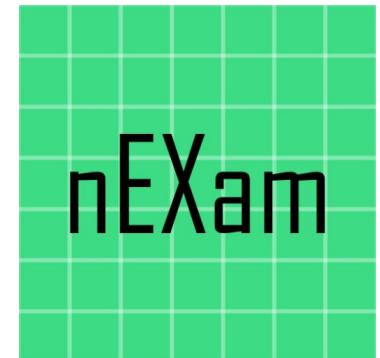


nEXam final presentation

by Franziska Ommer and Simon Reichle

Contents

- Our Vision
- Business plan
 - Use Case overview
 - Project Management
 - Rational Unified Process
 - Risk Management
 - Cost Estimation
 - Time spent
- Requirements
 - Environment
 - Architecture
 - Technologies
 - Testing & Metrics
 - Deployment
- Development
 - Class diagrams
 - Pattern implementation
 - Database Implementation
 - Backend
 - Frontend
- Live Demo



Our vision



Simple app to support learning



Better overview through sorting of upcoming exams



Suggestion of time to invest, including timer



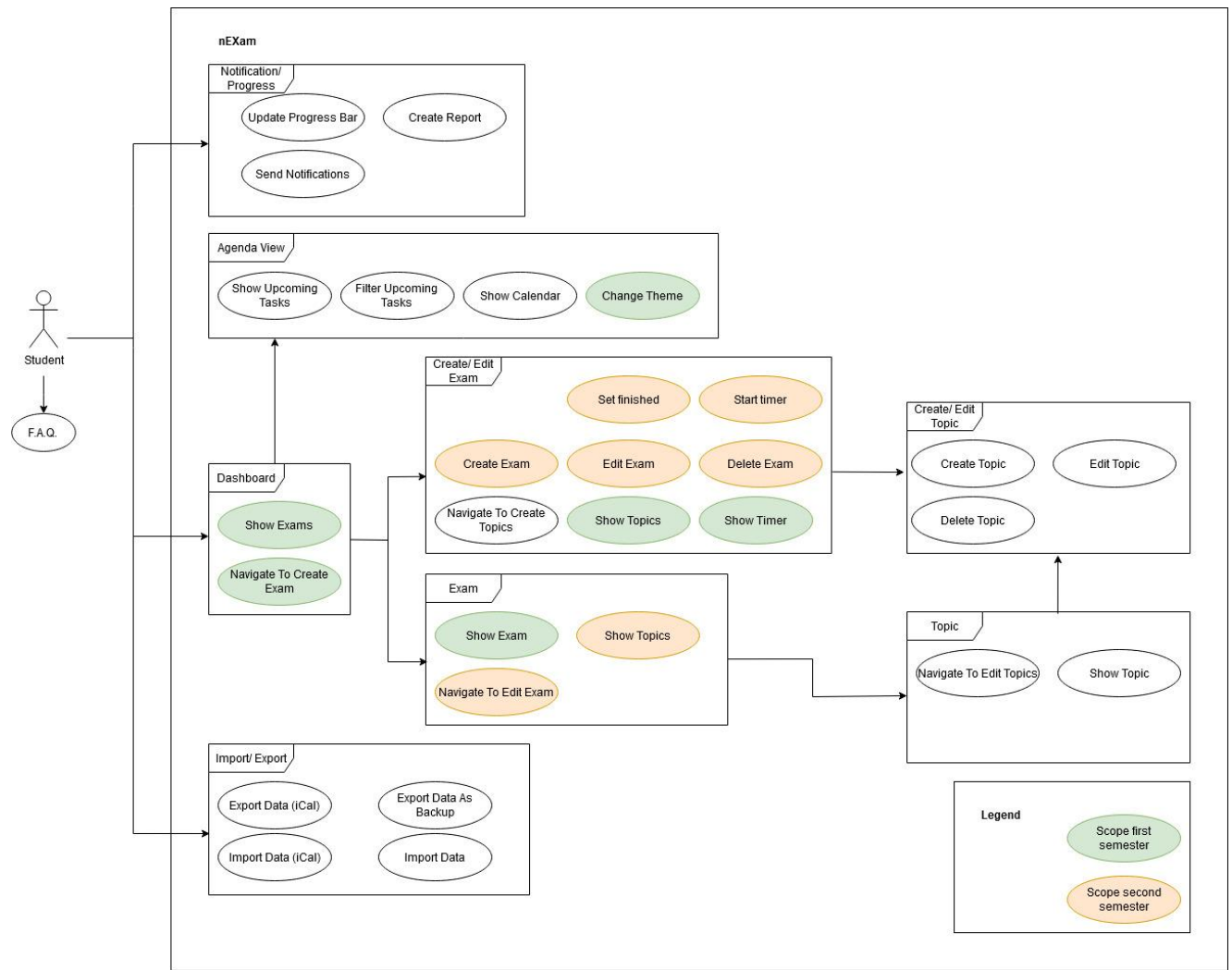
Reduces stress through coordinated learning



Many more great features in the future!

