Ceangal Messenger

YOUR OWN WAY OF COMMUNICATION

Final presentation

Table of Contents

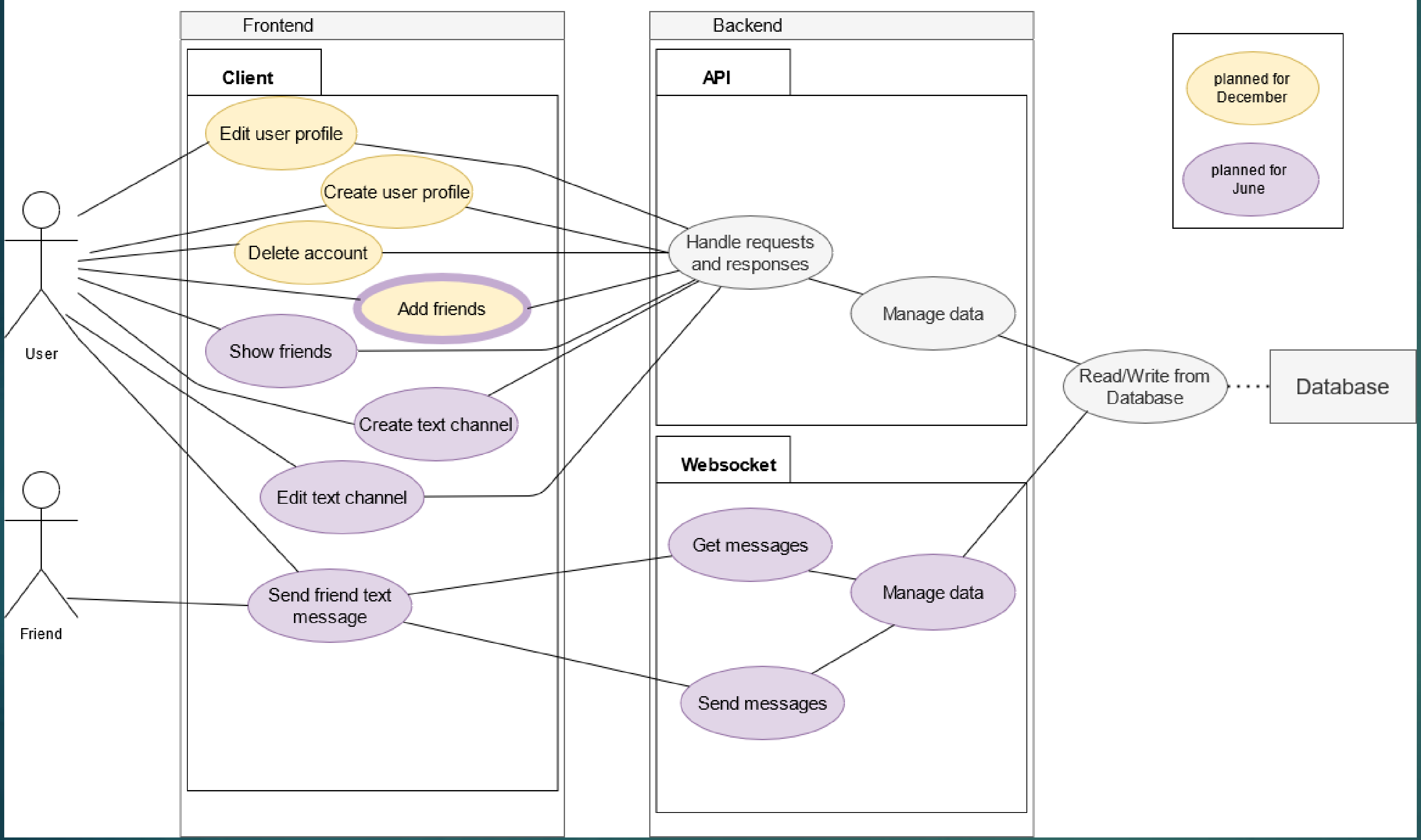
- Vision
- Business plan
  - Use case diagram
  - Non functional
  - Project management Scrum
  - Burndown chart
  - RUP
  - Risk management
  - Function points
- Technical requirements
  - Environment overview
  - Architecture
  - Technologies
  - Testing & Metrics
  - Deployment
- Development
  - Class diagram
  - Pattern
  - Database ER-Model
- Live presentation

Our Vision

- Simple social app to chat with friends
- Self deployable servers
- Create your profile and edit it
- Open source and expandable

Business plan

Use case diagram



Non functional

- Minimalistic
- Easy to understand
- Lightweight UI
- Fast
- Possibility to multiplatform

What helped us to get Project Management right?

