



Free version: Low quality pictures

[trackyourfit.wordpress.com](http://trackyourfit.wordpress.com)



# Contents

<b>1</b>	<b>2019</b>	<b>5</b>
1.1	October . . . . .	5
	Week 1: Track Your Fit (2019-10-08 10:35) . . . . .	5
	Week 2: Roles and Technologies (2019-10-10 20:01) . . . . .	6
	Week 3: SRS and UCD (2019-10-22 11:24) . . . . .	8
	Week 4: Use Case Specifications (2019-10-27 12:34) . . . . .	10
1.2	November . . . . .	11
	Week 5: .feature files (2019-11-05 18:12) . . . . .	11
	Week 6: PM tool + Integration (2019-11-12 14:40) . . . . .	16
	Week 7: Retrospective (2019-11-16 14:26) . . . . .	18
	Week 8: Class diagrams & DB schema (2019-11-23 12:54) . . . . .	21
1.3	December . . . . .	26
	Week 9: MVC & redefined scope (2019-12-02 14:49) . . . . .	26
	Week 11: Midterm Presentation (2019-12-10 19:04) . . . . .	30
<b>2</b>	<b>2020</b>	<b>35</b>
2.1	April . . . . .	35
	Week 11: Redefining Scope (2020-04-22 11:12) . . . . .	35
	Week 12: Risk management (2020-04-26 17:57) . . . . .	37
2.2	May . . . . .	38
	Week 13: Function Points (2020-05-02 13:33) . . . . .	38
	Week 14: Unit Testing (2020-05-09 17:08) . . . . .	40
	Week 15: Refactoring (2020-05-16 16:40) . . . . .	43
	Week 16: Design Patterns (2020-05-24 13:08) . . . . .	44
2.3	June . . . . .	50
	Week 17: Code Metrics (2020-06-04 11:33) . . . . .	50
	Week 19: Installation (2020-06-23 18:13) . . . . .	55



# 1. 2019

## 1.1 October

### Week 1: Track Your Fit (2019-10-08 10:35)

Manage your training plans, communicate with and support other Trackers on their training.



Our project is supposed to help sport enthusiasts and those who want to get sporty and active. So you will be able to:

- create, execute and track training plans
- compare your plans with other Trackers
- rate other training plans and give feedback to improve them

This blog will be used to track our progress in developing the application. We are happy to receive any kinds of feedback, tips and suggestions for future and existing features.

To start of, we are going to develop a webapplication. In the future we would like to turn the webapp to a mobile app. The reason for using these types of applications is, that you should be able to use the app wherever you are (as long as you have got an internet connection).

The Team:

Who are "we"? We are four computer science students from the course TINF18B3.

Janis Fix, Tobias Bühler, Toni Einecker & Johannes Zipfel

Stay tuned...

TheLord (2019-10-09 08:30:10)

Hello, we really like your Project. But you have to keep the App motivating for the User. As a Hint you can think of a Reminder, that reminds you about you missing some Training hours or you didn't train for xx amount of days. You can also think about preset Trainingplans to get in more easily. Best regards, Tim, Anna, Elia

TrackYourFit (2019-10-09 09:15:20)

Thanks for your feedback! We'll definitely work on some kind of feature to keep the user motivated. Your hint was one of our ideas to ensure that aswell!

hypercasualgame (2019-10-09 08:40:39)

Hey, this is a short, well written description of what you're up to. I'd rewrite the paragraph describing your motivation ("To start of, we are going to develop a[...]"). Maybe shorten it or write more about it in a different context. Greetings!

TrackYourFit (2019-10-09 09:15:57)

Thank you! We'll adjust that :)

logicgametinf18b3 (2019-10-09 09:21:31)

Hello there! Your blog is clearly sturctured and I like the way you introduced your project. But I have some questions: 1. Are you going to seperate every blog entry on a single website on your blog or are you going to make it like a long Text? We seperated the introduction of our team and our project to several sites which can be accessed from our frontpage. 2. I really like your logo. How did you create it? Is it a Stockphoto? Best Regards Marcel

TrackYourFit (2019-10-09 09:41:37)

Hey, Seperating the introduction of our team seems like a good idea! We will probably have this post stick on top and then have the rest of the entries below this, with a seperate link to them. We created our logo with an online tool <https://www.logomaker.com/de/> pretty easy to use, nothing special but works fine :D

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:30)

[...] Week 1: Track Your Fit [...]

## **Week 2: Roles and Technologies (2019-10-10 20:01)**

Hello guys,

we established the team roles and the technologies we prefer to use during the project.

Let's have a look into the Roles, they are inspired by the RUP teminology:

Implementation Janis Fix, Tobias Bühler, Toni Einecker Johannes Zipfel

Requirements Janis Fix, Tobias Bühler, Toni Einecker Johannes Zipfel

Test-Manager Johannes Zipfel

Deployment-Manager Tobias Bühler

Product-Manager Janis Fix

Design/UX Janis Fix, Tobias Bühler, Toni Einecker Johannes Zipfel  
Configuration-Manager Toni Einecker

And here you can see our technologies we've chosen:

App Xamarin / Flutter, XML (for design)  
Web ASP .Net Core / Entity Framework Core, HTML, JS, CSS  
Programming Languages C #, SQL  
Database MySQL  
Project-Management Tool YouTrack, GitHub

Currently we are using GitHub as the "Project-Management-Tool". In the future we would like to upgrade to YouTrack, but we don't have the license for YouTrack yet (license is needed for groups with more than 3 people).  
[1]<https://github.com/users/tobi4321/projects/1>

Don't hesitate to give us feedback to the roles and the technologies.

Greetings,  
Track Your Fit Team

1. <https://github.com/users/tobi4321/projects/1>

---

Jonas Heinrich (2019-10-12 13:13:10)

Hello Track Your Fit Team, I don't have any feedback on the distribution of your roles, however I am curious about your technology choices. By reading your first blog post I understand your vision and can imagine an app that realizes it. I thought that Xamarin and Flutter are alternatives for building a cross platform mobile application, so how do you plan to combine those two? Or are you just postponing that decision? Looking at the rest of your stack I would guess that you end up with Xamarin since it fits with the backend in C # quite nicely. Nevertheless it is a university project for learning about Software Engineering and new technologies so I guess it could be a great learning opportunity. We chose PostgreSQL instead of MySQL, since it is open source and got recommended to us by different people. I am in no way an expert on the differences between database implementations, so do you have any technical reasons for using it? Last but not least I want to point out that you included XML and HTML into your programming language category. I think you accidentally misplaced them. Nonetheless I think its funny since it is a bit of a [meme]([https://www.reddit.com/r/ProgrammerHumor/comments/7k6s0o/its\\_not\\_a\\_programming\\_language\\_ffs/](https://www.reddit.com/r/ProgrammerHumor/comments/7k6s0o/its_not_a_programming_language_ffs/)). No offence ;) Best Regards, Jonas from unitasks.net PS: If my link is broken you should probably enable markdown support in your wordpress settings.

TrackYourFit (2019-10-14 08:12:01)

Thank you for your feedback! You guessed it right, we are going to postpone the decision of using Flutter/Xamarin since two of our team members are going to present their ePortfolio on these two. We actually just chose MySQL since where kind of used to it, but we will definitely take a look at PostgreSQL! Thanks for pointing the XML and HTML mistake out! Obviously they are Markup Languages (as it literally says in their name lol). Best Regards, Janis from TrackYourFit

Jeanne (2019-10-13 18:09:21)

Hello trackyourfit-Team, i like that you decided to define that many roles for your team but i am missing the exact differences and exercises for all of those roles. Maybe it's because we're just using a different terminology in my team. :D What exactly is a configuration manager? Will you work together by using scrum? In that case I think Tobias is the Product Owner, not the Product Manager. Maybe it would be good to add another column for your roles, to define briefly what exactly one role member has to do. According to your technologies I have to admit, that i didn't know Xamarin before. As some other groups are using Flutter, it would be cool when you decide to use Xamarin so we can look in more new technologies in the next semesters. Do you already know, how you will test your application? I would recommend to use some testing framework for automatically testing (But you can't cover 100 % of your code with that!). It can guarantee that your system will still work for example after refactorings or someone can understand your code much better by looking into your tests too. I think the importance of tests is very often underestimated. It should be discussed and used from the beginning because you will seldom reprogram it. I will follow your project with interest. :) Best wishes, Jeanne :)

TrackYourFit (2019-10-21 17:21:36)

Hi Jeanne, for our team roles we decided to use the RUP terminology. You can check it out here: <http://www.ibm.com/developerworks/rational/library/apr05/crain/> We are still unsure about what we will use to develop the app, but we are leaning towards Xamarin since we already have a C # backend through the Webapplication. In the future there's going to be a lecture about Cucumber, so we will await this lecture for picking a testing framework! Thanks for your feedback, Your TrackYourFit-Team

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:31)  
[...] Week 2: Roles and Technologies [...]

### **Week 3: SRS and UCD (2019-10-22 11:24)**

Hello everybody,

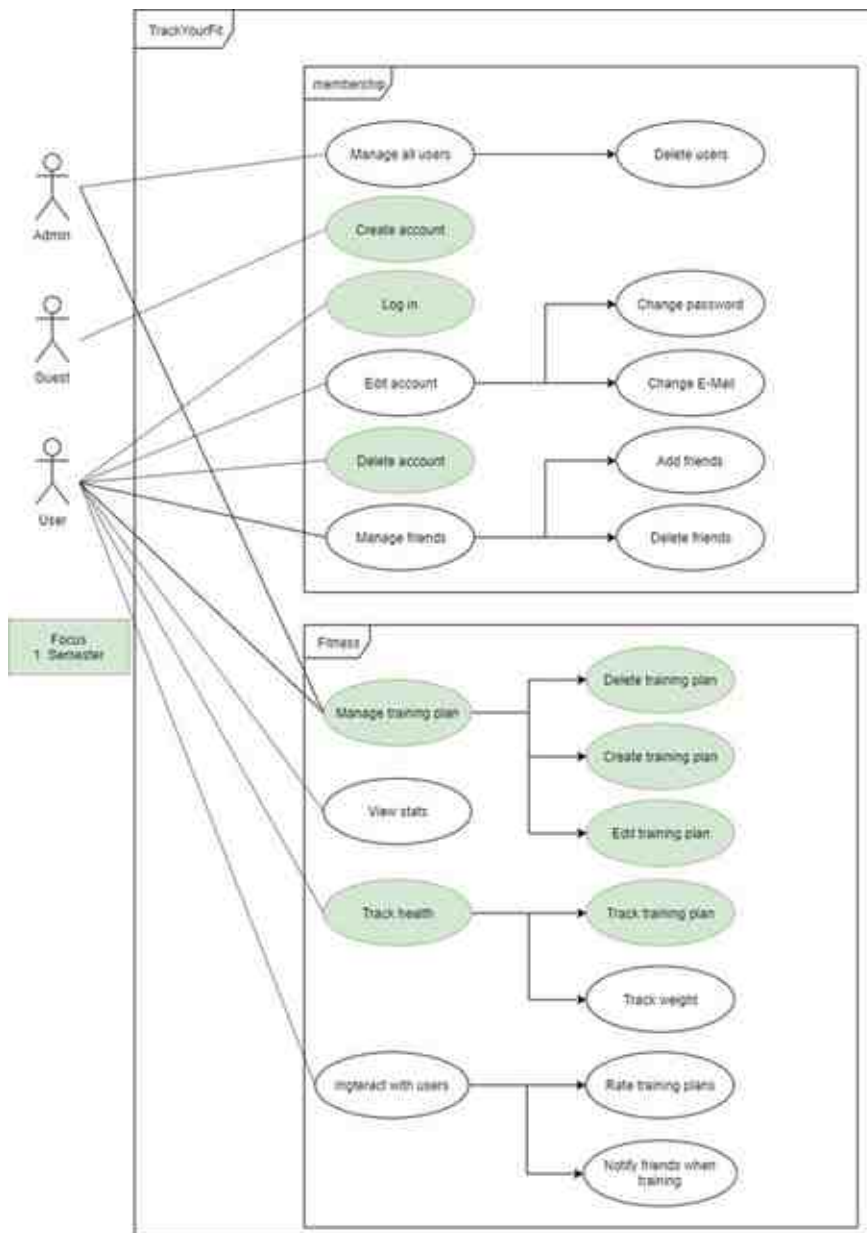
We've started filling out the SRS as you can see here:

[1]<https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/SRS.md>

Our Use-Case-Diagram can be found here:

[2]<https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UseCaseDiagram.jpg>





Feel free to give feedback and comment!

Your YouTrackTeam

1. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/SRS.md>
2. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UseCaseDiagram.jpg>

TheFriendsLook (2019-10-23 06:58:38)

Hey guys, I really like that you kept your blog entry simple and understandable. Maybe you could improve it by directly embedding your files instead of referring to your GitHub repository. Best regards, The TheFriendsLook Team

TrackYourFit (2019-10-28 12:43:30)

Thanks for the feedback! We've added the UC-Diagram, as you mentioned:)

Arerrac (2019-10-23 09:34:09)

Hello track your fit-team, We really like your usecase diagram. It is very clarifying and we now have clear understanding of the uses of every person. We hope you will accomplish the goals you defined in the diagram for this first semester! Your SRS looks promising, though you could reconsider using a better formatting for the text. For example the headers which do not differ from the rest of the text and have no spaces after the enumeration make the text hard to read. Furthermore we see more inelegant format errors in your text. For example: 1. 2. 3.6Design Constraints 1. 3.6.1MVC Architecture Greetings, Your Arerrac

TrackYourFit (2019-10-28 12:44:51)

Thanks for the feedback! We've had a hard time getting the word file into a .md file, but we're working on it!

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:33)

[...] Week 3: SRS and UCD [...]

## **Week 4: Use Case Specifications (2019-10-27 12:34)**

Hello guys, this week we took a deeper look in two of our UseCases. The following links will get you there!

