About

Do, 05 Okt 2017 11:40:41, necoproject, []

This site shows the progress of our project, developed in the lecture Software Engineering I & II in the Coorperative School in Karlsruhe, Germany.

Contact

Do, 05 Okt 2017 11:40:40, necoproject, []

Contact:

- Mirko Müller: mueller.mirko@student.dhbw-karlsruhe.de
- Maurice Heumann: heumann.maurice@student.dhbw-karlsruhe.de
- Alexander Rengers: rengers.alexander@student.dhbw-karlsruhe.de

You like our project?

Leave us a comment and share your ideas to improve our app with us.

[contact-form][contact-field label="Name" type="name" required="1"/][contact-field label="E-Mail-Adresse" type="email" required="1"/][contact-field label="Website" type="url"/][contact-field label="Kommentar" type="textarea" required="1"/][/contact-form]

W1: NeCo Project - Vision

Do, 05 Okt 2017 11:40:39, necoproject, [category: allgemein]

We are a group of three students studying B. Sc. Computer Science at the *DHBW Karlsruhe*. During our lecture of Software Engineering we have to implement a program. On this website we will inform you about the progress of our project.

Our idea is to develop a Xamarin App for chatting with strangers around you.

The basic idea behind NeCo is to use localization technologies to create chatrooms with people around you, similar to the popular app "Jodel". The difference is that you chat with random people in your local area, to meet, to party, or just to have fun.

For security reasons we'll use latest encrypting technology like RSA&AES-Encrypting.

Ideas for future features could be:

- A friend list to keep in contact.

We will publish parts of our source code here and hope to get your comments to help us create our app the best way possible.

Thank you for reading our blog!

Dev Team

Mo, 09 Okt 2017 10:14:22, necoproject, []

The Team consists of 3 active devs:

Maurice Heumann, Alexander Rengers and Mirko Müller (see contact)

Maurice Heumann

- Front-End-Developer

Mirko Müller

- Server-Administrator
- Software Architect
- Test Manager

Alexander Rengers

- Project Manager
- Business Process Analyst
- Responsible for Blog Posts

For detailed responsibilities see our team roles.

W2: Team Roles & Technologies

Mo, 16 Okt 2017 10:28:51, necoproject, [category: allgemein]

Team Roles

Here are our Team Roles defined, according to the RUP Model.

Each role comes with certain responsibilities.

Front-End-Developer: Maurice Heumann

- Responsible for
  - Client development
  - UI/UX development

Server-Administrator: Mirko Müller

- Responsible for
  - Server- and Infrastructure security

Project Manager: Alexander Rengers

- Responsible for
  - Creating business cases
  - Planing, tracking and managing risks

Business Process Analyst: Alexander Rengers

- Responsible for
  - Discovering all business use cases

Software Architect: Mirko Müller

- Responsible for
  - Deciding on technologies for the whole solution

Test Manager: Mirko Müller

- Responsible for
  - Implementation of automated portions of the test design
  - Testexecution

Responsible for **Blog posts:** Alexander Rengers

Technologies

Our App is going to be a Xamarin App, developed using C#. Therefore we use the Visual Studio IDE for development and, Git for version control.

For continous integration we use Jenkins CI.

For UI-Tests we use the SpecFlow plugin vor VS.

For project management we're going to use Jira.

W3: SRS & OUCD

Mo, 23 Okt 2017 12:20:39, necoproject, [category: allgemein]

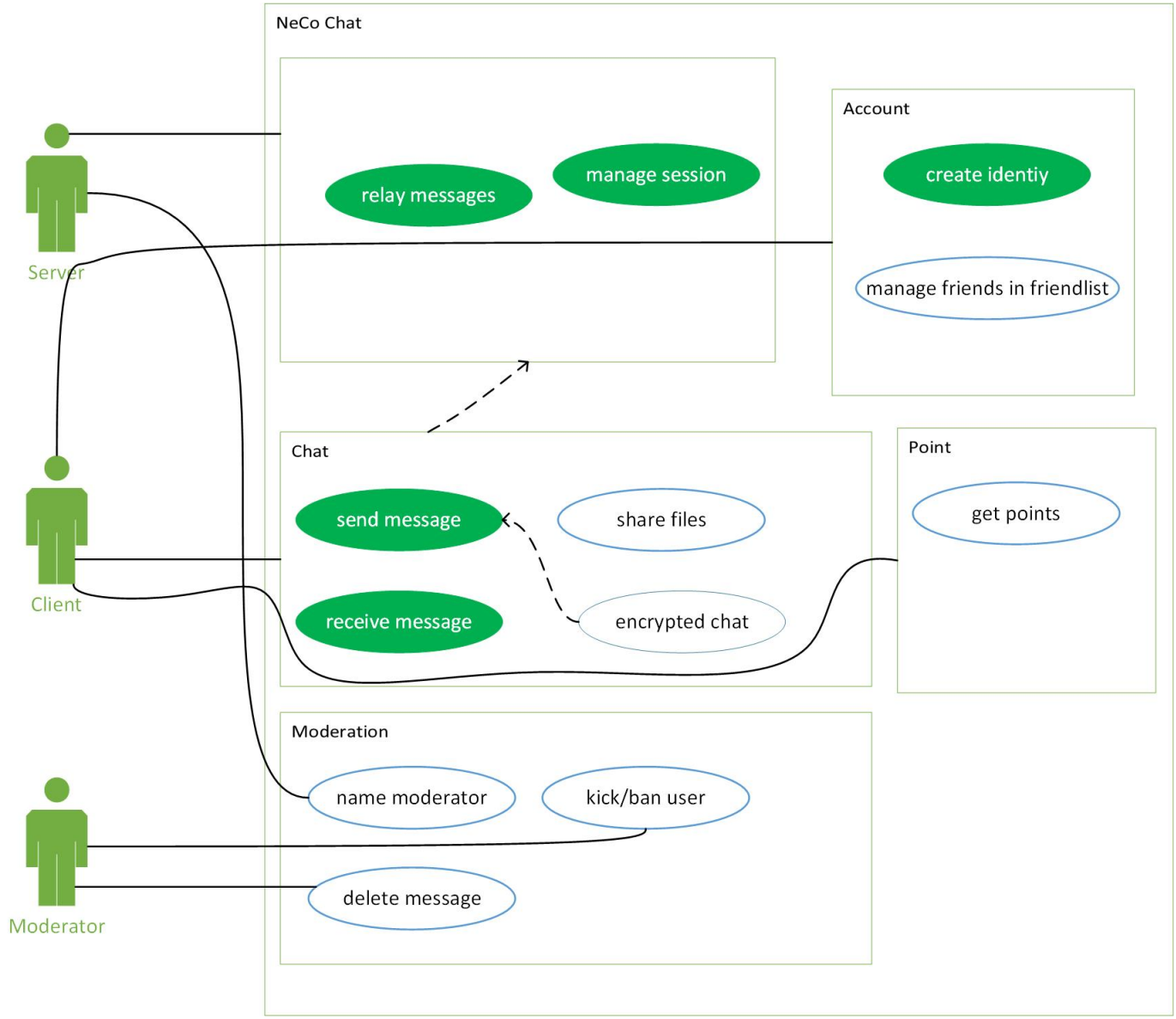
This week we started planing our Software Requirement Specification and our Overall Use Case Diagramm in detail. You can find the latest version on Github.