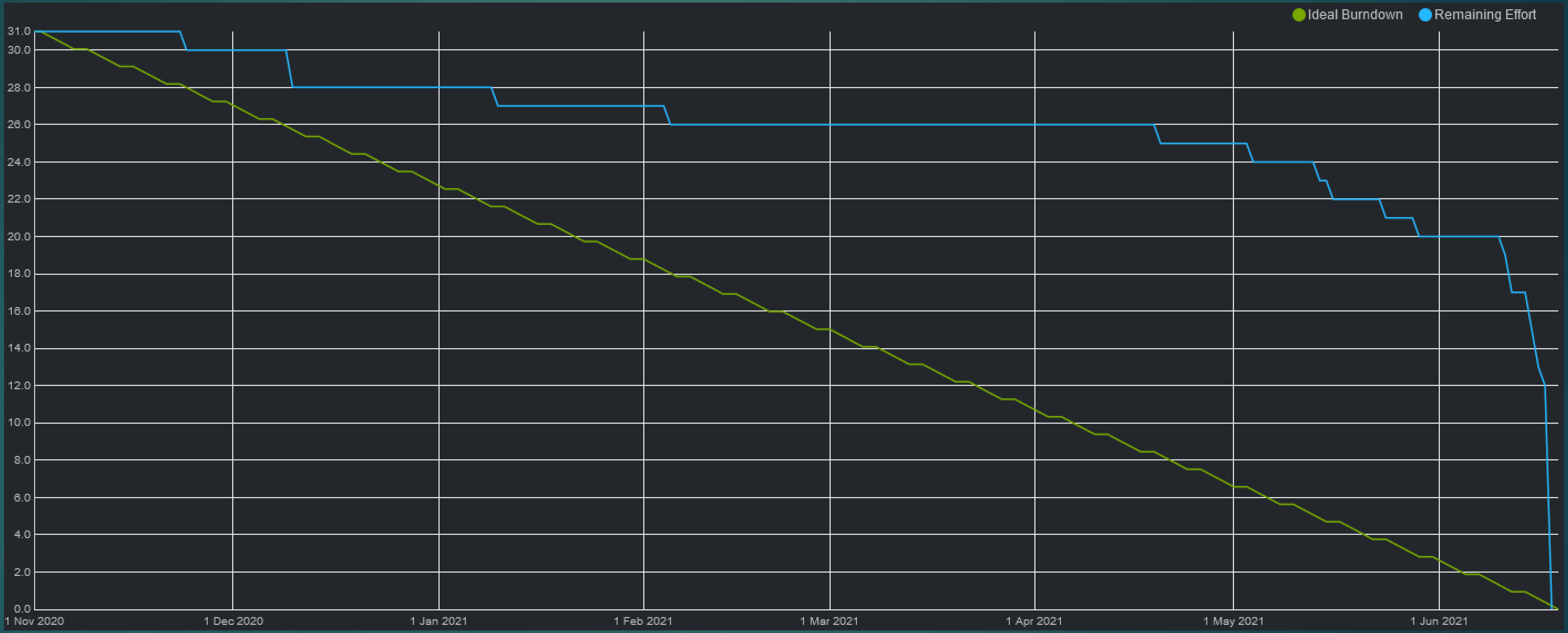
- Regular sprint meetings
- Split up workload to small pieces and allocate time right
- Time tracking
- Regular coding sessions
- Working with Git (Master, different branches, automatic code reviews)
- Proper risk management (risk chance, risk weight, responsible person)

Why Scrumming / Youtrack?

- Weekly sprint goals
- Distribution of tasks
- Check results of a sprint

- Real-time agile board
- Clear view over tasks
- Easy integration into Intellij
- Good and automatic time tracking

