mixthatblog.wordpress.com

Contents

1. 2017

1.1 October

Week No.1 (2017-10-12 09:07)

We decided to make a website to control some kind of an external device like a robot. After a couple more moments the idea of an Cocktail-mixing website occurred to our minds.

The basic idea behind this is, that you have a website with many cocktail recipes and a login function. When you are logged in you can add your own recipes and with a button "Mix it!" the Cocktail can be mixed on an external device which will be build by the Mechanical Engineers.
Our website for the Mix-It! Project will be build with PHP for dynamics and functions, HTML for the content base and CSS for the style.
JavaScript with JQuery will be used for visual effects.
In order to save the user data and recipes we will be setting up an MySQL-Database.
For the interface and communication between the site and the hardware we will be using Ajax in order to send Json-files with the recipes.
On the raspberry itself, the Json-file will be interpreted by a Python program which then will control the proper outputs.

After the idea was created, we made a github repository for version control and added ZenHub for the projectmanagment.

(By Mix-It team)

---

thomaspoetzsch (2017-10-12 11:13:31)
Hey, I really like your idea, and it should be quite useful to have a nice interface for the machine provided by the other students. You could also implement some functions for finding drinks for given ingredients, etc. For getting drink recipes, you could fetch data from the International Bartenders Association site (http://iba-world.com/contemporary-classics/) or Wikipedia. As I said during the lecture, you should provide offline functionality, so that I can mix a drink if I my Internet is down or just not working. Are you going to provide some easy tool for accessing another person's machine by scanning a QR-Code or authenticating in a similar way? You should shorten your blog entry, so that it meets the "1-2 paragraphs" requirement. You could leave out information about the blog itself and the introduction. Sincerely, Thomas

Alexander Kimmig (2017-10-12 11:31:14)
Hey Thomas Thank you for your input, we will be including sorting and searching of the cocktails :) The offline functionality sounds like a good idea, so that people without internet access can still use the provided functions. We were already thinking about easy authenticating and also thought about QR-Codes, which we will most likely use. Thank you for the reminder with the paragraphs requirement, we will be editing it to meet those.

freshmangoshake (2017-10-12 11:32:55)
Hello there! Indeed this is a very good idea! What do you think about adding a task queue on your interface to order several drinks at a time? This might be very useful for parties. Also I like the idea of creating my own receips and sharing them to others. I am realy looking forward to seeing the results of your implementation and the final product! Kind regards, Fabian

Alexander Kimmig (2017-10-12 11:37:33)
Hey, Fabian, The queue sounds like a good idea which we will include for shure :) We will be thinking about the implementation which you will be able to see in future posts. Greedings, Alex

Midterm Summary | Mix It! (2017-12-15 12:15:15)
[...] Week No. 1 (Introduction) [...]

Final Post | Mix It! (2018-07-01 20:12:52)
[...] Week No.1 [...]

Week No. 1 (old) (2017-10-12 11:31)

In our course of Softwareengeneering we made small teams to realize a software idea of any kind. At the first day we decided to form a team of two people and made a brainstorming to find a project to realize.

After many ideas we decided to make some kind of website to control an external device like a robot. After a couple more moments the idea of an Cocktail-mixing website occurred to our minds.

The basic idea behind this is, that you have a website with many cocktail recipes and a login function. When you are logged in you can add your own recipes and with a button “Mix it!” the Cocktail can be mixed on an external device which will be build by the Mechanical Engineers.

For technology we firstly chose a blog with “Wordpress”, as it is very easily set up and maintained. We made small changes to the site like new images and a contact page, which will send the comment as an E-mail to Jenny and me.
Our website for the Mix-It! Project will be build with PHP for dynamics and functions, HTML for the content bases and CSS for the style.
JavaScript with JQuerry will be used for visual effects.
In order to save the user data and recipes we will be setting up an MySQL-Database.
For the interface and communication between the site and the hardware we will be using Ajax in order to send Json-files with the recipes.

On the raspberry itself, the Json-file will be interpreted by a Python program which then will control the proper outputs.

When the idea was created, we made a git repository for version controll and made this blog to document what we do. We configured the blog, changed the design and made a contact form, because the mechanical engeneers or people that are interested in out project can contact us.

Week No.2 **(2017-10-16 16:45)**

Technology and Roles

After we showed of our ideas last week, we continue by introducing our roles within the project as well as the used Technologies.

Technology:

Languages:
We will be using PHP to create dynamic Websites with login function.
Python programs on a Raspberry pi will receive and control the Hardware. With Ajax we will transfer data from PHP to Python.
HTML will be the basic construction of our website and we will be using CSS to style our website. JavaScript will be used for animations.

Hosting / Server:
We will use a Apache Tomcat container locally to test the Server while developing, once we have a release ready, we will setup a proper server with online access.
To store the user data as well as data for cocktails we will be using a MySQL database.

Version Control / Developing:
We will use agile development in order to quickly response to requirement changes or new ideas. In order to realize this, we will use Scrum development.
For a version control we are using GitHub with the ZenHub extension.

Roles:

As our team consists out of only two persons, it is quite hard to specify these roles directly, which is why roles like Tester are assigned to both of us.

Role
Jennifer
Alexander

Designer
X

Requirements Specifier

X

Implementer
X
X

Tester
X
X

Project Manager
X

Change Control Manager

X

---

necoproject (2017-10-18 19:15:46)
Dear members of the "Mix It!"-team, I think you’re still lacking roles for the categories "deployment" and "environment" according to the RUP model (https://www.ibm.com/developerworks/rational/library/apr05/crain/). Apart from that I really like the way you assigned two responsible team members for one category, because it assures a consistent workflow even if one of you gets sick. Best Regards, Team NeCo

Jennifer Hauß (2017-10-19 10:03:11)
Hello necoproject! We decided against using these two roles, because we think they are not really needed /used in our project. When we latet realize that we need them, we will assign them then. Thank you for your feedback! Regards Team Mix-It!

teamCCR (2017-10-19 10:38:06)
Hello Team Mix-It! It’s very interesting to follow your weekly updates. As for the entries, I suggest to mark the subheadings (Languages, Hosting/Server, etc.) in bold to improve readability. Besides from that, the technologies you use and the RUP assignements seem very reasoned. Yours faithfully, Team CCR

Mix-It! (2017-10-19 10:42:16)
Hello Team CCR! That's a good idea. I marked the subheadings in bold now. Thanks for feedback! Best regards Team Mix-It

dffcblog (2017-10-19 11:16:14)
Hello Team Mix It!, Your team roles look very good, we can see that you have used RUP for defining the roles. We noticed that you have 2 Testers but no one specified for designing the tests.

Midterm Summary | Mix It! (2017-12-15 12:15:18)
[…] Week No. 2 (Team roles & Technologies) […]

Final Post | Mix It! (2018-07-01 20:12:55)
[…] Week No.2 […]

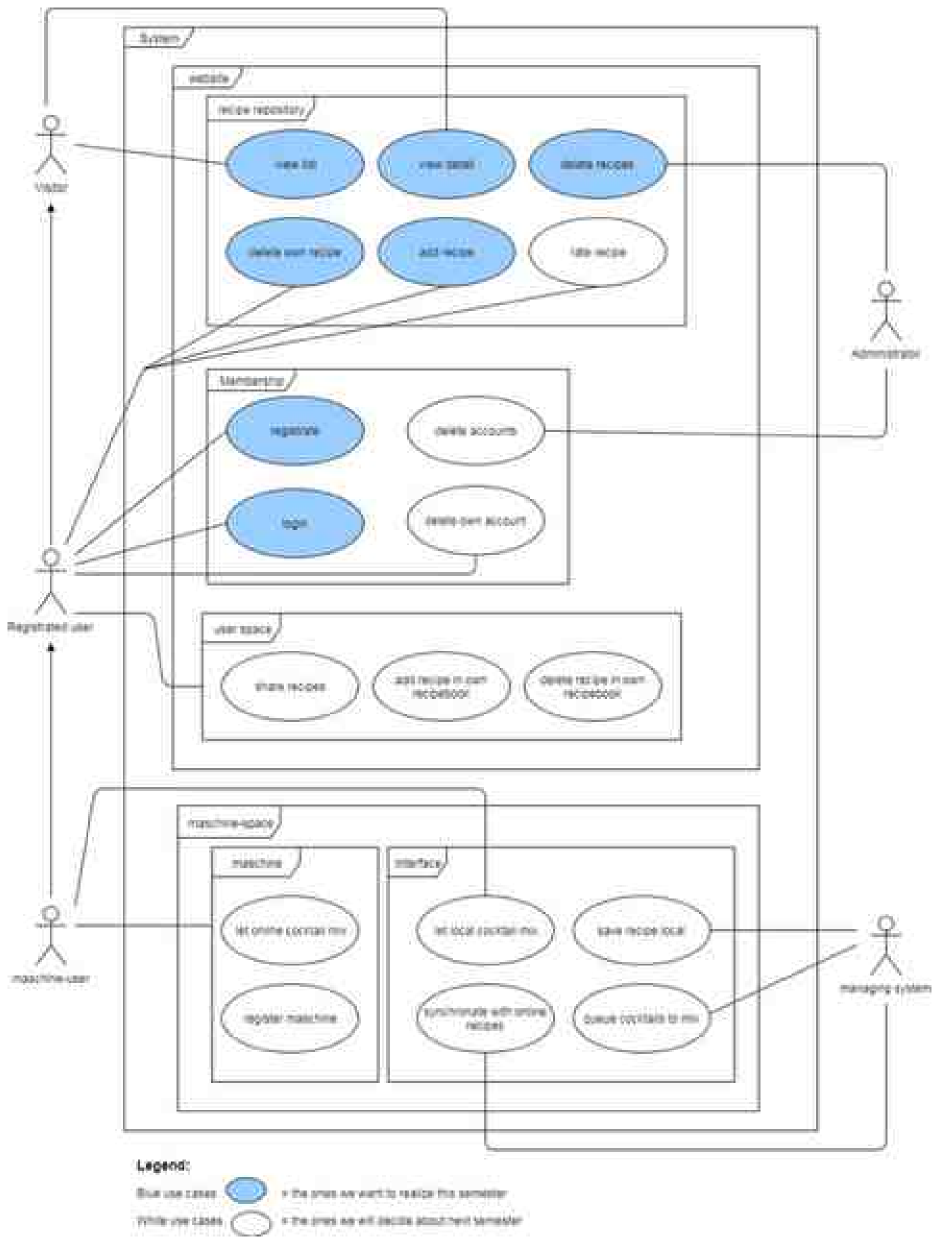
Week No. 3 (2017-10-22 18:41)

Hey All!

Here's our Software Requirements Specification(SRS) and our Use Case Diagram.

You can find them in our Git repository:
[1]SRS
[2]Use Case Diagram

The Use Case Diagram can also be found here in the blog:



The blue use cases are those, we want to realize this semester. Most of it will be setting up an webiste with an recipe repository and an personal space for registrated user.
In the next semester we will decide what we want to realize then.

Feel free to comment and ask questions!

Please note: The documents are not finished yet and will be updated soon!

1. `https://github.com/Mit-It/Documentation/blob/master/docu.md`
2. `https://github.com/Mit-It/Documentation/blob/master/Use_case.jpg`

possiblynotpotatoguy (2017-10-25 19:53:56)
Hello, Your Use Case Diagram seems to be structured quite well, but it could use some improvements to offer better readability. You connected your actors to the each use case with a seperate line. Naturaly you got a lot of lines which makes viewing the diagramm confusing. We advise you to just connect one line from an actor to a use case group. For instance registrated user and user space. You could just connect one line to the user space rectangle and it would be pretty clear that the registrated user can do all those use cases. Furthermore we would add a small legend into the diagram, which describes the colors, because the diagram should also make sense when viewed standanlone. Best regards Team react

Mix-It! (2017-10-26 09:23:11)
Hello Team React, Thank you for your hints! We updated our use case diagram. It has less lines and a legend now. Thanks for opinion. Best regards Team Mix-It!

thomaspoetzsch (2017-10-26 11:07:18)
Hey there, I think a lot of thought went into your overall use case diagram and your SRS, and I liked the improvements you made after Team reacts comment. When I look at your SRS, i get the expression that you somehow messed up your formatting. Have a look at https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet. If you still don't know which license to choose, take a look at https://choosealicense.com/. Best regards, Thomas

superwomantinf16b4 (2017-10-26 11:08:17)
Hello team Mix-It! From our point of view your UseCaseDiagram contains all important use cases. Also your SRS seems to be well structured and gives an overview of the most important requirements (of course you still have to add some aspects, but you can do it during the development process). :) We suggest you to fix the layout (the headlines should be bold/underlined etc.) and the table of contents of your SRS. Apart from that we really like your work! Best regards, Team SuperWoman

Mix-It! (2017-10-26 12:49:35)
Hey Team SuperWoman! Thanks for your feedback. We will fix the Layout of the SRS during the next days. Best regards Team Mix-It!

Week No. 8 | Mix It! (2017-11-29 21:13:42)
[...] but not least: We updated our scope. You can view our new scope here or on our github [...]

