



Free version: Low quality pictures

[perfecttime608150251.wordpress.com](https://perfecttime608150251.wordpress.com)



# Contents

<b>1</b>	<b>2018</b>	<b>5</b>
1.1	October . . . . .	5
	Hello (2018-10-05 15:49) . . . . .	5
	Breaking Down our Application (2018-10-12 10:28) . . . . .	8
	Defining Roles (2018-10-13 15:17) . . . . .	11
	Defining our software requirements (2018-10-21 23:59) . . . . .	14
	Use Cases - Getting closer to our application and to Cloud Nine (2018-10-28 17:42) . . . . .	18
1.2	November . . . . .	21
	Testing and .feature (2018-11-03 23:00) . . . . .	21
	Creating our class diagram and our database schema (2018-11-10 18:07) . . . . .	23
	Scrum, YouTrack and Coding (2018-11-18 18:05) . . . . .	26
	Working with the (Scrum) professionals (2018-11-25 01:10) . . . . .	30
1.3	December . . . . .	34
	Coding like mad (wo)men and describing our architecture (2018-12-02 19:26) . . . . .	34
	The final push (2018-12-09 17:46) . . . . .	38
	Checkpoint reached - handing in the midterm documents (2018-12-23 18:40) . . . . .	40
<b>2</b>	<b>2019</b>	<b>45</b>
2.1	April . . . . .	45
	Back on (You)Track (2019-04-07 12:00) . . . . .	45
	A risky business? (2019-04-14 19:55) . . . . .	47
	Yes, we can! (finish our defined scope within this semester) (2019-04-21 12:00) . . . . .	49
	Working on the next Unit... and Testing it. (2019-04-28 21:43) . . . . .	52
2.2	May . . . . .	56
	Learning Refactoring from M. Fowler (2019-05-05 18:00) . . . . .	56
	"I don't trust words, I even question actions, but I never doubt patterns." (2019-05-12 18:00) . . . . .	58
	Looking back and moving forward (2019-05-19 18:00) . . . . .	62
	Getting the numbers right - working on our metrics (2019-05-26 18:00) . . . . .	68
2.3	June . . . . .	73
	Testing in a nutshell (2019-06-02 14:56) . . . . .	73

Getting ready for the grand finale (2019-06-02 16:12) . . . . .	82
Creating your own Perfect Time - An installation guide (2019-06-09 21:37) . . . . .	86
It's looking good guys - our little ad video (2019-06-11 00:13) . . . . .	91
The grand finale - We made it! (2019-06-16 17:00) . . . . .	93

# 1. 2018

## 1.1 October

Hello (2018-10-05 15:49)



Hey Guys,

last week, at the fifth of October, we built Teams to realize an agile project in our course "Software Engineering I" at DHBW Karlsruhe. This is the blog of Team 7. We will build an Application called "Perfect Time – plan your trip". We want to invite you to join us on our journey from the draw board to the finished software. This blog entry will give you insights into our first project meeting.

At first, we brainstormed the project. What would be an innovative idea for a cool new Application? We considered several topics, including event planning support and e-learning software. But in the end, we decided to solve a problem that we all shared: Planning the next holiday trip.

This problem is probably known to most of mankind. Before you start traveling you want to plan which locations you want to visit, what you want to pack and what activities you want to participate in. So, you start searching on the Web. You use TripAdvisor, Google Places and travel blogs. You start watching YouTube videos, reading travel guides and making notes on everything. Then in the next step you start organizing the material and planning your timetable. But there is another problem: Which software should you use? There are only a few software

solutions available on the market. You would probably go with a more generic solution, such as OneNote.

With our software this challenge should become way easier. Our vision is to create an Application that allows you to plan your trip by setting different locations or stages. You will also be able to insert generic data for the whole vacation. After your holiday you can add a blog in the App using the already available data from planning your trip. Afterwards you can publish your trip and share it with your friends. They can use your plan as an outline for their own trip by simply importing your published data.

After having decided which problem our Application will solve, we thought about how our project should look like. Should it be a Smartphone, a Web or a Desktop app?

Based on the assumption that most people will plan their trip on a PC and not on a Smartphone, we decided to build a Web App. Besides the obvious fact that it is online, it also provides the clarity of a desktop App. We will use the React-Framework to build our Application. This will help to create a modern UI while using a MVC-Architecture.

We will use Scrum to organize our Project and since this is our first blog entry, we would like to introduce the team.

Product Owner: Tim, our data expert and passionate firefighter. He remains calm even under the highest pressure and he will keep our data safe.

Scrum Master: Lea, our power woman and organization talent. She will keep all of us in line and the process running smoothly.

Leading Developer: Lukas, our UI-expert and a handball enthusiast. He will make our application look good and inspire us all with his effort.

Leading Tester: Jan, former business-major and our documentation specialist. He will make sure our product meets its requirements and runs smoothly.

We would be pleased if you would follow our journey and test our Application. You can also look at our code, which we keep in a public Git Repository. It can be found at [1]GitHub.

Your

Perfect Time Crew

---

1. <https://github.com/Tallround3r/PerfectTime>

EventLAB (2018-10-10 11:07:27)

Hey Guys, after reading your blog we think that you have chosen a good idea for your project. As you said, we also experienced the problem of planning a trip and everything that comes with it. Therefore we are very interested in your project and of course your progress. We are excited to get some insights and at the end to see your result. Cheers, Your EventLAB team

perfecttimese (2018-10-10 11:21:47)

Thanks Guys, we are happy that you like our idea and we will keep sharing more and more insights, as our project progresses. We would appreciate if you keep following our progress and sharing your ideas and remarks with us. Your Perfect Time Team

commonplayground (2018-10-10 11:14:15)

Hello there, thanks for all the information about your project. We like your idea to make an application where you can plan your trip easily. We are not sure how your application is going to work. We would like to know what your subsystems are and what task or function they are going to fulfill. Also you could sum up your vision and make it more precise. Keep up the good work. We are interested in further updates :) Sincerely, CommonPlayground

perfecttimese (2018-10-10 11:20:18)

Thank you for your comment. We are glad that you like our idea. Regarding our vision and the possible features, we will go into further detail with our next blog entry. So stay tuned :D Regarding the subsystems of our application, we are still in the process of determining what they will look like. As soon as we know more, we will tell you. We would appreciate if you keep following our progress and sharing your ideas and remarks with us. Your Perfect Time Team

ChristianSchweigel (2018-10-10 11:21:20)

Hello Perfect Time team, your application idea sounds very interesting and useful. Planning a trip with a centralized tool would be great to save time. Are there going to be features to plan trips for groups, so friends can plan together one trip? I'm looking forward to see your progress over the next weeks. Sincerely, Christian

perfecttimese (2018-10-10 11:23:55)

Hi Christian, we are glad that you like our idea and share our passion for travelling. Thank you for sharing your idea with us. We like it. We will add it to our project backlog. We would appreciate if you keep following our progress and sharing your ideas and remarks with us. Your Perfect Time Team



Hey Guys,

last week, we told you about our project idea, our team and our vision. This week we did work on our project setup. We installed node.js, WebStorm and YouTrack, so we can start working. We also started to exchange our impressions with the other project from our class. It became obvious, that we will have to split our idea into smaller subsystems. In this blog post, we want to describe the subsystems we identified.

- Basic Web Application
  - Goal: Build a running website that provides our basic application.
  - Part 1: Setup a server.
  - Part 2: Provide a database for the user data.
  - Part 3: Design the application layout.
  - Part 4: Design deployment mechanism.
- Location Pages
  - Goal: Allow subpages for individual locations / events. One page should contain all the relevant data (address, date, map, etc.) on one location.
  - Part 1: Provide subpage skeleton and data structure for a single location.
  - Part 2: Implement page design form.
  - Part 3: Link pages.
  - Part 4: Add navigation between pages.
  - Part 5: Implement media support.



- Vacation Info
  - Goal: Allow general information for the whole vacation and statistics. I.e. your flight schedule, your budget, etc.
  - Part 1: Design data structure.
  - Part 2: Design overview subpage.
  - Part 3: Implement interface for usage of the data on other subpages.
  - Part 4: Implement statics feature (km travelled, € spent).
- Travel Blog
  - Goal: Allow redesign of location subpages into a travel blog. The individual location pages will look like a travel blog (pictures, videos, text, ...).
  - Part 1: Implement blog page design.
  - Part 2: Implement a blog mode, which shows the blog subpages instead of the location pages with travel plans.
  - Part 3: Allow videos insertion.
- Share
  - Goal: Allow other users to use the travel plan as a layout for their own travel. The System will create the subpages and the vacation info based on the other project.
  - Part 1: Design Data Interface.
  - Part 2: Implement Export / Import;
  - Part 3: Design and Implement selective data transfer (Which data from the other travel plan do I want to use?).
  - Part 4: Design hidden and public setting (plan can only be copied with an invitation or by everyone).
- Multiple Users (idea based on blog comment by Christian)
  - Goal: Provide different users and change tracking. This feature should make planning with multiple persons easier.
  - Part 1: Design and implement user administration.
  - Part 2: Implement chat and tasks, which can be assigned to users.
  - Part 3: Implement change log.
  - Part 4: Implement calendar synchronization.
  - Part 5: Implement synchronous change (very optional).

These are the subsystems that we want to use, to break down our big application. We know that we have a lot of features and we probably won't implement everything right away. But we want to show all the possible features of our application. What do you think about our segmentation? Please tell us in the comment section. We will make sure, that we will integrate your input into our project.

Our application has only few external dependencies. We want to create a standalone application, so we are only dependent on our hardware and the frameworks / the software we use to build our application.

Our lead developer has also designed a logo for our app. You can see it next to this blog entry. We love it. What do think about it? Tell us in the comments.

We would be pleased if you continue to follow our journey and test our Application.

Our logo:



Your

Perfect Time Crew

---

eventlabproject (2018-10-16 14:52:27)

Hey Guys, thank you for your detailed blog posts about the different roles in your team and the detailed overview of your project and possible extensions. The different steps you will need to take to implement the features gave us also the opportunity to get to know how much effort you are putting into this project. We really enjoyed your logo, as we instantly had the wish to visit the mountains during our next vacation (and of course use your app to plan this wonderful trip ;)). We hope that every team member enjoys his role(s) in your project and we can go on vacation soon. Cheers, Your EventLAB team

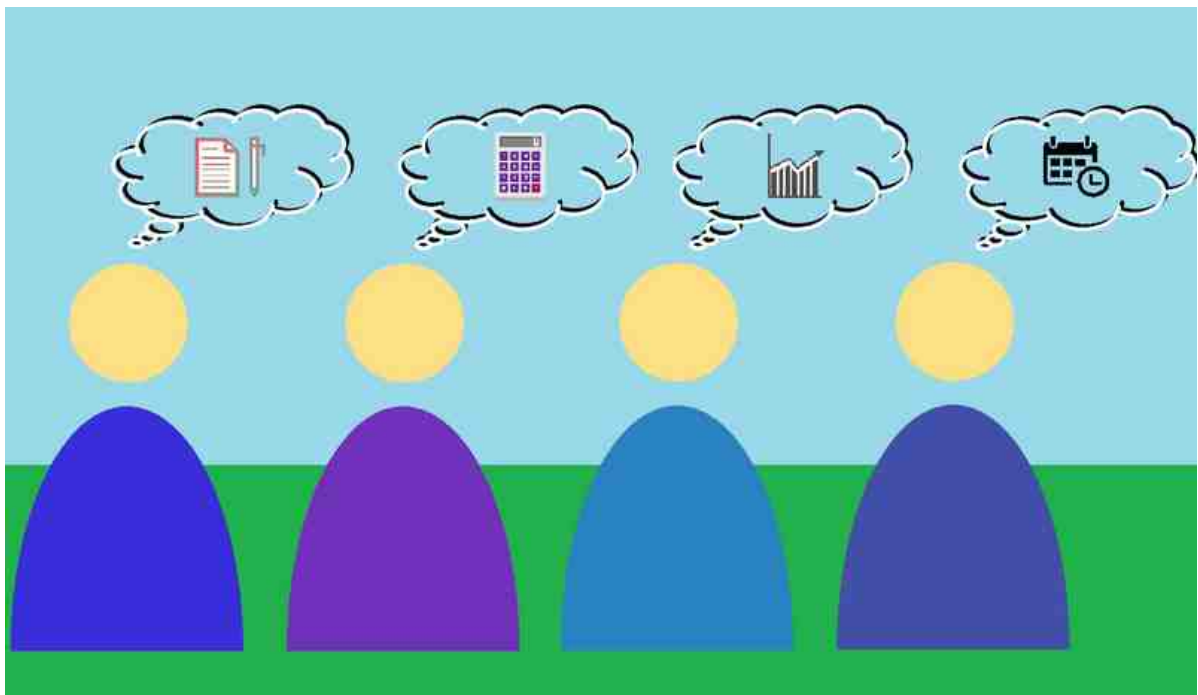
commonplayground (2018-10-17 11:42:47)

Hello there, at first I want to point out that I really like your logo. You also divided your application in smaller parts. Is it right that the users can post anything they want on a blog page of a location? How do you want to make sure that there aren't any misleading images and texts? What if they post an image they took in a different location? To your other post about your roles and technology: Your role allocation table is very well structured and you made clear who has which role and tasks. Also your Youtrack board is well organized but have you thought about tracking your issues by the different RUP phases (with tags)? Do you already know which integration or building tools you want to use? Thanks a lot for all the information. Keep us updated :) Celina and Denis@CommonPlayground

perfecttimese (2018-10-17 11:57:14)

Hi Celina and Denis, thank you for sharing your ideas with us. The blog can only be created by the user(s) that created the trip. So they first created a plan for their trip and after their trip they change the plan into a blog and publish it. So in our opinion they should know where they did take the pictures during their trip. We still have to add the RUP phases to our YouTrack board. We will do that as soon as our YouTrack is running the way we want it to. So it might still take a little time. But thank you for bringing it to our attention. Since we have a React application running on Google Firebase we don't have to use a specific deployment tool ourselves. Firebase will do it for us. Thank you for your support and your thoughts. We will use them to improve our project. So please stay tuned, since there will be more stuff coming soon. :) Your Perfect Time Crew

## Defining Roles (2018-10-13 15:17)



Hey Guys,

last time we gave you an in depth look into our application features. We also explained our dependencies and showed you our logo. In this blog post, we want to specify out team roles in further detail. We therefore (have to) use RUP ([1]<https://www.ibm.com/developerworks/rational/library/apr05/crain/>) . We will also detail the technology we use and show you our project management tool.

The following table shows the RUP disciplines, roles and which team member is filling this role. The Scrum roles we defined in our first blog entry are still valid and guide our team efforts.

## **Business Modeling**

### **Business Process Analyst**

Discovers all business use cases.

**Lea**

## **Requirements**

### **Systems Analyst**

Discovers all requirement use cases.

**Jan**

## **Analysis and Design**

### **Software Architect**

Decides on technologies for the whole solution.

**Lukas**

## **Implementation**

### **Integrator**

Owns the build plan that shows what classes will integrate with one another.

**Tim**

## **Test**

### **Test Manager**

Ensures that testing is complete and conducted for the right motivators.

**Test Analyst**  
Selects what to test based on the motivators.

**Test Designer**  
Decides which tests should be automated vs. manual and creates automations.

**Lea**

**Jan**

**Lukas**

## **Deployment**

### **Deployment Manager**

Oversees deployment for all deployment units.

**Lukas**

## **Project Management**

### **Project Manager**

Creates the business case and a coarse-grained plan; makes go / no go decisions.

**Lea**

## **Environment**

### **Process Engineer**

Owns the process for the project.

**Tim**

We have decided to use YouTrack as our project management tool because it is integrated in our IDE (WebStorm). You can look at our Scrum board and backlog following this link ([2]<https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-5>).

As previously mentioned, we will build a web application in React. Our backend will use Google Firebase to store the data. We use WebStorm as our IDE.

So, what do you think about our project layout? Let us know in the comments. As always, we love to hear your feedback and we will always try to integrate your ideas into our project.

We would be pleased if you continue following our journey and test our Application.

Have great time guys and see you next week.

Your

Perfect Time Crew

1. <https://www.ibm.com/developerworks/rational/library/apr05/crain/>

2. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-5>

---

GummibandBoris (2018-10-18 11:52:58)

Heyho, you've done very well in giving us a detailed insight into the roles of your project. We have noticed you only assigned one person to "Implementation" and wondered whether Tim really is the only one who will be implementing the Application. Using YouTrack because it is already integrated in your IDE is smart. We are looking forward to see your future progress and read more about your project. Best regards Luis@DHBWieWarsEssen

CodeCrunch (2018-10-18 13:13:39)

Hello Team perfecttime, You have a good idea and i can image that it would be very useful. Im traveling a lot and could need a app like this. If you have any questions to a future user, it would be a pleasure for me to help you. Do you have just one implementer? I think this is a huge part and you should split it up. I would like to know more about your technical

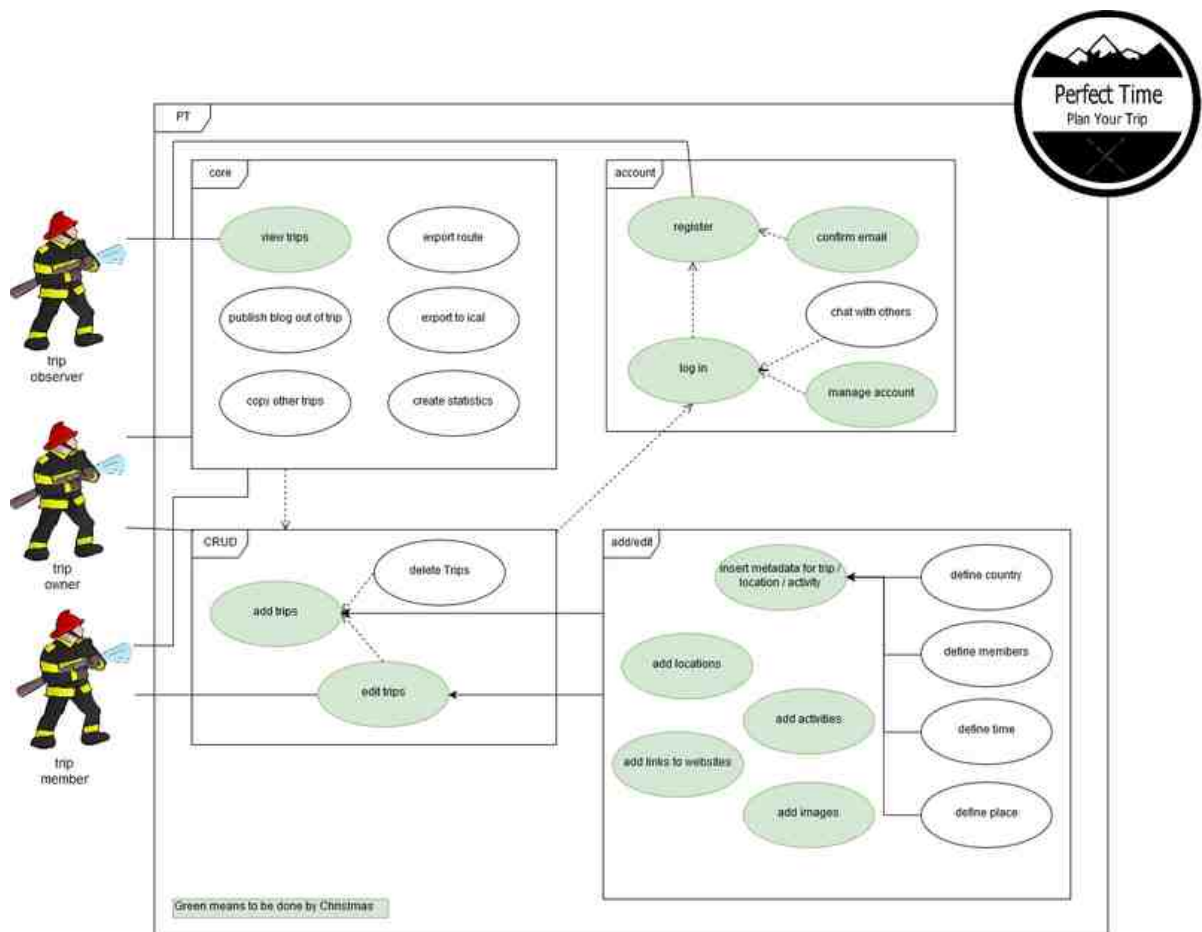
specifications. Do you have already planned where to keep your data? What will you use as your repository. As we are required to deliver and get feedback between courses it would be glad if you evaluate our project too. I'm looking forward to read your next updates! Greetings, Nirjan@CodeCrunch.

### Defining our software requirements (2018-10-21 23:59)



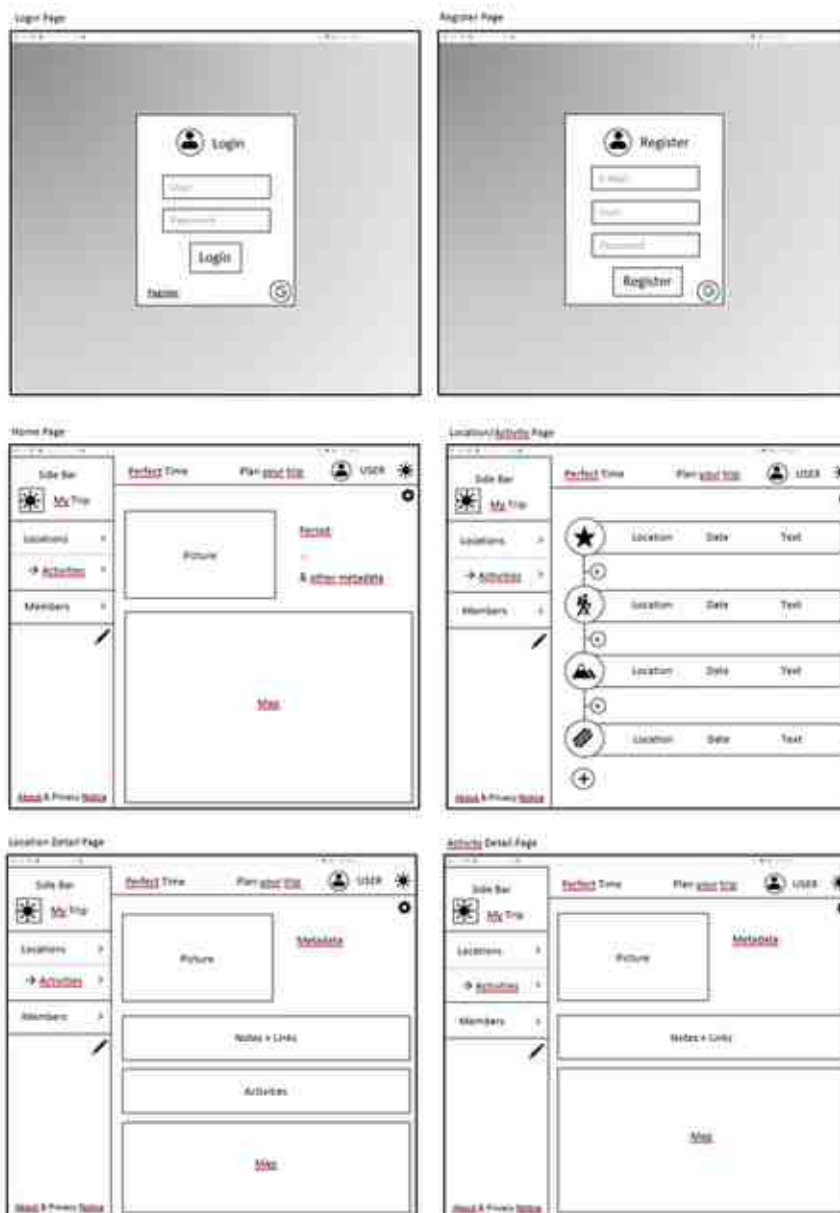
Hey Guys,

last week we defined our software requirement specifications (SRS). They can be found in our Git [1]repository. If you are interested in them, please take the time to read the document provided. We have included a use case diagram in our SRS.



It is obvious that this image was created by our very own firefighter. We like the outcome. What do you think? Please tell us your thoughts.

We also drew a first design of our application layout.



What do you think of our design and our defined requirements? Please leave a comment and as always, stay tuned, since there is more to come soon.

Your

Perfect Time Crew

1. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SRS.md>

CodeCrunch (2018-10-22 14:20:35)

Hello Team perfecttime, Your UC and SRS is very simple and easy to understand, well done. You should use a different format for your SRS because you can not mark text or use the links, maybe markdown? The UC chart seems to be possible for the period we have this semester. Your plan for the second semester is well chosen. But I miss more details about the data for the statistics (budget, km). There are a few spelling and grammatical errors like in the first line of your blogpost. The App-Desgin is as it should be, simple and not overloaded. Are you planning activity suggestion as soon as I add a location? I recently read an article about Google Maps integrating a new feature to planning travels with friends.



