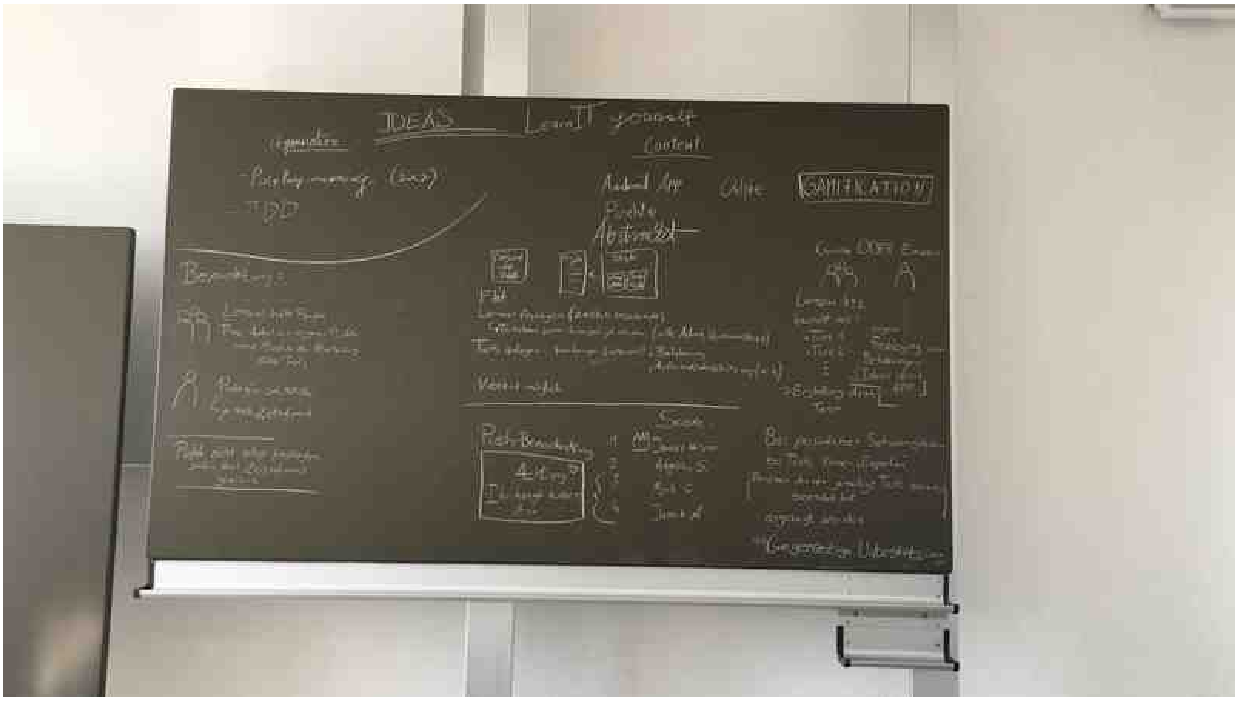
learnityourselfdhbw.wordpress.com

Contents

1. 2018

1.1 October

Gamificication: Learn it yourself (2018-10-05 16:25)



Our Vision? Applying the concepts of gamification to your studies and tasks in a way no one ever has! Our team consisting of Angelika S., Jannik A., Jonas H. and Mert G. plans to develop an App for Android that will blow up your motivation levels to a maximum.

The goal is to develop an application which enables users to turn their mundane learning goals into a game, alone or with others. Get points for the completion of tasks and spend them on personal rewards or just strive to be the best member in your group.

However the competitive nature of all of us is not the only thing that will be used for motivation. Every participant in a group may reflect upon him/herself by giving an individual rating after finishing a task. This way you will keep track on the parts you have mastered well, but also areas where you still could need some polishing.
As an additional reward factor we will be adding a series of achievements, e.g. being the first one in a group to finish a task. Of course you'll be rewarded with little cosmetics, for instance special icons to be able to show off how much you already achieved!

Apart from this, the app should also offer various other features, for example:

- A Simple account system using databases and a server so that you everyone has their own respective accounts.

- Creation of “Missions” which consist of a group of people partaking. We leave you plenty of room as to how you may define a Mission, since it does not necessarily have to be about studying. Be creative! Missions will then be partitioned into various tasks, so you have a clear separation on the different areas you will engage in.
- Points for completing tasks, which will serve as a Highscore system to help you compare yourself with others and track your own improvements.
- Statistics with neat little graphs inside of your groups, to provide you with a quick grasp on who is first and who is last. On top of that, you will receive push notifications if you’re last to make sure you step up your game.
- A reward system outside of your groups. You’ll be able to individualize the amount of points rewarded for each task. In addition to this, you can define your own personal rewards, such as eating a Snickers. Try to be fair to yourself!

Make sure to follow us to see this project take shape and be up to date to the newest features.

SofwareEngineeringEnthusiast (2018-10-10 08:52:46)
Hi Guys, thank you for giving us a first impression of your project. We are very interested in your project and your progress. We are looking forward to get further insights on your project, what roles you fill out, what project management tools you will be using and which features we can expect from you. Keep up the good work! Your travelling Software Engineering Enthusiasts :D

learnityourselfdhbw (2018-10-10 11:01:18)
Thanks for the review! We are glad to hear that our vision sparks some interest in you. We will update you on the technologies we will be using and on our team roster very soon! Sincerely, MertGündüz@learnityourselfdhbw

flashcardcommunity (2018-10-10 10:06:09)
Hello there, your idea sounds very interesting and may be useful when it comes to learning for future exams. But for now we’re interested how you will structure your team. We’re looking forward to see your project developing over the next months! Keep up your work! Greetings, FlashCardCommunity

learnityourselfdhbw (2018-10-10 10:47:06)
Hey Guys, that’s good to hear that you like our general idea. Our Application though is not only intended to be used for exams, also for every other learning Goal or "Mission" you want to accomplish. We are going to publish our team structure in our next blog post. One more advice for you is that you should edit your signature to make is visible who of your team wrote the comment (Prof. Berkling mentioned it). Thank you and have a great day :) Sincerely, JannikAdam@learnityourselfdhbw

flashcardcommunity (2018-10-10 11:56:33)
Thanks for this information, sadly one can’t edit the comment on other pages afterwards. Greetings, Moritz@FlashCardCommunity

DHBWieWarsEssen (2018-10-18 12:40:17)
Hi guys, thanks for this blog post about your project. Your project is really interesting and I can see it being very useful and beneficial. While reading this I could see it being used to motivate 1th-9th graders with the competetive aspect. You also mentioned "personal rewards" in the blog post but I couldn’t immediatly think of anything that could be. So giving an example would be really helpful there. Im really hyped to see more of your blog post and the progression of the project. Best regards,

Hasan@DHBWieWarsEssen

learnityourselfdhbw (2018-11-21 09:55:18)
Hi, personal rewards can be everything that motivates you for example watching a movie or eating a cookie. Everyone can generate them by themselves to make sure that they really motivate the individual. Best regards, Angelika@learnITyourself

Midterm summary – Learn it yourself (2018-12-11 14:00:19)
[...] Project Vision [...]

Introduction of team roles and technologies (2018-10-10 10:41)

This week we came up with a fresh team roster depicting every team-member’s role! On top of that, we’ve settled down on what technologies we will be using.

Our Roles

- Project Manager: Jonas H.
- Tool Specialist: Mert G.
- Deploy Manager: Jannik A.
- Designer: Angelika S., Jonas H.

- Configuration Manager: Jannik A.
- Implementation:
  - Main-Frontend-Developer: Jonas H.
  - Main-Backend-Developer: Mert G.
  - Note that every team member will participate in the implementation process
- Test Manager: Angelika S.
- Tester: Angelika S., Jannik A., Mert G.
- Documentation will be managed by every team member.

The roles name are based on the [1]Rational Unified Process. The roles may change over time.

Tools and Technologies

Tools and technologies were chosen by popularity and personal experience of the developers.

Agile Project Management Tool:

- YouTrack (you can find our issue list [2]here)

Version administration:

- GitHub
  - Git
  - Markup

Application development IDE:

- IntelliJ IDEA with Android SDK plugin

Blog documentation:

- Wordpress

App platform:

- Android

Build system:

- Maven or Gradle ([3]Evaluation has to be done)

Server and database:

- Apache
- MySQL ([4]Evaluation has to be done)

Continuous Integration and Testing:

- TravisCI

1. `https://www.ibm.com/developerworks/rational/library/apr05/crain/`
2. `https://learnityourselfdhbw.myjetbrains.com/youtrack/issues`
3. `https://learnityourselfdhbw.myjetbrains.com/youtrack/issue/LEARNIT-10`
4. `https://learnityourselfdhbw.myjetbrains.com/youtrack/issue/LEARNIT-12`

---

commonplayground (2018-10-14 11:49:24)
Hello there! In my opinion you've shared your roles well. I think this will work out. You have also given some thought to the different technologies you want to use. I would still like to know which of the tools and technologies you would like to use for which area of your app or for which part of the developing process. One last thing: You should also link to your current issue list and shortly explain how you classify the issues by phase and workflow. Thanks a lot for all the information. Keep us updated :) Sincerely, Celina@CommonPlayground

learnityourselfdhbw (2018-10-17 08:49:24)
Hello, thank you for the information. We will change it. :-) Sincerely, Angelika@learnityourselfdhbw

MNZ-Team (2018-10-15 15:13:52)
Hello guys, It is nice to hear about the roles in your team, but we would like to know more about your tools, technologies and how you want to use them. Maybe you could also tell us more about the general structure of your project. Your MNZ-Team.

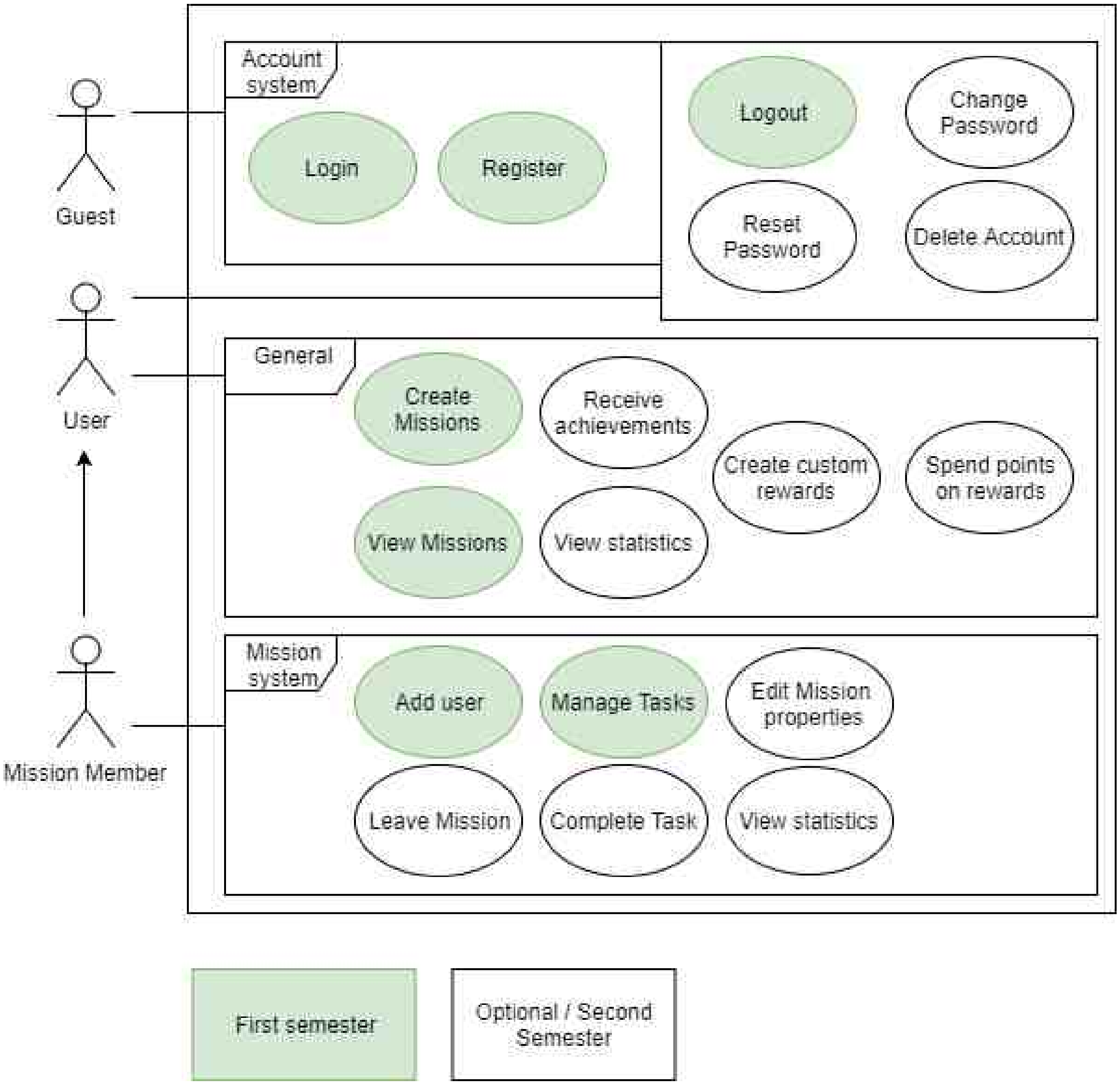
learnityourselfdhbw (2018-10-17 08:52:58)
Hello, thank you for the information. We will change our blogpost. :-) One more advice for you is that you should edit your signature to make visible who of your team wrote the comment (Prof. Berkling mentioned it). Sincerely, Angelika@learnityourselfdhbw

Midterm summary – Learn it yourself (2018-12-11 14:00:21)
[…] Project Roles […]

Week 3 – SRS and UC-Diagram (2018-10-21 21:37)

Hi everyone,

this week we finished the first version of the [1]Software-Requirements-Specification and created our [2]Use-Case-Diagram.
We will keep it updated on our GitHub repository.



Kind regards,

Your Learn it yourself-Team

1. `https://github.com/Mert-Guenduez/learnityourself/blob/master/SRS.md`
2. `https://github.com/Mert-Guenduez/learnityourself/blob/master/UseCases.png`

---

deminder18 (2018-10-22 09:50:09)
Hey guys, I like your use case diagram, it is clear and comprehensible. I have some suggestions for your SRS though. First of all there is still a placeholder in "2. Overall Description". Then for the conventions, do you have some set up for your Project? in "3.5 Supportability" you only mention common Java conventions. While this is good of course, it is probably a good idea to set up some conventions for your project. These could include conventions for git commits, git version control in general or your proejct structure in java (for example do you create few classes with many methods or a lot of classes with only few methods to support modularity). Especially the git conventions can ensure that everyone on the team knows what each commit really does and not just the original commiter. Overall, your SRS seems complete and well thought out though, so good Job ;). Sincerely, Your Deminder team.

learnityourselfdhbw (2018-12-11 16:12:32)
Hello DeminderTeam, Thanks for the detailed answer! We removed all placeholders and added more and specified conventions. Sincerely, Mert@learnityourselfdhbw

MNZ-Team (2018-10-22 12:50:50)
Hey guys, Your Project sound very interessting. Your use case diagram is very descriptive. You wrote in your SRS that a user can log in with a password and username. I would like to know how you want to protect these informations. Otherwise, your SRS is very thoughtful. Sincerely, Your MNZ team.

learnityourselfdhbw (2018-12-11 16:17:40)
Hello MNZ-Team, we protect this information by using a hashing algorithm (with multiple iterations), while also using a salt to hash (the salt string is saved in the database). Of course, the communication between Frontend and Backend is protected by a certificate we created. This means that even we, the developers, have no way of knowing the passwords of our users. Sincerely, Mert@learnityourselfdhbw

DHBWieWarsEssen (2018-10-23 08:52:16)
Hey guys, Your structure seems clear and I could understand it pretty easily. But what confused me at first was, that one box of the user overlaps with the guest. Is it intentional to show a special function? If so it's unclear what it means. Other than that SRS seems clear and written well. I had no issues with the description of yours. Sincerely, Hasan@DHBWieWarsEssen

learnityourselfdhbw (2018-12-11 16:21:04)
Hello DHBWieWarsEssenTeam, I can understand this can appear confusing at first, but this was made intentionally by us to show that these UseCases are also related to the "Account Systems", but do not apply for a guest. Sincerely, Mert@learnityourselfdhbw

Midterm summary – Learn it yourself (2018-12-11 14:00:22)
[...] SRS and UC Diagram [...]