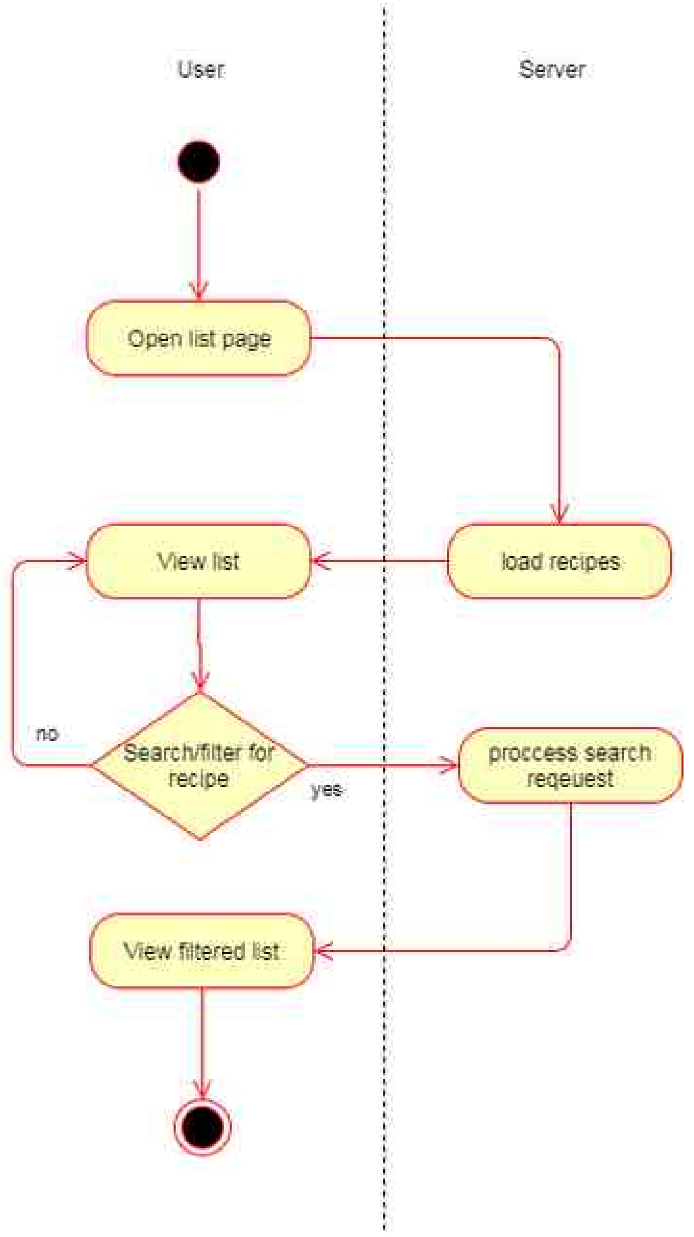
Midterm Summary | Mix It! (2017-12-15 12:15:21)
[...] Week No. 3 (SRS) [...]

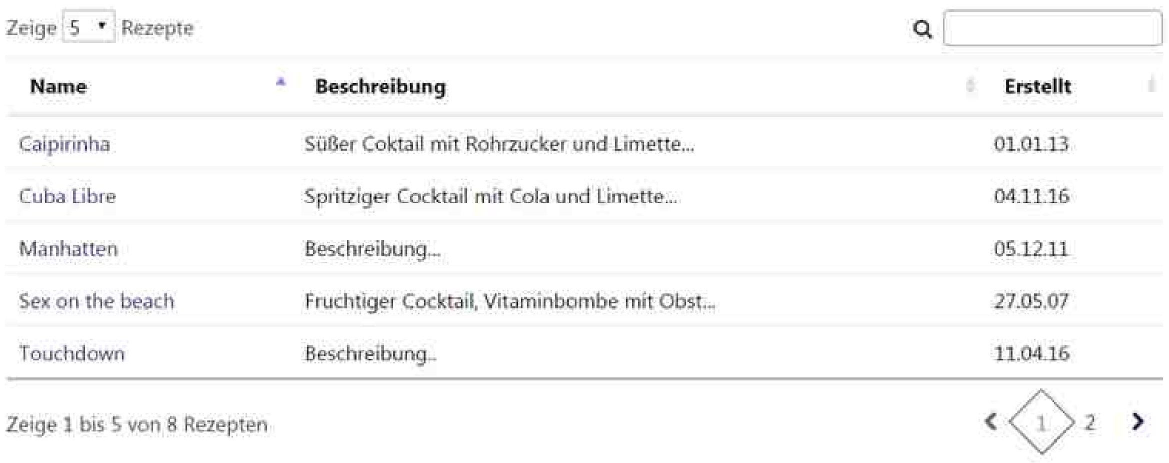
Final Post | Mix It! (2018-07-01 20:12:58)
[...] Week No. 3 [...]

Week No. 4 (2017-10-28 08:30)

This week we specified two use cases and made a first screendesign for both of them. You can view the results on [1]github or here on the blog:

view list

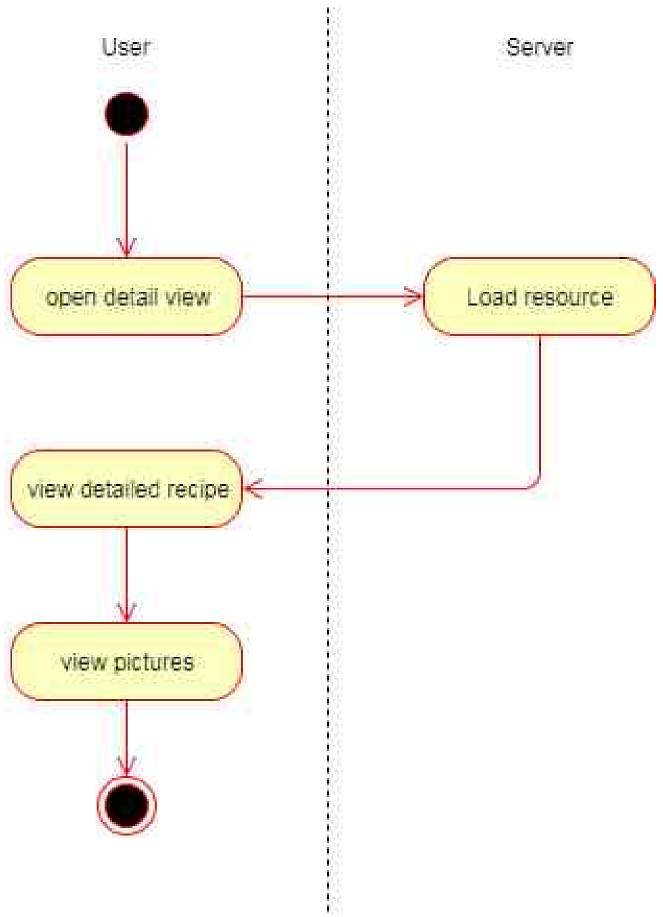

User
Server
Open list page
View list
load recipes
no
Search/filter for recipe
yes
proccess search reqeuest
View filtered list

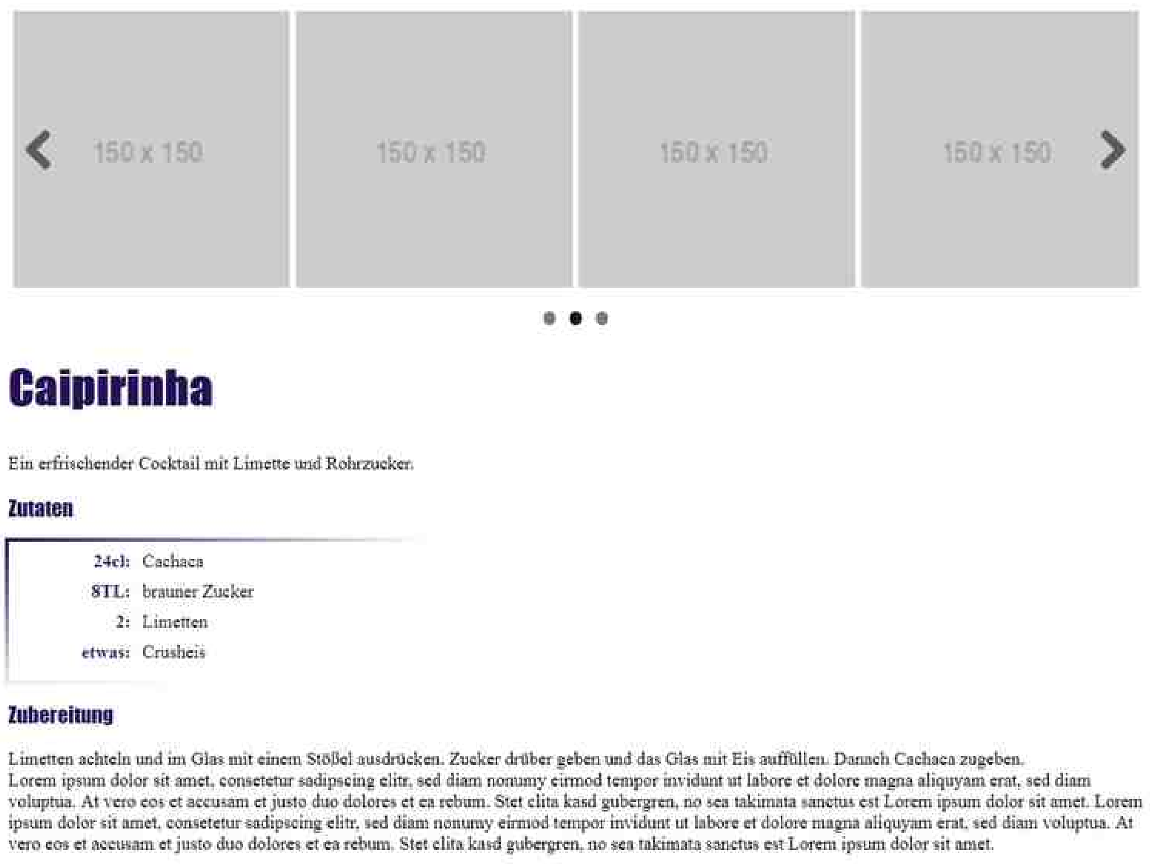

Zeige 5 Rezepte

Name	Beschreibung	Erstellt
Caipirinha	Süßer Coktail mit Rohrzucker und Limette...	01.01.13
Cuba Libre	Spritziger Cocktail mit Cola und Limette...	04.11.16
Manhatten	Beschreibung...	05.12.11
Sex on the beach	Fruchtiger Cocktail, Vitaminbombe mit Obst...	27.05.07
Touchdown	Beschreibung..	11.04.16

Zeige 1 bis 5 von 8 Rezepten

view detail


User
Server
open detail view
Load resource
view detailed recipe
view pictures



Caipirinha

Ein erfrischender Cocktail mit Limette und Rohrzucker.

Zutaten

24cl:	Cachaca
8TL:	brauner Zucker
2:	Limetten
etwas:	Crusheis

Zubereitung

Limetten achteln und im Glas mit einem Stößel ausdrücken. Zucker drüber geben und das Glas mit Eis auffüllen. Danach Cachaca zugeben.
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Please note: The screendesign are only first ideas and should help us to implement the use cases. The final look can still change.

1. `https://github.com/Mit-It/Documentation/tree/master/Week%204`

---

thomaspoetzsch (2017-11-01 19:25:46)
Hey there, I really like your mockups, they seem to be really close to your end product and helpful. The condition that the view list activity diagram contains could be moved in front of "view list", to make the diagram a lot clearer. All the UI activities that your second diagram contains could be grouped. You should also think about creating separate and detailed documents for the specified use cases and referring to them from your SRS. If you put some time in to your documentation repository sorting and reorganizing it, it would be easier to follow along with your progress and orient oneself. Best regards, Thomas

Mix-It! (2017-11-02 11:17:44)
Thank you for your feedback! We will change the diagrams the next days and also we will make the documents tor our SRS. Best regards Team Mix-It!

dffcblog (2017-11-02 12:27:23)
Hello team mix it, We really like your work. Your activity diagrams are good and provide decent information about your use cases. In your second activity diagram there is no server thread :) does that mean all the data is offline and u can freely access

it, or is a internet connection required?

Mix-It! (2017-11-02 12:29:33)
Hello Team dffcblog! Thank you! You need an internet connection. We will add a server thread in that diagram. Thank you very much for your feedback Regards Team Mix-It!

Midterm Summary | Mix It! (2017-12-15 12:15:24)
[...] Week No. 4 (Use Cases) [...]

Final Post | Mix It! (2018-07-01 20:13:01)
[...] Week No. 4 [...]

1.2 November

Week No. 5 (2017-11-04 20:45)

Hello together!
We are now using Zenhub for organizing of our Scrum. Zenhub can't be integrated in our IDE, but as Zenhub is an extension for Github that doesn't impair us, because we integrated Github to our IDE.
On Friday our first sprint startet. It will durate for two weeks and end on Friday 24th.

You can view our board and burndown chart over the Zenhub web app, you have to log in with your github account.
[1]Board
[2]Burndown chart

1. `https://app.zenhub.com/workspace/o/mit-it/mix-it/boards?repos=106662421`
2. `https://app.zenhub.com/workspace/o/mit-it/mix-it/reports?report=burndown&milestoneId=2883316`

---

dennyfl (2017-11-06 16:24:07)
Hello Mix It!-Team, I was curious about your progress so i tried to check out your current sprint, but i can't seem to access it. Even after installing the add on, it won't show me your board or chart. Is it possible to enlarge those pictures a bit, because it is too small to read. Since i cannot really comment on your content i at least noticed something else. You set your sprint duration to 3 weeks, is there a reason for this long time? I think especially for a small team, shorter sprints would make more sense. best regards team dffc

Mix-It! (2017-11-07 09:30:41)
Hello Team dffc! You should have access over the web app of Zenhub now with your github account. Thanks for your hint! We also enlarged the picture of the board and splitted it into to pictures. The duration of our sprint is so long, because we wanted to realize an big task, what can't be done in a shorter time. Because everthing in this task belongs together it was hard to split it. Thanks for your comment! Team Mix-It!

Jannik Upmann (2017-11-07 07:09:47)
Hello Team Mix it, it is intresting to see someone working with another tool then JIRA. Can you use Zenhub with your IDE ? It looks like you Build the Application from two different point of views and meet at the middle. Or do you plan to develop the frontend first ? Your links Board and Burndownchart might be broken. You should fix this! Best regards Team React

Mix-It! (2017-11-07 09:34:05)
Hello Team React! We can not use Zenhub directly with our IDE. But we have git-integration and Zenhub is an extension for git, so it somehow belongs togheter. We plan to develop the frontent first. In this way we can test new functions directly. For the content we will first add some placeholders. Later they will be replaced by dynamic content. We fixed the links. You can now access our board and burndown chart, after login with your github account. Thanks for your comment! Team Mix-It!

Team VSS (2017-11-07 11:12:40)
Hey Mixers, :) I don't think you need a guest-account. I can watch your boards with my own account. Best regards, Team VSS PS. Part of the Homework is it to setup IDE Integration. So you should mention that part in the block post, even if you don't got it. So that you describe how much that effects you. And how you work around it.