Unfortunately i couldnt find the article anymore. Maybe you can take some ideas from there. By reading your SRS I was able to get a good understanding of your project. I'm looking forward to see the first results. Greetings, Nirjan@CodeCrunch.

perfecttimese (2018-10-23 10:37:06)

Hi Nirjan, Thank you for your comment and for sharing your thoughts with us. We will switch to a markup document for future versions of our SRS, even if we don't like the available markup editors. Regarding the statistics function of perfect time, we still have to determine which variables the user can insert (costs, etc.). This is why our description of the statics feature is still broad. We will make it more specific, once we decided on our implementation. The integration of Google maps is one of our biggest tasks ahead. We will put a lot of effort into this aspect. So thank you for your input, we will look for that article. Thanks again for your comment. Your Perfect Time Crew

Team EventLAB (2018-10-22 22:07:54)

Hey Guys, we are very impressed by the big progress you have done since your last update. The first thing we really like is that you've made some really good layouts for your application. It helps a lot to get an impression how you want to implement your ideas. Moreover it is useful for you because now you have a structure and can build your frontend way easier. Just some thoughts we had while appraising it: - the login repectively the register site are simple but that's completely enough - the menu bar and the field for the user are good - the amount of things within one site is alright till maybe a little bit too much. In case you change something about the design (what we think you will do, because you didn't start coding yet), try not to put more on one site, because the more a user has to look at the less he really notices - just an idea: Maybe it is easier for the user if you put the settings icon/link on the menu bar too? All in all we really liked the ideas you came up for your designs. Furthermore we also looked at your SRS and the use case diagram. To start with the SRS you have done a lot of work to sum up all the things you want to do and it is a good start. Of course yo didn't fill in everything, but we hope this will be completed during this project. Like the comment above us we thought that you should have used markdown than an other format. But to continue, the use case diagram we liked a lot. It gives a good overview over your application use cases and is very detailed. In conclusion you have done a good job this week and we hope to get a response from you even though this comment got so long ;) Cheers Your EventLAB team

perfecttimese (2018-10-23 10:46:15)

Hi Guys, thank your for sharing your thoughts with us. We are glad that you like so much about our project. The settings icon allows the user to edit the individual page. The menu bar on the left allows the user to work on the whole trip. Settings in the bar on top will allow the user to edit the application and account settings. In the real application we will emphasize the differences between those settings. Thank you for bringing it to our attention. As mentioned above, we will probably switch to markup for future versions, since Git works badly with PDF documents. Thanks again for your comment and stay tuned. There is a lot more to come. Your Perfect Time Crew

profdh (2018-10-24 16:03:33)

CRUD maker ... this may or may not be useful: <https://www.npmjs.com/package/react-crud-admin>

## Use Cases - Getting closer to our application and to Cloud Nine (2018-10-28 17:42)



Hey Guys,

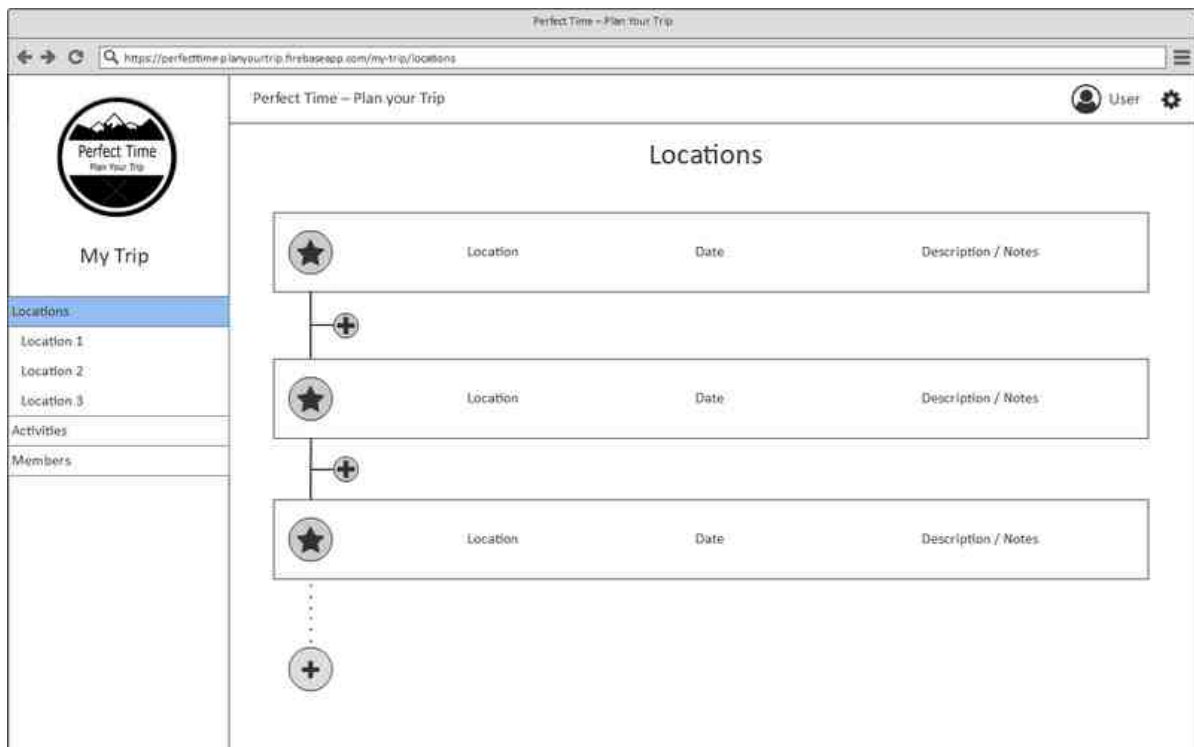
this week we defined our first two use cases:

- edit a location
- add an activity.

You can find our use cases in our [1]GitHub repository.

With these use cases, step by step we are getting closer to coding our application. We are getting more excited with each week and with each milestone we complete on our journey. We hope, you can also see our progress.

With those use cases we improved our layouts. So here is an example.



If you want to see more, please look into the use cases on our GitHub repository.

We also wanted to bring our pictures to your attention. Any picture we publish on our blog, was taken on a trip of a Perfect Time team member. We hope that with the support of our applications, many more people will plan and go on amazing trips. And we hope that many more pictures will be taken and published for the world to see. :-D

Well that is it for this week. We hope you like our use cases and as always we are interested in your thoughts. So please leave a comment and tell us what you think. Thanks guys and stay tuned for next week, since there is a lot more to come.

Your

Perfect Time Crew

1. <https://github.com/Tallround3r/PerfectTime/tree/master/documentation/useCases>

---

EventLAB Team (2018-10-29 17:52:39)

Hey Perfect Time Team, we've had a detailed look at your two use cases - well done! The activity diagrams are easy to understand and provide all necessary information. Your mock-ups look like your application will have an intuitive and simple user interface, that's pretty good! In our eyes, there is nothing that requires any improvements. Nevertheless, we've got some additional questions that came up recently: Do you want to allow sharing trips only for a limited set of users (friends, colleagues, etc.) or also allow making them completely public? Another nice feature would be some kind of tags, so that people can easily find a trip they are looking for: #hiking #Australia #outback. Have a nice week! Your EventLAB team P.S: You took some really beautiful images! We're already saving money to go on vacation when your

application is finished... :)

perfecttimese (2018-10-30 14:38:33)

Hi EventLAB, thank you for your comment. We are glad to hear that you liked our use cases. Regarding your questions, we have to admit, that we haven't thought about tags so far. We thought that grouping our trips by location, duration, time of travel and budget should be sufficient, to find a trip fitting your purpose. However tags are a nice feature and will take them into consideration for an additional feature. Thank you for bringing it to our attention. Our trips can be private (only accessible to travel mates) or public to everyone in the web. For our current setup owners will have to invite friends to become members in order to grant them access to their private trip. For the future we could also imagine having special share options. So we will put your idea into our backlog. Thanks again for sharing your thoughts with us. It is always helpful for our progress. So please continue to do so. Your friends from Perfect Time

learnityyourselfdhw (2018-10-31 10:17:05)

Hello Perfecttime team, I immediately see that you put in a lot of effort this week, your use case specifications are very detailed and the effort definitely paid off, I'm amazed! Just like the Eventlab team already mentioned, I had no problems understanding the process, the flowchart is well done and the mock-ups are great as well. I only have a few minor suggestions/questions: When your Backend validates the inputted data and the data is invalid, shouldn't the user see an error message rather than just showing the perspective again? Also, I think it is better for every Blog-reader if you provide two links that directly navigate to the .md files of your use cases rather than one link that navigates to a folder. But I'm really nitpicking right now :D Overall I think you did great and I expect more in the future! With kind regards, Mert@learnityyourselfdhw

perfecttimese (2018-10-31 10:30:22)

Hi Mert, thank you for your comment. You have raised two valid points. We will show an error message, if the input data was invalid. And yes you are right, from now on we will directly link the .md-files in our blog entry. Thank you for making us aware of those issues. Thank you for your effort and for sharing your thoughts with us. And as always stay tuned, since there is a lot more to come. Your friends from Perfect Time Crew

Testing and .feature (2018-11-04 12:24:12)

[...] Having installed Cucumber, we created our first feature files for our use cases: [...]

## 1.2 November

Testing and .feature (2018-11-03 23:00)



Hi Guys,

this week we started working on our testing-environment. We installed cucumber.js in our workspace. Cucumber.js works nicely with React and our chosen IDE (WebStorm) provides us with code completion and syntax highlighting.

Cucumber will allow us to do framework and GUI testing. For further information you can look up the cucumber [1]documentation and guides.

Having installed Cucumber, we created our first feature files for our [2]use cases:

- [3]edit a location
- [4]add an activity to a location

We inserted the feature files into our use cases, so you will also find them there.

We also created a first running Cucumber test, which checks if our application is running. So we know for sure, that our testing framework is running as it should be.

Well that is all for this blog entry. Please tell us what you think about our project and our progress in the comments. And as always stay tuned, since there is more to come soon.

Your

Perfect Time Crew.

1. <https://docs.cucumber.io/>
2. <https://perfecttime608150251.wordpress.com/2018/10/28/use-cases-getting-closer-to-our-application-and-to-cloud-nine/>
3. <https://github.com/Tallround3r/PerfectTime/blob/master/features/editActivity.feature>
4. <https://github.com/Tallround3r/PerfectTime/blob/master/features/addActivity.feature>

---

EventLAB Team (2018-11-05 17:17:57)

Hey guys, again you did a really good job this week. This time the scope of your tasks weren't that big, so we don't have so much to comment. After looking at your feature file we think the scenarios you chose are by itself comprehensible. The only thing we want to mention is that it would be better if you name the first two scenarios of both use cases more understandable. Just an example that you get what we mean: Scenario: User is not logged in and cannot find the editLocation button Given I have opened a location Then There is no edit button The scenario sounds a little bit like the user has problems finding the button, but what you wanted to say is probably that you have to be logged in to edit a location. Maybe just try to find a name of the scenario that shows more the fact that the task "editLocation" comes with that fact that person has to be logged in. Keep on working :) Cheers Your EventLAB team

PerfectTimeCrew (2018-11-07 00:41:48)

Hi EventLAB, thank you for your comment. We are glad to hear that our feature files are comprehensible. Regarding the feature file you mentioned: We will act on your advise and update this feature file. Thanks again for sharing your thoughts with us. It is always helpful for our progress. So please continue to do so. Your friends from Perfect Time

CodeCrunch (2018-11-06 21:31:40)

Hey there, Your feature files are looking good and easy to read. There are just two things I would change. -> I agree with the essence of what EventLAB team has said about the scenario names. I would recommend putting the information whether I am logged in or not to "Given". So you could keep the name of the Scenario shorter and more understandable. -> When I would use your feature files for testing, i wouldnt know what kind of input data is invalid or valid. Maybe you have to specify the input in your feature files. Other than that, good job! Greetings, Nirjan@CodeCrunch.

PerfectTimeCrew (2018-11-07 00:50:23)

Hi Nirjan, thank you very much for your comment. You mentioned two important things we will consider in our feature files. It is maybe better to put the information whether the user is logged in or not into "Given". We will act on your advise and update the feature files. Thanks again for sharing your thoughts with us. It is always helpful for our progress. So please continue to do so. Your PerfectTimeCrew

Testing in a nutshell (2019-06-02 14:56:53)

[...] We are using feature test to analyze our implemented functionality. Therefor we use Cucumber and TestCafe. You can look at our feature files here. Because of our limited time for this project we can't implement tests for each use case. You will see that by looking at the comments in our feature files. (Original blog post) [...]



Hi Guys,

this week we did a lot of work.

We drew our class diagram. We had a few problems doing that. Since React is based on JavaScript, we do not have classes. But with a new little helper called WAVI we managed to overcome this obstacle. This is what got out of it (Click it to get a full-sized version):

[1]



What do you think? Do you like it? Is there something we should change? Please let us know in the comments.

We also decided on the structure / schema of our database. Since we are using Google Firebase to host our application and our database, we had a choice to either go with a SQL or a NoSQL database. We decided to proceed with a NoSQL database. Our schema will be based on this design:

[2]





2. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/DatabaseLayout.svg>
3. <https://perfecttime-planyourtrip.firebaseio.com/>

---

MNZ Team (2018-11-11 12:18:44)

Hi Perfect Time-Team, we also faced some problems with our Ionic app, because it is based on JS and Angular. You found a good work around with wavi, sadly it doesn't work for us. Your DB schema was well structured and easy to read. Keep up your work, I hope we can read more soon. Your MNZ-Team

perfecttimese (2018-11-13 14:14:35)

Thank you for your feedback. If you have any questions regarding WAVI, please feel free to contact us.

EventLAB Team (2018-11-13 11:15:28)

Hey guys, you did a great job with your class diagrams. Using WAVI you got a result very different to us which is very interesting. Looking at you database and your db schema you said you had to chose between a SQL and a NoSQL database. What was the reason that in the end you decided to go with a NoSQL database? The db schema you created is easy to understand and functional. Keep on going :) Cheers EventLAB team

perfecttimese (2018-11-13 14:22:24)

Hey guys, thank you for your comment. We decided to go with a no SQL Database for several reasons. First of all, FireBase provides a NoSQL database. With some work arounds, you can make it it behave like a SQL DB, but internally it is still a NoSQL DB. Secondly, our documents in the Database have a clear hierarchy. A trip contains several locations and each location may contain several activities. This hierarchy is very intuitive to implement in a NoSQL DB. And last but not least, we have the idea, that our application will allow the user to create custom fields for individual locations / activities (e.g. ticket number for concert at Sydney Opera House). This can be implemented very easily in NoSQL DBs but its almost impossible in SQL DBs to include all possible variations. We hope you you can understand our decision for NoSQL. Thanks again for sharing your thoughts with us. Your friends from Perfect Time

„I don't trust words, I even question actions, but I never doubt patterns.“ (2019-05-12 20:45:05)

[...] pattern. We tried to generate some class diagrams (before and after), but as we told you in past blog posts, we do not use classes. And the tool (WAVI) we used in the past to „fake“ a class [...]



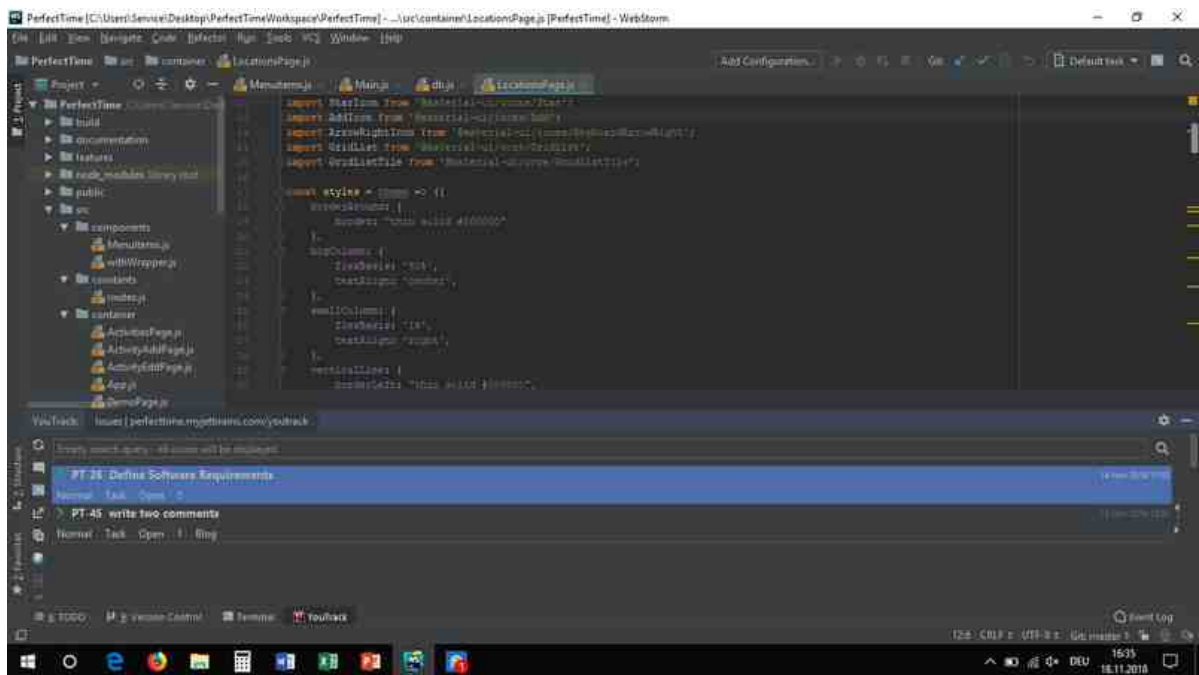
Hey guys,

this week is different from the ones so far. Our class homework is very limited. Next week we will have the opportunity to work with Scrum professionals. Therefore we have to prepare our project. So for the next class we shall prove, that our Scrum-project on YouTrack is running and connected to our other tools.

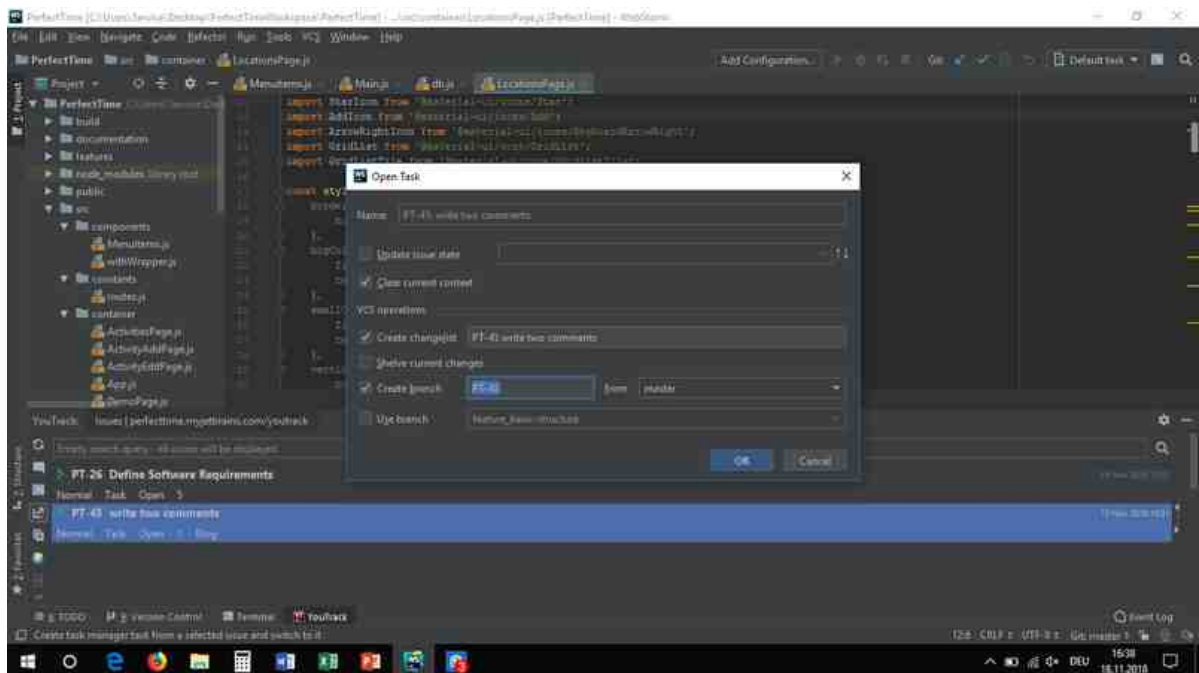
That got us in a lot of trouble, because JetBrains decided to downgrade their free YouTrack-Plan to 3 users only and as you may know, if you have been following this blog, we are 4 team members. So thank you JetBrains for not supporting students.

However we will continue to work on our Project with now only two users for our team on YouTrack (one user is the guest user). With these limited users our YouTrack is running as it should.

We can view, create and edit issues from our IDE (WebStorm):

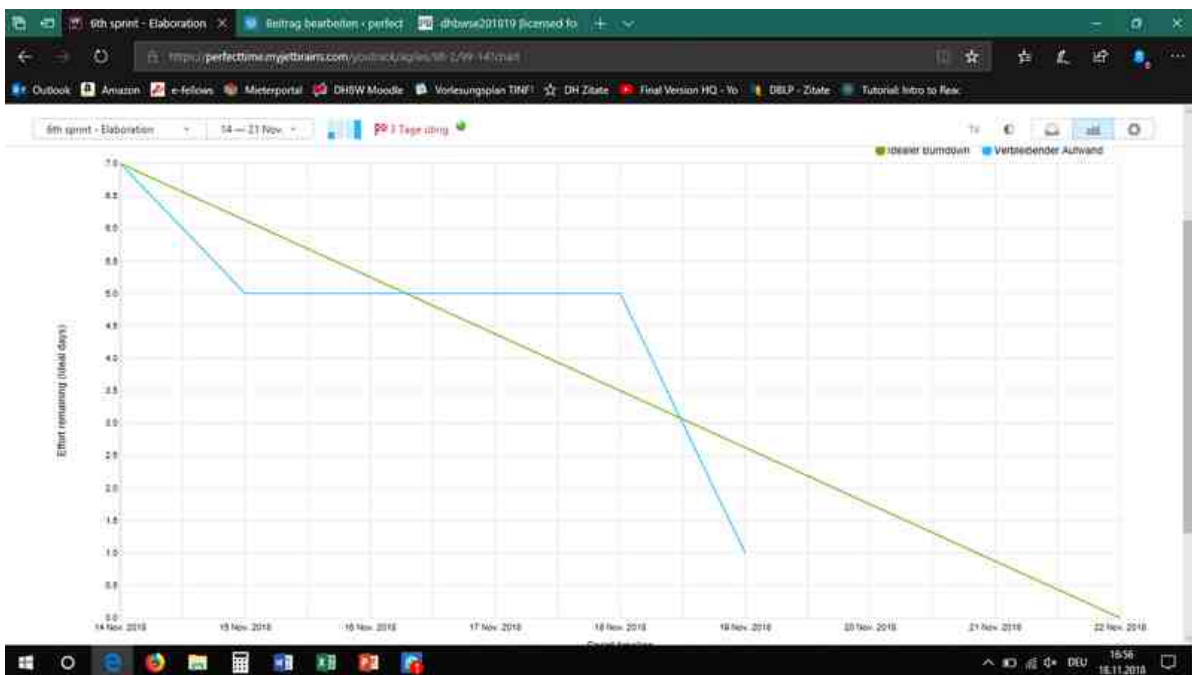
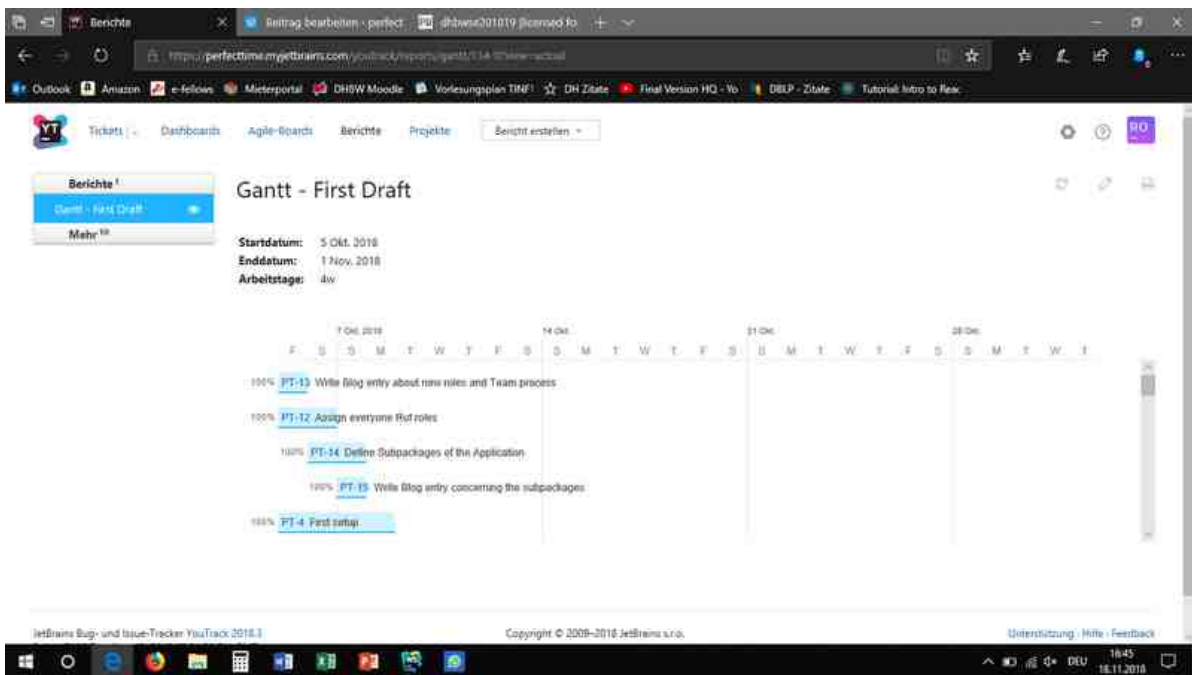


We can create branches in our Git repository from our tasks:



Since we do not use a deployment tool but a npm-command to upload the current version of our app on FireBase, we can't integrate that into our YouTrack project.