Week 4 - Use Cases (2018-10-28 12:44)

After we have presented our Software Requirements Specification last week, we took it a step further and thought about our use cases. In order to update you on what functionalities our App will be offering in the user's perspective, we specified two use cases in detail. For each use case we created a diagram and a mock-up.

The first use case is about the process of creating (and specifying) a mission, you can find an overview [1]here.

The second use case is about managing tasks within a mission, you can view it [2]here.

1. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/CreateMission/UC_CreateMission.md
2. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/ManageTasks/UC_ManageTasks.md

PerfectTimeCrew (2018-10-29 15:30:20)
Hi Guys, thank you for giving us more information about the current status of your project. Unfortunately your UML-Diagrams for creating missions and managing tasks are a bit confusing because they are too detailed. In the Diagram for creating missions, you drew direct lines from a fork to a join. We are not sure if this is the correct way to draw such UML-Diagrams. Maybe you could go through it again. Did you already created any mock-ups for your use cases? We couldn't find them in your Git repository. Anyway, we are looking forward to your next post, so keep up the good work. Your Perfect Time Crew

learnityourselfdhbw (2018-12-11 16:25:18)
Hello PerfectTimeCrew, thanks for the criticism! We worked over our diagram a little. We also created mockups, you can find them on our GitHub repository (here is an example: https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC /Login/Login _Mock.png) Sincerely, Mert@learnityourselfdhbw

MNZ Team (2018-10-30 15:07:45)
Hi, We tried to review your new UML-Diagram, but the github-Link is broken. If you fix the link I will review your Diagram. Greetings MNZ-Team

MNZ Team (2018-10-31 10:01:54)
After you fixed the links I am back for a full review. The Flow-Diagram was easy to read and the Mock-Up well structured. Your app looks very intuitive. On the first look your Manage Tasks Flow of Events was very confusing. But after reading the details it was easy to understand. See you soon MNZ-Team

learnityourselfdhbw (2018-12-11 16:26:45)
Hello MNZ-Team, I am glad that you liked our work. Thank you very much for the feedback! Sincerely, Mert@learnityourselfdhbw

Midterm summary – Learn it yourself (2018-12-11 14:00:24)
[...] Use Cases [...]

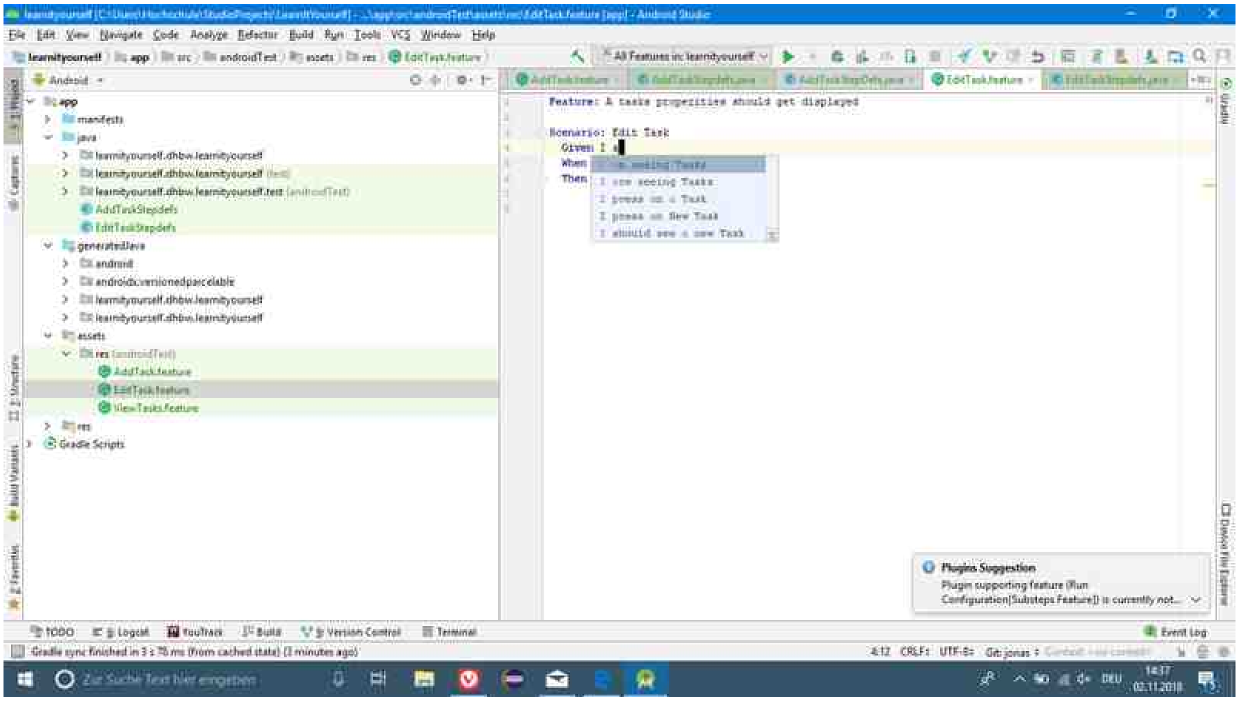
1.2 November

Week 5 - Feature Files (2018-11-04 18:41)

After setting up the project files for the Frontend Application it was time to create the first basic Feature Files for testing with Cucumber.

The files for the Use Case [1]Create Mission as well as [2]Manage Tasks each contain a few Scenarios the App will be able to fullfill. Additionally the Step definitions for both [3]Create Mission and [4]Manage Tasks were generated using the Cucumber Plugin for Android Studio.

Both Highlighted Terms as well as the Autocomplete feature of the IDE were used as shown in the following screenshot:



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
learnityourself app src androidTest assets res EditTask.feature
All Features in: learnityourself
Android
app
manifests
java
learnityourself.dhbw.learnityourself
learnityourself.dhbw.learnityourself (test)
learnityourself.dhbw.learnityourself.test (androidTest)
AddTaskStepdefs
EditTaskStepdefs
generatedJava
android
androidx.versionedparcelable
learnityourself.dhbw.learnityourself
learnityourself.dhbw.learnityourself
assets
res (androidTest)
AddTask.feature
EditTask.feature
ViewTasks.feature
res
Gradle Scripts
AddTask.feature
AddTaskStepdefs.java
AddTaskStepDefs.java
EditTask.feature
EditTaskStepdefs.java
Plugins Suggestion
Plugin supporting feature (Run Configuration[Substeps Feature]) is currently not...
TODO
Logcat
YouTrack
Build
Version Control
Terminal
Event Log
Gradle sync finished in 3 s 76 ms (from cached state) (3 minutes ago)
4:12 CRLF UTF-8 Git: jonas
Zur Suche Text hier eingeben
DEU
14:37
02.11.2018

```
Feature: A tasks properitiss should get displayed

Scenario: Edit Task
  Given: I a
  When   I am seeing Tasks
  Then   I am seeing Tasks
         I press on a Task
         I press on New Task
         I should see a new Task
```

1. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/CreateMission.feature
2. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/ManageTasks.feature
3. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/java/CreateMissionStepdefs.java
4. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/java/ManageTasksStepdefs.java

---

flashcardcommunity (2018-11-06 17:44:41)
Hi guys, I can say, you did a very good job again. Especially you wrote several scenarios for the "Create Mission" UC and for both UCs you gave us an view to the created .java stub files. Your screenshot shows the syntax highlighting and the autocompleteion of your IDE. This picture seems to be downscaled a lot, maybe a link to the original image would have been nice ecpecially for people who have worse eyes. Just one more little thing: Look at "...A new Mission ... and d'i'splayed" So keep on your good work! Regards Rainer@FlashCardCommunity

learnityourselfdhbw (2018-11-21 10:27:09)
Hi, thanks for your feedback. We will change it. Regards Angelika@learnITyourself

MNZ Team (2018-11-06 23:35:58)
Hey learnityourselfdhbw-Team, you had many Scenarios and they were described very detailed. What we very much were the .java filed where you show us how you use your feature files. Like Rainer already mentioned, the picture is a little bit small. Maybe it could get bigger after clicking on it. See you soon your MNZ-Team

learnityourselfdhbw (2018-12-11 16:28:06)
Hello MNZ-Team, thanks for taking the time! We tried to improve the picture a little. Sincerely, Mert@learnityourselfdhbw

Find your Farm-Team (2018-11-07 11:29:45)
Hello, good work on your feature files. You made it clear, what happens in which scenario. The syntax highlighting shows that you're using all the help your IDE offers. Greetings, Natalie

learnityourselfdhbw (2018-12-11 16:29:11)
Hello FindyourFarmTeam, thanks a lot for the feedback! Sincerely, Mert@learnityourselfdhbw

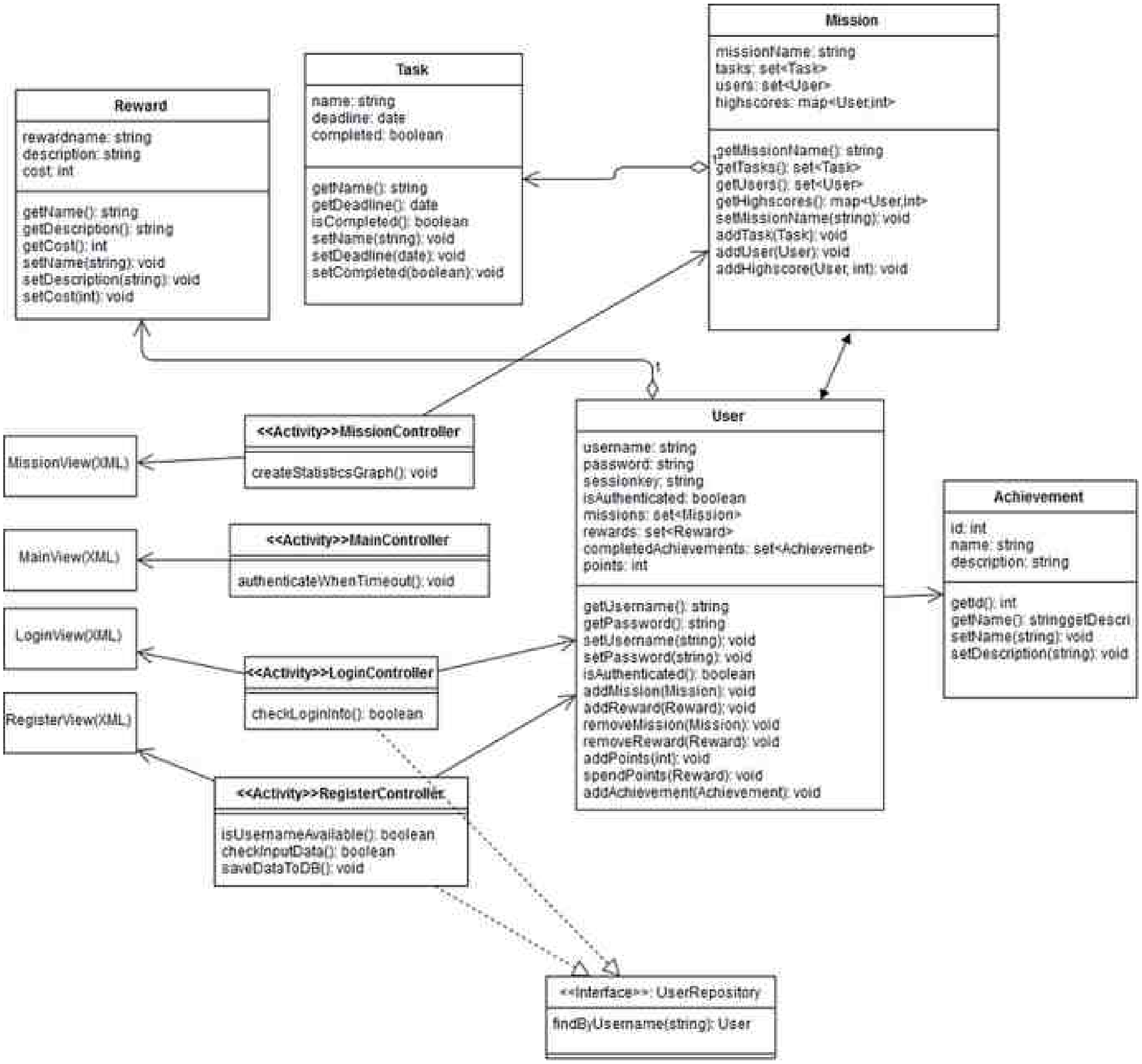
Midterm summary – Learn it yourself (2018-12-11 14:00:26)
[...] Feature Files [...]

Week 6 - Class Diagram / DB Scheme (2018-11-11 20:14)

This week we created our database scheme and our class diagram.

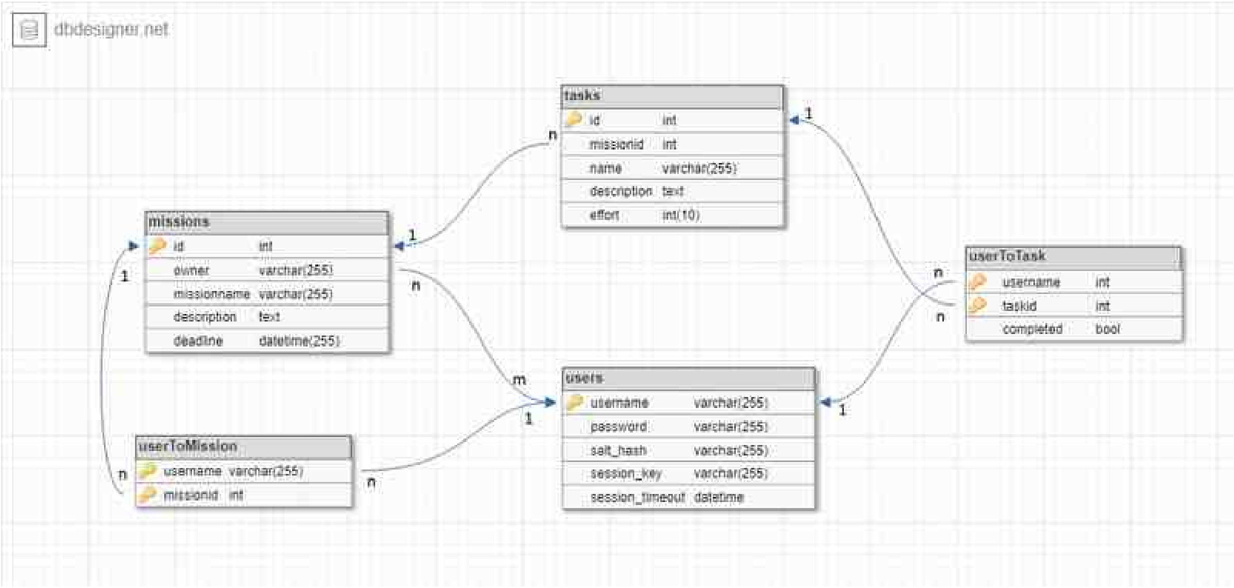
Class Diagram

We generated our [1]Class Diagram due to our previous considerations which you can see in our SRS, UCs and Mockups. Our Class Diagram is generated by hand and we might decide to change it up a little, also of course depending on your feedback!