Business plan

Use Case overview



Project Management

YouTrack

SCRUM with one-week sprints

Weekly sprint meetings

Issues for every task with the necessary information

Time tracking

Git workflow

Automatic analysis with SonarCloud

Proper risk management



Project Management in



Sprint management

Detailed tasks

Development state and RUP workflow

Use of swim lanes for better overview

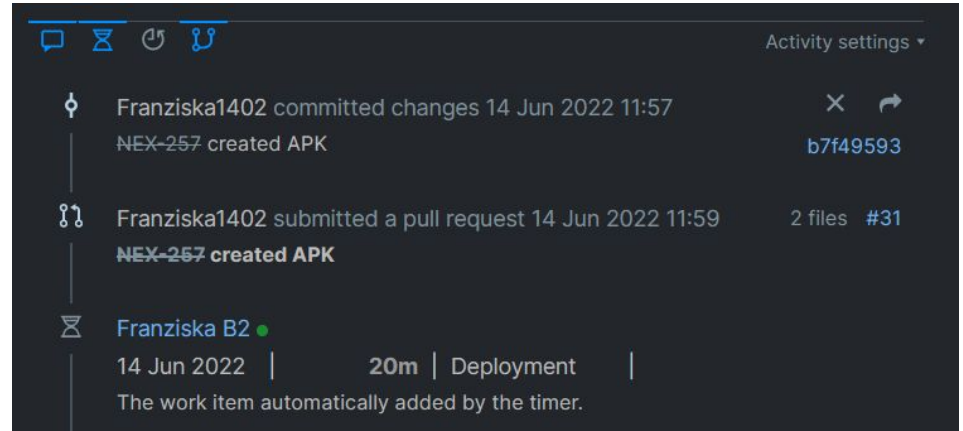
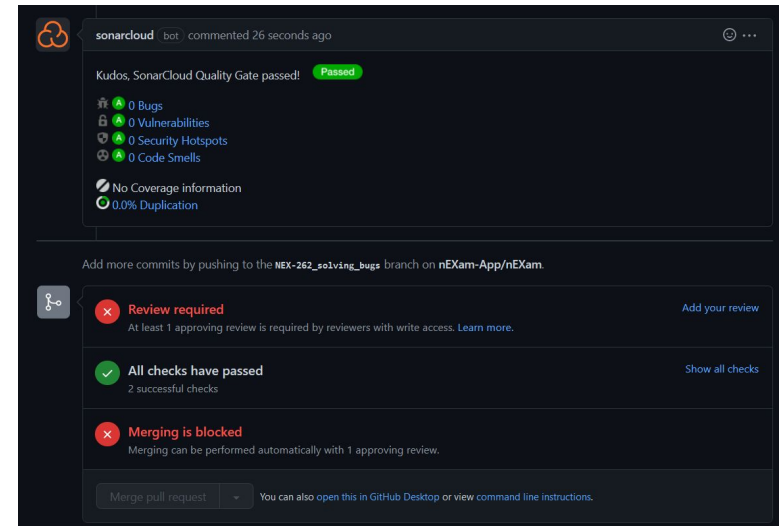
Time tracking with timer