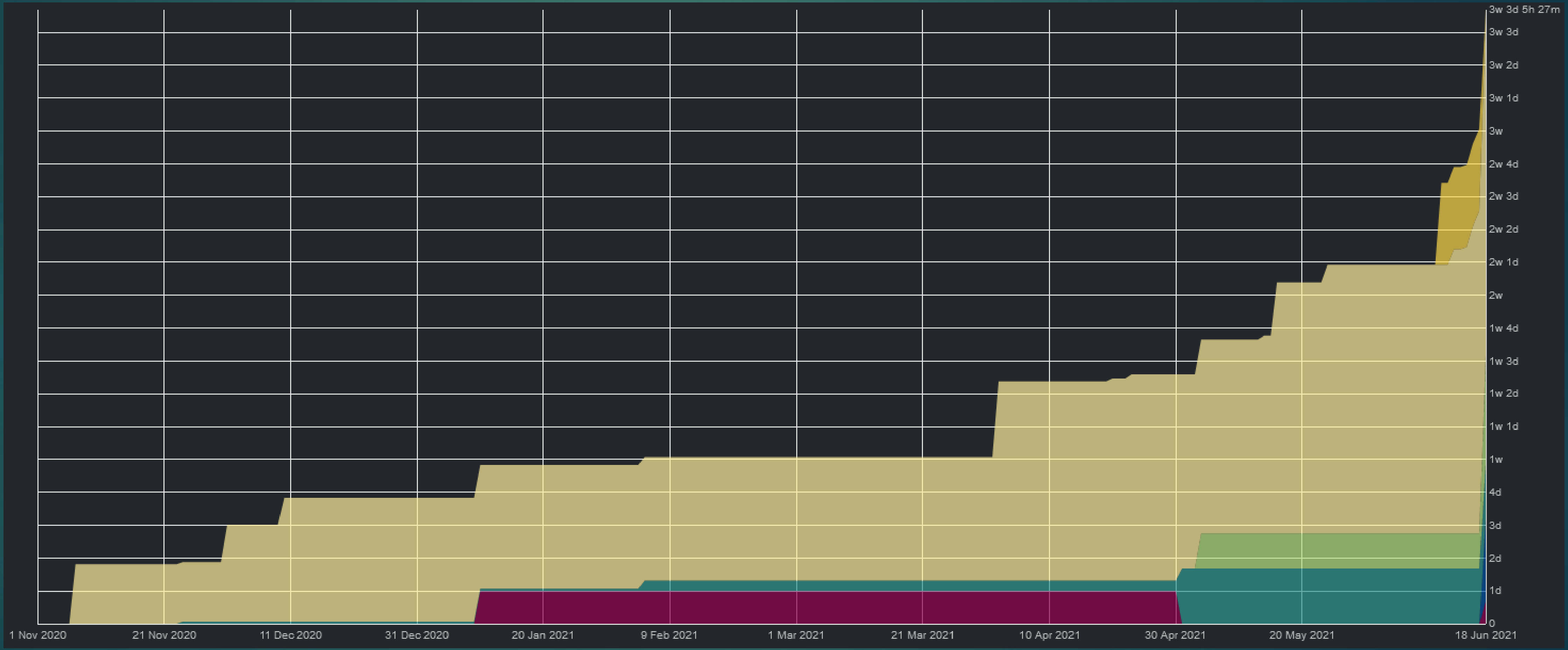
Burndown chart



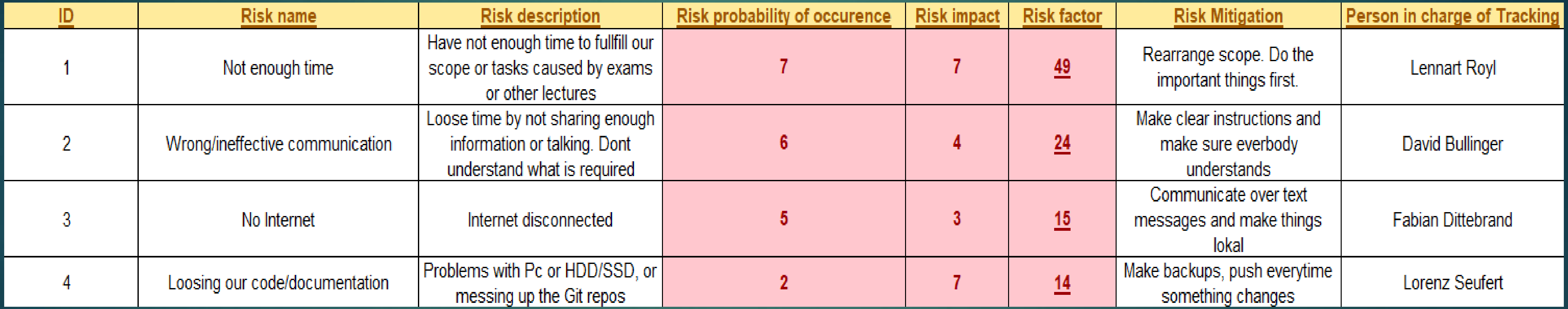
RUP roles

Member	Discipline	Area in Project
David Bullinger	Implementation, Test Manager	Backend
Lennart Royl	Implementation, Database design	Backend, Database
Fabian Dittebrand	Implementation, Test Manager	Websocket, Frontend
Lorenz Seufert	Implementation, Deployment Manager	Frontend, Design

