

Week 1: The Vision for Great Success

👤 Corvin Klein ⏰ October 5, 2021December 30, 2021 💼 1 Minute

Welcome to our wonderful first blog entry. In this blog, you should be able to follow us on our journey to building a working app. But what is the “nEXam” app supposed to do? “nEXam” will be an app which helps you plan out when you should start learning before an exam and how much time you should spend learning.

How does it work?

You need to enter your exam information, including date, subject, and topic. Based on this information, the app will create an agenda when you should study which topic. While studying, you can run the in-app timer, so the app will change the background color of the topic to visualize what has already been learned. Also, you should be able to export the iCal file to import it in your calendar of choice.

What will we use?

The app should be available on iOS and Android. To code, we will use Kotlin, because it supports an easy way to create smartphone applications with common business logic for both iOS and Android apps. It can also be used for front- and backend. To store the data, we planned to use SQLite, but decided to stick with an XML file due to the small amount of data.

Advertisements



REPORT THIS ADPRIVACY



Published by Corvin Klein

[View all posts by Corvin Klein](#)

3 thoughts on “Week 1: The Vision for Great Success”

official4kills says:

October 11, 2021 at 9:49 am | Edit

Concise blog post and good structure.

Propositions for improvement:

Content:

1. What are “chapters”?
2. How can Kotlin create both Android and iOS apps? Is it possible to compile from Kotlin to Objective-C / Swift required by iOS?
3. Why “hopefully” (!) use SQLight?
4. Why should one use your app instead of the 100 others that exist? What’s your USP?
5. Are you sure you want to use Kotlin as it might not be possible to auto-create UML diagrams from it? I am not sure about this, though.
6. Perhaps, expand your idea a bit, because the idea might be lacking a little complexity for a team of least 3 devs for a year. Maybe teaser a few more features that could be added in the future.

Spelling, punctuation

1. Put title in all title case.
2. “blog entry” not “blog-entry”
3. Please review capitalization (app not App, journey not Journey)
4. Check punctuation (commas):
5. in-app
6. which

↳ Reply

st3fan1e says:

October 11, 2021 at 6:05 pm | Edit

Hi dear nEXam-team,

we really like the idea of the app. We would however like it better if the plan generated by the app could be exported into iCal format and synchronized into an existing calander system. It would also be nice to get an insigh into your know-how and learn about the team members. How about adding a page introducing yourselves?

Can you please explain the changing of background color further. I did not quit get this concept.

Best wishes,

your LectureFeed-team 

↳ Reply

uhrsch says:

December 27, 2021 at 10:14 am | Edit

Hello LectureFeed-team

we decided to not use an DBMS and store our data just in an *.xml file. We also read directly from it so you can easily took your data to another device.

I think you are refering to the changing background color while the timer is running? We expected to show a complete finished exam in green and a new exam in red. While the timer is

counting your spent time on that topic the color should change so you can see faster how much time you will need to spend on that topic.

Best regards,
your nEXam team

↳ Reply



Week 2: First Step – Organization

👤 uhrscl 🕒 October 12, 2021December 30, 2021 💼 1 Minute

We would like to start by thanking you for your interest and comments on our first blog entry.

To turn our vision into a solid project, we had to set up some platforms to manage and structure our work. So we started with setting up the groups for the blog and working on our YouTrack project. Again we decided on the groups and responsibilities. Then we created a first sprint and discussed the needed actions. All of them were created and assigned by our project manager. We also set up our version control system in the form of a GitHub repository.

To avoid the process of starting too early and running into the wrong direction, we will use the first week to collect more information. The goal will be to update the requirements and used technologies. But we also need to earn some knowledge. In our current situation we would have to research every step, but that would slow us down, so we decided to research first and then start more efficiently.

Additionally, everyone must set up their workstation for their tasks.

Of course everybody will be a Developer and will need an IDE, but we also got the following roles assigned:

- Project Manager – Corvin
- Software Architect – Corvin
- Project Admin – Franziska, Corvin
- Business Process and System Analyst – Simon
- Test Manager – Simon
- Blog Manager- Mareike

Certainly, we will not work only in our roles. We will try to help each other to achieve a balanced level of work and shared knowledge of the systems.

Edit: Since Mareike has unfortunately left us in the meantime, we will take over her responsibilities together.

Advertisements



Like an undo
button for your
WordPress site

[Back up your site](#)

[REPORT THIS AD](#)[PRIVACY](#)



Published by [uhrschen](#)

[View all posts by uhrschen](#)

4 thoughts on “Week 2: First Step – Organization”

[DueTo](#) says:

October 17, 2021 at 9:39 am | Edit

Hi nEXam Team,

we enjoyed reading your post. It shows what kind of roles you have and how you divided your responsibilities.

It is also nice that you started early with your sprints. This way you can get quickly accustomed to it and avoid mistakes later on.

What we think is missing is your tech stack. Maybe you could create a table showing what technologies you want to use and why.

We are looking forward to seeing your progress.

Regards,

Your DueTo Team

↳ Reply

[uhrschen](#) says:

October 25, 2021 at 1:06 pm | Edit

Hello DueTo Team, thank you for your review.

We mentioned some thoughts about our tech stack in our first post, we would like to use the opportunity to refer you to our new post “Getting Concrete”

(nexam955203221.wordpress.com/2021/10/19/getting-concrete/), there we're explaining our tech stack and why we decided to use those programms.

Regards,
Your nEXam Team

↳ Reply

crixos86 says:

October 18, 2021 at 7:34 pm | Edit

Hi nEXam Team,

we really like your idea because it could be really beneficial for a student's life.

We are looking forward to see some of your first results 😊

It was quite a good idea to took a long time on planning to avoid early mistakes.

Kind regards

Your Papatohu Team

↳ Reply

Pingback: [Midterm – nEXam](#) | [Edit](#)



Week 3: Getting specific

👤 uhrscl 📅 October 19, 2021December 30, 2021 ⏳ 2 Minutes

Last Tuesday we finished our first Sprint. We are pleased to have successfully completed all tasks on time and feel good about our agenda.

After our research, we are even more confident that our tech stack from our first post is a good fit for us and our project. (You can find the post [here](https://nexam955203221.wordpress.com/2021/10/05/blogpost-vision/) (<https://nexam955203221.wordpress.com/2021/10/05/blogpost-vision/>))

We all agreed to use VS Code as our IDE because it gives us more flexibility to use other programming languages (if needed) in the same application. Yes, we have to adjust and customize the VS Code a bit, but we think that's fine. Some of us have also had great experiences with the 'Live Share' extension, which allows us to work on code together and simultaneously. This is useful when more than one person is working on a topic or someone needs help.

As we mentioned in our first blog, we will use Kotlin and SQLite.



The main argument why we will use Kotlin (even if nobody has any experience with it) is usability. With Kotlin, we are able to create the front- and backend, also Kotlin provides us with the possibility to develop the app for Android and iOS without a huge additional workload.

Kotlin logo

We originally planned to use SQLite as the database. Our goal would be to not need a large database and store all the data we need on the client. There will be no need to 'work' with the client's data, and everything should be able to run only on the user's device.

Edit: However, since we used an XML file to generate test data for our frontend, we have revised this decision and now store the data as an XML file instead.



SQLite logo

Edit: We used Android Studio to create the views and found that it meets our requirements better



Android Studio logo

than VS Code, so we will move our project there completely. Android Studio supports Kotlin, offers Android Native as a framework by default, and even provides accessibility tips. Another plus is that it automatically offers a dark and light theme.

Additionally, we have started to create some use cases for our application. We will continue to work on them and present them in a few weeks (hopefully next week). We also created some documentation on our GitHub repository.

Published by uhrscl



[View all posts by uhrscl](#)

6 thoughts on “Week 3: Getting specific”

Planit Se says:

October 25, 2021 at 2:39 pm | Edit

Hey neXam Team,

seems like you make lots of decisions last week. We like that you think about what influence has you decision on your project. Your ideas sounds like you will have a good teamwork.
Unfortunately we d'ont understand if your intention to use SQLite is to increase the data security of the users or just to prevent to use a big database.
We are looking forward to see your use case diagramm and your next steps.

Best,
Your PlanIt-Team

↳ Reply

Corvin Klein says:

October 26, 2021 at 8:35 am | Edit

Thanks for your feedback,

regarding your question, we chose SQL-Lite because we only will store data locally. Also it isn't a lot of data so we want to prevent unnecessary overhead. The increased security is a nice

benefit.

kind regards
nEXam Team

↳ Reply

Team Papatohu says:

October 25, 2021 at 3:37 pm | Edit

Hi nEXam Team,

we like your idea of using VSCode because it provides you more flexibility than other IDE's.
We think its kind of brave from you to choose Kotlin despite the fact no one used it before, but we're sure that you'll manage it well.

The fact that you stayed behind your decission regarding your tech stack seems that you're well prepared.

kind regards

Team Papatohu

↳ Reply

uhrschr says:

December 27, 2021 at 10:18 am | Edit

Hello Team Papatohu

reflecting the semester it was a good choice to use Kotlin, but we needed to change our IDE (at least partly) to Android Studio. We got there a benefit from Kotlin by a quite easy handling of the frontend which is realized with *.xml files.

Best regards,

you nEXam Team

↳ Reply

Pingback: [Week 7: database schema – nEXam](#) | [Edit](#)

Pingback: [Midterm – nEXam](#) | [Edit](#)



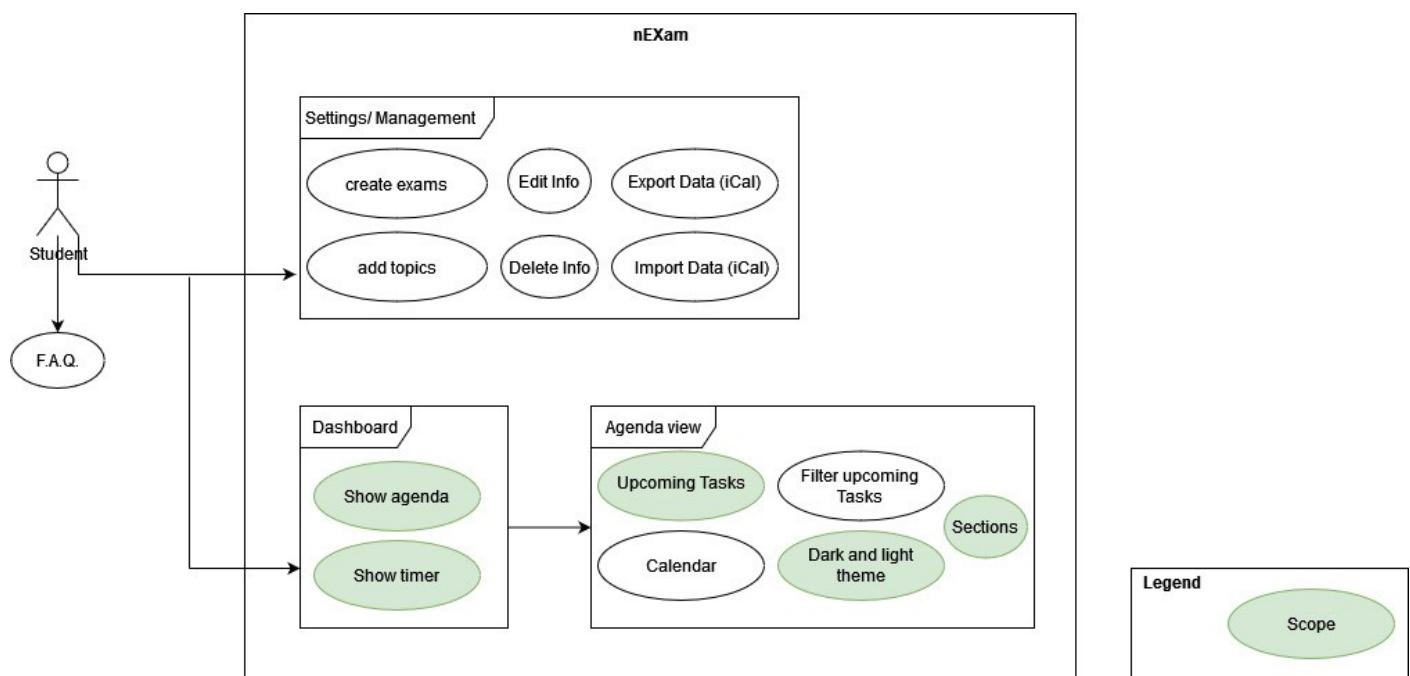
Week 4: Check out our GitHub repo

👤 uhrscht ⌚ October 26, 2021December 30, 2021 Ξ 1 Minute

On our last post, we missed sharing the first version of our SRS and our use case diagram. So we would like to share them now with you. Also, on that point we would like to invite you to take a look at our [GitHub repository](https://github.com/Calco2001/nEXam) (<https://github.com/Calco2001/nEXam>), there you will find of course our code but also the latest versions of our documentation.

Here you will be redirected to our [SRS](https://github.com/Calco2001/nEXam/blob/main/docs/SRS.md) (<https://github.com/Calco2001/nEXam/blob/main/docs/SRS.md>).

Below you will find our use case diagram. The ovals show the different use cases, the green marked are planned to be realized until the end of the year.