IDE Plugin for issues

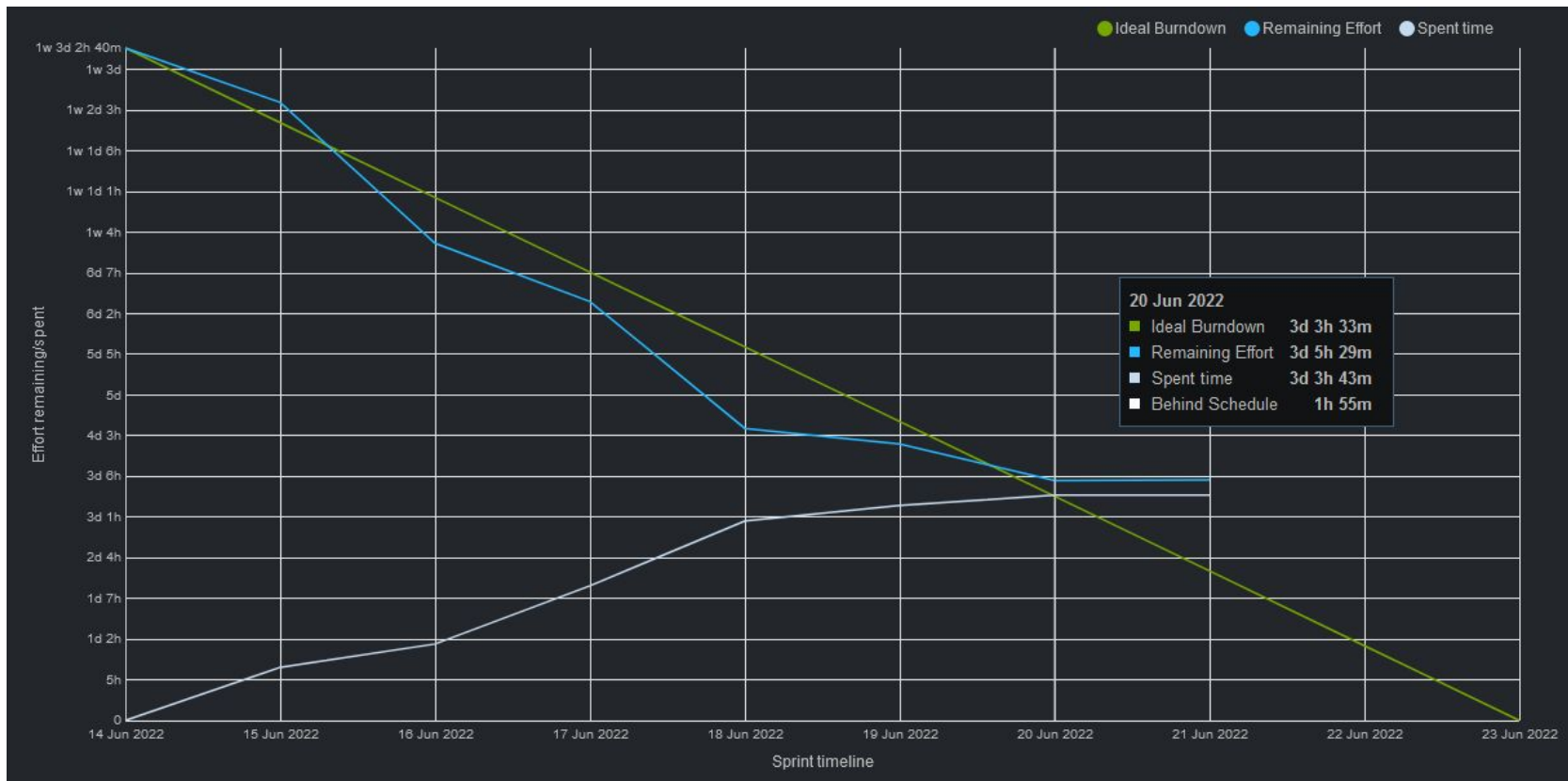
Backlog

GitHub

- Integration with YouTrack
- Integration with SonarCloud
- Organization with repos for documentation and application
- Application main branch is protected



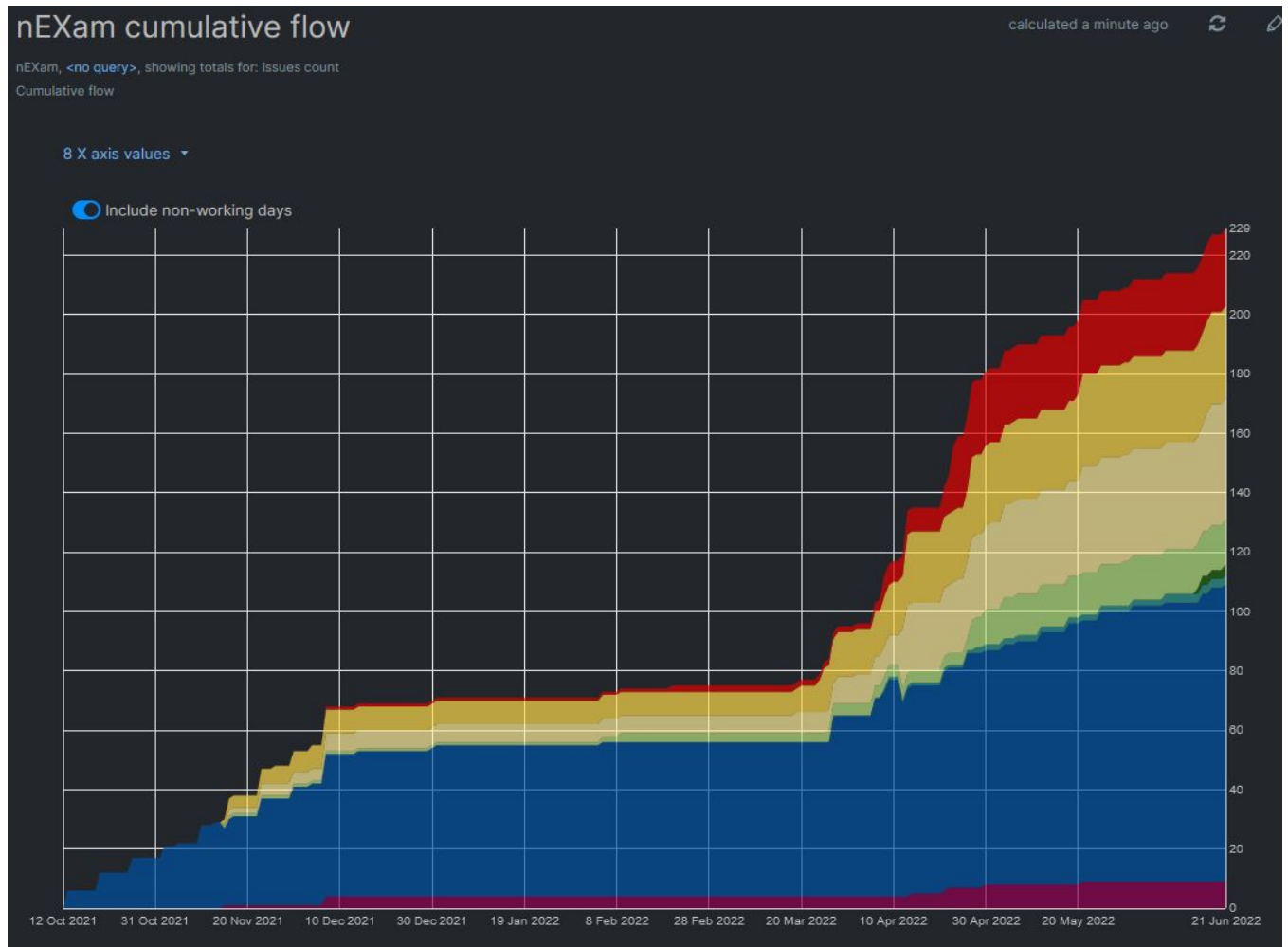
Burndown Chart



RUP roles

Member	Role
Franziska	Deployment, Frontend, Design, Testing
Simon	Backend, Database, Deployment