RUP flowchart

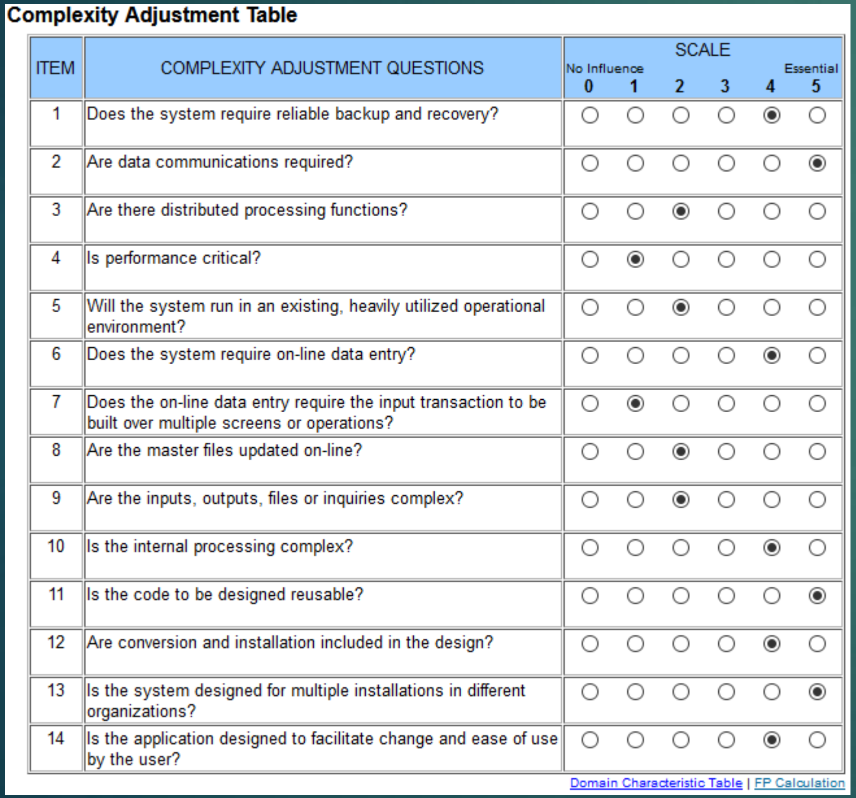


Risk management



ID	Risk name	Risk description	Risk probability of occurence	Risk impact	Risk factor	Risk Mitigation	Person in charge of Tracking
1	Not enough time	Have not enough time to fullfill our scope or tasks caused by exams or other lectures	7	7	49	Rearrange scope. Do the important things first.	Lennart Royl
2	Wrong/ineffective communication	Loose time by not sharing enough information or talking. Dont understand what is required	6	4	24	Make clear instructions and make sure everbody understands	David Bullinger
3	No Internet	Internet disconnected	5	3	15	Communicate over text messages and make things lokal	Fabian Dittebrand
4	Loosing our code/documentation	Problems with Pc or HDD/SSD, or messing up the Git repos	2	7	14	Make backups, push everytime something changes	Lorenz Seufert

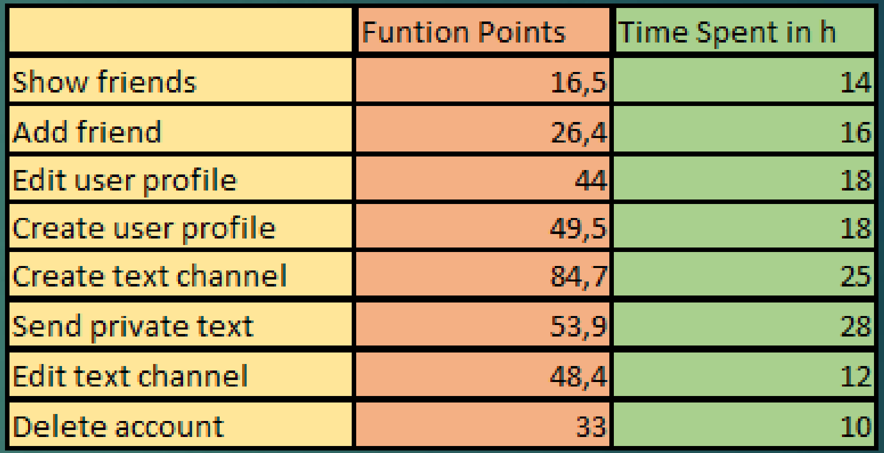
Function points (Cost/Time estimation)



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



	Funtion Points	Time Spent in h
Show friends	16,5	14
Add friend	26,4	16
Edit user profile	44	18
Create user profile	49,5	18
Create text channel	84,7	25
Send private text	53,9	28
Edit text channel	48,4	12
Delete account	33	10