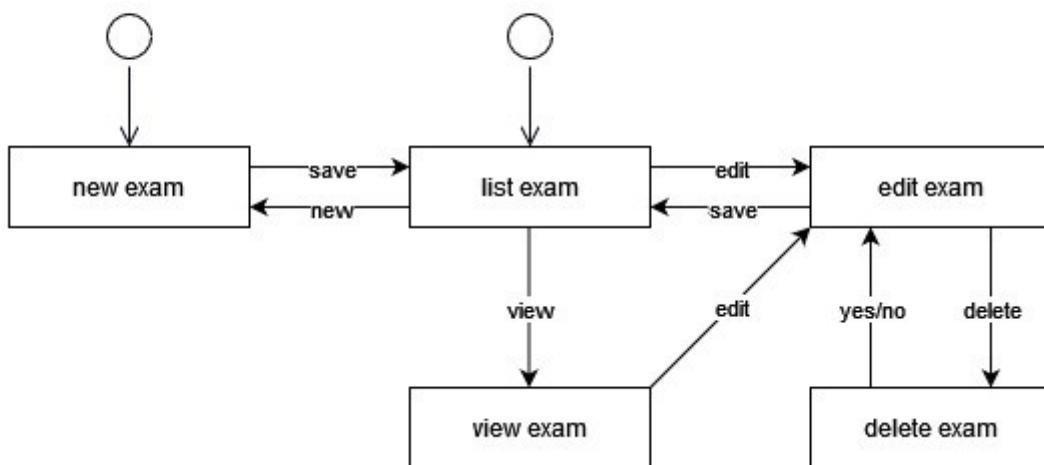


In our planning are currently few use cases shown, but those should be a solid ground, so we are able to expand our app after we finished those cases, if there are ideas and time. The user itself will only be able to interact with the two frames 'Dashboard' and 'Setting/Management' but of course there are some functionalities needed behind the 'Dashboard'.

If you are interested in how we adjust the diagram or what changes, you can also find it on our [GitHub page](https://github.com/Calco2001/nEXam) (<https://github.com/Calco2001/nEXam>) or directly (<https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/use%20case%20diagram.jpg>).

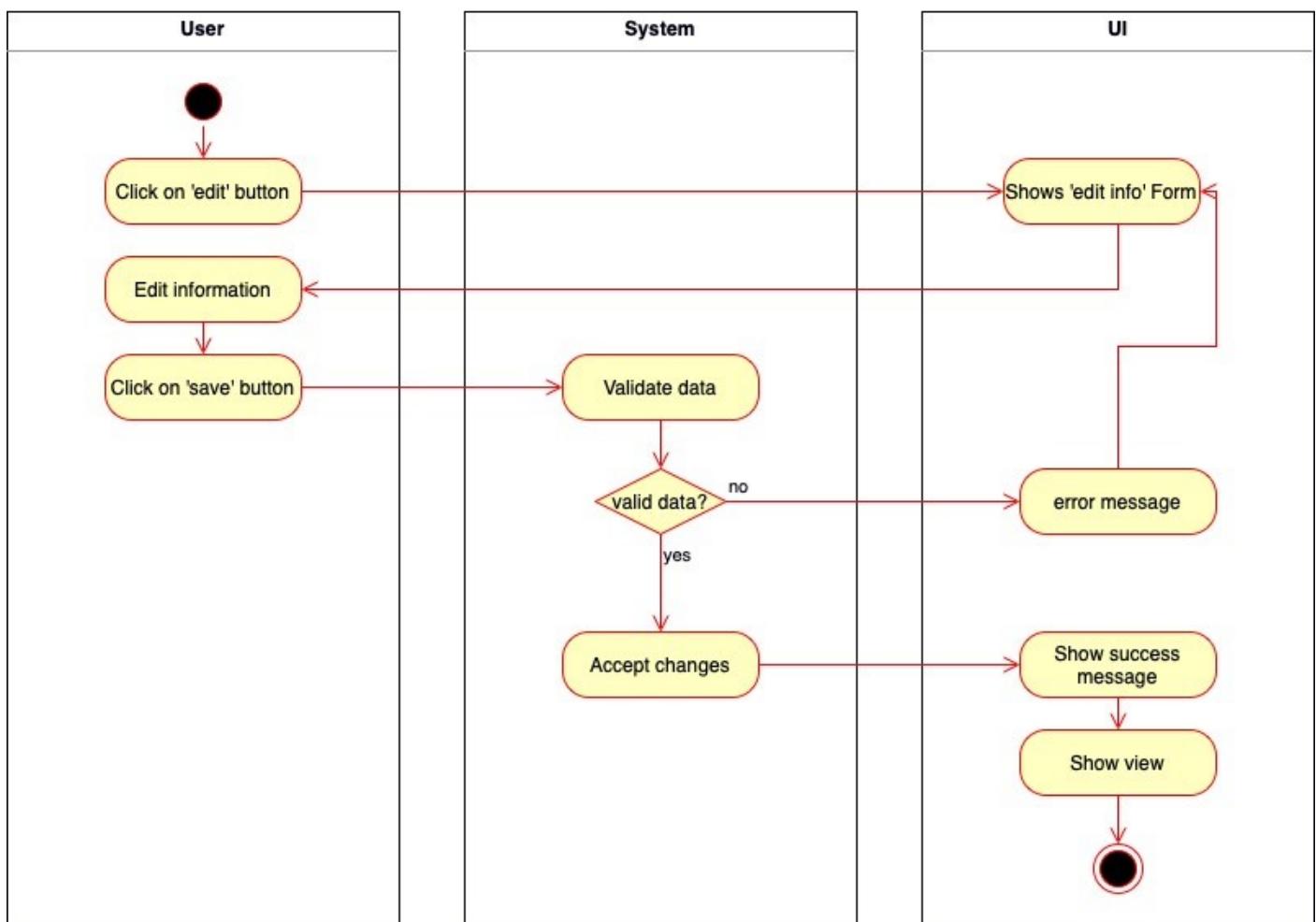
Next we would like to show you our [basic flow diagram](https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/basic%20flow.jpg) (<https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/basic%20flow.jpg>) which shows the simplest version of our workflow.

flow of events - overview

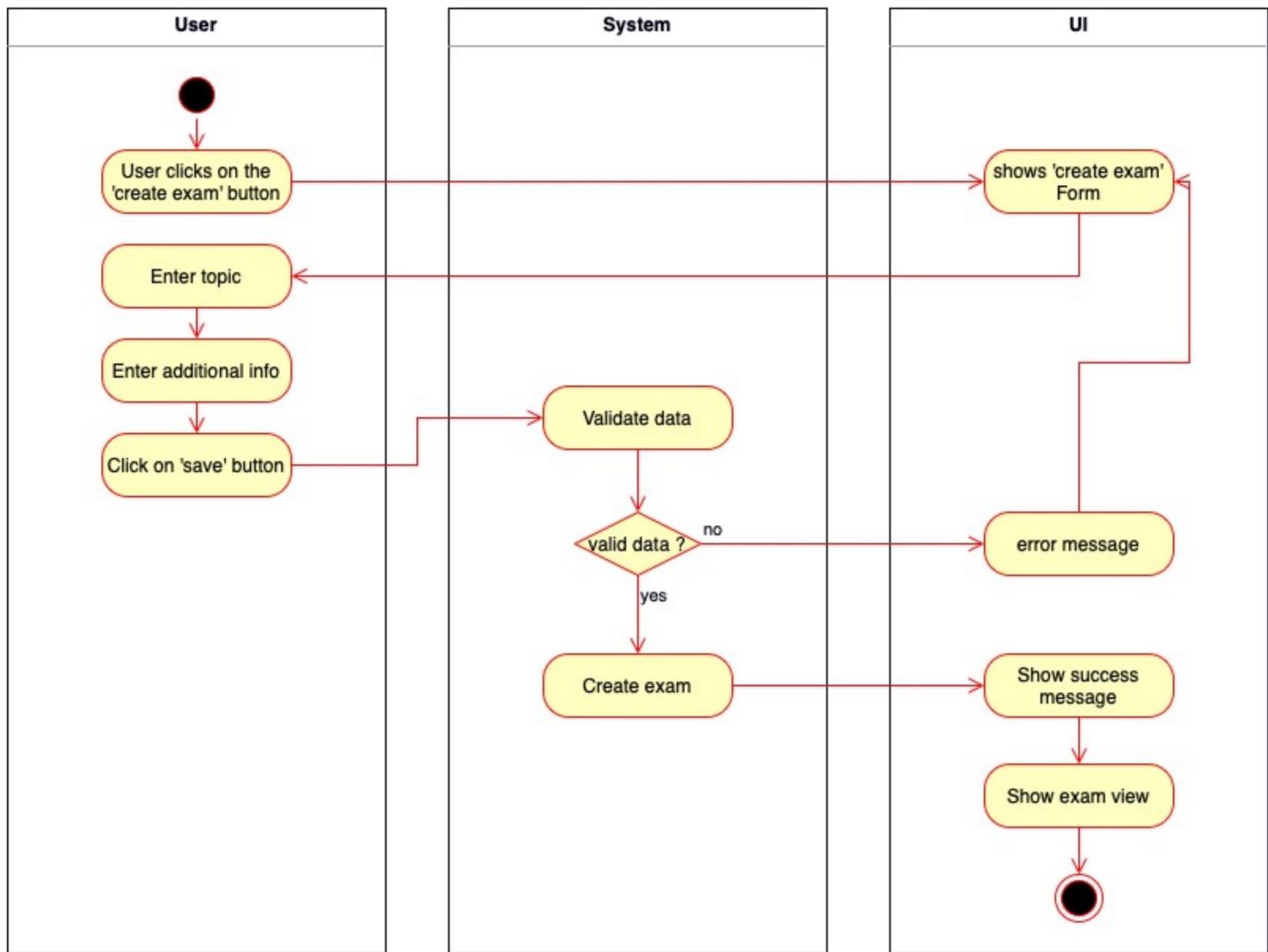


More complex and detailed are our activity diagrams ([edit info](https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/activity_diagram_EditInfo.jpg) (https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/activity_diagram_EditInfo.jpg) and [create exam](https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/activity_diagram_CreateExam.jpg) (https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/activity_diagram_CreateExam.jpg)), here for two of our first five use cases.

Edit Info



Create Exam



Or to get all of them at once you could visit our [use-case specification](https://github.com/Calco2001/nEXam/tree/main/docs/Use_Case_Specification) (https://github.com/Calco2001/nEXam/tree/main/docs/Use_Case_Specification) where all those diagrams are included and explained.

We are looking forward to your opinion and reviews.

Published by uhrscl



[View all posts by uhrscl](#)

6 thoughts on “Week 4: Check out our GitHub repo”

Lukas H. [B2] says:

November 1, 2021 at 9:27 pm | Edit

Hey nEXam Team,

we really enjoyed reading through your blogpost!

Although you mentioned that use-case diagram shows rather few user cases, we have to say that we really don't mind, as we think that even very few use-cases can be quite some work if you really pay attention to details like encryption of user data.

One great feature would be to make .ics files publicly available via a long, hard to brute force url! This would enable user to use the web-calendar feature of most common calendars like Outlook. That way, calendars can update in real-time across multiple devices. But that a feature for the next semester!

Keep up the good work,

Lukas from DHBW-Experts [B2]

↳ Reply

uhrsch says:

December 27, 2021 at 10:22 am | Edit

Hello Lukas,

thank you for your great idea! We will keep that in mind.

Currently we dont just want to show the exam in the calender, you are also able to run a timer to see how much time you spent for a topic. We currently havent researched about the possibility to do that with other applications.

I think for sure we will come to that question next semester!

Best regards,

your nEXam Team

↳ Reply

undecided1550 says:

November 1, 2021 at 10:40 pm | Edit

Dear nExam Team,

your activity diagrams are good, nothing to add there. I believe your use case diagram/basic flow diagram would be more understandable if you would add some mock ups to visualize those functionalities you define in your diagrams. We are looking forward to see your mock ups and your general design to deliver a great and intuitive user experience – that's one of the most important aspects in creating an app!

Your UNDECIDED Team

↳ Reply

Corvin Klein says:

November 2, 2021 at 9:36 am | Edit

Dear UNDECIDED Team,

We have added some Wireframes in our use case specification. If you want you can have a look at them.

Kind regards

your nEXam Team

↳ Reply

Pingback: [Week 5: Its not a bug its a .feature – nEXam](#) | [Edit](#)

Pingback: [Midterm – nEXam](#) | [Edit](#)

W

Week 5: It's not a bug, it's a .feature

👤 uhrscht 🕒 November 2, 2021December 30, 2021 ⏳ 1 Minute

This week, we created Gherkin files, which describe our first two use cases (mentioned in the [post last week](#) (<https://nexam955203221.wordpress.com/2021/10/26/week-4-check-out-our-githubrepo/>)). Those files are based on the cucumber language, which we also added to our VS Code extensions.