SRS:

https://github.com/Haus4/NeCo/blob/master/docs/SRS.md

OUCD:

https://github.com/Haus4/NeCo/blob/master/docs/UseCaseDiagramm.jpg



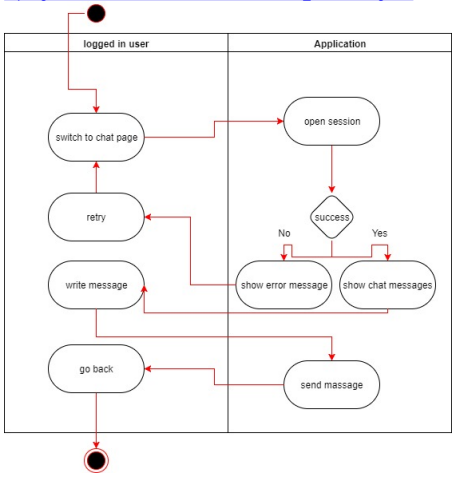
W4: 2 Use Cases 1 & 2

Mi, 01 Nov 2017 20:16:21, necoproject, [category: allgemein]

This week we started creating and specifing 2 Usecases. We also setup the first hello world page on our client and server. You can find the latest version on Github.

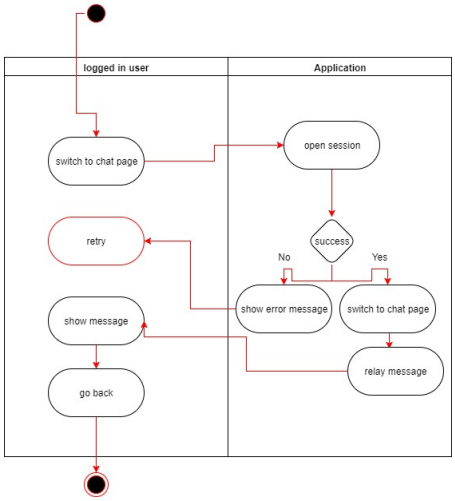
UC1: Send Message

https://github.com/Haus4/NeCo/blob/master/docs/UC1_SendMessage.md



UC2: Receive Message

https://github.com/Haus4/NeCo/blob/master/docs/UC2_ReceiveMessage.md



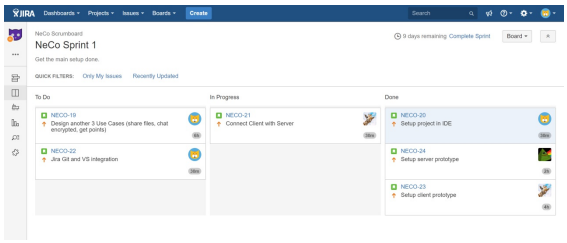
W5: Scrum & Prototype

Fr, 03 Nov 2017 12:04:19, necoproject, [category: allgemein]

This week we dealt with project management tools and scrumming.

For our project, we decided to use the tool Jira from Atlassian.

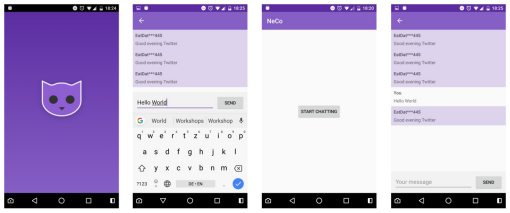
We've also created our first sprint , which you can see on our Jira scrumboard or on the screenshot below. Here you can see the actual status of our sprint.



The contains 5 basic tasks, assigned to a member of our team for processing.

Therefore we also defined our sprint length to 2 weeks.

We've also uploaded our first prototype of the NeCo App this week. You may look into it on our GitHub repository. The screenshot might give you a feeling, of how the app will look like.



You can find our prototype here

And our Scrumboard here

If you have any ideas or thougts on our process on the project, feel free to comment here.

W6: Gherkin .feature files

Di, 14 Nov 2017 15:46:27, necoproject, [category: allgemein]

Hi there,

this week we advanced a step further into Behavior Driven Development, therefore we wrote .feature files for some of our Use Cases.

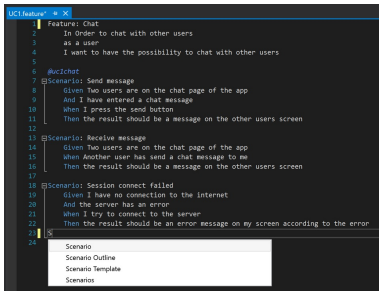
You can find those files right here:

1. Feature "Chat"
2. Feature "Chat Local"

Furthermore we added these feature files in our Use Case Specifications.

1. Use Case "Chat"
2. Use Case "Chat Local"

For a better syntax highlighting in Visual Studio we used the plugin "SpecFlow.VsIntegration". The screenshot below shows additionaly the autocompletion.