Estimated time per Use Case

Technical requirements

Environment

- API Testing with JUnit and Spring
- Unit tests with JUnit
- Cucumber

IntelliJ IDE -

Dev Environment

Git -

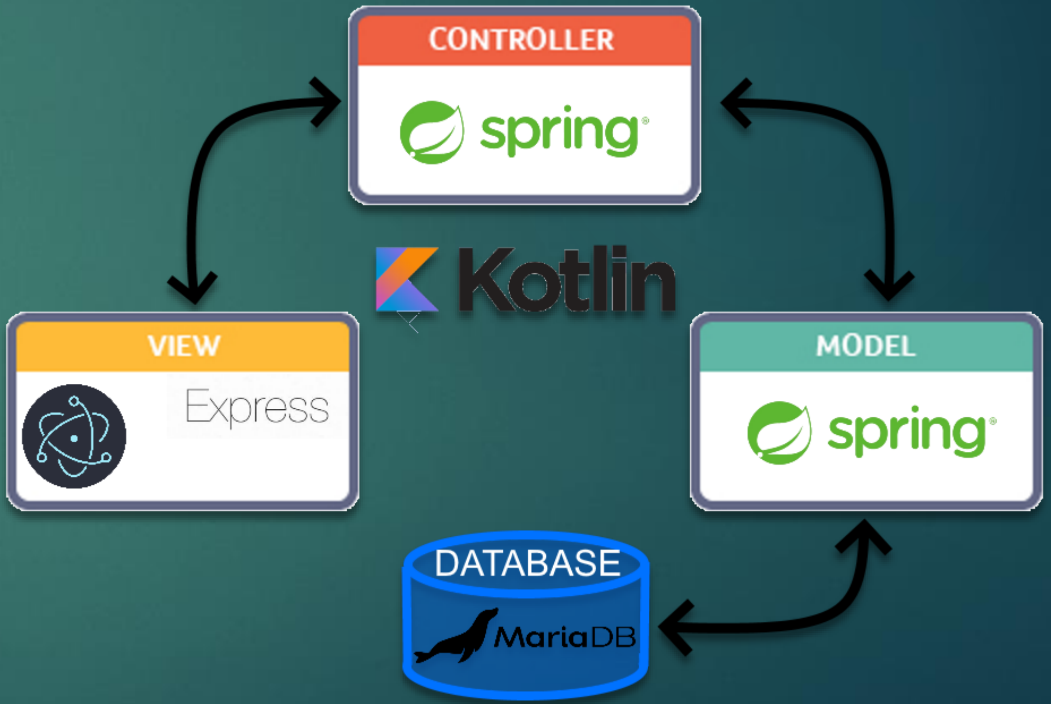
Version Control

YouTrack –
Management Tool

Discord –
Conferencing Tool

Architecture

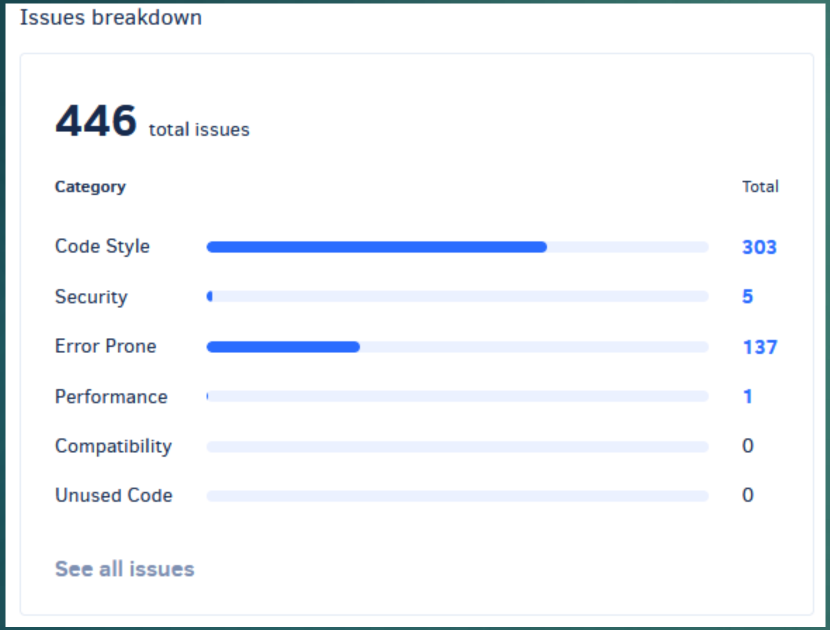
- MVC-Design
- REST-Paradigma
- Cross Platform



Technologies

- Frontend
  - Electron
  - Express Framework
  - EJS
- Backend
  - Spring Boot
  - Spring Data JPA
  - Kotlin
  - Maven
- Database
  - MariaDB
- Testing
  - JUnit
  - API tests with JUnit and Spring

Testing & Metrics



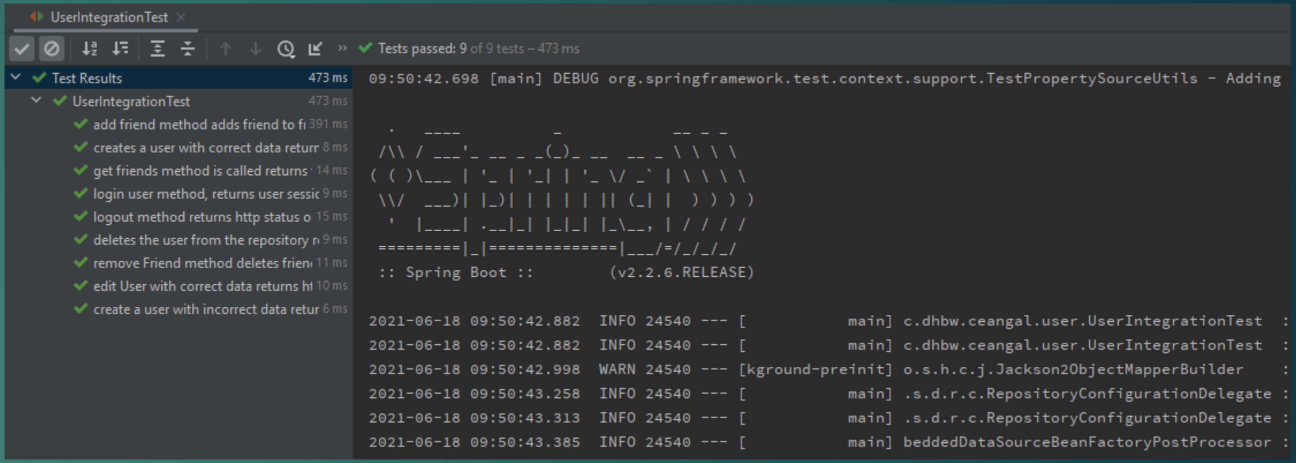
Issues breakdown

446 total issues

Category	Total
Code Style	303
Security	5
Error Prone	137
Performance	1
Compatibility	0
Unused Code	0