Mix-It! (2017-11-07 12:58:41)
Hello Team VSS! You are right! I removed the guest account and edited the blog entry. Thanks for your hints Regards Team Mix-It!

Midterm Summary | Mix It! (2017-12-15 12:15:27)
[...] Week No. 5 (Scrum) [...]

Final Post | Mix It! (2018-07-01 20:13:03)
[...] Week No. 5 [...]

Week No. 6 (2017-11-12 21:14)

Hey together!

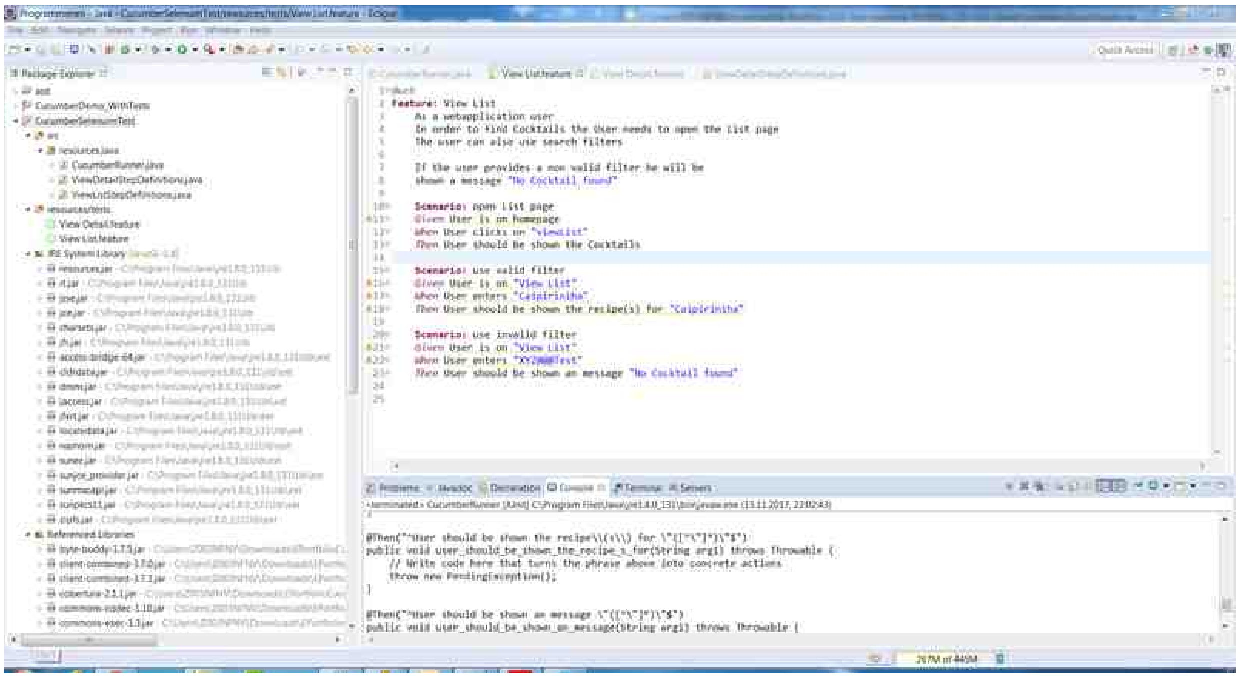
This week we learned to use Cucumber an Gherkin. We created feature files for our use cases view list and view detail.

You can find them on this github repository:

[1]view list

[2]view detail

Our IDE also supports Syntax Highlighting as you can see here:



```
#Author
Feature: View List
    As a webapplication user
    In order to find Cocktails the User needs to open the List page
    The user can also use search filters

    If the user provides a non valid filter he will be
    shown a message "No Cocktail found"

    Scenario: open List page
    Given User is on homepage
    When User clicks on "viewList"
    Then User should be shown the Cocktails

    Scenario: use valid filter
    Given User is on "View List"
    When User enters "Caipirinha"
    Then User should be shown the recipe(s) for "Caipirinha"

    Scenario: use invalid filter
    Given User is on "View List"
    When User enters "XY2###Test"
    Then User should be shown an message "No Cocktail found"
```

```
<terminated> CucumberRunner [JUnit] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (13.11.2017, 22:02:43)

@Then("^User should be shown the recipe\\(s\\) for \"([^\"]*)\"$")
public void user_should_be_shown_the_recipe_s_for(String arg1) throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Then("^User should be shown an message \"([^\"]*)\"$")
public void user_should_be_shown_an_message(String arg1) throws Throwable {
```

1. `https://github.com/Mit-It/Mix-It/blob/master/CucumberSelenuimTest/resources/tests/View%20List.feature`
2. `https://github.com/Mit-It/Mix-It/blob/master/CucumberSelenuimTest/resources/tests/View%20Detail.feature`

---

necoproject (2017-11-16 08:30:42)
Hey Team Mix It, your feature files look very good, but it seems that you're missing a proof that your IDE supports syntax highlighting. Other than that it looks very good. Best regards, Team NeCo

Mix-It! (2017-11-16 11:18:28)
Hello Team NeCo! Thanks for the reminder. We added a Screenshot of the Syntax-highlighting Regards Team Mix-It!

possiblynotpotatoguy (2017-11-16 08:41:43)
Hey Team Mix It, your feature files looks really good, but you called your feature in viewDetail.feature "View List". Also it has only one scenario. A Feature File should consist at least of two scenarios. In View LIst.feature you have the Given "User is on Home". Does it mean the user is on the homepage or something else? Best regards Team react

Mix-It! (2017-11-16 11:19:12)
Hey Team react! We added new scenearious and fixed what you mentioned Regards Team Mix-It!

Midterm Summary | Mix It! (2017-12-15 12:15:30)
[...] Week No. 6 (Testing) [...]

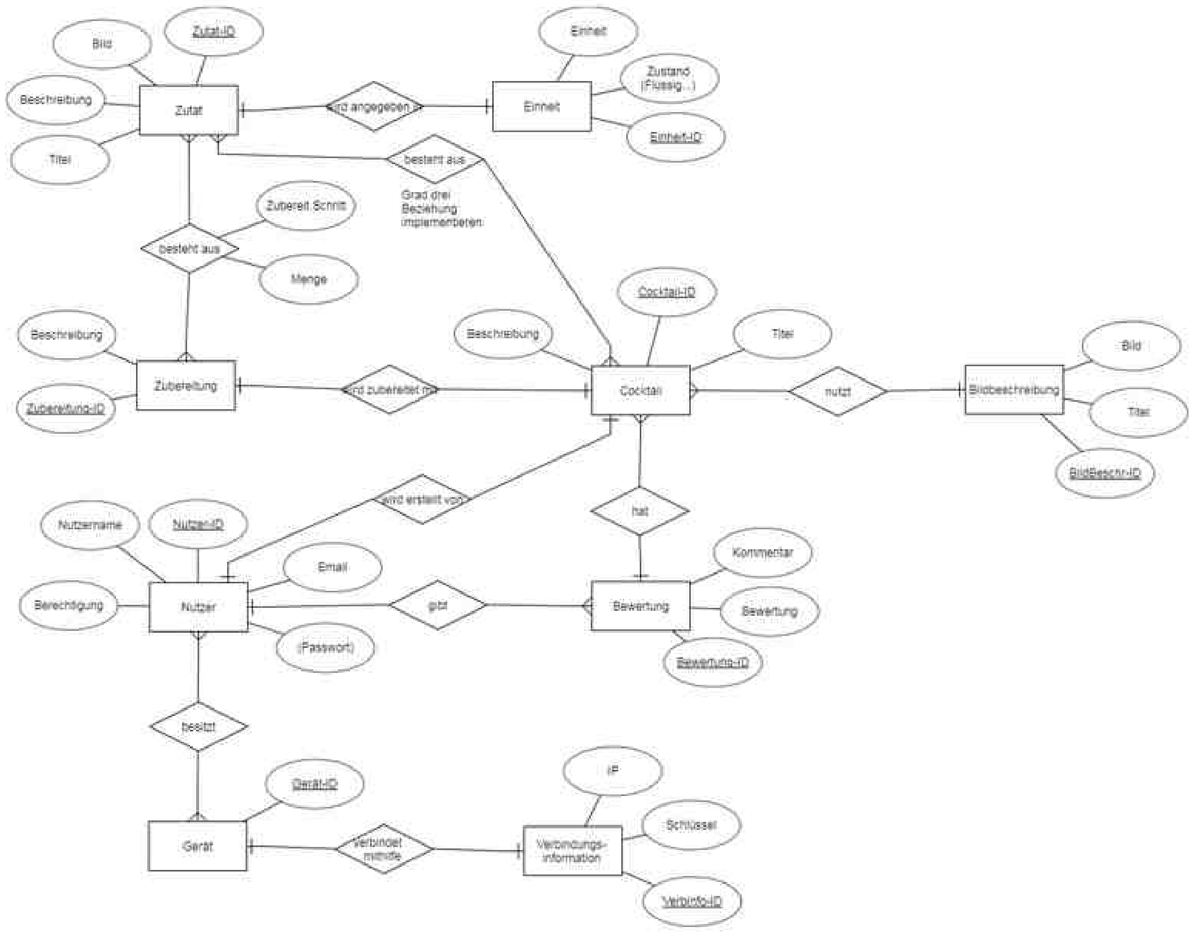
Final Post | Mix It! (2018-07-01 20:13:06)
[...] Week No. 6 [...]

Week No. 7 (2017-11-22 14:21)

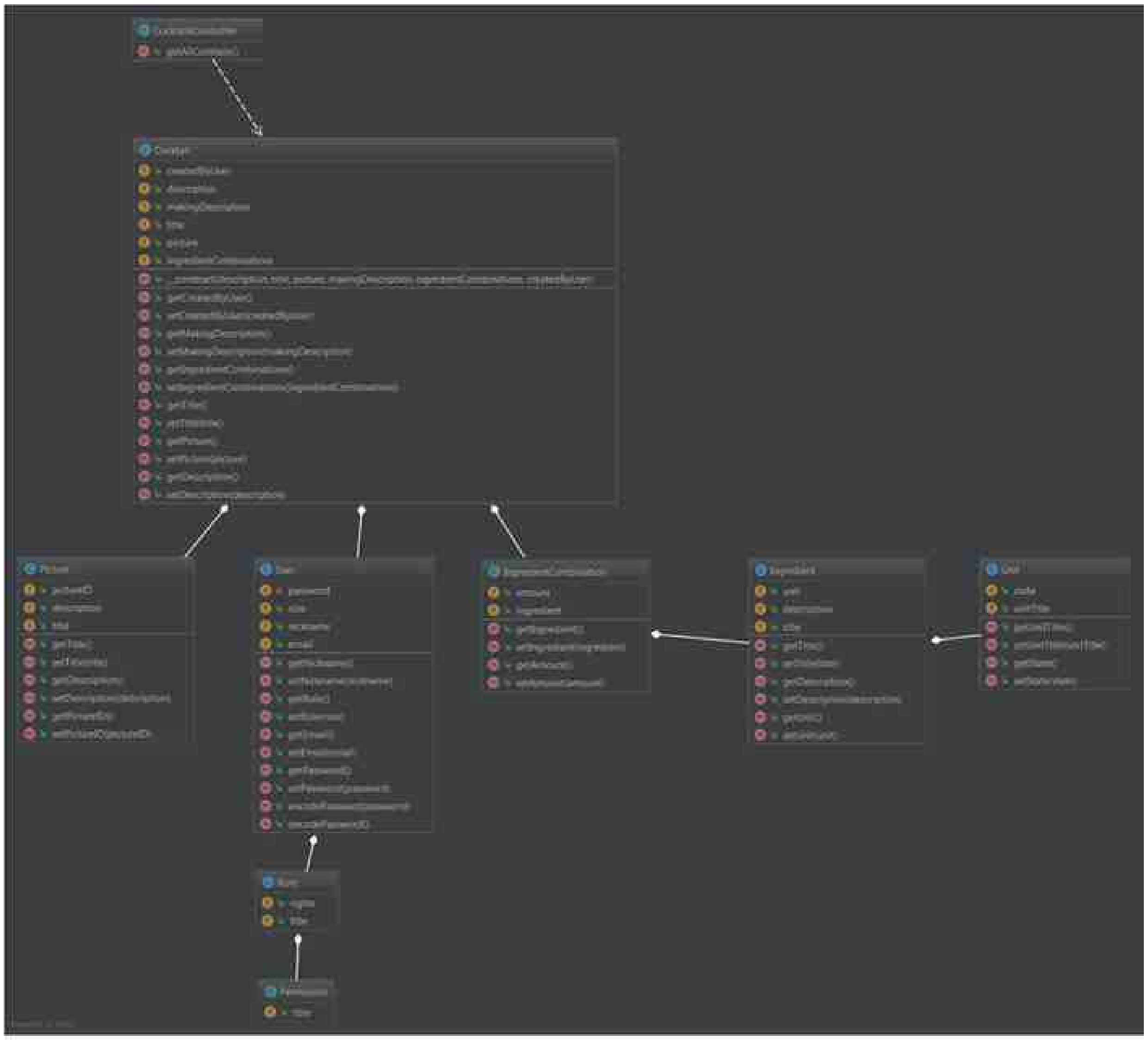
Hey Guys!

This week we create a database diagram and a class diagram from our already existing classes. We made the class diagram with phpstorm, but as it only shows arrows for "extends" and "implements" relationship, we had to add arrows manually.

Database Diagram:



Class diagram:



Let us know what you think about our diagrams!

necoproject (2017-11-23 09:52:01)
Hey Team MixIt, your database diagram looks very good. Maybe adding colors would have been cool, but it’s not that important. Your class diagram looks good as well. Greetings, NeCo

Jennifer Hauß (2017-11-23 11:23:57)
Hey Team Neco! That’s a good idea. We will add some colors soon, cause the database model will change anyway Thanks for your comment Regards Team Mix-it!

schmiefri (2017-11-23 10:05:22)
Hi Folks, youd diagrams look fine, but you seem to have no real business logic right now, e.g. talking to the mix-bot. Detail question: Do you have thoughts on password storage? Like how do you plan to encrypt them? Will you salt them? Algorithms? Kind regards

Jennifer Hauß (2017-11-23 11:26:08)
Hey! Yeah that's right. We don't have logic yet, but we will do this soon and then update our diagram We haven't thought about this yet, but we will also do this soon. I think we will be using laravels login-system. Best regards Team Mix-It!