UCS-CRUD-TrainingPlan:

[1]<https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-CRUD-Trainingplan.md>

UCS-TrackHealth:

[2]<https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackHealth/UseCaseTrackHealth.md>

As always, we're happy for any provided feedback!

Your TrackYourFit-Team

1. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-CRUD-Trainingplan.md>

2. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackHealth/UseCaseTrackHealth.md>

lilolunatic (2019-10-27 15:39:52)

Dear trackyourfit-Team, We liked your use-cases and the flow-diagrams. It's quite detailed and you get a clear image of the application and its functions. The only thing that caught our eye is that you should maybe choose a different name for the first use-case, since they ususally describe an activity. Creating, editing and deleting are standard-operations so it can make sense to merge them. The second use-case looks fine, but maybe you should check the spelling on the mockups. But in general, they are pretty clear and give a good impression of the application. Best, Your Orchestra-Team

TrackYourFit (2019-10-28 12:47:36)

Dear Orchestra-Team, Thanks for your feedback! Regarding the first use-case, we named it CRUD since it is all about creating, reading, updating and deleting entries. Regarding the second one, we really have to look into the spelling, might've been a little late when we did it! Best, TrackYourFit

/ Ardragos (2019-10-27 16:31:35)

Heya trackyourfit Team, beside the things another group already mentioned, we wanted to add a few things: First in your first usecase the black circle and red arrow is not a part of a CRUD diagramm. Second, your last mockup (2.4) looks very poor made in comparison to the ones above it, maybe you could consider making it a bit more appealing? Lastly we wanted to remind you that the usecases should be linked in the SRS document which is missing currently. Anyway, the diagrams look very nice and all pre and postconditions are clearly stated. Have a good day, Team Itemize

TrackYourFit (2019-10-28 12:50:38)

Hey Team Itemize, thank you very much for your feedback, we've deleted the circle in the first use case! Regarding the mockup for 2.4 we forgot to state that its only going to be a modal popping up on your screen so that's why it doesn't look as good as the others! Have a good one, TrackYourFit-Team

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:35)

[...] Week 4: Use Case Specifications [...]

## 1.2 November

### Week 5: .feature files (2019-11-05 18:12)

Hello everyone,

quick little update from us! We've sat down and made a few feature files for our use cases. Regarding our CRUD use case, we decided to add a feature file for each CRUD action, as you can see in our Markdown file for that UseCase. Unfortunately we didn't start testing yet, since we have not started developing a proper UI, where testing would be useful.

.feature files for use case "CRUD-Trainingplan":

```

1 Feature: List all Trainingplans
2   as a User
3   I want to list all my Trainingplans and be asked to create one if I don't have any
4
5   Scenario: User has Trainingplans
6     Given I am logged in
7     Given I am on the Page Index
8     When I press the button with the id "overviewTP"
9     Then Section "Overview" should open
10    And the table with the id "allTrainingplans" should have more than one row.
11
12   Scenario: User does not have any Trainingplans
13     Given I am logged in
14     Given I am on the Page Index
15     When I press the button with the id "overviewTP"
16     Then Section "Overview" should open
17     And I am prompted to create a Trainingplan.

```

```

1 Feature: Update Trainingplan
2   as a user
3   I want to update my Trainingplan
4
5   Scenario: Rename Trainingplan
6     Given I am logged in
7     And I have navigated to section "Edit" on an trainingplan
8     When I change the name in the field with id "name"
9     And I press the Button named "Save"
10    Then The confirmation popup should be displayed.
11
12   Scenario: Accept rename
13     Given I am logged in
14     And I have navigated to section "Edit" on an trainingplan
15     And the confirmation popup is displayed
16     When I press the Button named "Confirm"
17     Then The popup should close
18     And the name should have changed
19
20   Scenario: Abort rename
21     Given I am logged in
22     And I have navigated to section "Edit" on an trainingplan
23     And the confirmation popup is displayed
24     When I press the Button named "Cancel"
25     Then The popup should close
26     And the name should not have changed

```

```

1 Feature: Create trainingplan
2   as a user
3     I want to Create and name a trainingplan.
4
5   Scenario: Valid trainingplan name
6     Given I am logged in
7     And I have navigated to section "Overview"
8     When I enter a text into input field with id "TPname"
9     And I press the button with id "TPaddBTN"
10    Then Section "Edit" should open
11
12   Scenario: No trainingplan name
13     Given I am logged in
14     And I have navigated to section "Overview"
15     And I enter no text into input field with id "TPname"
16     When I press the button with id "TPaddBTN"
17     Then Alert with text "Please input a valid name" should be displayed

```

```

1 Feature: Delete Trainingplan
2   as a user
3     I want to delete my trainingplan.
4
5   Scenario: User accepts to delete the trainingplan
6     Given I am logged in
7     Given I have navigated to the section "Edit"
8     Given I have pressed the button with id "askDeleteTP"
9     When I press the button with the id "deleteTP"
10    Then section "Overview" should open.
11
12   Scenario: User declines to delete the trainingplan
13     Given I am logged in
14     Given I have navigated to the section "Edit"
15     Given I have pressed the button with id "askDeleteTP"
16     When I press the button with the id "cancelDeleteTP"
17     Then section "Edit" should open.

```

Feature files for use case CRUD-Trainingplan

Github link for UC md file:

[1]<https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-CRUD-Trainingplan.md>

Following the feature files for our second use case "TrackHealth":

```
1 Feature: trackTrainingsplan
2
3   This Feature enables the user to submit trainingsprogress on a by the user chosen Trainingplan.
4
5
6 Scenario: GoToSubmitTrainingprogress
7
8   Given I am logged in
9   When I press the Menue Button having the name "Track Health"
10  And I press the Button having the name "Trainingprogress"
11  Then Section "Submit Trainingprogress" should open
12
13
14 Scenario: SubmitTrainingplanFully
15
16   Given Section "Submit Trainingprogress" is open
17   When I check all the checkboxes
18   And I press the Button Submit
19   Then Then I should see alert text as "You Successfully submitted your weight"
20   And Section "Mainwindow" should open
21
22
23 Scenario: SubmitTrainingplanPartially
24
25   Given Section "Submit Trainingprogress" is open
26   When I check a potion of the checkboxes
27   And I press the Button having name "Submit"
28   Then Then I should see alert text as "Is it correct that you haven't done These Workouts:"
29   And Window having name "Summary of unchecked Workouts" should be present
30
31 Scenario: CorrectIHaventDoneTheWorkout
32
33   Given Window having name "Summary of unchecked Workouts" is open
34   When I press the Button having name "Correct"
35   Then Then I should see alert text as "You Successfully submitted your weight"
36   And Section "Mainwindow" should open
37
38 Scenario: InCorrectIHaventDoneTheWorkout
39
40   Given Window having name "Summary of unchecked Workouts" is open
41   When I press the Button having name "Incorrect"
42   Then Window having name "Summary of unchecked Workouts" should close
```



```

Feature: Submit Weight
  as a user
  I want to submit my weight

  Scenario: Accept weight
    Given I am logged in
    And I have navigated to section "TrackHealth" on an trainingplan
    And I have navigated to section "weight"
    When I change the weight in the field with id "weight"
    And I press the Button named "Submit"
    Then The View stats page is displayed

  Scenario: Reject weight
    Given I am logged in
    And I have navigated to section "TrackHealth" on an trainingplan
    And I have navigated to section "weight"
    When I press the Button named "Submit"
    Then Alert with text "Please input a valid weight" should be displayed

```

Github link for UC md file:

[2]<https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackHealth/UseCaseTrackHealth.md>

/UC-

Feel free to leave feedback and questions if there are any!

1. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-CRUD-Trainingplan.md>
2. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackHealth/UseCaseTrackHealth.md>

---

Leonhard Gahr (2019-11-07 14:46:26)

Hi Trackyourfit team, based on the coverage your feature files look quite good to me. However, you sometimes use for multiple preconditions "Given ... And ... And ..." which is good, but sometimes you use "Given ... Given ... Given ..." and I think this would be nice to be updated, because it looks better and for consistency. But this is only my personal opinion :) Also you put your feature files in your UCs which I think is really good. Keep it going! Leonhard@GameBase

thefriendslook (2019-11-19 19:15:21)

Hi everyone from Trackyourfit Team, It's nice to be up to date, but I would be happy to know when the test phase can finally

begin. When will you start to develop a proper UI? I am really interested how your project will continue! Mo@TheFriendsLook

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:36)

[...] Week 5: .feature files [...]

## Week 6: PM tool + Integration (2019-11-12 14:40)

This week we linked github, YouTrack and Visual Studio.

We decided to use sprints of one week, since we want to integrate the peer reviews, blogposts and homework into our sprints as well. Our agile board, we use for scrumming can be found here:

[1]<https://dhw-karlsruhe.myjetbrains.com/youtrack/agiles/108-2/109-31>

As you can see in the following pictures, we are using estimation and time spent fields for time tracking.



The image shows a screenshot of a task card in the YouTrack application. The card is titled 'TrackYourFit' and contains various fields for task management. The fields are arranged in a list-like format with labels on the left and values on the right. Some fields have small icons to their right, likely for quick actions or status indicators. The background is a light gray, and the text is in a clean, sans-serif font.

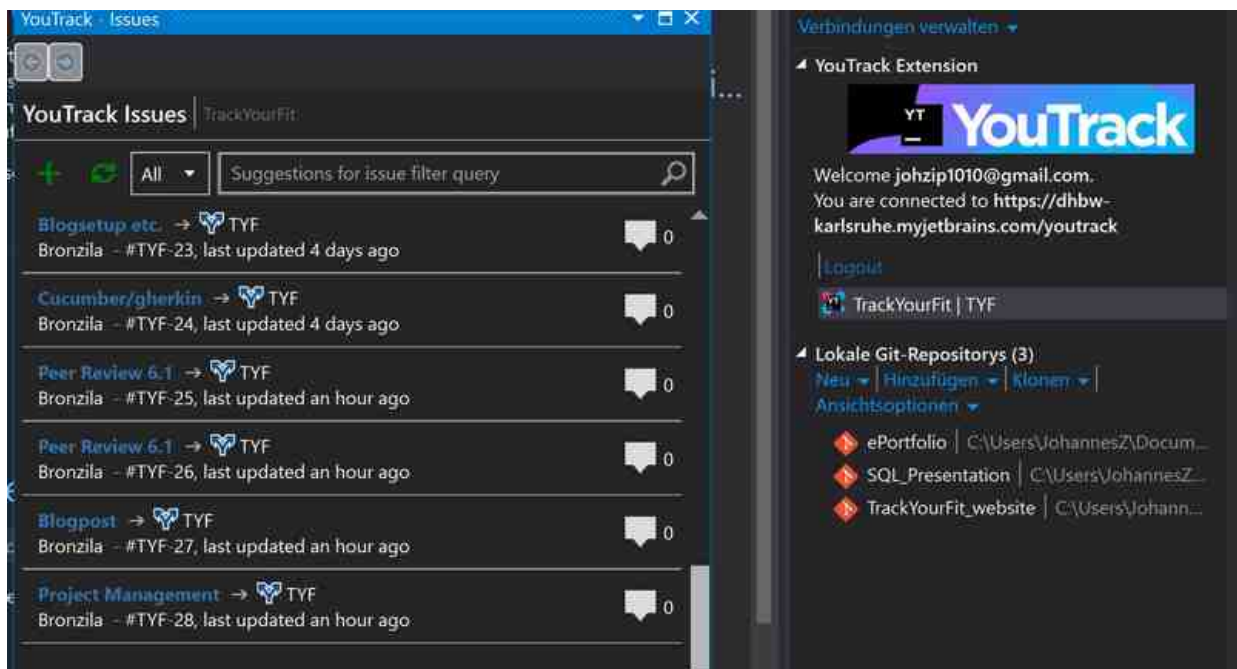
Project	TrackYourFit	
Priority	Normal	
Type	Task	
State	Done	
Assignee (multiple)	Bronzila	
Sprints	Week 5, Week 6	
Scope	Implementation	
Time spent	2h.45m	
Estimation	4h	
RUP workflow	Implementation	
RUP Phase	Elaboration	

Since we linked our YouTrack with GitHub, you can see the commit messages on YouTrack if they are mentioned in the commit.



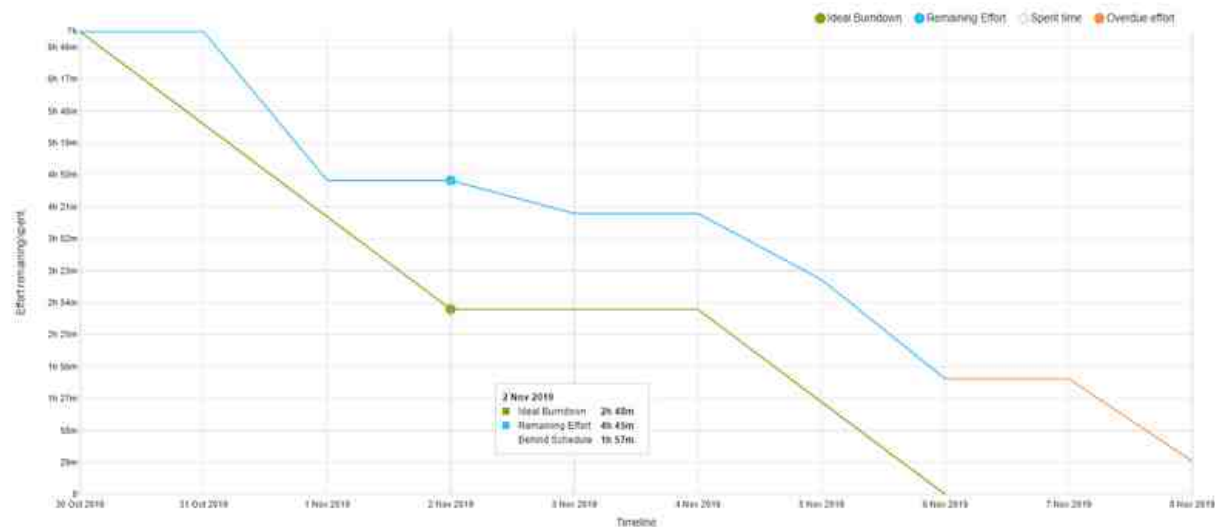


Here's the integration in our IDE (Visual Studio). As you can see there is our issue list:



Last but not least here's our first burndown-chart. As you can see we haven't quite been able to finish all our tasks, that's why the effort line didn't go all the way down:

[2]<https://dhbw-karlsruhe.myjetbrains.com/youtrack/reports/burndown/147-2>



With YouTrack, you also have the possibility to create a Gantt-Chart, but unfortunately we didn't add relations between issues yet.