See all issues

Issue breakdown of Codacy



UserIntegrationTest

✓ Tests passed: 9 of 9 tests – 473 ms

- ✓ Test Results 473 ms
  - ✓ UserIntegrationTest 473 ms
    - ✓ add friend method adds friend to fı 391 ms
    - ✓ creates a user with correct data returr 8 ms
    - ✓ get friends method is called returns 14 ms
    - ✓ login user method, returns user sessic 9 ms
    - ✓ logout method returns http status o 15 ms
    - ✓ deletes the user from the repository rı 9 ms
    - ✓ remove Friend method deletes frien 11 ms
    - ✓ edit User with correct data returns hı 10 ms
    - ✓ create a user with incorrect data retur 6 ms

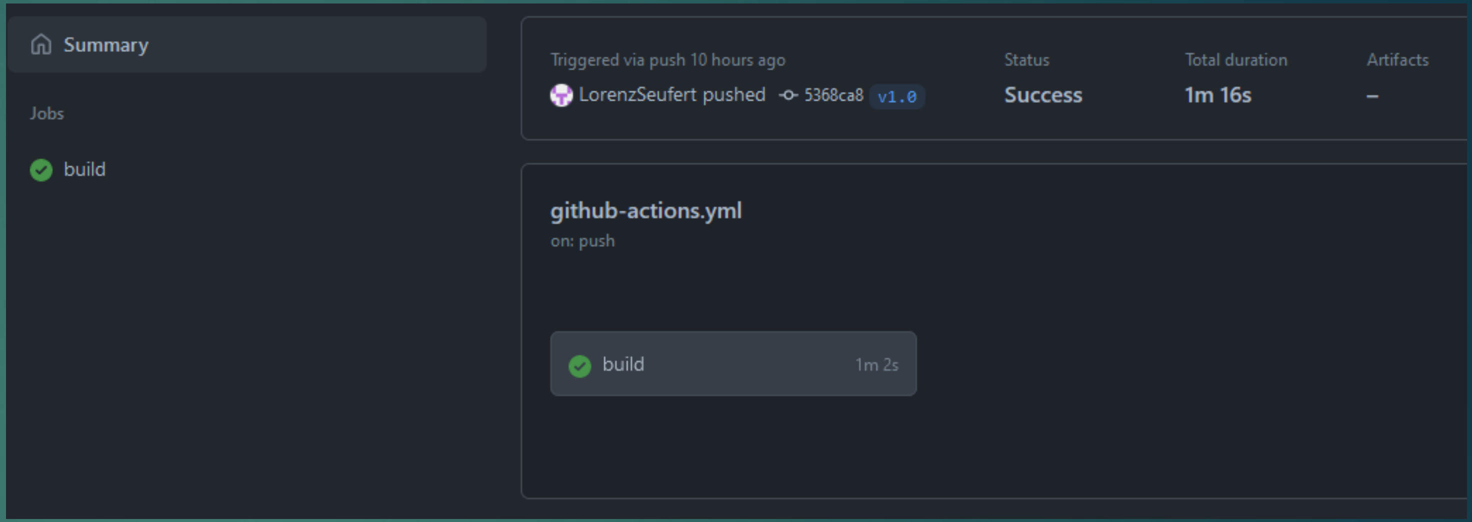
```
09:50:42.698 [main] DEBUG org.springframework.test.context.support.TestPropertySourceUtils - Adding

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.6.RELEASE)

2021-06-18 09:50:42.882  INFO 24540 --- [           main] c.dhbw.ceangal.user.UserIntegrationTest  :
2021-06-18 09:50:42.882  INFO 24540 --- [           main] c.dhbw.ceangal.user.UserIntegrationTest  :
2021-06-18 09:50:42.998  WARN 24540 --- [kground-preinit] o.s.h.c.j.Jackson2ObjectMapperBuilder    :
2021-06-18 09:50:43.258  INFO 24540 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate :
2021-06-18 09:50:43.313  INFO 24540 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate :
2021-06-18 09:50:43.385  INFO 24540 --- [           main] beddedDataSourceBeanFactoryPostProcessor :
```