In those files, you can see a specific how a use case could be performed. We added them to our [repository](#), so (<https://github.com/Calco2001/nEXam>) you can take a look. Below you will see the current versions, but for the updated version please visit us on GitHub.

```

1  Feature: Use Case 1 Create Exam
2    As a USER
3      I want to create a new exam with a topic and further information.
4      Therefore I will be asked for the topic and additional information (subject, content and date).
5
6      Background:
7          Given The application has the needed local permissions
8
9      @postsession-feature:
10     Scenario Outline: Create new Exam
11         When The user clicks on the Create Exam button
12         And The user types the topic <topic>
13         And The user types the subject <subject>
14         And The user types the content <content>
15         And The user types the date <date>
16         And The user clicks on the Save button
17         And All the data is valid
18         Then The exam is created
19         And The success message with <topic> is shown
20
21     Examples: Exam
22     | topic           | subject           | content           | date           |
23     | Formale Sprache | Regulaere Ausdruecke | Regex            | 2021-12-17    |
24     | Statistik       | Deskriptive Statistik | Histogramme     | 2021-12-23    |
25
26     @postsession-feature:
27     Scenario: Leaving the Activity New Exam without creating an exam
28         When The user presses the Back button
29         Then No exam is saved
30         And The user returns to the previous screen

```

Use case 1: Create new exam

```

1 Feature: Use Case 2 edit an existing Exam
2   As a USER
3     I want to edit the information of an existing exam.
4     Therefore I will get shown the topic and additional information (subject, content and date) of the selected exam.
5
6   Background:
7     Given The application has the needed local permissions
8     And At least one exam is existing
9     And The user wants to edit all information
10
11 @postsession-feature:
12 Scenario Outline: Edit all information on an existing exam
13   When The user clicks on the existing exam in the Dashboard view
14     And The user edit the topic <topic>
15     And The user edit the subject <subject>
16     And The user edit the content <content>
17     And The user edit the date <date>
18     And The user clicks on the Save button
19     And All the data is valid
20     Then The exam is updated
21     And The success message with <topic> is shown
22
23   Examples: Exam
24   | topic          | subject           | content      | date       |
25   | Formale Sprache | Regulaere Ausdruecke | Regex        | 2021-12-17 |
26   | Statistik       | Deskriptive Statistik | Histogramme | 2021-12-23 |
27
28   Background:
29     Given The application has the needed local permissions
30     And At least one exam is existing
31     And The user wants to edit the date
32
33 @postsession-feature:
34 Scenario Outline: Edit only one information of an existing exam (example <date>)
35   When The user clicks on the existing exam in the Dashboard view
36     And The user edit the date <date>
37     And The user clicks on the Save button
38     And All the data is valid
39     Then The exam is updated
40     And The success message with <topic> is shown
41
42   Examples: Exam
43   | topic | subject | content | date      |
44   |      |         |         | 2021-12-17 |
45   |      |         |         | 2021-12-23 |
46
47 @postsession-feature:
48 Scenario: Leaving the Activity New Exam without creating an exam
49   When The user presses the Back button
50   Then No exam is updated
51   And The user returns to the previous screen

```

Use case 2: Edit existing exam



Published by uhrscl

[View all posts by uhrscl](#)

4 thoughts on “Week 5: It’s not a bug, it’s a .feature”

PiXS says:

November 5, 2021 at 12:54 pm | Edit

Howdy nEXam team,

we read your blog post and want to give you some feedback:

We really like your title because it is a creative play on words.

Your blog post and feature files are very concise but we are not sure why you split the second use case into two separate screenshots (You might change that).

Btw: you might move the .feature files to a dedicated folder in your repository, in order to structure your repository 😊

Cheerio, the PiXS team

↳ Reply

uhrsch says:

December 27, 2021 at 10:44 am | Edit

Hello PiXS team,

you might have seen that we moved our *.feature files and also structured our repository.

Maybe you would like to take a look again 😊

Best regards,

your nEXam Team

↳ Reply

P3lina says:

November 16, 2021 at 10:54 am | Edit

Pai rā Team nEXam,

I think your features are pretty detailed which hopefully helps you later in development.

As PiXS already said, moving the .feature files into a seperate folder defenitely helps you organizing your repository, but I've senn you've already did that, so: Well done!

Apart from that I think that there is nothing more to criticize and I wish you good luck with the further realization of your project.

Your Team Papatohu

↳ Reply

Pingback: [Midterm – nEXam](#) | [Edit](#)

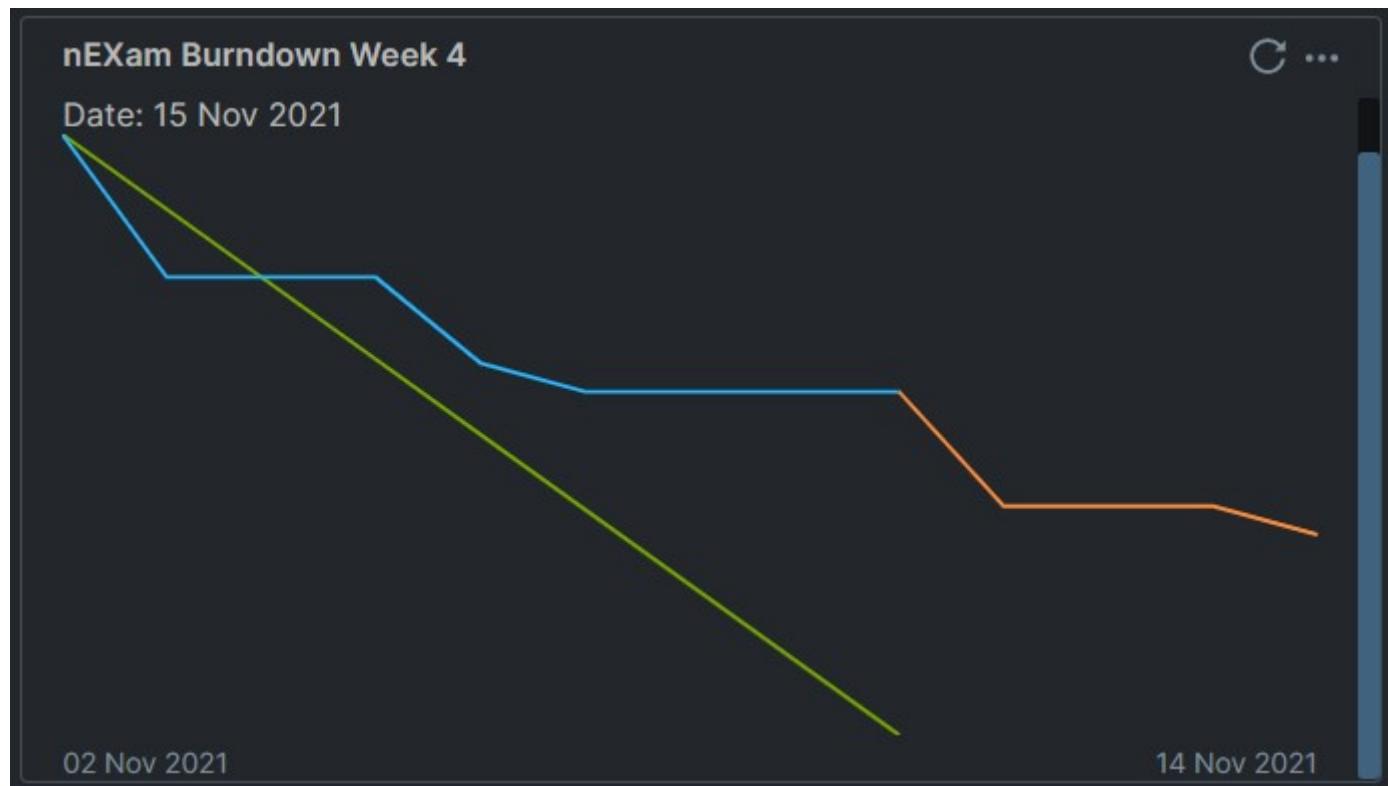


Week 6: our dashboard

👤 uhrscht 🕰 November 15, 2021December 30, 2021 🗓 1 Minute

Now we are already working 6 weeks on our project and so it's time to take a first look of our own evaluations.

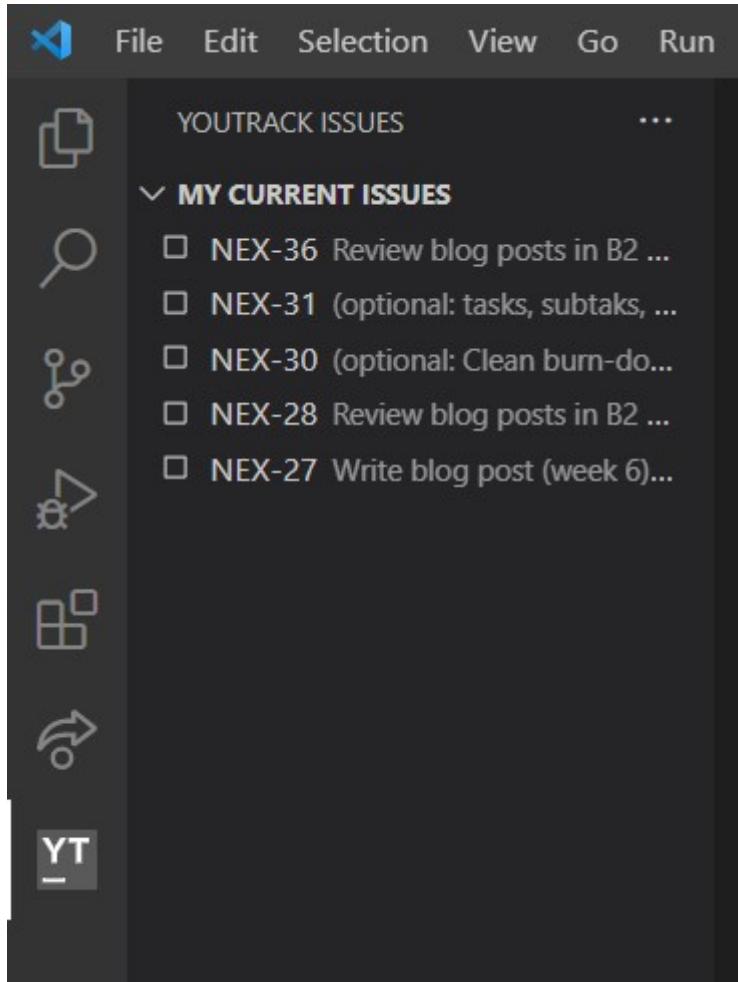
On our dashboard, we created a few burndown charts and also a cumulative flow.



We are still confronted with a planning and organization problems. We are working on it and the burnouts looks better and better, but we need to adjust a little more. Also, we are having a little trouble with our work on YouTrack, as you see we are setting up something wrong and as a result we are getting the orange overdue.

If you like to see our other burndown charts or any other of our evaluations, you can see them on our [YouTrack dashboard](https://dhwkarlsruhe.myjetbrains.com/youtrack/dashboard?id=fc14affe-ae44-4123-8d77-b5828e4aa5eb) (<https://dhwkarlsruhe.myjetbrains.com/youtrack/dashboard?id=fc14affe-ae44-4123-8d77-b5828e4aa5eb>).

Additionally, to hopefully increase our structure with YouTrack we added a YouTrack extension to our IDE.



After a short configuration, we are able to see there our assigned issues, and we are able to add times directly.

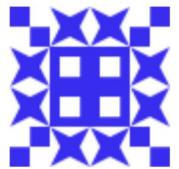
A screenshot of a YouTrack Kanban board. The board has four columns at the top: "Open" (0m), "In Progress" (2h30m), "To Verify" (0m), and "Done" (1d2h). Below these columns is a header row with status filters: "Elaboration 1", "9 — 16 Nov", "1 day left", "Open", "In Progress", "To Verify", "Done", and a "New swimlane" button. The main area contains a list of cards under the heading "Uncategorized Cards". There are two cards in the "In Progress" column: "NEX-28 Review blog posts in B2 and B4" and "NEX-27 Write blog post (week 6)/ evaluation criteria". There are seven cards in the "To Verify" column: "NEX-26 Integrate youtrack in IDE", "NEX-32 Integrate youtrack with github", "NEX-34 Set up environment", "NEX-29 Personalize Dashboard", and "NEX-33 Time tracking in IDE". The bottom of the screen shows the text "Board owner: root".

For our non coding issues, we added a timer in YouTrack. So we can just start the timer and stop it after work, if we for example write a blog post.

Do you have similar problems with your burndown chart? We would like to hear some feedback for that from you.

Kind regards,

your nEXam-Team



Published by uhrschr

[View all posts by uhrschr](#)

3 thoughts on “Week 6: our dashboard”

ncryptedv1 says:

November 15, 2021 at 6:51 pm | Edit

Greetings Team nEXam,

first of all – your YouTrack is starting to look awesome! I hope you’re able to see the organisational benefits and in which ways this improves your overall workflow. Especially the various charts which give you insight into what might be going wrong with your workflows just with a glimpse.

From the technical perspective I’d like to offer you some ideas on how to ease your life with YouTrack. You could e.g. (if not already done) automate the creation of tasks like “Writing blog post” or “Reviewing blog posts” as these occur in every weekly sprint. For this you could take a look at Workflow rules (especially On-change or Action Rules: <https://www.jetbrains.com/help/youtrack/incloud/Workflow-Rules.html>) to automatically create these tasks on a weekly basis. Regarding the burndown charts: we faced a few similar problems – try tweaking around with the settings. We also had an issue where the time of a card being moved to “In Progress” and being moved to “Done” was tracked separately and taken into account – despite dedicated time tracking being enabled. This seems to have been caused by a separate plugin you may disable. Following just a few recommendations regarding the blog post itself: Try to proof-read your blog posts as there are some grammar mistakes, especially regarding punctuation (commas) and consistency regarding e.g. capitalization. In case you don’t want to invest two people’s time into this you may also use some 0815 online language checker like <https://languagetool.org/de>. A few changes work wonders for the recipients! 😊

Keen on seeing your next blog post!

Best regards,

Oliver from Team Newtry

↳ Reply

undecided1550 says:

November 16, 2021 at 7:10 am | Edit

Hey nEXam Team,

it’s nice that you are using the plugin in your IDE, i think that will give you a better overview in the long run. Unfortunately we don’t work with YouTrack, meaning we can’t really help you on your issue but we are looking forward to other comments that may help you. Maybe you could write in your next blog post how you managed to resolve that issue.

Likewise it would have been great if you included some more screenshots of your dashboard to

this blog post and maybe told us something about your structure and your thoughts behind it.
E.g.:

What about your workflows? Do you have any?

Or some kind of Automation like moving issues automatically into another status after an associated commit was made?