Database Scheme

In dependence on our Class Diagram we created our [2]Database Scheme with the web application [3]DbDesigner. Just as our Class Diagram we might also change our Database Scheme.



1. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/ClassDiagram.png`
2. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/dbScheme.png`
3. `https://www.dbdesigner.net/`

---

perfecttimese (2018-11-13 11:20:20)
Hey Guys, amazing work with your diagrams. They look great. We can clearly see you application structure there. One little thing came to our attention: You did not specify numbers to your table relations for your database. You might want to add that. Apart from that everything looks good. So keep up your great work. Your friends from Perfect Time

learnityourselfdhbw (2018-12-11 23:01:27)
Hello PerfectTimeTeam, thanks for your feedback. We have now added the table relations. Sincerely, Mert@learnityourselfdhbw

Team DigiWill (2018-11-14 09:39:14)
Hello, nice to see your progress! One thing about the user is that it could be a good idea to have a unique id as the primary key. Otherwise there could be some problems I think.. In the class diagram I do not see the session key used anywhere. So I wonder how you want to use it later. Other than that I really like your diagrams as they are very clear and good to understand. Kind regards, Morten Harter

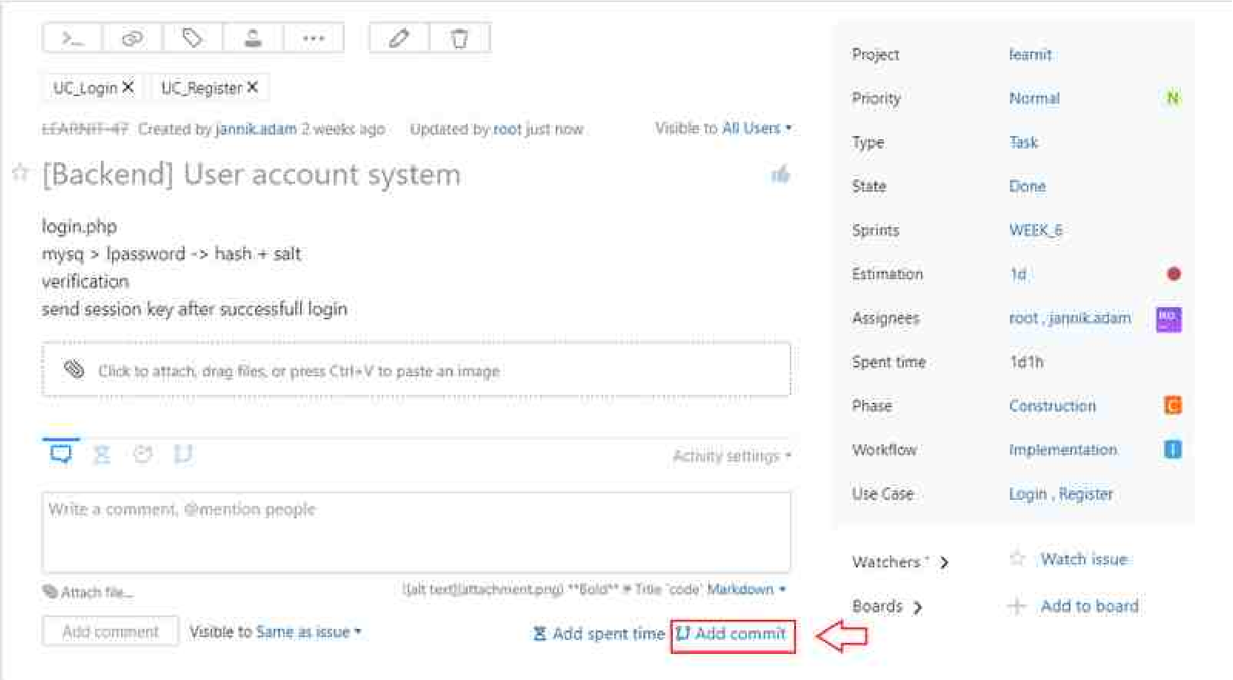
learnityourselfdhbw (2018-11-14 09:48:04)
Hello Morten, Thank you for your for taking time to inspect our diagrams, we like your answer. We decided that all usernames have to be distinct (this is checked when somebody tries to register a username that already exists), therefore we do not really need a user-id, do we? You can find the session key as an attribute of the user class. Sincerely, Mert@learnityourselfdhbw

Midterm summary – Learn it yourself (2018-12-11 14:00:28)
[...] Class & Database Diagram [...]

Week 7 - Scrum (2018-11-17 15:09)

This week we will provide you with a concise insight on our Project Management process. Since we intended to have agile development from the very beginning of our project, we are using Scrum with YouTrack as our Project Management tool to record our issues and reports.
First we will show you some proof that we actually integrated GitHub and AndroidStudio to be able to make issues more detailed and precise:

GitHub Integration:



UC_Login ✕ UC_Register ✕

~~LEARNIT-47~~ Created by jannik.adam 2 weeks ago Updated by root just now Visible to All Users ▾

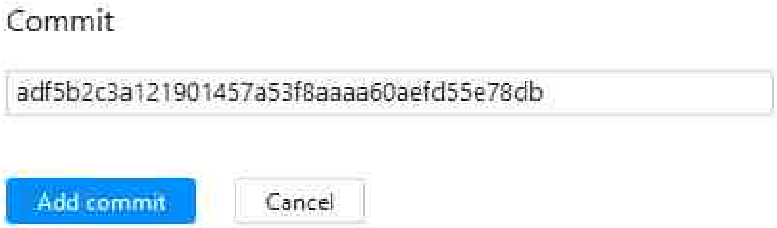
[Backend] User account system

login.php
mysq > lpassword -> hash + salt
verification
send session key after successfull login

Project	learnit
Priority	Normal
Type	Task
State	Done
Sprints	WEEK_6
Estimation	1d
Assignees	root , jannik.adam
Spent time	1d1h
Phase	Construction
Workflow	Implementation
Use Case	Login , Register

Click to attach, drag files, or press Ctrl+V to paste an image
Activity settings ▾
Write a comment, @mention people
Attach file...
![alt text](attachment.png) **Bold** # Title `code` Markdown ▾
Add comment
Visible to Same as issue ▾
Add spent time
Add commit
Watchers ›
Watch issue
Boards ›
Add to board
N
C
I

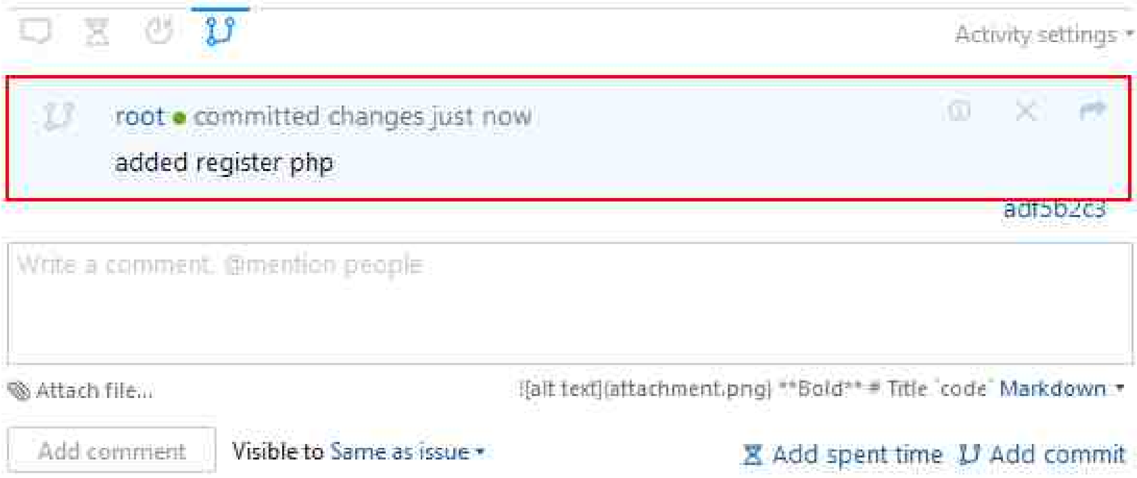
Here you can see an example of one of the issues we created. After we have worked on an issue and committed files to GitHub, we can add the commit by entering the commit hash.



Commit

adf5b2c3a121901457a53f8aaaa60aefd55e78db

This way we can also keep track of the progress by directly viewing our commits in YouTrack.



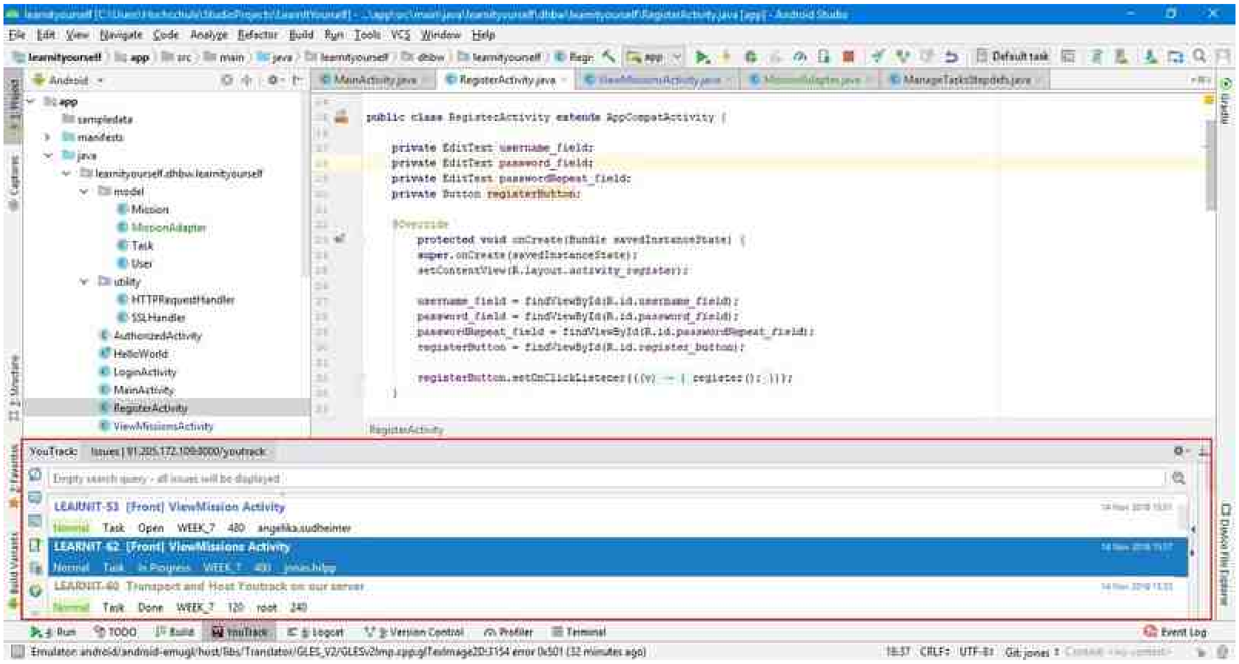
root committed changes just now

added register php

adf5b2c3

Activity settings
Write a comment, @mention people
Attach file...
![alt text](attachment.png) **Bold** # Title `code` Markdown
Add comment
Visible to Same as issue
Add spent time
Add commit

AndroidStudio Integration:



```
public class RegisterActivity extends AppCompatActivity {

    private EditText username_field;
    private EditText password_field;
    private EditText passwordRepeat_field;
    private Button registerButton;

    @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        username_field = findViewById(R.id.username_field);
        password_field = findViewById(R.id.password_field);
        passwordRepeat_field = findViewById(R.id.passwordRepeat_field);
        registerButton = findViewById(R.id.register_button);

        registerButton.setOnClickListener((v) -> { register(); });
    }
```

YouTrack: Issues | 91.205.172.109:8000/youtrack

Empty search query - all issues will be displayed

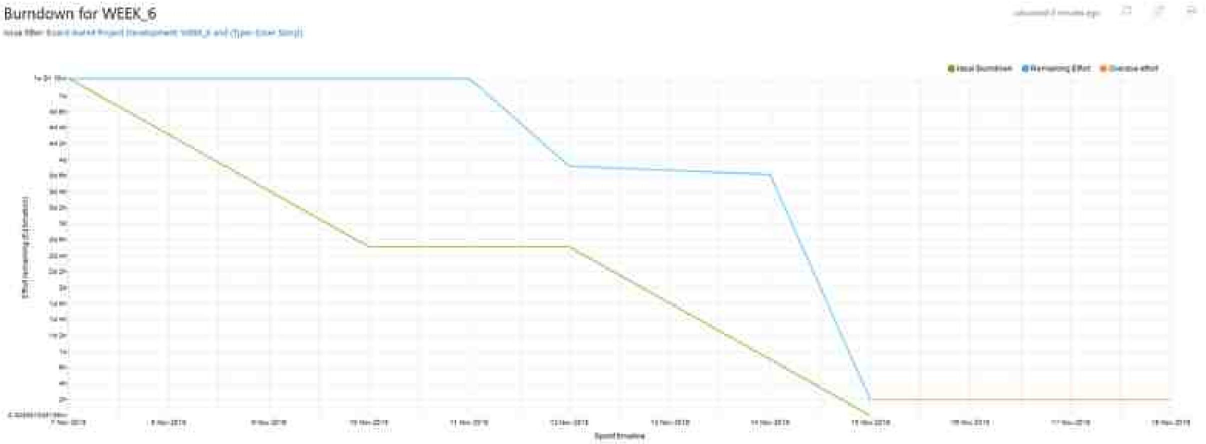
LEARNIT-53 [Front] ViewMission Activity
Normal Task Open WEEK_7 480 angelika.sudheimer

LEARNIT-62 [Front] ViewMissions Activity
Normal Task In Progress WEEK_7 480 jonas.hilpp

LEARNIT-60 Transport and Host Youtrack on our server
Normal Task Done WEEK_7 120 root 240

Emulator: android/android-emugl/host/libs/Translator/GLES_V2/GLESv2Imp.cpp:glTexImage2D:3154 error 0x501 (32 minutes ago)

Lastly, we also want to show you one of our Sprint-Burndown Charts:



As you can see, the Remaining Effort line is constantly above the Ideal Burndown. You will see this in pretty much all of our Sprints and it has several reasons. Firstly, we tend to work way more on the weekends, when the week is already almost over, rather than within the week. On top of that, most of the time our issues are in the "To be verified"-category, since we usually review our work at the very end of the week and also wait on your peer review feedbacks. These are the reasons why there is such a rapid fall of the remaining effort at the end. Lastly, one issue has been left on the "To be verified"-category that week, counting as Overdue effort, because we couldn't decide yet wether we were satisfied with the result.