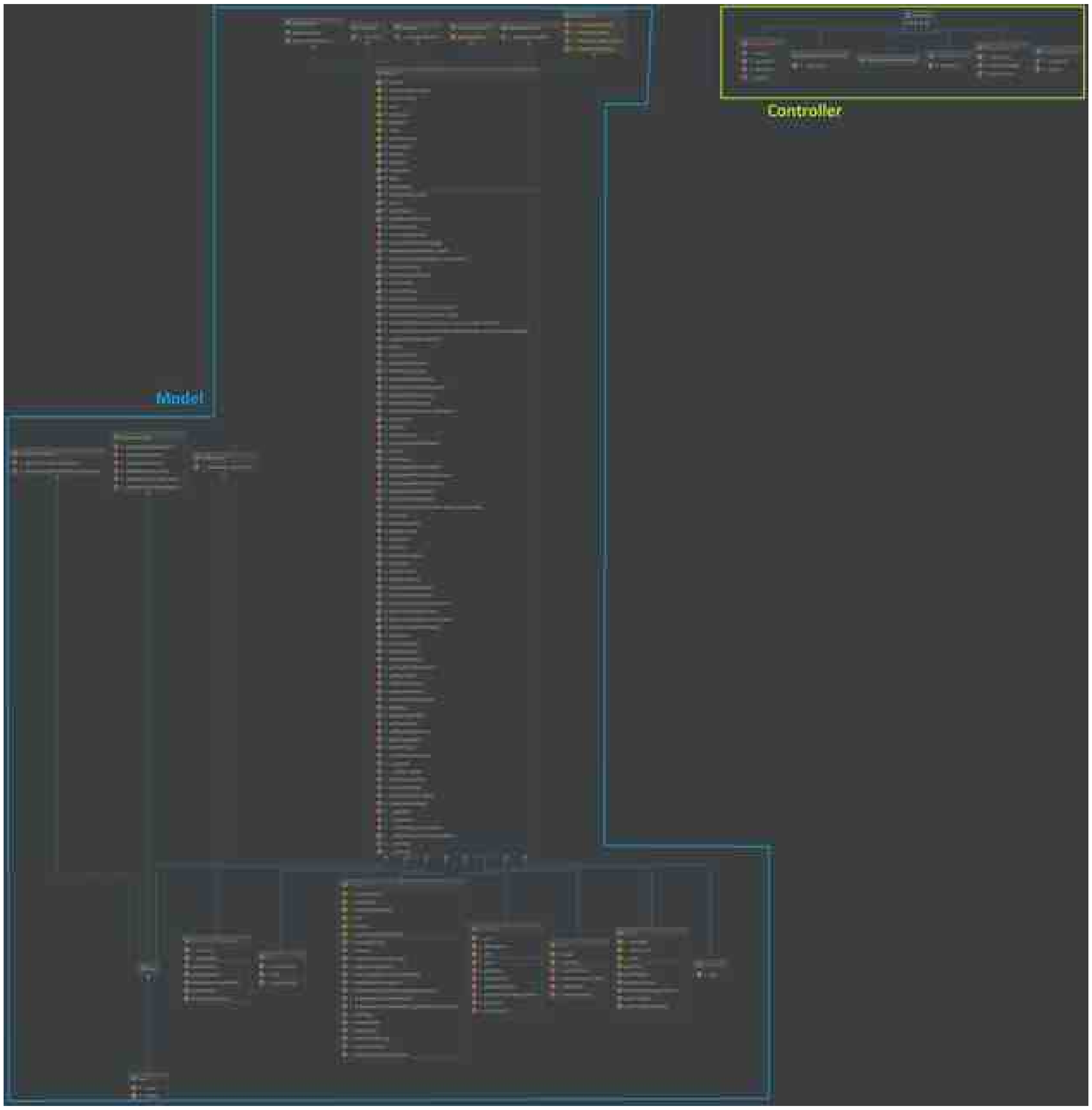
Midterm Summary | Mix It! (2017-12-15 12:15:33)
[...] Week No. 7 (Class Diagram) [...]

Final Post | Mix It! (2018-07-01 20:13:08)
[...] Week No. 7 [...]

Week No. 8 **(2017-11-25 10:54)**

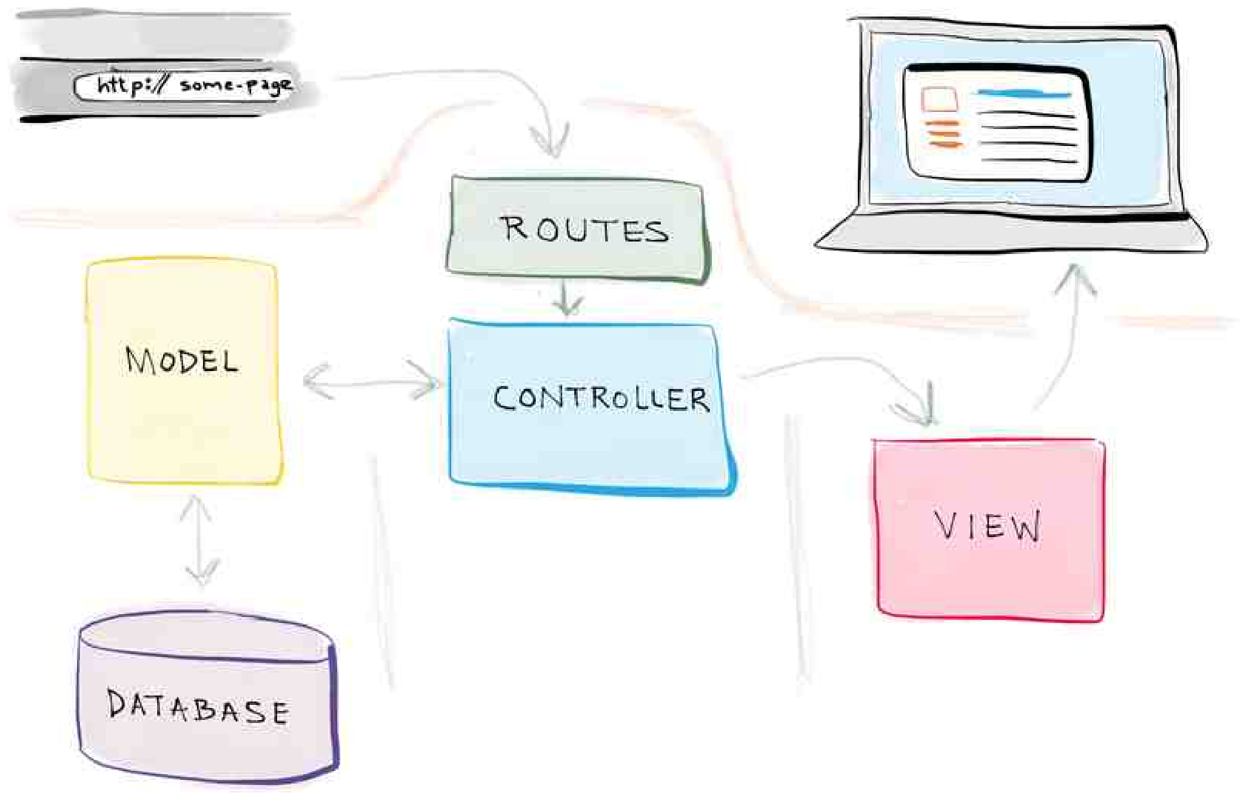
Hey togheter!

This week we updated our class-diagram and divided the classes into Model, View and Controller. As our views don't consist of classes they aren't contained in the diagram. Phpstorm adds only classes in the diagram.



[1]view on github

As we are using Laravel our MVC works like this:



(source: https://selftaughtcoders.com)

You can find both in our [2]Software Architeture Document.

Last but not least: We updated our scope. You can view our new scope [3]here or on our github repository

1. `https://github.com/Mit-It/Documentation/blob/master/Development/ClassDiagram.png`
2. `https://github.com/Mit-It/Documentation/blob/master/Documentation/SAD.md`
3. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/Use_case_diagram.png`

---

dennyfl (2017-11-30 07:59:48)
Hello Mix It! I really like your newest update, the pictures are good and your document has the right amount of details, to keep me informed but not bore me ;) I only found a few spelling mistakes: in chapter 3 structure without k; a diagram, no 'n' added. And chapter 9 diagram only has one m. Apart from that, great work. Keep up the good work Team dffc

superwomantinf16b4 (2017-11-30 09:00:56)
Hi Team Mix It, your Software Architecture Document looks very nice, but it is difficult to read your Class Diagramm. It is to small so we can´t understand how the classes are associated to each other and it is not possible to zoom it. It would be nice if you could make the picture maximised in your github. Best Regards, Team SuperWoman

Midterm Summary | Mix It! (2017-12-15 12:15:36)
[…] Week No. 8 (MVC) […]

Final Post | Mix It! (2018-07-01 20:13:11)
[…] Week No. 8 […]

1.3 December

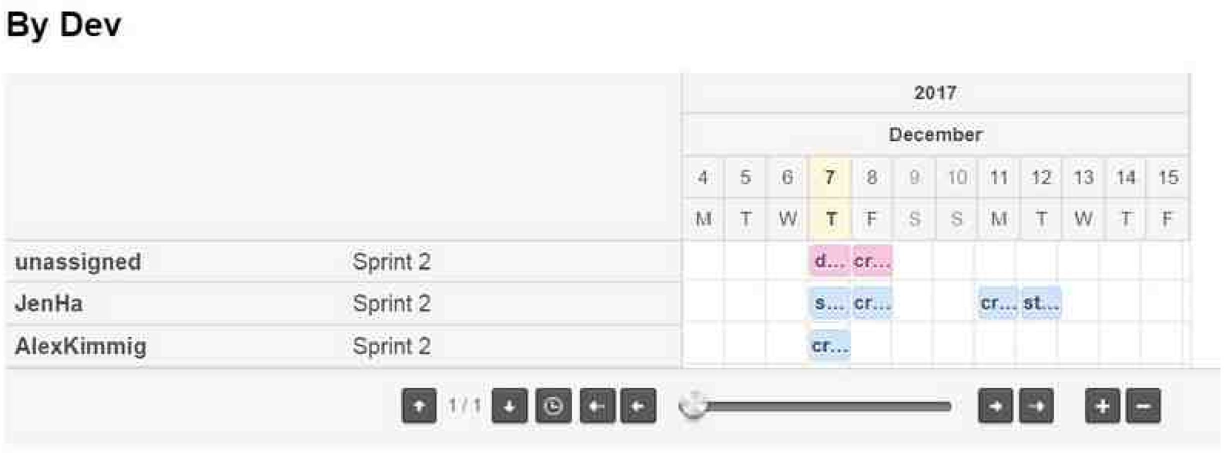
Week No.9 (2017-12-07 09:50)

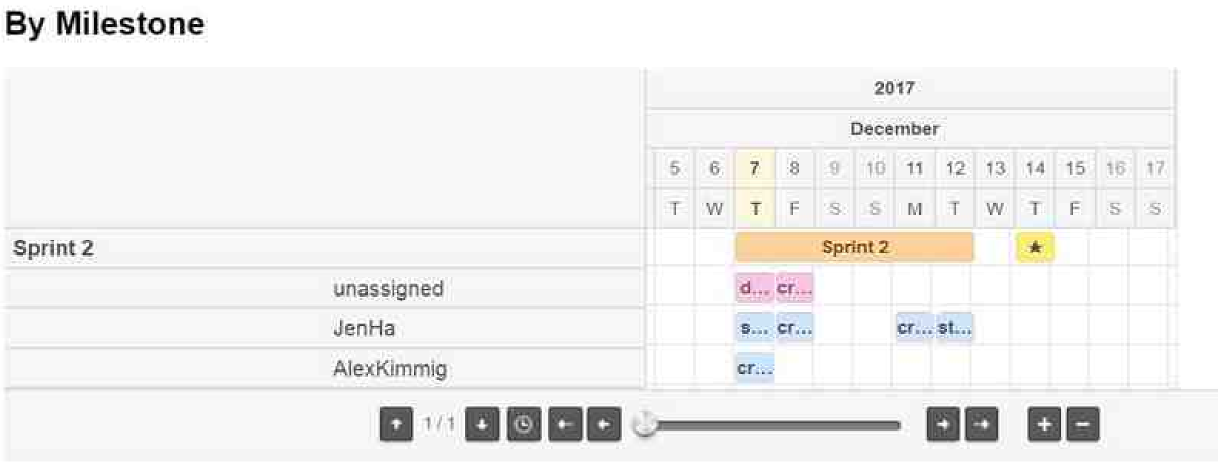
Hey together. This week we should create a gantt chart.

As we are using Zenhub for our projectmanagment, we had some Problems with this, because Zenhub doesn't offer any options to do this.

We found a tool for creating a guntt chart out of github issues, but we are still not satisfied with it. For some reason we haven't found out yet, it only show milestones that end in the future. It does not show milestones that has ended in the past.

Nevertheless we want to show you what it produced:





We are still looking for a better solution and will update our blog entry when we found one.

---

schmiefri (2017-12-07 10:38:25)
Hi there, once you figure out how to do the charts with github issues, would you be so kind and write a small summary of the tools and steps it took? Aksing from personal interest :D Kind Regards

Midterm Summary | Mix It! (2017-12-15 12:15:39)
[...] Week No. 9 (Gantt) [...]

Final Post | Mix It! (2018-07-01 20:13:14)
[...] Week No.9 [...]

Midterm Summary **(2017-12-12 12:27)**

Hey together.

This is the final Blogpost for this Semester before our Midterm Presentation.
In this post we want to summarize our first half of Software engineering.
For that we linked our documents and posts.