Don't be afraid to tell us everything you did until now 😊

Best wishes,
your UNDECIDED Team

↳ Reply

Pingback: [Midterm – nEXam](#) | [Edit](#)



Week 7: database schema

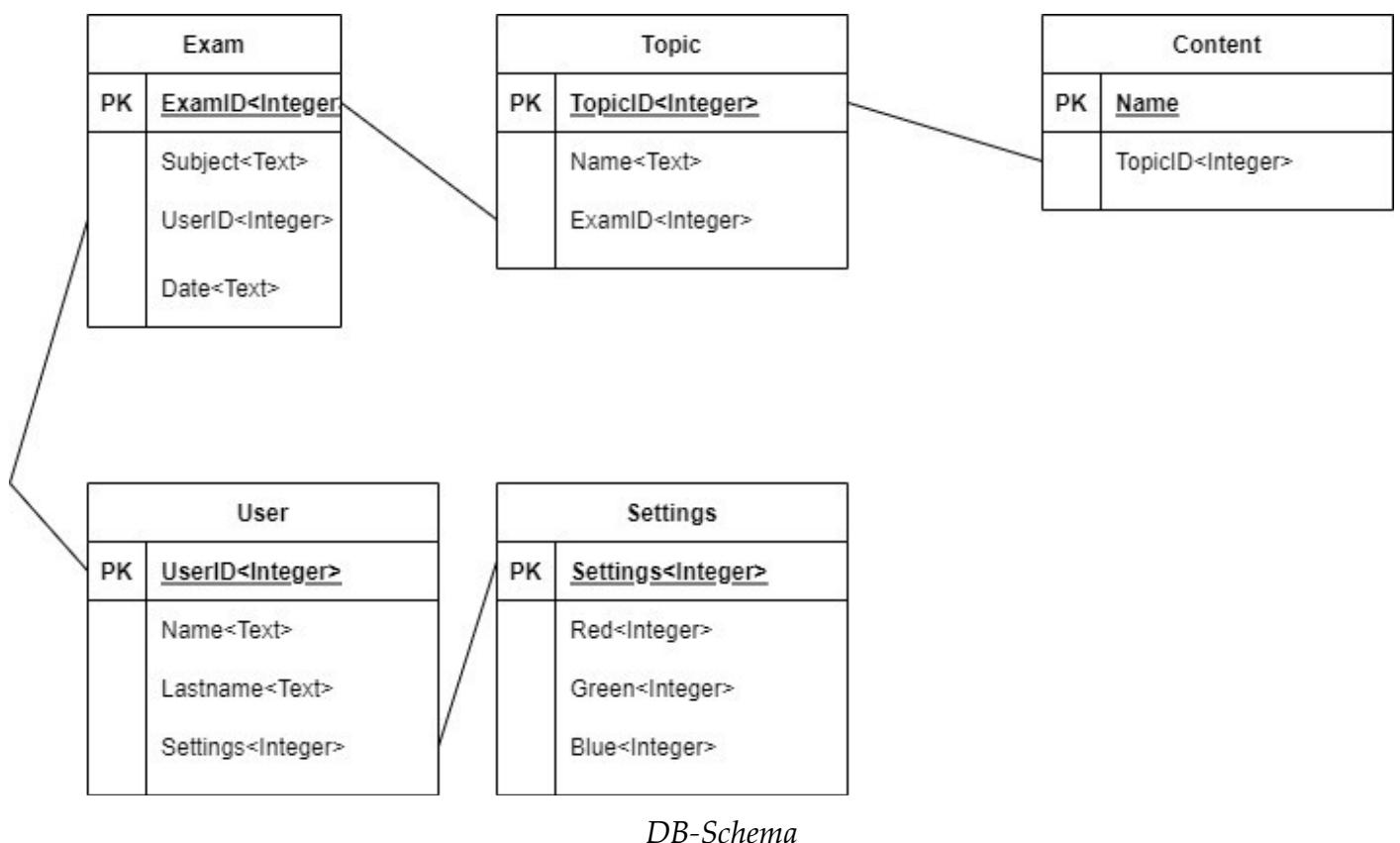
👤 uhrscht 🕒 November 22, 2021December 30, 2021 ⏳ 1 Minute

This week, we created a schema of how we think our database should look.

Soon after we started with planning, we realized our first thoughts weren't that precise, and we needed to adjust our planning. While thinking about the schema and working on it, our project was getting more and more concrete. It started to become real, which was a great feeling for all of us.

Take a look at our GitHub repository (<https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/DB-Schema.jpg>) to stay up to date if you're interested in our progress.

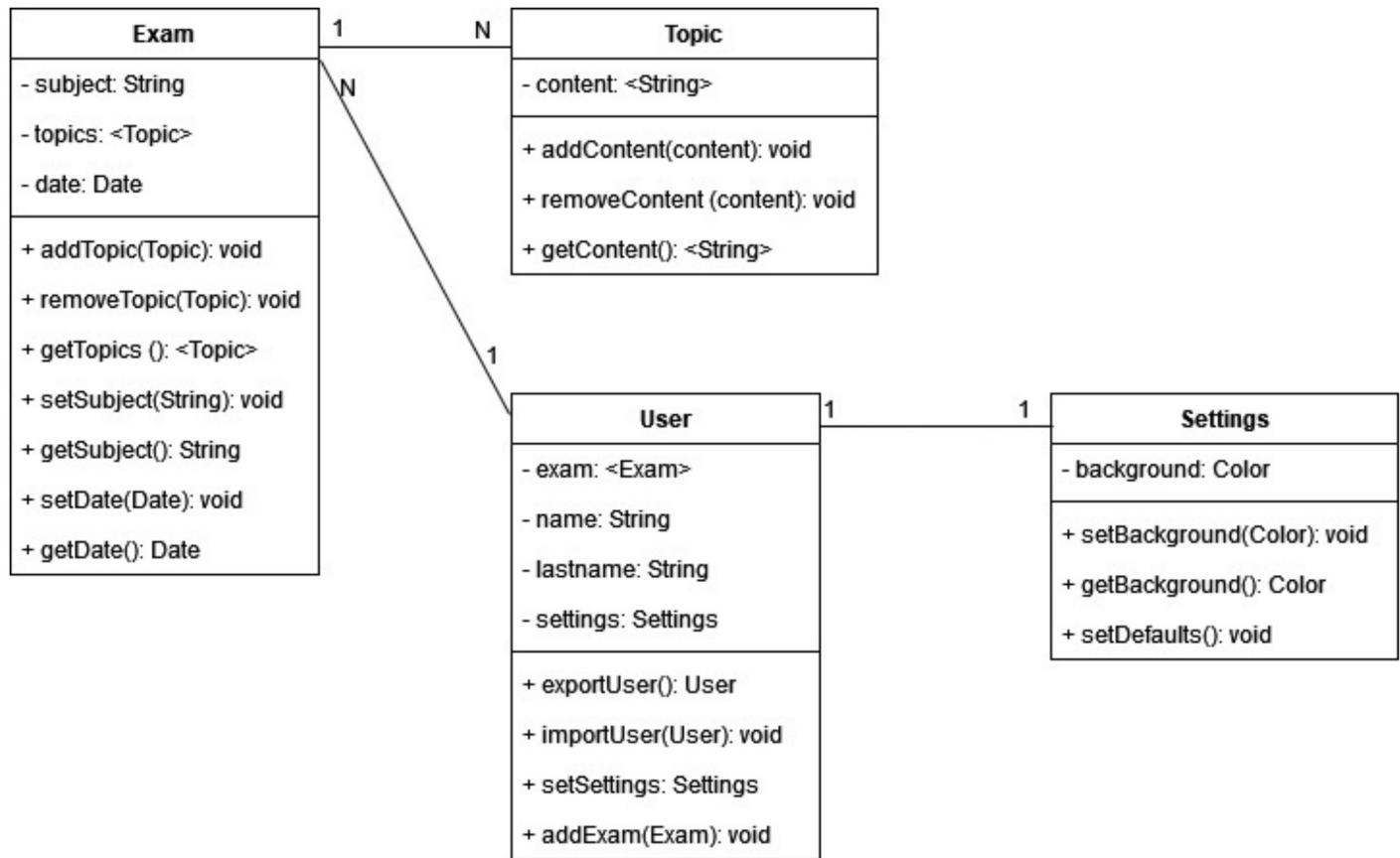
Here's our current db schema as well.



After we created this schema, we started thinking about to set it up. As we mentioned in our [blog post](#) (<https://nexam955203221.wordpress.com/2021/10/19/getting-concrete/>) 'Getting specific' we planned using SQLite as DBMS. We found an extension for our IDE, so we are able to create and modify our db in our IDE, which makes it really comfortable to work with. But each of us has never worked with databases or hasn't done it for a long time. That's why creating it first required some research, and we spent more time than estimated.

If you would like to see how it affected our burndown, take a quick look at our next week's dashboard (<https://dhw-karlsruhe.myjetbrains.com/youtrack/dashboard>).

EDIT: here we show you our class diagram, which you can also find on our [repository](https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/nEXam-UML%20class%20diagram.jpg) (<https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/nEXam-UML%20class%20diagram.jpg>).



Kind regards,

your nEXam Team



Published by uhrscl

[View all posts by uhrscl](#)

5 thoughts on “Week 7: database schema”

[The PiXS Team](#) says:

November 22, 2021 at 10:07 pm | Edit

Howdy,

first of all, we think your blogpost on DB schema is well done. We also like how you report how your planning went and is going.

However, we would have liked to see your class diagrams this week.

We are looking forward to see these diagrams.

Cheerio
The PiXS Team

↳ Reply

uhrscl says:

November 23, 2021 at 8:44 am | Edit

Hello PiXS Team, thanks for your feedback. I just added the class diagramm so you could take a look on it too.

Kind regards,

The nEXam Team

↳ Reply

Team Papatohu says:

November 23, 2021 at 10:28 am | Edit

Hey there,

its's nice to hear that your idea becomes more and more reality! In the edit area you have small spelling mistakes. Besides that is your diagram very clear and well done.

Kind regards

Team Papatohu

↳ Reply

h says:

November 24, 2021 at 6:14 pm | Edit

Hi nEXam Team,

your blog post looks really good. Both the DB schema and the class diagram look well structured and thought out. Good work!

regards

deX-Team

↳ Reply

Pingback: [Midterm – nEXam](#) | [Edit](#)



Week 8: Retrospective

👤 uhrscl 🕒 November 25, 2021December 30, 2021 ⏳ 1 Minute

This week we had our first retrospective meeting, unfortunately not all of us could attend, but we tried our best. Also, this is a usual problem we have to deal with in every project, so we took it as a situation we could count to our experiences.

Below you will find a screenshot of our solution (also in our [repository \(\[https://github.com/Calco2001/nEXam/blob/9bc73b87c7d46e2285744805d9d45d8c1b5ff90e/docs/diagrams/first%20retro/verbesserungspotenzial_vorschlaege.PNG\]\(https://github.com/Calco2001/nEXam/blob/9bc73b87c7d46e2285744805d9d45d8c1b5ff90e/docs/diagrams/first%20retro/verbesserungspotenzial_vorschlaege.PNG\)\)\). We figured out, that we created a good working routine for the 'weekly' tasks, but we are having a lack in responsibilities and coding tasks. Therefore, we set our goal for the coming weeks to define more concrete responsibilities and also schedule some coding sessions.](https://github.com/Calco2001/nEXam/blob/9bc73b87c7d46e2285744805d9d45d8c1b5ff90e/docs/diagrams/first%20retro/verbesserungspotenzial_vorschlaege.PNG)

Was läuft toll im Projekt?	Welche Themen haben noch Verbesserungspotenzial?	Welche Ideen/Maßnahmen fallen euch für die genannten Themen ein?
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">Planung</div> <div style="text-align: center;">Routine</div> </div>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">Aufgaben frühzeitiger erledigen</div> <div style="text-align: center;">Zeitmanagement</div> <div style="text-align: center;">"Rückstand" aufholen</div> <div style="text-align: center;">zu ungenau definierte Arbeitspakete</div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">Abarbeitung des sprints</div> <div style="text-align: center;">kein "Zuständig"-Fühlen für bestimmte Aufgaben</div> <div style="text-align: center;">Youtrack nicht immer auf dem neusten Stand</div> </div>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">Regelmäßige Teammeetings</div> <div style="text-align: center;">Aufgaben in kleinere Arbeitspakete unterteilen</div> <div style="text-align: center;">Aufgaben eindeutig zuordnen</div> <div style="text-align: center;">Kommentare bei Tasks zum momentanen Stand/ wichtigen Infos</div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">klare Verantwortlichkeit im Team</div> <div style="text-align: center;">Youtrack mehr verwenden</div> <div style="text-align: center;">selbstständiger Aufstellen aus Youtrack bearbeiten</div> <div style="text-align: center;">mehr Coden</div> </div>

first retrospective

Apart from this, we added Android Studio to our tech stack. With this application, we can create a virtual device and test our app more easily than if we constantly transfer it to a physical device.