That's it for this week, feel free to leave any feedback!

1. <https://dhbw-karlsruhe.myjetbrains.com/youtrack/agiles/108-2/109-31>
2. <https://dhbw-karlsruhe.myjetbrains.com/youtrack/reports/burndown/147-2>

---

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:38)  
 [...] Week 6: PM tool + Integration [...]

## Week 7: Retrospective (2019-11-16 14:26)

Hello everyone,  
 this weeks blogpost is about the retrospective we had in class.  
 Here's our whiteboard-page:

## Wir feiern uns für...

Track Your Fit

- ... Zeitmanagement (Hausaufgaben?)
- ... fertige ePortfolios
- ... ~~finden~~ die Idee
- ... YouTrack setup
- ... funktionsfähige VK + Integration IDE
- ... Raclette (Teambuilding)

## Wo liegt Verbesserungspotential?

- 4 - Arbeitsverteilung
- 1 - Statusinformationsaustausch
- 3 - Entwicklungsgeschwindigkeit
- 2 - KISS 
- 5 - Raclette-Ausgleich 

Alle Fortschritte kommunizieren  
=> Whatsapp + YouTrack-connector

Klar an Aufgaben (Issues) halten

Issues noch schneller integrieren

As you can see we thought about what went well, where we see potential and what solutions we have in mind to reach the potential. The topics regarding the "Raclette" is more of a fun addition, since we live together and bought a raclette set (which is simply amazing). Since we're trying to develop an application to help you stay sporty and healthy we thought this is kinda ironic ;)

Let's get in to the points a little deeper:  
Part 1: What makes us great?

The first thing that came to mind was that we (almost) always get all the homework and peer reviews or other topics done to when they are planned, so we noted Timemanagement. Another thing we are happy about is, that every team member has done his ePortfolio, so that's not getting in the way of working on the project! Moreover we agreed, that our idea is pretty sick and we're looking forward on how we're going to realise it. Another point is, that we liked how we setup our YouTrack, since this took a little more time than expected. Along with that we're happy that we have a working VCS in git and it's integrated in our IDE + connected with our YouTrack.

#### Part 2: Where's potential for improvement

Now we had to think about what's not going that well (yet). Communication is key (and a part of teamwork you can pretty much always improve). Moreover we thought about our application itself. What came to mind is, that we should start focusing on smaller goals for now and KISS (keep it simple and stupid). Another point we weren't really happy about is, how fast/slow we're developing, since the beginning of a project is often pretty slow and clunky (at least that's what it feels like). We also thought about improving the workdistribution.

#### Part 3: Solutions to reach our potential

In order to have a better exchange and communication, we agreed on telling everyone how far we've come through a textmessage in WhatsApp and a quick little comment on our YouTrack issue. To KISS we just really need to do what's written in our issue description and not do more/something else. To enhance our development speed, we just think that a good sprint planning and then working on the sprint is the key.

That's it for this week. Can't wait for your feedback!

---

Mr Card Game (2019-11-18 07:09:13)

Hi, I like what you did there. I think you covered every point very well and your solutions to improve your communication seems very good. I would just recommend to try making dailys that's what helped us to stay on track. This also might enhance the developing speed a bit.

TrackYourFit (2019-11-18 13:09:53)

Hey Mr Card Game- Team, Thank you for this comment. We will think about a daily, since you are right about, it would speed up the process. Best Regards TrackyourFit

Arerrac (2019-11-20 09:26:21)

Hello guys, First of all, you divided your points very clear so it's easy to read your blog entry. Your "team building idea" with the raclette is a nice touch, we didn't see in other groups. Your plan to stay in touch with your team mates after every task is helpful for sure. But don't forget to coordinate who works on which issue. Regular meetings may help as well. Kind regards, Arerrac

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:40)

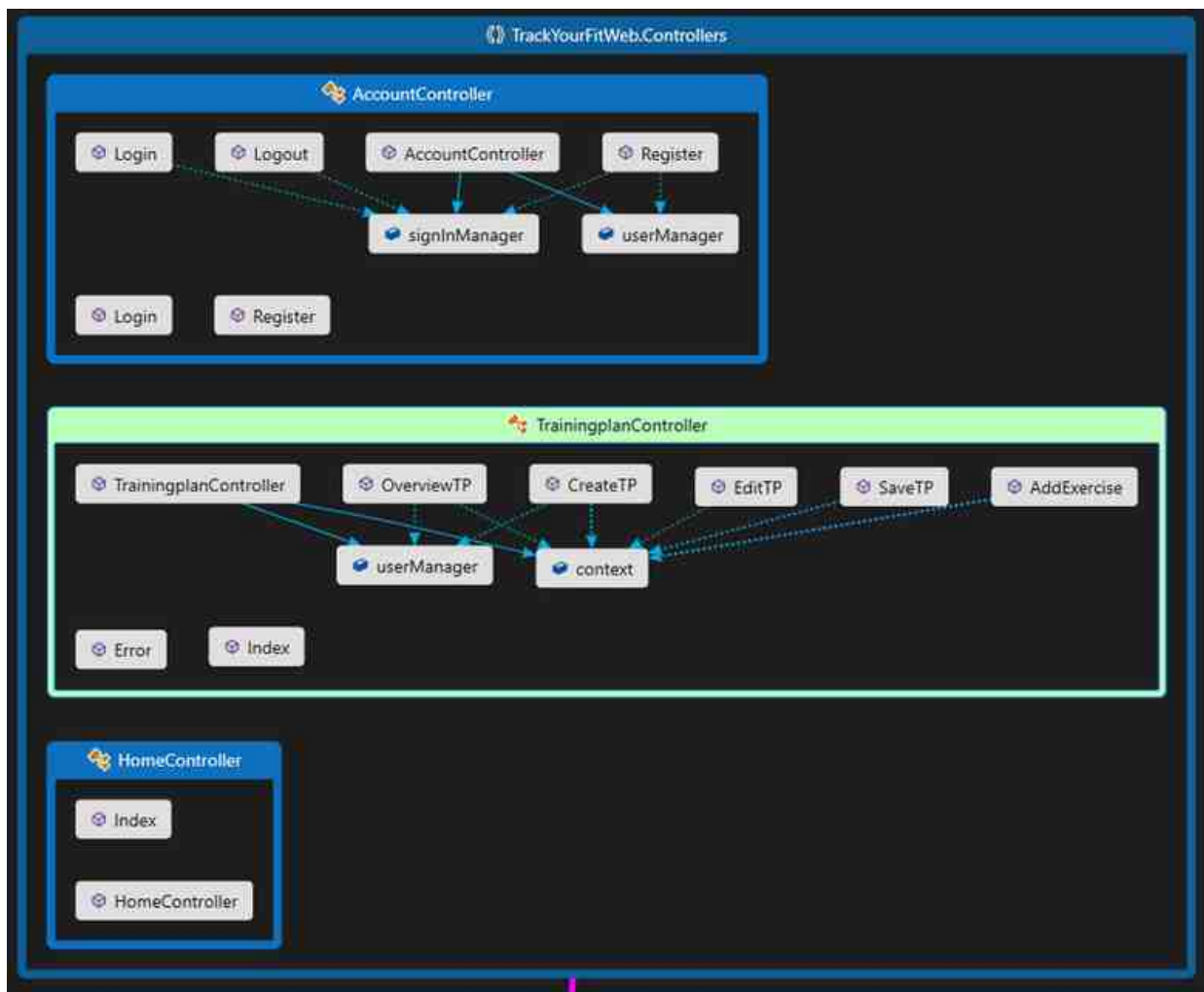
[...] Week 7: Retrospective [...]

## Week 8: Class diagrams & DB schema (2019-11-23 12:54)

Hey everyone! This weeks blog post is all about classes and diagrams! We would like to show you our currently used classes and since we're using a database you will also be pleased by a database diagram!

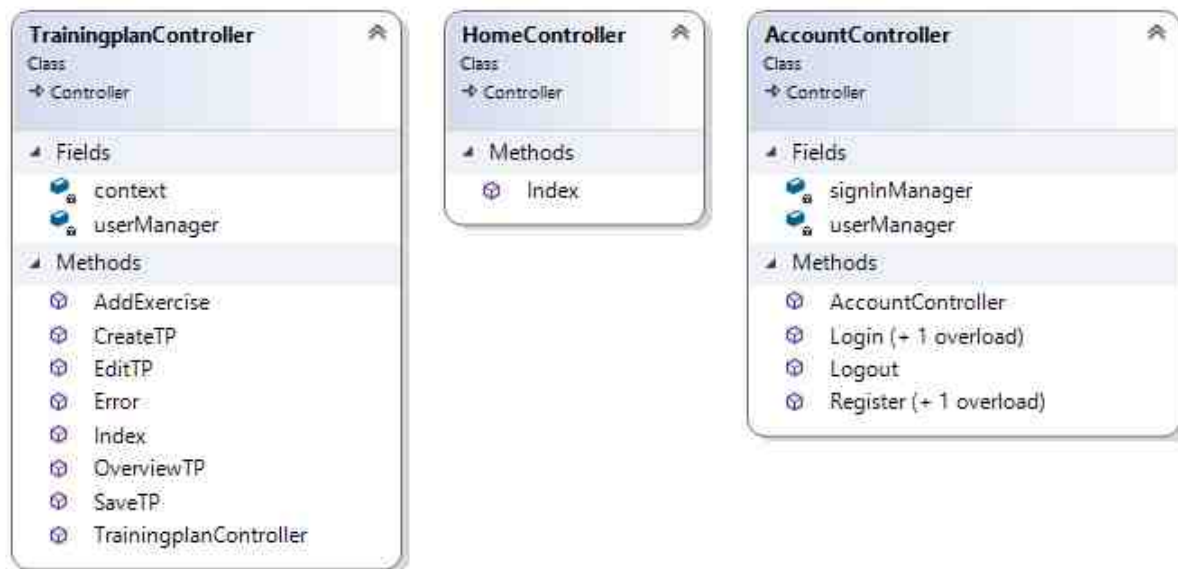
Our application is developed in Visual Studio, but sadly VS doesn't really have a "beautiful" way of creating class diagrams. There's two different approaches to getting a class diagram: There's a thing called the codemap, which shows the relation between multiple classes and even their methods/properties/fields. Sadly you can't really see what attributes the fields have. Therefore there's another way of creating a somewhat class diagram with the class diagram extension. Unfortunately these don't really show any connections, well see for yourself:

This is the codemap of our controllers:

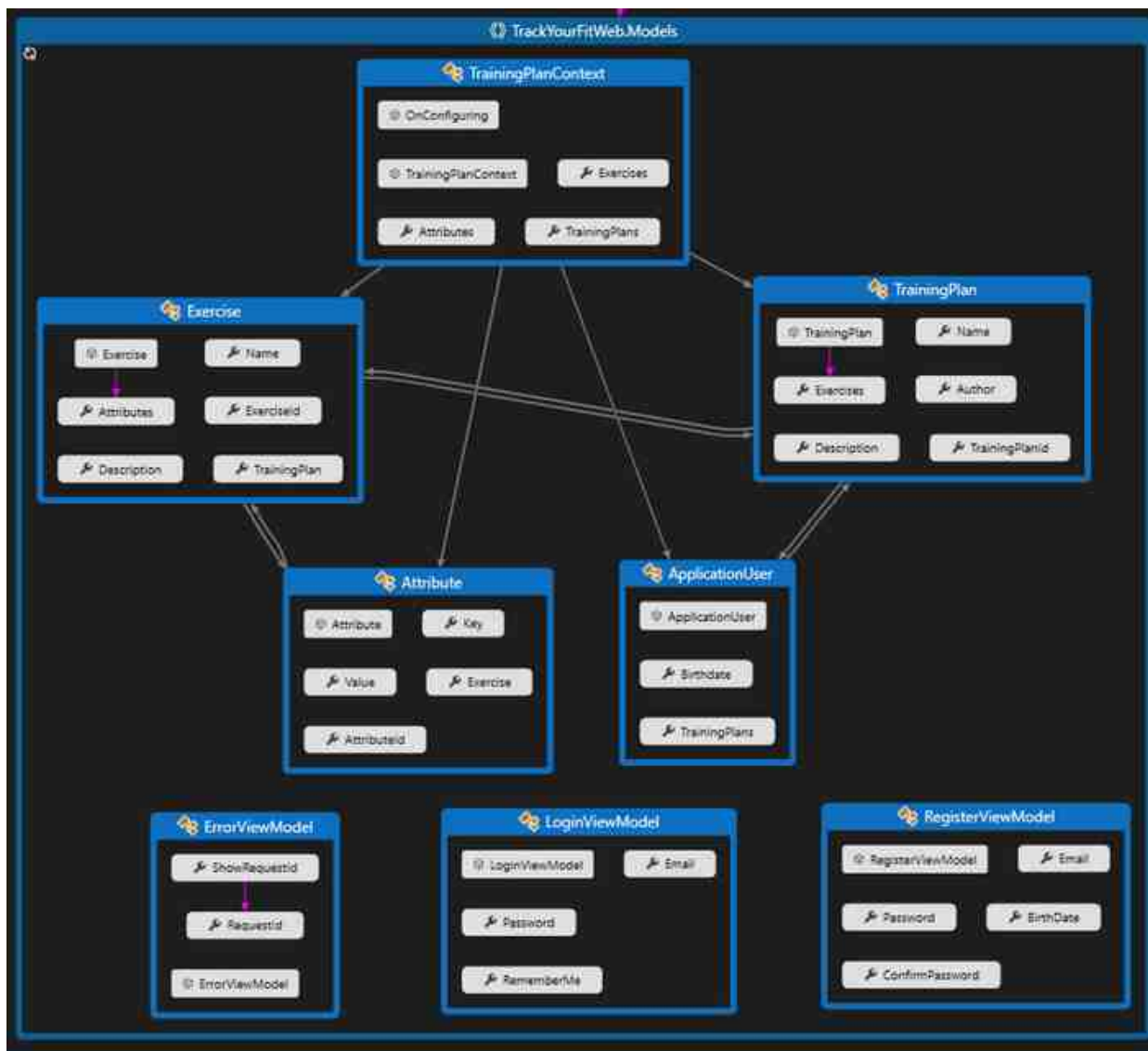


All purple cubes are methods and the blue ones are fields.