W7: Class Diagram

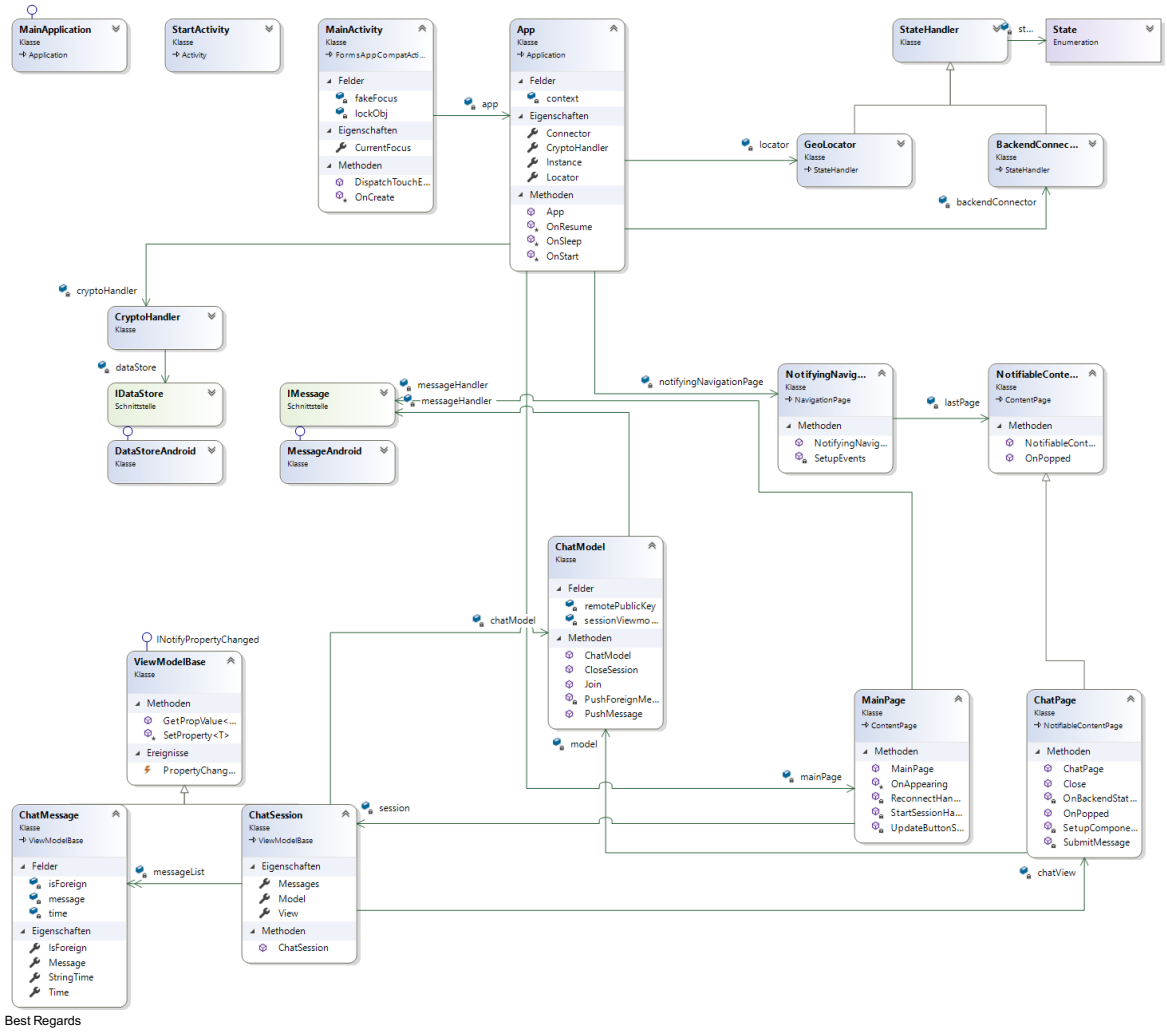
Di, 21 Nov 2017 12:20:33, necoproject, [category: allgemein]

Hi everyone,

this week we created a first version of our class diagrams.

The class diagram uses Visual Studios class diagram creation tools. First we created the C# code and then generated a diagram. We used as mentioned the Visual Studio toolkit for generating it automatically.
These diagram is a first versions, we will adapt and improve it while working at our project, so it will change in time. You can inform about the progress at our Software Architecture Document which also describes our project more in detail.

We also added our class diagram to our Software Architecture Document.



Best Regards

The NeCo Team

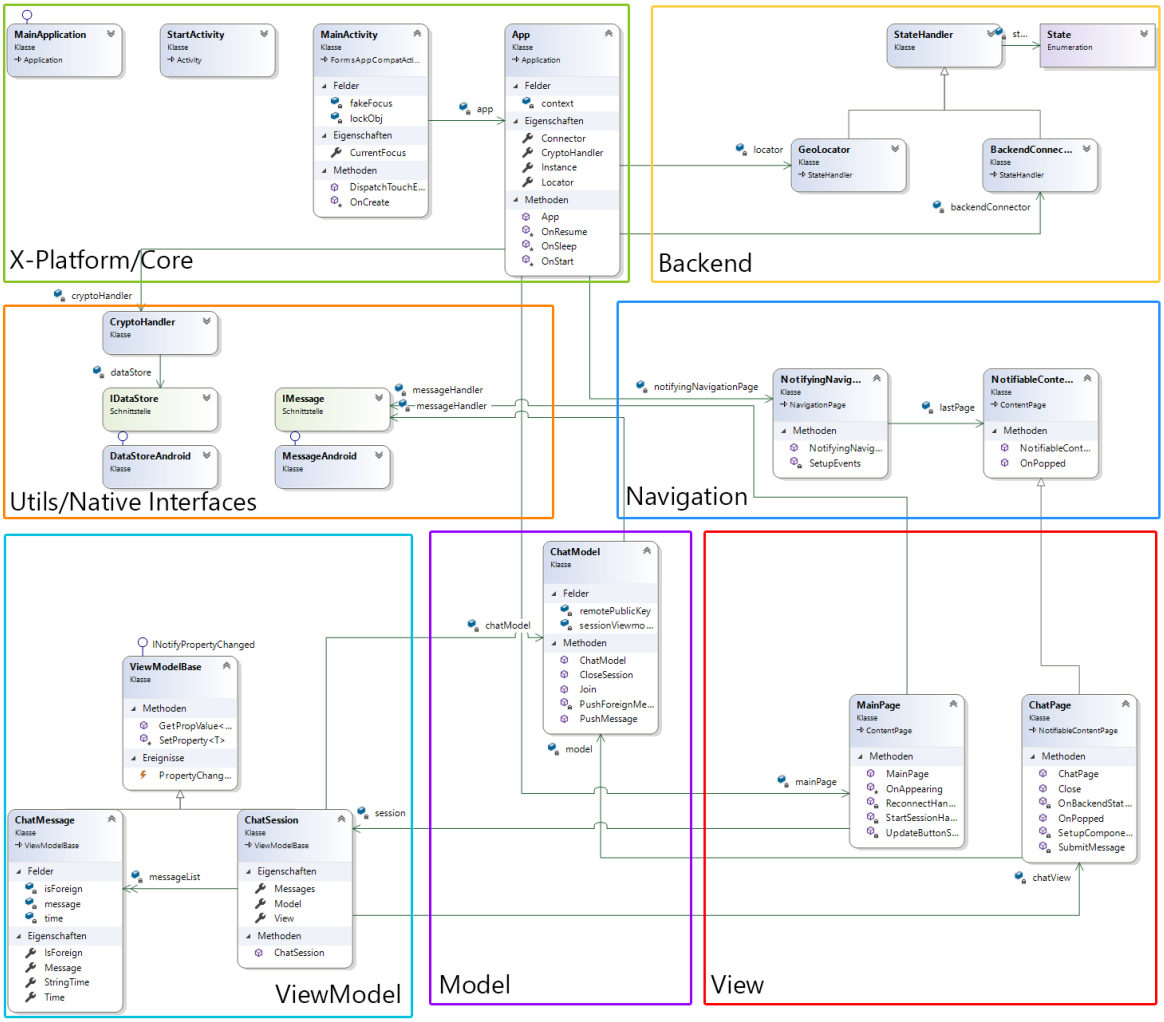
W8: Architecture

Di, 28 Nov 2017 14:25:16, necoproject, [category: allgemein]

Hi there,

Today we created the *Software Architecture Document* to give you an overview of our architecture model.