Kind regards,

Your nEXam Team



Published by uhrscl

[View all posts by uhrscl](#)

2 thoughts on “Week 8: Retrospective”

Pingback: [Midterm – nEXam](#) | [Edit](#)

Pingback: [SE2: Week 5: Retrospective the second – nEXam](#) | [Edit](#)

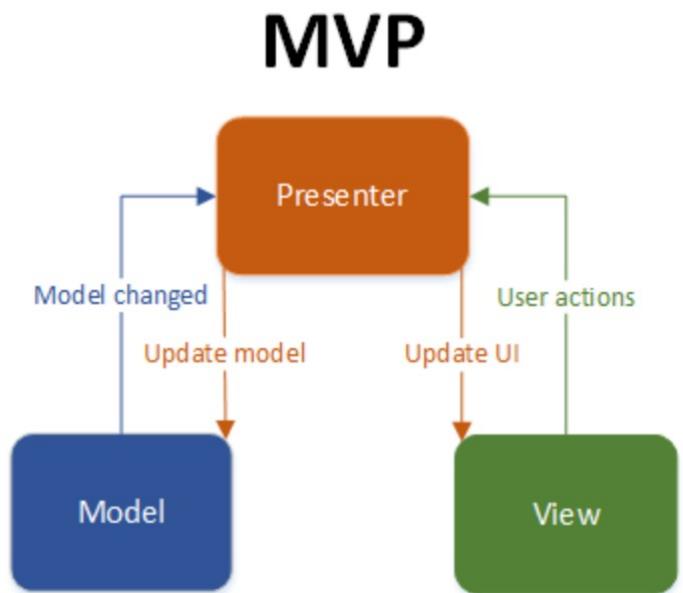


Week 9: Update

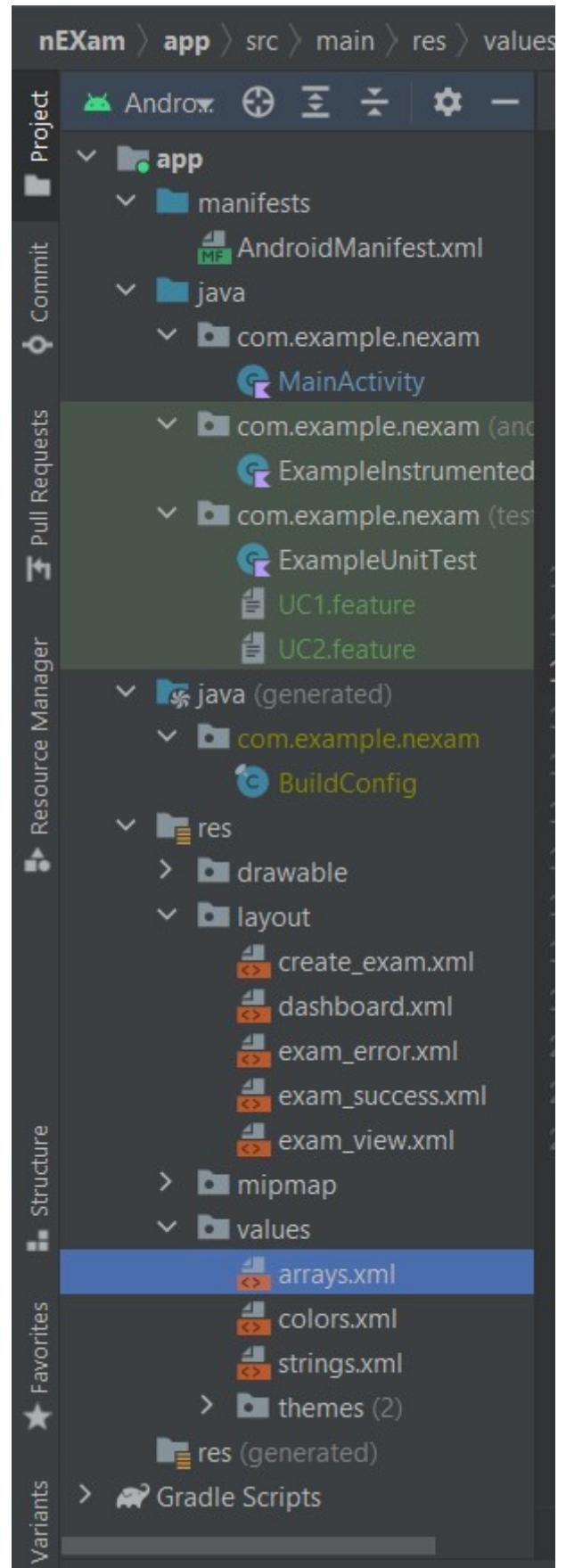
👤 uhrscht 🕒 December 6, 2021December 31, 2021 💼 1 Minute

We are currently in a changing process from maven to gradle. This process started, because we added Android Studio to our tech stack last week. We used Android Studio to start creating our frontend. It's more simple to use than VS Code, we were able to see an instant preview, and we can also simulate smartphones to test the application. Because we created a test project in gradle, we decided to change our build automation tool. Further, we currently aren't using a framework. Beside the switch and the absence of team members we are still behind our plan and caused by this lack of progress we didn't spend focus on a framework. So we couldn't create an architectural UML diagram and compare or overlay it with the UML diagram we already did.

Edit: We didn't know that Android Studio automatically provides an [MVP diagram](#) (<https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/mvp%20android.PNG>). That's why we have added it here afterwards.



You can also find our current structure of the project [here](#) (<https://github.com/Calco2001/nEXam/blob/main/docs/diagrams/mvc%20from%20android%20studio.png>):



We also haven't tried already to run the test because there isn't that much code available. But we have two feature files, which you can review in our [repository](https://github.com/Calco2001/nEXam) (<https://github.com/Calco2001/nEXam>).

Our main task currently is a working demo for the midterm presentation. Until now, we just could show some views, I would recommend you to check them out in our [repository](https://github.com/Calco2001/nEXam/tree/main/docs/wireframes) (<https://github.com/Calco2001/nEXam/tree/main/docs/wireframes>) too (those are available in light and dark theme)

We try to update our [SRS](https://github.com/Calco2001/nEXam/blob/main/docs/SRS.md) (<https://github.com/Calco2001/nEXam/blob/main/docs/SRS.md>) weekly, so

it represents the current state of our project and comparing to the first weeks it looks more filled and how it should look, but there's still a lot to do.

The hope is to get more things done by next week, despite the exams, so that we can give you an interesting presentation with a hopefully working demo.



Published by uhrscl

[View all posts by uhrscl](#)

5 thoughts on “Week 9: Update”

Moritz Sternjakob says:

December 7, 2021 at 10:56 am | Edit

Although you couldn't create a MVC diagram this week, you wrote a very interesting blog and we also see that you did a lot this week.

We also have great understanding for the fact that you have other priorities with your current progress. From our side, we can only advise you to make progress as soon as possible so that you have something interesting to show midterm.

All the best from your workout-log team 😊

↳ Reply

uhrscl says:

December 10, 2021 at 4:24 pm | Edit

Hello Workout-log,

thanks for your advise. We found out, we are able to create a MVC, its implemented in the native Android framework. We are using it without knowing by creating our app in Kotlin. We think knowing this will make coming projects more easy. So i think we will be able to impress you with our midterm presentation 😊

Your nEXam Team

↳ Reply

lukashrnle says:

December 8, 2021 at 9:50 pm | Edit

Dear team nexam,

great post! We certainly hope you can work out the issues with your demo before having to present it.

I looked at your feature-files. They look good to me.

Did you implement all 5 use-cases so far?

Best regards,

Team LectureFeed



↳ Reply

uhrscl says:

December 10, 2021 at 4:28 pm | Edit

Hello Team LectureFeed,

thanks for your review 😊

The good news are that we were able to create a little demo and it works better than we expected a week ago. We made a good progress and even in the current situation we enjoy the progress.

Unfortunately we are not able to implement all use case as we planned. Lets see if we are able to adjust them to hold the scope.

Kind regards,

your nEXam Team

↳ Reply

Pingback: [Midterm – nEXam](#) | Edit

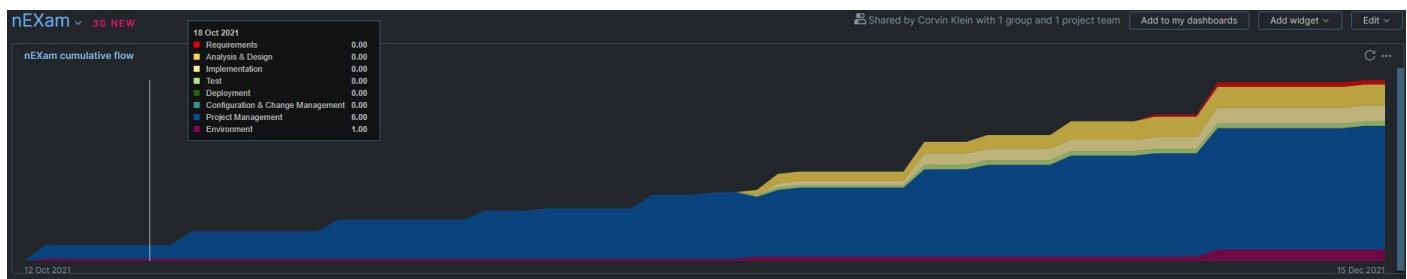


Week 10: Midterm

👤 uhrscht 🕒 December 23, 2021 April 7, 2022 🗓 1 Minute

Hello guys, it was a busy term, and we are happy about what we did and the experiences we collected. Here you will find our [GitHub repo](https://github.com/Calco2001/nEXam) (<https://github.com/Calco2001/nEXam>) and below our most important links to summon up our project so far.

[Project Management \(YouTrack\) \(<https://dhw-karlsruhe.myjetbrains.com/youtrack/dashboard?id=fc14affe-ae44-4123-8d77-b5828e4aa5eb>\):](https://dhw-karlsruhe.myjetbrains.com/youtrack/dashboard?id=fc14affe-ae44-4123-8d77-b5828e4aa5eb)



cumulative flow



burn down week 9

Quality:

- [SAD document \(<https://github.com/Calco2001/nEXam/blob/main/docs/SAD.md>\)](https://github.com/Calco2001/nEXam/blob/main/docs/SAD.md)

Requirements:

- [SRS \(<https://github.com/Calco2001/nEXam/blob/main/docs/SRS.md>\)](https://github.com/Calco2001/nEXam/blob/main/docs/SRS.md)
- [Use Case Specification \(directory\) \(\[https://github.com/Calco2001/nEXam/tree/main/docs/Use_Case_Specification\]\(https://github.com/Calco2001/nEXam/tree/main/docs/Use_Case_Specification\)\)](https://github.com/Calco2001/nEXam/tree/main/docs/Use_Case_Specification)
- [Testing files \(<https://github.com/Calco2001/nEXam/tree/main/app/src/test/java/com/example/nexam>\)](https://github.com/Calco2001/nEXam/tree/main/app/src/test/java/com/example/nexam)

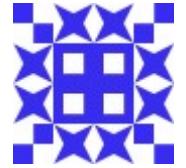
Blog posts:

- [Week 1: The Vision for Great Success](https://nexam955203221.wordpress.com/2021/10/05/blogpost-vision/) (<https://nexam955203221.wordpress.com/2021/10/05/blogpost-vision/>)
- [Week 2: First Step – Organization](https://nexam955203221.wordpress.com/2021/10/12/first-step-organisation/) (<https://nexam955203221.wordpress.com/2021/10/12/first-step-organisation/>)
- [Week 3: Getting specific](https://nexam955203221.wordpress.com/2021/10/19/getting-concrete/) (<https://nexam955203221.wordpress.com/2021/10/19/getting-concrete/>)
- [Week 4: Check out our GitHub repo](https://nexam955203221.wordpress.com/2021/10/26/week-4-check-out-our-githubrepo/) (<https://nexam955203221.wordpress.com/2021/10/26/week-4-check-out-our-githubrepo/>)
- [Week 5: It's not a bug, it's a .feature](https://nexam955203221.wordpress.com/2021/11/02/week-5-its-not-a-bug-its-a-feature/) (<https://nexam955203221.wordpress.com/2021/11/02/week-5-its-not-a-bug-its-a-feature/>)
- [Week 6: Our Dashboard](https://nexam955203221.wordpress.com/2021/11/15/week-6-our-dashboard/) (<https://nexam955203221.wordpress.com/2021/11/15/week-6-our-dashboard/>)
- [Week 7: Database schema](https://nexam955203221.wordpress.com/2021/11/22/week-7-database-schema/) (<https://nexam955203221.wordpress.com/2021/11/22/week-7-database-schema/>)
- [Week 8: Retrospective](https://nexam955203221.wordpress.com/2021/11/25/week-8-retrospective/) (<https://nexam955203221.wordpress.com/2021/11/25/week-8-retrospective/>)
- [Week 9: Update](https://nexam955203221.wordpress.com/2021/12/06/238/) (<https://nexam955203221.wordpress.com/2021/12/06/238/>)

Other:

- [Handout midterm presentation](https://github.com/Calco2001/nEXam/blob/main/presentation/nEXam%20-%20midterm%20presentation.pdf) (<https://github.com/Calco2001/nEXam/blob/main/presentation/nEXam%20-%20midterm%20presentation.pdf>)
- [midterm presentation](https://docs.google.com/presentation/d/1STyQueeX2z3PaKITfged837gBu4IunW3-VaVGUIJJeE/edit?usp=sharing) (<https://docs.google.com/presentation/d/1STyQueeX2z3PaKITfged837gBu4IunW3-VaVGUIJJeE/edit?usp=sharing>)

Published by uhrscl



[View all posts by uhrscl](#)