However we and you can use our [1]YouTrack project to get an overview of our work. You will also see some interesting reports there, such as a the Gantt-chart and burndown charts.



So please take a look at our [2]YouTrack and tell us what you think about it.

Apart from all the management stuff, we also did some coding and our locations-page looks promising. It is not ready yet to be shown to the world, but we will show it to you soon.

Please tell us what you think in the comments. And as always stay tuned, since there is a lot more to come.

Your

Perfect Time Crew

1. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-14>
2. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-14>

---

EventLAB Team (2018-11-19 13:59:02)

Hey guys, thank you for giving us a detailed overview of your project management tools. Sad to hear that you are facing huge issues with YouTrack, we hope they will offer a solution soon. Nevertheless, your Scrum Board looks well structured and we don't miss any information about tasks. Your burndown charts gives details about your work in the past week(s). Keep up the good work. We are looking forward to your next steps. Your EventLAB-team

perfecttimese (2018-11-21 10:10:35)

Hey Eventlab Team, we also hope to get a solution soon, our professor is in contact to JetBrains and tries to get a study licence as fast as possible. Thanks for following our project and giving us helpful feedback! Stay tuned :D Best regards  
PerfetTime Team

learnityyourselfdhbw (2018-11-20 19:25:46)

Hey :) We also struggled with the downgrade YouTrack made to their free plan. But we found out that YouTrack offers up to 10 persons for free with the "New YouTrack Standalone License" (<https://www.jetbrains.com/youtrack/buy/#edition=standalone>). Maybe this could resolve your problems or you could migrate to another issue tracker. I hope you can find a way to get your whole team back together! Kind regards, jannik@learnityyourself

perfecttimese (2018-11-21 10:09:10)

Thank you for your comment and thank you for bringing up the 10 persons plan. However this is the standalone version of YouTrack, so we would need our own server to deploy it. Unfortunately we do not have our own private server. So this option won't work for us. But we hope it will work for you. Your friends from Perfect Time

Working with the (Scrum) professionals (2018-11-25 01:10:09)

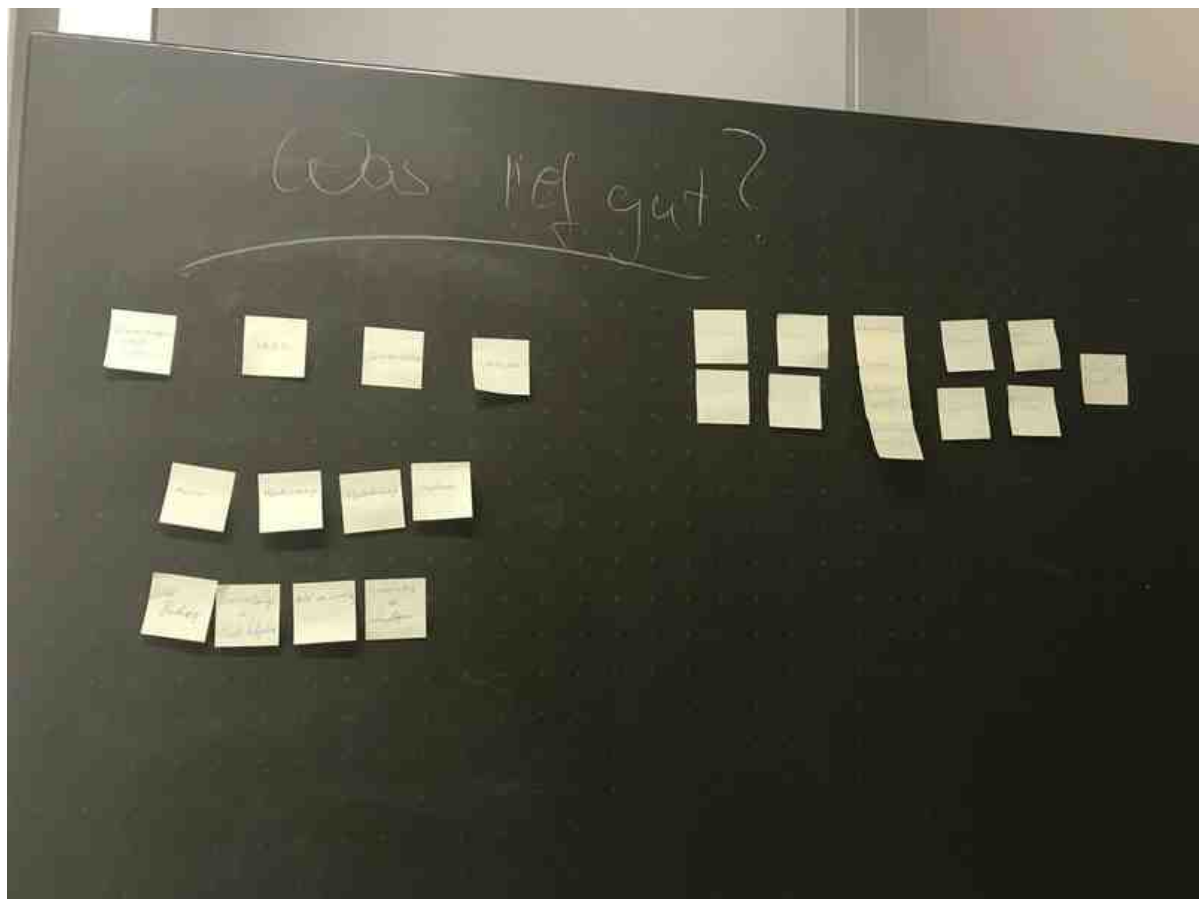
[...] go as we hoped for? First, we are very unhappy with YouTrack and their subscription policy (see out last blog entry). But hopefully our professor managed to strike a deal with JetBrains we should get our full access [...]



Hey guys,

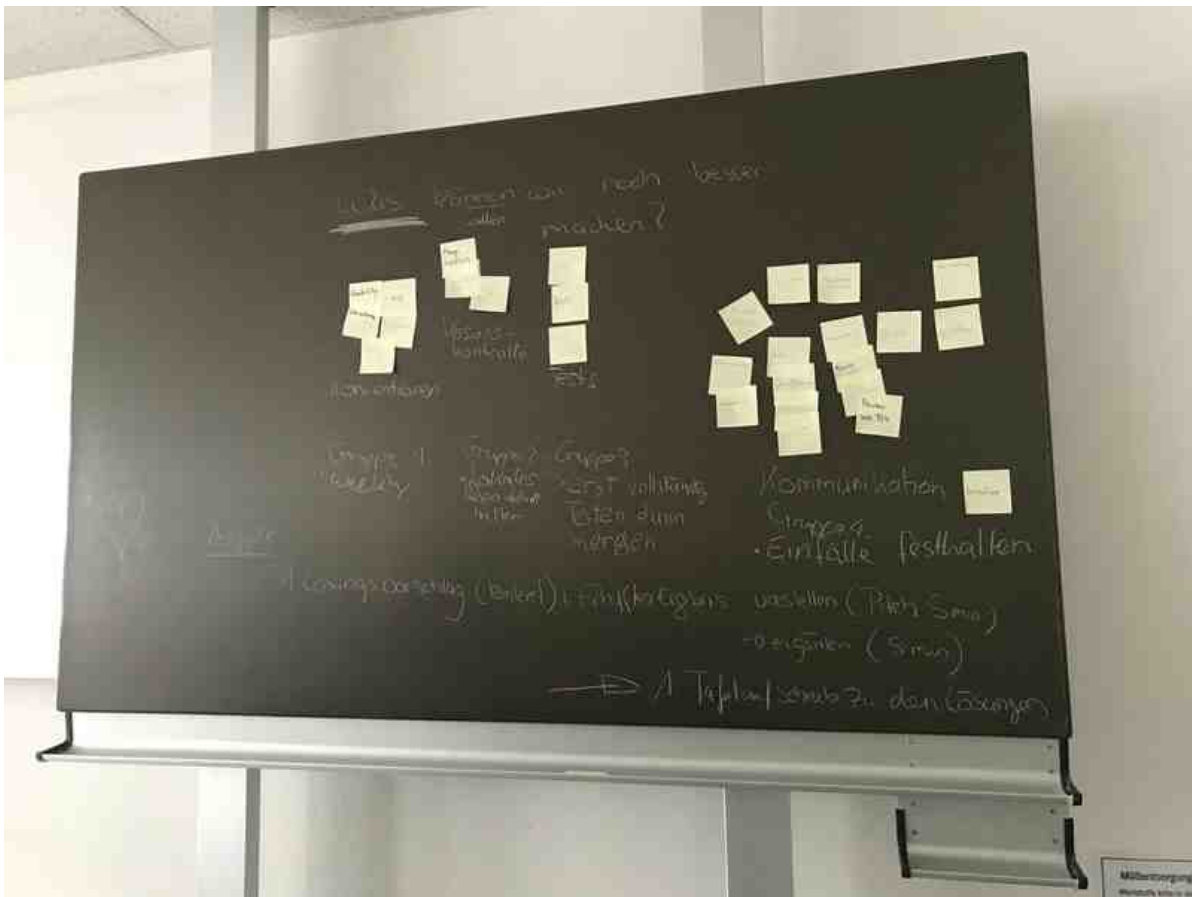
last week was special. We had the opportunity to do a retrospective with a professional Scrum / agile coach. In this event we got some new ideas for our project. In this blog post, we want to tell you about this event.

First, for those of you, who didn't have a [1]retrospective before, let us give you a brief summary of what we did. We reviewed our last sprint(s) and look at what went well, what didn't and how could improve the latter. You usually do that with a lot of post-its.



What did go well in our project. First of all, we really like our team. We support each other and we have each other's back. We like the way we communicate and how we coordinate our work. It is running quite smoothly. We are also quite satisfied with our progress so far. By the way, our [2]current version of the app is online, if you want to check it out. But you need to click the logo and to register to do so.





What didn't go as we hoped for? First, we are very unhappy with YouTrack and their subscription policy (see out [3]last blog entry). But hopefully our professor managed to strike a deal with JetBrains we should get our full access back. Secondly, we thought that our implementation efforts aren't getting enough attention. Our homework and the whole documentation consume most of our time in each sprint. So, we decided to make it a habit to have an implementation task in each sprint, starting next week. And last but not least, we recognized that testing wasn't really on any of our minds. But it should be, because we want to create a quality application. So, we decided to create a dev-branch in our Git repository. Any changes that are merged from dev to master-branch must be tested to their full extend prior to the merge.

These were our lessons learned from our professional retrospective. We liked having the opportunity to work with a true pro and to exchange opinions and strategies with other teams.

What are your thoughts on our retrospective? Tell us in the comments, please. And as always, stay tuned since there is a lot more to come.

Your

Perfect Time Crew.

PS: The past week was Thanksgiving and we also want to show our appreciation for people who supported us. First a personal thank you, to my other team mates, for your help and your efforts. Secondly, to the other teams, who help us with their advice and experience. Special thanks go to [4]EventLAB. And finally thank you dear readers, for showing an interest in our blog and our project. And thank you for your comments and for sharing your thoughts and opinions with us.

THANK YOU. WE REALLY APPRECIATE YOU.



Jan, in the name of the whole Perfect Time Crew

1. [https://de.wikipedia.org/wiki/Scrum#Sprint\\_Retrospective](https://de.wikipedia.org/wiki/Scrum#Sprint_Retrospective)
2. <https://perfecttime-planyourtrip.firebaseio.com/>
3. <https://perfecttime608150251.wordpress.com/2018/11/18/scrum-youtrack-and-coding/>
4. <https://eventlab.jupiterspace.de/>

---

flashcardcommunity (2018-11-27 10:57:34)

Hey Perfect Time Crew, it is always a good experience to review your steps because you can get so much out of it. Especially that you can rely on each other is one of the most important aspects of working in a group. And your progress is quite insane as well. Having a working app running on a server at this point is really good. And it becomes even more amazing when I hear that you had to deal with technical issues at the same time. Concerning your testing, having a dev branch is always a good option. But do you plan on securing that everything was tested before you merge something? Or do you just rely on the teamwork you mentioned and that everyone sticks to the rules? The last thing I would like to mention is that you should try to be more consistent concerning your language. For example the Wikipedia-link redirects me to the german page while your blog post is kept in english. Or in your app, when I try to create an account one of your fields says "Passwort wiederholen" while the rest of your website is again kept in english. Kind regards Benjamin@FlashCardCommunity PS: Thank you for providing the blog every week. It is always nice to read something about your progress. Keep up the good work!

perfecttimese (2018-11-27 18:22:55)

Thanks for your comment and thank you for your praise. We are humbled by it. Regarding your remark about our inconsistent language, thank you for bringing it to our attention. We will be only use English terms in the future. Making sure things are tested before merging with the master should be simple, because we will use merge request, so in the end two team members will have looked at the code and used our automated tests to uphold quality. Thanks again for your comment and for sharing your opinion with us. Your friends from Perfect Time

EventLAB Team (2018-11-27 17:41:56)

Hey guys, this time you wrote a very different blog compared to the updates you give us normally. It is interesting to see your own opinion about your work and your progress. We really liked to hear that the communication inside your team is working and that the developing of your application is running. The things you like to improve seems reasonable for us and you gave good ideas how to fix them. But anyway we got a question about your testing. You said that everything that merges from dev to master branch has to be tested before. How should your testing look like? Do you want to write tests or is it enough if another group member is reviewing the code? Good thoughts this time. Keep on going :) Cheers, Your EventLAB team

perfecttimese (2018-11-27 18:26:40)

Hey guys, thanks for your comment. We want to have automated tests that we can run before finalizing the merge. Apart from that we will use merge request, so every merge to the master branch will be review by at least two people. We hope to thereby adhere high quality standards. Thanks again for sharing your thoughts with us. Your friends from Perfect Time

Looking back and moving forward (2019-05-20 00:54:15)

[...] we looked back at the last retrospective and what changed since then. (We wanted to get YouTrack working again, focus on implementation and [...])

## 1.3 December

Coding like mad (wo)men and describing our architecture (2018-12-02 19:26)



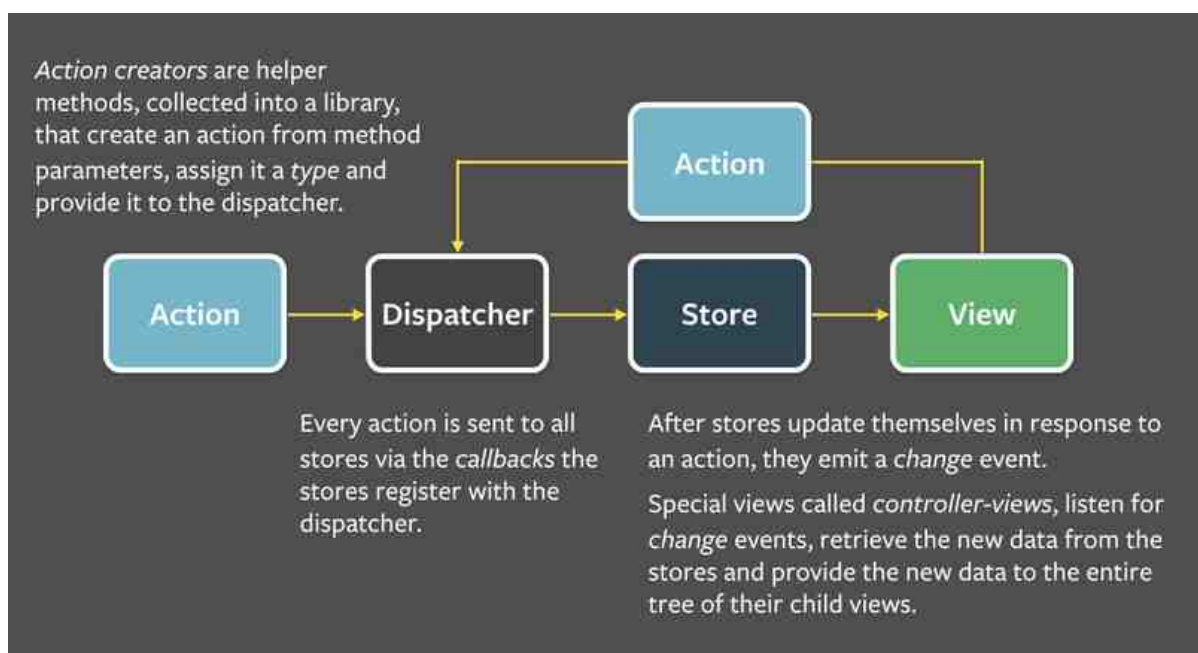
Hey guys,

This week we focused on implementing our features. Since our time is running out and we have to present the current status of our application in two weeks, we have to work like mad people. Between all the other projects from our DHBW schedule and the upcoming exams, we have to prepare for, our (Perfect) time is very limited. ;-)

We implemented our Location View, our Edit Location and our Add Location Page.

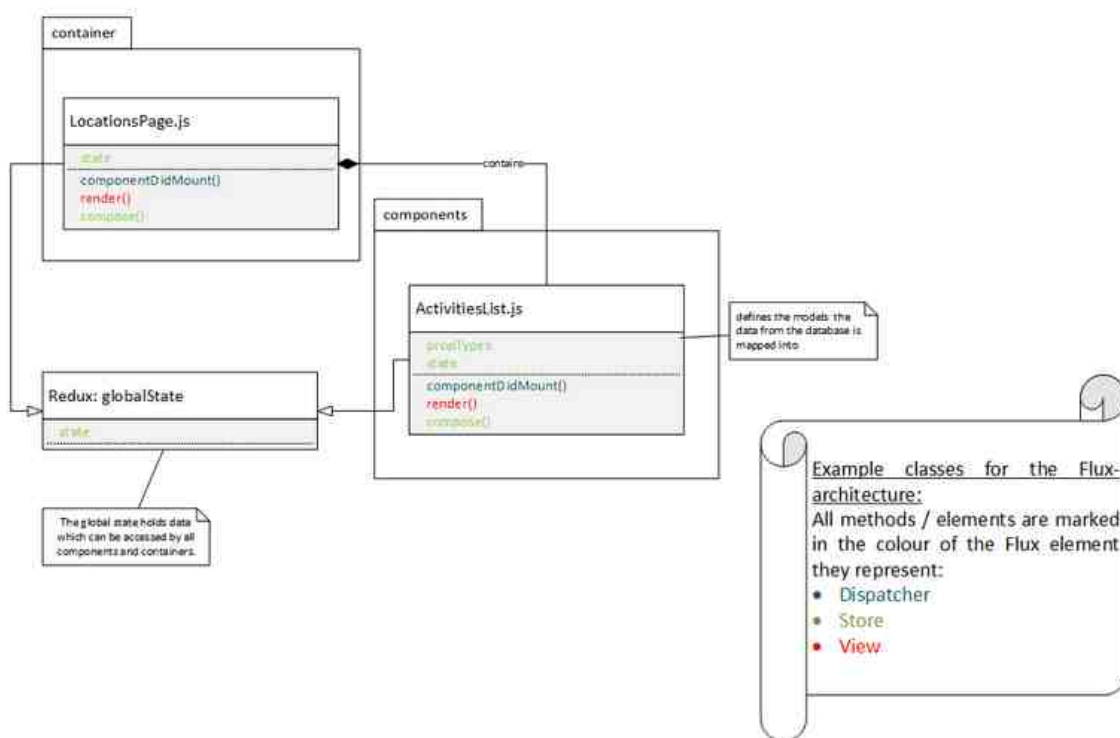
We have worked on our maps integration and implemented a nice slide show for the activities. So from now on, looking at our [1]application will be worth your time. Just click the logo and login/register.

Apart from that we worked on many more use cases and implemented many tests. We also had to describe our architecture. You get the full description here. As you may recall from previous blog posts, we are developing a react application. Thereby it is based on a [2]FLUX architecture. So it is not a pure MVC-pattern.



Flux architecture

Our whole architecture is more complex. The following shows a simple example.



In React the individual Flux elements are present in most of the classes. They can even be mixed up in one method. If you want to know more, please check our Software Architecture [3]document.

Last but not least, we have some good news. Our YouTrack project is working again for all four of us. JetBrains put us back in the old plan. So thank you JetBrains.

What do you think about our progress? Do you like it? What should we change? Please let us know in the comments. And as always, stay tuned since there is a lot more to come (very soon, we promise ;-)).

Your

Perfect Time Crew

1. <https://perfecttime-planyourtrip.firebaseio.com/>
2. <https://facebook.github.io/flux/docs/overview.html#content>
3. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SAD.md>

CommonPlayground (2018-12-02 22:35:36)

Hello Perfect Time Crew! At first, I want to say that it is astonishing how much you implemented in the last few days. Furthermore, it is really interesting to see your web application how it comes alive and to use it by oneself. Also, your SAD is really detailed. Therefore, it would be nice if it could contain a table of contents, so that the reader is able to see everything he wants very quickly. Moreover, I have two suggestions how you could do even better. Is it possible that

the picture of the example for your architecture in your blog gets big when there is a mouse click? Secondly, your class diagrams of the container and components are really large. It looks like you pasted the different pictures next to each other. Wouldn't it be more readable if you put some of them underneath? Other than that, great job. Keep us updated. Best, Celina@CommonPlayground

perfecttimese (2018-12-04 11:17:56)

Hi Celina, thank you for your comment. Creating a table of content in the Mockup file has proven to be difficult with our editor. It is available in the DOCX and PDF version of the document. However we will try to put that in our MD version as well. [Update: We won't do it... The documentation of the editor is only available in Chinese... [1]See here ] Secondly, the "class" diagram cannot be changed or rearranged by us, since it is generated by WAVI. It is very difficult to find a tool like WAVI that generates a class diagram for Java script. So we created it as a SVG, which you can click in our MD file to open. But thanks again for sharing your thoughts with us, Your friends from Perfect Time

1. <http://pad.haroopress.com/page.html?f=markdown-toc>

EventLAB Team (2018-12-04 00:22:27)

Hey guys, you impressively showed that you have done a lot of work and a pretty good job this week! Your Software Architecture Document is comprehensible and provides many details on the structure of your application. It is also great to hear that you are prepared for a rapidly increasing number of users and can easily upgrade to a paid plan. We cannot wait for your final application and are keen on planning our own trips with it! As we do not want you to spend the next weeks bored and without anything to do, we also have got some questions and tips for you! ;) First of all, we tried your current application ourselves and really enjoyed the intuitive user interface design! The problem we encountered is that we could not add or edit any location. In your screenshot, you show an "edit location" screen, but for us this screen appeared to be completely empty. Have you released these features already, or are they still being developed? We also have a question regarding internationalization: Does your framework provide an easy way to translate any text in your application into other languages? Perhaps, you could give us some insights on how you deal with this topic. We mainly came up with this question, as we noticed that your login screen currently has some German and English phrases on it. And last but not least: In your Architecture Document, you wrote that: "Google is not allowed to access the user data which Perfect Time is storing in Firebase. Still the Perfect Time team cannot promise that the NSA or other US agencies won't collect data of Firebase users." Of course, there is not much to do about US agencies accessing your data. But what about attackers, for example? Are there any special steps you take regarding the protection of sensitive data (e.g. not storing plain-text passwords, protection against malicious data, etc.)? Thanks very much in advance for your reply! Best regards, Your EventLAB Team

perfecttimese (2018-12-04 11:31:30)

Hey guys, thank you for your comment and for your advice. We are glad that you like our application. Yes, you are correct, the Edit and Add Location features are still in testing and we are slightly behind our deployment schedule. They are running locally, but we haven't put them on Firebase yet. But we plan to do that during this week. So far we haven't planned for different languages. The time we have available for this project is simply too short. We want to keep the project in English in order to reach the largest target audience. But thank you for bringing the inconsistencies to our attention. We will fix that. Regarding the app security, we don't save the passwords in our own database. Firebase provides user administration which saves the users and their passwords in a protected mode with encoding. Regarding malware, we will only allow certain file types (e.g. JPEG) to be uploaded and stored. Apart from that we will integrate a malware detection tool. We are simply lacking the time to engineer that. We hope that you will still use our application and that you will not attempt to upload malware. Thank you again for sharing your thoughts. Your friends from Perfect Time



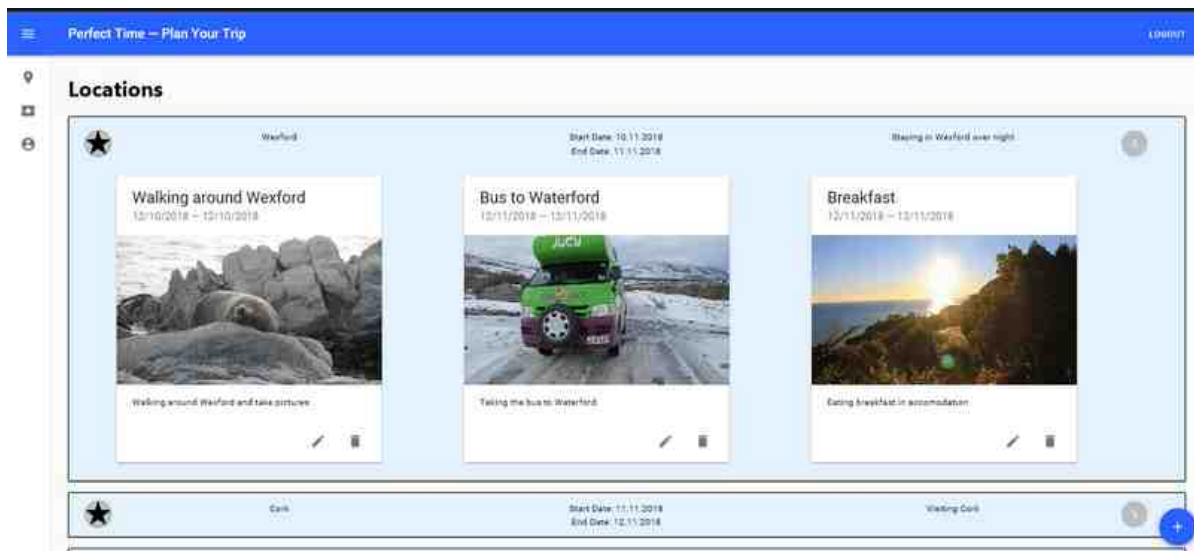
Hey guys,

this is the last real blogpost for this year. The next one will only have a list of all the documentation we have done so far. What happened this week. Well it was a lot.