Here you can see our redrawed ClassDiagram where you can see which classes are the View, ViewModel, Model and X-Plattform/Core.



Best Regards

The NeCo Team

W9: Gantt Chart

Di, 05 Dez 2017 13:02:53, necoproject, [category: allgemein]

Hi there,

this week we worked on our Gantt Chart showing all the work we have done so far. Because we are using Jira for issue tracking, we could export the task list as a comma-separated values (CSV) file quite intuitive. However, we failed at importing this file into MS Project, as it doesn't recognizes the file's format.

Furthermore we started working with Jira quite at the mid of our projects first part, therefore we had to add tasks, that we did before we used Jira.

The export of the Gantt Chart was also quite unhandy, because fitting it into an A4 PDF file was possible, but you need to zoom in realy deep. So if you have MS Project installed, it is better to take a look at our original MPP file, which you can download here.

Otherwise, you can find the PDF version here on our Github, preferably you download it to be able to zoom in properly.

For us a Gantt Chart isn't quite good at telling us how long we spent on what task or phase. As we are not using any time tracking tool, therefore we depend on our Jira Sprints for time management. Logging work in Jira is for us a good overview of our workload. At the sprint overview you can also see how long you spent on an issue and what you were doing in the past, and are currently doing.

Best regards,
Team NeCo

NeCo App realeased in Play Store

Do, 07 Dez 2017 10:38:45, necoproject, [category: allgemein]

Hi everyone,

today we launched our App prototype on the Play Store.

If you are interested you can get it here:

Kind regards,

The NeCo Team

Midterm

Fr, 15 Dez 2017 13:11:12, necoproject, [category: allgemein]

Hey everyone,

this is the last post for this year. Here you can find our *Software Requirement Specification* and the *Software Architecture Document*.

Below you can see all the posts for each week:

Week 1: Vision
Week 2: Team Roles & Technologies
Week 3: SRS & OUCD
Week 4: Use Cases 1 & 2
Week 5: Scrum & Prototype
Week 6: Gherkin .feature files
Week 7: Class Diagram
Week 8: Architecture
Week 9: Gantt Chart

Here you can find a link to the midterm presentation.

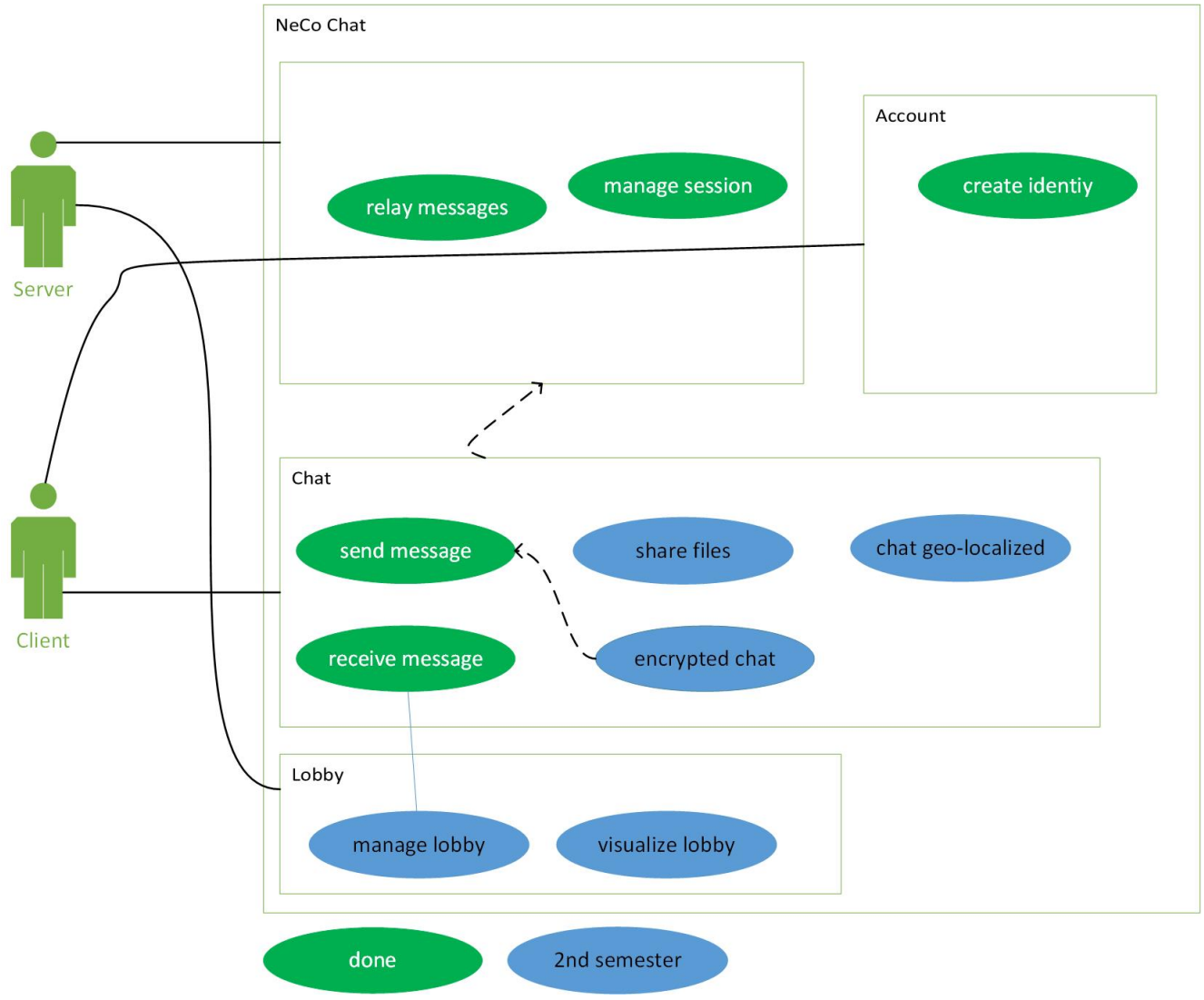
2nd Semester - W1

Mi, 04 Apr 2018 09:27:32, necoproject, [category: allgemein]

Hi everyone,

for the first week of the new semester we revisited our code and documentation and set all up, to work further on our project.

We changed our scope to new requirements and updated our Overall Use Case Diagramm:



We still rely on our well know technologies.

You can see how many hours we spend on each use case and how many LOC are written per use case, in the following. All values are estimated, using our Jira issues and our GitHub commits:

- Relay messages
  - 8h / 1 PD
  - 250 LOC
- Manage session
  - 16h / 2 PD
  - 150 LOC
- Create identity
  - 8h / 1 PD
  - 200 LOC
- Send message
  - 16h / 2 PD

    - 1000 LOC

- Receive message
    - 16h / 2 PD
    - 1000 LOC

Fortunately our team consists of the same team members as last semester :)

W2 - Scope & Risk Management

Mi, 11 Apr 2018 10:35:08, necoproject, [category: allgemein]

Hi there,

this week, we digged further into project management, defining our scope more detailed and identifying risks of this project.