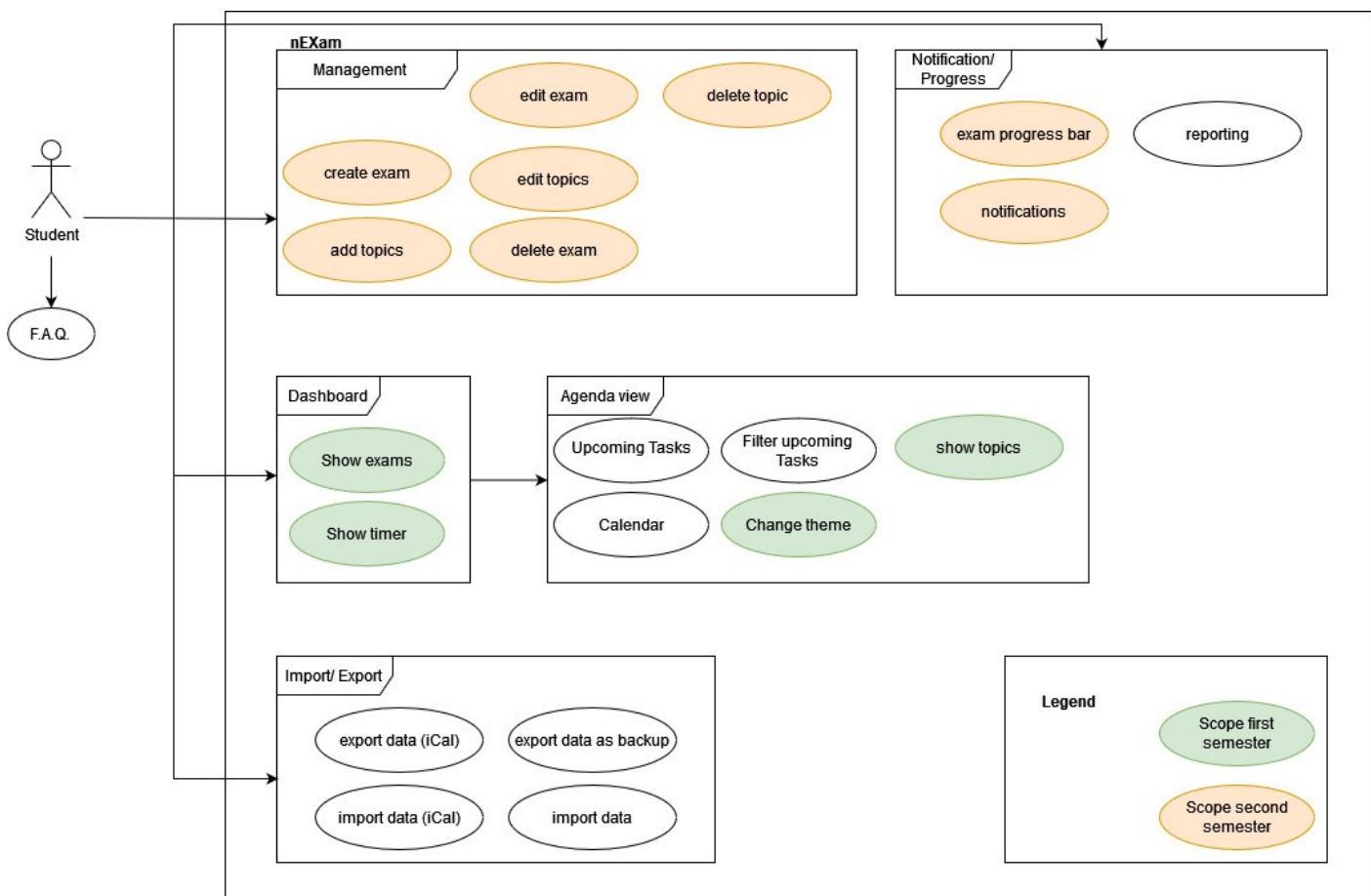


SE2 Week 1: Welcome back!

👤 Franziska_B2 ⏰ April 9, 2022April 11, 2022 🗓 1 Minute

Hi guys, welcome back to our blog post. Today we want to update you on the latest changes since our midterm post last semester.

Unfortunately, we lost another team member. So we did our best to adapt our use case diagram, so hopefully we can still make progress on our project. Below, you can see the new use case diagram. The current version of it can be found [here \(https://github.com/nEXam-App/nEXam-doc/blob/main/diagrams/nEXam-use%20case%20diagram.jpg\)](https://github.com/nEXam-App/nEXam-doc/blob/main/diagrams/nEXam-use%20case%20diagram.jpg).



Use case diagram

As you can see, we made a different choice on how we want to store the user data. This is because we had some issues with adding new data to our xml file. The current choice is to store it locally in SQLite using the Room abstraction layer. There are still some issues, but hopefully we will be able to fix them.

Also, we added some new features to our diagram. We want to include a progress bar, so the user can see how much is left to learn. Besides this, the user should receive notifications in advance, so they can start learning when they have enough time left.

Additionally, we're working on the feedback for the midterm. Therefore, we have created a [new GitHub organization](#) (<https://github.com/nEXam-App>) and separated the documents and the code from each other. So make sure, you use the right link!

We look forward to continuing our project, even though we're down to two team members, and making as much progress as possible!

Published by Franziska_B2



[View all posts by Franziska_B2](#)



SE2 Week 2: Consider your risks

👤 Franziska_B2 ⏰ April 12, 2022April 15, 2022 💻 1 Minute

To be aware of possible risks for our project, we have discussed them. Below, you can see the current list of risks, we could find. Since the image is quite small, you will also find a link to our [spreadsheet](https://docs.google.com/spreadsheets/d/1D37-90Dhg_P-9yCL7CjMfnPYa3YZMsxcbcvK8pcW6_Z8/edit?usp=sharing) (https://docs.google.com/spreadsheets/d/1D37-90Dhg_P-9yCL7CjMfnPYa3YZMsxcbcvK8pcW6_Z8/edit?usp=sharing).

Risk Management - SE2 nEXam						
Risk	Description	Prob. of Occurrence	Damage	Risk Factor	Mitigation Strategy	Person in Charge of Tracking
Scope too big	planned to many things for this semester	60,00%	7	4,7	more communication about the tasks and progress	All
Database crashes	the data isn't backed up	40,00%	8	3,2	implement export option to backup the data	All
new technology	project involves the use of new technology	80,00%	3	2,4	learn new technology	All
Database get compromised	there is no auth so everyone has access	70,00%	3	2,1	set a pin to open the app	All
Use cases	misunderstands what is meant with each use case	30,00%	6	1,8	regular meetings, ask questions in case of ambiguity	All
inexperience	inexperience project managers	60,00%	3	1,8	use tools regularly to get used to the workflow	All
Loose more team members	small team	20,00%	8	1,6	all members need to know about all issues so they could continue	All
User	dissatisfaction with the app	40,00%	3	1,2	get feedback from others	All
Data is invalid	Entered data can not be saved	20,00%	3	0,6	check data before saving	All

Risk management

As you can see, we assume that our greatest risk is a too big scope. Even though we know we can adjust it if needed, it's difficult to estimate how much we can handle as a team of two in addition to preparing for the other lessons. We have so many ideas about what we want to do, but we also know that we won't be able to do it all. The strategy for this risk is a good communication and monitoring progress. Another risk, we would like to point out, is that our project has a high risk of loosing its members since it happened in the past, almost like the Bermuda Triangle. But that should be able to occur less now, since the team can't really get smaller anymore.

Additionally, we would like to mention, that we're making progress with the integration of Room respectively SQLite.

We're looking forward to seeing you again next week, when we report on our time spent on the use cases and the function points.

Published by Franziska_B2



[View all posts by Franziska_B2](#)

2 thoughts on “SE2 Week 2: Consider your risks”

Raynex says:

April 15, 2022 at 7:40 pm | Edit

Kia Ora nEXam-Team,

the risk of a scope that is too big is understandable, especially with your small team. All the risks you have in your spreadsheet are understandable. Hopefully, you do not lose any more team members, as it would probably terminate your project.

I hope you can finish your work, without any of the risks you mentioned or any unforeseen risks happening.

~Team Papatohu

↳ Reply

undecided1550 says:

April 27, 2022 at 10:01 pm | Edit

Dear nEXam Team,

first of all I really like your list of risks, you can see that you put much thought into it and it all makes sense. Usually you would not set all people of your team in charge of a risk, but considering that you're only a team of two people it's fine.

I really hope that you won't loose any more members and finish your project without any major issues

Kind regards

The UNDECIDED Team

↳ Reply

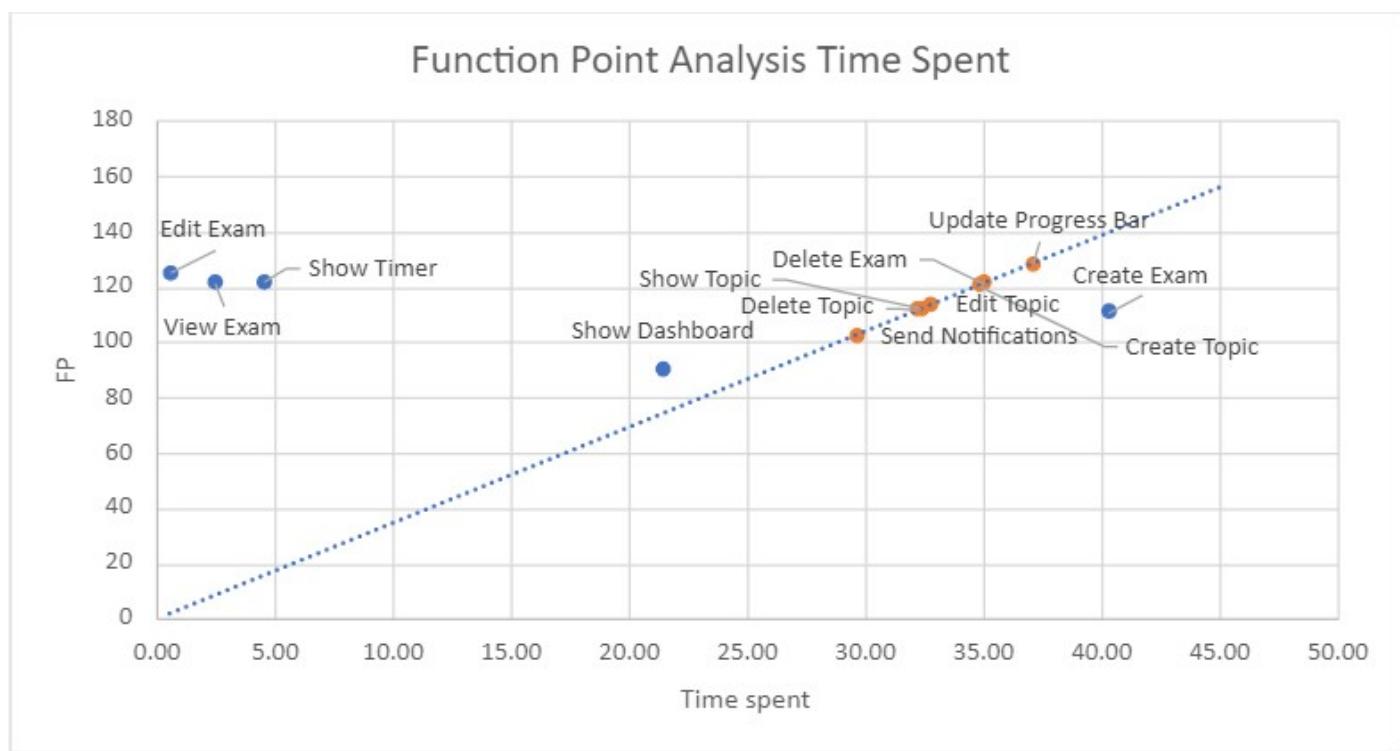
W

SE2 Week 3: Function Points Diagram

👤 uhrschr 📅 April 23, 2022 🕒 April 23, 2022 ⏳ 1 Minute

This week we created a diagram which shows the needed time and a calculation of function points for our use cases from last semester. The function points are calculated by a [website](http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP_Calc.html?tCountVal=0#Complexity) (http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP_Calc.html?tCountVal=0#Complexity), where we just entered some values. To verify our solutions, we put screenshots with our values for every use case in the [diagram directory](https://github.com/nEXam-App/nEXam-doc/tree/main/diagrams/FP) (<https://github.com/nEXam-App/nEXam-doc/tree/main/diagrams/FP>) of our repository, or you can also find them in the [use case documents](https://github.com/nEXam-App/nEXam-doc/tree/main/Use_Case_Specification) (https://github.com/nEXam-App/nEXam-doc/tree/main/Use_Case_Specification).

With the trend line through our use cases, we could estimate the time we will need for our use cases this semester after calculating again the function points. The values for those use cases are in our repository as well.



The blue points are the use cases we implemented last semester, the orange ones will follow this semester.

You may notice, we got more outliers than helpful values. That's because in our current structure, an object like exam is build in many files. We created the app this way, because it works better with our room database. The high number of files increases the function points. But we just needed to create the files once (e.g. create exam) and just adjust those files a little for other use cases (e.g. edit exam). So the high function point use cases consumed a low amount of time. Another reason is that we started from coding with a basic setup and could not really transfer the time for this to the individual

use cases. However, now that we know how to better divide and document this, better values should emerge in the future. We also needed training time, but that is difficult to subtract retroactively.

We are still positive that we will be able to realize our use cases for this semester despite the high time specifications in the graph. See you next week!



Published by uhrscl

[View all posts by uhrscl](#)

2 thoughts on “SE2 Week 3: Function Points Diagram”

binmin1.3 says:

April 25, 2022 at 3:57 pm | Edit

Hey nEXam team,

first of all: thank you very much for the diagram and for the detailed description and reasoning! Your diagram is very clear and understandable. The reasons for your outliers are well formulated and clearly explained. In itself, a good block post that has fulfilled this week's task well. We wish you good luck with the realization of the use cases!

Kind regards
workout-log

↳ Reply

P3lina says:

April 25, 2022 at 9:16 pm | Edit

Hello nEXam team,

Like the workout-log team, I thank you for your diagram, which helps understand Function Points.

I think we even had the same problems determining the time spent on each use case as one has to get familiar with the used technologies, which isn't really assignable to a specific use case.

Overall your block post had a good structure and was a pleasure to read.

Good luck with the realization of the remaining use cases

~Team Papatohu

↳ Reply

W

SE2 Week 4: Testing

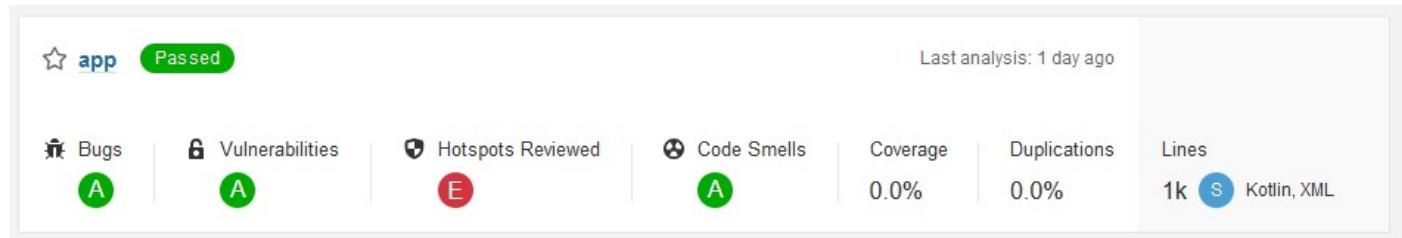
👤 Franziska_B2 ⏰ April 30, 2022May 1, 2022 💡 1 Minute

Hey everyone,

This week, we want to share with you our testing strategy. Therefore, we want to use three different types of tests: JUnit tests, Cucumber tests and of course user tests.