First, we are still coding like there is no tomorrow and we are getting better and faster with every day. We implemented add, view and edit activity. Which in our application is unfortunately no simple CRUD operation which you can autogenerate with a few commands. But still, we got it running. We have also integrated the remaining models into our dispatchers. We activated the confirm email-address function in Firebase. We switched to a new (better) map provider and got our location on the pages (we are still working on making it look amazing). And finally we got our tests running with the current implementation. We are pretty happy with all the work we got done.

But still, we have some more points to work on until Christmas. This focuses mainly on the ability to edit the whole trip. We are confident, that will complete this soon. So feel free to take a look at our [1]current version.

[2]



We also prepared for our Midterm presentation... If you want a sneak peak, go find it in our [3]Git Repository. If not, you will find it in the blog post before Christmas.

We took a brief moment to look at the hours spent on this project this year. We will reach 276 hours by Christmas. So we hope all this effort will be worth it in the end.

Apart from that, all of us have a ton of exams coming up. We are uncertain, if it is still okay to sleep or if we just should study and work on our projects 24/7. If there is another blog post published before Christmas on this website, you will know that we have somehow survived this examination madness.

If you have any thoughts on our progress, please let us know in the comments. And as always, stay tuned since there is a lot more to come, if we survive the exams. ;-)

Your

Perfect Time Crew

1. <https://perfecttime-planyourtrip.firebaseio.com/>
2. <https://perfecttime-planyourtrip.firebaseio.com/>
3. <https://github.com/Tallround3r/PerfectTime/tree/midterms/documentation/Midterms>

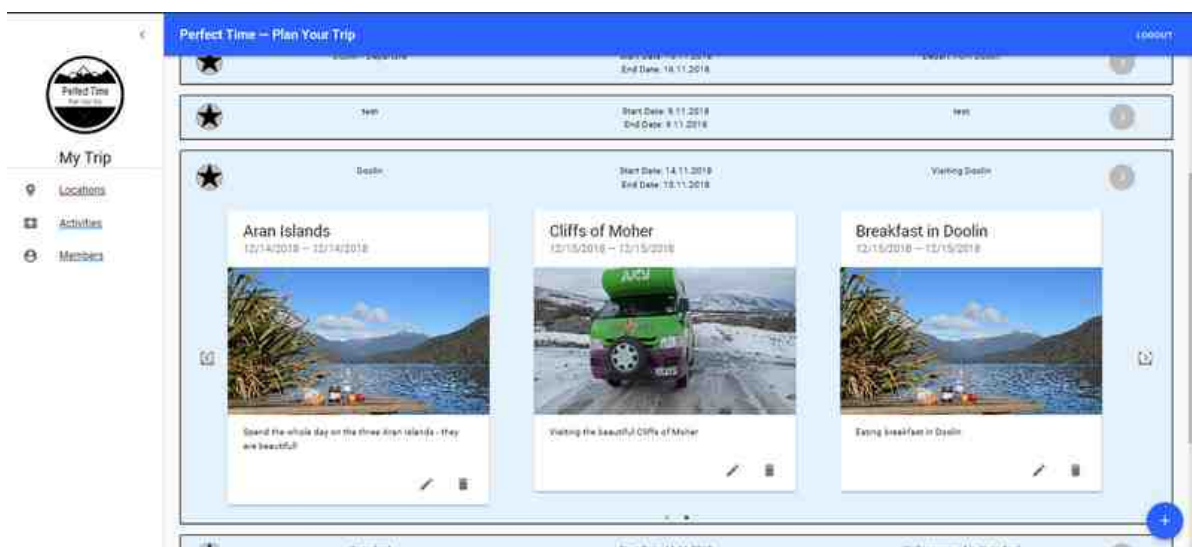


## Checkpoint reached - handing in the midterm documents (2018-12-23 18:40)



Hey guys,

this is our last blog post for this year. As you might noticed, we have somehow managed to survive our exams. And we completed implementing our target use cases for this year. You can find our current version of the Perfect Time Application [1]here. Just click the logo and sign in or register.



Apart from that, you can find all our other documents and "achievements" by using the following links.



## **Hand-in materials**

- Use Cases

[2]Overview and [3]folder

- SRS

[4]Software Requirement Specifications file

- Test Cases

[5]Test Cases folder (incl. .feature and step definitions)

- Test Log

(screen shot in red/green)

[6]example test log

- Functional Test

[7]link to example test video

### *Project Management*

- RUP Gantt chart (past)

[8]YouTrack Gantt Chart (not realistic, since MS Project was not covered / required in class yet, so for now, we are stuck with YouTrack)

- burndown

[9]YouTrack burndown whole year,

[10]Ideal sprint burndown

### *Ability to run*

- Demo

[11]Perfect Time Application

- Code

[12]Code repository

### *Architecture*

- SAD document

[13]Software Architecture Document

does not include pattern highlighted as part of architecture (Flux architecture, as discussed)

### *Configuration Management/ Environmental Setup*

- Automated Testing

No Jenkins or similar tool use for deployment. Firebase uses its own commands (without automated testing).

Cucumber and TestCafe are used to test before deployment. See[14] feature files.

### *Other*

- Presentations

[15]Midterm presentation & [16]handout

We hope that you like our progress and that you will continue to follow our progress. The whole team is disbanding for the next three month. We are all heading back to our companies to work on our actual jobs. However, we will start to develop and post again in April. So tune in again in April since there is a lot more to come.

We wish you all a pleasant Christmas holiday and a good start of 2019.

See you next year guys.

Your

Perfect Time Crew

1. <https://perfecttime-planyourtrip.firebaseio.com/>
  2. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/UseCaseDiagram\\_redesigned.png](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/UseCaseDiagram_redesigned.png)
  3. <https://github.com/Tallround3r/PerfectTime/tree/master/documentation/useCases>
  4. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SRS.md>
  5. <https://github.com/Tallround3r/PerfectTime/tree/master/features>
  6. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/testCoverage.jpg>
  7. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/2018\\_12\\_11\\_Cucumber\\_Running.mp4](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/2018_12_11_Cucumber_Running.mp4)
  8. <https://perfecttime.myjetbrains.com/youtrack/reports/gantt/134-1?view=actual>
  9. <https://perfecttime.myjetbrains.com/youtrack/reports/burndown/119-2>
  10. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-14?chart>
  11. <https://perfecttime-planyourtrip.firebaseio.com/>
  12. <https://github.com/Tallround3r/PerfectTime/tree/master/src>
  13. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SAD.md>
  14. <https://github.com/Tallround3r/PerfectTime/tree/master/features>
  15. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/MidTerms.pptx>
  16. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/MidTerm\\_Report\\_PerfectTime.docx](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/MidTerm_Report_PerfectTime.docx)
-

## 2. 2019

### 2.1 April

Back on (You)Track (2019-04-07 12:00)



Hey guys,

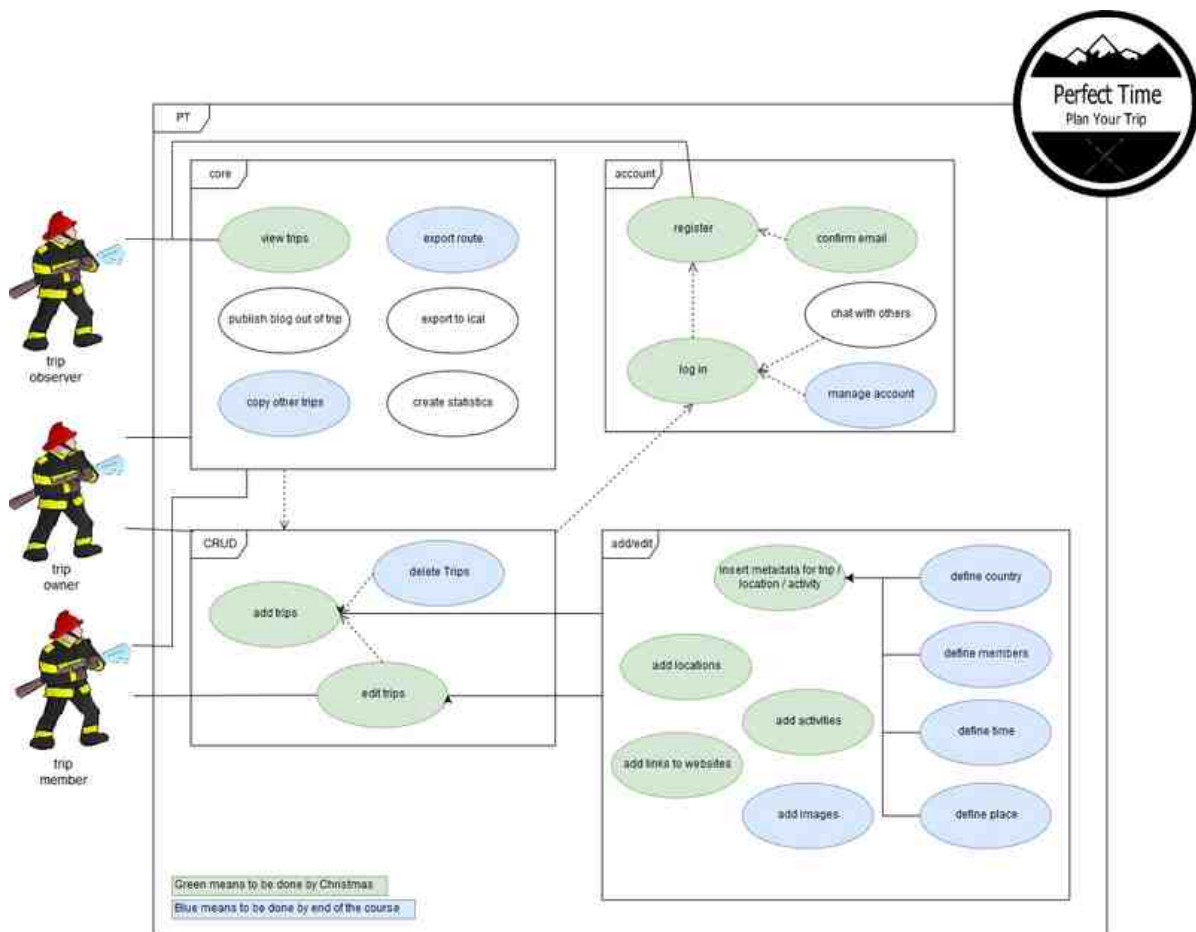
we are back - at the DHBW and working on our Perfect Time project.

Due to our full workload at our companies, we didn't have the time to work on Perfect Time. Nevertheless we are ready to tackle the project again.

Since there is no time to waste, we got right back at it. We defined our scope for this term. We decided to focus on the following issues:

- Implementing a user management
- Implementing the export function and the copy trip feature
- Finalizing our travel plan (including meta data for the whole trip and uploading pictures)

You can see our tasks for this semester in the following diagram (marked in blue):



We believe it is best for the project to focus on the quality of the core functionality. Even if that implies, that we don't have the time to implement all functions by the end of this semester.

Apart from that, we decided to use [1]Travis CI for our deployment process. We are also working on fixing the issues, recognized by our professor in our documentation (we got some broken links). This should be fixed by next week.

The other parts of our project stay unchanged. It is still us four folks working on Perfect Time. We are still happy with the functionality Firebase provides us with. We also enjoy coding in the React Framework.

So as always stay tuned, since there is a lot more to come.

Your,

Perfect Time Crew

1. <https://travis-ci.org/Tallround3r/PerfectTime>

## A risky business? (2019-04-14 19:55)



Hey guys,

this week we had to work on the dangerous stuff - namely identifying the risks of our project and how we could mitigate them. We identified six major risks. (We didn't want to break down the risks into the smallest possible units, because the units would have considerable dependencies on one another. Therefore we would still have to handle them together.) You can find our risks in our [1]GitHub repository. (You will need to download the Excel-File.)

What do you think? Did we identify all the important risks or did we miss or misjudge some? Please let us know.

We also worked on our new use cases. You can look them up [2]here.

Regarding our use cases we also looked at the time we spent on the different use cases. You will see, that we put most of our effort in the Add and Edit Location Use Cases. We required the time and effort to work with our maps API and to find our design. If you want to take a closer look, you can check out our [3]UC-time-spent-table or our [4]YouTrack report.

Well that's all for this weeks effort. Next week we will evaluate the function points of our application. So stay tuned, since there is a lot more to come...

Your

Perfect Time Crew

BTW: You also find the [5]risk and the [6]time table on our Google Drive.

1. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/RiskManagement.xlsx>
2. <https://github.com/Tallround3r/PerfectTime/tree/master/documentation/useCases>
3. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/TimeEstimationUC.xlsx>
4. <https://perfecttime.myjetbrains.com/youtrack/reports/time/145-1>
5. <https://drive.google.com/open?id=1DhfD6We8WK3xnM-8WzH4JZiLLyPSlczX>
6. [https://drive.google.com/open?id=1ZQ\\_V2uwXbj0vQeNRhskU6kZrGtW4QiVQ](https://drive.google.com/open?id=1ZQ_V2uwXbj0vQeNRhskU6kZrGtW4QiVQ)

---

DHBWieWarsEssen (2019-04-15 09:44:38)

Hello and a wonderful day, I was really impressed by your excel sheet, it looks marvelous. Just a few ideas I wanted to share, maybe give your second and sixth risk a higher impact because it's maybe a little bit more crucial then you might have thought. Stay motivated and keep your good work going. Best regards Thimo - DHBWieWarsEssen

perfecttime (2019-04-15 11:25:38)

Hi Thimo, thanks for your remarks. We gave a lot of thought into the risk evaluation of our project. We don't think that loosing one team member would impact our project in a success-threatening way. We are four people. Other teams have to accomplish their goals with only three contributors. And our scope is defined that we could publish the product without some of the defined functions. The functions could be added later as a major update. Therefor we do not see the risk of not finishing all our functions within this semester as threatening to crucial parts of the projects. But thank you for your ideas and we will continue to pay close attention to the points you mentioned. Your friends from Perfect Time

learnityourselfdhw (2019-04-15 13:20:28)

Hey, Your risks seem to be well-thought-out. And the time sheet gives a nice overview of your efforts. The only thing I want to mention is: You could use a .csv file in GitHub instead of using an Excel-file. So nobody will need to download the Excel-file. Best regards, Angelika@LearnItYourself

perfecttime (2019-04-15 13:56:16)

Hi Angelika, thanks for the tip. We tried it. However we are using Line Breaks in cells, to achieve a nice formatting. This doesn't work with csv-Files. So we can either make it available without a download or easy to read... We will stick with the xlsx-format for now. But we can make it an md-table once we are done with the project, so it will look nice. But thank you for sharing your ideas with us. We appreciate it. Your friend from Perfect Time

7lofficial (2019-04-15 13:36:36)

Hi all, your risk depictions are very detailed, so its very easy to understand what you mean. I think your risk factor for the "buggy product" is relatively low, because the impression arises that the risk of not getting finished is higher than a buggy product. In my opinion these two risks belong to the more or less same category which should be refelcted within the risk factor. I also had a look on your time spent table. It looks good. You are a very efficient team. I'm looking forward to see your project growing. Best Regards Sven @dashup

perfecttime (2019-04-15 14:04:13)

Hi Sven, thank you for your comment. We get your point. Little time and low quality output are definitely related. However we have seen in the last semester, that if we are faced with the trade-off between promoting quality and delivering on time, we as a team are more likely to focus on quality. Therefor we believe the risk of not getting our tasks done this semester is greater than the risk of having many bugs. The impact of both risks should be similar, just like you said, and they are both given the same grade in the table. Thank you for sharing your thoughts with us. We appreciate your input. Your friends from Perfect Time

EventLAB Team (2019-04-16 08:49:25)

Hey guys, you've done a good job specifying the possible risks that might effect your project. As far as we are concerned, there is nothing to add to this. Your measures for risk mitigation also seem very plausible. Regarding your report of the time spent on each use case, we were wondering why the "spent time" field almost exactly matches your time estimation



and always is a full hour value. Perhaps you could track the actual implementation time within you IDE, so that you've got the exact time in minutes. Furthermore, this would make it easier for you, as you do not need to track the time manually anymore. Additionally, you might (perhaps) think of using another format for creating your tables. Binary files (Excel) are not really suitable for GitHub. We can recommend Markdown files, as they are very simple and natively supported. Additionally, this allows you to see the actual changes in each commit (any not only "Binary file changed"). Best regards, Your EventLAB Team

perfecttimese (2019-04-16 11:45:45)

Hey guys, thanks for your comment. Regarding our time measurement, we do not use our IDE to track it, because many tasks (documentation, ...) are not IDE related. Therefor we only use YouTrack to enter time spent. And for simplicity reasons we only use full hours to track time. We put the tables on Google Drive as well, so you will see it there too. Thanks again for sharing your thoughts with us. Your friends from Perfect Time

The grand finale – We made it! (2019-06-16 17:01:23)

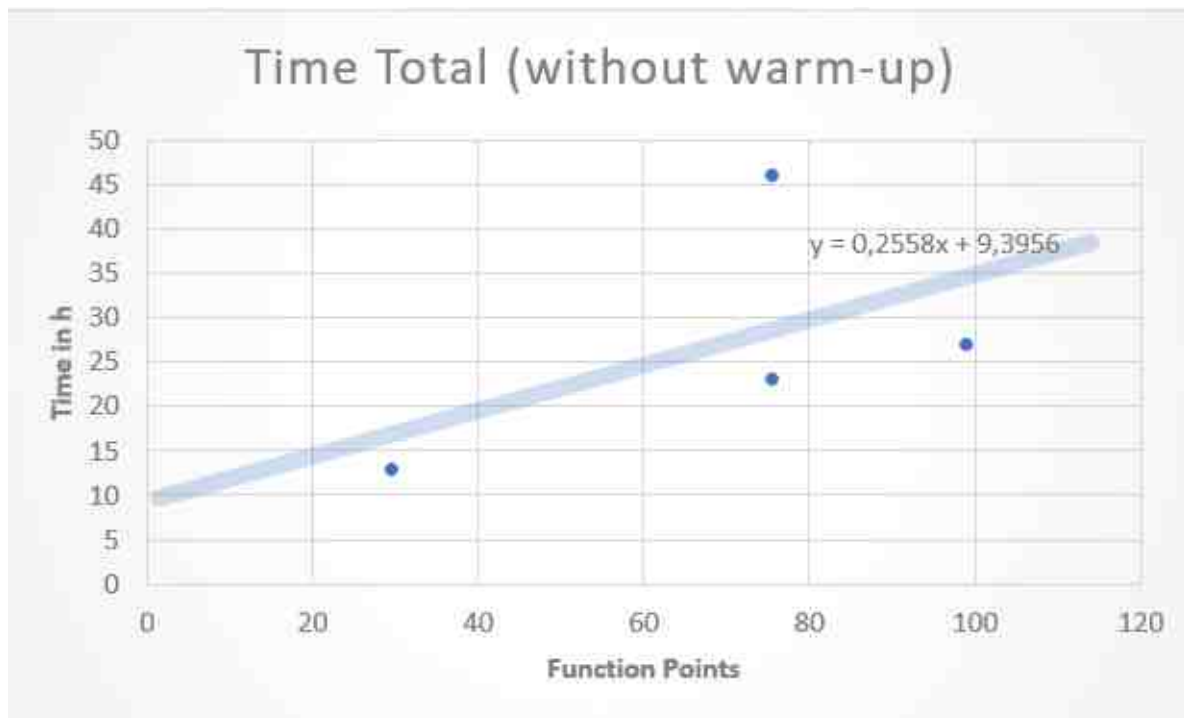
[...] Risk calculation (explained here) [...]

**Yes, we can! (finish our defined scope within this semester) (2019-04-21 12:00)**



Hey guys,

this week we finished assigning function points to our [1]use cases. The document we created to calculate our complexities (DET, RET, FTR) can be found [2]here (you need to download it). We looked at how many hours it took us, to achieve the functions. From that we calculated how much time we will probably need to implement the future functions. The graph shows our h/FP in the past UCs. The points at 46h and at 99 FP are separate use cases but with similar UI-elements. Therefor some work of one UC could be copy-pasted into the other UC. In total the graph should be quite accurate for the work we have done so far.



This [3] calculation for the future scope (110,72 FP in total) resulted in 55h (13,75h per person). This is a very low number. However the Use Cases from last semester are not finished yet (apart from register). They will still need some work, since they require an existing user management to have all their functions available (e.g. only trip members can edit a trip and only the owner can delete it). We estimate that we will have to put another 50 h in the old UCs. The calculation should then add another 20h to the future UCs. Thereby we have a workload of 120h this semester for UCs (30h per person).

Since we have 8 weeks remaining this will be doable. In average each team mate will put 10-20h of work into the project each week. This is consistent with many weeks in the past.

Apart from the planning effort we put a lot of time into implementing our user management. It is still untested, but we are confident that we can deploy our progress soon.

Testing was an other part of our work load this week. We implemented some new tests for existing features. We want to make sure that our application has a high test coverage to avoid a buggy performance.

So that's all for this week. Thank you for reading our blog and stay tuned, since there is a lot more to come.

Your

Perfect Time Crew.