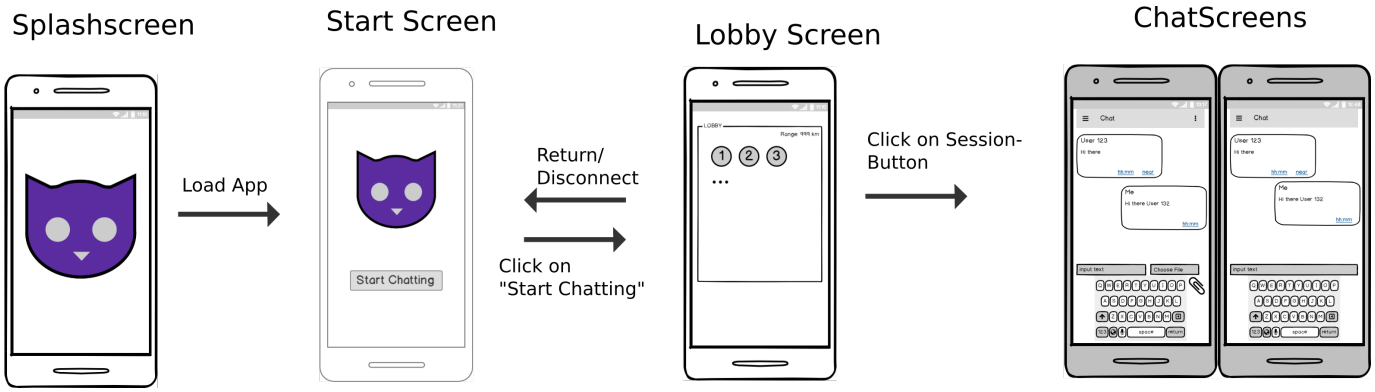
- You can find our **risk management** table here.

identified	Description	prob of occurence (in %)	damage/impact (1-10)	mitigation strategy	person in charge	risk factor (0-10)
showstopper at critical time	changes in code lead to an fatal error, before presentation	0,4	7	splitting branches in master, develop (Git Workflow); regular testing in the end; feature freeze	Maurice	2,8
server shutdown	shutdown due to failure/updates	0,3	7	update freeze before endpresentation; backups; docker	Mirko	2,1
underestimated workload	to much workload of this course regarding the workload of other courses	0,4	5	flexible workload management	Alex	2
loss of data	due to device failure;	0,1	10	code management using GitHub; track code-sharing	Alex	1
illness	fatal	0,3	2	code sharing; communication via WhatsApp; regular commits to feature branches		0,6
loss of team members	loss of team members due to exmatriculation	0,02	8	share knowledge via GitHub; document detailed	Alex	0,16
						0
						0
						0

- We made a **time estimation** for our use cases done in the last semester, which you find here here.

UC	Documentation	Coding	Testing	Total	Warm-Up Time FP
relay messages	2h	12h	5h	20h	5h
manage session	1h	10h	1h	12h	5h
create identity	1h	6h	1h	8h	3h
send message	2h	10h	3h	15h	8h
receive message	2h	10h	2h	14h	6h
share files	1h				
chat geo-localized	1h				
encrypted chat	1h				
manage lobby	1h				
visualize lobby	1h				

This is the **screenflow** of our application, showing the flow using our application:



For an overall overview you can visit our first post of this semester, where you find our overall use case diagramm.

Best regards,
Team NeCo

W3 - FUNCTION POINT ESTIMATION

Mi, 02 Mai 2018 08:38:00, necoproject, [category: allgemein]

Hello!

Today we want to show you our function point estimation.

First of all we will give you a little explanation about function points.

Function points are used for estimating the time that will be spent on a certain Use Case. They are calculated in reference on External Inputs, Outputs, Inquiries as well as Internal and External Logical Files from the User's view, thereby they do not depend on the used technology.
To estimate how long it will take to implement a Use Case, function points of UC that have been already implemented are put in relation to the time we spent on it.
We used the calculation for our function points from „TINY TOOL„.

Now to the actual work we done.

The following table shows our completed Use Cases with the time we needed to implement. You can also see the at the beginning explained references we calculated the function points on.

use case	esitmation (h)	logged (h)	transaction	DET	RET	FTR	Complexity	function points
relay messages	20h		external input	0			low	
			external output	2			low	
			external inquieries	0			low	
			internal logical files		7		avg	
			external interface files	0			low	81,9
manage session	12h		external input	0			low	
			external output	1			low	
			external inquieries	1			low	
			internal logical files			3	low	
			external interface files			1	low	34,6
create identity	8h		external input	0			low	
			external output		1		low	
			external inquieries	0			low	
			internal logical files	1			low	
			external interface files	1			low	16,8
send message	15h		external input	1	1		low	
			external output		1		low	
			external inquieries	0			low	
			internal logical files			3	low	
			external interface files			3	low	48,3
receive message	14h		external input	0			low	
			external output	1	1		low	
			external inquieries	0			low	
			internal logical files			3	low	
			external interface files			3	low	46,2
share files			external input	1			low	
			external output	1			low	
			external inquieries	0			low	
			internal logical files			1	low	
			external interface files			1	low	19,9
chat geo-localized			external input	1			low	
			external output	1			low	
			external inquieries	0			low	
			internal logical files	1		1	low	
			external interface files	1			low	27,3
encrypted chat			external input	0			low	
			external output	0			low	
			external inquieries	0			low	
			internal logical files	1		3	low	
			external interface files	1		1	low	34,6
manage lobby			external input	1			low	
			external output	0			low	
			external inquieries	0			low	
			internal logical files			3	low	
			external interface files			4	low	46,6
visualize lobby			external input	0			low	
			external output	1			low	
			external inquieries	1			low	
			internal logical files			3	low	
			external interface files			1	low	34,9

The acronyms DET, RET and FTR mean Data Element Type, Record Element Type and File Type Reference. Through these we identified the complexity which can be low, average or complex.
Here you can find the whole document which also includes the data for the new Use Cases. In this document you also see the estimation for the new Use Cases these get explained later in this blog entry.

For the calculation of function points with TINY TOOL you need some extra information about the whole application.

Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE No Influence 0	1	2	3	4	Essential 5
1	Does the system require reliable backup and recovery?	◉	○	○	○	○	○
2	Are data communications required?	○	○	○	○	○	◉
3	Are there distributed processing functions?	○	○	○	○	◉	○
4	Is performance critical?	○	○	○	◉	○	○
5	Will the system run in an existing, heavily utilized operational environment?	○	○	○	○	◉	○
6	Does the system require on-line data entry?	○	○	○	○	◉	○
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	○	○	○	◉	○	○
8	Are the master files updated on-line?	○	○	○	◉	○	○
9	Are the inputs, outputs, files or inquiries complex?	○	○	○	○	◉	○
10	Is the internal processing complex?	○	○	○	◉	○	○
11	Is the code to be designed reusable?	○	◉	○	○	○	○
12	Are conversion and installation included in the design?	○	◉	○	○	○	○
13	Is the system designed for multiple installations in different organizations?	○	○	○	○	○	◉
14	Is the application designed to facilitate change and ease of use by the user?	○	◉	○	○	○	○

Domain Characteristic Table | FP Calculation

In this screenshot you can see what we choose for our project.

We calculated the function points for all our Use Cases. For example in the next picture you can see the table for the Use Case „Relay Message".

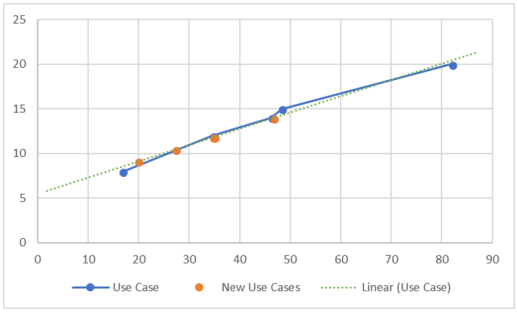
MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input		◉	○	○
Number of User Outputs	2	◉	○	○
Number of User Inquiries	0	◉	○	○
Number of Files	7	○	◉	○
Number of External Interfaces	0	◉	○	○

Domain Characteristic Table

This Use Case has 81,9 function points. Here you can see the whole document.

With the calculated function points we generated a diagram which shows the interact of the function points and the person hours.

FP calculation and time estimation diagram



We used this graph to estimate the time for our remaining Use Cases. In the graph you can see these as orange dots. You can only see four dots, because two Use Cases („encrypted chat" and „visualize lobby") have the same amount of function points. Below you can see our time estimation for our new use cases.

New Use Cases		
share files	19,9	9,11845
chat geo-localized	27,3	10,46895
encrypted chat	34,6	11,8012
manage lobby	46,6	13,9912
visualize lobby	34,9	11,85595

I hope you liked it and if you want give us some feedback.

Greetings,

NeCo

W4 - Unit Testing

Mi, 16 Mai 2018 08:41:50, necoproject, [category: allgemein]

Hi everyone,

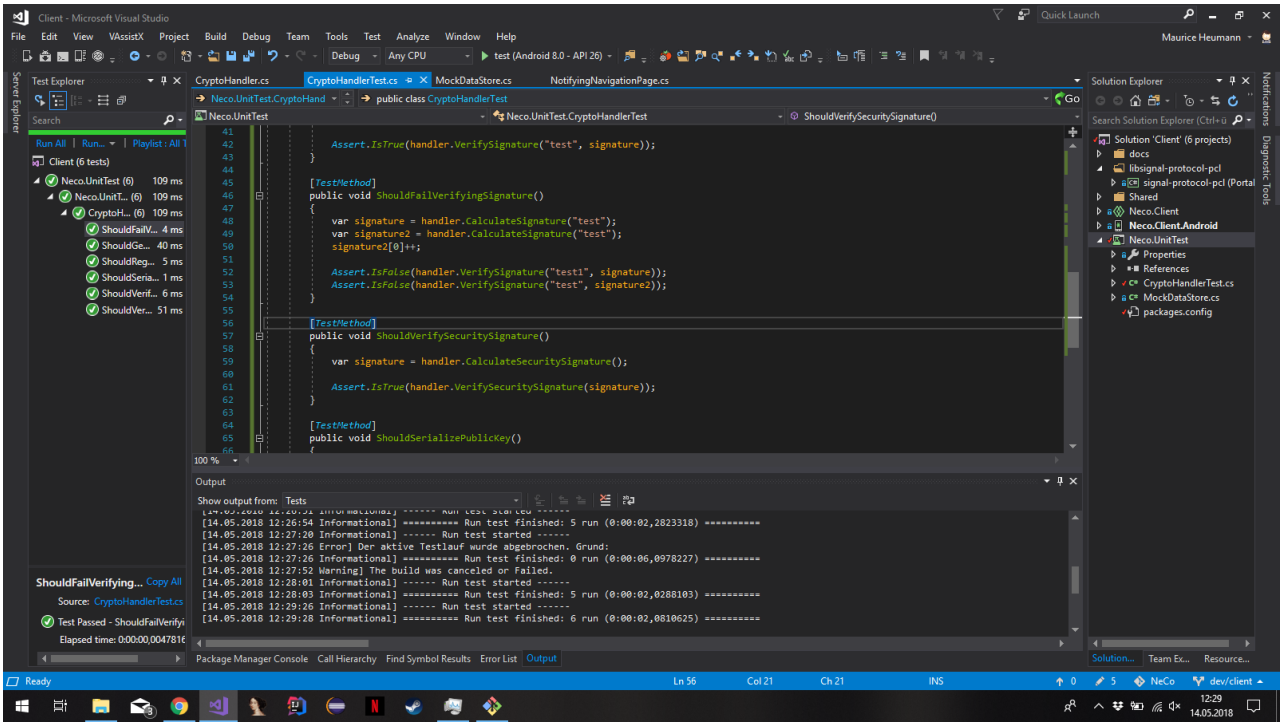
this week, we created our test plan

this week's topic was testing our project. So we started writing tests for our application improving our applications by finding bugs in early stage of development. Therefore we first wrote a Testplan which you can find here. It's going to be completed with all information needed in the future, while we work more on testing our project.

We use the MSTest framework for C# and it has been added as dependency to our project here.

A link to our tests can be found here.

For your overview, in the screenshot below, you can see the tests running in our IDE.



We're looking forward to your feedback

Best regards,

your Team Neco

W5 - Refactoring

Di, 22 Mai 2018 10:24:27, necoproject, [category: allgemein]

This week we learned more about refactoring. We studied the introduction to refactoring from Fowler, you can learn about it here. For learning purposes each of us trained the idea of refactoring on some exercise project.

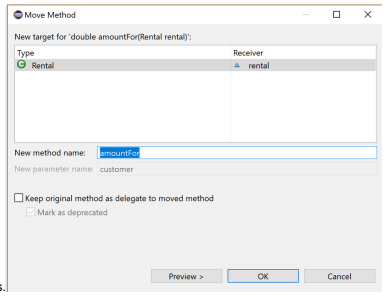
You can find each of our projects here on our team member's GitHub Account.