All in all we will try to work on this by reviewing our work earlier, thus being able to set issues quicker into the "done"-category.

Stay tuned for more updates in the future!

Felix Hausberger (2018-11-18 11:19:20)
Hey learnityourselfdhbw, I really appreciate that you too use YouTrack as your project management tool. Don't worry, I think it is quite common in software projects that a burndown chart will never match with the ideal curve. Remember to add a reference to the use case, workflow and phase according to RUP to each task. ou can do this by defining custom fields. Just go to projects > [projectname] > Edit Project > fields > add fields to project. This is especially important for the cumulative flow chart (by workflow) and other charts to track your project progress. I'm looking foward for seeing your next blog post!

Felix Hausberger (2018-11-18 11:20:44)
And please make your comment section editable ;)

learnityourselfdhbw (2018-11-20 15:42:52)
Hello Felix, thank you for your comment! I have now added the fields you suggested. The Screenshot will be updated later on. Sincerely, Mert@learnityourselfdhbw

perfecttimese (2018-11-18 19:23:32)
Hey guys, you did a great work with your PM tools and the different integrations. Well done! Concerning your burndown chart: You mentioned that you work way more at the weekends than within the week. Maybe you can track this also in YouTrack. The default working time in YouTrack is from Monday to Friday. Because of that your ideal burndown chart isn't straight. You can

edit your working time by going into the main settings. There you click on "Time tracking" or "Zeiterfassung". Anyway, keep up the good work. We are looking forward to your next steps. Your PerfectTimeCrew

learnityourselfdhbw (2018-11-20 15:51:13)
Hello PerfectTimeTeam, thanks a lot, we really appreciate your feedback! Your suggestion was really helpful, we changed it so Saturday and Sunday are also workdays now. Kind Regards, Mert@learnityourselfdhbw

Midterm summary – Learn it yourself (2018-12-11 14:00:30)
[...] Scrum [...]

Week 8 - Retrospective (2018-11-25 13:46)

For this weeks Blogentry we will take a look at the Retrospective we did, taking the technique introduced by the agile coach that visited us this week into account.

First of all, we really liked the Retrospective and think that it will definitely improve our abilities as a team. The fact that we were not only supposed to reflect on ourselves in our own team, but also collected ideas together with the other teams as a whole group provided us different perspectives on how to view the problems we face. Additionally it also made the whole process more fun and interactive, which is why we enjoyed the coach’s style of teaching us how a retrospective may look like.

The basic steps were collecting points that worked well, then the problematic ones and finally working out ways to improve ourselves.

First off, the things that went well were:

- Communication
- Making decisions
- Design

Points that need improvement are:

- Motivation to finish work on time and avoid impeding the work of others
- Planning our steps to make sure the different parts of development are coordinated well
- Sticking to Code and Git Conventions we agreed on

From all this we derived three goals for the future:

- Weeklys: While we did talk about the tasks for each week, it didn’t have any kind of formality to it, therefore we are seeking to make our plans for each week clearer and record them with the help of Youtrack.

- Code Patterns: To avoid confusion and make our code easier to understand for each developer, we are looking forward to using a more consistent structure throughout

- Tests: To make sure the code we commit works, as well as catching up with the missing Tests we want to make sure they are written with equal vigor to the actual code

---

CommonPlayground (2018-11-26 20:50:38)
Hi folks, thanks for sharing your thoughts with us. The grading criteria requires some reflection on how the Retro was for you. Just a short opinion maybe? Did you find it useful? Do you think the time spent on it was adequate? Did you like the coaches style? Just anything that came to your mind really. Everything else you've covered! Keep it up! Inga@CommonPlayground

learnityourselfdhbw (2018-11-27 09:39:56)
Thanks for your Feedback, we will revise the Blogentry and take your advice into account. Jonas@LearnItYourslef

CodeCrunch (2018-11-28 00:44:56)
Hey learnityourself team, we enjoyed reading your update. As far as we can see the post meets all the requirements listed in the grading criteria for this week. You got three great goals for the future to improve your work but we can not help but notice that you did not mention on how you want to approach the lack of motivation mentioned in your points to improve. Maybe you can tell us a little bit more about your plans in this regard. Cheers Kai@CodeCrunch

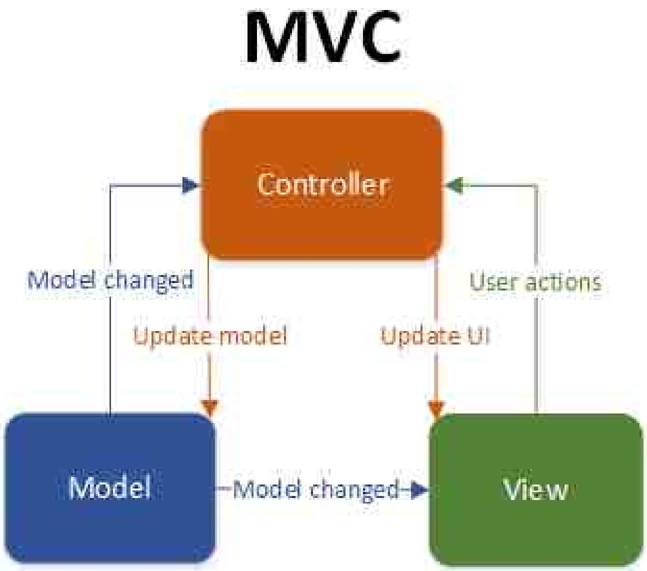
learnityourselfdhbw (2018-12-11 23:03:54)
Hello CodeCrunch-Team, thank you for your review! We have specified our plans a little more now ;) Sincerely, Mert@learnityourself

Midterm summary – Learn it yourself (2018-12-11 14:00:32)
[...] Retrospective [...]

1.3 December

Week 9 - MVC (2018-12-02 20:04)

This weeks homework was to properly define the software architecture of our application. The basic pattern corresponds to MVC as shown in the following image



However, contrary to this depiction, our Controllers cannot directly make changes to the View. This is dictated by the nature of the App since the Activities, which represent the View, get started first and instantiate the corresponding controller.

To make the clear distinction between View and Controller possible in the diagram, we had to edit our Code by extracting logic from the Activity classes and put them in their respective Control Classes.

The exact description can be seen in our new [1]Software Architecture Document.

1. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/SRA.md`

flashcardcommunity (2018-12-03 23:53:34)
Hi guys, it is interesting to see how your project has been developed during the last weeks. Your SAD seems to be completed, you can see the Use-Cases and those you want to implement in this semester, as well as the other diagrams which give us a detailed view in your program logic and your database. You also added a table of contents and the links to the Use-Case diagrams what makes it easy to navigate to all of the information. We are really looking forward to the presentation of your application. So, keep on your good work! Regards, Rainer@FlashCardCommunity

learnityourselfdhbw (2018-12-11 23:05:02)
Hello Rainer, We are glad you enjoyed our work! Thanks for your review. Sincerely, Mert@learnityourselfdhbw

MNZ Team (2018-12-05 09:35:36)
Hi guys, At first, we want to say that it is nice to see what you have done since you started your Project. Furthermore, we are really interested to see your web application live and to test it. Your SAD seems completed and very detailed. It is nice that you have a table of contents, so we are able to see everything we want very quickly. Additionally, we want to tell you that your use case view is not a valid UML-diagram. Keep us up to date. Best, Your MNZ-Team

Midterm summary – Learn it yourself (2018-12-11 14:00:33)
[…] MVC […]

Midterm summary (2018-12-06 12:10)

Hello everyone,

this week we want to summarize everything we have done and achieved so far. In the following we will provide you links to all relevant parts of our project as well as all of our previous blog posts.

Use Cases:

- [1]UC Login
- [2]UC Register
- [3]UC CreateMission
- [4]UC ViewMissions
- [5]UC EditMissionMember
- [6]UC ManageTask
- [7]UC AddUser

Feature Files:

- [8]Feature: Login
- [9]Feature: Register
- [10]Feature: CreateMission
- [11]Feature: ViewMissions
- [12]Feature: EditMissionMember
- [13]Feature: ManageTask
- [14]Feature: AddUser

GitHub:

- [15]GitHub (Repository)

Software Specifications:

- [16]SRS
- [17]SAD
- [18]Overall Use Case Diagram

Youtrack:

- [19]Agile Board
- [20]Issues
- [21]Time Report (The user "root" is Mert G.)
- [22]Burndown
- [23]Gantt Chart
- [24]Cumulative Flow

Blog Posts:

1. [25]Project Vision
2. [26]Project Roles
3. [27]SRS and UC Diagram
4. [28]Use Cases
5. [29]Feature Files
6. [30]Class & Database Diagram
7. [31]Scrum
8. [32]Retrospective
9. [33]MVC

Demo:

- [34]Learnityourself Online Demo
- [35]Learnityourself APK Download

1. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/Login/UC_Login.md
2. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/Register/UC_Register.md
3. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/CreateMission/UC_CreateMission.md
4. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/ViewMissions/UC_ViewMissions.md
5. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/EditMissionMember/UC_EditMissionMember.md
6. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/ManageTasks/UC_ManageTasks.md
7. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC/AddUser/UC_AddUser.md
8. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/Login.feature
9. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/Register.feature
10. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/CreateMission.feature
11. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/ViewMissions.feature
12. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/EditMissionMember.feature
13. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/ManageTasks.feature
14. https://github.com/Mert-Guenduez/learnityourself/blob/master/app/src/androidTest/assets/res/AddUser.feature
15. https://github.com/Mert-Guenduez/learnityourself
16. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/SRS/SRS.md
17. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/SRA.md
18. https://raw.githubusercontent.com/Mert-Guenduez/learnityourself/master/Documentation/UC/UseCases_Overview.png
19. http://91.205.172.109:8000/youtrack/agiles/
20. http://91.205.172.109:8000/youtrack/issues
21. http://91.205.172.109:8000/youtrack/reports/time/134-4
22. http://91.205.172.109:8000/youtrack/reports/burndown/116-1
23. http://91.205.172.109:8000/youtrack/reports/gantt/137-0?view=actual
24. http://91.205.172.109:8000/youtrack/reports/cumulativeFlow/135-1
25. https://learnityourselfdhbw.wordpress.com/2018/10/05/erster-blogbeitrag/
26. https://learnityourselfdhbw.wordpress.com/2018/10/10/our-project-roles/
27. https://learnityourselfdhbw.wordpress.com/2018/10/21/week-3-srs-and-uc-diagram/
28. https://learnityourselfdhbw.wordpress.com/2018/10/28/week-3-use-cases/
29. https://learnityourselfdhbw.wordpress.com/2018/11/04/week-5-feature-files/
30. https://learnityourselfdhbw.wordpress.com/2018/11/11/week-6-class-diagram-ds-schema/
31. https://learnityourselfdhbw.wordpress.com/2018/11/17/week-7-scrum/
32. https://learnityourselfdhbw.wordpress.com/2018/11/25/week-8-retrospective/
33. https://learnityourselfdhbw.wordpress.com/2018/12/02/week-9-mvc/
34. https://appetize.io/app/x27f6h9t6d94n61jx6egvdahdw?device=nexus5&scale=75&orientation=portrait&osVersion=7.1

35. `https://91.205.172.109/download/learnityourself.apk`

2. 2019

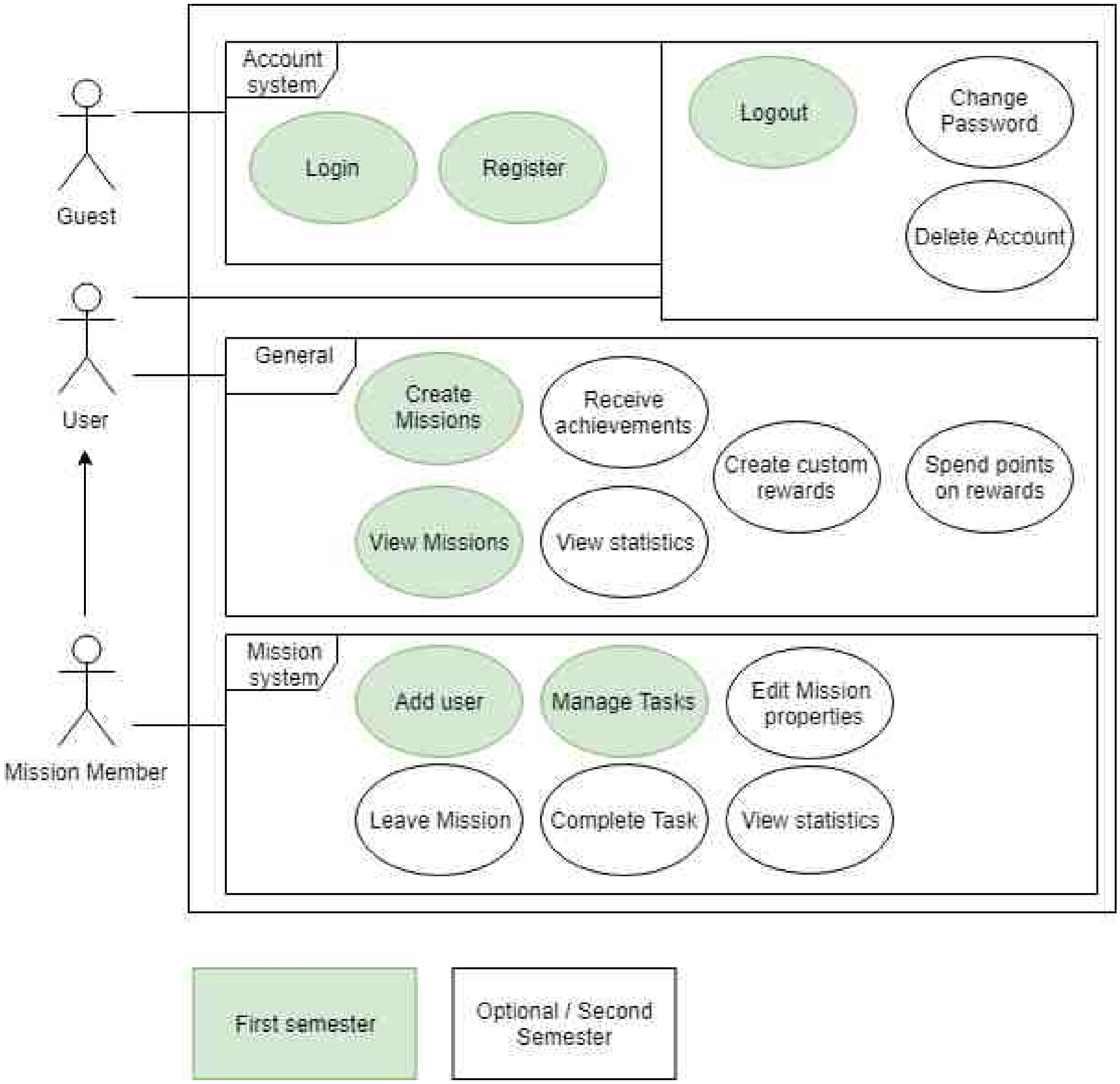
2.1 April

Confirmation of our Scope (2019-04-10 15:54)

Hello everyone,

after a three month break working at our companies we are now picking our project back up. This week we will have a look at our scopes we defined and examine whether our goals are realistic or not. First of all we are happy to inform you that our team composition stays the same and that we do not plan to deviate a lot from our original aims.