1. <https://github.com/Tallround3r/PerfectTime/tree/master/documentation/useCases>

2. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/PT\\_function\\_points\\_compl](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/PT_function_points_compl)

exity\_calc.ods

3. [https://drive.google.com/open?id=1ZQ\\_V2uwXbj0vQeNRhskU6kZrGtW4QiVQ](https://drive.google.com/open?id=1ZQ_V2uwXbj0vQeNRhskU6kZrGtW4QiVQ)

---

learnityyourselfdhw (2019-04-22 13:24:37)

Over all your Blog entry looks good. The FP/Time Plot gives a good oversight on how FP correlates with time spent in your project. Additionally the stray data point was explained well within the blogpost. The only thing I'm missing is the Complexity Adjustment Table in case you used the Tiny Tools FP Calculator or a similar tool. With regards Jonas@LearnItYourself

perfecttimese (2019-04-22 21:06:45)

Hey Jonas, thanks for your positive feedback! Eventough it is Easter we tried to do as much as we were able to and this shows us that it worked. :) We havent added the Complexity Adjustment Table in our post because we have two different ones for Use Cases with file output and without file output. To find the best way to show it to you, without losing the overview, we will talk about this when we meet tomorrow and update our post. Stay tuned and have a nice evening. Your PerfectTime Crew

EventLAB Team (2019-04-22 21:20:30)

Hey guys, this week, you've done a great job again! Your diagram gives an intuitive overview of the velocity (h/FP) you reached in your previous use cases. The outliers are explained very well, so the interpolation looks quite accurate. Perhaps, you could also share some insights into the DET/RET/FTR calculation. Apart from that, all required information is given in the individual use case specifications and all grading criteria are fulfilled (from our point of view :) ) And of course, it's nice to hear, that you will be able to complete your project this semester! Keep up the good work! Your colleagues from EventLAB

perfecttimese (2019-04-23 11:48:12)

Hey EventLAB Team, thanks again for your positive feedback. We have added a link to the document where you can see our complexity calculations. If you want to look at it. Have a nice day and stay tuned! PerfectTime Crew

CodeCrunch (2019-04-23 22:33:52)

Greetings PERFECTTIMESE, I think it is fair to say that you have provided some valuable and interesting insight on your project with this blog post. Having a list of every element belonging to the "DET, FTR and RET" categories for all the UCs used for the complexity calculation specifically, rather than just providing totals of elements in every complexity category without explanation makes the presented data much easier to interpret than the other UC complexity calculation I have seen. The explanation given to why the data points are located where they are on the coordinate system which holds the h/fp plot also helps with understanding the presented data. But the conclusion provided with the calculation result is the most crucial and therefore best part of the blog post in my opinion, as it provides the reader with a really good understanding of where you are in the project in the grand scale of things and how the h/fp plot helped you determine your future work load. Having said that I also have some questions regarding the blog posts content. Firstly I am curious why you decided to use the y axis to display the time in your h/fp plot. Is there a specific reason? Of course this is simply a formality but as a general rule the independent variable, which in this case is the time, should be represented on the x axis rather than the y axis. To be clear, the way you plotted the graph is not false, it just makes it harder to compare your graph with the graph of one of your se course members naturally, since most of them followed the "independent variable x axis convention" and therefore have switched axis descriptions. Secondly you mention a term called "warm up" in your blog post but as far as I can tell, this term has not been accurately defined anywhere in the blog post. Of course I have an idea what is supposed to be meant by it. (probably that you needed some time to read up on some concepts which were foreign to you and therefore could'nt start with the implementation of a defined UC right a way) I would have still liked a short sentence that would have explained the "warm up" concept. Staying with the "warm up" concept for a moment I am also wondering why you even went out of your way to remove the time allocated to warming up from the total time you spend working on a UC. Be free to correct me if you disagree but is not the part where you "warm up" to write the code for the implementation of a UC the or at least one essential ingredient of the UC implementation process? In my past experience as a programmer at least, I found that the stage of a program implementation where you are educating yourself on some new ideas is far more challenging than when you are using code completion in the IDE of your choice to

write code. Do yourself a favour and include the "warm-up" in your work duration estimation. And Lastly I am wondering, did you provide an explanation to why the resulting complexities in your "PT\_function\_points\_complexity\_calc.od" excel file are the way they are (Low,Medium, High) somewhere in your blog post because I seem to miss it. In case you did not provide the explanation. I would appreciate if you could just very roughly say a couple of words about the "resulting" complexities of your UCs regarding EI, EO,EQ,ILF and EIF. Since most of the resulting complexities are valued "low" with the exception of the EI of the UC "CRUD Trips", which has a resulting complexity valued "high", I am really not expecting a UC by UC complexity explanation. Despite these few points of confusion though,I am quite impressed with what I have seen here, as I have mentioned already in my opening remarks. I hope that you can finally finish up on the use cases of last semester, so you can focus your attention to the challenges which lie ahead. Best Regards, Falko@CodeCrunch

The grand finale – We made it! (2019-06-16 17:00:55)  
[...] FP planning and graph (see blog post) [...]

**Working on the next Unit... and Testing it. (2019-04-28 21:43)**



Hey guys,

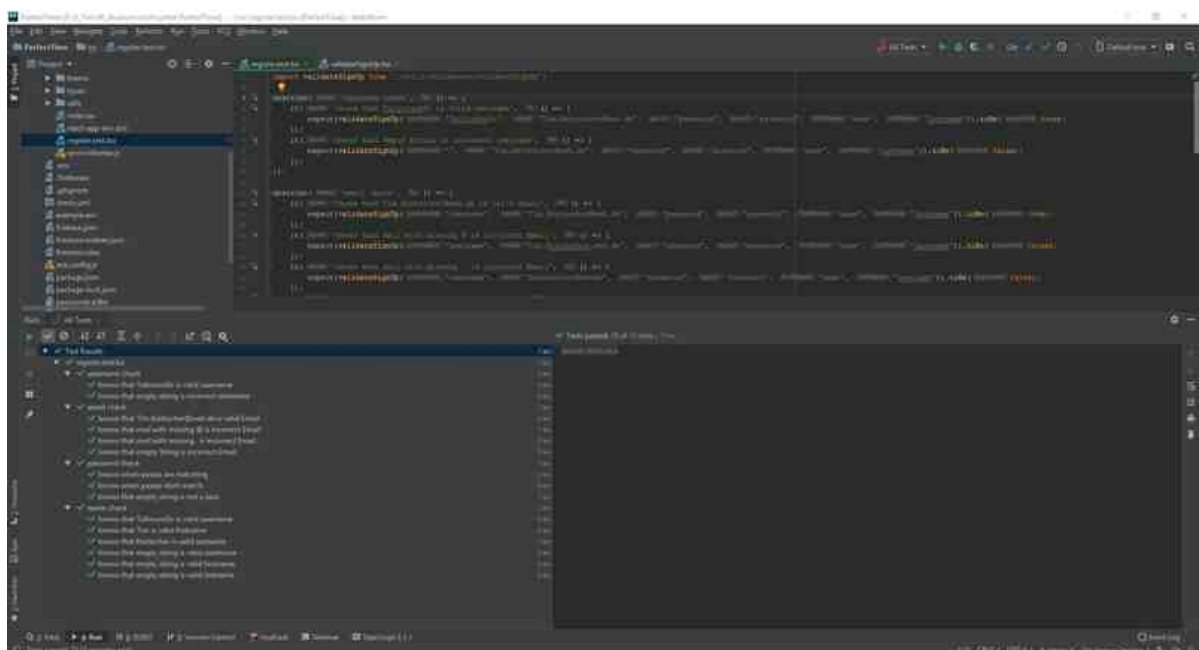
so this week we worked on two big issues.

First of all, we implemented our user management. So all our registered users can edit their accounts, delete them and "follow" other users. So we made some significant progress there.

The second big topic was testing. We are already using Cucumber and TestCafe to test our application. However we didn't have Unit Tests in our application so far. Well, now we do. We integrated [1]Jest in our project, so we can test our React application using WebStorm. Jest allows us to mock data and dependencies while testing our application in WebStorm. That is working quite smoothly. There are many tutorials and examples of how to do that [2]online. And even Test Driven Development (TDD) is possible within our setup (see [3]JetBrains' tutorial for PyCharm; can be adapted for WebStorm).

We don't use Maven or Gradle to build our application (it's a JS and not a Java application). Our build works with Node.js and Yarn. During the build process we are also running our tests. This process is also included into our automated deployment with [4]Travis CI.

We implemented our first unit tests for our [5]register use case . The tests are up and running. The results are successes. You can find our test code on [6]GitHub.



```
npm
PASS src/register.test.tsx
  username check
    ✓ knows that Tallround3r is valid username (6ms)
    ✓ knows that empty string is incorrect username (1ms)
  email check
    ✓ knows that Tim.Kertzsch@web.de is valid Email
    ✓ knows that mail with missing @ is incorrect Email
    ✓ knows that mail with missing . is incorrect Email (1ms)
    ✓ knows that empty String is incorrect Email
  password check
    ✓ knows when passes are matching
    ✓ knows when passes dont match
    ✓ knows that empty string is not a pass (1ms)
  name check
    ✓ knows that Tallround3r is valid username
    ✓ knows that Tim is valid firstname
    ✓ knows that Kertzsch is valid lastname
    ✓ knows that empty string is valid username (1ms)
    ✓ knows that empty string is valid firstname
    ✓ knows that empty string is valid lastname

Test Suites: 1 passed, 1 total
Tests: 15 passed, 15 total
Snapshots: 0 total
Time: 1.89s, estimated 2s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

However we have to admit, that unit tests are no reliable quality tool for our project. Every component within our application relies heavily on communicating with firebase, our Redux store and our other components. So mocking all these interactions will take away most of the functionality of each unit. Therefore our existing tests with Cucumber are more realistic scenarios for our application. Nevertheless we will use both, function and unit testing, to ensure Perfect Time is running smoothly. There are some other test related decisions we had to make. You can find all of them in our [7]test plan on GitHub.

You can always check our Test status with our status badges on [8]GitHub. There you will also see our Code Coverage and Code Quality. We are using Codecov to analyze our Code Coverage and Codacy to analyze Code Quality.

So that is all we accomplished this week. But stay tuned, since there is always more to come.

Your

Perfect Time Crew

1. <https://jestjs.io/>
2. [https://www.jetbrains.com/help/webstorm/react.html#react\\_testing\\_application](https://www.jetbrains.com/help/webstorm/react.html#react_testing_application)
3. [https://www.jetbrains.com/pycharm/guide/tutorials/react\\_typescript\\_tdd/](https://www.jetbrains.com/pycharm/guide/tutorials/react_typescript_tdd/)
4. [https://github.com/Tallround3r/PerfectTime/blob/feature\\_testing/.travis.yml](https://github.com/Tallround3r/PerfectTime/blob/feature_testing/.travis.yml)

5. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/register/register\\_useCase.md](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/register/register_useCase.md)
6. [https://github.com/Tallround3r/PerfectTime/blob/feature\\_testing/src/register.test.tsx](https://github.com/Tallround3r/PerfectTime/blob/feature_testing/src/register.test.tsx)
7. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Testing/TestPlan.md>
8. <https://github.com/Tallround3r/PerfectTime>

---

EventLAB Team (2019-04-30 08:51:03)

Hey guys, this week you have proven that you can manage to accomplish a lot of tasks in a short period. You did a great job implementing new functionality and also creating tests for your code. The test plan you wrote is very detailed and gives a good overview about all the points regarding your testing. (Just a hint: In your test plan at point 1.5 where you linked your use cases the links 'UC Copy Trip' and 'UC Edit Activity' are broken). In addition you also inserted some great badges which we really like. You fulfilled all grading criteria this week and we are looking forward to see your progress in the next weeks :)  
Your EventLAB Team

perfecttimese (2019-04-30 09:06:58)

Links are fixed. Thanks for the hint. Your Perfect Time Crew

jonny9298 (2019-05-07 09:14:51)

Hey, First of all I really like that you already have badges up and running on your github. I've still got some things to criticize, though: This is about you "using Cucumber and TestCafe to test [y]our application", while you haven't even had tests. The point of those test suites is to have test running and certifying that your code actually works. Thus without any tests, you haven't really been using those tools (while you still might have already integrated them in your project). Furthermore (which isn't about the topic of testing, though) your Readme on Github seems to be some tutorial about React, which you have copy-pasted in there. This is imo a big no-go, as this doesn't give someone who tries to test or use your application any instructions on how to setup and run your project, while the information you have given there can be found with a simple search query on thousand of different websites on the internet. Other than that, you have fulfilled the grading criteria (and even some of the additional points) to my liking. Your Jonny from turnie.re

Testing in a nutshell (2019-06-02 14:56:58)

[...] We use unit tests to verify our application logic is working. For our unit tests we are using JestJs. You can look at our test files here. Our code coverage metric is based on the logic covered by unit tests. You can see a successful unit test below. All unit tests run automatically with every Travis build. (Original blog post) [...]



## 2.2 May

### Learning Refactoring from M. Fowler (2019-05-05 18:00)



Hi guys,

this week we received a special assignment from our professor. We should read the first chapters of M. Fowler's *Refactoring - improving the design of existing code*. We did that and were amazed how many of the issues he mentioned in his books were standard functionality of our modern IDEs. E.g. Alt+Shift+R (in Eclipse) is a perfect standard solution to rename variables, classes, methods and packages. Alt+Shift+M extracts code into a separate and it also uses the intended method signature (well, most of the time ;-)). We couldn't picture an IDE without it.

To prove how well versed we already are with refactoring and what we could learn from Fowler's book, we were challenged to individually refactor a Java code example of the books. We used polymorphism to replace switch-case-statements, renamed methods whose name could lead to wrong assumptions and extracted many lines-of-code from long methods. Here are our results and some issues we focused on:

- [1]Jan: He used Java 8 and the Assertions library to make things easier.
- [2]Lea: She likes using the good old Java 1.2.
- [3]Lukas: He renamed almost every variable and method for a better readability.
- [4]Tim: Liked enums, but then remembered "OO Best Practice" lecture.

But obviously we couldn't slow down in our work on Perfect Time, so despite our additional homework we continued coding. We finish the follow other users basic functions and we started working on a search functionality.



We also continued to work on testing our application.

So that's it for this week. We hope that you will take a look at our refactored code and please let us know what you think. And as always, stay tuned since there is a lot more to come.

Your

Perfect Time Crew

1. [https://github.com/jasonmf123/DHBW\\_SE\\_REFACTORING\\_HW](https://github.com/jasonmf123/DHBW_SE_REFACTORING_HW)
2. [https://github.com/LeaFries/Fowler\\_Refactoring](https://github.com/LeaFries/Fowler_Refactoring)
3. <https://github.com/kr4cher/dhbw-se-fowler>
4. [https://github.com/Tallround3r/SE\\_Fowler\\_Refactoring](https://github.com/Tallround3r/SE_Fowler_Refactoring)

---

Deminder Team (2019-05-06 12:00:33)

Hi guys! I looked into your git repositories and want to leave a comment for each of them: Jan: You have a lot of commits and one can see, that you did all the work, very good! But you could have committed a few more times. When it came to generating those price-classes, three refactor-methods were used in that process, but I can not see all of them in your commits. Lea: Good committing. One can see your process. Lukas: You did some fixes and refactoring, but one can get the impression that you didn't work yourself through the first chapter completely. Maybe you can consider doing that. Tim: You too still have to work yourself through the first chapter. That's all folks! Keep up the good work! Regards, Thomas

EventLAB Team (2019-05-06 13:15:14)

Hi guys, you have done a good job as always! It's great to see how you could improve the code quality a lot by following the thoughts in the first chapter of the book. Furthermore, you could transfer knowledge from other courses (OO Best Practice) to create simple and intuitive code that everyone can understand! From our point of view, there is nothing to criticize ;) Also, we really like the explanations of the automatic refactor functionality in your IDE. This shows how easy refactoring can be if you use the right tools! Keep up your good work! Your friends from EventLAB

perfecttimese (2019-05-06 18:41:23)

Thank you for all your comments. We appreciate them. Since the points you mentioned are not related to our project and we will not continue to work on the refactored code, we won't reply to all the points mentioned. Nevertheless, we all read your remarks and we will take them into consideration when we will refactor our Perfect Time source code. So thank you again for taking the time to read our blog and work through our repositories. Your friends from Perfect Time

"I don't trust words, I even question actions, but I never doubt patterns." (2019-05-12 18:00)



Quote origin: [1]unknown.

Hey guys,

last week we worked with patterns. Yes we know it is quite late in the development process to introduce a pattern now, but hey, we just covered it in our lecture.

Our task was to introduce a new pattern into our project and to refactor our existing code accordingly. Unfortunately we weren't allowed to use the React Flux Pattern as our pattern, because it is provided by our framework. That challenged us quite a bit, because the patterns that we had been introduced to in class, were designed for object-oriented languages. But we are using React and TypeScript, so these patterns cannot be applied to Perfect Time.

To master this challenge, we did some digging online and we found design patterns for React. You can read the [2]article and the [3]blog post under the given links. We recognized that we are already using some of the mentioned patterns (apart from those React forces us to use anyway). We use the Global App Data (Pattern) by having added Redux into our project to store the state across all our components. We are using Higher Order Components (HOC Pattern) to simplify our application structure. And we use the [4]Container-Pattern in some cases to build our website structure in a more intuitive way. However we do that "by chance" in some area, so we decided to do it by choice for our whole code in this week's sprint.

The **container pattern** uses the separation of data fetching and rendering the view elements. To say it with the words of Jason Bonta: "A container does data fetching and then renders its corresponding sub-components. That's it." The container is the data logic layer and utilizes stateful APIs. The container calls its subcomponents

in the render()-method. The subcomponents deal with the individual UI-elements. They utilize stateless APIs. Thereby we have contained our state within the container and do not have to worry about synchronization and similar concerns in our components. We can also reuse our defined components in different containers and we only have to set / change their looks and behaviors once. We hope you can understand why we think using the container pattern makes sense for Perfect Time.

As mentioned before, we are already using the container pattern in some areas (e.g. activity card) of our application but didn't recognize it as an overall design approach. Therefor we looked at our existing code and identified three areas in which we have to refactor to implement the component pattern universally:

- Make each trip in the trips overview an individual component
- Make each location in the locations overview an individual component
- Make the follow / unfollow button an own component.

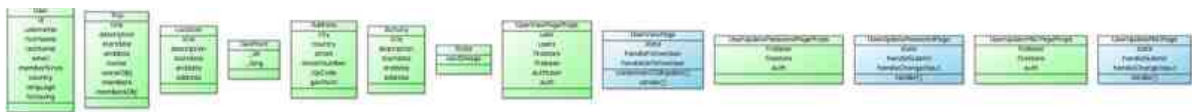
We made these changes and you can see our code [5]before and [6]after the implementation of the component pattern. We tried to generate some class diagrams (before and after), but as we told you in [7]past blog posts, we do not use classes. And the tool (WAVI) we used in the past to "fake" a class diagram, does not work with Typescript. So we looked at some tools for Typescript. We found [8]tsviz and [9]tsuml. However both tools are unable to resolve the dependencies of our project, because the frameworks that we are using are written in JavaScript while we are using Typescript. The tools cannot handle the different file dependencies. The results look like this:



wavi result (misses all .tsx files)

```
C:\Users\Service\Desktop\PerfectTimeWorkspace\PerfectTime\src\container\documentations\classwavi\beforewavi\pattern.png
Unable to resolve type: 'React.Component<ActivityAddPageProps, State>'
Unable to resolve type: 'React.Component<ActivityEditPageProps, State>'
Unable to resolve type: 'React.Component<ActivityViewPageProps, State>'
Unable to resolve type: 'React.Component<Props, State>'
Unable to resolve type: 'React.Component<Props, State>'
Unable to resolve type: 'React.Component<Props, State>'
Unable to resolve type: 'React.Component<Props>'
Unable to resolve type: 'React.Component<Props, State>'
Unable to resolve type: 'React.Component<Props, State>'
Unable to resolve type: 'React.Component<Props, State>'
Unable to resolve type: 'React.Component<Props, State>'
Unable to resolve type: 'React.Component<Props>'
Unable to resolve type: 'React.Component<UserDeletePageProps, State>'
Unable to resolve type: 'React.Component<UserEditPageProps, State>'
Unable to resolve type: 'React.Component<Props>'
Unable to resolve type: 'React.Component<UserUpdateMailPageProps, State>'
Unable to resolve type: 'React.Component<UserUpdatePasswordPageProps, State>'
Unable to resolve type: 'React.Component<UserViewPageProps, State>'
Found 22 module(s)
Could not find @agoric in PATH.
events.js:113
    throw err; // Unhandled 'error' event
    ^
Error: Could not find @agoric in PATH
    at Function.resolve (C:\Users\Service\Desktop\PerfectTimeWorkspace\PerfectTime\src\container\documentations\classwavi\beforewavi\pattern.png:1:1)
```

tsviz result (error)



tsuml result (no dependencies)

Apart from that struggle we continued implementing additional functionality and tests. We finally completed the follow user functionality. You can follow other user by clicking the add person icon next to the user name. If you are already following another user, another icon will be shown. You can click this outlined person icon to stop following the user.

In addition we started implementing export trip. We wrote additional tests. We worked on the backend authorization validation to improve security and we worked on our applications performance.

So we had a busy week. We hope you can see our progress and you understand our design pattern choice. Thanks for reading. Please leave a comment if you want to share your thoughts and as always, stay tuned since there is a lot more to come.

Your

Perfect Time Crew

1. <https://medium.com/gc-entrepreneur/i-dont-trust-words-i-even-question-actions-but-i-never-doubt-patterns-9278f12f655e>
2. <https://medium.com/@joomiguelcunha/react-patterns-you-should-know-da86568372fa>
3. <https://lucasmreis.github.io/blog/simple-react-patterns/>
4. <https://medium.com/teamsubchannel/react-component-patterns-e7fb75be7bb0>
5. <https://github.com/Tallround3r/PerfectTime/tree/e7881274c8003e2162c12d1df69de54cb2365672>
6. <https://github.com/Tallround3r/PerfectTime/tree/533f115ef27eda450447eacf6c6cc2e6714f6cd6>

7. <https://perfecttime608150251.wordpress.com/2018/11/10/creating-our-class-diagram-and-our-database-schema/>
8. <https://github.com/joaompneves/tsviz>
9. <https://github.com/remojansen/TsUML>

---

EventLAB Team (2019-05-13 16:02:23)

Hey guys, you have done a lot of work last week! We're sorry to hear that it took so much effort to find suitable patterns for your project and to generate some kind of "class diagram". Nevertheless, you have done a good job with a lot of research, and all the grading criteria seem to be fulfilled. We are also following your implementation and are looking forward to your final application! Keep up the good work! Best regards, Your EventLAB Team

MNZ Team (2019-05-14 08:59:44)

Hey guys, we are happy to read that you mastered the task even though it is hard to find a pattern that makes sense to implement for React and isn't already handled by default. It's obvious that you put a lot of work into this and it looks like it was well worth it. You should probably also include links to the full images you included in the blog entry since it is barely readable if at all. Besides that, you did a really good job and we are excited for next week. Best regards MNZ-Team

perfecttimese (2019-05-14 10:19:32)

Thank you for your comments. We appreciate it. The diagram pictures do not show any relevant data, they only show failure. We also didn't upload them to Git, because they have no further value for this project. We hope that you understand our reasoning. :-D Your friends from Perfect Time

Looking back and moving forward (2019-05-20 00:54:17)

[...] now. We additionally decided to assign tasks to people at the beginning of each sprint. And we dropped creating class Diagramms, because we want to code type safe with TypeScript. If you want to see some issues the other groups [...]

The grand finale – We made it! (2019-06-16 17:01:16)

[...] Blog post for Pattern [...]

## Looking back and moving forward (2019-05-19 18:00)



Hey guys,

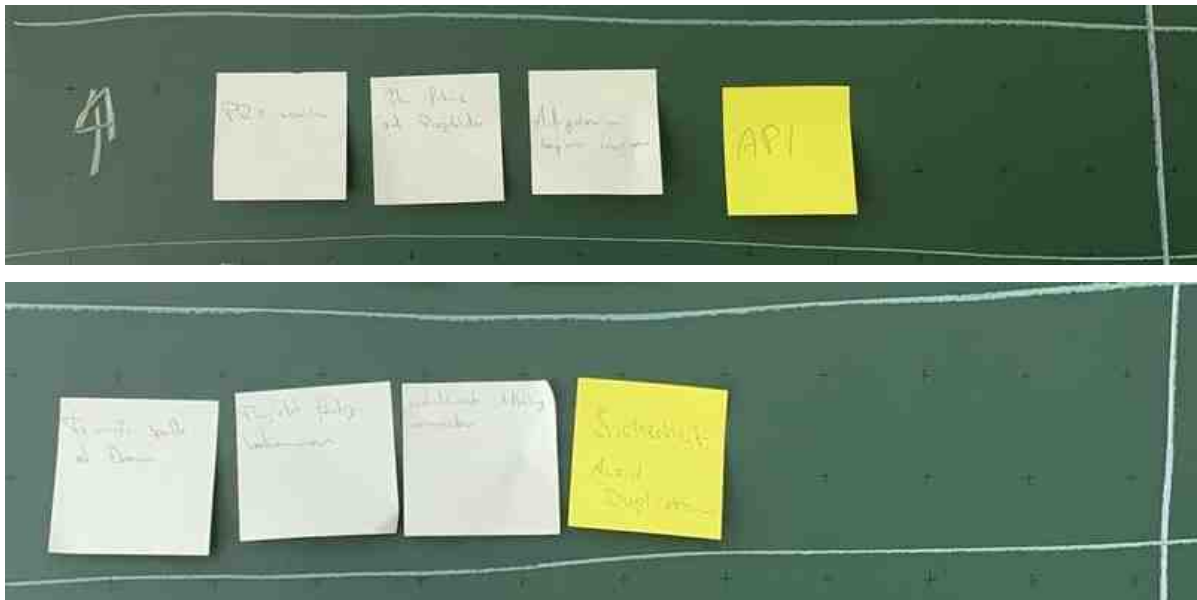
last week we had the second retrospective in our class. During this meeting with the whole class we looked at several questions.

First we looked back at the last [1]retrospective and what changed since then. (We wanted to get YouTrack working again, focus on implementation and test our results before merging.) We decided to keep all those points and they are working now. We additionally decided to assign tasks to people at the beginning of each sprint. And we [2]dropped creating class Diagramms, because we want to code type safe with TypeScript. If you want to see some issues the other groups mentioned, please take a look at the next picture.



We also decided on three additional rules to comply to this semester.

1. We want to define interfaces in our application before we start implementing a new feature / use case. Thereby we want to ensure security and reduce the bug potential of new implementations.
2. We want to use Pull Requests to match our 'To Verify'-Column of our SCRUM-board. We currently have many tasks remaining in 'To Verify' until the last day of the sprint. We want to avoid that in the future to finish tasks more quickly.
3. We want to make the idea behind Perfect Time the focus of our efforts. We recognized that with all the additional tasks, we often loose focus of our customer benefit. That will change for the remaining sprints. We are 100 % dedicated to finish the defined features in this semester.



We liked this retrospective. It showed us how far we have come and what we still need to improve. We also got to know an agile method called Fishbowl. We think that this is a fun way to share your thoughts with your colleagues. We will definitely join the next Fishbowl-Events in our companies. We think that many of the issues (e.g. about defining interfaces and APIs prior to implementation) can be applied our professional careers.

Apart from the retrospective, we continued implementing tests and working on implementing export trips. We continued to work on improving our application's performance. BTW our current version has been deployed, so you can take a look [\[3\]here](#).

In Terms of testing we implemented more and more [\[4\]unit tests](#) for classes where they are useful. Also we configured [\[5\]CodeCov](#) to check the coverage of our unit (Jest) tests with every Travis build. As described in the post on April 28 it is really difficult to implement unit tests in most areas of our application because of firebase and storage calls (which extend the meaning of unit tests). Due to this we decided to **exclude** these files from our code coverage. Files we can't even tests should no appear in a coverage report. To make it easier for you guys to see our coverage we implemented a badge in our [\[6\]GitHub repo](#). In our existing and new tests we implemented [\[7\]faker](#), a tool that makes generating random data like emails, names, adresses and more very easy.



A new way of testing our application is security and access control testing. Therefore we also use Jest as known from our unit tests. What we want to achieve by security and access control testing is to be sure that for example only users who own the trips are able to delete them also only users who are logged in and owner or member should be allowed to edit trips. There are many other rules we are testing, you can see them by looking at our [8]Jest files. Inside these test files we send API requests to the Firebase backend with different configurations (most of the configuration is faked with [9]faker again) and check the results with expectations. Firebase helps us a lot in terms of controlling access by allowing us to define detailed rules for database access.

The following two picture will show you how our test-suites covers our activity diagram branches by the example of creating trips:

```
describe('name: 'create TRIPS', fn() => {

  const trip = {
    title: fakeTitle,
    description: fakeDescription,
    startdate: fakeStartDate,
    enddate: fakeEndDate,
    public: false,
    owner: process.env.ID_BRUCE_LEE,
    member: [process.env.ID_TIM_TESTER],
  };

  it('name: 'is possible to create a TRIP for logged in user', fn async () => {
    let tripCreated = false;

    await dbTripRef.doc(randomTripID).set(trip).then(() => {
      tripCreated = true;
    }).catch(() => {
      tripCreated = false;
    });

    expect(tripCreated).toBe(expected: true);
  });

  it('name: 'is not possible to create a TRIP as not logged in', fn async () => {
    await auth.signOut();

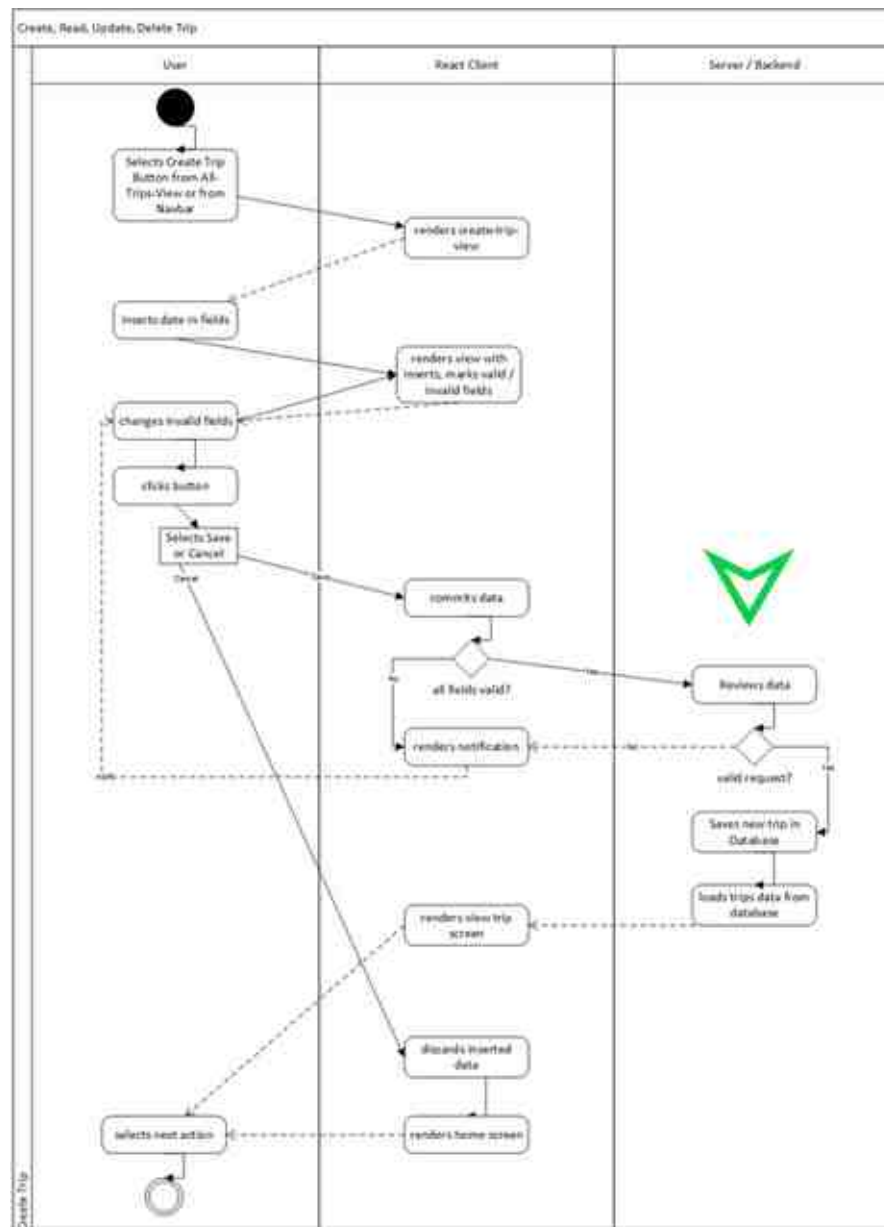
    let tripCreated = false;

    await dbTripRef.doc(randomTripID).set(trip).then(() => {
      tripCreated = true;
    }).catch(() => {
      tripCreated = false;
    });

    expect(tripCreated).toBe(expected: false);
  });
});
```

Here you are able to see the [10]test code that tests if only logged in users are able to create trips





The API calls executed by the Jest tests will enter the activity diagram of "[11]CRUD Trips" at the green arrow, as if it came from a user

That's all for this week. Please let us know what you think in the comments. And as always stay tuned since there is a lot more to come.

Your

Perfect Time Crew

PS: We weren't asked in the retrospective, but if we should choose a car to illustrate our situation, we would choose this:



1. <https://perfecttime608150251.wordpress.com/2018/11/25/working-with-the-scrum-professionals/>
2. <https://perfecttime608150251.wordpress.com/2019/05/12/i-dont-trust-words-i-even-question-actions-but-i-never-doubt-patterns/>
3. <https://perfecttime-planyourtrip.firebaseio.com/>
4. <https://github.com/Tallround3r/PerfectTime/tree/master/src/unitTests>
5. <https://codecov.io/gh/Tallround3r/PerfectTime>
6. <https://github.com/Tallround3r/PerfectTime/blob/master/README.md>
7. <https://www.npmjs.com/package/faker>
8. <https://github.com/Tallround3r/PerfectTime/tree/master/src/securityTests>
9. <https://www.npmjs.com/package/faker>
10. [https://github.com/Tallround3r/PerfectTime/blob/testing\\_security/src/securityTests/trips.test.tsx](https://github.com/Tallround3r/PerfectTime/blob/testing_security/src/securityTests/trips.test.tsx)
11. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/CRUDTrips/CRUD\\_trips.md](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/CRUDTrips/CRUD_trips.md)

---

CodeCrunch (2019-05-21 23:19:05)

Greetings PERFECTTIMESE, I must say that I am really impressed with the blogpost you published this week. While reading the first sentences of the blogpost it is immediately clear that you put some real effort in to this blogpost. From the way the blogpost follows a clear structure to the content which resides in the structure itself, the blogpost ticks all the right boxes. The only thing which I could complain about is the fact that the future blogposts of my team and all other software engineering teams for that matter have to be measured relative to what you have achieved here. What I enjoyed most out of the blogpost... well is the quantity and the quality of information you provided about virtually everything you could give information about in the context of this blogpost's, the way that the written part is supported with links and pictures was also a nice treat. As a proof that I actually read your blogpost and did not just give you a lot of compliments with no bases, with the hopes that doing so will provide me with minimum effort to write a comment for this blogpost. I want to mention that I find it very meaningful that you have decided to overhaul your code coverage by excluding files, which you are aren't able to test currently and I can definitely emphasize that at the current stage

of your project, you feel like a truck having to move a great load under difficult conditions. I really hope that you can find a way of putting some of the same effort you put in to this blog post in the service you are building. If you manage to do so, I think that at some point you will not feel like a truck on the verge of breaking down but instead like something more favourable, which is capable of bringing your project to completion easily !!! ;) Best regards, Falko@CodeCrunch

CodeCrunch (2019-05-21 23:21:35)

Greetings PERFECTTIMESE, I must say that I am really impressed with the blogpost you published this week. While reading the first sentences of the blogpost it is immediately clear that you put some real effort in to this blogpost. From the way the blogpost follows a clear structure to the content which resides in the structure itself, the blogpost ticks all the right boxes. The only thing which I could complain about is the fact that the future blogposts of my team and all other software engineering teams for that matter have to be measured relative to what you have achieved here. What I enjoyed most out of the blogpost... well is the quantity and the quality of information you provided about virtually everything you could give information about in the context of this blogpost's, the way that the written part is supported with links and pictures was also a nice treat. As a proof that I actually read your blogpost and did not just give you a lot of compliments with no bases, with the hopes that doing so will provide me with minimum effort to write a comment for this blogpost. I want to mention that I find it very meaningful that you have decided to overhaul your code coverage by excluding files, which you are aren't able to test currently and that I can definitely emphasize that at the current stage of your project, you feel like a truck having to move a great load under difficult conditions. I really hope that you can find a way of putting some of the same effort you put in to this blog post in the service you are building. If you manage to do so, I think that at some point you will not feel like a truck on the verge of breaking down but instead like something more favourable, which is capable of bringing your project to completion easily !!! ;) Best regards, Falko@CodeCrunch

Getting the numbers right – working on our metrics (2019-05-26 18:47:45)

[...] week we worked on our project metrics. As we mentioned before, we use Codacy to measure our code [...]

Testing in a nutshell (2019-06-02 14:57:02)

[...] We are testing the security and access control of our application. Therefor we use specialized feature tests. You can access the test files here. The security tests user faker to generate testing data, so we don't have use our database for that. The images below show a security test and how it covers the different paths through the activity diagram. (Original blog post) [...]

## Getting the numbers right - working on our metrics (2019-05-26 18:00)



Hey guys,

this week we worked on our project metrics. As we mentioned [1]before, we use [2]Codacy to measure our code quality.



Every build and every pull request runs through our Codacy and Codecov evaluation.

The image shows a GitHub pull request interface. At the top, a list of commits is visible, including 'Implemented firebase function onDeleteTrip', 'remove pr', 'Merge pull', 'refactor', and 'refactor'. A status bar indicates 'All checks have passed' and '1 successful check'. A Codacy/PR Quality Review badge is shown with a green checkmark and the text 'Up to standards. A pos...'. A 'Details' link is present.

Below the pull request, a 'Merged' badge is shown. The commit message is 'Testing security #58' by 'jasonmfi23', merged 11 commits into 'master' from 'testing\_security' 18 hours ago.

The main part of the image is a Codecov Report. It features the Codecov logo and the title 'Codecov Report'. The report states: 'Merging #58 into master will not change coverage. The diff coverage is n/a.' Below this is a large green rectangular area. A table shows coverage metrics for 'master' and '#58'.

	Coverage	Diff	
	master	#58	+/-
Coverage	100%	100%	
Files	5	6	+1
Lines	21	29	+8
Branches	0	0	+0
+ Hits	21	29	+8

Below the table is a section titled 'Impacted Files' with a table showing the coverage change for 'src/firebase/firebase.tsx'.

Impacted Files	Coverage Δ
src/firebase/firebase.tsx	100% <0%> (0)

At the bottom, a link says 'Continue to review full report at Codecov.'

We wanted to improve our Codacy project grade, so we worked on the issues Codacy identified. You can find our current grade on our [3]GitHub . The code [4]before and [5]after working on our issues can be found on GitHub. For this process we used different metrics which will be explained in this blog post.

We learned during our class about different metrics. We looked at the different measurement tools (Sonar, CRAP, ...) mentioned in our course. However we decided to stick with our metrics that Codacy provides us with. So we focused on

- Number of complex files (Complexity > 20)

This metric includes several complexity calculations (e.g. cyclical complexity). We cannot calculate an overall complexity, because our application is separated in different containers, which are independent. Therefore it is better to calculate complexity for the different files, and only look at the complex files. The number of complex files is zero for our project, because React is based on different components which are by default non cyclical.

- Error prone issues

This is a number of issues which are likely to cause errors. One might want to argue that these issues are code smells, but this is only partially true. Code smells can be error prone issues, but not every one of these issues has to be a code smell. E.g. a lacking default case in a switch-case-statement is no code smell (even if the whole switch case might be). It also does not stop the project from compiling. Nevertheless not defining a default case can lead to an error if an unexpected and undefined case occurs. Having no error prone issues shows a good reliability and correctness (goals of metrics) of the project.

**features/support/pages/const-selection.js**

**Expected a default case. (default-case)**

```
32 switch(selection){
```

**features/support/pages/const-url.js**

**Expected a default case. (default-case)**

```
11 switch (url) {
```

Another important aspect for our application is performance/speed. Therefore we used [6]Google lighthouse to measure our application's performance.

- Performance

Google Lighthouse uses several metrics and combines them to a performance score. The metrics are different response times. This helps the metrics goal of efficiency.



As you can see above our application is currently very slow to start. It takes over 5 seconds between the http-request and the first loaded screen. That kills our performance. We are working on decreasing the start-up time. But so far we haven't achieved any significant progress for that matter. But we will keep working on it. Another issue we did not attempt, was the issue of our testController interface (for our feature tests). It is considered to be an error prone issue, but we cannot change it. Our test framework cucumber requires us to use the controller in that specific way. Therefore we instructed Codacy to ignore the issue.

```
'testController' is not defined. (no-undef)
6  const button = select(element).with({boundTestRun: testController});

'testController' is not defined. (no-undef)
15 await testController.typeText(select('email').with({boundTestRun: testController}), 'test@user.de');

'testController' is not defined. (no-undef)
22 await testController.typeText(select('first name').with({boundTestRun: testController}), 'test');

'testController' is not defined. (no-undef)
23 await testController.typeText(select('last name').with({boundTestRun: testController}), 'test');
```

testController seems to be undefined for Codacy, but Cucumber requires this command

There were some other issues with imports, that we didn't fix, because our frameworks require us to use them.

All in all we worked on the issues Codacy pointed us to. We changed the code to make it less cyclical (if possible at all) and less error prone. Thereby we are certain, that our code quality has improved and we are less likely to run into problems (bugs, hard to maintain code,...) in the future. Codacy grades our project with an A now. We have to admit, it is way more fun to improve your code, if the grade increases and you get an instant feedback. ;-)



## Project certification

### Quality evolution

Last 7 days

Last 31 days



### Issues breakdown



### Coverage



Apart from the metrics we worked on implementing export trip and upload pictures (for locations and activities). We are also working on increasing our performance.

So that's all for this week. We hope you can understand why we used these metrics. Let us know what you think in the comments. And as always, stay tuned since there is still more to come.

Your

Perfect Time Crew

1. <https://perfecttime608150251.wordpress.com/2019/05/19/looking-back-and-moving-forward/>
2. <https://www.codacy.com/>
3. <https://github.com/Tallround3r/PerfectTime>
4. <https://github.com/Tallround3r/PerfectTime/tree/3d06e97d07c44fd83e2cb38b28e32db1feb9be50>
5. <https://github.com/Tallround3r/PerfectTime/tree/79eec3d505004f705a0f4dafdda1081a6c7ce2a8>
6. <https://developers.google.com/web/tools/lighthouse/>

EventLAB Team (2019-05-27 10:35:49)

Hey guys, you've done a good job this week. You clearly stated the metrics and tools you used and how they are integrated into your build and deployment process. Furthermore, you showed how you modified your code so that the evaluation from Codacy could be improved. You also gave good reasons for why you will not modify certain spots where Codacy



expects you to make changes. We're sorry to hear about the poor performance of your application (initial loading time) and hope you will find something to improve there as well. But nevertheless, we enjoy your application and are looking forward to your demo tomorrow! Best regards, Your EventLAB Team

Felix Hausberger (2019-05-27 21:40:57)

Dear Team PercetTime, I really enjoyed reading through your blog post this week! Especially it was quite informational for me reading about Google Lighthouse to measure performance, I didn't know this tool before. Perhaps dashup will copy this in future ;). I as well appreciated to hear, that your choice of technology stacks prevents you from bad metrics (like using component-based frameworks like react). Furthermore, you proved that you have integrated metrics tools in your deployment process, included commits to compare to each other, posted images showing your improvement progress and talked about your problem spot of performance. Even the test plan is updated. So in the end, you really did a great job! Nothing left to mention. Keep going like that! Yours sincerely, Team dashup

Testing in a nutshell (2019-06-02 14:57:09)

[...] Both tools have badges shown on our GitHub main page. (Original blog post) [...]

The grand finale – We made it! (2019-06-16 17:00:43)

[...] Test plan with function, UI, security, installation and Unit testing; includes the used tools and the metrics [...]

## 2.3 June

### Testing in a nutshell (2019-06-02 14:56)



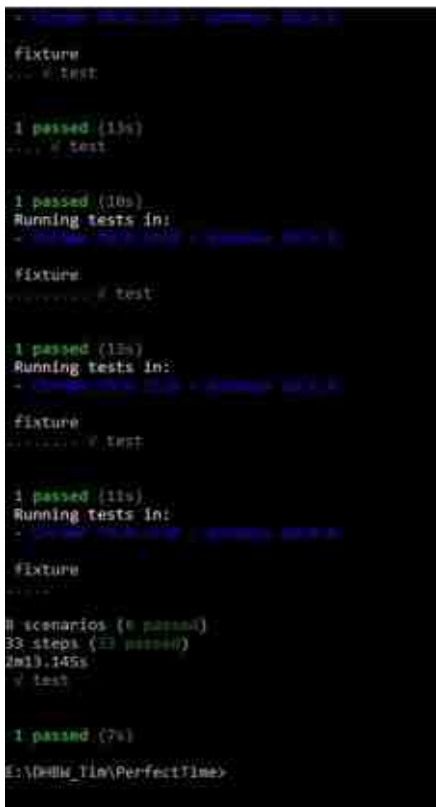
Hey guys,

it has come to our attention that we explained our tests in a variety of blog posts. It is quite difficult to see the big picture across all these different pages. So we decided to provide all our readers with a combined overview: testing in a nutshell.

## What tests ware we using?

- feature tests

We are using feature test to analyze our implemented functionality. Therefor we use [1]Cucumber and [2]Test-Cafe. You can look at our feature files [3]here. Because of our limited time for this project we can't implement tests for each use case. You will see that by looking at the comments in our feature files. (Original [4]blog post)



```
1 passed (13s)
Running tests in:
Fixture
1 passed (13s)
Running tests in:
Fixture
1 passed (11s)
Running tests in:
Fixture
4 scenarios (4 passed)
33 steps (33 passed)
2m11.145s
1 passed (7%)
E:\DHLW_T1a\PerfectTime>
```

- unit tests

We use unit tests to verify our application logic is working. For our unit tests we are using [5]JestJs. You can look at our test files [6]here. Our [7]code coverage metric is based on the logic covered by unit tests. You can see a successful unit test below. All unit tests run automatically with every [8]Travis build. (Original [9]blog post)

```
CA: npm
PASS src/register.test.tsx
  username check
    ✓ knows that Tallround3r is valid username (6ms)
    ✓ knows that empty string is incorrect username (1ms)
  email check
    ✓ knows that Tim.Kertzsch@web.de is valid Email
    ✓ knows that mail with missing @ is incorrect Email
    ✓ knows that mail with missing . is incorrect Email (1ms)
    ✓ knows that empty String is incorrect Email
  password check
    ✓ knows when passes are matching
    ✓ knows when passes dont match
    ✓ knows that empty string is not a pass (1ms)
  name check
    ✓ knows that Tallround3r is valid username
    ✓ knows that Tim is valid firstname
    ✓ knows that Kertzsch is valid lastname
    ✓ knows that empty string is valid username (1ms)
    ✓ knows that empty string is valid firstname
    ✓ knows that empty string is valid lastname

Test Suites: 1 passed, 1 total
Tests:       15 passed, 15 total
Snapshots:   0 total
Time:        1.89s, estimated 2s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

- security tests

We are testing the security and access control of our application. Therefore we use specialized feature tests. You can access the test files [10]here. The security tests use [11]faker to generate testing data, so we don't have to use our database for that. The images below show a security test and how it covers the different paths through the activity diagram. (Original [12]blog post)

```

describe( name: 'create TRIPS', fn () => {

  const trip = {
    title: fakeTitle,
    description: fakeDescription,
    startdate: fakeStartDate,
    enddate: fakeEndDate,
    public: false,
    owner: process.env.ID_BRUCE_LEE,
    member: [process.env.ID_TIM_TESTER],
  };

  it( name: 'is possible to create a TRIP for logged in user', fn async () => {
    let tripCreated = false;

    await dbTripRef.doc(randomTripID).set(trip).then(() => {
      tripCreated = true;
    }).catch( 'onrejected' () => {
      tripCreated = false;
    });

    expect(tripCreated).toBe( expected: true);
  });

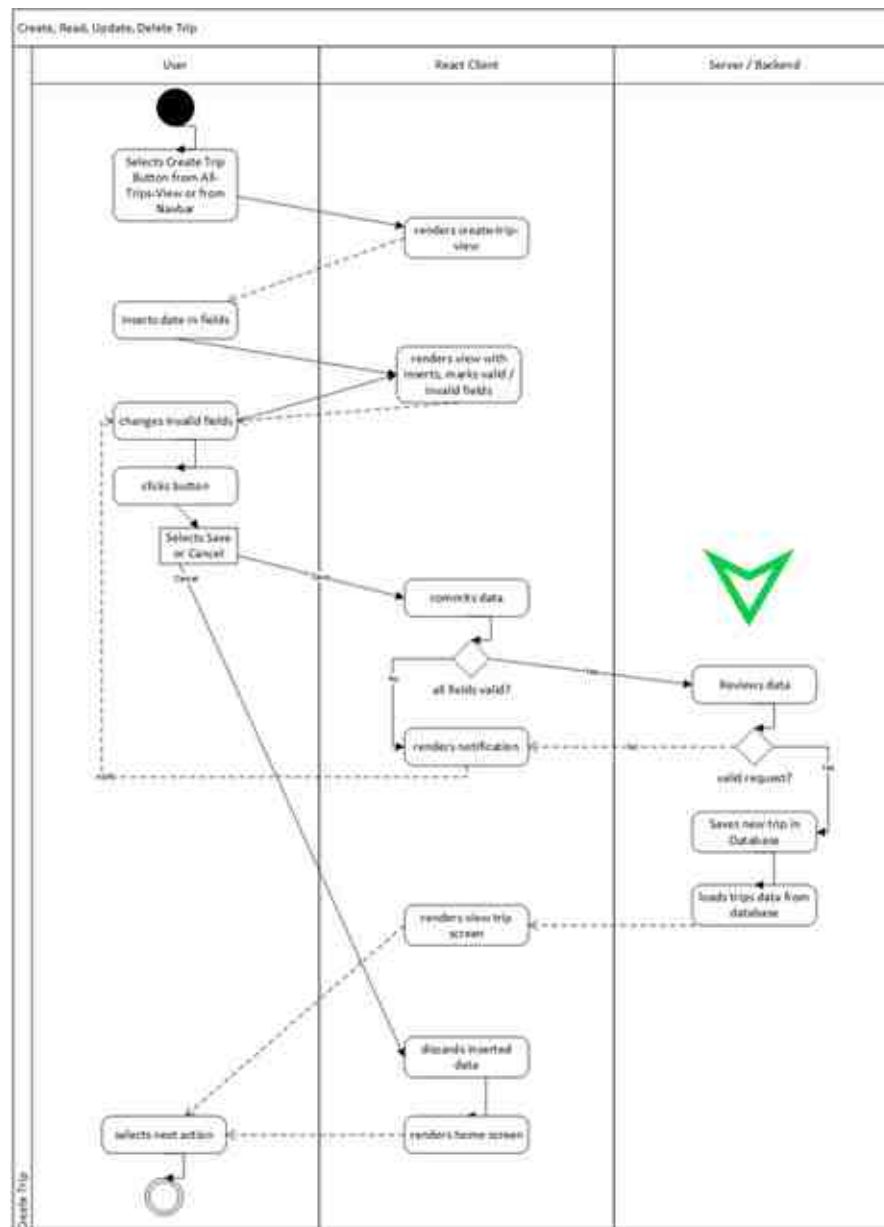
  it( name: 'is not possible to create a TRIP as not logged in', fn async () => {
    await auth.signOut();

    let tripCreated = false;

    await dbTripRef.doc(randomTripID).set(trip).then(() => {
      tripCreated = true;
    }).catch( 'onrejected' () => {
      tripCreated = false;
    });

    expect(tripCreated).toBe( expected: false);
  });
});

```



How are our tests documented?