GitHub repos:

- [1]GitHub (Documentation)
- [2]GitHub (Code)

Zenhub:

- [3]Burndown Chart
- [4]Agile Board

Blogposts:

- [5]Week No. 1 (Introduction)
- [6]Week No. 2 (Team roles & Technologies)
- [7]Week No. 3 (SRS)
- [8]Week No. 4 (Use Cases)
- [9]Week No. 5 (Scrum)
- [10]Week No. 6 (Testing)
- [11]Week No. 7 (Class Diagram)
- [12]Week No. 8 (MVC)
- [13]Week No. 9 (Gantt)

Use-Cases:

- [14]Overall Diagramm
- [15]Use-Case View-Detail
- [16]Use-Case View-List
- [17]Use-Case Add-Recipe
- [18]Use-Case Delete-Own-Recipe
- [19]Use-Case Delete-Recipes
- [20]Use-Case Login
- [21]Use-Case Rate-Recipe
- [22]Use-Case Registrate

Architecture:

- [23]Class diagram
- [24]MVC

- [25]SAD

Demo:

- [26]Mix-It Homepage

Presentation:

- Presentation Slides

1. `https://github.com/Mit-It/Documentation`
2. `https://github.com/Mit-It/Mix-It`
3. `https://app.zenhub.com/workspace/o/mit-it/mix-it/reports?report=burndown&milestoneId=2948694&showPRs=false`
4. `https://app.zenhub.com/workspace/o/mit-it/mix-it/boards?repos=106662421`
5. `https://mixthatblog.wordpress.com/2017/10/12/week-no-1/`
6. `https://mixthatblog.wordpress.com/2017/10/16/week-no-2/`
7. `https://mixthatblog.wordpress.com/2017/10/22/week-no-3/`
8. `https://mixthatblog.wordpress.com/2017/10/28/week-no-4/`
9. `https://mixthatblog.wordpress.com/2017/11/04/week-no-5/`
10. `https://mixthatblog.wordpress.com/2017/11/12/week-no-6/`
11. `https://mixthatblog.wordpress.com/2017/11/22/week-no-7/`
12. `https://mixthatblog.wordpress.com/2017/11/25/week-no-8/`
13. `https://mixthatblog.wordpress.com/2017/12/07/week-no-9/`
14. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/Use_case_diagram.png`
15. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/view-detail.png`
16. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/view-list.png`
17. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/add_recipe.png`
18. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/delete_own_recipe.png`
19. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/delete_recipes.png`
20. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/login.png`
21. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/rate_recipe.png`
22. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/registrate.png`
23. `https://github.com/Mit-It/Documentation/blob/master/Development/ClassDiagram.png`
24. `https://github.com/Mit-It/Documentation/blob/master/Development/MVC-Laravel.png`
25. `https://github.com/Mit-It/Documentation/blob/master/Documentation/SAD.md`
26. `http://hauss.web-commerce.eu/`

Final Post | Mix It! (2018-07-01 20:13:17)
[…] Midterm Summary […]

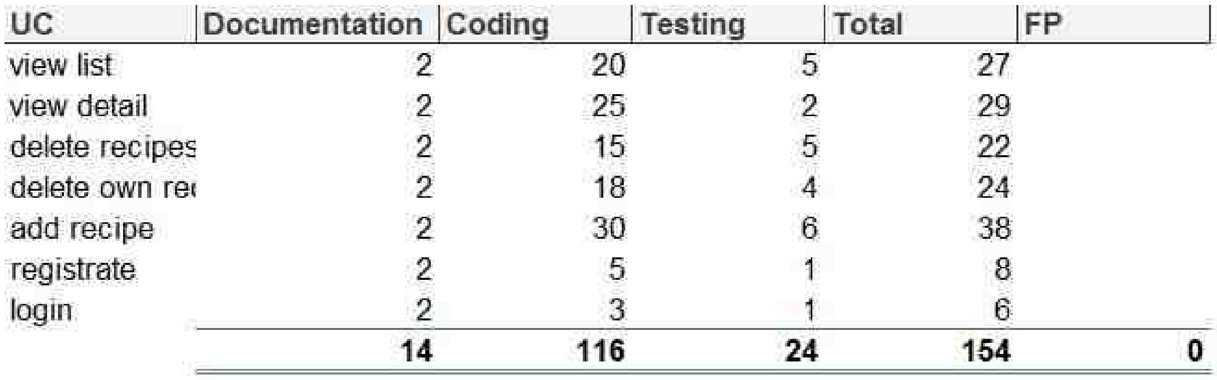
2. 2018

2.1 April

Risks, UC-Time and new Scope (2018-04-17 09:22)

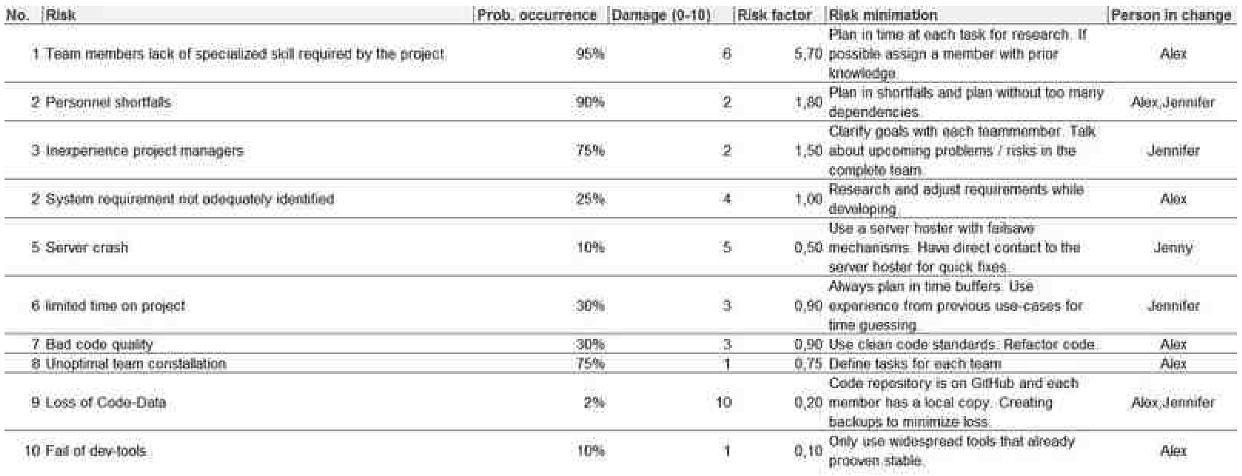
Hey together!

The next Semester startet, so here we are back.
This week we estimated the time for our already finished UC from last semsester. You can see the needed time in hours here:



UC	Documentation	Coding	Testing	Total	FP
view list	2	20	5	27	
view detail	2	25	2	29	
delete recipes	2	15	5	22	
delete own re	2	18	4	24	
add recipe	2	30	6	38	
registrate	2	5	1	8	
login	2	3	1	6	
	14	**116**	**24**	**154**	**0**

Also we analyzed what risk our projekt has, how much damage they could do and their possibility to happen. You can see the ten highest risks sorted by risk-factor here:



No.	Risk	Prob. occurrence	Damage (0-10)	Risk factor	Risk minimation	Person in change
1	Team members lack of specialized skill required by the project	95%	6	5,70	Plan in time at each task for research. If possible assign a member with prior knowledge.	Alex
2	Personnel shortfalls	90%	2	1,80	Plan in shortfalls and plan without too many dependencies.	Alex,Jennifer
3	Inexperience project managers	75%	2	1,50	Clarify goals with each teammember. Talk about upcoming problems / risks in the complete team.	Jennifer
2	System requirement not adequately identified	25%	4	1,00	Research and adjust requirements while developing.	Alex
5	Server crash	10%	5	0,50	Use a server hoster with failsave mechanisms. Have direct contact to the server hoster for quick fixes.	Jenny
6	limited time on project	30%	3	0,90	Always plan in time buffers. Use experience from previous use-cases for time guessing.	Jennifer
7	Bad code quality	30%	3	0,90	Use clean code standards. Refactor code.	Alex
8	Unoptimal team constallation	75%	1	0,75	Define tasks for each team	Alex
9	Loss of Code-Data	2%	10	0,20	Code repository is on GitHub and each member has a local copy. Creating backups to minimize loss.	Alex,Jennifer
10	Fail of dev-tools	10%	1	0,10	Only use widespread tools that already prooven stable.	Alex

Also we updated our scope. You can find it on [1]github.

1. `https://github.com/Mit-It/Documentation/blob/master/Use%20Cases/Use_case_diagram.png`

---

anaschwed (2018-04-25 10:58:20)
Hey team MixThat! We think your time estimation table provides an overview of your progress in different categories and has a good structure. The risk table looks pretty good as well: the risks are described in a clear way, sorted by the risk factor and so on. But to make your risk table perfect, you could add more columns such as risk mitigation and person in charge. Probably, you could also change the scale for damage (according to the grading criteria "risk impact can be a number from 1 (negligible) -10 (large)"). Kind regards, Team VSS

Jennifer Hauß (2018-04-25 11:09:08)
Hey! Thanks for your review. We edited the points you mentionted and updated our blockentry Best regards Team Mix-It!

Final Post | Mix It! (2018-07-01 20:13:19)
[...] Risks, UC-Time and new Scope [...]

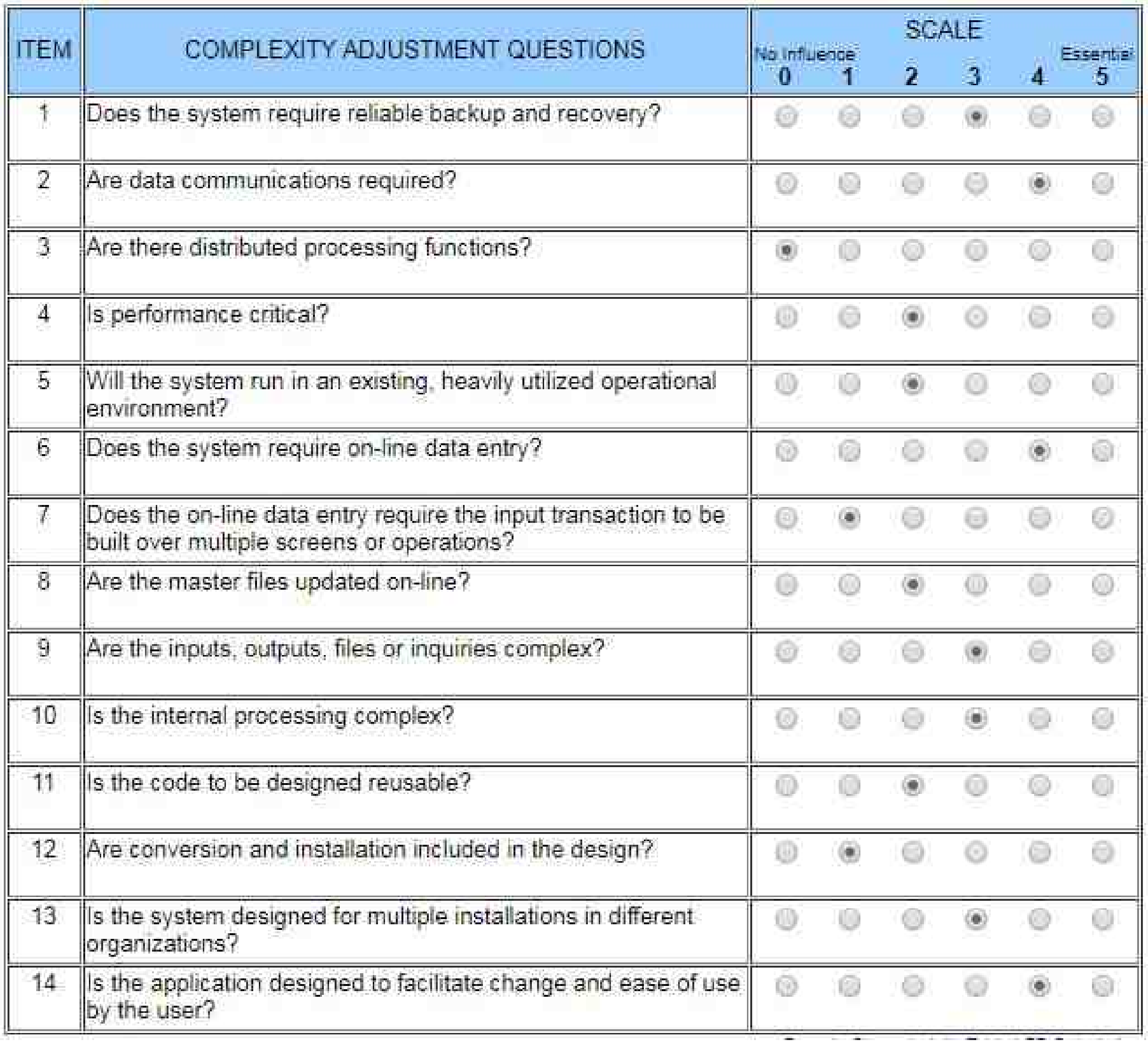
Function Points (2018-04-29 09:17)

Hey together!

This week we dealed with function points.