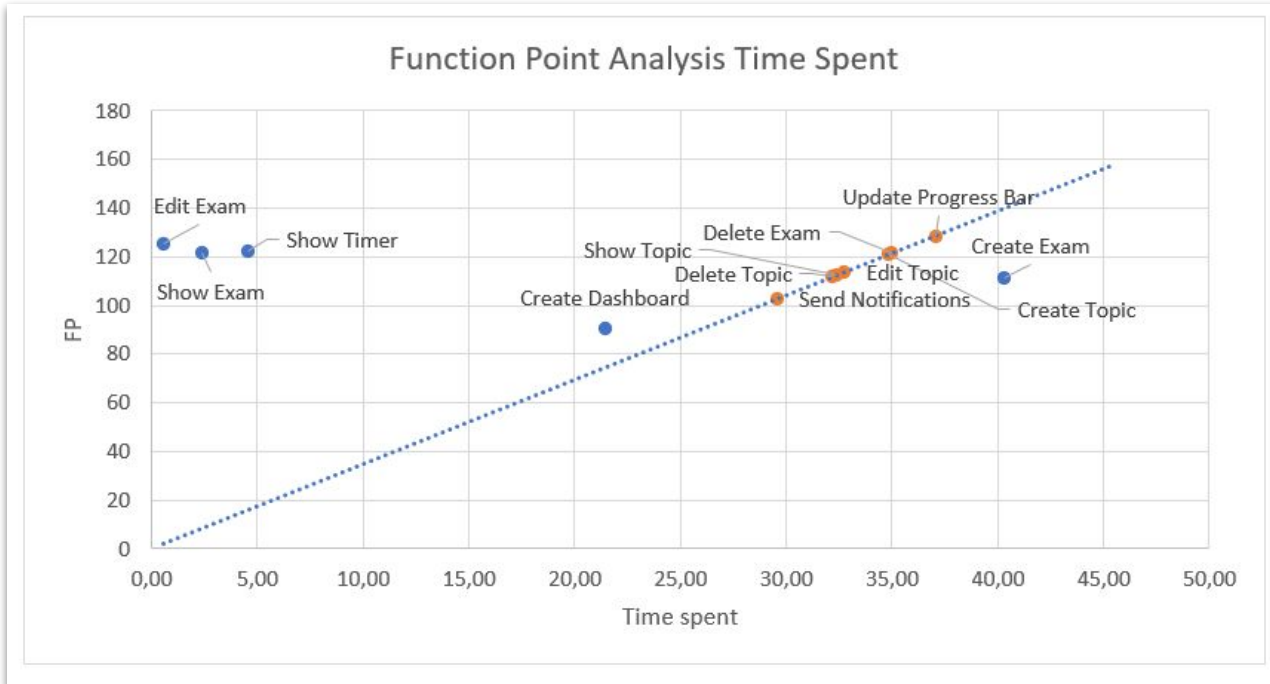
RUP workflow



Risk Management

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

Cost estimation (Function Points)



Estimation

Time first semester: 230



Time second semester: 258


Real time spent

Time first semester: 172

Time second semester: 167

Time spent by Use Case

nEXam edit topic		1h 10m	12m	58m	12m
NEX nEXam		1h 10m	12m	58m	12m
nEXam exam error		5h 00m	2h 57m	2h 03m	2h 57m
NEX nEXam		5h 00m	2h 57m	2h 03m	2h 57m
nEXam exam success		2h 30m	1h 03m	1h 27m	1h 03m
NEX nEXam		2h 30m	1h 03m	1h 27m	1h 03m
nEXam exam_view		7h 30m	5h 15m	2h 15m	5h 15m
NEX nEXam		7h 30m	5h 15m	2h 15m	5h 15m
nEXam progressbar		8h 00m	5h 16m	2h 44m	5h 16m
NEX nEXam		8h 00m	5h 16m	2h 44m	5h 16m
nEXam send notifications		5h 30m	1h 53m	3h 37m	1h 53m
NEX nEXam		5h 30m	1h 53m	3h 37m	1h 53m
nEXam show timer		7h 00m	5h 07m	1h 53m	5h 07m
NEX nEXam		7h 00m	5h 07m	1h 53m	5h 07m
nEXam show topic		1h 00m	31m	29m	31m
NEX nEXam		1h 00m	31m	29m	31m

projects, grouped by Use case	Time estimated	Time spent	Blog-and-Peer-Review	Deployment	Development	Environment	Learning-New-Technology	No type	Project Management	Requirements-Documentation	Testing
Total time	486h 35m	338h 09m	14h 17m	3h 58m	143h 51m	11h 52m	2h 53m	92h 47m	5h 24m	48h 57m	14h 10m
No use case	284h 25m	195h 44m	14h 17m	2h 34m	18h 42m	11h 52m	2h 13m	90h 58m	5h 24m	39h 39m	10h 05m
NEX nEXam 	284h 25m	195h 44m	14h 17m	2h 34m	18h 42m	11h 52m	2h 13m	90h 58m	5h 24m	39h 39m	10h 05m

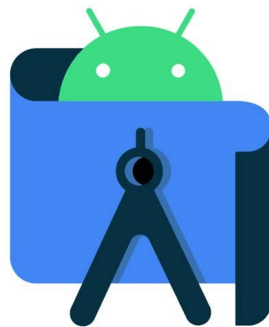
Requirements

Environment

- Android Studio
- Kotlin
- Gradle
- Android Native
- Room DB
- JUnit
- Cucumber
- Espresso with Green Coffee