- test plan

Our test plan covers all test related topics of our project. You can find it [13]here. It includes a detailed description of our different tests, our dependencies and requirements for testing, as well as the metrics related to testing.

- blog posts

There are several blog posts that touch the topic of testing. The major ones are linked under the relating category of this blog post.

- current status

To see our current status, one can always visit our [14]GitHub, [15]Travis CI, [16]Codacy and [17]Codecov. The test plan is also kept up to date.

✕

✕

✕

## How does testing work in our project?

- Integration in Travis CI and our merge process (Codacy & Codecov)

With each Travis CI build, the [18]tests are run and only if they pass, the merge will be executed. Before each merge a Travis CI test build is running, to see if the application is still ready for execution.

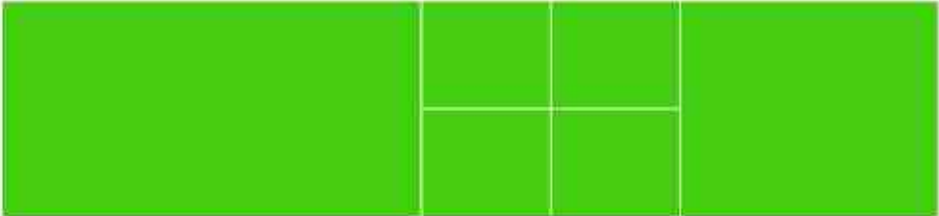
Furthermore Codecov will look at the test coverage for the new logic. Thereby we want to ensure a high test coverage for our application logic.

**Merged** Testing security #58  
jasonmf123 merged 11 commits into master from testing\_security 18 hours ago

codecov-io commented 3 days ago • edited •

## Codecov Report

Merging #58 into master will not change coverage.  
The diff coverage is n/a.



	Coverage	Diff	
	100%	100%	
Files	5	6	+1
Lines	21	29	+8
Branches	0	9	+1
+ Hits	21	29	+8

Impacted Files	Coverage Δ
src/firebase/firebase.tsx	100% <9% (0)

[Continue to review full report at Codecov.](#)

In addition Codacy will look at code smells and metrics to analyze, if the new files are up to code quality standards.

Implemented firebase function onDeleteTrip ✓ efla156

remove pr ✓ ecb00b

Merge pull request 1e5c445

refactor j517683

refactor 67a7149

**All checks have passed**

↑ successful check

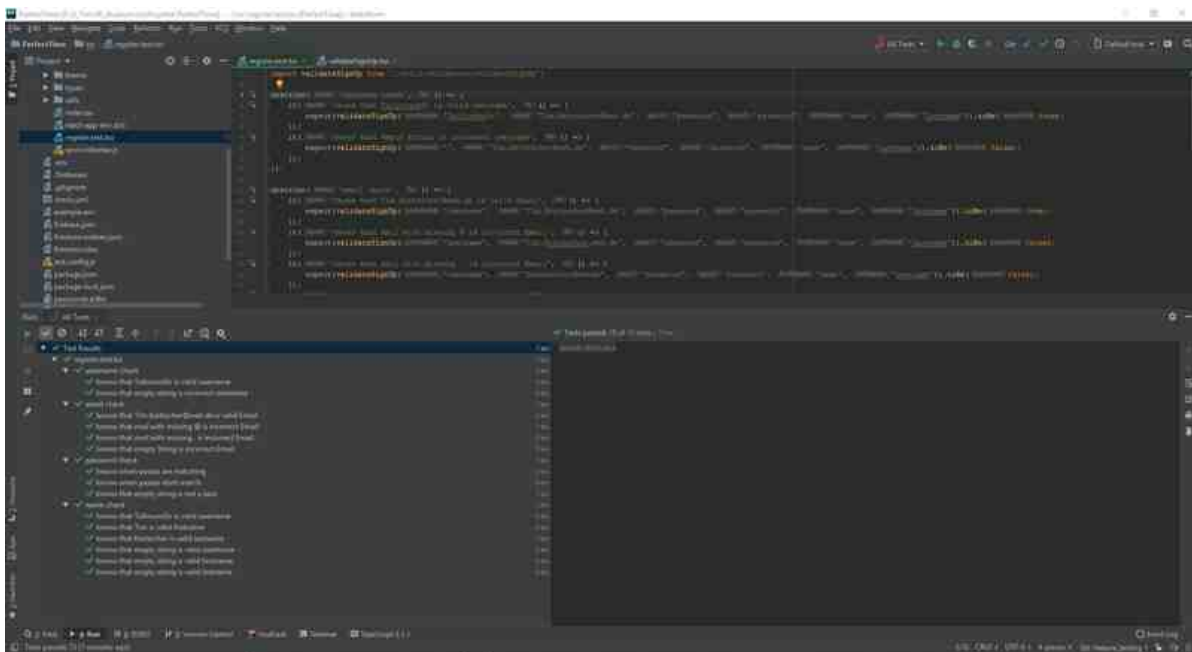
✓ Codacy/PR Quality Review — Up to standards. A post... [Details](#)

Only if those requirements are met, we will merge a branch.

Both tools have badges shown on our [19]GitHub main page. (Original[20] blog post)

- TDD possible with Webstorm

Webstorm allows us to write (unit) tests first and then implement the logic. There are some tutorials that show TDD with [21]Webstorm and [22]PyCharm ( similar) available. The following image shows how easy it is to execute unit tests locally in Webstorm. (Original [23]blog post)



- Using Faker to provide test data

We use [24]FakerJS (NPM dependency). It allows us to generate fake test data. This results in random email addresses and fake usernames. (Original [25]blog post)

- Separating test data and test code

This is quite simple for us. Our data is stored in the (test) database. To feed it for test purposes we use faker in order to provide random input. You can see the implementation in the test below. (Original [26]blog post)



```

describe('name: 'create TRIPS', fn() => {

  const trip = {
    title: fakeTitle,
    description: fakeDescription,
    startdate: fakeStartDate,
    enddate: fakeEndDate,
    public: false,
    owner: process.env.ID_BRUCE_LEE,
    member: [process.env.ID_TIM_TESTER],
  };

  it('name: 'is possible to create a TRIP for logged in user', fn async () => {
    let tripCreated = false;

    await dbTripRef.doc(randomTripID).set(trip).then(() => {
      tripCreated = true;
    }).catch((onRejected) () => {
      tripCreated = false;
    });

    expect(tripCreated).toBe(expected: true);
  });

  it('name: 'is not possible to create a TRIP as not logged in', fn async () => {
    await auth.signOut();

    let tripCreated = false;

    await dbTripRef.doc(randomTripID).set(trip).then(() => {
      tripCreated = true;
    }).catch((onRejected) () => {
      tripCreated = false;
    });

    expect(tripCreated).toBe(expected: false);
  });
});

```

So this is how our testing works. We hope you guys liked our little wrap up. If you have any questions or remarks, please let us know in the comments. And as always stay tuned, since there is, well not much but still a little more yet to come. :-)

Your

Perfect Time Crew

1. <https://cucumber.io/>
2. <https://devexpress.github.io/testcafe/>
3. <https://github.com/Tallround3r/PerfectTime/tree/master/features>
4. <https://perfecttime608150251.wordpress.com/2018/11/03/testing-and-feature/>
5. <https://jestjs.io/>
6. <https://github.com/Tallround3r/PerfectTime/tree/master/src/unitTests>
7. <https://codecov.io/gh/Tallround3r/PerfectTime>
8. <https://github.com/Tallround3r/PerfectTime/blob/master/.travis.yml>
9. <https://perfecttime608150251.wordpress.com/2019/04/28/working-on-the-next-unit-and-testing-it/>
10. <https://github.com/Tallround3r/PerfectTime/tree/master/src/securityTests>

11. <https://www.npmjs.com/package/faker>
12. <https://perfecttime608150251.wordpress.com/2019/05/19/looking-back-and-moving-forward/>
13. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Testing/TestPlan.md>
14. <https://github.com/Tallround3r/PerfectTime>
15. <https://travis-ci.org/Tallround3r/PerfectTime>
16. <https://app.codacy.com/project/Tallround3r/PerfectTime/dashboard>
17. <https://codecov.io/gh/Tallround3r/PerfectTime>
18. <https://travis-ci.org/Tallround3r/PerfectTime/builds/538080294>
19. <https://github.com/Tallround3r/PerfectTime>
20. <https://perfecttime608150251.wordpress.com/2019/05/26/getting-the-numbers-right-working-on-our-metrics/>
21. [https://www.jetbrains.com/help/webstorm/react.html#react\\_testing\\_application](https://www.jetbrains.com/help/webstorm/react.html#react_testing_application)
22. [https://www.jetbrains.com/pycharm/guide/tutorials/react\\_typescript\\_tdd/](https://www.jetbrains.com/pycharm/guide/tutorials/react_typescript_tdd/)
23. <https://perfecttime608150251.wordpress.com/2019/04/28/working-on-the-next-unit-and-testing-it/>
24. <https://www.npmjs.com/package/faker>
25. <https://perfecttime608150251.wordpress.com/2019/05/19/looking-back-and-moving-forward/>
26. <https://perfecttime608150251.wordpress.com/2019/05/19/looking-back-and-moving-forward/>

---

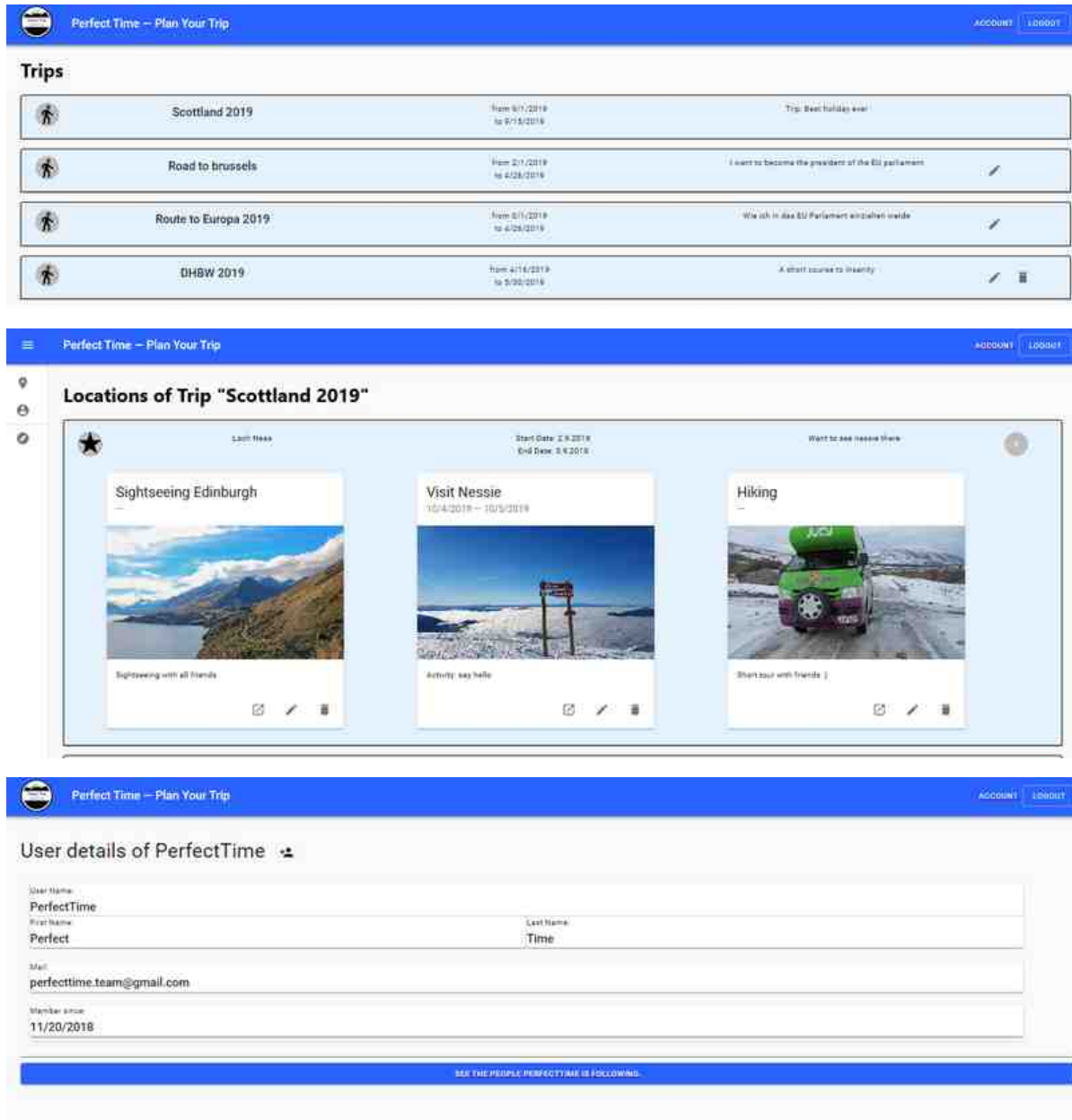
The grand finale – We made it! (2019-06-16 17:00:45)  
[...] For more information please read this summary [...]

## Getting ready for the grand finale (2019-06-02 16:12)



Hey guys,

last week we gave a short demo of our application in front of the whole course. We want to show you some screenshots of the stuff we showed.

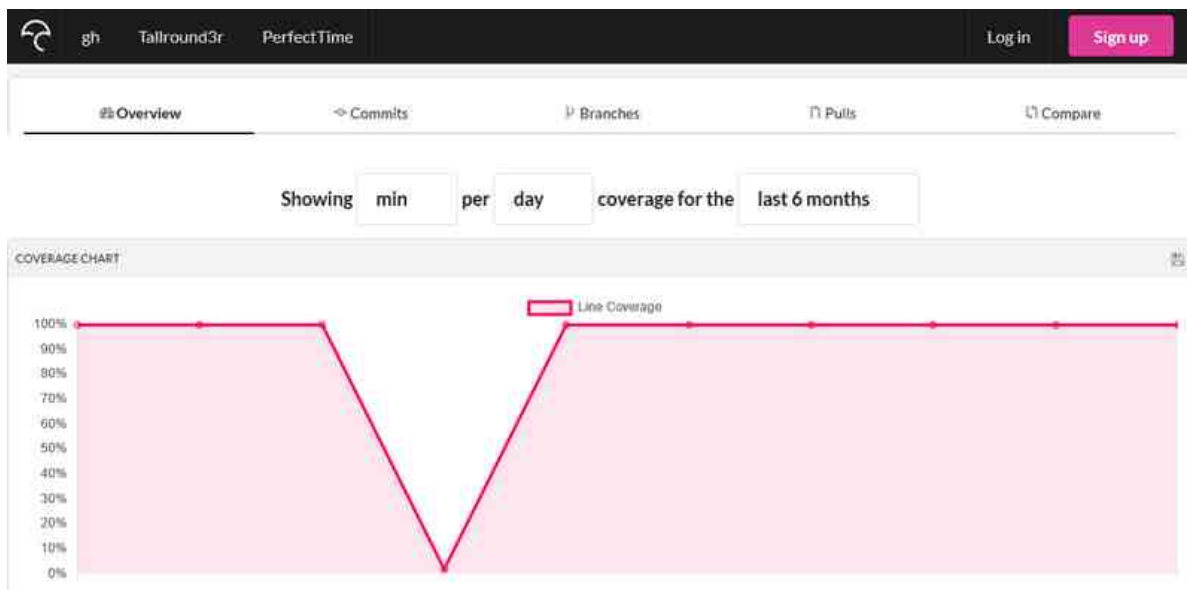


If you want to see more, please check out our current application [1]here. As you might know by now, just click the logo and login or register. (We write that quite often, we think. ;-))

Another thing we often mentioned is [2]Travis CI, but we never explained it in detail. So today, we will do that. We use Travis for our Continuous Integration (CI). So with each merge, it handles the [3]build process along with all the tests, we want to run. Travis also handles our automated deployment process. If we merge a branch into or push a commit onto our master (on GitHub), Travis will not only run the build process, but it will also deploy the successful build on our firebase server. You can see all the necessary steps in this process in our [4]travis.yml file.

We also wanted to mention our final [5]Codecov Score: 100 %. This shows that all our logic files are tested with unit tests. (We do not include any files that only create layout (HTML) or use framework logic in our test coverage

metric. Those files have no areas for reasonable unit testing.) You can look at our unit tests [6]here.



And as a last topic for this week, we wanted to talk about our tech stack. Since we first started with fire-base and React, we decided to use many more technologies. So [7]here is a little overview over all the stuff we are using today.

## PERFECT TIME – TECH STACK (I)

### FRAMEWORKS / DEPENDENCIES

- REACT
- REDUX
- FIREBASE
- FIRESTORE (NOSQL)
- MATERIALUI
- NODE & NPM

### LANGUAGES

- HTML
- CSS
- JAVASCRIPT
- TYPESCRIPT
- YAML

## PERFECT TIME – TECH STACK (II)

### PROCESS & DOCUMENTATION

- CODACY 
- CODECOV 
- TRAVIS CI 
- GITHUB 
- YOUTRACK 
- WORDPRESS 

### TESTING

- JEST 
- FAKER **faker.js** 
- CUCUMBER **cucumber** 
- TESTCAFE 
- TSLINT 

## PERFECT TIME - PROCESS



If you are wondering why we are using all of that tech (we sometimes do ourselves), we recommend you to read through our various blog posts. It is simply too much to explain in one blog entry.

So that's all for this week. We have to get back to coding. And as always stay tuned, since we will be showing you next week, how to set up your own Perfect Time. So see you then.



Your

## Perfect Time Crew

1. <https://perfecttime-planyourtrip.firebaseio.com/>
2. <https://travis-ci.org/>
3. <https://travis-ci.org/Tallround3r/PerfectTime/builds/538080294>
4. <https://github.com/Tallround3r/PerfectTime/blob/master/.travis.yml>
5. <https://codecov.io/gh/Tallround3r/PerfectTime>
6. <https://github.com/Tallround3r/PerfectTime/tree/master/src/unitTests>
7. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/FinalHandin/TechStack.pptx>

---

The grand finale – We made it! (2019-06-16 17:01:18)

[...] CI and AD with TravisCI; running on Firebase (see blog post) [...]

## Creating your own Perfect Time - An installation guide (2019-06-09 21:37)



Hey guys,

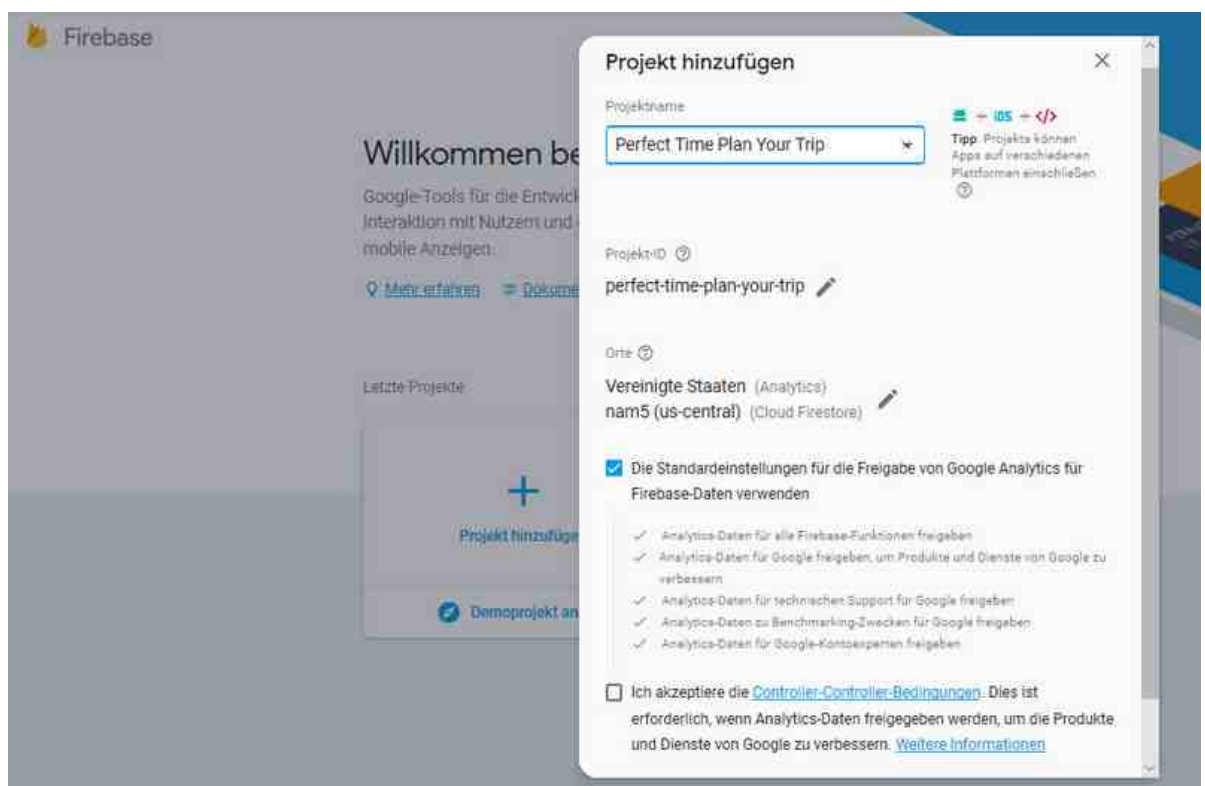
this is our second to last blog post. And it is special. We want to show you, how to set up your own Perfect Time application (on your own Firebase server).