We decided against using a CI/CD pipeline. A possible choice would have been Travis, but since we didn't have the ability to run the tests locally before pushing them and creating pull requests, that made our decision. Instead, we agreed to run the tests locally after making the changes. That way, we can still be sure that we don't break something that already worked before we push it to a branch.

We will be using SonarQube for our code coverage reports. Unfortunately, we are still having issues integrating it with our GitHub flow. As a short term solution, we had it analyzed locally. You can see a screenshot of it [here](https://github.com/nEXam-App/nEXam-doc/blob/main/diagrams/SonarQubeResult.PNG) (<https://github.com/nEXam-App/nEXam-doc/blob/main/diagrams/SonarQubeResult.PNG>).



SonarQube Analysis

We were able to use the time saved that we would otherwise have needed to integrate a pipeline to familiarize ourselves with the JUnit tests. As you can see below, we were able to implement some, which run successfully. These currently check the context of the app and the function to check user input. We can also use JUnit tests to check the integration of our database. So both the creation, display, editing, and deletion of checks are tested.

Status		6 passed	6 tests, 12 s 159 ms
Filter tests:		Q	L
Tests		Duration	HTC HTC U11
✓ Test Results		432 ms	6/6
✓ AddExamFragmentTest		101 ms	2/2
✓ isNotValid_ShouldReturnFalse		76 ms	✓
✓ isValid_ShouldReturnTrue		25 ms	✓
✓ ExampleInstrumentedTest		25 ms	1/1
✓ useApplicationContext		25 ms	✓
✓ SimpleEntityReadWriteTest		306 ms	3/3
✓ updateExam		126 ms	✓
✓ addAndReadExam		102 ms	✓
✓ deleteExam		78 ms	✓

Implemented JUnit-Tests

More detailed information about the different JUnit-tests can be found in the [source code folder](#). (<https://github.com/nEXam-App/nEXam/tree/main/app/src/androidTest/kotlin>) The required dependencies were added in our [build.gradle file](#) (<https://github.com/nEXam-App/nEXam/blob/main/app/build.gradle>).

Unfortunately, since we didn't get our cucumber tests up and running last semester, we don't know yet if we can make it this semester in terms of time. The previous attempts were unfortunately unsuccessful and extremely time-consuming. But within our limited capacity, we will try again as soon as there is an open time slot. The .feature files exists and can be accessed through the [use case specifications](#) (https://github.com/nEXam-App/nEXam-doc/tree/main/Use_Case_Specification).

Finally, we would like to point out our newly created [testing document](#) (https://github.com/nEXam-App/nEXam-doc/blob/main/test_plan_template.md), where we have recorded our testing strategies.

Greetings,
Your nEXam-Team



Published by Franziska_B2

[View all posts by Franziska_B2](#)

2 thoughts on “SE2 Week 4: Testing”

DueTo says:

May 1, 2022 at 10:18 pm | Edit

Hey nEXam-Team,

We have read your blog post with great pleasure.

It is nice to see that you have implemented working JUnit tests. We are looking forward to seeing how much they help you without CI/CD pipeline.

It's a pity you didn't have more time to look at the other testing strategies, they could have helped you catch bugs in your code early in the future.

Friendly regards,

DueTo

↳ Reply

2376daria says:

May 2, 2022 at 8:51 am | Edit

Hi eEXam-Team,

we enjoyed reading your blog post. It looks like your testing strategy works pretty good.

It's very nice that all of the unit-tests you have implemented already are working. We also looked into your documentation und we think you described your test plan for your project very accurate and struktured.

So, we like it!

Regards,

DHBWorld-Team

↳ Reply



SE2 Week 5: Retrospective the second

👤 Franziska_B2 ⏰ May 3, 2022June 17, 2022 💻 1 Minute

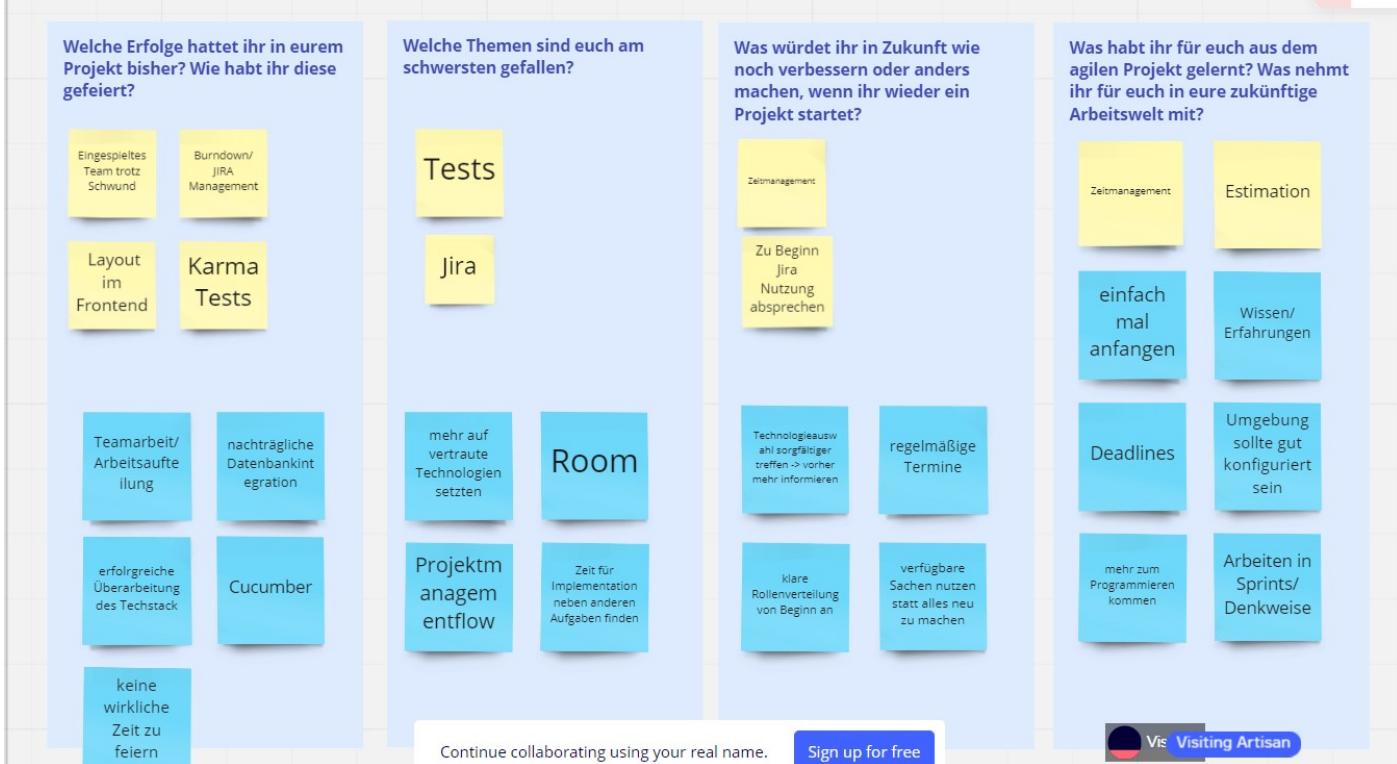
Today we had our second retrospective. To get a better feeling about our improvements, we want to compare it in the end to the one we did the semester before. The old blog post about it can be found [here](https://nexam955203221.wordpress.com/2021/11/25/week-8-retrospective/) (<https://nexam955203221.wordpress.com/2021/11/25/week-8-retrospective/>).

Every team started with the Check in, where we should picture and mention what agile working means for us.



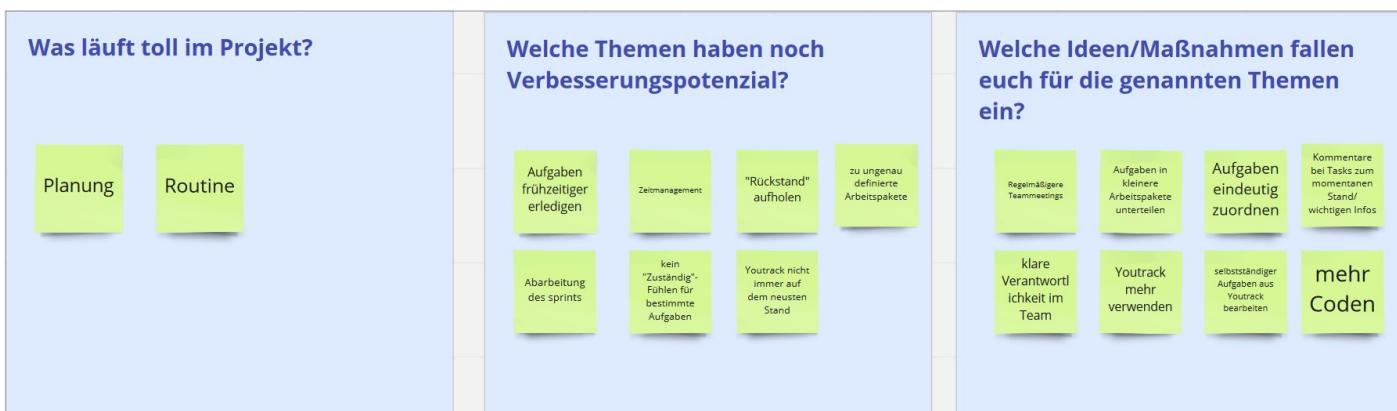
[Check in \(\[https://github.com/nEXam-App/nEXam-doc/blob/c8432e1f4653661187c9b33a99d76871e3c3adde/diagrams/Retro22_intro.PNG\]\(https://github.com/nEXam-App/nEXam-doc/blob/c8432e1f4653661187c9b33a99d76871e3c3adde/diagrams/Retro22_intro.PNG\)\)](https://github.com/nEXam-App/nEXam-doc/blob/c8432e1f4653661187c9b33a99d76871e3c3adde/diagrams/Retro22_intro.PNG)

After that, we continued working in smaller groups. Together with the team Papatohu, we created the following board in three sessions. As you can see, we learned a lot by now and know what to improve in the future or in other projects. We also now know that it's important to recognize and celebrate our own successes, rather than putting that at the end of the project. There is still significant room for improvement here, as we have simply continued working on another thing.



Our board (<https://github.com/nEXam-App/nEXam-doc/blob/c8432e1f4653661187c9b33a99d76871e3c3adde/diagrams/retro22.PNG>) (yellow – Papatohu, blue – nEXam)

If you compare the board from this and the last retro, you can see, that we also improved doing a retro itself. We now knew better what was expected of us, and this also made it much easier and more efficient for us.



Board from first retro (<https://github.com/nEXam-App/nEXam-doc/blob/c8432e1f4653661187c9b33a99d76871e3c3adde/diagrams/firstRetro.png>)

We hope you also learned something from your retro and look forward to seeing you again next week.

Published by Franziska_B2



[View all posts by Franziska_B2](#)

One thought on “SE2 Week 5: Retrospective the second”

DHBW Experts says:

May 6, 2022 at 12:38 pm | Edit

Hi nEXam,

It's nice to see that you made great progress both in your app and in the whole scrum process! Just like you, we want to focus more on celebrating success, although it's not always easy (especially if your team consists of only two people).

I wish you all the best for the future development process!

In case you wonder: Yes, we don't have to do peer reviews this week, but we somehow managed to miss one last semester, and now we have to make up for it... hehe 😊

Best regards and CYK CYK,
DHBW Experts

↳ Reply



SE2 Week 6: Refactoring

👤 Franziska_B2 ⏰ May 11, 2022May 19, 2022 ⏳ 1 Minute

This week we've been looking at refactoring. For this, we followed [Fowler's book](#) (<http://silab.fon.bg.ac.rs/wp-content/uploads/2016/10/Refactoring-Improving-the-Design-of-Existing-Code-Addison-Wesley-Professional-1999.pdf>) and improved the code step by step.

So that everyone can take something from this exercise, everyone created their own repo and worked there.

Below, you can find the links to our two:

Simon: [SE-refactoring](https://github.com/uHrsch/se_refactoring) (https://github.com/uHrsch/se_refactoring)

Franziska: [SE-refactoring](https://github.com/Franziska1402/SE-refactoring) (<https://github.com/Franziska1402/SE-refactoring>)

Although we used different IDEs, Eclipse and IntelliJ, two provided features stood out to us in a particularly positive way.

One is the rename command, which allows you to name variables, methods, classes, or even packages as well as their usages more descriptively with almost no effort.

The other command is the move command. The fact that the IDE already suggests which class is suitable for swapping was a new insight for both of us.

See you again next week,

Your nEXam team

Published by Franziska_B2



[View all posts by Franziska_B2](#)

2 thoughts on “SE2 Week 6: Refactoring”

llambdaa says:

May 13, 2022 at 9:39 am | Edit

Hello dear nexam team!

First, I like that the blog post is short. You know, I am lazy 😊

Nice to head, that you have learned something new about your IDE.
(SPOILER: There are sooo many more things to discover!)

Regarding your refactoring (@Franziska):

I think, it is overkill to create a separate class for each price type. First, it is work to do so and secondly,

you technically still allow to pass invalid price types into your function(s). You actually catch that case.

It might be much simpler to make an enum that encodes the price type. Then, expect that price type,

instead of an integer. That way, you will only have valid price types, namely one of the enumeration.

The integer has an unrestricted value range, but you only want a handful of them to encode something.