Maurice's GitHub

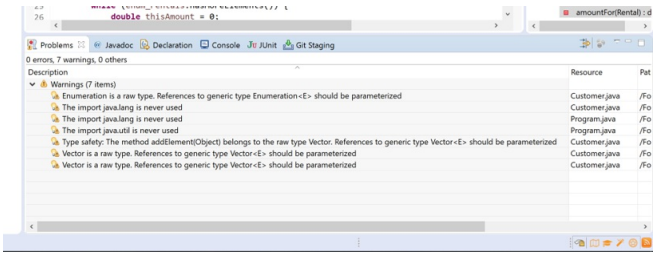
Mirko's GitHub

Alex' GitHub

In the screenshot below you can see, how Eclipse is helping us to refactor, first you can automatically move methods.



Furthermore Eclipse shows warnings like unused imports etc. while you code, so you might inspect these warnings early on.



Best Regards,

Team Neco

W6 - Design Pattern

Di, 29 Mai 2018 08:27:21, necoproject, [category: allgemein]

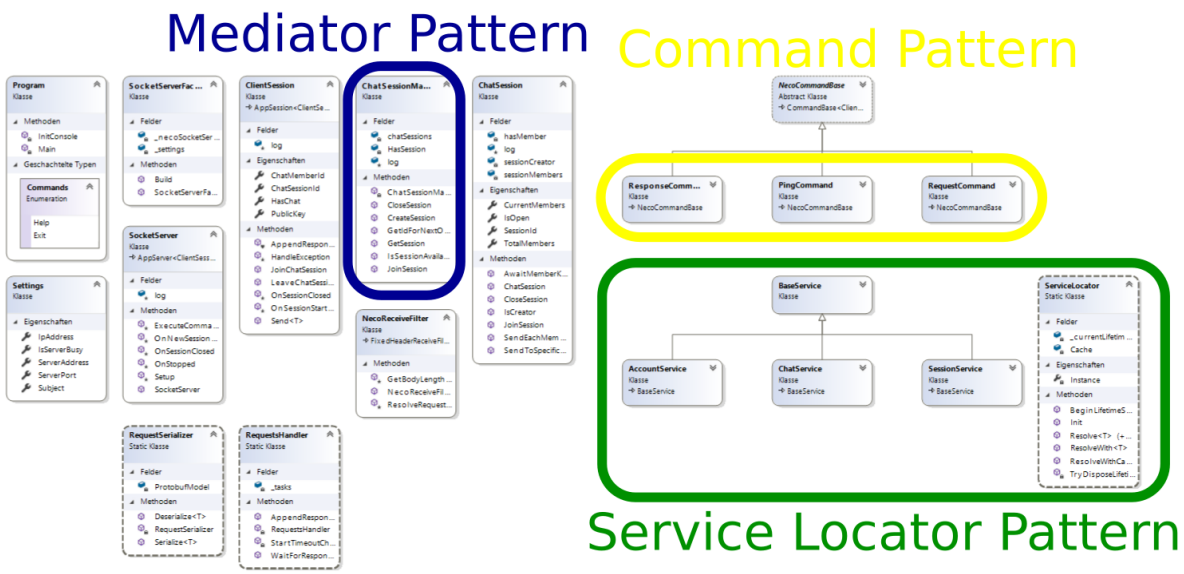
Hi everyone,

this week we diged further into refactoring and implemented some design patterns.

We've choosen a service locator pattern to be used on our server. This pattern is typical for c#.

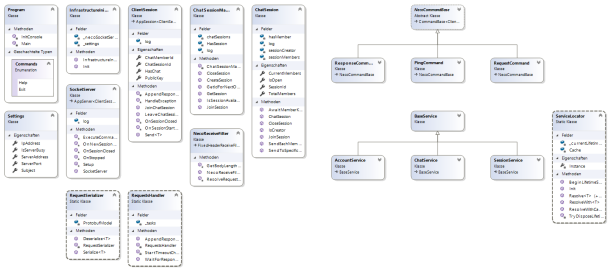
Therefore we rebuild our InfrastructureInitializer to a SocketServerFactory.

As we already used some patterns since the beginning of development, we decided not to compromise the code integrity completely. Therefore not realy much changed in our code.



The picture above unfolds our pattern used.

The pictures below show our class diagramm before we implemented the pattern



Kind regards,

Team Neco

W8 - Installation Test

Di, 12 Jun 2018 13:16:23, necoproject, [category: allgemein]

This week we published our app on the playstore. Therefore we needed a google developer account which we furtunately had. If you want to install it on your own Android device you can follow the link below.

On the playstore it is published to everyone and got already downloads. We also tested the successfull installation with some of our friends and family.

For a more detailed insight we propose you to take a short survey, which we made for us to know, how everything works and if there are any problems.

You can find the survey here, thanks for participating.

The results of the test can be inspected here.

W7 - Metrics

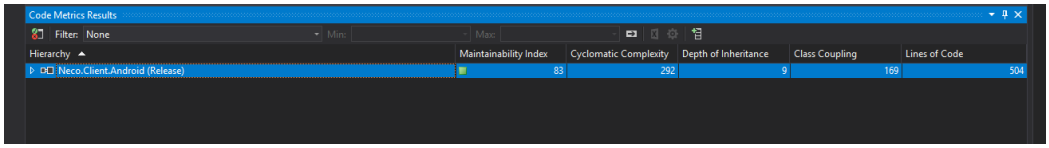
Mi, 06 Jun 2018 08:40:30, necoproject, [category: allgemein]

Hi,

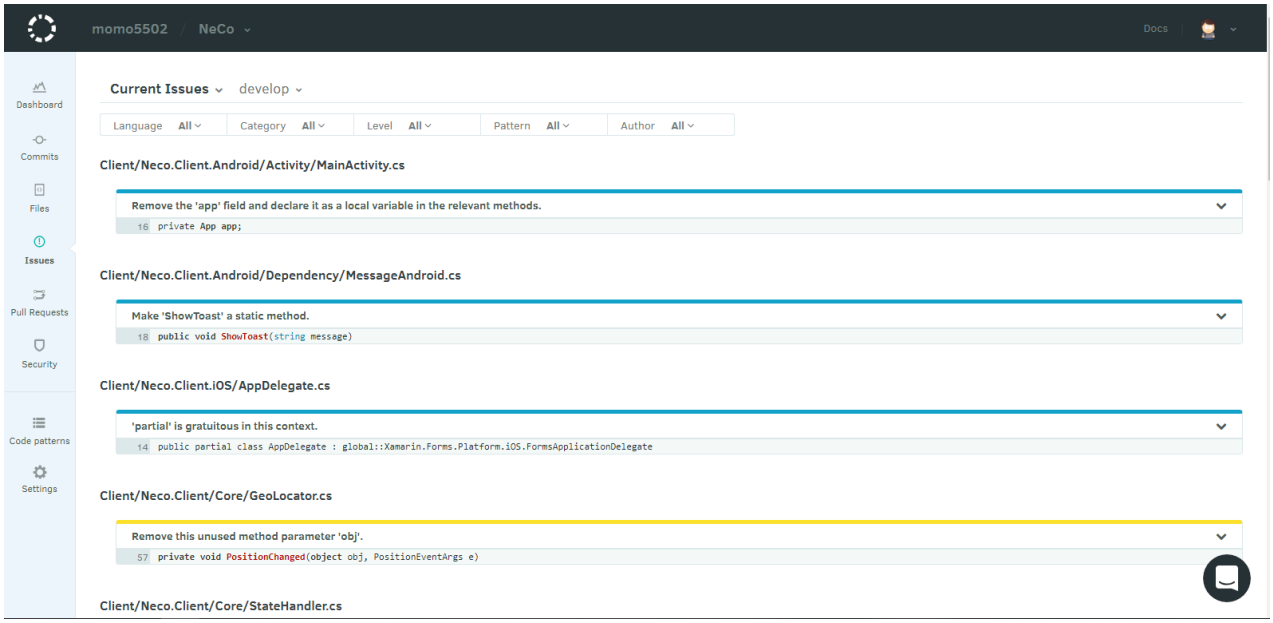
this weeke we looked closer at metrics. We used the metrics tool inside our IDE Visual Studio 17 and therefore got some detailed insights into our code.
Getting it to work for Xamarin took a bit of time and effort, due to the different architectures.