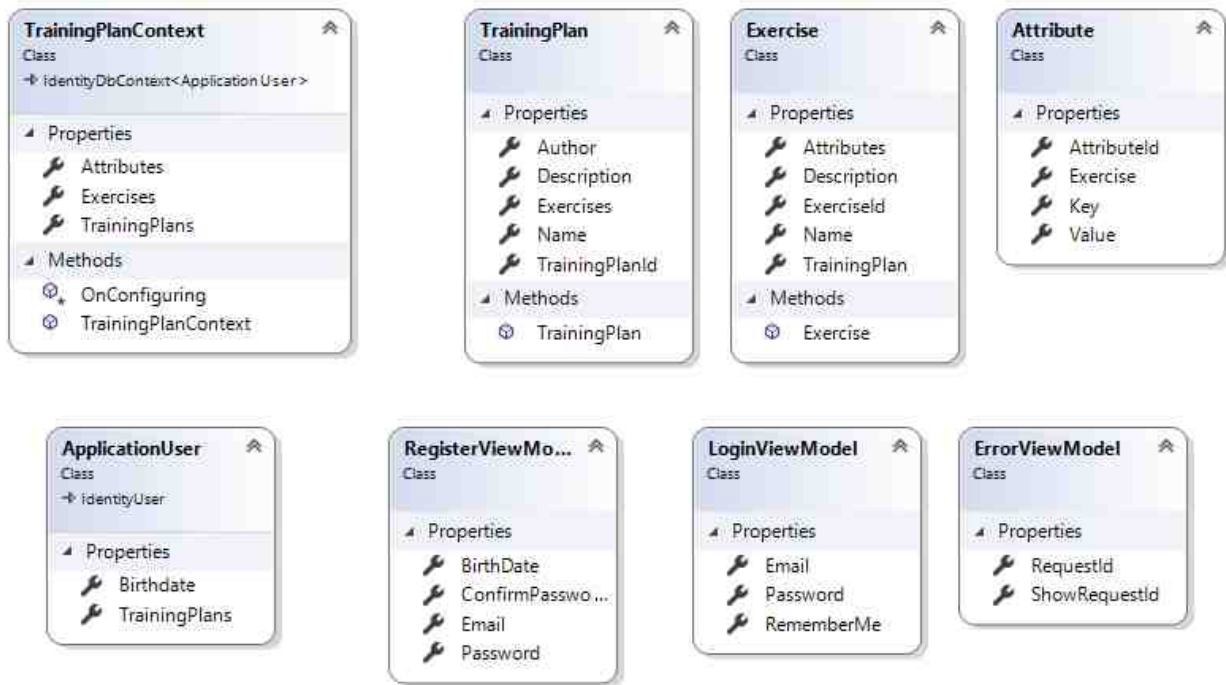
And now here's the class diagram for the same exact three controllers:



As you can see in here it's a little more "tidy".  
Now this is the codemap for our models:



Followed up by the class diagram for our models:



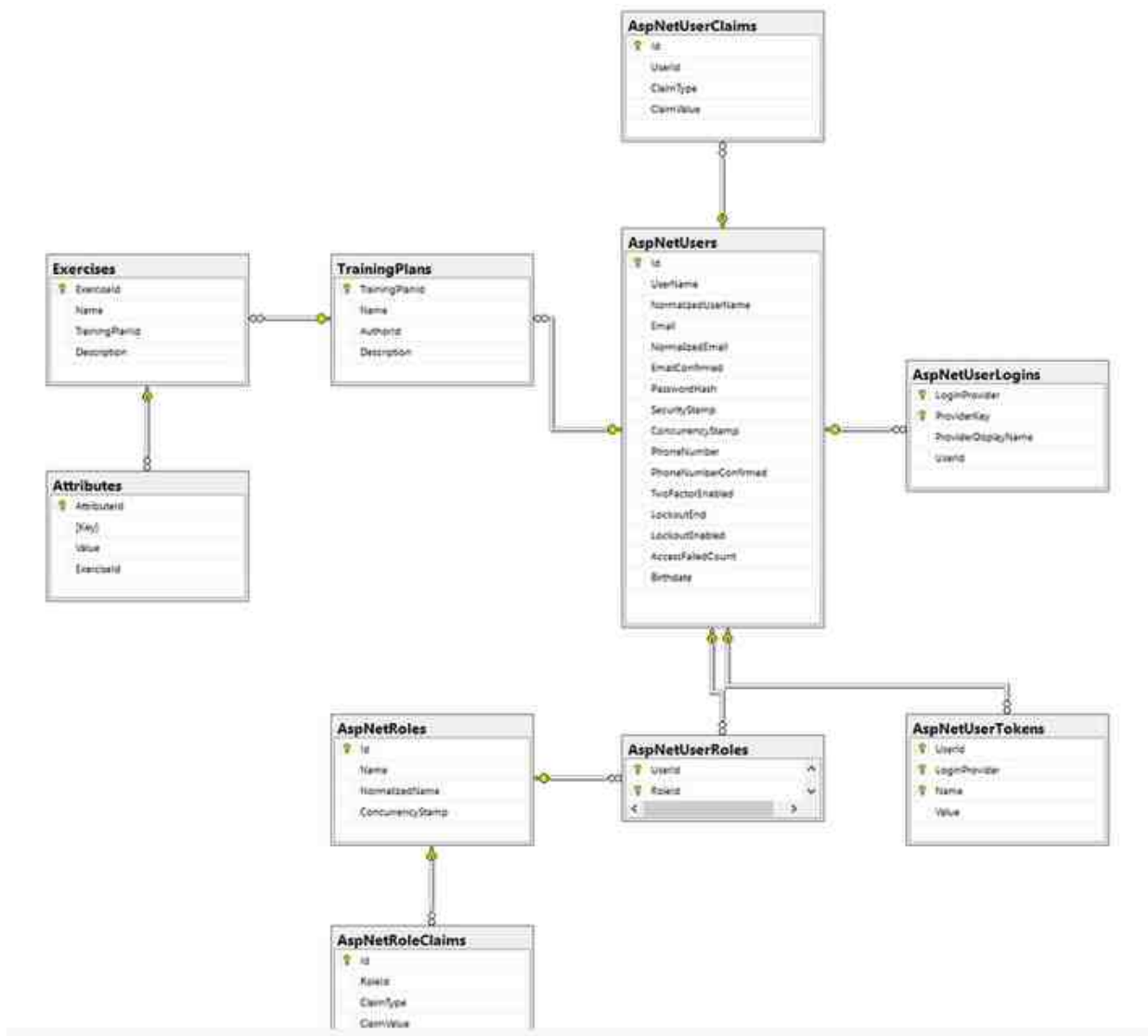
The good thing about the class diagram is, that you can actually have an insight on the classes themselves, by clicking on them. When you do so you can actually see all datatypes and modifiers of the properties/methods:

Class Details - TrainingPlanContext

Name	Type	Modifier	Summary	Hide
<b>Methods</b>				
OnConfiguring	void	protected		<input type="checkbox"/>
TrainingPlanContext		public		<input type="checkbox"/>
+ add method				<input type="checkbox"/>
<b>Properties</b>				
Attributes	ICollection<Attribute>	public		<input type="checkbox"/>
Exercises	ICollection<Exercise>	public		<input type="checkbox"/>
TrainingPlans	ICollection<TrainingPlan>	public		<input type="checkbox"/>
+ add property				<input type="checkbox"/>
<b>Fields</b>				
+ add field				<input type="checkbox"/>
<b>Events</b>				
+ add event				<input type="checkbox"/>

Now since we promised you a database diagram aswell, here you go:





Note that a lot of the tables were created by .NET Core Identity (an extension we use for managing our users), so not all of its functionality is in use (yet!).

Last but not least here's the link to our SAD-document (please don't criticize it too much, it's already SAD ;D)  
 [1]<https://github.com/tobi4321/TrackYourFit/blob/master/SW-Dokumentation/SAD.md>

We hope, that we could give you a good update on our classes!  
 If you know a better way of creating more beautiful diagrams, please let us know.  
 Feel free to leave feedback!

---

@rawbean (2019-11-24 14:04:26)

Hey ho, impressive work with your classes. It looks clean and well structured. It also looks very simplistic what brings easy reability with it, good job. What does your "Attribute"-Class do? I cannot find out by the values inside. Maybe rename either the class or the values for an external view. Does it already run (some) features or is it still in "prototype"-state? Keep up the good work, looking forward to the final result. Cheers, @raw bean (Foody, B4)

TrackYourFit (2019-12-04 10:18:10)

Hi, thank you for your comment. The Attribute Class contains the Attributes of an Exercise, for example key: Distance, value: 20km. The feature files are still in a prototype state, but we are working on it. The TrackYourFit Team

/ Ardragos (2019-11-24 19:21:20)

Heya, very nice and clear diagrams. These diagrams don't have to 'beautiful' as you stated, it's more important that it is understandable in the first place, and they are definitely! We just have one question: What is a field, we never heard of that before. Have a good time and We are looking forward seeing what you create the next weeks! Team Itemize

TrackYourFit (2019-12-04 10:21:37)

Hi, we think they have to be beautiful to be nicely understandable, and we are glad you like them. A field is a variable of any type that is declared directly in a class or struct and fields are members of their containing type. Thanks, have also a great time. The TrackYourFit Team

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:41)  
[...] Week 8: Class diagrams & DB schema [...]

## 1.3 December

### Week 9: MVC & redefined scope (2019-12-02 14:49)

Hello everyone!

This weeks post is all about MVC! (and a little bit of us paddling back)

## What is MVC?

MVC (Model-View-Controller) is a specific software architecture, that divides the software in three main sections. You guessed it right, the models, views and controllers! This way software may get a little less complex and faster to

develop. If you wanna go deeper, just look it up on google, you'll be presented by more than enough results to cover your thirst of knowledge!

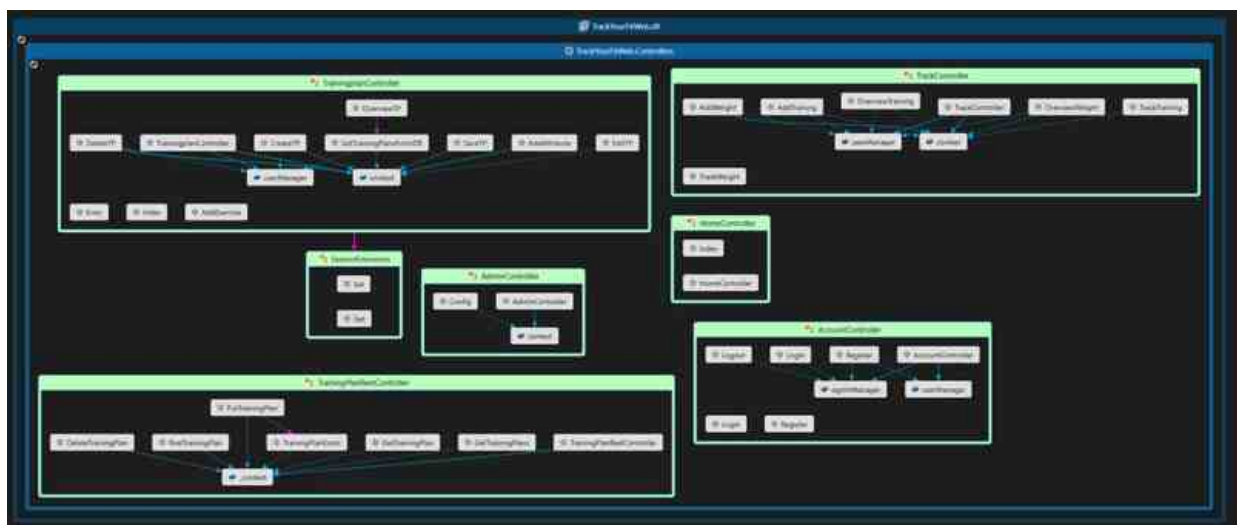
## What do you use for MVC?

Since we are creating a webapp, we are using the ASP.NET Core MVC framework, it's an open source framework by Microsoft and really does it's job! It even provides scaffolding views for the CRUD-actions of your models! But don't be too excited about that, since it can only handle single inputs and don't even think about handing it a list!