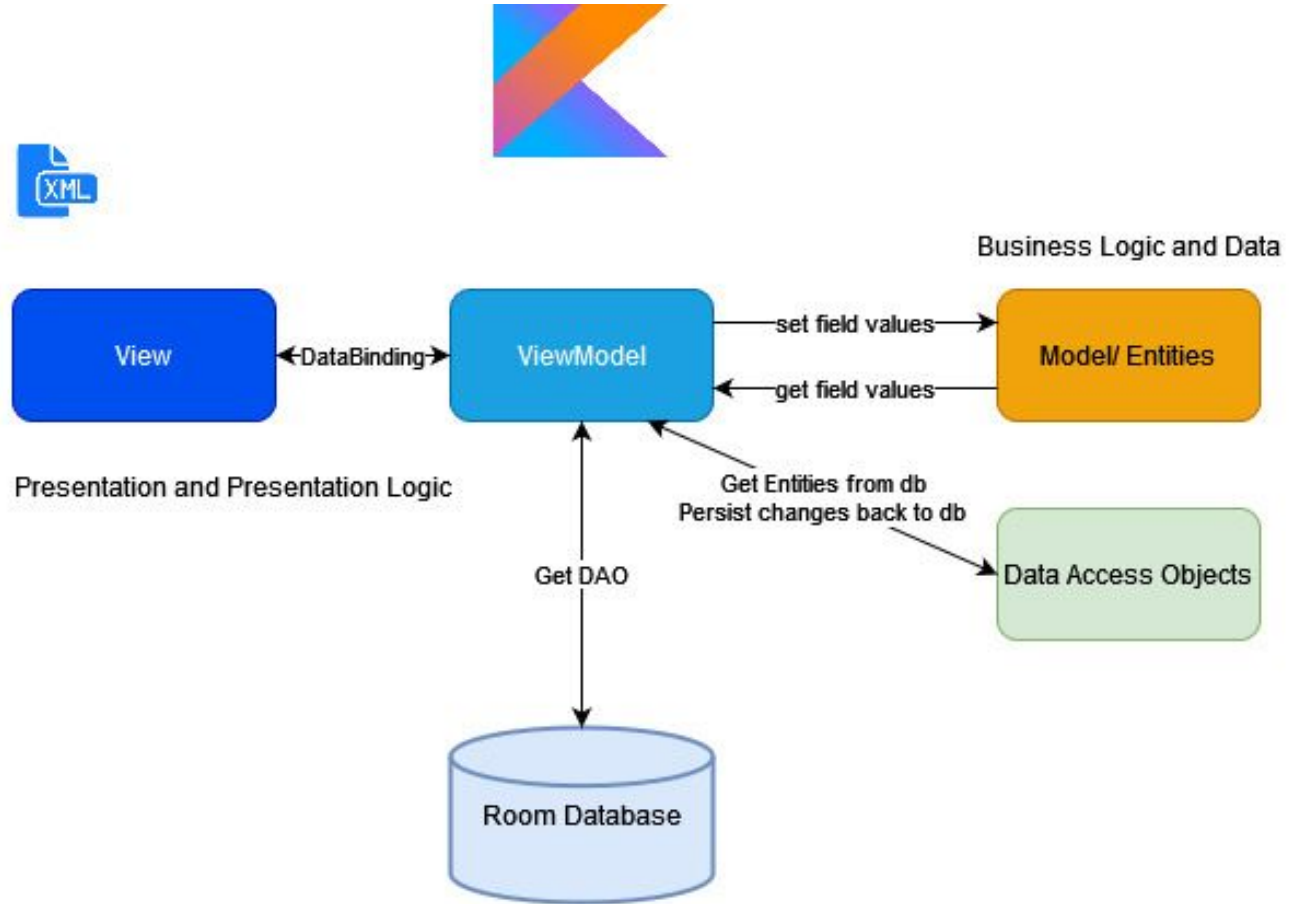


JUnit



Architecture

- Frontend
 - Kotlin
 - XML
- Backend
 - Room Database
 - ViewModel/ Model



Technologies

- Frontend
 - XML
 - Kotlin
- Backend:
 - Room Database
 - Gradle
- Metrics:
 - SonarCloud
- Deployment
 - APK file
- Testing:
 - Cucumber
 - JUnit
 - Espresso with Green Coffee

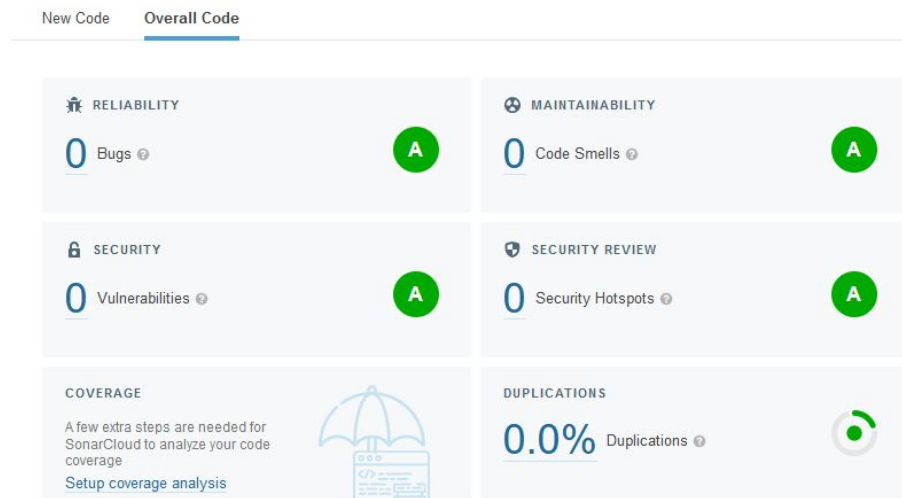
sonarcloud 



JUnit



Testing & Metrics



Example result of SonarCloud analysis

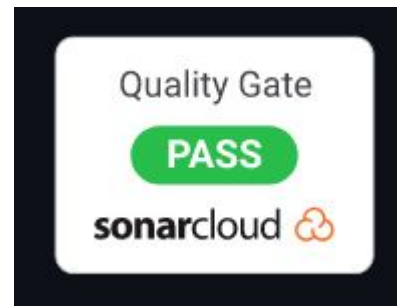
Run: app x AddExamFragmentTest x

Status: 6 passed | 6 tests, 12 s 159 ms

Filter tests: [Icons]