Example API test

Deployment



Summary

Jobs

✓ build

Triggered via push 10 hours ago

LorenzSeufert pushed 5368ca8 v1.0

Status

Success

Total duration

1m 16s

Artifacts

–

github-actions.yml

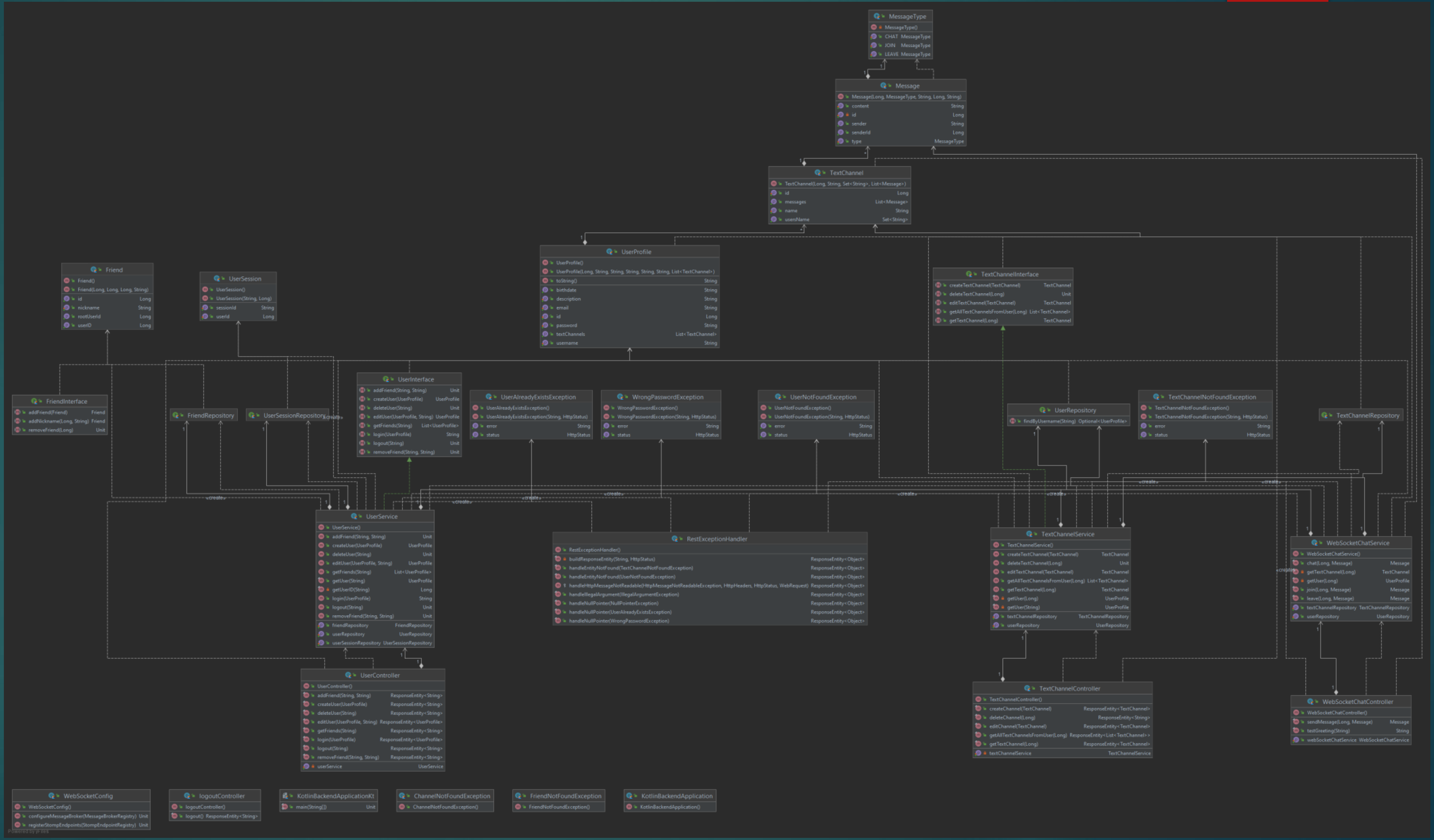
on: push

✓ build 1m 2s

- CI (Continuous Integration)
  - Pushes will be tested
- CD (Continuous Delivery)
  - Upload a release to GitHub when new features are available