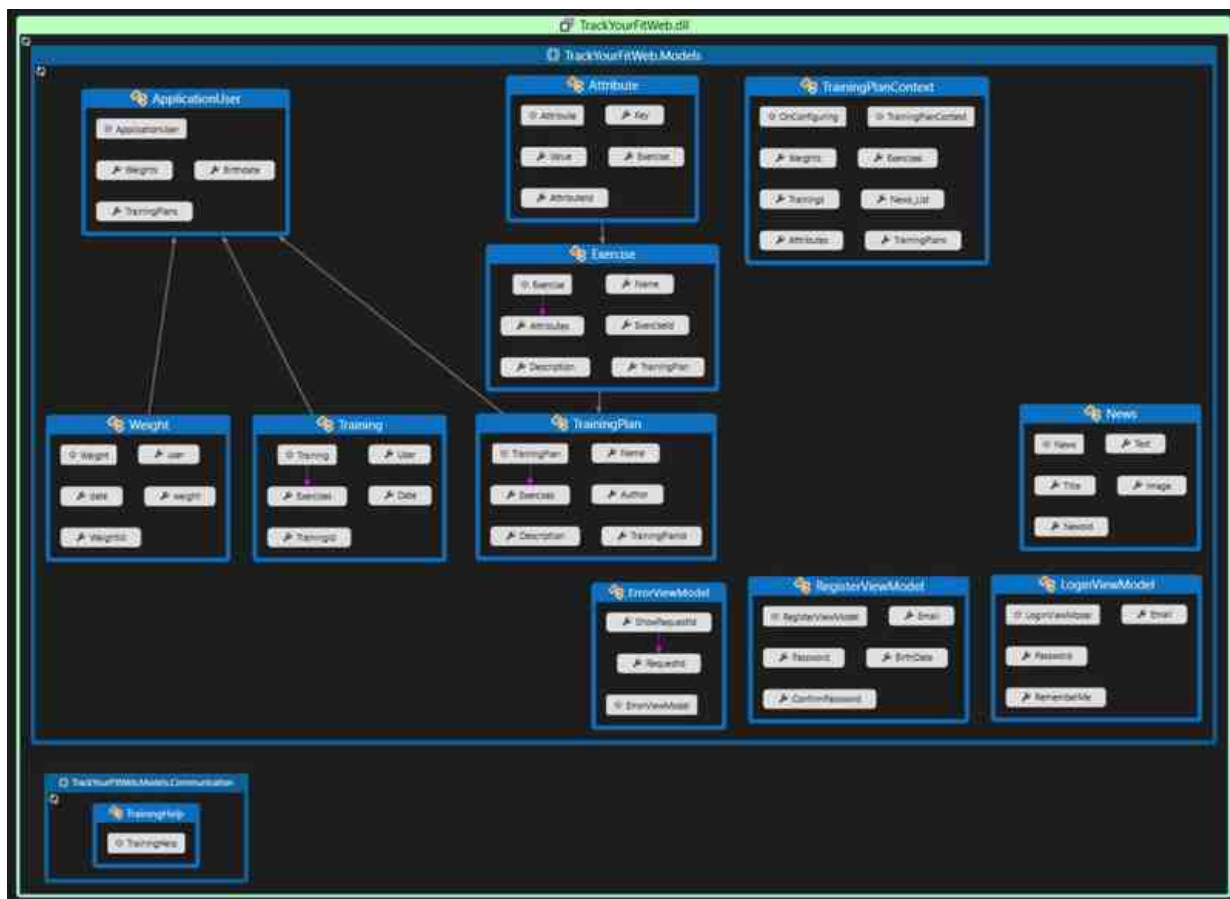
## What's it looking like?

We have already shown are current state of development regarding class diagrams of our controllers of models, but since you can never get enough of diagrams, here are some more:

Controller:



Models:



Where's the rest?

Further information can be found in our SAD-document:

[1]<https://github.com/tobi4321/TrackYourFit/blob/master/SW-Documentation/SAD.md>

What else happened?

Not only did we set up our SAD-document, we also took another look on our scope for the end of this semester, sadly we were a little too euphoric about what we can do in this short period of time and we had to redefine our scope! Tracking your training plans is a feature, we most certainly wont be able to implement this semester, since exams are

coming up and setting up everything did take quite a while! What we really want to ensure is, that you will be able to CRUD training plans and track your weight (and obviously create an account).

That's it for this weeks litte update, if you have any feedback feel free to comment!

Cheers,  
TrackYourFit

1. <https://github.com/tobi4321/TrackYourFit/blob/master/SW-Documentation/SAD.md>

---

Robin (2019-12-02 15:40:47)

Hello TrackYourFit, I had a look at your blogpost and your SAD and I like most of what I see. Your SAD contains relevant information to all required points. You do mention a web app and a mobile app: as far as I can tell, your SAD only covers the web app? Could you please clarify this point and flesh out Architectural Goals and Constraints some more. At point 5.2 I miss the views. I have no experience with ASP.NET Core (or Xamarin) and how it implements MVC. Is providing UML-Class-Diagrams impractical in your framework? Sidenote: why do you have low resolution jpg images on your blog, when you already have much prettier pngs on your github? Your pared down goals for this semester will still represent quite an achievement. I would enjoy seeing a demo of your project. Best regards, Robin from SaSEp: Clairvoyance

TrackYourFit (2019-12-04 10:28:27)

Hello, In the moment we are concentrating on the web app, but the plan is to extend it to a mobile app in the future. The Views are just single blocks without connections, so they would just be confusing and unnecessary. The Images are the same as on github and we don't know why they got so compressed and low resolution, but we try to fix it. The TrackYourFit Team

Nico (2019-12-02 16:14:26)

Hey there, although you give a good overview for the model and controller, I think you should put everything in one picture and also add the view to it. That way when someone outside of your group looks at it, it's easier to understand. I really missed some progress on actual development but I guess exams are coming soon and need preparation. I looked at your SAD and your scope is well chosen for this semester, it covers a great part of your app (Training Plan) and also login/register. So I don't think you are behind schedule :) Best Regards Nico (bookly)

TrackYourFit (2019-12-04 10:32:00)

Hello, we splitted everything in separate images because otherwise they would get quite big and just more confusing in our opinion. The View are missing because they are just single blocks without any connections and would just blow things up. We also think the scope is nice and we think everything should work on the exam. Greetings The TrackYourFit Team

Week 11: Midterm Presentation – trackyourfit (2019-12-10 19:04:43)  
[...] Week 9: MVC & redefined scope [...]

## Week 11: Midterm Presentation (2019-12-10 19:04)

Hey everyone!

This is our final post for our upcoming midterm presentation. It summarizes all important documents and such that are required by our stakeholder.

Have a good time and see you in the next Semester.

Your TrackYourFit-Team

---

Our Software:

- [1]Code (Git Repository)

Project Management on YouTrack:

- [2]Agile Dashboard
- [3]Issues list
- [4]Time Report
- [5]example burn down chart (Week 5)
- [6]example burn down chart (Week 9)

The Overview:

- [7]SAD (Software architecture document)
- [8]SRS (Software requirement specifications)
- [9]Class diagrams

Our Use Cases with their .feature files:

- [10]CRUD News
- [11]CRUD Trainingplan
  - [12].feature file Create Trainingplan
  - [13].feature file Delete Trainingplan
  - [14].feature file Read Trainingplan
  - [15].feature file Update Trainingplan
- [16]Track Trainingplan
  - [17].feature file TrackTrainingplan
- [18]Track Weight
  - [19].feature file Submit Weight

All the past blog posts:

- [20]Week 1: Track Your Fit
- [21]Week 2: Roles and Technologies
- [22]Week 3: SRS and UCD
- [23]Week 4: Use Case Specifications
- [24]Week 5: .feature files
- [25]Week 6: PM tool + Integration
- [26]Week 7: Retrospective
- [27]Week 8: Class diagrams & DB schema
- [28]Week 9: MVC & redefined scope

## Additional Links:

- [29]Midterm slides (.pptx)
- [30]Midterm handout (.docx)

1. <https://github.com/tobi4321/TrackYourFit/tree/website>
2. <https://dhw-karlsruhe.myjetbrains.com/youtrack/agiles>
3. <https://dhw-karlsruhe.myjetbrains.com/youtrack/issues>
4. <https://dhw-karlsruhe.myjetbrains.com/youtrack/reports/time/141-1?line=user>
5. <https://dhw-karlsruhe.myjetbrains.com/youtrack/reports/burndown/147-2>
6. <https://dhw-karlsruhe.myjetbrains.com/youtrack/reports/burndown/147-9>
7. <https://github.com/tobi4321/TrackYourFit/blob/master/SW-Documentation/SAD.md>
8. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/SRS.md>
9. <https://github.com/tobi4321/TrackYourFit/tree/master/SW-Documentation/ClassDiagrams%26DB>
10. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-News/UC-CRUD-News.md>
11. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-CRUD-Trainingplan.md>
12. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-Trainingplan-feature-create.feature>
13. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-Trainingplan-feature-delete.feature>
14. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-Trainingplan-feature-read.feature>
15. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-CRUD-TrainingPlan/UC-Trainingplan-feature-update.feature>
16. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackTrainingplan/UC-TrackTrainingplan.md>
17. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackTrainingplan/UC-TrackTrainingplan-Feature-TrackTrainingplan.feature>
18. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackWeight/UC-TrackWeight.md>
19. <https://github.com/tobi4321/TrackYourFit/blob/master/Projectmanagement/UC-TrackWeight/UC-TrackHealth-Feature-SubmitWeight.feature>
20. <https://trackyourfit.wordpress.com/2019/10/08/the-journey-begins/>
21. <https://trackyourfit.wordpress.com/2019/10/10/week-2-roles-and-technologies/>
22. <https://trackyourfit.wordpress.com/2019/10/22/week-3-srs-and-ucd/>
23. <https://trackyourfit.wordpress.com/2019/10/27/week-4-use-case-specifications/>
24. <https://trackyourfit.wordpress.com/2019/11/05/week-5-feature-files/>
25. <https://trackyourfit.wordpress.com/2019/11/12/week-6-pm-tool-integration/>
26. <https://trackyourfit.wordpress.com/2019/11/16/week-7-retrospective/>
27. <https://trackyourfit.wordpress.com/2019/11/23/week-8-class-diagrams-db-schema/>



28. <https://trackyourfit.wordpress.com/2019/12/02/week-9-mvc-redefined-scope/>
  29. <https://github.com/tobi4321/TrackYourFit/blob/master/Midterm/Midterm-Slides.pptx>
  30. <https://github.com/tobi4321/TrackYourFit/blob/master/Midterm/Handout.docx>
-



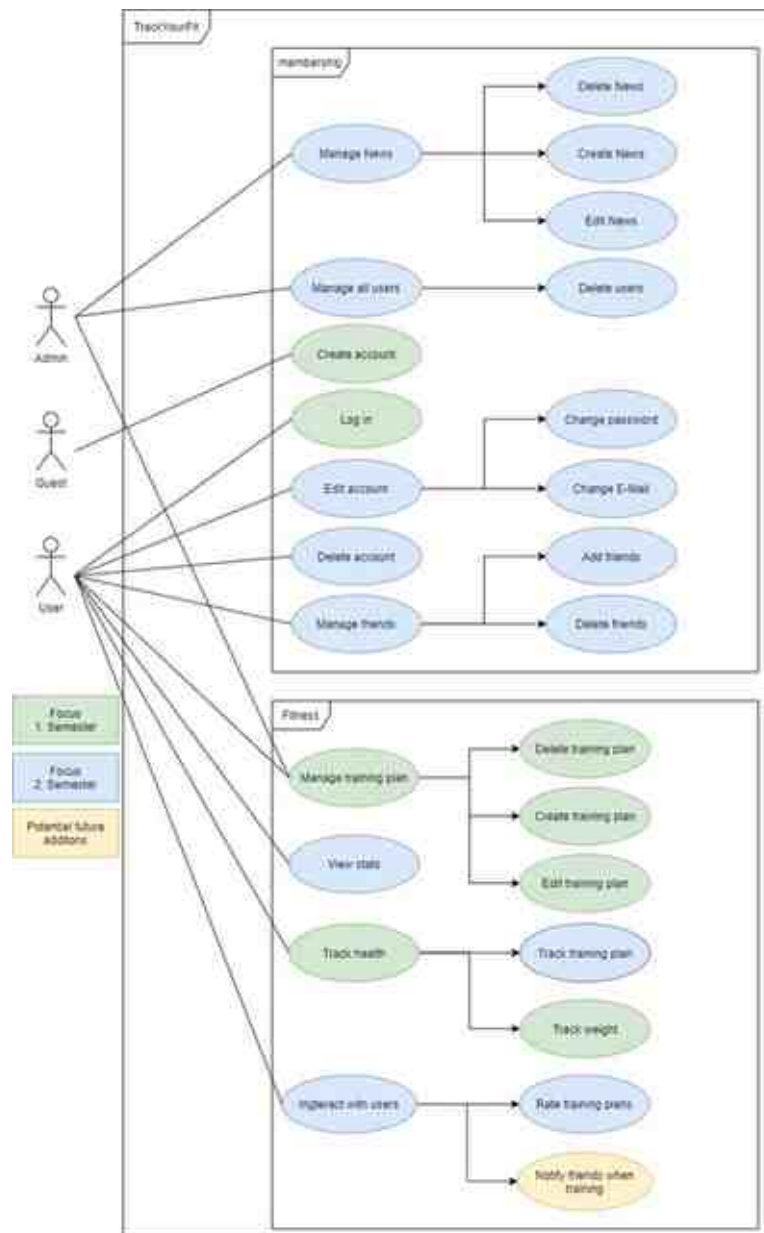
## 2. 2020

### 2.1 April

#### **Week 11: Redefining Scope (2020-04-22 11:12)**

This blogpost is all about us being anxious about how few time is left to finish the project. Therefore we redefine our scope and adjust some other environmental issues.

In the beginning we were keen on implementing a web-app and mobile-app to track training plans, now we are taking a step back. We will focus on implementing the web-application and making it (at least as far as possible) responsive. We also updated our UC-Diagram as you can see here:



As you can see the UC-Diagram is color coded. Regarding the point "notify friends when training", we have something like the app "Beer with me" in mind. This UC will only be realised when there's time left for it.

We also updated our git-repo, since our prior structure was really unhandy and chaotic. We now have two separate repos: one for the docs and one for code. This makes it a lot easier to develop and show others our code.