Tests	Duration	HTC HTC U11
✓ Test Results	432 ms	6/6
✓ AddExamFragmentTest	101 ms	2/2
✓ isValid_ShouldReturnFalse	76 ms	✓
✓ isValid_ShouldReturnTrue	25 ms	✓
✓ ExampleInstrumentedTest	25 ms	1/1
✓ useAppContext	25 ms	✓
✓ SimpleEntityReadWriteTest	306 ms	3/3
✓ updateExam	126 ms	✓
✓ addAndReadExam	102 ms	✓
✓ deleteExam	78 ms	✓

Example results JUnit/ Integration tests



Results
after
analysis
trigger

Feature Files (e.g. Show Dashboard)

Feature: Use Case Show Dashboard

As a USER

I want to open the app or navigate to the dashboard from another view.
Therefore, I will need to open the app or click the corresponding button.

Background:

Given The application has the needed local permissions

@postsession-feature:

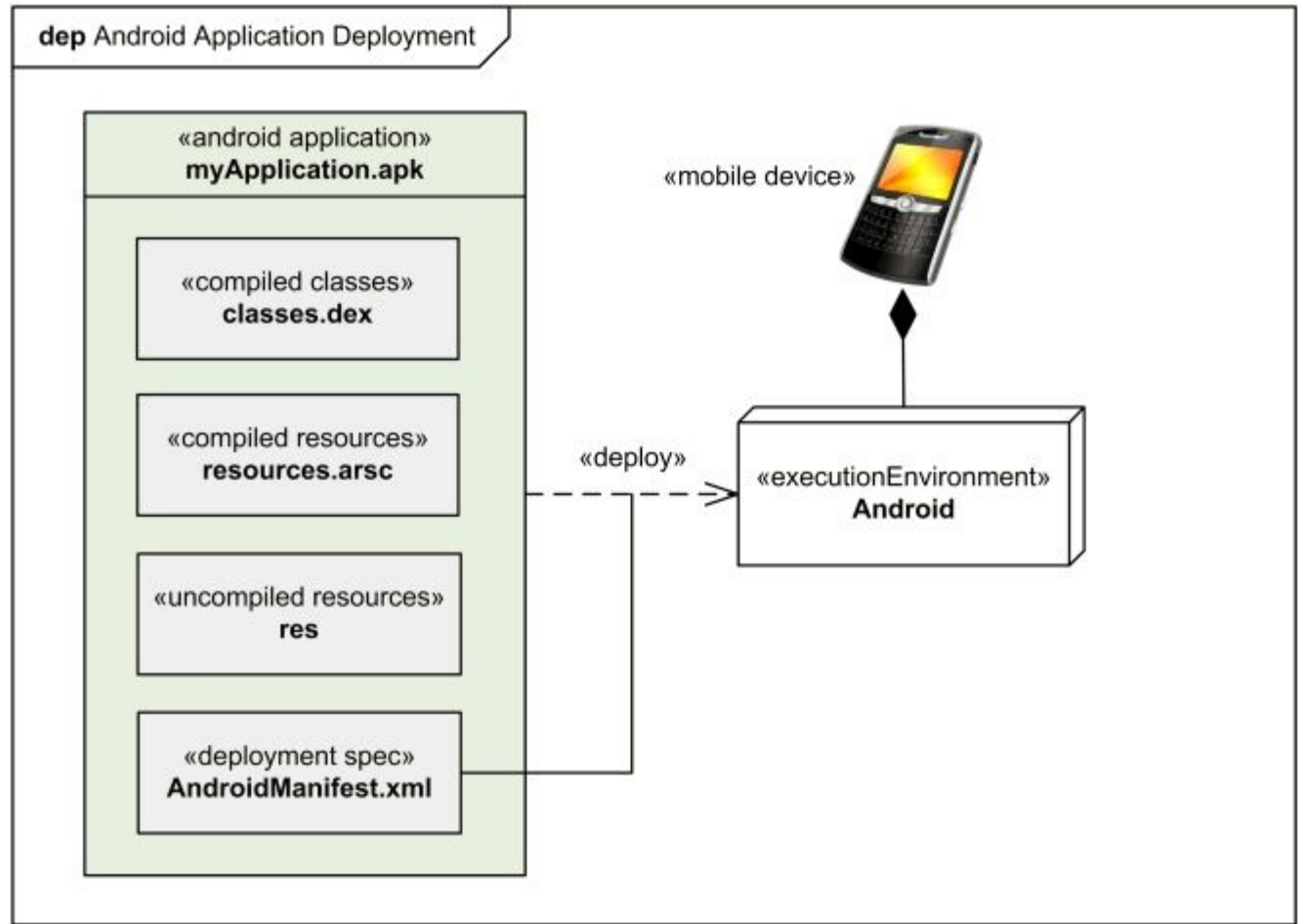
Scenario Outline: Show Dashboard

When The user clicks on the App Icon or the Dashboard button

Then The dashboard with is shown

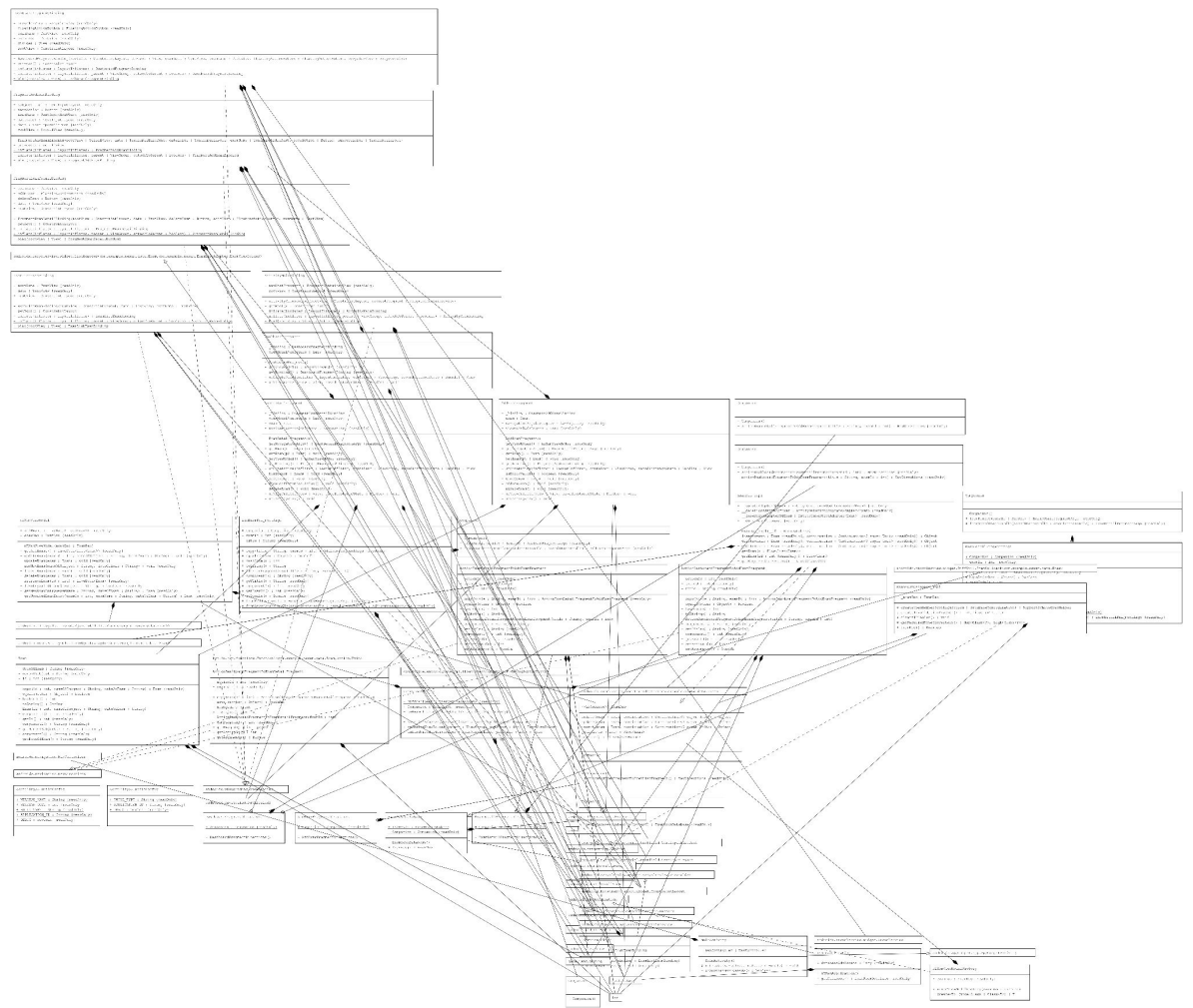
And the text view is loaded with the exams from the database

