDAOMysql Class and the NormalUserDAO Class which generally is helping us to exchange with our Mysql backend.

You can find the full new class diagram: [1]here

That’s all for this week.

See ya next week again.

[1] `https://raw.githubusercontent.com/JoinSports/Documentation/master/Class-diagram-UML/ClassDiagram_new.jpg`

---

vnvproject (2017-05-23 09:33:57)
Hey JoinSports! I really like your DAO pattern. The class diagramm is overseeable and points out the benefits of the DAO pattern. It seems to make your life more easy. But aren’t they called "Data Access Objects"? :) Keep up the good work! Greetings, vnv

SquadIT (2017-05-23 10:16:50)
Hey JoinSports, your pattern looks good. It would have been nice to see the difference to your previous class diagram directly in your blogpost. Greetings, SquadIT

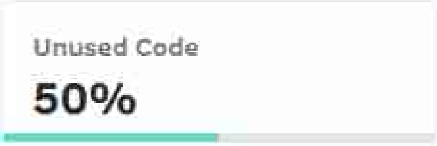
TheMidnightExpress (2017-06-10 14:19:17)
Hi JoinSports, I really like your pattern. You can oversee everything quiet well. A before/ after comparison would be nice. Nevertheless: Good Work Regards TheMidnightExpress

Metrics (2017-05-30 08:44)

Hey,

this week we did some work on metrics. First of all we used codacy. We configured the project there and now it will be automatically analyzed if someone does commit code to GitHub. Now there is a badge in our repository showing a grade for the code [1]see here. As metrics we will show complexity and unused code. As you can see on the images below we got a lot unused code as of today. This will change after the backend is implemented and the database is used by the user interface. The complexity is calculated per file (at the moment writing this there is a bug with codacy not showing an average percentage over all files). It uses lines of code, method count, method length, number of control flow statements (like "if,switch,...") and function parameters count.

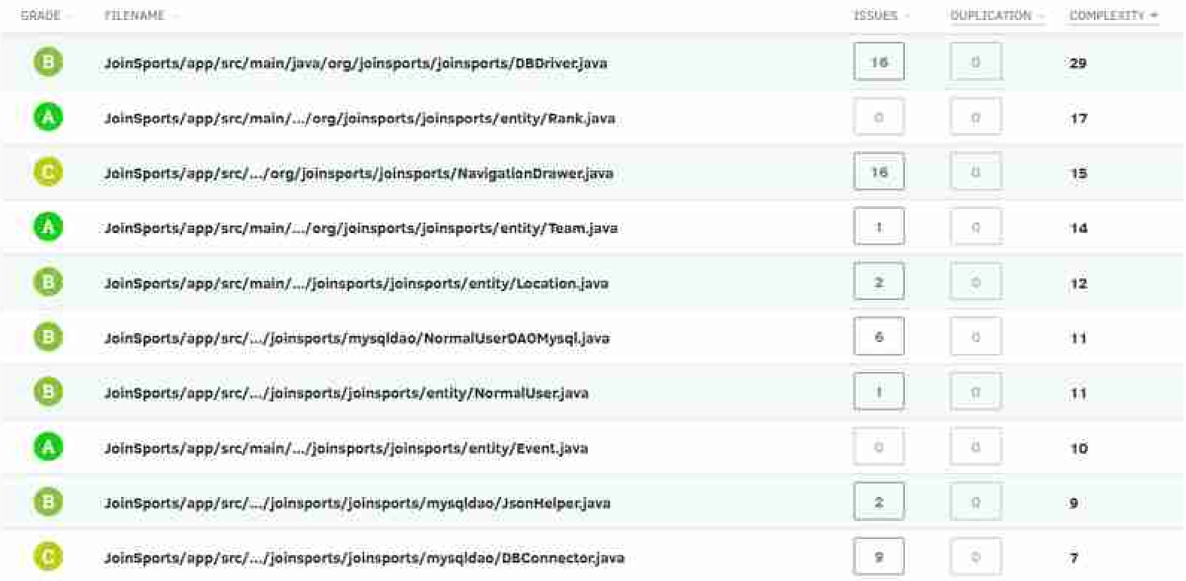
The current percentage of unused code is pretty high:



Unused Code

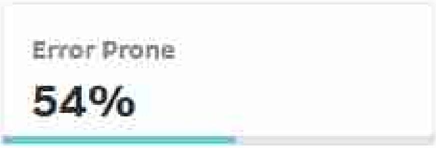
50%

The complexity is shown per file. Higher complexity count is considered bad (wiki page tells us below 30 is considered good). Everything seems fine here.