Thanks for checking in!

easyFinance Team (2020-04-29 09:37:58)

Hey Trackies, I appreciate that you want to focus at one channel, so in the end there is at least on functional application. Also I hope that you get all UCs done because there are more to finish than you did in the last semester, but there es estimation incoming, so I'm looking forward to them. According to that I think it has been the right choice to reduce your UCs by making 'interact with friends' optional. stay tuned an healthy, Nico - easyFinance

## Week 12: Risk management (2020-04-26 17:57)

Hello everyone,

this week we looked at our risks to discover our possible points of failure. So that we can try to migrate them and minimize the risk of their occurrence.

ID	Risk Name	Risk Description	Risk Probability of Occurrence	Risk Impact	Risk factor	Risk migration	Person in charge of tracking
1	Not enough time	Not finishing our scope of this semester	60%	10	6	Careful planning of the sprints.	All
2	Wrong Priorities	Not focusing on the scope.	35%	7	2,45	Not getting carried away by smaller, less needed tasks.	All
3	Underestimated complexity	The set goals are too hard to achieve	30%	5	1,5	Self-reflections and changes when sprints are not completed.	Toni Einecker
4	Coronavirus	Unexpected circumstances (e.g. internet failing, serverhosting platforms not available)	15%	8	1,2	Self hosting of a server, other than that there is nothing we could do.	Tobias Bühler
5	Unreliability of a team member	A team member doesnt complete his work.	15%	7	1,05	The other team member have to pit up the undone work and complete it.	Janis Fix
6	Dataloss	Loss through a github mistake or error.	5%	8	0,4	Careful git handling.	Johannes Zipfel
7	Loss of team members	Losing a team member (e.g. due to failing the exams)	5%	5	0,25	Everybody should have knowledge about the whole code.	All

Our biggest point of failure is the wrong planing and therefore not having enough time to finish the work in this semester.

As second we created a table to see our spend time per use case:

ID	Use Case	Documentation (h)	Coding (h)	Testing (h)	Warm-Up time (h)	Total (h)	FP
1	CRUD News	2	2,5	0	2	6,5	?
2	CRUD TrainingPlan	3,5	14	0	2	19,5	?
3	Track Training	2	4	0	5	11	?
4	Track Weight	2	1,5	0	4	7,5	?

Our testing time is zero because we couldn't figure out gherkin to work for our project. We are looking forward to using unit tests, since we will have a lecture about them in the future.

See you next week.  
The TrackYourFit-Team

---

Pascal Stephan (2020-04-26 19:37:29)

Hello TrackYourFit-Team, We think you did a good job this week. Your risk management table and your use case overview are clearly understandable. Maybe you can resize your risk management table to make it easier to read. Kind Regards,  
AniflixApp-Team

TrackYourFit (2020-04-29 06:51:36)

Heyho, Thanks for your feedback. We did change the embedded image to a gallery, so that you can just click on it. Best Regards, TrackYourFit-Team

Felix (2020-05-06 08:40:19)

Hello TrackYourFit-Team, your risk-management seems to be on point, you identified a lot of risks and evaluated their impact on your project. Also good to see you working on tests or at least keeping them in mind. Keep up the good work. Best Regards,  
Maphynn

## 2.2 May

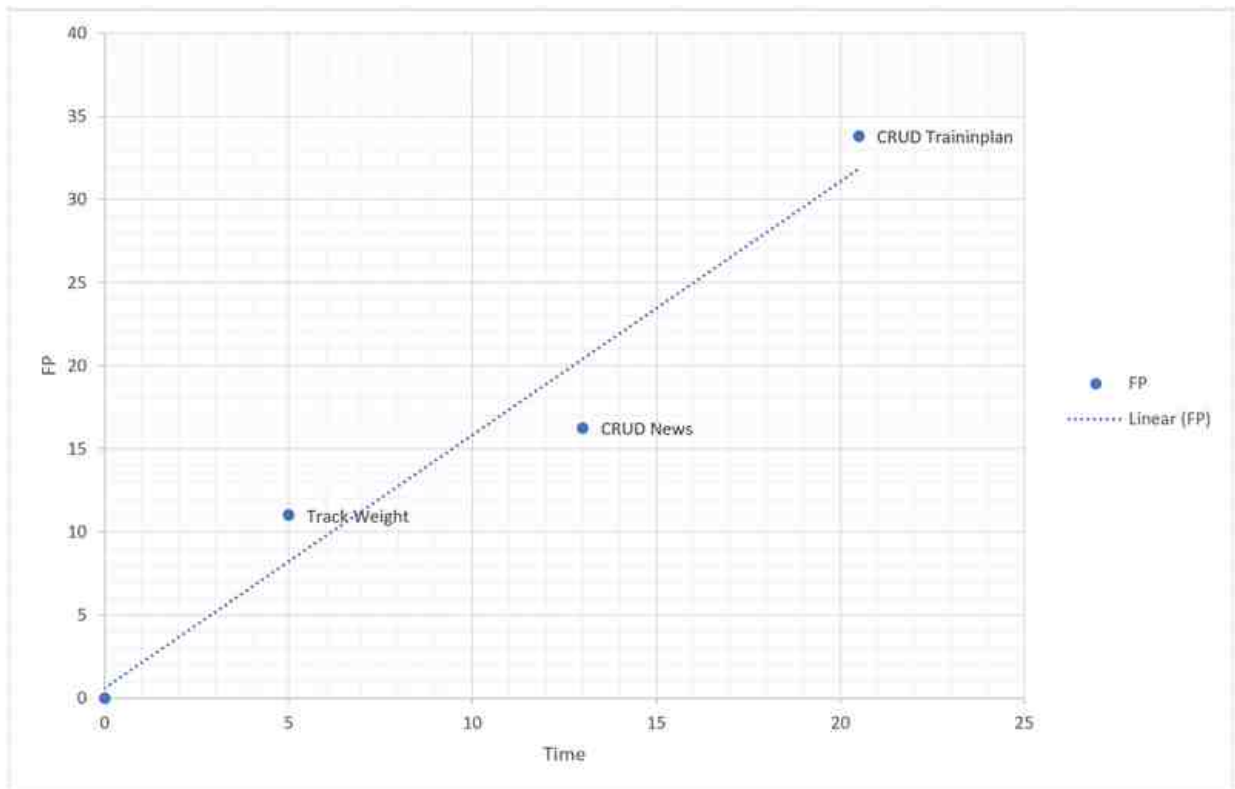
### Week 13: Function Points (2020-05-02 13:33)

Hello everyone,  
this week we calculated our function points of our current use cases. This is needed to measure our functional size of the software.

In this table you can find the use cases with their FP and Time we spend on it:

UC	time	FP
	0	0
Track Weight	5	11
CRUD News	13	16,25
CRUD Traininplan	20,5	33,8

With the help of this data we created the following graph:



You can see that we are balanced in our UCs.

See you next week.  
The TrackYourFit-Team

---

lilolunatic (2020-05-02 14:50:56)

Dear Trackyourfit-Team, Good job in tracking your spent time and measuring your implemented usecases :) Have you also estimated your upcoming usecases? It would be nice to know what you will during the semester and if your pensum is manageable :) Best, Your Orchestra-Team

partyplayerteam (2020-05-02 15:56:00)

Hey guys, your graph and table look nice. But I am missing the information about the process, how you got to your results. What exactly it is that you calculated, how you did it (...). Like Orchestra-Team already mentioned, it would be nice to see, what you are going to do next and what the time estimation is. Maybe you already did that somewhere but forgot to provide the link? Regards, Seba@PartyPlayer

Dennis (2020-05-03 13:08:47)

Dear trackyourfit Team, we like the upcoming UC., and we'd love to see more about the following training plans. Other than that we have to agree with the other commenters, it would be really useful to see what is going to follow and how much time you plan to spend on it and a quick introduction into FP-Calculation might be awesome for people who've never heard of it before. Maybe that's just personal opinion, but you could try to resize the table to align it with the text, we think it could improve the overall aesthetics. Keep the good work up. Stay cool. Stay organized. – Your Fridgify Team

## **Week 14: Unit Testing (2020-05-09 17:08)**

This weeks blogpost is all about unit testing. (Automatic) Testing is really important for any kind of application. Recognizing fatal bugs, that would cause the application to stop working (before sending the new version into production) is only one upside of testing.

After having struggled with setting up Gherkin and the .feature files for our project, we were keen on implementing unit tests. We are using the xUnit framework to test our ASP.NET Core web-app. We are by no means experts in implementing unit tests, so if anyone has already worked with xUnit framework, feel free to hit us up!

What we managed to do though, is setting up some basic tests. It's integrated into our IDE (Visual Studio) like this:



▲	✓	TrackYourFitTest (3)	378 ms
▲	✓	TrackYourFitTest (3)	378 ms
▲	✓	AccountControllerTest (2)	342 ms
	✓	AccountController_RegisterFailure	339 ms
	✓	AccountController_RegisterSuccessfull	3 ms
▲	✓	HomeControllerTest (1)	36 ms
	✓	HomeController_ReturnsAViewResult	36 ms

•

Especially testing with the Entity Framework Core (which we use for our database access) and the Identity Framework (User/Login-stuff) isn't quite the easiest, but we're getting there!

Our test code can be found on github: [1]<https://github.com/tobi4321/TrackYourFit/tree/master/TrackYourFitWeb/TrackYourFitTest>

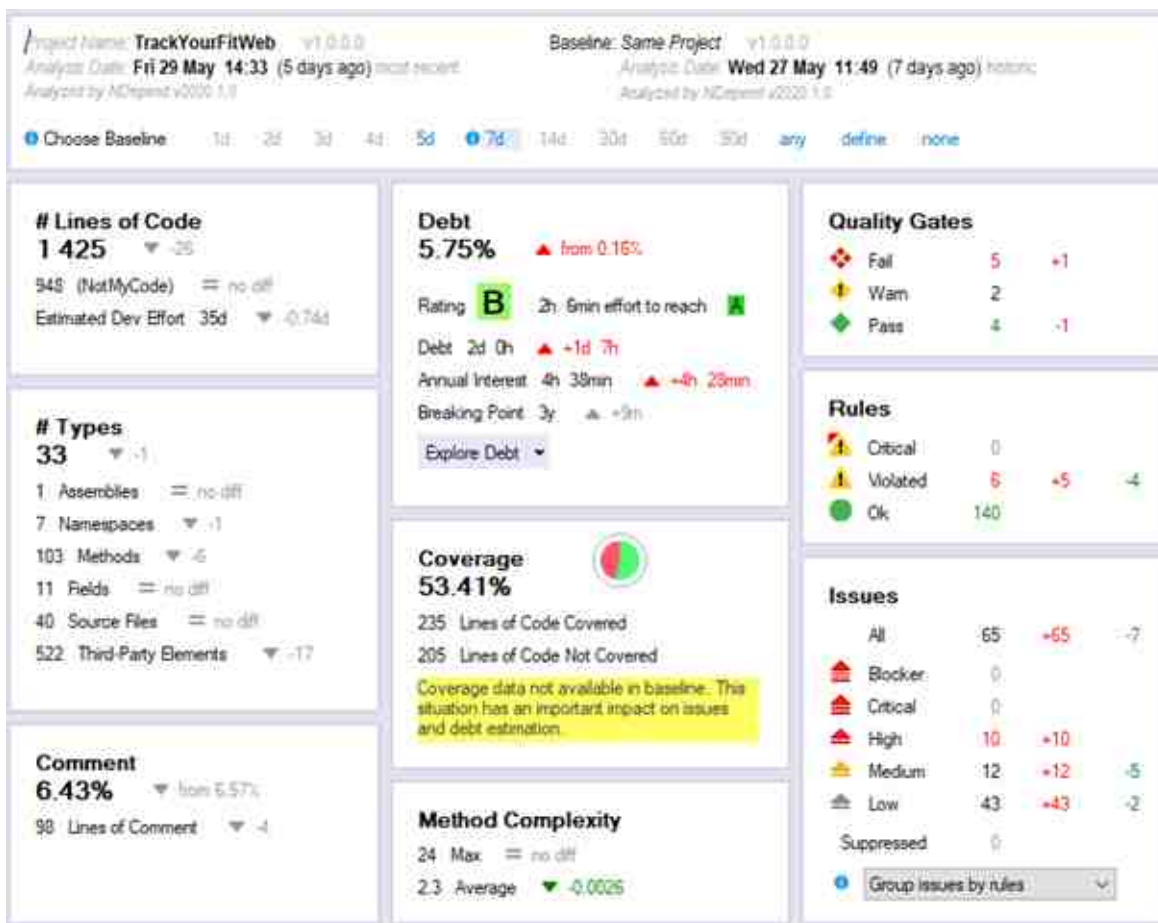
And here's our Test Plan:

[2][https://github.com/Bronzila/TrackYourFit\\_docs/blob/master/Projectmanagement/rup\\_tstpln.md](https://github.com/Bronzila/TrackYourFit_docs/blob/master/Projectmanagement/rup_tstpln.md)

Moreover we have started getting into Jenkins and are looking forward to using it in the near future.

## Update Week 18:

We are able to look at our Code Coverage through our Metric tool NDepend, it's displayed on their dashboard as follows:



As you can see we are currently sitting on a code coverage of 53 %, we will try to improve it as much as we can though!

Moreover, we setup Jenkins to automatically build and test the latest commits on our master branch, depending on the result we will either receive a green check or a red cross:



That's it for this weeks blogpost. Feel free to leave feedback to our work!