Deployment



Development

UML Diagramm



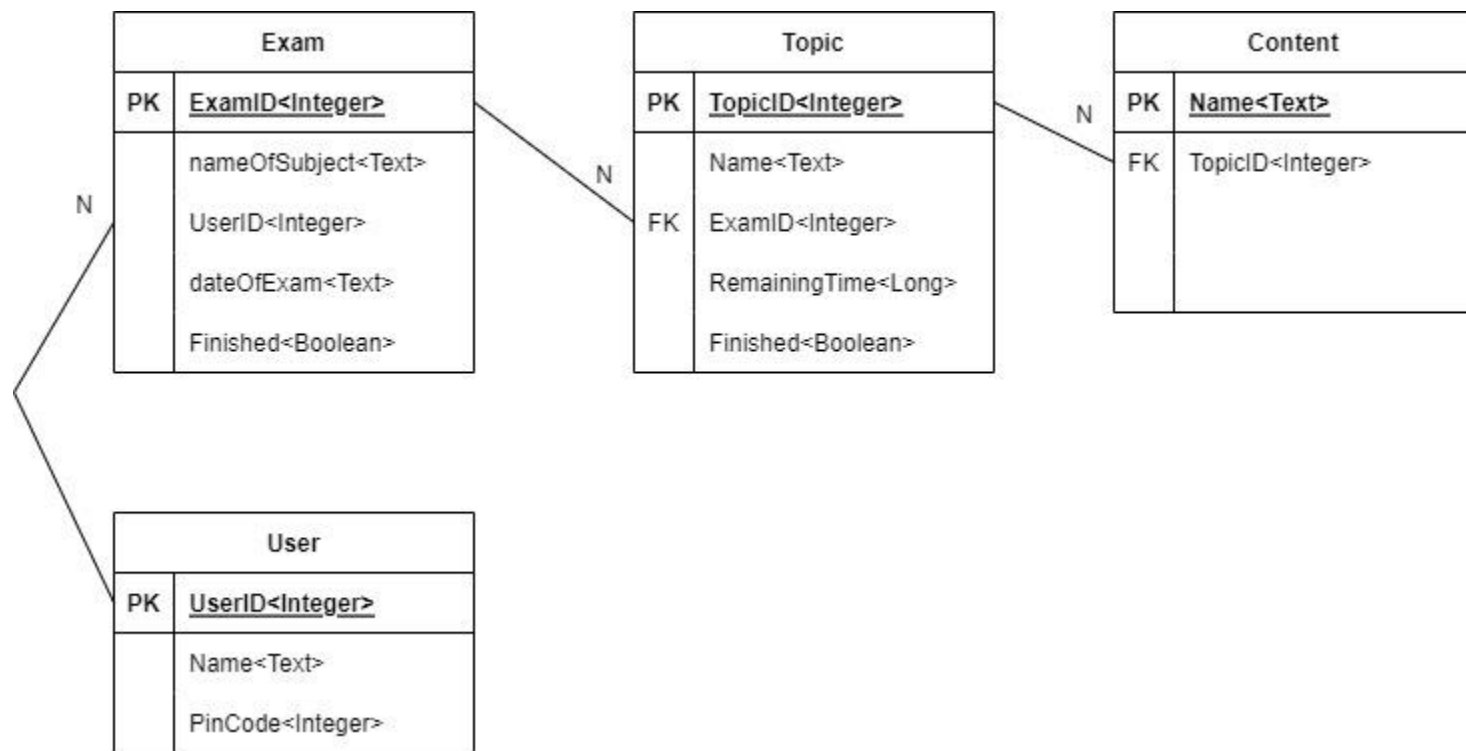
Implementation (e.g. observer pattern)

```
// Attach an observer on the allExams list to update the UI automatically when the data
// changes.
viewModel.allExams.observe(this.viewLifecycleOwner) { exams ->
    exams.let {
        adapter.submitList(it)
    }
}
```

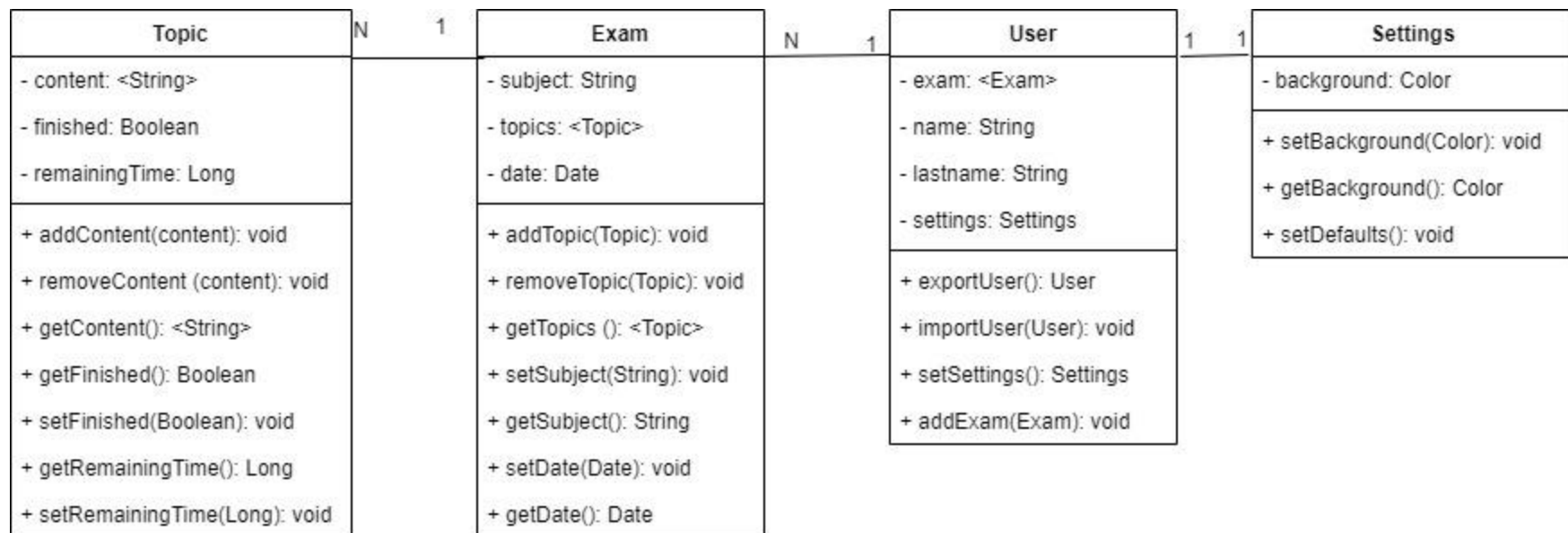
additional patterns:

- builder for database
- recyclerview for list items
- viewModel factory for the viewModel
- adapter for data binding

Database scheme



Class diagram



Backend

- Database Management with Room
- Access through DAO
- SQL underneath



Frontend



Intuitive use of the app



self explaining buttons



dark and light mode →
entirely according to the
preferences of the user



structure with Native
Android framework

Live Demo

Thank you for your attention!

Links to everything shown can be found in our [final blog post](#)