It seems that we defined our scope pretty accurately, which is why our Overall Use Case diagram stays almost the same. We have removed one Use Case, namely Reset Password, since we have no way of authenticating the User except for the password itself:



You can also find this Use Case Diagram in our [1]SRS. The specifications including UML diagrams and Mock-ups can be found in our [2]GitHub Repository.

Apart from focusing on the Use Cases, we started to continue developing on our project and re-read our code to find our way back into our everyday schedule. Both Frontend and Backend implementation of the Use Cases above are currently in progress.

Stay tuned for more updates!

Sincerly,

the learnityourself team

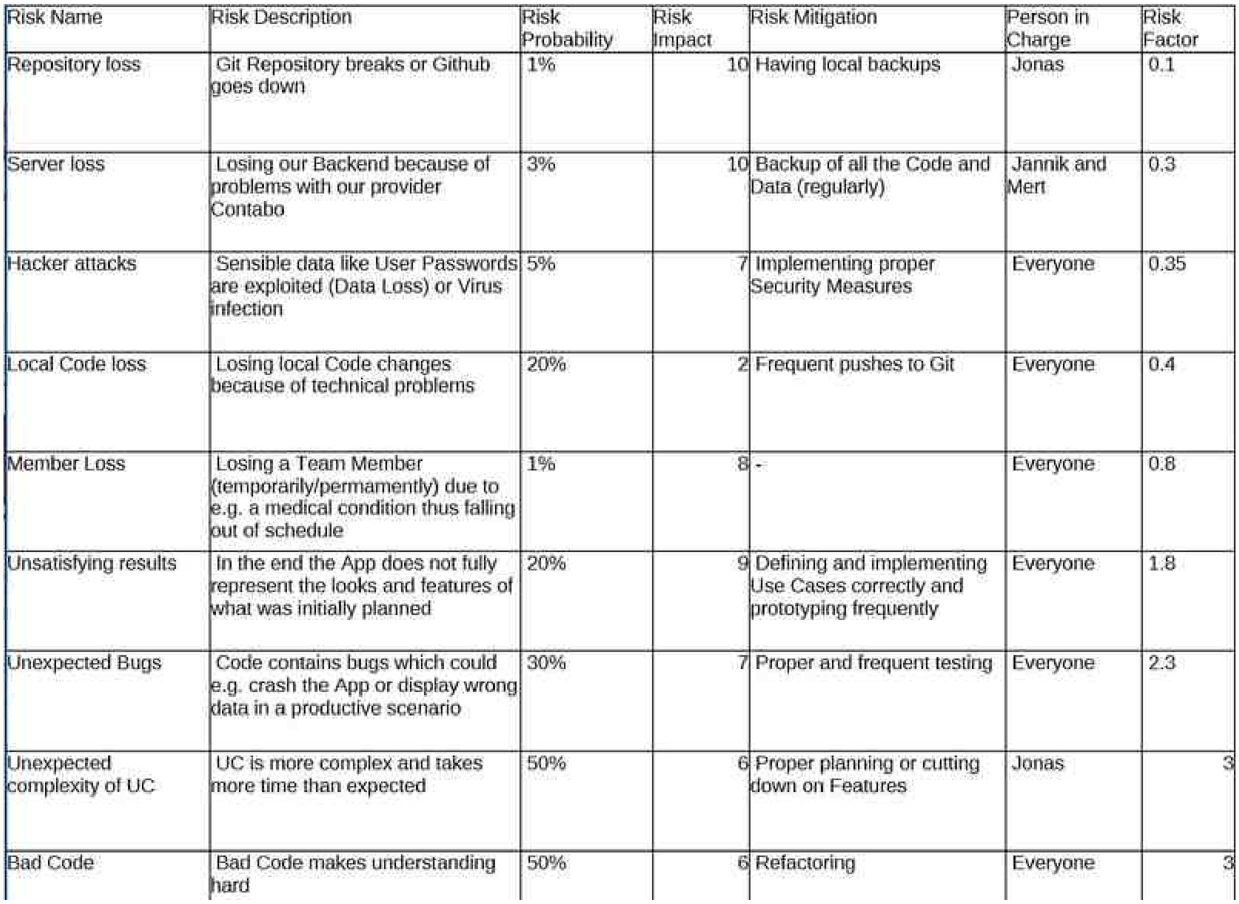
1. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/SRS/SRS.md
2. https://github.com/Mert-Guenduez/learnityourself/tree/master/Documentation/UC

Risk Assessment (2019-04-15 20:14)

Hello everyone,

this week we had a look at all the possible risks that are facing us and our project. Every project has its risks, some more potent than others, which is why it is quite sensible to think of every possible risk scenario and consider which preventive measures should be taken to avoid as many risks as possible.

We have created a list of all possible risks including details like probability and potential damage, which you can also find [1]here:



Risk Name	Risk Description	Risk Probability	Risk Impact	Risk Mitigation	Person in Charge	Risk Factor
Repository loss	Git Repository breaks or Github goes down	1%	10	Having local backups	Jonas	0.1
Server loss	Losing our Backend because of problems with our provider Contabo	3%	10	Backup of all the Code and Data (regularly)	Jannik and Mert	0.3
Hacker attacks	Sensible data like User Passwords are exploited (Data Loss) or Virus infection	5%	7	Implementing proper Security Measures	Everyone	0.35
Local Code loss	Losing local Code changes because of technical problems	20%	2	Frequent pushes to Git	Everyone	0.4
Member Loss	Losing a Team Member (temporarily/permamently) due to e.g. a medical condition thus falling out of schedule	1%	8	-	Everyone	0.8
Unsatisfying results	In the end the App does not fully represent the looks and features of what was initially planned	20%	9	Defining and implementing Use Cases correctly and prototyping frequently	Everyone	1.8
Unexpected Bugs	Code contains bugs which could e.g. crash the App or display wrong data in a productive scenario	30%	7	Proper and frequent testing	Everyone	2.3
Unexpected complexity of UC	UC is more complex and takes more time than expected	50%	6	Proper planning or cutting down on Features	Jonas	3
Bad Code	Bad Code makes understanding hard	50%	6	Refactoring	Everyone	3

Keep in mind that this is a Living Document and therefore the Image might not be up to date.

As you can see, we have covered all kinds of risks reaching from Bad Code to Member Loss. We will try to apply all risk mitigation methods with a special focus on the risks with a high risk factor.

We are glad that so far no risk has set us back in a significant manner. We hope this stays the same for the following weeks in order for us to be able to complete this project successfully!

Additionally we assessed our time spent on individual Use-Cases so far
which you are able to look up [2]here.

Thanks for reading and stay tuned for more updates,

learnityourselfdhbw

1. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/Risk-Chart.csv`
2. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/UC-Time-Chart.csv`

Christian Schweigel (2019-04-16 08:27:05)
Hi there, looks like you had a good start back into the project. Overall it looks good what you have covered this week. I have just a few little things that could be changed: Your chart has no color coding, which makes it harder to read it. I know that it is csv, but maybe a different file format with more styling options would be better. Best regards, Christian@DigiWill

learnityourselfdhbw (2019-04-16 10:52:16)
Hello DigiWill, While it would make sense to use color coding, we chose csv since it is easy for everyone to use since we all contributed to this file. If we used a format like excel it would take longer and we would have to ensure that everyone has excel, which is why we wanted to keep things simple. Sincerely, learnityourselfdhbw

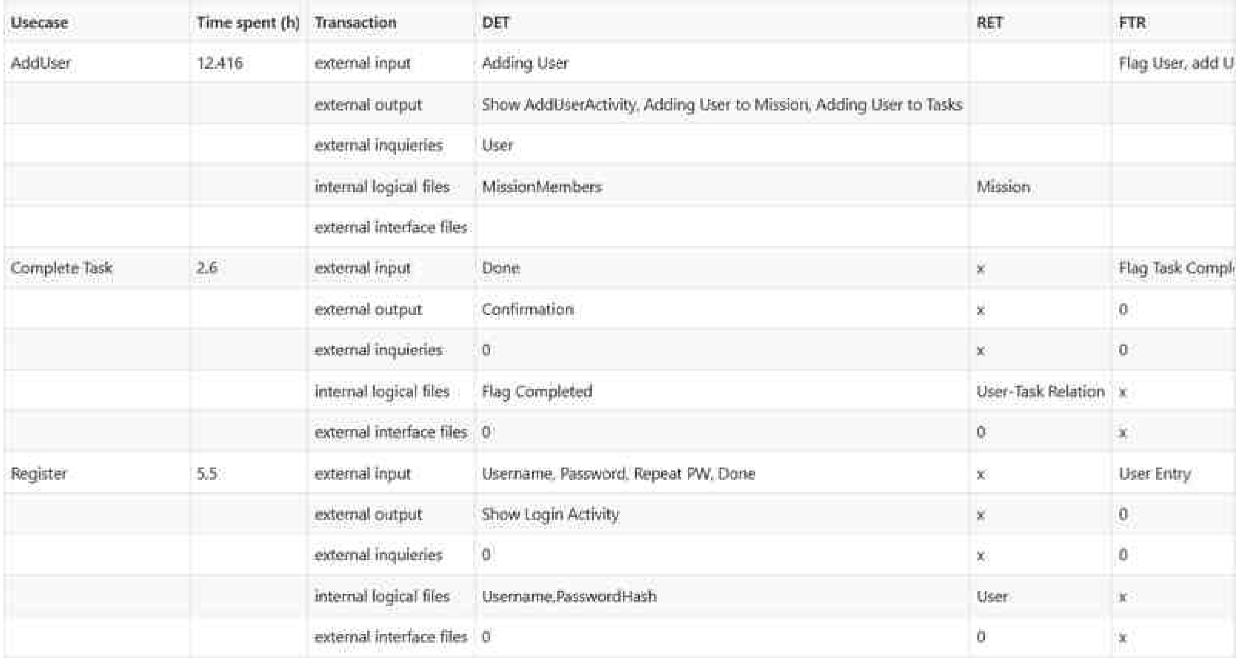
MNZ Team (2019-04-16 11:02:51)
Hey guys, the risks you thought of are precise and valid. Everything is well structured, detailed and still easy to read and understand. Most of the probabilities look reasonable, however, some of them appear to be pretty high. For example the probability for writing "bad code" is at 50 %. Besides that everything looks great - good work this week. I’m also glad to hear that none of these risks have significantly set you back. Keep up the good work. Best regards MNZ-Team

Function Points **(2019-04-22 12:24)**

Hey everyone,

this week we calculated so called "Function Points" for 7 of our Use Cases. Function Points are a way to describe what and how many functions are provided to the user and helps to predict how much effort it will take to implement other Use Cases.

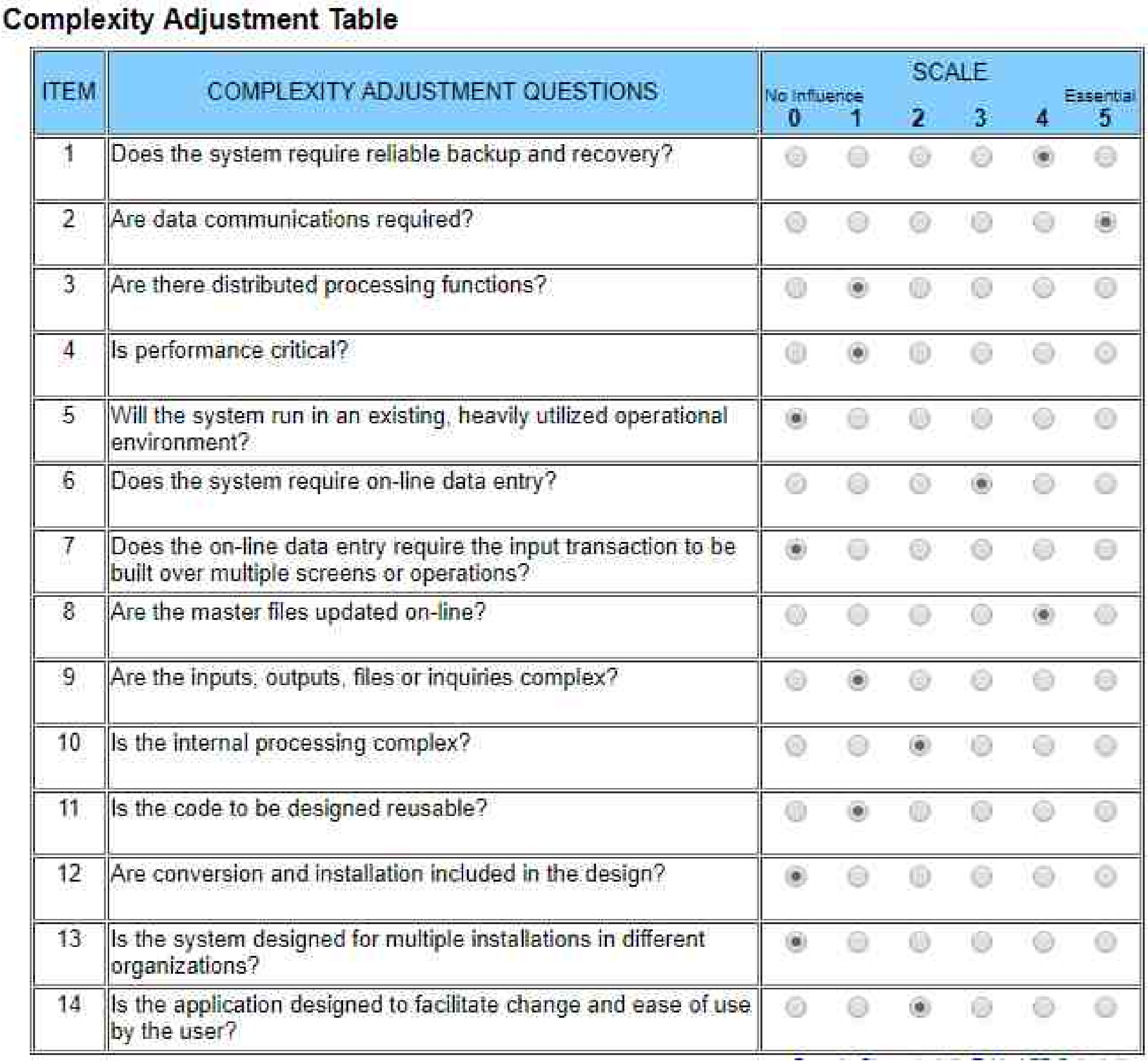
For this, we created a Function Point Table which contains all the detailed information. This includes the complexity of every transaction (e.g. external input/output, internal logical files, ...) of the Use Cases. Have a look at a snippet of that table:



Usecase	Time spent (h)	Transaction	DET	RET	FTR
AddUser	12.416	external input	Adding User		Flag User, add U
		external output	Show AddUserActivity, Adding User to Mission, Adding User to Tasks		
		external inquieries	User		
		internal logical files	MissionMembers	Mission	
		external interface files			
Complete Task	2.6	external input	Done	x	Flag Task Compl
		external output	Confirmation	x	0
		external inquieries	0	x	0
		internal logical files	Flag Completed	User-Task Relation	x
		external interface files	0	0	x
Register	5.5	external input	Username, Password, Repeat PW, Done	x	User Entry
		external output	Show Login Activity	x	0
		external inquieries	0	x	0
		internal logical files	Username,PasswordHash	User	x
		external interface files	0	0	x

The entire table can be found [1]here.