Furthermore, you would not have to check if the value matches the id of a price type in order to create

a specific object for that particular price type. However, when calculating the actual price, you have to

use a switch statement (or something equivalent) that triggers the code you have put into the classes

of each price type respectively.

Tip: If you have a simple value collection (such as price types) with fairly limited range, then use enumerations.

That way, you can expect that type where needed and be sure, that you only get valid values there. You spare

yourself much code and let the compiler do the work to ensure validity of the parameters.

That is only a suggestion tho, it might make things simpler and be more stable, because no invalid price types exist 😊

Have a nice day!

↳ Reply

undecided1550 says:

May 29, 2022 at 6:29 pm | Edit

Dear nexam Team,

your blog post is short and has everything important in it, what I really like. I also enjoyed the fact that you used sonar to scan your projects, because it's such a nice tool. It highlights code smells that you probably would've missed (at least it does this for me :D). If you would've left some words about sonar, it would be a nice addition to your blogpost.

Overall your code looks good and you followed the guidelines from the book.

I wish you good luck with further continuing your project.

Kind regards

The UNDECIDED Team

↳ Reply

SE2 Week 7: Design Patterns

👤 uhrscht ⌚ May 19, 2022May 20, 2022 ⌛ 2 Minutes

This week we learned about improving our code. There are several types of design patterns that are used to make the code cleaner and easier to understand.

Since we were struggling with our app and felt like we weren't making good progress, we were afraid to add complexity with Design Patterns. But luckily, after searching for design patterns, we realized that we were using not one, but five design patterns!

We would like to show you which patterns we already use and in which file you can find them in our repository. For the sake of simplicity we present you one in detail, for the others you will find a very good explanation with this link (<https://www.raywenderlich.com/28238259-android-app-distribution-tutorial-from-zero-to-google-play-store#toc-anchor-001>).

We are using an observer in our [DashboardFragment.kt](#) (<https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/DashboardFragment.kt>)

```
1  viewModel.allExams.observe(this.viewLifecycleOwner) { exams ->
2      exams.let {
3          adapter.submitList(it)
4      }
5 }
```

With this pattern, you can define a one-to-many dependency between objects. As soon as one object changes its state, the dependent objects receive a notification and are updated. Additionally, this pattern is versatile, it can be used for indeterminate time like API-calls or you can use it to respond to user input.

You define observable objects that output values and define subscribers that wait for these values and respond to them, in our case the exams with all their possible changes.

Also, we use a builder to create our database, here is a snippet from our code:

[ExamDetailFragment.kt](#) (<https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/ExamDetailFragment.kt>)

```
1  private fun showConfirmationDialog() {
2      MaterialAlertDialogBuilder(requireContext())
3          .setTitle(getString(android.R.string.dialog_alert_title))
4          .setMessage(getString(R.string.delete_question))
5          .setCancelable(false)
6          .setNegativeButton(getString(R.string.no)) { _, _ -> }
7          .setPositiveButton(getString(R.string.yes)) { _, _ ->
8              deleteExam()
9          }
10     .show()
11 }
```

In [DashboardFragment.kt](https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/DashboardFragment.kt) (<https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/DashboardFragment.kt>) we are using a recyclerview:

```
1 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
2     super.onViewCreated(view, savedInstanceState)
3
4     val adapter = ExamListAdapter {
5         val action =
6             DashboardFragmentDirections.actionDashboardFragmentToExamDe
7             this.findNavController().navigate(action)
8     }
9     binding.recycler_view.layoutManager = LinearLayoutManager(this.conte
10    binding.recycler_view.adapter = adapter
11    // Attach an observer on the allExams list to update the UI automat
12    // changes.
13    viewModel.allExams.observe(this.viewLifecycleOwner) { exams ->
14        exams.let {
15            adapter.submitList(it)
16        }
17    }
}
```

You will find a factory in our [nEXamViewModel.kt](https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/NexamViewModel.kt) (<https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/NexamViewModel.kt>):

```
1 class nEXamViewModelFactory(private val examDao: ExamDao) : ViewModelProvide
2     override fun <T : ViewModel> create(modelClass: Class<T>): T {
3         if (modelClass.isAssignableFrom(nEXamViewModel::class.java)) {
4             @Suppress("UNCHECKED_CAST")
5             return nEXamViewModel(examDao) as T
6         }
7         throw IllegalArgumentException("Unknown ViewModel class")
8     }
9 }
```

And we are using an adapter in our [ExamListAdapter.kt](https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/ExamListAdapter.kt) (<https://github.com/nEXam-App/nEXam/blob/main/app/src/main/java/com/example/nexam/ExamListAdapter.kt>):

```

1  class ExamListAdapter(private val onExamClicked: (Exam) -> Unit) :
2      ListAdapter<Exam, ExamListAdapter.ExamViewHolder>(DiffCallback) {
3
4      override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): Exam
5          return ExamViewHolder(
6              ExamListExamBinding.inflate(
7                  LayoutInflater.from(
8                      parent.context
9                  )
10             )
11         )
12     }
13
14     override fun onBindViewHolder(holder: ExamViewHolder, position: Int) {
15         val current = getItem(position)
16         holder.itemView.setOnClickListener {
17             onExamClicked(current)
18         }
19         holder.bind(current)
20     }
21
22     class ExamViewHolder(private var binding: ExamListExamBinding) :
23         RecyclerView.ViewHolder(binding.root) {
24
25         fun bind(exam: Exam) {
26             binding.examName.text = exam.nameOfSubject
27             binding.date.text = exam.dateOfExam.toString()
28         }
29     }
30
31     companion object {
32         private val DiffCallback = object : DiffUtil.ItemCallback<Exam>() {
33             override fun areItemsTheSame(oldExam: Exam, newExam: Exam): Boolean
34                 return oldExam === newExam
35             }
36
37             override fun areContentsTheSame(oldExam: Exam, newExam: Exam): Boolean
38                 return oldExam.nameOfSubject == newExam.nameOfSubject
39             }
40         }
41     }
42 }

```

We have also added the new uml class diagram with the patterns in [our SAD](https://github.com/nEXam-App/nEXam-doc/blob/main/SAD.md) (<https://github.com/nEXam-App/nEXam-doc/blob/main/SAD.md>).

Since we are using them from the beginning, we are not able to show you a before and after comparison, so we decided to just show you which we are using. The used tutorials, for example one from Google for creating an android app, already included them, and we thought it would be good practice. Nice to know that it saved us a lot of time this week!

Published by uhrscl



[View all posts by uhrsche](#)

3 thoughts on “SE2 Week 7: Design Patterns”

DueTo says:

May 20, 2022 at 11:31 am | Edit

Hello Nexam

It's nice to see that you already used different design patterns in your source code from the beginning.

It would have been interesting if you would have compared your design patterns with other patterns to show why you used them.

We hope you will use the time you gained during the week well and get further in your project!

Best regards

DueTo

↳ Reply

uhrsche says:

June 17, 2022 at 10:46 pm | Edit

Hello DueTo

because we never created an app where worked quite close to tutorials from google with recommanditions for android apps. So we started using the patterns without knowing.. 😊

Looking back we need to say the patterns which slowed our start are helping a lot after understanding them.

↳ Reply

2376daria says:

May 30, 2022 at 8:09 pm | Edit

Hi nEXam-Team,

we enjoyed reading your blog post. It looks like you already have some experience with design patterns and best practices. Five design patterns from the beginning is quite a lot!

We think, your blog post (or documentation) is missing a short explanation, why it was necessary to use the Observer pattern. Overall, your repos and diagrams look very nice.

Regards,

DHBWorld-Team

↳ Reply



SE2 Week 8: Metrics

👤 uhrschn ☰ June 6, 2022June 17, 2022

🕒 1 Minute

Hello there,

this week we should take a look at metrics. Metrics can be used to help you to ensure the quality of your code.

To keep maintain the quality and metrics we use SonarCloud, we added SonarCloud for the refactoring a few weeks ago.

Since there are a lot of metrics possible, we will just focus on two of them:

- Cyclomatic complexity
- Vulnerabilities

Cyclomatic complexity

This metrics indicates how complex a class is. The calculation is based on the number of ways to get through the code.

 app/src/main/java/com/example/nexam/AddExamFragment.kt	12
 app/src/main/java/com/example/nexam/NexamViewModel.kt	12
 app/src/main/java/com/example/nexam/ExamDetailFragment.kt	7
 app/src/main/java/com/example/nexam/ExamListAdapter.kt	5
 app/src/main/java/com/example/nexam/MainActivity.kt	3
 app/src/main/java/com/example/nexam/DashboardFragmnt.kt	2
 app/src/main/java/com/example/nexam/data/ExamRoomDatabase.kt	1
 app/src/main/java/com/example/nexam/data/Exam.kt	0
 app/src/main/java/com/example/nexam/data/ExamDao.kt	0
 app/src/main/java/com/example/nexam/NexamApplication.kt	0

Cyclomatic complexity of the nEXam repository by SonarCloud

The complexity of many files is higher than we expected. The high complexity may be caused by the high amount of design patterns we use. We still have some trouble when we add features to understand the working between the files, which costs us a lot of time. The high score shows us that, our calculation of the [function points \(<https://wordpress.com/post/nexam955203221.wordpress.com/357>\)](https://wordpress.com/post/nexam955203221.wordpress.com/357) is correctly high. Unfortunately, the recommendations of Google lead to a high complexity, so we aren't able to reduce the high score.

Vulnerabilities

This metric shows the security of the code.

app/src/main/java/com/example/nexam/AddExamFragment.kt	0
app/src/main/java/com/example/nexam/DashboardFragment.kt	0
app/src/main/java/com/example/nexam/data/Exam.kt	0
app/src/main/java/com/example/nexam/data/ExamDao.kt	0
app/src/main/java/com/example/nexam/ExamDetailFragment.kt	0
app/src/main/java/com/example/nexam/ExamListAdapter.kt	0
app/src/main/java/com/example/nexam/data/ExamRoomDatabase.kt	0
app/src/main/java/com/example/nexam/MainActivity.kt	0
app/src/main/java/com/example/nexam/NexamApplication.kt	0
app/src/main/java/com/example/nexam/NexamViewModel.kt	0

Security hotspots of nEXam by SonarCloud

A benefit of the high amount of design patterns and as a part of the usage of Kotlin we already got here a low score by creating.

We think metrics are a great way to identify the parts of your project you should refactor. And these will be very interesting if you read a bit more and find out what you can do.

We hope could show you some interesting parts and will be happy to read your comments.

Your nEXam-Team



Published by uhrschatz

[View all posts by uhrschatz](#)

4 thoughts on “SE2 Week 8: Metrics”

Planit Se says:

June 13, 2022 at 12:09 pm | Edit

Hello Nexam,

the complexity of classes makes code-maintance after finish of development more difficult. That's why I think it's very good to get an impression of how complex your code is. As you stated, you were surprised by the evaluation of your tool.

Reducing current complexity might require major refactoring and can be very effortful. In future projekts I recommend to keep complexity as low as possible from the start.

Your PlanIt-Team

↳ Reply

Newtry says:

June 13, 2022 at 8:24 pm | Edit

Hello Team Nexam,

nice to see that you have no known vulnerabilities in your code. Also your complexity looks pretty good, as a number below 10 is considered a "Simple program, without much risk".

I look forward to seeing your final product.

Best regards Norman from Team Newtry

↳ Reply

uhrsch says:

June 17, 2022 at 10:36 pm | Edit

Hello Newtry,

thanks for your compliment.

But as we know in the meantime, SonarCloud only scans the code. The DependerBot of GitHub will also scan the dependencies where you will get the most vulnerabilities.

So the metric from SonarCloud isn't, reliable unfortunately

↳ Reply

P3lina says:

June 14, 2022 at 8:55 am | Edit

Hi Nexam-Team,

as Planit Se already said, of course it's better to keep cyclomatic complexity low right away but sometimes it's actually quite easy to 'drop some paths' to reduce complexity, just a question how important that's to you.

I look forward to seeing your final application, keep up the good work!

~Team Papatohu

↳ Reply



SE2 Week 10: Kickoff

👤 Franziska_B2 ⏰ June 16, 2022June 17, 2022 ⏳ 1 Minute

This week's blog post is about how you can install and use our app on your own Android phone. Since it stores the data locally, the installation is quite simple, just follow the steps.

First, you need your smartphone. Go to your mobile browser and go to our [GitHub](#) (<https://github.com/nEXam-App/nEXam/blob/main/app/release/app-release.apk>). Download the **app-release.apk** file.

After the download is finished, your smartphone might directly ask you if you want to install the application. If not, go to your downloads and click there on the file.

Then just follow the installation guide on your phone. Sometimes you need to configure your settings, so you can install an application which is not from the play store. It properly also asks you if you trust the download and the authors of the app, so please consider us as trustworthy.

If you're facing other difficulties while installing our great app, think about asking google if it knows a way for your specific smartphone. An example of a tutorial can be found [here](#) (https://praxistipps.chip.de/android-apk-installieren-so-gehts_13423). Otherwise, you can of course contact us as well, and we'll try it together!

Good luck with the installation and see you back with our last blog post for the final.
Greetings nEXam

Published by [Franziska_B2](#)



[View all posts by Franziska_B2](#)

One thought on “SE2 Week 10: Kickoff”

Lukas ~ DHBW Experts says:

June 18, 2022 at 10:11 pm | Edit

Hi there!

I really enjoyed reading through your second last blog post! Since your project is not dependent on a backend server, installation was pretty straightforward. I just went to your GitHub page with

my phone, downloaded the APK, pressed install and everything was ready to go. Nevertheless, there are two tiny things I want to mention. Firstly, your GitHub-APK link is outdated and secondly it would be great if you try to embed your APK file in GitHubs releases tab. That way all releases are in one central spot. Besides that! Well done!

Lukas ~ DHBW-Experts

↳ Reply