It calculates several metrics:

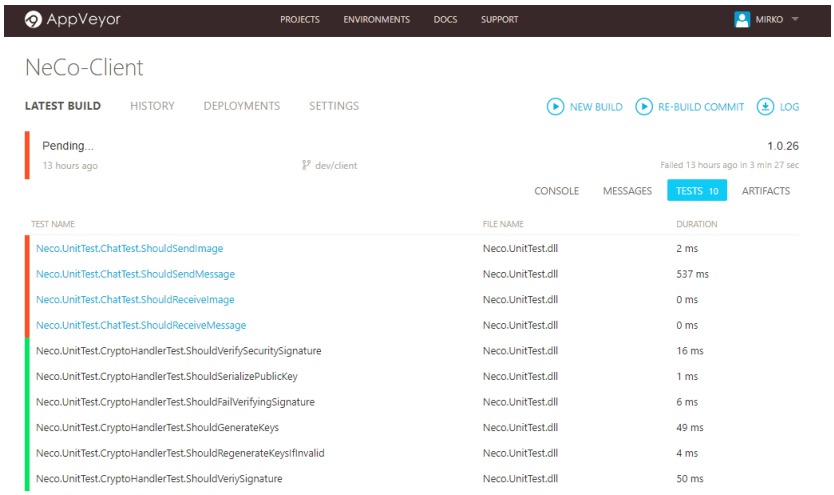
- Cyclomatic complexity
- Depth of inheritance
- Class coupling
- Maintainability index
- Lines of code



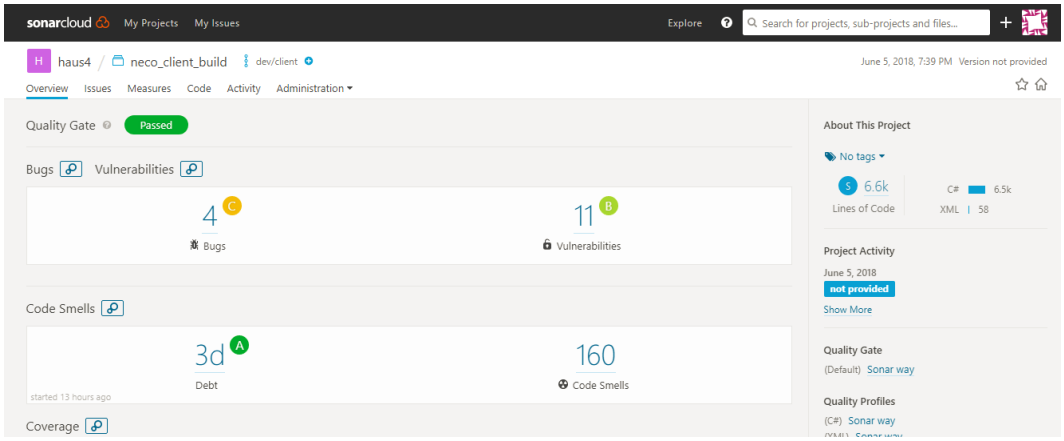
Additionally, we integrated codacy to our project which shows us additional issues and points of optimization:



Also here is a picture of our build-server running our tests:



Finally we also added sonarqube to our project to track bugs/codesmells and coverage:



Kind regards,

Team Neco

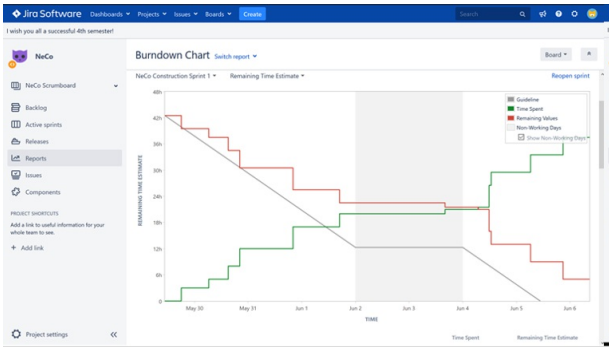
W9 - Final Exam

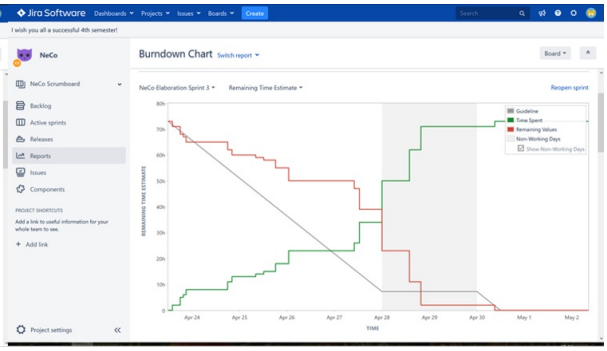
Mi, 20 Jun 2018 09:00:13, necoproject, [category: allgemein]

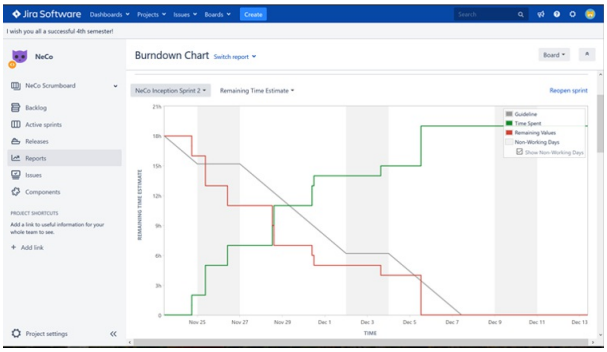
This week we have our final exam, therefore this is going to be our last post on this blog for this second semester.

So finaly we can take a short break and resume what we have done.

First we want to show three clean sprints, which we made this semester, using Jira







Kind regards,

Team NeCo