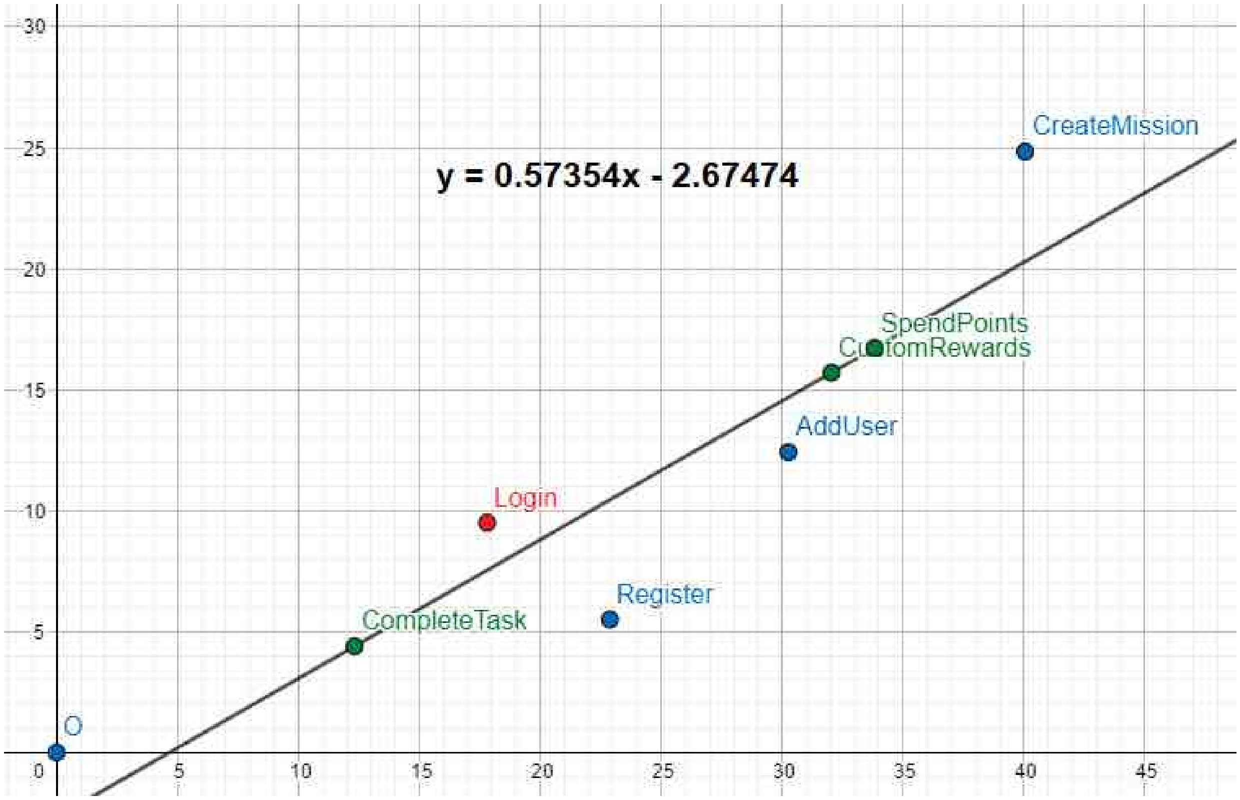
For calculating the FP we used [2]TINY TOOLS with this Complexity Adjustment:



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?	○	○	●	○	○	○

Plotting three points of finished usecases results in the following graph. The stray point (red) can be ignored since it was a task completed at the beginning of the project and therefore needed more time than it would now. Using the remaining three points (blue) to make a linear regression results in the function y=0.57354x-2.67474



From this predictions about the time needed for future Usecases can be made. The FP of CompleteTask, CustomRewards and SpendPoints each can be used to calculate a point (green) on the line.

Using the information from our cummulative Flow Diagram we will be able to make a judgement on whether we will be able to complete all these Tasks within a given Time frame.

Sincerely,

learnityourselfdhbw

1. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/FP-Table.csv`
2. `http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP_Calc.html`

CommonPlayground (2019-04-22 20:05:31)
Hi learnityourselfdhbw :) At first, I want to say that your calculations of function points seem to be reasonable. Well done! After some research, I found out that you already implemented more than just three use cases. Wouldn't be the estimation for the future use cases way more accurate when you add more of your implemented use cases in the calculation? Beside, the grading criteria says that you should do the FP calculation for 5 old use cases and at least 3 new use cases. Furthermore, I am curious to see the calculation for the function points more detailed. Did you use the website Tiny Tools (http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP _Calc.html)? What are the numbers you picked? Just add a screenshot of the complexity adjustment table to your blogpost. And maybe also a screenshot of the domain characteristic table for each use case in the UC specifications would be great. Is it possible to do the line in your diagram just

through the use cases that are important for the estimation? A polygonial line doesn't seem right to me. One last question: After the estimation, are you able to say if you have enough time to finish the remaining use cases before the end of semester? But all in all, you did a great work. Keep up, Celina@CommonPlayground

learnityourselfdhbw (2019-04-23 08:40:46)
Thanks for your feedback! We already had a screenshot of the complexity adjustment table and now added it to the blogpost. The diagram shall be fixed as well, as you pointed out, having a line to a Datapoint thats being ignored is pointless. With regards Jonas@LearnITYourself

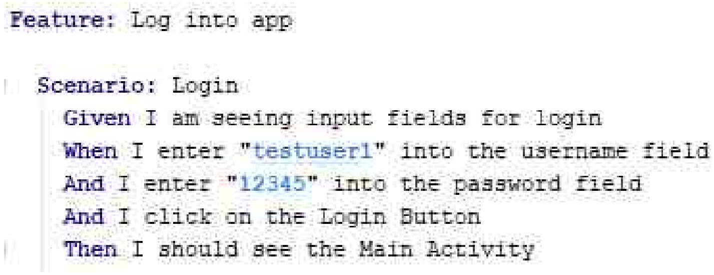
MNZ (2019-04-22 22:38:26)
Hi learnityourselfdhbw, I really liked your work for this week. I would have liked it if you listed all the new use cases you added and the use cases you used for the graph. Also like CommonPlayground already mentioned you should change your graph to a linear graph to estimate the Time you need for your future use cases. Your MNZ-Team

Week 10 - Cucumber Testing (2019-04-25 15:01)

In this weeks blogpost we will deal with Cucumber Testing and Feature Files within our Project.

Cucumber is a Testing Framework using so-called Feature files to describe possible scenarios that will be tested. You are able to find our Feature files within our [1]Git-Repo.

For example, a scenario in the Login Feature File looks like this:



```
Feature: Log into app

  Scenario: Login
    Given I am seeing input fields for login
    When I enter "testuser1" into the username field
    And I enter "12345" into the password field
    And I click on the Login Button
    Then I should see the Main Activity
```

Executing this with Cumumber looks as shown in this [2]Video

1. https://github.com/Mert-Guenduez/learnityourself/tree/master/app/src/androidTest/assets/res
2. https://streamable.com/4xyem

Week 14 – Unit Testing – Learn it yourself (2019-04-29 20:16:26)
[...] this weeks blogpost we will deal with Testing. While there already was a Blogpost about Cucumber Testing where you are able to see a Frontend Test in action, this weeks entry will deal with Unit [...]

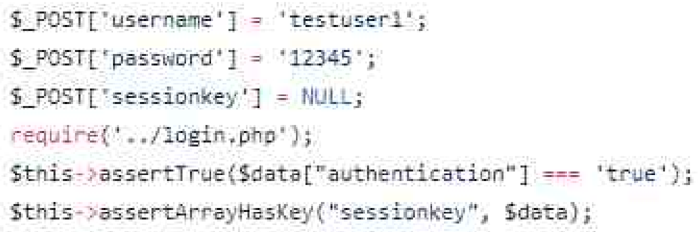
Week 15 - Unit Testing (2019-04-29 20:16)

Hello Everyone,

in this weeks blogpost we will deal with Testing. While there already was a Blogpost about [1]Cucumber Testing where you are able to see a Frontend Test in action, this weeks entry will deal with Unit testing.

Since most of the applications logic is taking place on the server side, we decided to Use phpUnit. You can find our Backend tests [2]here.

For example this Snippet of the Login Test makes sure that a login with correct inputs works:



```
$_POST['username'] = 'testuser1';
$_POST['password'] = '12345';
$_POST['sessionkey'] = NULL;
require('../login.php');
$this->assertTrue($data["authentication"] === 'true');
$this->assertArrayHasKey("sessionkey", $data);
```

All of this will be automated using [3]Jenkins. Which is running on [4]our server.
With Jenkins we were able to provide a Build Badge on our [5]GitHub

With this we covered two of the three kinds of Tests required, all of this will be recorded in our [6]Testplan

1. `https://learnityourselfdhbw.wordpress.com/2019/04/25/week-10-cucumber-testing/`
2. `https://github.com/Mert-Guenduez/learnityourself/tree/master/Backend/tests`
3. `https://jenkins.io/`
4. `http://91.205.172.109:8080/`
5. `https://github.com/Mert-Guenduez/learnityourself`
6. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/Testplan.md`

flashcardcommunity (2019-04-30 08:38:38)
Hello, i see your UI performs tests in the frontend and integration tests in the backend. But I don’t see your third type of test (unit tests) documented. You have several broken links in your markdown in chapters 4.1 and 7.1. Maybe you can add them. Greetings Sascha@flashcardcommunity

learnityourselfdhbw (2019-04-30 08:53:08)
Thanks for your comment. We will go through our markdown and make sure the links work. Regarding the third type of Test, we will add that soon. With regards Jonas@LearnItYourself

perfecttimese (2019-04-30 08:40:12)
Hi guys, nice work with the unit tests. The test plan is also nice. And it's great that it is an MD-File. However there are some smaller issues. There are some broken links to pictures in it (our experience: check the File type and if they are written in capital or small letters. and you didn't fill out chapter 6 (but we believe Entry and Exit Criteria are important points to consider). For the future you might also want to consider using Codecov with a badge to quantify your test coverage. But still, you are doing a great job. So keep working hard on your project and we are certain that many people will learn IT themselves. ;-) Your friends from Perfect Time

learnityourselfdhbw (2019-04-30 08:54:58)
Thanks for your detailed reply. We will make sure to fix the broken links and add the missing points. We wanted to add a code coverage Badge but didn't find a good service yet so thanks for the Recommendation of Codecov With regards Jonas@LearnItYourself

2.2 May

Week 16 - Refactoring (2019-05-06 19:53)

Hello everyone,

this weeks task was to refactor some code. As this wasn't a group work, everyone will show and link to their own GitHub Repository so you can see how we each refactored the given code.

Angelika:
My Refactoring can be seen in this [1]GitHub-Repository. I used the Fowler Code from this [2]GitHub-Repository. I used the first chapter of [3]Refactoring by Fowler to refactor the Code

as well as this [4]Catalog of Refactorings , making sure to name each commit after the issue fixed.

Jannik:

My Refactoring can be seen in this [5]GitHub-Repository. I used the Fowler Code from [6]here. While working i used the first chapter of [7]Refactoring by Fowler as well as this [8]Catalog of Refactorings, making sure to name each commit after the issue fixed.

Jonas:
My Refactoring can be seen in this [9]GitHub-Repository. I used the Fowler Code from [10]here. While working i used the first chapter of [11]Refactoring by Fowler as well as this [12]Catalog of Refactorings, making sure to name each commit after the issue fixed.

Mert:

The Refactored Code I have written can be found in my [13]GitHub-Repository. I used the Fowler Code from [14]here. While working i used the first chapter of [15]Refactoring by Fowler as well as this [16]Catalog of Refactorings, making sure to name each commit after the issue fixed.

1. https://github.com/AngelikaSudheimer/Hochschule-SE

2. https://github.com/thedomdom/Fowler-Refactoring/tree/70fc87873f1eeb10693c2cf1dd93882f3bfd7549

3. https://www.csie.ntu.edu.tw/~r95004/Refactoring_improving_the_design_of_existing_code.pdf
4. https://refactoring.com/catalog/
5. https://github.com/JannikAdam97/Fowler
6. https://github.com/gnilkreb/Fowler/tree/c0e1c7a21a5335d7e475c2c795ed77deec37b776
7. https://www.csie.ntu.edu.tw/~r95004/Refactoring_improving_the_design_of_existing_code.pdf
8. https://refactoring.com/catalog/
9. https://github.com/JoHi21/SE_Fowler
10. https://github.com/gnilkreb/Fowler/tree/c0e1c7a21a5335d7e475c2c795ed77deec37b776
11. https://www.csie.ntu.edu.tw/~r95004/Refactoring_improving_the_design_of_existing_code.pdf
12. https://refactoring.com/catalog/
13. https://github.com/Mert-Guenduez/Fowler
14. https://github.com/gnilkreb/Fowler/tree/c0e1c7a21a5335d7e475c2c795ed77deec37b776
15. https://www.csie.ntu.edu.tw/~r95004/Refactoring_improving_the_design_of_existing_code.pdf
16. https://refactoring.com/catalog/

Christian Schweigel (2019-05-07 08:32:43)
Hello learnityourself-Team, as far as I can see your repositories are looking good. It's also good that you named your commits after the refactoring methods you used, so it's easier to get an overview. But there are only two of four repositories available right now, maybe you forgot to put the other links in your blog post as well. Best regards, Christian@DigiWill

Tenniskönig (2019-05-07 09:27:24)
Hey guys, as Christan already had a look at two of our repos. I had a look at Jonas. The Repo from Jannik is still missing! First of all you did well. I am just wondering a little bit if your test are running, as the logs say something diffrent. I didnt execute them, but maybe you should have a look. Greeting Tenniskönig

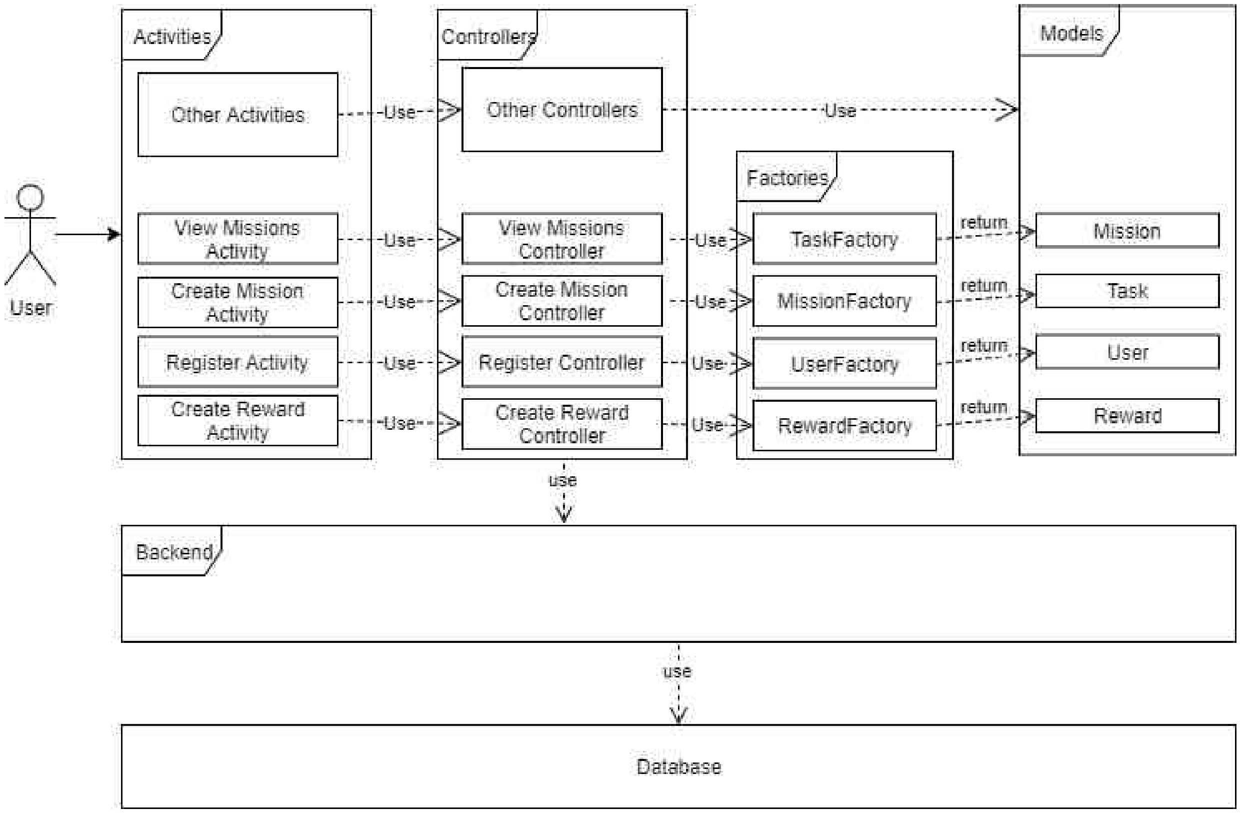
jonny9298 (2019-05-13 11:31:55)
Hey, I've noticed that the link to your current blogpost in the peer review central seems to be broken. Just wanted to inform you :) Your Jonny from turnie.re

Week 17 - Design Patterns (2019-05-13 01:22)

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

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.