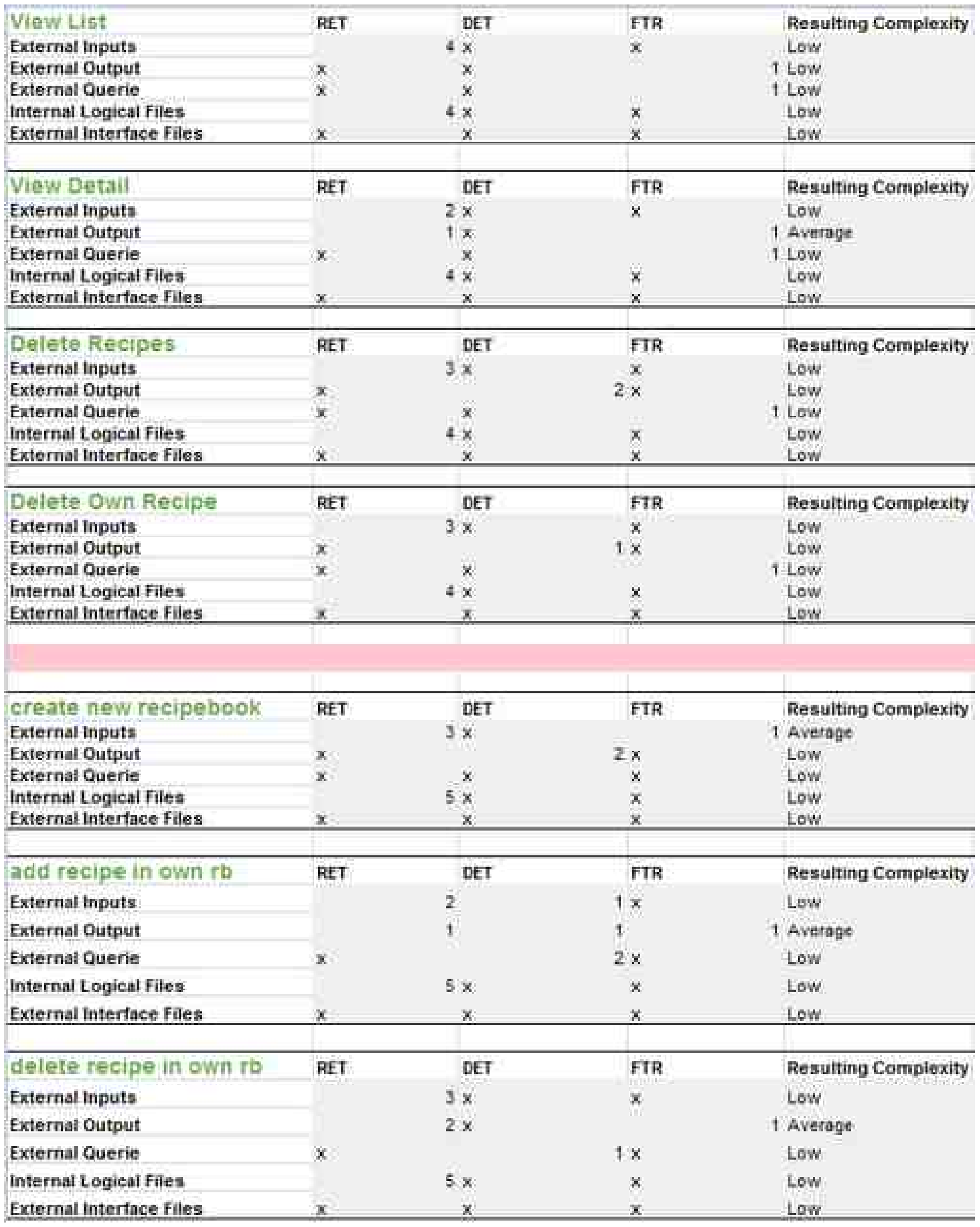
With help of [1]this tool we estimated the function points for four of our finished use cases and three of the to-be-finished use cases.

For our whole project we used this configuration in the tool above:



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?	○	○	○	○	●	○

We estimated the function points for the single use cases like this:



View List	RET	DET	FTR	Resulting Complexity
External Inputs	4	x	x	Low
External Output	x	x	1	Low
External Querie	x	x	1	Low
Internal Logical Files	4	x	x	Low
External Interface Files	x	x	x	Low

View Detail	RET	DET	FTR	Resulting Complexity
External Inputs	2	x	x	Low
External Output	1	x	1	Average
External Querie	x	x	1	Low
Internal Logical Files	4	x	x	Low
External Interface Files	x	x	x	Low

Delete Recipes	RET	DET	FTR	Resulting Complexity
External Inputs	3	x	x	Low
External Output	x	2	x	Low
External Querie	x	x	1	Low
Internal Logical Files	4	x	x	Low
External Interface Files	x	x	x	Low

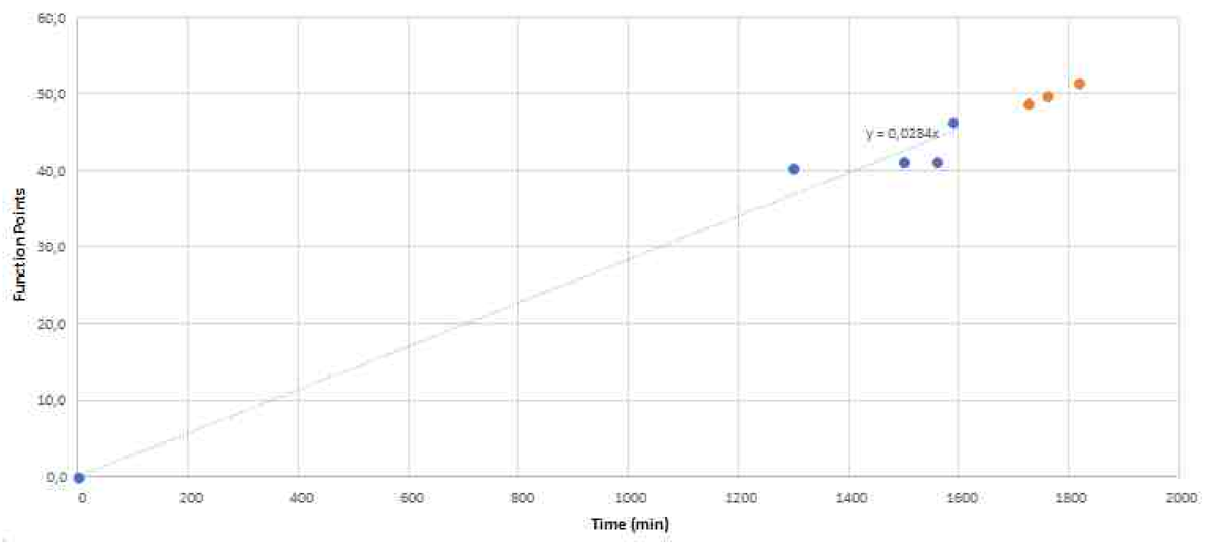
Delete Own Recipe	RET	DET	FTR	Resulting Complexity
External Inputs	3	x	x	Low
External Output	x	1	x	Low
External Querie	x	x	1	Low
Internal Logical Files	4	x	x	Low
External Interface Files	x	x	x	Low

create new recipebook	RET	DET	FTR	Resulting Complexity
External Inputs	3	x	1	Average
External Output	x	2	x	Low
External Querie	x	x	x	Low
Internal Logical Files	5	x	x	Low
External Interface Files	x	x	x	Low

add recipe in own rb	RET	DET	FTR	Resulting Complexity
External Inputs	2	1	x	Low
External Output	1	1	1	Average
External Querie	x	2	x	Low
Internal Logical Files	5	x	x	Low
External Interface Files	x	x	x	Low

delete recipe in own rb	RET	DET	FTR	Resulting Complexity
External Inputs	3	x	x	Low
External Output	2	x	1	Average
External Querie	x	1	x	Low
Internal Logical Files	5	x	x	Low
External Interface Files	x	x	x	Low

And from this calculation we made this diagram:



1. `http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP_Calc.html`

---

necoproject (2018-05-02 09:28:44)
Hey team mixit, could you show your complexity table, so we see how you judged your project? Also you could add the spreadsheet of the function point calculation in your blog post. Best regards, NeCo

thomaspoetzsch (2018-05-02 10:10:05)
Hey there, Your diagram seems fine, but we would love to see the Complexity Adjustment table you used with the tool. It would also be interesting to see some sort of spreadsheet with the underlying data. Best regards, Team poest

Final Post | Mix It! (2018-07-01 20:12:47)
[...] Blog post [...]

2.2 May

Unit Testing (2018-05-16 09:44)

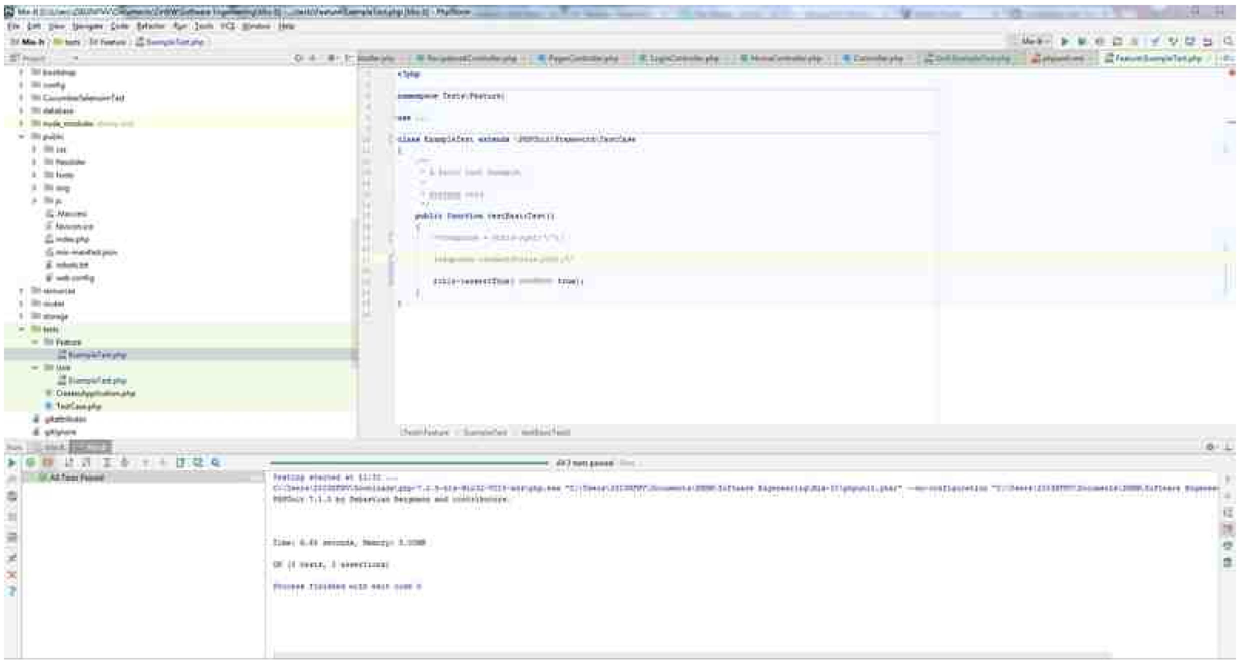
Hello together!

This week we worked on unit testing. By testing our code on each commit we are ensuring that our code has stable quality. Bugs are easily found and can quickly be corrected.

Testing-Framework

As we are using the Laravel framework for our application, we are already including PHPUnit for unit tests. They can be found [1]here

However we don't use any tests right now, as the controllers do not implement any logic that could be tested at the moment. Once more logic is implemented we already got Unit testing and CI-Process set up. Here is a screenshot to show you that the Unit tests are running.



```
<?php

namespace Tests\Feature;

use ...

class ExampleTest extends \PHPUnit\Framework\TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testBasicTest()
    {
        //$response = $this->get('/');

        //$response->assertStatus(200);

        $this->assertTrue( condition: true);
    }
}
```

```
Testing started at 11:11 ...
[illegible]
PHPUnit 7.1.5 by Sebastian Bergmann and contributors.



Time: [illegible] seconds, Memory: [illegible]

OK (2 tests, 2 assertions)

Process finished with exit code 0
```

We are using [2]TravisCI to automaticly build our application. TravisCI doesn't require the setup of any CI-Infrastructure and can be easily bound to a Git-Repository. It detects new commits and runs our test-setup described in our [3].travis.yml file.

Check out our [4]badge and [5]test-plan ;)

1. `https://github.com/Mit-It/Mix-It/tree/master/tests`
2. `https://travis-ci.org/`
3. `https://github.com/Mit-It/Mix-It/blob/master/.travis.yml`
4. `https://github.com/Mit-It/Documentation/blob/master/README.md`
5. `https://github.com/Mit-It/Documentation/blob/master/Test-Plan.md`

---

dffcblog (2018-05-16 10:21:07)
Hello Team MixIt, I took a look on your blog entry and it looks really nice! You already got a badge and you are into Travis already so you have more time in week 7. But you should add the Testplan to your blog entry and add a screenshot of your

running tests ;) Best regards, Florian from Team DFFC

Alexander Kimmig (2018-05-16 10:42:37)
Hey there, Thank you for your reply, the test plan is now linked. However we don’t have logic to test for our unit tests as mentionedr. I will update the post as soon as we have logic to test :) Greedings Alex

Final Post | Mix It! (2018-07-01 20:13:22)
[...] Unit Testing [...]

Refactoring (2018-05-20 18:07)

Hey together!