Have a good one,

Your TrackYourFit-Team

1. <https://github.com/tobi4321/TrackYourFit/tree/master/TrackYourFitWeb/TrackYourFitTest>
2. [https://github.com/Bronzila/TrackYourFit\\_docs/blob/master/Projectmanagement/rup\\_tstpln.md](https://github.com/Bronzila/TrackYourFit_docs/blob/master/Projectmanagement/rup_tstpln.md)

---

RawBean (2020-05-09 17:54:29)

Hey guys, your tests look really good, as far as you have done them. Think of implementing all test before end of the semester as we need 3 different methods. I really like you plan of testing you made. It looks complete when it's done. Keep it up, I like your ideas. Cheers, RawBean (Foody, B4)

TheFriendsLook (2020-05-09 18:25:43)

Hey trackyourfit Team, i think we have the same problems with our .feature and behavior development tool. but even that ya also struggled with this things .. it looks really good what you did with the planning and tests .. i think it looks different with us because we are developing an app with android studio. I hope that I hope that we can do everything on time and that you get on well. Stay save! Mo TheFriendsLook

partyplayerteam (2020-05-11 13:05:25)

Hey guys, haha gherkin ... we know that struggle as well. But as long as you got something working, you are on a good path to improve the quality of your tests and with that also the quality of your code. Keep your heads up and you will be fine. "We are by no means experts in implementing unit tests". That's not a problem. Like you said you are keen to implement unit test after gherkin, so as long a s you keep the spirit you will learn fast and maybe some deay you will be an expert ;) I personally like the part with jenkins. After setting it up properly it will be a smooth operation and you wont get any headache afterwards. Rely on your automation and you can focus on the important parts. Good work! Reagrds, Seba@PartyPlayer

## **Week 15: Refactoring (2020-05-16 16:40)**

Hey guys,

this weeks task, was to refactor an intentionally bad written code in Java. In order to improve the code, we added JUnit tests and refactored the code, by using IDE Tools of the IDE Eclipse.

These are links to our git reposetories, so you can look up the steps we took.

- [1]Toni's reposetory
- [2]Janis's reposetory

- [3]Johannes's repository
- [4]Tobi's repository

For refactoring your code, Eclipse gives you a collection of tools to use. On [5]this site there is a in depth summary of those tools.

1. <https://github.com/EinToni/Refactoring>
2. <https://github.com/Bronzila/FowlerRefactoring>
3. <https://github.com/johzip/refactor>
4. <https://github.com/tobi4321/RefactoringSWE>
5. [https://wiki.fernuni-hagen.de/eclipse/index.php/Refactoring-Tools\\_in\\_Eclipse](https://wiki.fernuni-hagen.de/eclipse/index.php/Refactoring-Tools_in_Eclipse)

---

Nathalie (2020-05-18 08:19:07)

Hi TrackYourFit team, your repositories have tests and commits with description how you improved the code, which is good. Only Johannes repository just has 2 commits by now with only tests added. Maybe you want to pick a refactoring feature of your IDE out of the linked wikipedia-site to describe here. Keep up the good work. Best regards, Nathalie from Orchestra-Team

TrackYourFit (2020-05-27 15:21:15)

Hey Nathalie, Thanks for the feedback! You are right about Johannes repo, since it was not finished at the time of your feedback. Now it is :D Best regards TrackYourFit

Evendo (2020-05-18 09:06:07)

Hi TrackYourFit, thanks a lot for your blog entry. All of your codes are looking pretty good. I like that you have implemented testing in your code. Just a short question: who of you uses which IDE - just for my interest :)? Have a great day Cheers Evendo

TrackYourFit (2020-05-27 09:38:05)

Hey Evendo, Thanks for the feedback we all used Eclipse, since we all normally write in C # and visual studio TrackYourFit

## Week 16: Design Patterns (2020-05-24 13:08)

This week we took a look at design patterns to implement in our code. For our backend we decided to implement dependency injection and for our frontend we wanted to tidy up our JavaScript-Code by modularizing our AJAX-Calls.

## Frontend

Let's start with the modularization of AJAX-Calls. If you have ever worked with jQuery and AJAX-Calls you might know that every call has a similar structure. We found ourself copy-pasting alot, so that's why we decided to modularize these calls.

Before we were copy and pasting like this:

```
7 function addTP() {
8     var name = $('#TPname').val()
9     $.ajax({
10         type: "Post",
11         url: "/Trainingplan/CreateTP",
12         cache: false,
13         data: {
14             TPname: name
15         }
16     }).done(function (msg) {
17         editTP(msg);
18     });
19 }
20
21 function addExercise(id) {
22     var exname = $('#exerciseName').val()
23     var exdesc = $('#exerciseDescription').val()
24     $.ajax({
25         type: "Post",
26         url: "/Trainingplan/AddExercise/",
27         cache: false,
28         data: {
29             TrainingPlanId: id,
30             exName: exname,
31             exDesc: exdesc
32         }
33     }).done(function (msg) {
34         $("#exercises").html(msg);
35     }).fail(function (msg) {
36         alert(msg.responseText);
37     });
38 }
```

What we did was adding a new function that's returning the AJAX-promise, since the action performed when the call is done varies.

Here's our after view:

```
27 function postData(dataUrl, useCache, data) {
28     return $.ajax({
29         type: "Post",
30         url: dataUrl,
31         cache: useCache,
32         data: data
33     })
34 }
35
36 function addTP() {
37     var name = $('#TPname').val()
38     var data = {
39         TPname: name
40     }
41
42     postData("/Trainingplan/CreateTP", false, data)
43         .done(function (msg) {
44             editTP(msg);
45         });
46 }
```

## Backend

Now for our backend we implemented dependency injection. Our framework ASP.NET Core took us by its hand and guided us through it, since it has a few built-in features that make working with dependency injection easier.

As a short description: When using dependency injection, clients (= the class using an object) won't create the objects they need on their own, the objects (= services) will be injected by an (guess what) injector. One advantage is of this

method is, that the client is much more flexible and configurable. Only the behaviour of the client is fixed, but it can act on anything that supports the interface it expects. Another upside is, that unit testing these clients is much easier, since you can mock the services, that are going to get injected into the client.

As an example we will look at our `TrainingPlanContext`, which we use to access our database.  
Before refactoring we instantiated the context-object in every controller it has been used in like this:

```
private TrainingPlanContext context;
0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
public NewsController()
{
    context = new TrainingPlanContext();
}
```

Now that we included dependency injection in our project, we can add it to the services in our `Startup`-class and we even made it configurable by adding what SQL-Server should be used:

```
public void ConfigureServices(IServiceCollection services)
{
    services.Configure<CookiePolicyOptions>(options =>
    {
        // This lambda determines whether user consent for non-essential cookies is needed for a given request.
        options.CheckConsentNeeded = context => false;
        options.MinimumSameSitePolicy = SameSiteMode.None;
    });
    services.AddIdentity<ApplicationUser, IdentityRole>()
        .AddEntityFrameworkStores<TrainingPlanContext>();

    services.AddDbContext<TrainingPlanContext>(
        options => options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
    services.AddSession();
}
```

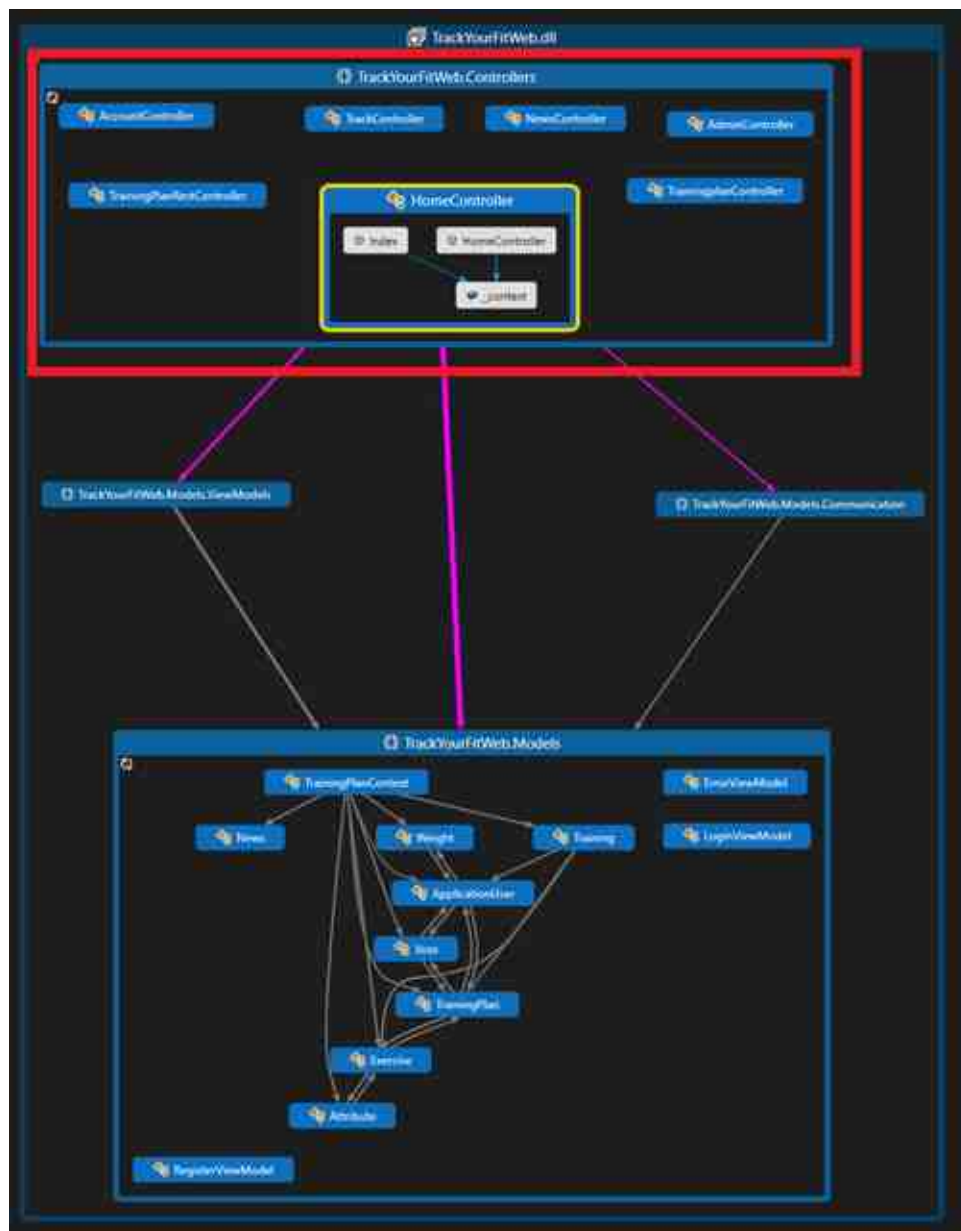
In the Controller-class we wont need to instantiate it now, we will just let it get injected:

```
private TrainingPlanContext context;  
0 references | Bronzila, 8 days ago | 1 author, 1 change | 0 exceptions  
public NewsController(TrainingPlanContext con)  
{  
    context = con;  
}
```

*Sadly there is no difference in the code map (UML-Diagram of VS), so that's why we couldn't provide a screenshot of the UML. :(*

**Update:**





On the so called "Code Map" we highlighted the spots where the dependency Injection is used and as you can see it's in every controller! The reason for that is, that we are either using some kind of Identity-Framework (our User-Login/Logout Framework) objects or our context object (needed to connect to the database), as shown in the unfolded HomeController above.

Thats it for this week, feel free to leave feedback!  
Have a good one.

Your TrackYourFit-Team

---

The hopper team (2020-05-24 13:51:16)

Dear trackyourfit-Team, your blog post looks great. It is very nice that you show real code in your post. So it is a lot simpler to understand your refactoring and use of the design patterns. The choice of design patterns is also very good. Although your class diagram has not changed, it would be good to show and mark where you have added the design patterns. Also, it would be great to add a reference to the code version before and after adding the design patterns. All in all a very good blog post. We are looking forward to your further progress! Best regards, The hopper team

TrackYourFit (2020-06-10 06:54:10)

Heyho, thanks for your feedback, as you mentioned, we added the VS-Class Map and highlighted the parts where our pattern is being used. Have a good one!

CardGameAppTeam (2020-05-26 06:10:31)

Hi there, your post is awesome as far as I can tell. Even I have to fight a bit with Ajax ones in a while and at some point the same idea popped into my head, regarding the reusage of the Ajax function, so nicely done on this one. As a matter of fact I also happen to develop a lot in .net framework, and it is nice that it guides you threw with a pattern which is used a lot in todays development. As the post above my one also stated, I'd like to see a little UML diagram with the class highlighted that would be nice. And like you did with the Ajax example, maybe a little code snippet of the dependency injection (before and after), or did you have the dependency injection in the code beforehand? All in all I like your post very much. You mentioned everything I wanted to hear and even have given a little explanation about dependency injection. So if someone had no idea, he should know by now. Nicely done. Cheers.

TrackYourFit (2020-06-10 06:57:58)

Heyho, thank you for your great feedback! We updated our post and included the VS Code-Map with the highlighted segment. It seems crazy on first sight, but the .NET Core Framework helps you a lot when working with dependency injection, the code we have showed in the section "backend" is all we needed to do. Have a good one!