[2]This example commit shows how the new Pattern was integrated into our Code.
Using a **TaskFactory** a task 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.

1. https://www.oodesign.com/
2. https://github.com/Mert-Guenduez/learnityourself/commit/bf8f407522281273be9363790ff0ca5ed2d4488e

jonny9298 (2019-05-13 11:57:32)
Hey, I've noticed that your link to this blog post is corrupt in the peer review central. I've already commented about this [here](https://learnityourselfdhbw.wordpress.com/2019/05/06/week-16-ref actoring/ #comment-92), while said comment could be deleted, if you want to. Now onto your actual post: I like that you've explained to us why you wouldn't have used this, if you wouldn't have had to do this homework. Nevertheless the question arises, whether there are no different design patterns that you could've easily implemented, that were useful. Furthermore you forgot to link or showcase the differences

in code before and after you've implemented the pattern, aswell as you didn't show a class diagram of before the pattern has been implemented. Since you're using some sort of version control system, it shouldn't be too much of a hassle to give us said information, even after the code has been committed. Last but not least you should probably also take a look at other sources for information, since informing yourself from solely one source makes you extremely subjective to misinformation or even manipulation. Your jonny from turnie.re

learnityourselfdhbw (2019-05-13 20:09:55)
Thanks for you comment! We fixed the link and will soon add some code to show the changes we made. We did look into multiple sources but chose to only list one here. Jonas@LearnItYourself

EventLAB Team (2019-05-13 23:49:58)
Hey guys, we liked the choice you made regarding the design pattern you used (we also used the factory method :D). As mentioned by jonny it would be great if you could link the code before and after implementing the pattern and also create class diagrams to make it easier to see the changes you have made. Stay on track Your EventLAB Team

simonleitl (2019-05-14 13:40:47)
Hey Guys, we really like your blogpost. It's possible to understand what you've done here without previous knowledge. It seems like it makes the code clearer. That's nice. Keep up the good work! Greetings, Simon/ FyF-Team

Week 18 - 2nd Retrospective (2019-05-19 22:45)

Hey Everyone,

this week we had our second retrospective. Just like the last time, everyone took some time to think about what was working well in our team and what still needs some refining.

What worked well? In the last retrospective we stated that we had problems with distributing our tasks properly, which improved a lot since then. Also, we hoped to stick more to our coding and git conventions, which also improved. Additionally, the speed in which we produced code increased marginally.

What didn't work so well? Firstly, we realized, that more often than not we have a different picture of the product we want to create. This creates misunderstandings regarding our interfaces, as an example. We planned to fix this by having more meetings in general.
Secondly, our risk management did not turn out to work very well in a certain aspect. One of our team member's laptop and computer both stopped functioning for multiple weeks and we didn't really have a plan B for this.
Lastly, we caught ourselves often completing our work hours before it is due. This also makes it difficult to test, fix and verify our work. To combat this, we set an internal deadline one day before the official deadline.

However, all in all we are pretty happy with our workflow. We are eager to see the end product and are glad to keep working together as a team!

Stay tuned for more!

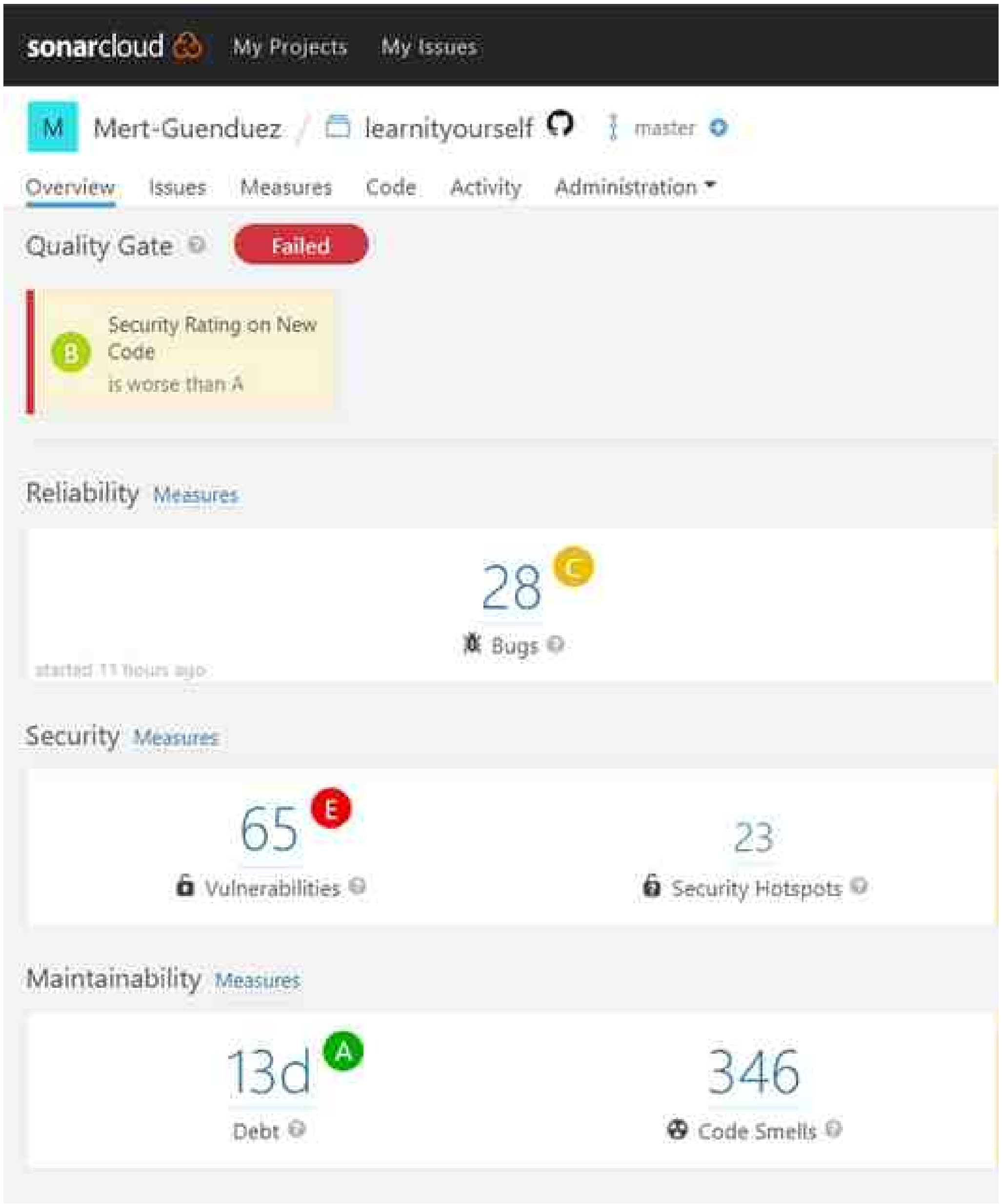
Sincerely,

learnityourselfdhbw

Week 19 - Metrics (2019-05-27 18:17)

This weeks task was to use Metrics for our Code. This enables us to look at our Code in numbers, which gives us an idea about its quality.

We went through different Metric Tools as integrating them into our Build Process turned out to be complicated. To make sure the code can be refactored in time we used the manually generated results with MetricsReloaded in our IDE. Therefore the tool we ended up using in our Jenkins is [1]Sonarcloud, which analyzes our code after every commit to the master branch and uploads the metrics results. This will be specified in our [2]Testplan. Additionally this provided a badge for code quality on our Github repository. Our first general code quality analysis results look like this:



Quality Gate Failed

B Security Rating on New Code is worse than A

Reliability Measures

28 C

Bugs

started 11 hours ago

Security Measures

65 E

Vulnerabilities

23

Security Hotspots

Maintainability Measures

13d A

Debt

346

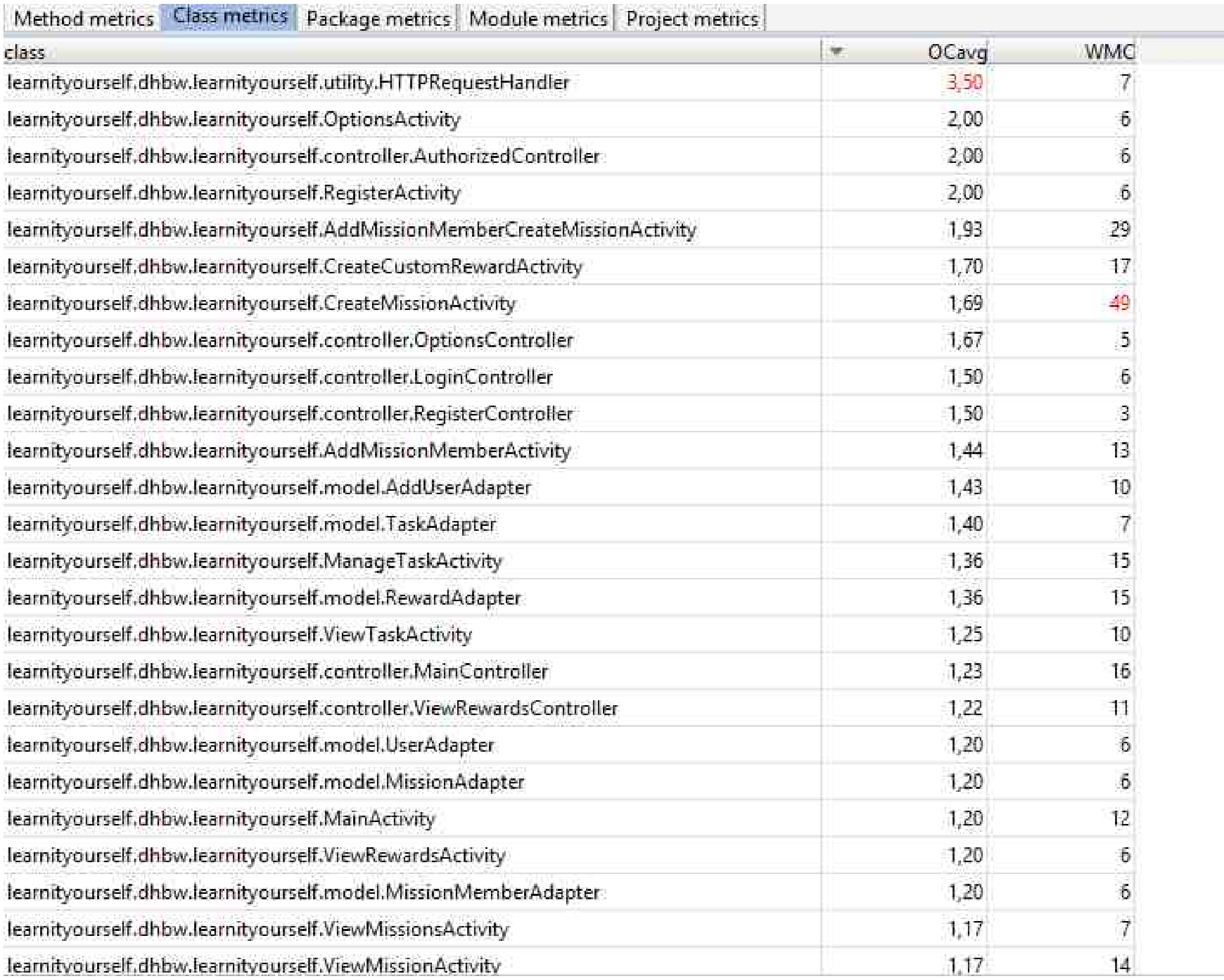
Code Smells

As you can see, we have quite some vulnerabilities and bugs we will try to work on. You can view the dashboard [3]here.