This week we worked trough the first chapter of a book about Refactoring from Martin Fowler. To practise what we read, we refactored a little program from him. You can see the refactored code including all commits on our github repositorys:

[1]Jennifer

[2]Alexander

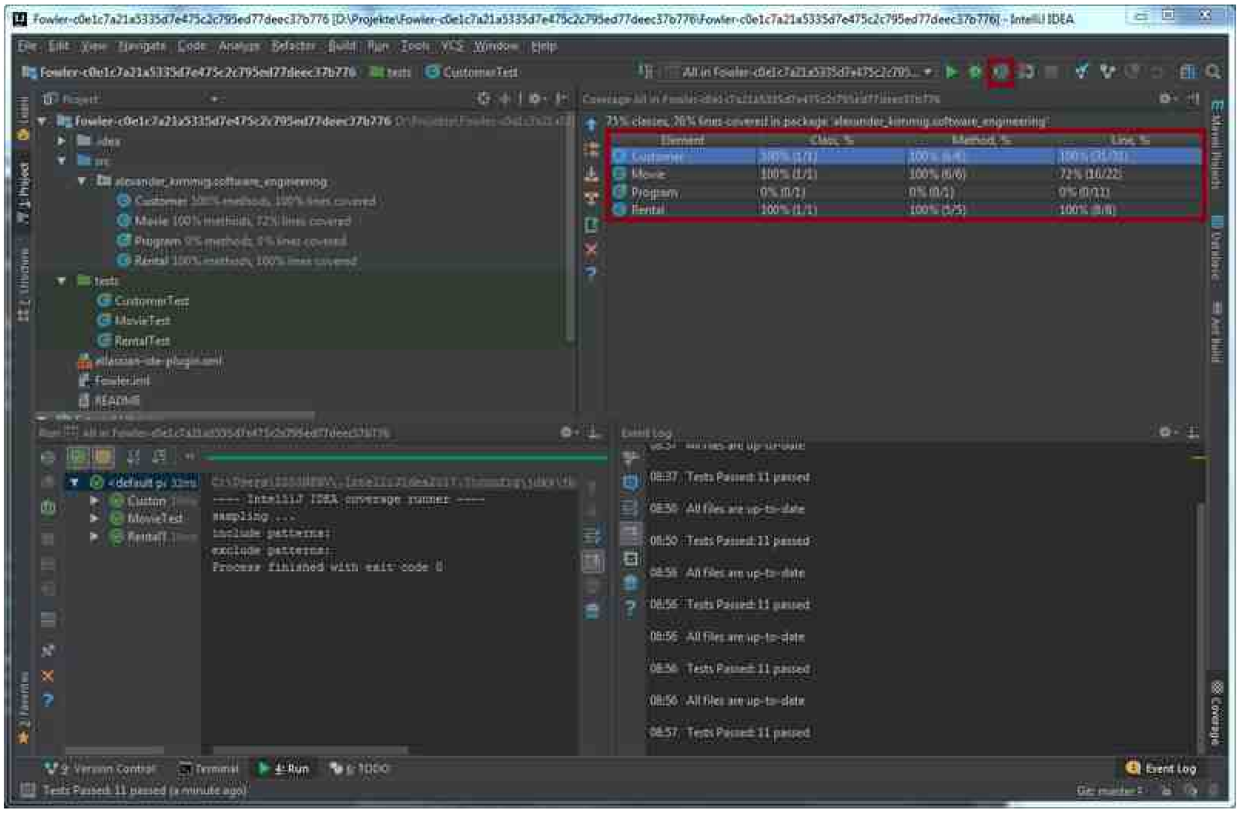
Useful IDE features

The intelliJ IDE provides syntax highlighting for unused code or imports. You can then quickly delete those with a shortcut.

Renaming a variable, class, method or package is also greatly supported by the IDE. Using the "refactor" feature the IDE will refactor all the instances which makes sure that you don’t forget renaming it in any place.

As far as Unit-Tests go, you can easily compare the different test-outputs when they fail to track down the error.

The IDE also provides a feature to run the tests with coverage. Returning the values for Class-, Method- and Line-coverage.



75% classes, 76% lines covered in package 'alexander_kimmig.software_engineering'

Element	Class, %	Method, %	Line, %
Customer	100% (1/1)	100% (6/6)	100% (35/35)
Movie	100% (1/1)	100% (6/6)	72% (16/22)
Program	0% (0/1)	0% (0/1)	0% (0/11)
Rental	100% (1/1)	100% (5/5)	100% (8/8)

```
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
exclude patterns:
Process finished with exit code 0
```

```
08:37 Tests Passed: 11 passed
08:50 All files are up-to-date
08:50 Tests Passed: 11 passed
08:56 All files are up-to-date
08:56 Tests Passed: 11 passed
08:56 All files are up-to-date
08:56 Tests Passed: 11 passed
08:56 All files are up-to-date
08:57 Tests Passed: 11 passed
```

1. `https://github.com/JenHa/RefactoringPractise/commits/master`
2. `https://github.com/AlexKimmig/fowler_refactoring/commits/master`

---

necoproject (2018-05-23 08:36:40)
Hey, your repos look good, but I think adding one or two pictures why and how your IDE helped you might be good. Other than that, there is nothing to add from out side. Greetings, NeCo

Alexander Kimmig (2018-05-23 08:54:31)
Hey there, thank you for your comment. We added a picture of the test coverage feature with the important outputs marked in red. Greetings Mix-It

superwomantinf16b4 (2018-05-23 09:00:10)
Hi team MixThatBlog, you did a great job! I can see that you understood the principles of 'Refactoring' and why it is worth to use it in your project. I really like that you also added a paragraph about the features of your IDE intelliJ, it is always good to know how your IDE can help you with your work. Best regards, Team SuperWoman

Alexander Kimmig (2018-05-23 09:04:36)
Hey, thank you for your comment. I'm happy to see that you like our blog post. I can agree that it really is helpful to know about the features that the IDE provides. When you use them it can really save some time and nerves :) Greetings, Mix-It

Final Post | Mix It! (2018-07-01 20:13:25)
[...] Refactoring [...]

2.3 June

Patterns (2018-06-05 10:43)

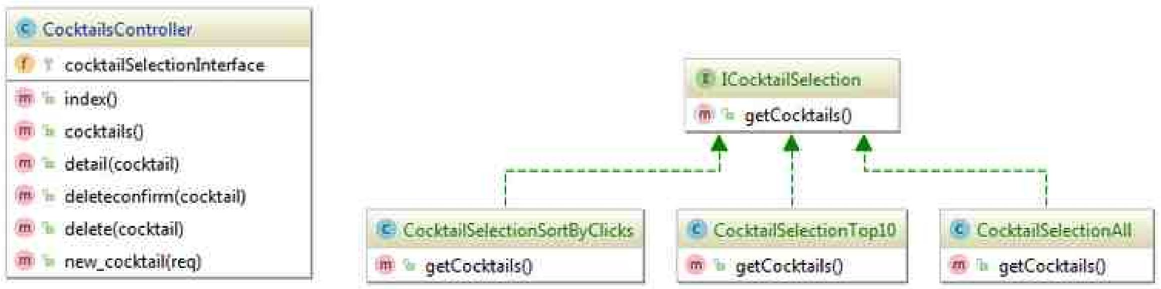
Hey together!

This week we worked through different software patterns. If you are curious about it you can read more about this subject [1]here. Our goal was to decide on a useful pattern and then intregrate it into our project.

We decided on using the behavioral design pattern "Strategy". Our website provides a wide veriety of situations were similar classes only differ in their behavior. An example is the implementation of different cocktail-listings. One class provides an lexicographic listing and another one e.g. a listing by rating.

In the project we pattern firstly to redo our cocktail listing. Therefor we added an interface for our cocktail-Controller, that provides access to the different listing strategies. With the interface the controller is able to use a strategy without ever changing any code of the controller again. The strategies need to inherit the interface and implement the inherited method.

Here the class diagram:



To summerize it, the strategy pattern is very handy for project that thend to implement exchangeable classes / methods. The class that is using those does not need to be changed anymore after the strategy is implemented and therefore is less likely to get broken. The implementation of the strategy itself is also relativley quickly done.

1. `https://www.oodesign.com/`

Final Post | Mix It! (2018-07-01 20:13:28)
[...] Patterns [...]

Metrics (2018-06-06 10:17)

Hey there,

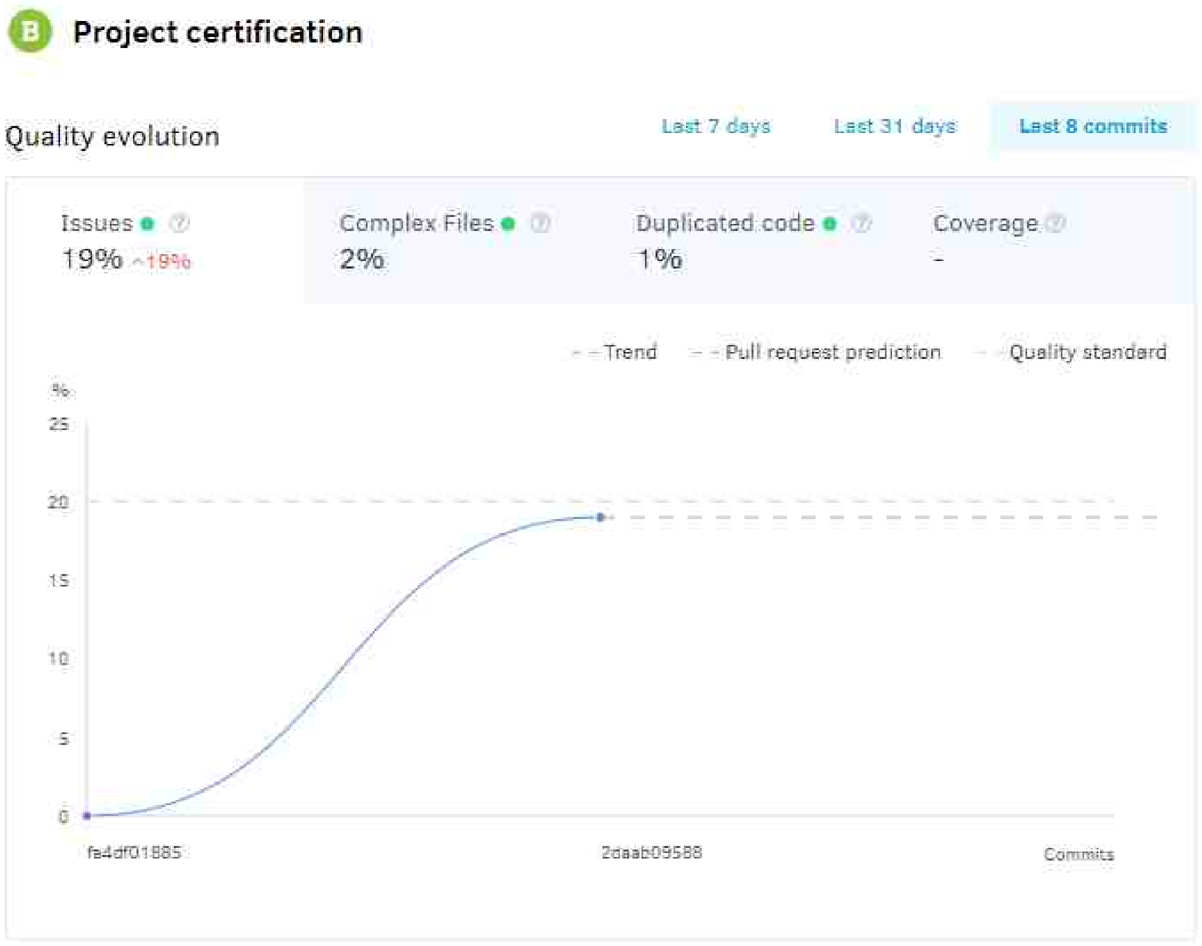
this week we were working on getting metrics for our project. The metrics are used to get an overview on the code style.

We decides on using Sonarqube and Codacy for evaluating our metrics. The Sonarqube config can be found [1]here. Codacy automatially checks the code when we commit to our git repository. It also provides an option to include test coverage. However the test coverage does not recognize our cucumber tests.
It gives us several metrices for

- Issues
- Complex Files
- Duplicated code
- Code quality analysis
- Security analysis

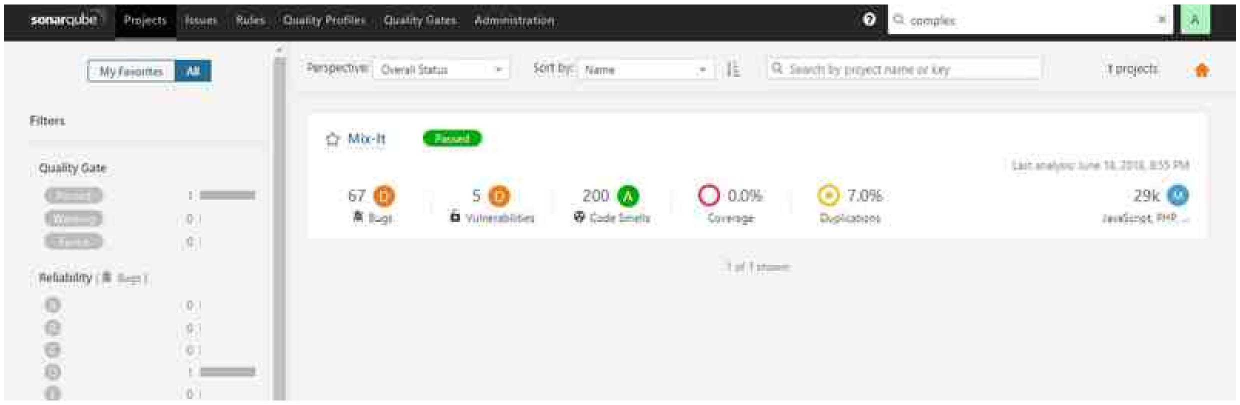
Here is our code metrics before refactoring:

Codacy:



Issues	Complex Files	Duplicated code	Coverage
19% ^19%	2%	1%	-

Sonarqube:



Mix-It Passed

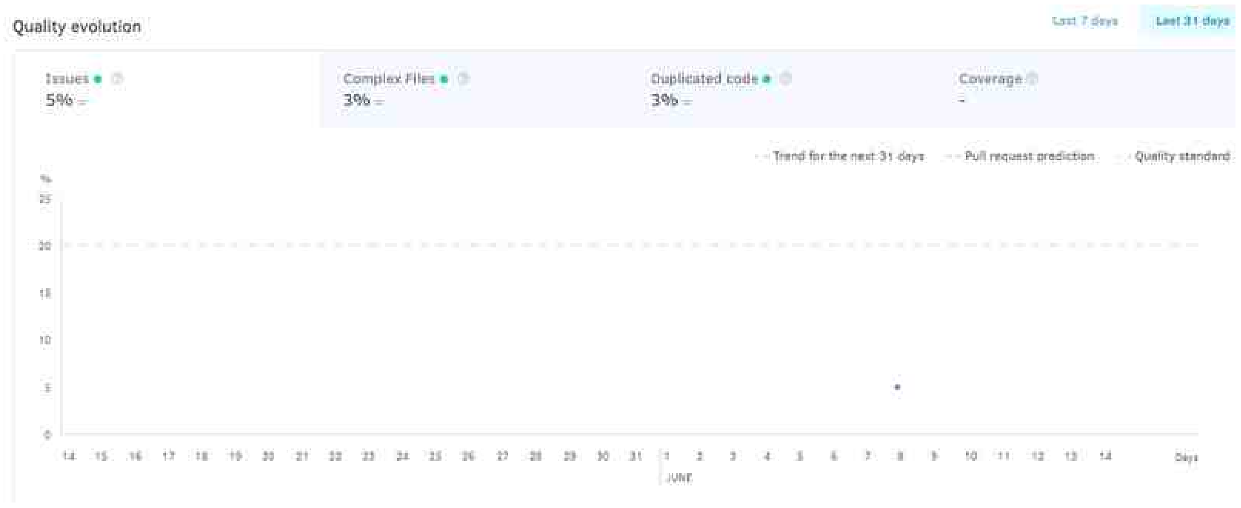
Last analysis: June 14, 2018, 8:55 PM

67 D	5 D	200 A	0.0%	7.0%	29k M
Bugs	Vulnerabilities	Code Smells	Coverage	Duplications	JavaScript, PHP, ...

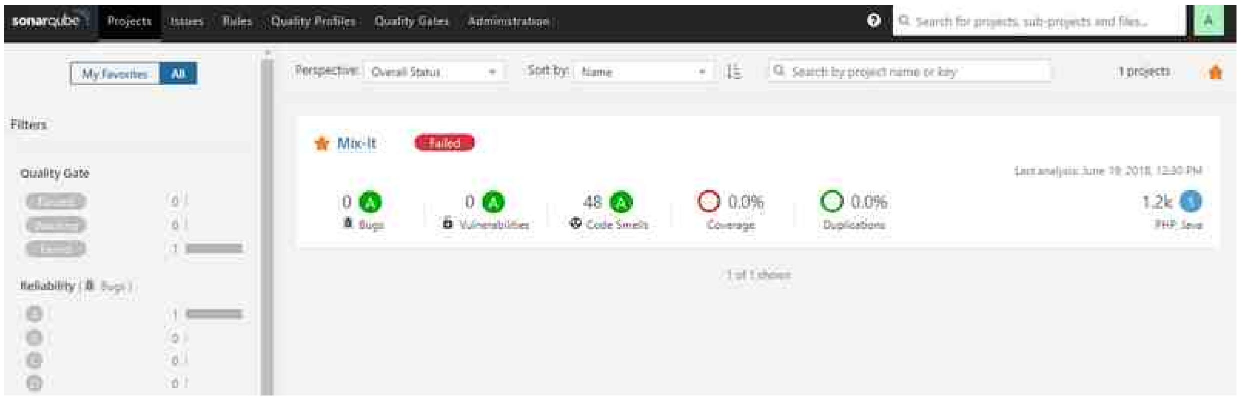
1 of 1 shown

and here after:

codacy:

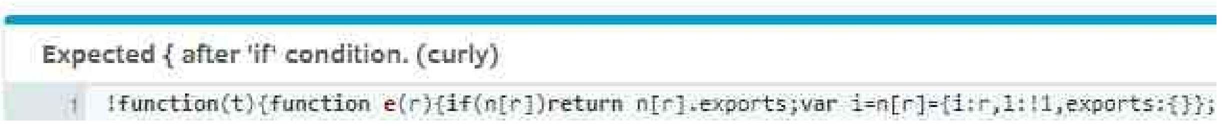


sonarqube:



As you can see we got down to 5 % issues with codacy.

As you can see the the Issues before refactoring is at 19 % the number in red represents the "Comparison of the number of issues with the industry average". Codacy also lets you exactly see where you issues are and why there are there.



Expected { after 'if' condition. (curly)

```
!function(t){function e(r){if(n[r])return n[r].exports;var i=n[r]={i:r,l:!1,exports:{}};
```

Metrics that Sonarqube offers

Cognitive Complexity

Cognitive Complexity is described as a "new measurement of how hard code is to understand – one that strikes developers as intuitively right."

So it descibes how difficult it it, to understand a unit of code, to comprehend and read it.
Developers then can see that they should maybe split it into further logical units and redo some of the code until the score fits.

Code Duplication

Duplicated code can lead to software that is hard to understand and difficult to change. The Don't Repeat Yourself (DRY) principle states:

When you violate DRY, bugs and maintenance problems are sure to follow. Duplicated code has a tendency to both continue to replicate and also to diverge (leaving bugs as two similar implementations differ in subtle ways).

Thank you for reading our post.

1. `https://github.com/Mit-It/Mix-It/blob/master/sonar-project.properties`

---

anaschwed (2018-06-06 10:32:01)
Deat team MixThat, we think your refactoring and code improvements are just great. However, codacy and other tools are not perfect, so we wonder if there are issues you didn't want to change although codacy suggested to do so. Kind regards Team VSS

Alexander Kimmig (2018-06-06 10:39:22)
Hey team VSS, Thank you for your comment. Codacy seems to get SQL-Qeries wrong. But after knowing about the function to ignore files, almost every issue seemed reasonable for us. Greetings, Team Mix it

thomaspoetzsch (2018-06-06 10:37:07)
Hey there, We liked your blog article, but we will check back to see the improvements once you merged them into the master branch ;-) . There are also a lot of IDE plugins to better ones coding style. Have you thought about using them? Do you think the tips Codacy gave you were useful and applicable? Best regards, Team PPR

Alexander Kimmig (2018-06-06 10:42:11)
Hey there, thanks for your comment. We did not really look into IDE plugins yet, but it seems like a good idea and i think we will be starting on using them. As far as codacy goes, i really think that most of the tips it gave seemed reasonable. Greedings, Team Mix it

Final Post | Mix It! (2018-07-01 20:13:30)
[...] Metrics [...]

Installation manual **(2018-06-14 10:44)**

Hey togheter!

This week we want to show you, how you can install our Application on your own computer.

We created a [1]github repository for the installation files.

So here is our installation manual:

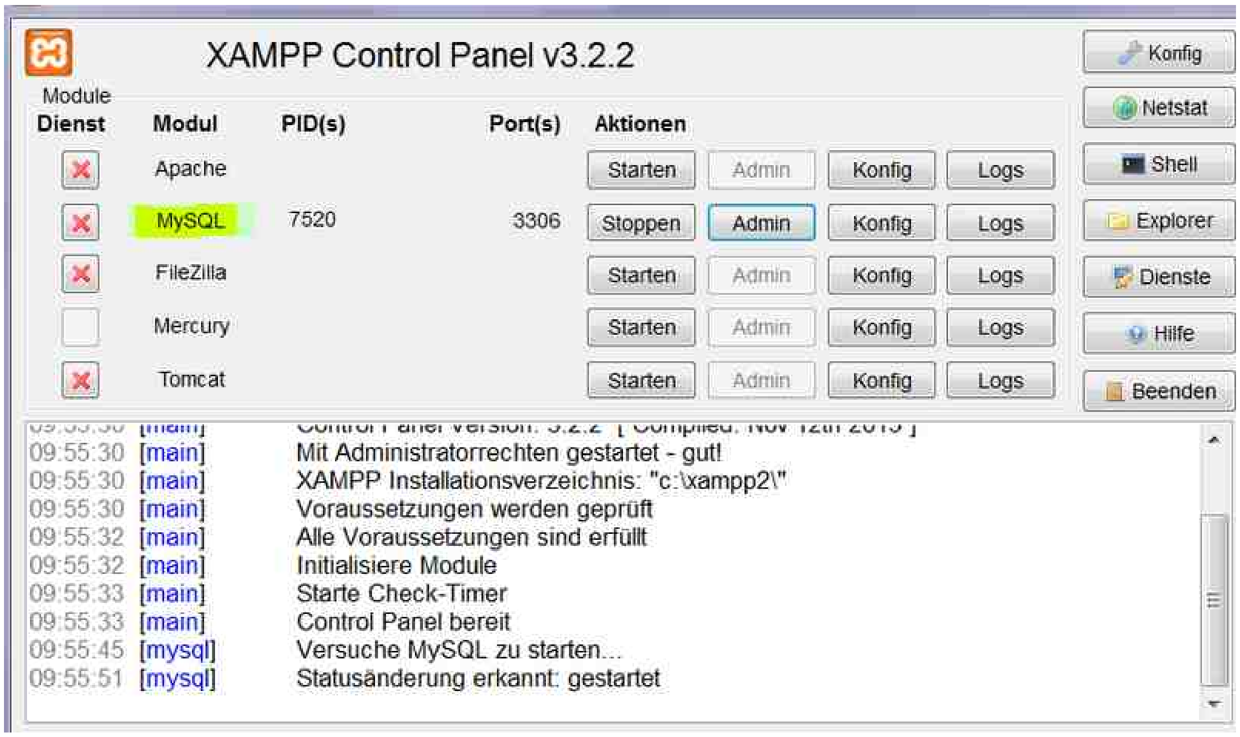
Local:

You need php and mysql on your computer.

Step 1: Download the files from our repository

Step2: Install Xamp with mysql if you haven’t already

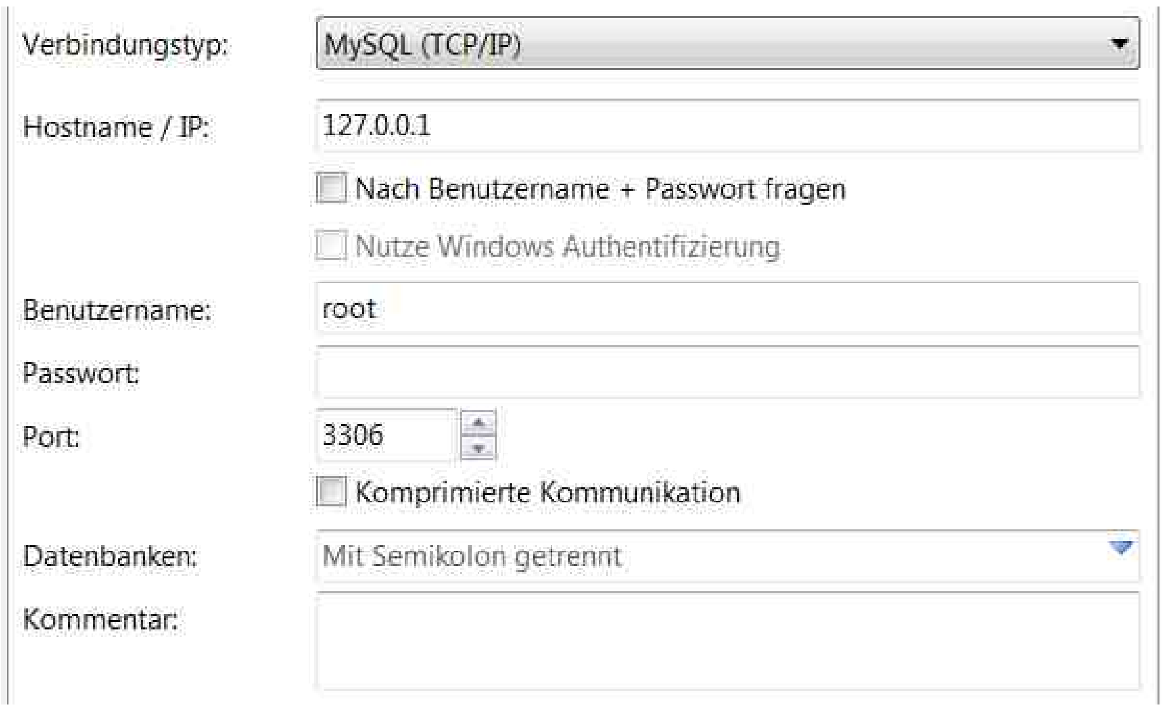
Step3: Start Xamp and start mysql. If it is the first time you have to configure it first. Choose “root” as user and no password (otherwise you have to change the .env file and the bootstrap/cache/config.php and change username and password on each position)



Dienst	Modul	PID(s)	Port(s)	Aktionen			
✗	Apache			Starten	Admin	Konfig	Logs
✗	MySQL	7520	3306	Stoppen	Admin	Konfig	Logs
✗	FileZilla			Starten	Admin	Konfig	Logs
	Mercury			Starten	Admin	Konfig	Logs
✗	Tomcat			Starten	Admin	Konfig	Logs

```
09:55:30  [main]     Mit Administratorrechten gestartet - gut!
09:55:30  [main]     XAMPP Installationsverzeichnis: "c:\xampp2\"
09:55:30  [main]     Voraussetzungen werden geprüft
09:55:32  [main]     Alle Voraussetzungen sind erfüllt
09:55:32  [main]     Initialisiere Module
09:55:33  [main]     Starte Check-Timer
09:55:33  [main]     Control Panel bereit
09:55:45  [mysql]    Versuche MySQL zu starten...
09:55:51  [mysql]    Statusänderung erkannt: gestartet
```

Step4: Start a database-management tool like HeidiSQL or MySQL workbench and open your local mysql



Verbindungstyp:	MySQL (TCP/IP)
Hostname / IP:	127.0.0.1
	☐ Nach Benutzername + Passwort fragen
	☐ Nutze Windows Authentifizierung
Benutzername:	root
Passwort:	
Port:	3306
	☐ Komprimierte Kommunikation
Datenbanken:	Mit Semikolon getrennt
Kommentar:	

(screenshot from HeidiSQL)

Step 5: Create a new database named “test” (otherwise you have to change the files like in Step 3 mentioned), import the mix.sql and let it run on the “test”-database

Step6: Open your command line (cmd), navigate to the “Mix-It”-folder and type: “php artisan serve”

Step7: Now you can open your personal mix-it website on http://127.0.0.1:8000 , admin-email is: [2]admin@mail.com and password is: supersecret

Step8: Enjoy your very own cocktail-website

You can also find our installation manual on [3]github

1. `https://github.com/Mit-It/Installation`
2. `mailto:admin@mail.com`
3. `https://github.com/Mit-It/Installation`

Team VSS (2018-06-19 12:22:23)
Works! Thank you for the great instructions! https://imgur.com/a57mlaY Team VSS

Alexander Kimmig (2018-06-19 12:54:56)
Thank you for installing it :) Greeding team Mix-It

Final Post | Mix It! (2018-07-01 20:13:33)
[...] Installation manual [...]

BlogBook v1.0,
$\LaTeX\, 2_\varepsilon$ & GNU/Linux.
`https://www.blogbooker.com`

Edited: July 2, 2018