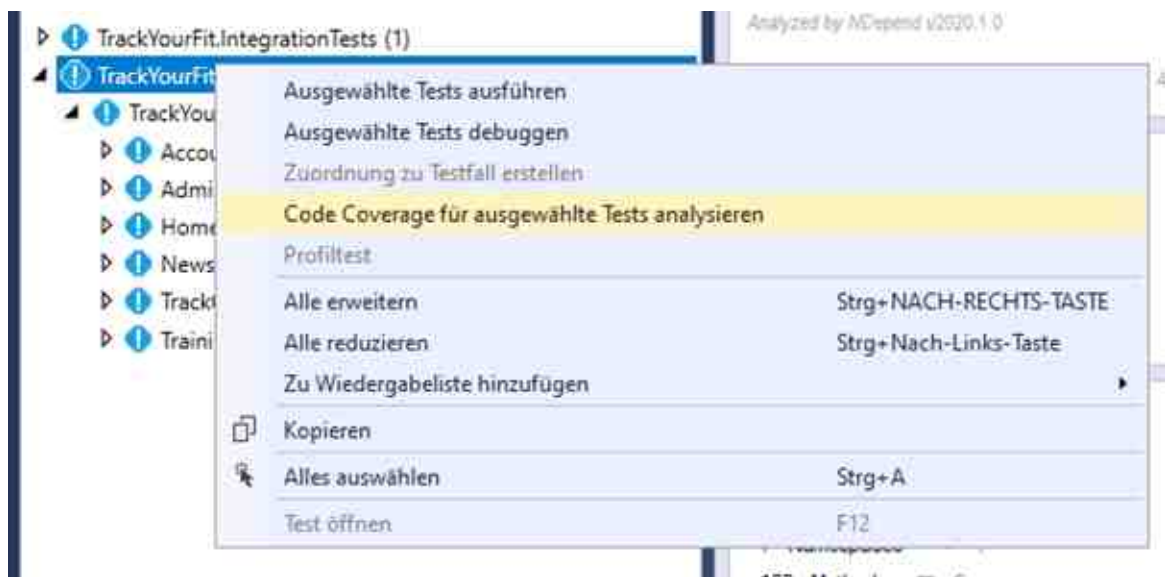
## 2.3 June

### Week 17: Code Metrics (2020-06-04 11:33)

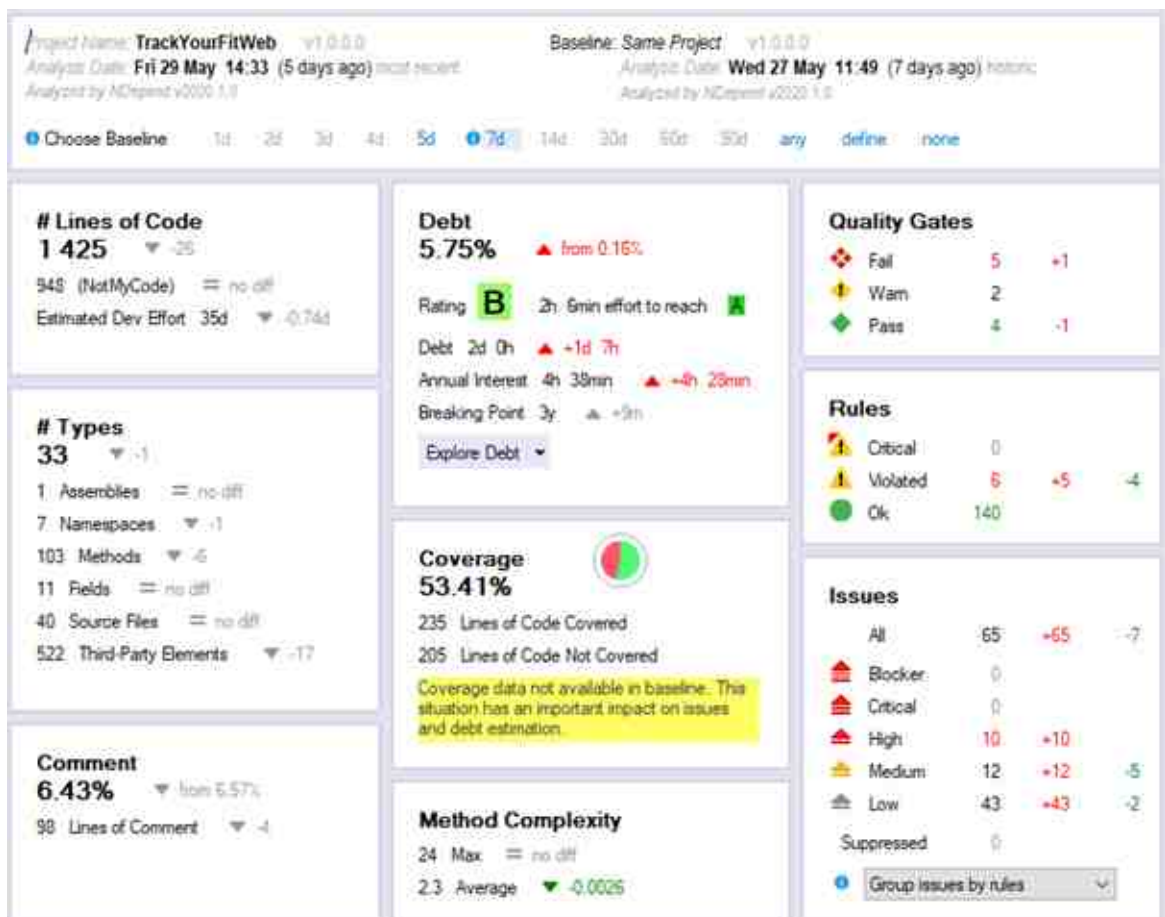
In the last 2 weeks we have started getting into NDepends. NDepend is a static code metrics tool for .NET/.NET Core projects.

After setting up NDepend and linking it to our project, we had to create a code coverage file for NDepend to analyse.

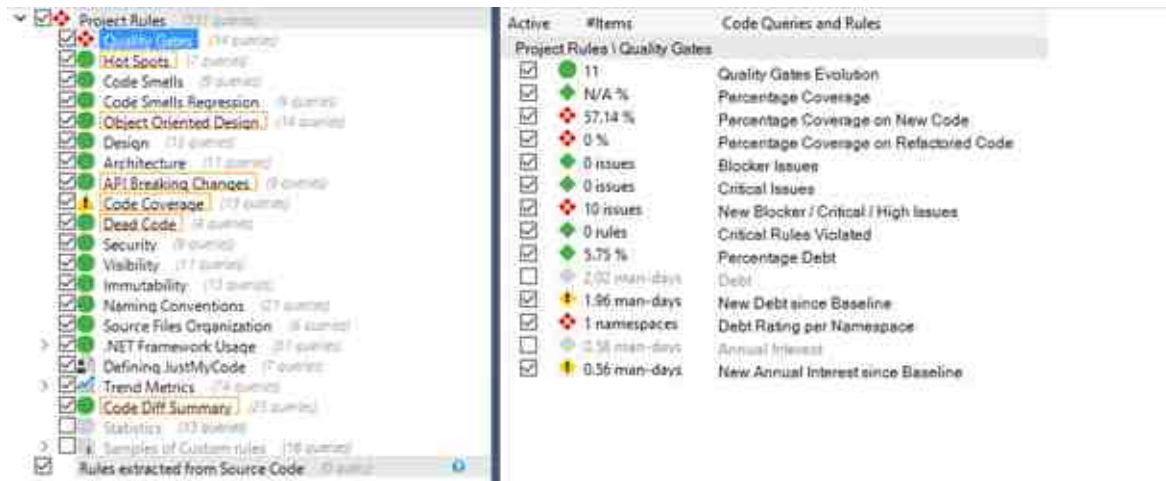
In VS you can create one by right clicking the tests, you want to have a code coverage file created for:



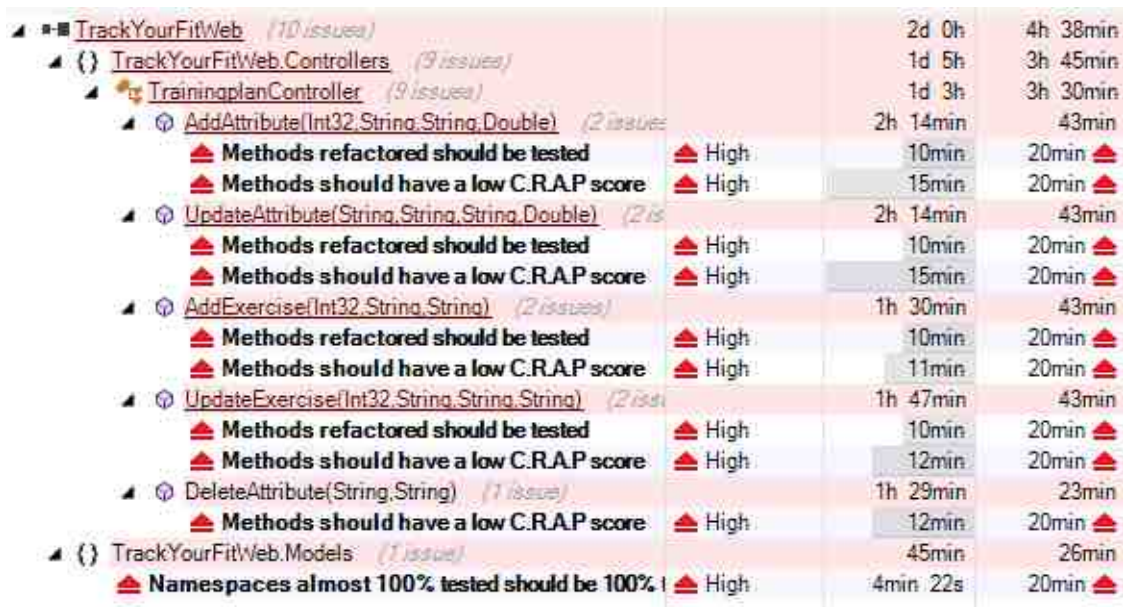
When the Code Coverage has been created, you simply need to add the file to your NDepend Project. Now you can look at your dashboard, where the Code Coverage and all other analysed values can be found:



As you can see we have a coverage of 53 %. The reason for that is, that we haven't implemented tests for our Models, since these are classes with just their attributes and getters/setters. Since we don't have that much time left, we aren't going to implement tests for the models, since they aren't useful. Sadly as a result, we have some Quality Gates, since there's some code that's not tested, but more on that later...



The figure above shows our used Rules, on the right side you can see the precise descriptions for the Quality Gates.



The "New Blocker / Critical / High Issues" are the problems above. As mentioned above, that's the case since we aren't testing our models and since we have 5 Methods in one of our Controllers, that we aren't going to test, since we are using a session-extension, which is awful to mock in unit tests (trust me, I still have nightmares of trying to do it).



What we realised is, that we should've started using NDepend right from the start, since our graph wont show big changes/improvements.

Now let's look at some of our improvements for the metrics...

One of the issues were naming conventions, since methods in .NET Core should always start with an uppercase letter:

```
private bool attributeNameAlreadyTaken(List<TrainingPlan> plans, int trainingPlanId, string exName, string attrName)  
private bool AttributeNameAlreadyTaken(List<TrainingPlan> plans, int trainingPlanId, string exName, string attrName)
```

The next issue we attacked, was the immutability. It indicates, that some specific objects, for example our context-object we use for reading our Database, aren't allowed to be modified in Controllers. This can be fixed by adding the readonly keyword to the controller attribute:

```
public class TrainingPlanController : Controller
{
    private TrainingPlanContext context;
    private readonly TrainingPlanContext context;
    private readonly UserManager<ApplicationUser> _userManager;
```

Another Issue was, to remove unused code. We actually found a class, that isn't in use anymore, but hasn't been removed yet:

```
1  -using System;
2  -using System.Collections.Generic;
3  -using System.Diagnostics.CodeAnalysis;
4  -using System.Linq;
5  -using System.Threading.Tasks;
6  -
7  -namespace TrackYourFitWeb.Models.Communication
8  -{
9  -    [ExcludeFromCodeCoverage]
10 -    public class TrainingHelp
11 -    {
12 -    }
13 -
14 -}
```

Our conclusion to NDepend:

NDepend is a good tool for analyzing .NET/.NET Core projects, but it is almost mandatory to use it since the beginning of the project to get all Values/Metrics calculated properly. Moreover NDepend is really powerful, since there are a lot of predefined rules, which can also be expanded by your own rules.

#### UPDATE:

NDepend didn't meet ours (and Prof Berklings) expectations. What seemed to be a software for calculating high level metrics is rather a software for enhancing some smaller metrics for your code. Nevertheless it is enough to fulfill the grading criteria.

We're currently on the Finishing Straight and try to add as many features and improvements as possible to our application, so stay tuned!

Your TrackYourFit Team

---

lumanuel (2020-06-06 08:02:44)

Hi TrackYourFit Team, nice post! I like your detailed description of your refactoring. Unfortunately I miss a link to your code where you have been refactored. I'm not sure about your code metrics either. Does the code coverage count as a software metric (I don't know)? Nevertheless, I would like it if you could explain the two metrics you use, as this is also part of the grading criteria. Kind regards Manuel from Orchestra

TrackYourFit (2020-06-30 07:20:19)

Heyho, thanks for your feedback! We were pretty unsure about these metrics from NDepend as well. We did check it with Prof Berkling though and she told us, that she was expecting more from NDepend aswell. But for now it's fine fixing these minor issues/metrics for full points. Kind Regards, Janis from TrackYourFit

annoua (2020-06-06 10:23:59)

Hi Track your Fit, it's very nice how you explain how you fixed the issues. According to the grading criteria, you should explain two code metrics, you analyzed and fixed, and a spot, where you don't want to fix the issues and why. Regards, OneTouchNextGen

TrackYourFit (2020-06-30 07:22:01)

Hey, thank you for your feedback! As stated in our answer above, the expectations for NDepend were high, but couldn't NDepend couldn't meet them. Best Regards, TrackYourFit

#### **Week 19: Installation (2020-06-23 18:13)**

This weeks blogpost is going to be short and all about installing the proper IDE for developing our application. This can be found here.

We also added a document showing our deployment on IONOS. This can also be found here.

Sadly we couldn't include the deployment to our jenkins, since we are using IONOS and combining these two would take **a lot** of time.

Our installation will be performed by the group "TheFriendsLook" of our course.

That's it from us, stay healthy and

*Track your fit*

---

thefriendslook (2020-06-29 11:10:30)

Installation: We installed the application to our pc's. The installation was very easy, we only had to Install VisualStudio and the .net Core package. After the download of the Git repository we were able to load the projekt folder by Visual Studio. We started the application and the web browser opened. In the web browser we could log in with the admin Password and user name. After the login we could see the different menu items. We test to create a new training plan and how to add different exercises to them. We also checked out the administrator surface in which you get an overview over all user activities. All functions we had tested worked very well and it was great to see how many functionality the group realized in the short time. Best wishes TheFriendsLook





BlogBook v1.2,  
 $\text{\LaTeX}$  2 $\epsilon$  & GNU/Linux.  
<https://www.blogbooker.com>

Edited: July 14, 2020