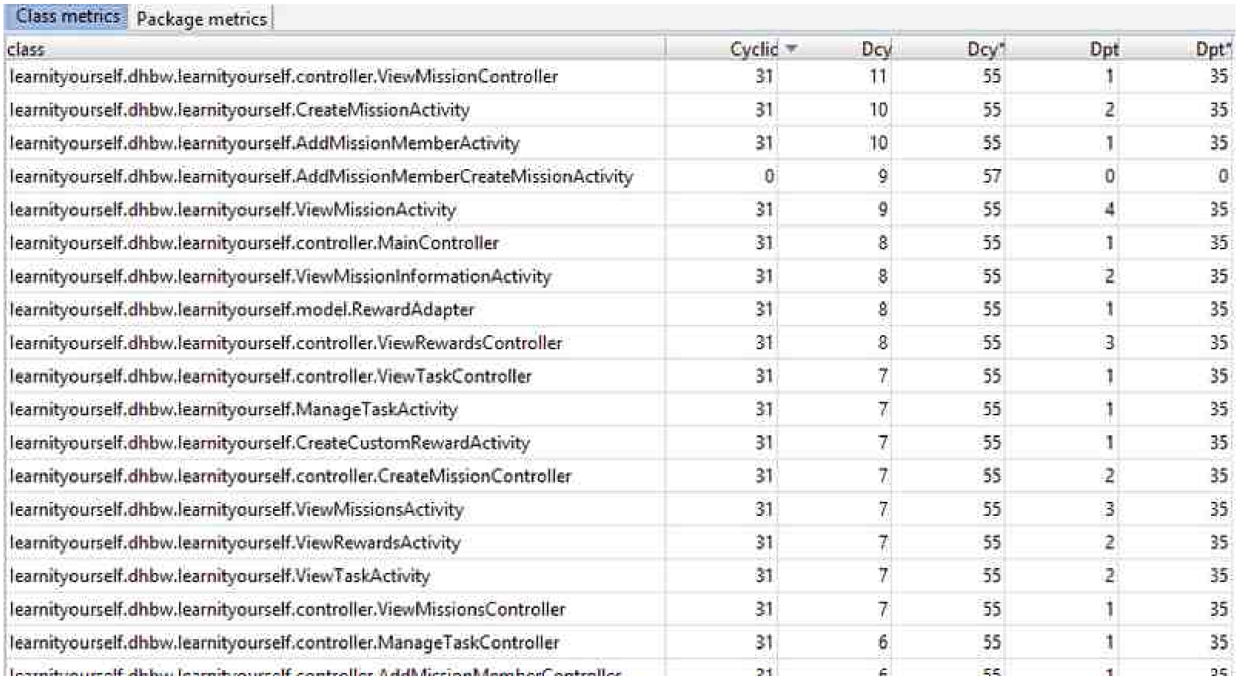
Further we used another tool to show us more detailed metrics:

Initial results

As a manual metrics analysis tool we chose to use Average Cyclic Method Complexity (OCavg) and Direct Dependencies of each class (Dcy). The initial result for our Code looks like this:



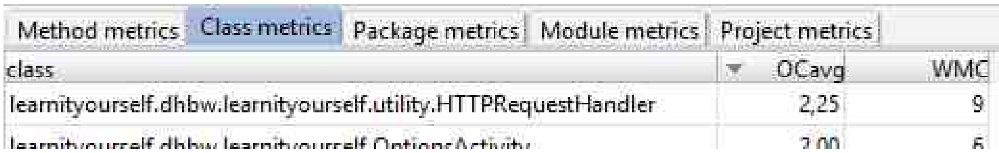
class	OCavg	WMC
learnityourself.dhbw.learnityourself.utility.HTTPRequestHandler	3,50	7
learnityourself.dhbw.learnityourself.OptionsActivity	2,00	6
learnityourself.dhbw.learnityourself.controller.AuthorizedController	2,00	6
learnityourself.dhbw.learnityourself.RegisterActivity	2,00	6
learnityourself.dhbw.learnityourself.AddMissionMemberCreateMissionActivity	1,93	29
learnityourself.dhbw.learnityourself.CreateCustomRewardActivity	1,70	17
learnityourself.dhbw.learnityourself.CreateMissionActivity	1,69	49
learnityourself.dhbw.learnityourself.controller.OptionsController	1,67	5
learnityourself.dhbw.learnityourself.controller.LoginController	1,50	6
learnityourself.dhbw.learnityourself.controller.RegisterController	1,50	3
learnityourself.dhbw.learnityourself.AddMissionMemberActivity	1,44	13
learnityourself.dhbw.learnityourself.model.AddUserAdapter	1,43	10
learnityourself.dhbw.learnityourself.model.TaskAdapter	1,40	7
learnityourself.dhbw.learnityourself.ManageTaskActivity	1,36	15
learnityourself.dhbw.learnityourself.model.RewardAdapter	1,36	15
learnityourself.dhbw.learnityourself.ViewTaskActivity	1,25	10
learnityourself.dhbw.learnityourself.controller.MainController	1,23	16
learnityourself.dhbw.learnityourself.controller.ViewRewardsController	1,22	11
learnityourself.dhbw.learnityourself.model.UserAdapter	1,20	6
learnityourself.dhbw.learnityourself.model.MissionAdapter	1,20	6
learnityourself.dhbw.learnityourself.MainActivity	1,20	12
learnityourself.dhbw.learnityourself.ViewRewardsActivity	1,20	6
learnityourself.dhbw.learnityourself.model.MissionMemberAdapter	1,20	6
learnityourself.dhbw.learnityourself.ViewMissionsActivity	1,17	7
learnityourself.dhbw.learnityourself.ViewMissionActivity	1,17	14



class	Cyclic	Dcy	Dcy*	Dpt	Dpt*
learnityourself.dhbw.learnityourself.controller.ViewMissionController	31	11	55	1	35
learnityourself.dhbw.learnityourself.CreateMissionActivity	31	10	55	2	35
learnityourself.dhbw.learnityourself.AddMissionMemberActivity	31	10	55	1	35
learnityourself.dhbw.learnityourself.AddMissionMemberCreateMissionActivity	0	9	57	0	0
learnityourself.dhbw.learnityourself.ViewMissionActivity	31	9	55	4	35
learnityourself.dhbw.learnityourself.controller.MainController	31	8	55	1	35
learnityourself.dhbw.learnityourself.ViewMissionInformationActivity	31	8	55	2	35
learnityourself.dhbw.learnityourself.model.RewardAdapter	31	8	55	1	35
learnityourself.dhbw.learnityourself.controller.ViewRewardsController	31	8	55	3	35
learnityourself.dhbw.learnityourself.controller.ViewTaskController	31	7	55	1	35
learnityourself.dhbw.learnityourself.ManageTaskActivity	31	7	55	1	35
learnityourself.dhbw.learnityourself.CreateCustomRewardActivity	31	7	55	1	35
learnityourself.dhbw.learnityourself.controller.CreateMissionController	31	7	55	2	35
learnityourself.dhbw.learnityourself.ViewMissionsActivity	31	7	55	3	35
learnityourself.dhbw.learnityourself.ViewRewardsActivity	31	7	55	2	35
learnityourself.dhbw.learnityourself.ViewTaskActivity	31	7	55	2	35
learnityourself.dhbw.learnityourself.controller.ViewMissionsController	31	7	55	1	35
learnityourself.dhbw.learnityourself.controller.ManageTaskController	31	6	55	1	35
learnityourself.dhbw.learnityourself.controller.AddMissionMemberController	31	6	55	1	35

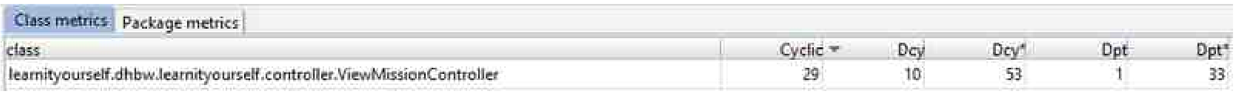
Changes to our code

We decided to take a look at HTTPRequestHandler regarding it's OCavg. As you are able to see in this [4]Commit, it consisted of a huge method. Extracting parts of that method into two new methods, reduced the OCavg of this Class noticeably:



class	OCavg	WMC
learnityourself.dhbw.learnityourself.utility.HTTPRequestHandler	2,25	9
learnityourself.dhbw.learnityourself.OptionsActivity	2,00	6

Additionally we decided to bring down the amount of Dependencies for the MissionViewController down to 10 by removing the unnecessary use of the Helper class as you can see [5]here. This changed the amount dependencies like this:



class	Cyclic	Dcy	Dcy*	Dpt	Dpt*
learnityourself.dhbw.learnityourself.controller.ViewMissionController	29	10	53	1	33

What we didnt change

Another piece of code the metrics suggested us to change the dependencies to our HTTPRequestHandler. It causes a huge number of dependencies because it is used in all of our Controller Classes. That is because it provides static functions we need for the communication between Frontend and Backend. As these function calls are very similar each time, it wouldn't make sense to implement it for each Controller individually.

1. `https://sonarcloud.io/`
2. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/Testplan.md`
3. `https://sonarcloud.io/organizations/mert-guenduez-github/projects`
4. `https://github.com/Mert-Guenduez/learnityourself/commit/2d846407c37a55b1da48c235f0d4098e03681539?diff=split`
5. `https://raw.githubusercontent.com/Mert-Guenduez/learnityourself/master/Documentation/Screenshots/MetricsReloaded_Dependency_refactored.png`

---

mljenne (2019-05-28 08:41:27)
Hi Lern it yourself-Team, you did a good job this week. To use sonarcloud is very useful to see what the quality of your code is about. You have to do some changes in your code as well. I see that the WMC-value of the CreateMissionActivity is the highest of all, is there any posibility to reduce this? Or is that a part you do not want to do like your HTTP request handler? Kind regards, FyF-Team

perfecttimese (2019-05-28 08:53:35)
Hey guys, Your sonarcloud looks impressive. It seems to be a perfect tool for your project. You also explained all grading criteria issues. So did well during the last week. The only point we did not get is, what does it mean that Average Cyclic Method Complexity (OCavg) is a MANUAL TOOL? Maybe you can enlighten us. THX. But great work anyway. Keep it up. ;-) Your friends from Perfect Time

Joshua Schulz (2019-05-28 08:55:05)
Hi guys, first of all: Good job! This blog post looks very good. You have provided a good overview about your metrics and what you have changed because of them. There is one point I want to mention regarding the grading criteria: I think you should explain what kind of issue is represented by metrics like "OCavg" so anyone can understand all the numbers and your changes. This is also a requirement in the grading criteria. In the future it would be nice to see your second metrics tool in your automated build process. Maybe you should consider change your metrics tool to codacy or something that is very easy to integrate to github and jenkins. All in all, well done. Regards, Joshua

2.3 June

Week 18 - Testing summarized (2019-06-03 11:36)

For the final presentation of our Project we needed three kinds of tests in total.
The first two we implemented were UI-Tests using Cucumber and then Unit-Test using phpUnit.

For our UI Tests we covered 66 % of our Activities with Feature files.

As a third type of test we decided to create our own [1]Score-Sheet
to enable Test-Users to evaluate our application from a real human perspective.

To conclude, we used the following types of testing, all of this is ofcourse specified in our [2]Testplan

[3]UI-Tests
[4]Unit-Tests
[5]User-Tests

The test coverage can be seen on our Jenkins

1. https://github.com/Mert-Guenduez/learnityourself/blob/master/UserTesting/ScoreSheet.xlsm
2. https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/Testplan.md
3. https://github.com/Mert-Guenduez/learnityourself/tree/master/app/src/androidTest
4. https://github.com/Mert-Guenduez/learnityourself/tree/master/Backend/tests
5. https://github.com/Mert-Guenduez/learnityourself/tree/master/UserTesting

Week 20 - Installation (2019-06-10 10:40)

After finally finishing our Application we're giving you the opportunity to install it on your own server and run the app on our own mobile phone.

Instructions on how to intall the Server can be found [1]here.

The APK File for our App can be downloaded [2]here.

If the installation worked for you please let us know with a comment, you can of course also point out if there are any issues.

1. `https://github.com/Mert-Guenduez/learnityourself/blob/master/Documentation/Backend%20Installation.md`
2. `https://github.com/Mert-Guenduez/learnityourself/blob/master/learnityourself.apk?raw=true`

Felix Hausberger (2019-06-11 09:29:16)
Hey Team learnityourself, I was able to successfully install your server locally using xampp, but please add to your installation guide, that people have to create a technical user WITH a password for the database. Hope the error messages I received when connecting to the server via the browser are not serious? Installing the app itself was no problem for me. Here a screenshot as a proof of installation: https://raw.githubusercontent.com/raphaelmue/dashup/master/docs/project _management/cross _project _work/learnityourself.jpg All in all, good job! Yours sincerely, Felix

learnityourselfdhbw (2019-06-11 09:31:33)
Thanks for your effort to install our project! The errors displayed in the browsers are mere notices and won't cause any problems. We will add the information about needing a user with password for the database. With Regards Jonas@LearnItYourself

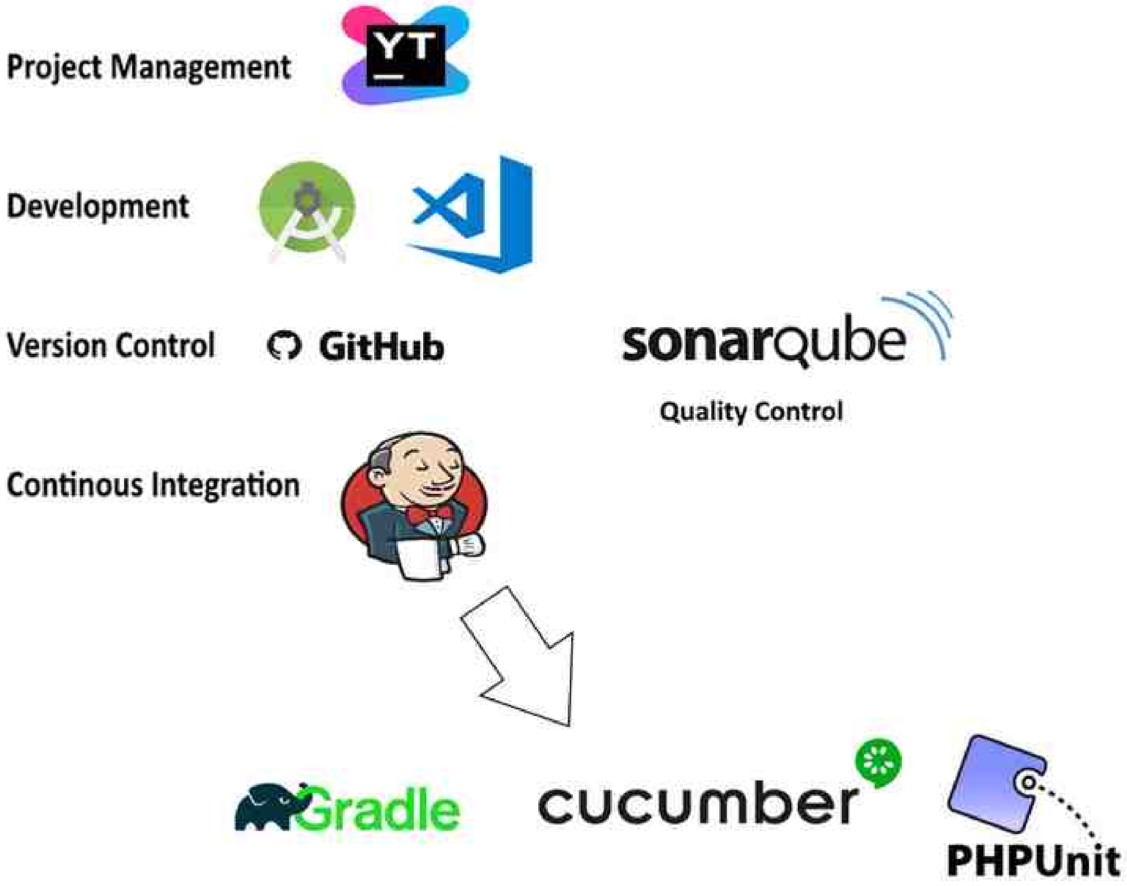
Continuous Integration and Tech Stack (2019-06-11 18:34)

Hey everyone,

this week we want to present you our Continuous Integration tool and on top of that tell you which technologies we used for the entire project.

We use Jenkins for Continuous Integration. We have integrated the build (Gradle), tests (PHPUnit), code coverage and metrics ([1]SonarQube) into this deployment cycle, which is triggered by every commit to the master branch of our [2]GitHub repository.

The programming languages we used are Java (Android) and PHP with a MySQL Database.
Our Tech Stack looks like this:



Stay tuned for the final weeks!

Sincerely,

learnityourselfdhbw

1. `https://sonarcloud.io/dashboard?id=Mert-Guenduez_learnityourself`
2. `https://github.com/Mert-Guenduez/learnityourself`

BLOGBOOKER

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

Edited: June 16, 2019