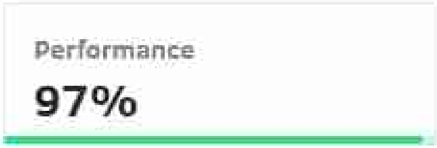
GRADE	FILENAME	ISSUES	DUPLICATION	COMPLEXITY
B	JoinSports/app/src/main/java/org/joinsports/joinsports/DBDriver.java	16	0	29
A	JoinSports/app/src/main/.../org/joinsports/joinsports/entity/Rank.java	0	0	17
C	JoinSports/app/src/.../org/joinsports/joinsports/NavigationDrawer.java	16	0	15
A	JoinSports/app/src/main/.../org/joinsports/joinsports/entity/Team.java	1	0	14
B	JoinSports/app/src/main/.../joinsports/joinsports/entity/Location.java	2	0	12
B	JoinSports/app/src/.../joinsports/mysqldao/NormalUserDAOMysql.java	6	0	11
B	JoinSports/app/src/.../joinsports/joinsports/entity/NormalUser.java	1	0	11
A	JoinSports/app/src/main/.../joinsports/joinsports/entity/Event.java	0	0	10
B	JoinSports/app/src/.../joinsports/joinsports/mysqldao/JsonHelper.java	2	0	9
C	JoinSports/app/src/.../joinsports/joinsports/mysqldao/DBConnector.java	9	0	7

Error prone is high right now. The main reason is the unfinished/unused code. The Perfomance of the working code is pretty good.



Error Prone

54%



Performance

97%

That's all for this week, we are doing our best to improve the code usage and decrese the error prone ratio.

See ya next week,

Team JoinSports.

1. `https://github.com/JoinSports/AndroidApp`

gtagroup (2017-05-30 12:05:23)
Hey there, It seems like you know how to use Codacy - looks pretty good. Are these the values before or after you refactored your code? It would be very interesting to see the progress :) Lovely greetings <3 GTA-Group

TheMidnightExpress (2017-06-10 14:22:08)
Hi JoinSports, I really like your progress and I think that you will be doing good on decreasing your unused code and the errors as the project will develope. I nice chart with your progress on getting better would be good. Regards TheMidnightExpress

2.3 June

Installation **(2017-06-12 18:33)**

Hi everybody,

as you probably know, the 4. Semester is almost finished and the end of our project work is approaching fast. Therefore the last weekly homework was to write an Installation Guide for installing our application on other systems.

So if you want to test our JoinSports Android App just follow the steps below:

1. First of all you need an Android smartphone.
2. Download the "JoinSports.apk"-File on the following [1]link.
3. Copy the file to the on your mobile device.
4. Go to your settings menu on your smartphone and enable the installation of applications from unknown sources.
5. Go to the corresponding file path on the file system and open the file. => The Installation of the JoinSports-App is now running.
6. After the installation succeeded the system has created a shortcut icon in the phone menu. If you like you can drag and drop the icon on your home screen to start the app quicker.
7. The installation process is now finished. To run the app just click on the created shortcut. Have fun!

We hope you like the JoinSports-App and we are looking forward to some feedback.

Additional info:
Within the next weeks we complete our Software Engineering Project and publish our last Final Handin. Stay tuned!

Greetings,
the JoinsSports-Team

1. `https://drive.google.com/open?id=0B9TxrfClR7EIdlhDaE9scEhKX2M`

---

scalaqua (2017-06-13 15:55:19)
Hey JoinsSports-Team, We Installed your App with your instructions and it worked. It’s vary easy to install your app. Greatings TextVenturer

BlogBook v0.9,
LaTeX 2$_\varepsilon$ & GNU/Linux.
`https://www.blogbooker.com`

Edited: June 18, 2017