First, there are a few requirements, you need to fulfill before you can start. You need

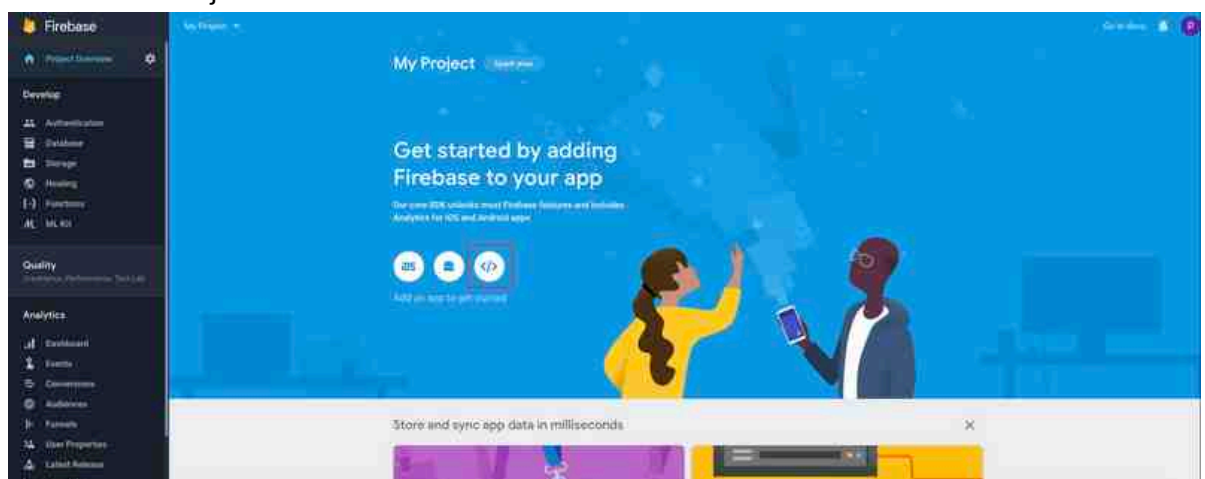
- A computer with an active internet connection
- Node version 11 (NOT 12) and NPM running on your computer ([1]installation guide)

Secondly, you need to set up your own Firebase account. To do that, please follow these steps:

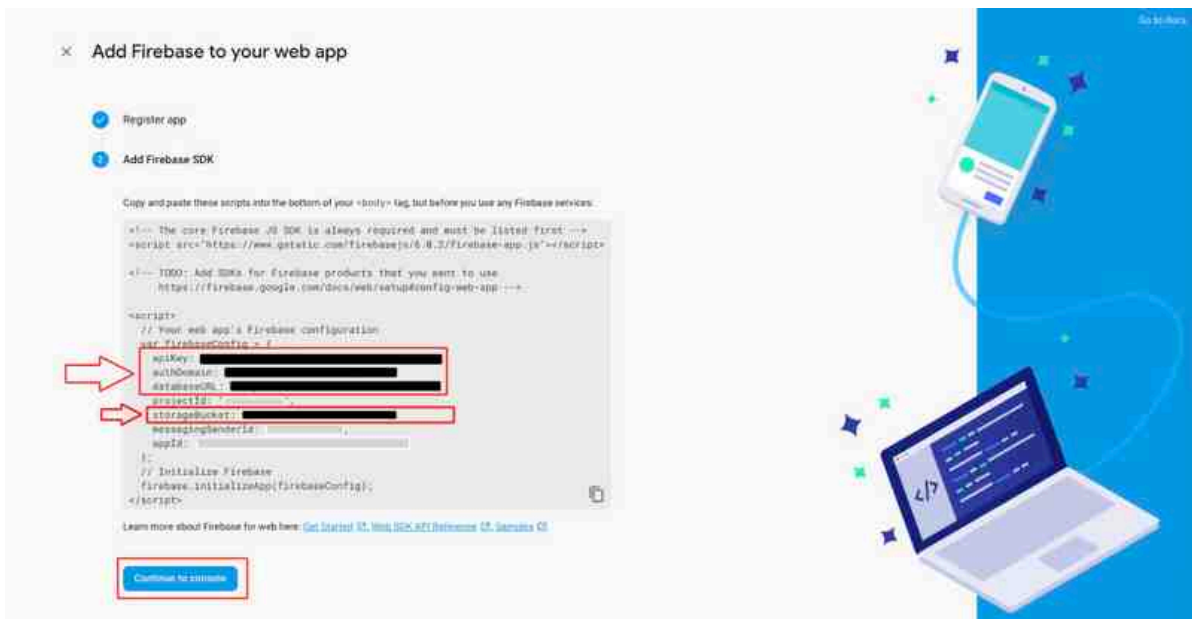
- Visit [2]<https://console.firebase.google.com/> and either login with your Google account or create a new account.
- Add a new Project to your account. (Video Tutorial available at the [3]Firebase website) We recommend to follow this guide [4]HERE (but ignore the rewriting the rules part). You should be fine with the [5]free subscription of Firebase.
  - **IMPORTANT:** After creation firebase will show you your **API-Key, your Auth-Domain, etc.** (see picture below). PLEASE TAKE A **SCREENSHOT OF IT OR COPY IT**. You will need the information later.



Create a new Project

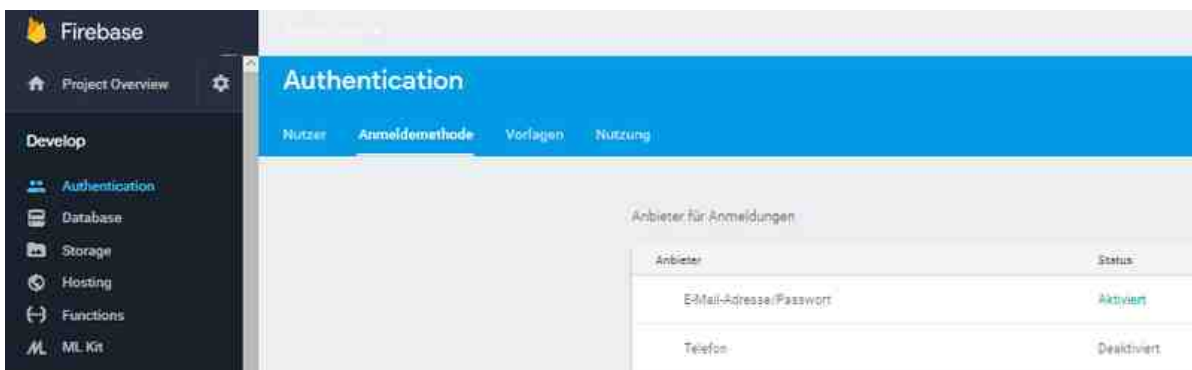


Choose the Web Symbol <> and set up your app.



COPY THIS INFORMATION. YOU WILL NEED IT LATER!

- Please check the following settings in the firebase "console" (browser). First select the tile *Develop* in the left menu.



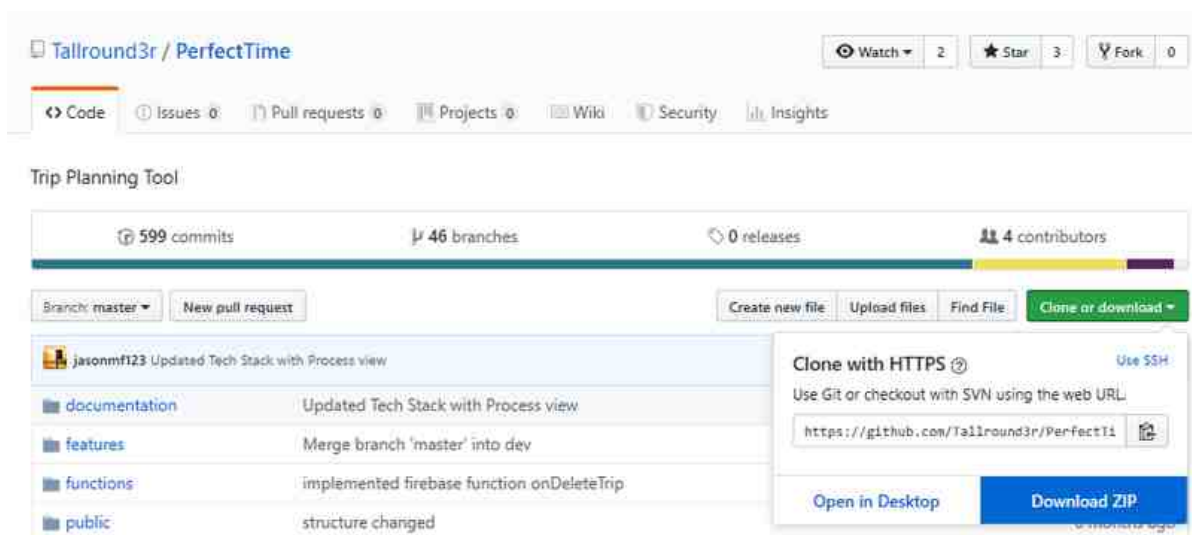
- Select *Authentication*:
  - Choose *Sign In Method* / *Anmeldemethode* and active the email / password option.
  - Choose *Templates* / *Vorlagen*. Please read through the different messages and change the wording to your linking.
- Select *Database* and make sure the *Cloud Firestore* is active (can be accessed)



- Select *Storage* and ensure that the storage is available (can be accessed). The storage will allow the application to save pictures and GIFs. (After you uploaded the first pictures inside Perfect Time, a directory/folder *images/* and two subdirectories *locations/* and *activities/* should be created. Should you get any problems with that functionality, please create those folders manually.)
- Choose *Hosting* (if necessary click Get Started and follow the instructions). Once it is finished, please write down the URL that the application will respond to later on.
- Install the firebase tools on your computer. Open your CLI and run the command "npm i -g firebase-tools". Wait for the process to finish.

Next, you have to download the Perfect Time Source Code to your computer and prepare it for deployment on your Firebase server.

- Visit our [6]GitHub Repository.
- Download our Repository as a ZIP-file to your computer (if you have a git account you can also clone the repo).



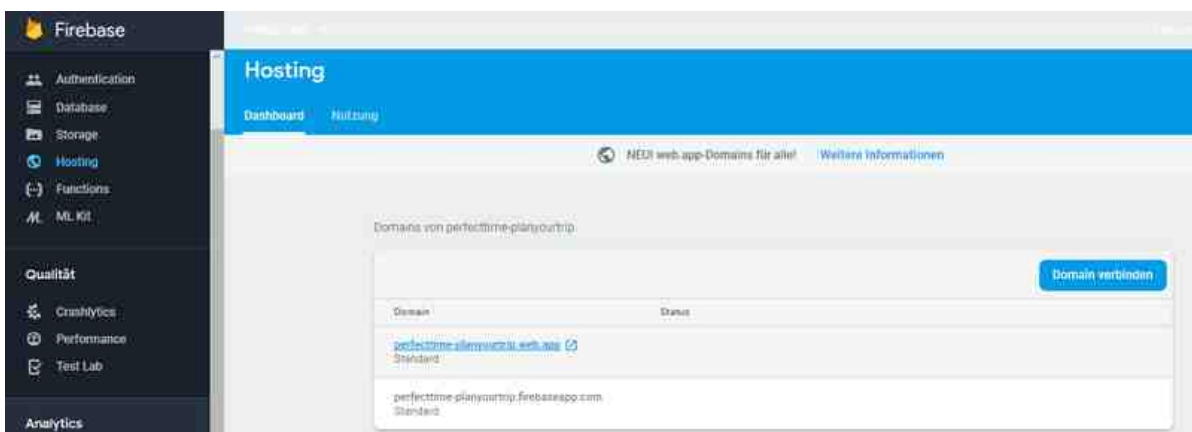
- Unzip the file at a location of your choosing.
- Open the file *example.env*. Fill out the missing values after the '='. You should have copied the information you need to insert after creating the project.
- Rename the file *example.env* to *.env*.
- Open the file *.firebase* and insert your Project-ID from your Firebase console (browser).
- Open your command line in your Perfect Time project folder.
- Insert the command "npm install" and execute it. (It may take some time)
- After finishing change the directory to the functions directory ("cd functions")

- and also run "npm install".
- after that change directory back to project folder ("cd..") and run "npm run build".

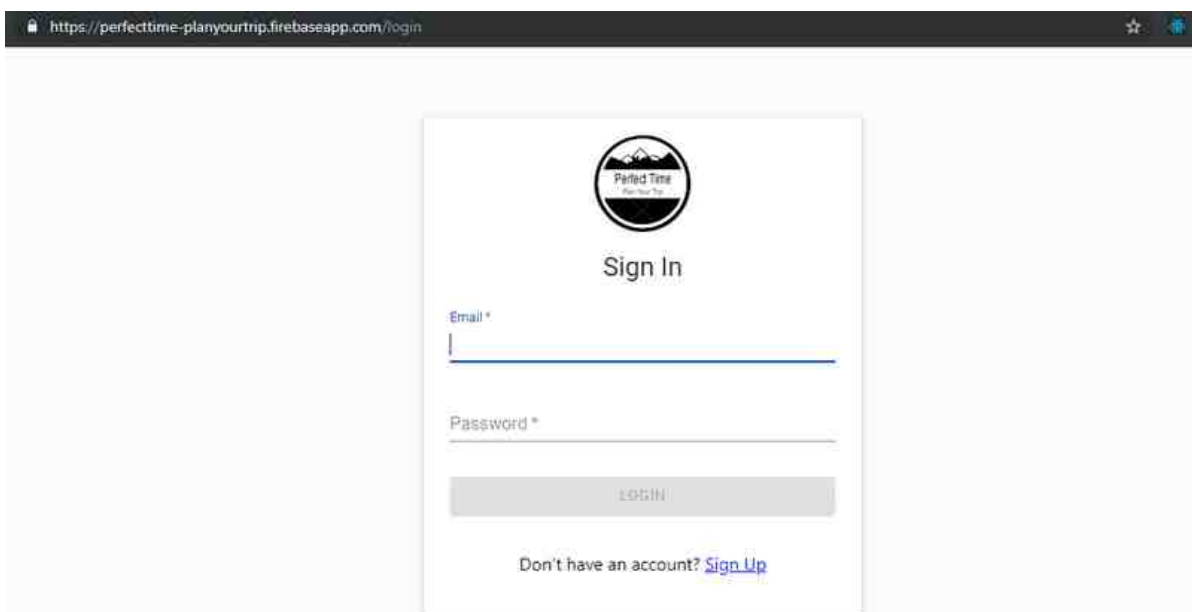
(For additional (optional) Information on React Code please read our [7]ReadMe-File.)

Finally, you have to deploy the application to firebase.

- Open your command line in your Perfect Time project folder.
- Run the command 'firebase deploy' and wait until the process has finished. (You might have to login. Insert "firebase login" and then open the URL. Just follow instructions given to you.
- Open the URL stated in the hosting section of your project in the Firebase console.



- Register your first user and start planning your first trip.



So this it. We completed implementing all the functions, we aimed for during this semester. You now have the choice to either use our "official" [8]application or set up your own application. So tell us what you think in the comments. And as always stay tuned since the grand finale is yet to come (And we shot a cool promo video for it!). ;-)

So see you next week.

Your

Perfect Time Crew

PS: Special thanks to [9]EventLAB for testing this guide.

1. <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>
2. <https://console.firebase.google.com/>
3. <https://firebase.google.com/>
4. <https://www.appypie.com/faqs/how-can-i-get-api-key-auth-domain-database-url-and-storage-bucket-from-my-firebase-account>
5. <https://firebase.google.com/pricing/>
6. <https://github.com/Tallround3r/PerfectTime>
7. <https://github.com/Tallround3r/PerfectTime/blob/master/README.md>
8. <https://perfecttime-planyourtrip.firebaseio.com/>
9. <https://eventlab.jupiterspace.de/>

---

EventLAB Team (2019-06-10 16:16:24)

Hi guys, we followed your instructions step by step and finally... IT WORKED! It's great to see how everyone can set up your project with just a normal computer and 20 minutes of their time. You explained everything really well and in detail, so that it was easy for us to follow your guide. We are looking forward to your final presentation and hope you will rock it the same way as this blog post ;) Best regards, Your friends from EventLAB

perfecttimese (2019-06-10 20:34:10)

Perfect! :-D Thank you for testing our guide. You guys are amazing. Your friends from Perfect Time

The grand finale – We made it! (2019-06-16 17:00:50)

[...] tests (covered by special functional tests)Installation test (see blog post and comment by [...])

**It's looking good guys - our little ad video (2019-06-11 00:13)**



Hey guys,

one of our final tasks was to shoot a little advertisement video. That wasn't our everyday homework and a lot to do for a small team. For this challenge we teamed up with our friends from [1]EventLAB. We used their and our app to plan the whole thing. We shot an entire day and spent another one editing the video. We are quite proud of what we accomplished. So without any further delay: this is our finished advertisement clip:

<https://www.jupiterspace.de/perfectevent/perfecttime.mp4>

What do you think? Do you like it? Do you think that we should get an award for it. Maybe an Oscar or at least a Red-Dot-Award since they seem to give those away for no reason whatsoever. :-P

By the way, our friends from [2]EventLAB had their own perspective and Story line for our trip. This is what they edited:

<https://www.jupiterspace.de/perfectevent/eventlab.mp4>

Amazing, right? So make sure you register to their [3]app. And while your at it, look at our [4]application as well, we implemented some last minute features (such as a search bar).

So we hope you are all exited for our final and last blog post, which we publish next Sunday. So stay tuned, since there is one more thing to come.

Your

Perfect Time Crew

1. <https://eventlab.jupiterspace.de/>

2. <https://eventlab.jupiterspace.de/>

3. <https://eventlab-project.herokuapp.com/>

4. <https://perfecttime-planyourtrip.firebaseio.com/>

---

The grand finale – We made it! (2019-06-16 17:01:28)

[...] blog post [...]

### **The grand finale - We made it! (2019-06-16 17:00)**



Hey guys,

THIS IS IT! Our last blog post. We made. Our [1]application is finished and with this post we hand in our work. We want to thank all the people that followed our progress and that helped us along the way. You guys rock!

THANK YOU!

Your Perfect Time Crew (Lea, Lukas, Tim and Jan)

Below you see links to all our documents and achievements:

GRading CriteriaLinks

### Requirements:

- UC [2]Overview and [3]folder
- SRS [4]SRS file
- Test Cases  
[5]Test Cases folder (incl. .feature and step definitions)
- Test Log (for example screenshot in red/green)
- test coverage for [6]Unit and [7]functional tests  
[8]Test coverage (Unit Tests)
- Test Plan includes 3 different tests  
+ include lifecycle management tools and metrics  
+ change [9]Test plan with function, UI, security, installation and Unit testing;  
includes the used tools and the [10]metrics  
For more information please read [11]this summary
- Functional Test [12]link to example test video
  - + unit test
  - + one other test (like installation, stress test, ...)

[13]Unit tests;

[14]Security tests (covered by special functional tests)

Installation test (see [15]blog post and [16]comment by EventLAB)

### Project Management

- RUP gantt chart (past)  
future long-term plan using FP estimation  
(including hours/team member breakdown) Gantt chart for [17]2018 and [18]2019 (generated with YouTrack, inaccurate,  
as discussed in class)

[19]FP planning and [20]graph (see [21]blog post)

Time tracking on YouTrack for each [22]User, [23]Use Case, [24]Phase and [25]Workflow

- YouTrack / burndown [26]YouTrack project

[27]Total burn-down, Burndown [28]S1 and [29]S2

Close to ideal burn-downs ([30]1, [31]2, [32]3, [33]4, [34]5, [35]6)


- FP calculation

(include estimates and actual time spent comparison) FP calculation (see [36]Use Cases), Complexity [37]calculation

[38]FP planning ([39]original file, [40]explanation)

Time tracking on [41]YouTrack

**Ability to Execute** Build status: 

Application status: 

- Demo

[42]application link

- Code

- installation (manual / or virtual machine) [43]GitHub Repository

[44]Installation guide

## Quality:

### Architecture

SAD document

+ includes pattern highlighted as part of architecture [45]SAD (with Flux architecture and Container pattern explained)

Blog post for [46]Pattern

Blog post for [47]Testing

Configuration Management / Environmental Setup CI and CD with [48]TravisCI; running on [49]Firebase (see [50]blog post)

- Metrics

show how code change improves a metric.

Explain 2 metrics

show code samples when explaining metric

explain, why some parts of code have bad metric.

but no need to change

(can be in SAD document) See [51]blog post (Complexity, Error prone Issues and Performance)

Using [52]Codacy, [53]Codecov and [54]Lighthouse (see [55]blog post)

Integrated in [56]SAD

#### - Risk Management

show living document - correctly filled [57]Risk calculation (explained [58]here)

(for changes, see its history log)

#### - Automated Testing

- functional
- unit
- automatic with deploy

Using [59]TravisCI (see [60].travis.yaml)

explained in [61]blog post

Automated with every pull request (see [62]Screen Shot), Master is deployed with every push / merge

Pattern (place in SAD) explain pattern [63]SAD

explaining [64]blog post

### **Other**

#### - Presentations presentation [65]file

#### - Other Materials:

- Advertisement video
- Hand in document
- Tech Stack

Links:

- see [66]blog post or
- [67]Project Report
- [68]Tech Stack

1. <https://perfecttime-planyourtrip.firebaseio.com/>

2. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/UseCaseDiagram\\_redesigned.png](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/UseCaseDiagram_redesigned.png)

3. <https://github.com/Tallround3r/PerfectTime/tree/master/documentation/useCases>

4. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SRS.md>



5. <https://github.com/Tallround3r/PerfectTime/tree/master/features>

6. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Testing/Unit\\_Testing\\_CLI.PNG](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Testing/Unit_Testing_CLI.PNG)

7. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Testing/Function\\_Testing\\_Coverage.jpg](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Testing/Function_Testing_Coverage.jpg)

8. <https://codecov.io/gh/Tallround3r/PerfectTime>

9. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Testing/TestPlan.md>

10. <https://perfecttime608150251.wordpress.com/2019/05/26/getting-the-numbers-right-working-on-our-metrics/>

11. <https://perfecttime608150251.wordpress.com/2019/06/02/testing-in-a-nutshell/>

12. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/2018\\_12\\_11\\_Cucumber\\_Running.mp4](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/MidtermsHandin/2018_12_11_Cucumber_Running.mp4)

13. <https://github.com/Tallround3r/PerfectTime/tree/master/src/unitTests>

14. <https://github.com/Tallround3r/PerfectTime/tree/master/src/securityTests>

15. <https://perfecttime608150251.wordpress.com/2019/06/09/creating-your-own-perfect-time-an-installation-guide/>

16. <https://perfecttime608150251.wordpress.com/2019/06/09/creating-your-own-perfect-time-an-installation-guide/#comment-80>

17. <https://perfecttime.myjetbrains.com/youtrack/reports/gantt/134-1>

18. <https://perfecttime.myjetbrains.com/youtrack/reports/gantt/134-2?view=actual>

19. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/TimeEstimationUC.xlsx>

20. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/FPgraph.png>

21. <https://perfecttime608150251.wordpress.com/2019/04/21/yes-we-can-finish-our-defined-scope-within-this-semester/>

22. <https://perfecttime.myjetbrains.com/youtrack/reports/time/145-4>

23. <https://perfecttime.myjetbrains.com/youtrack/reports/time/145-1>

24. <https://perfecttime.myjetbrains.com/youtrack/reports/time/145-5>

25. <https://perfecttime.myjetbrains.com/youtrack/reports/time/145-3>

26. <https://perfecttime.myjetbrains.com/youtrack/>

27. <https://perfecttime.myjetbrains.com/youtrack/reports/burndown/119-6>

28. <https://perfecttime.myjetbrains.com/youtrack/reports/burndown/119-2>

29. <https://perfecttime.myjetbrains.com/youtrack/reports/burndown/119-5>

30. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-25?chart>

31. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-14?chart>

32. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-24?chart>

33. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-23?chart>

34. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-21?chart>

35. <https://perfecttime.myjetbrains.com/youtrack/agiles/98-2/99-20?chart>

36. <https://github.com/Tallround3r/PerfectTime/tree/master/documentation/useCases>

37. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/PT\\_function\\_points\\_complexity\\_calc.ods](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/PT_function_points_complexity_calc.ods)

38. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/useCases/TimeEstimationUC.xlsx>

39. [https://drive.google.com/open?id=1ZQ\\_V2uwXbj0vQeNRhskU6kZrGtW4QiVQ](https://drive.google.com/open?id=1ZQ_V2uwXbj0vQeNRhskU6kZrGtW4QiVQ)

40. <https://perfecttime608150251.wordpress.com/2019/04/21/yes-we-can-finish-our-defined-scope-within-this-semester/>

41. <https://perfecttime.myjetbrains.com/youtrack/reports/time/145-1>

42. <https://perfecttime-planyourtrip.firebaseio.com/>

43. <https://github.com/Tallround3r/PerfectTime>

44. <https://perfecttime608150251.wordpress.com/2019/06/09/creating-your-own-perfect-time-an-installation-guide/>

45. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SAD.md>

46. <https://perfecttime608150251.wordpress.com/2019/05/12/i-dont-trust-words-i-even-question-actions-but-i-never-doubt-patterns/>

47. <https://perfecttime608150251.wordpress.com/2019/06/02/testing-in-a-nutshell/>

48. <https://travis-ci.org/Tallround3r/PerfectTime>
  49. <https://console.firebase.google.com/>
  50. <https://perfecttime608150251.wordpress.com/2019/06/02/getting-ready-for-the-grand-finale/>
  51. <https://perfecttime608150251.wordpress.com/2019/05/26/getting-the-numbers-right-working-on-our-metrics/>
  52. <https://app.codacy.com/project/Tallround3r/PerfectTime/dashboard>
  53. <https://codecov.io/gh/Tallround3r/PerfectTime>
  54. <https://developers.google.com/web/tools/lighthouse/>
  55. <https://perfecttime608150251.wordpress.com/2019/06/02/testing-in-a-nutshell/>
  56. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SAD.md>
  57. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/RiskManagement.xlsx>
  58. <https://perfecttime608150251.wordpress.com/2019/04/14/a-risky-business/>
  59. <https://travis-ci.org/Tallround3r/PerfectTime>
  60. <https://github.com/Tallround3r/PerfectTime/blob/master/.travis.yml>
  61. <https://perfecttime608150251.wordpress.com/2019/06/02/testing-in-a-nutshell/>
  62. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/Metrics/automatedCodacyCheck.PNG>
  63. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/SAD.md>
  64. <https://perfecttime608150251.wordpress.com/2019/05/12/i-dont-trust-words-i-even-question-actions-but-i-never-doubt-patterns/>
  65. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/FinalHandin/FinalPresentation.pptx>
  66. <https://perfecttime608150251.wordpress.com/2019/06/11/its-looking-good-guys-our-little-ad-video/>
  67. [https://github.com/Tallround3r/PerfectTime/blob/master/documentation/FinalHandin/Final\\_Report\\_PerfectTime.docx](https://github.com/Tallround3r/PerfectTime/blob/master/documentation/FinalHandin/Final_Report_PerfectTime.docx)
  68. <https://github.com/Tallround3r/PerfectTime/blob/master/documentation/FinalHandin/TechStack.pptx>
-



BlogBook v1.1,  
L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> & GNU/Linux.  
<https://www.blogbooker.com>

Edited: June 17, 2019