Development

UML diagram

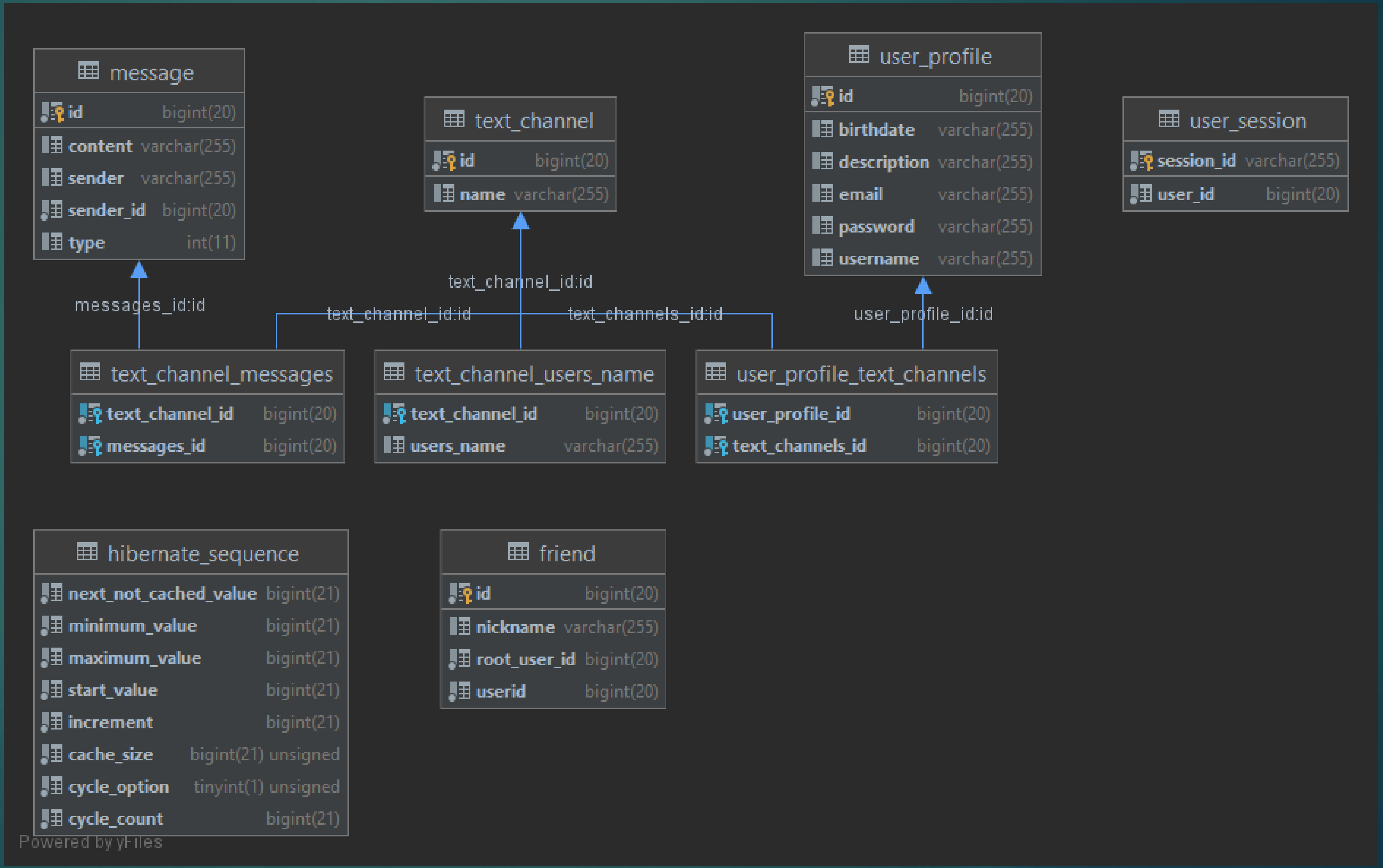

MessageType
MessageType()
CHAT MessageType
JOIN MessageType
LEAVE MessageType
Message
Message(Long, MessageType, String, Long, String)
content String
id Long
sender String
senderId Long
type MessageType
TextChannel
TextChannel(Long, String, Set<String>, List<Message>)
id Long
messages List<Message>
name String
usersName Set<String>
UserProfile
UserProfile()
UserProfile(Long, String, String, String, String, String, List<TextChannel>)
toString() String
birthdate String
description String
email String
id Long
password String
textChannels List<TextChannel>
username String
TextChannelInterface
createTextChannel(TextChannel) TextChannel
deleteTextChannel(Long) Unit
editTextChannel(TextChannel) TextChannel
getAllTextChannelsFromUser(Long) List<TextChannel>
getTextChannel(Long) TextChannel
Friend
Friend()
Friend(Long, Long, Long, String)
id Long
nickname String
rootUserId Long
userID Long
UserSession
UserSession()
UserSession(String, Long)
sessionId String
userId Long
FriendInterface
addFriend(Friend) Friend
addNickname(Long, String) Friend
removeFriend(Long) Unit
FriendRepository
UserSessionRepository
UserInterface
addFriend(String, String) Unit
createUser(UserProfile) UserProfile
deleteUser(String) Unit
editUser(UserProfile, String) UserProfile
getFriends(String) List<UserProfile>
login(UserProfile) String
logout(String) Unit
removeFriend(String, String) Unit
UserAlreadyExistsException
UserAlreadyExistsException()
UserAlreadyExistsException(String, HttpStatus)
error String
status HttpStatus
WrongPasswordException
WrongPasswordException()
WrongPasswordException(String, HttpStatus)
error String
status HttpStatus
UserNotFoundException
UserNotFoundException()
UserNotFoundException(String, HttpStatus)
error String
status HttpStatus
UserRepository
findByUsername(String) Optional<UserProfile>
TextChannelNotFoundException
TextChannelNotFoundException()
TextChannelNotFoundException(String, HttpStatus)
error String
status HttpStatus
TextChannelRepository
«create»
UserService
UserService()
addFriend(String, String) Unit
createUser(UserProfile) UserProfile
deleteUser(String) Unit
editUser(UserProfile, String) UserProfile
getFriends(String) List<UserProfile>
getUser(String) UserProfile
getUserID(String) Long
login(UserProfile) String
logout(String) Unit
removeFriend(String, String) Unit
friendRepository FriendRepository
userRepository UserRepository
userSessionRepository UserSessionRepository
RestExceptionHandler
RestExceptionHandler()
buildResponseEntity(String, HttpStatus) ResponseEntity<Object>
handleEntityNotFound(TextChannelNotFoundException) ResponseEntity<Object>
handleEntityNotFound(UserNotFoundException) ResponseEntity<Object>
handleHttpMessageNotReadable(HttpMessageNotReadableException, HttpHeaders, HttpStatus, WebRequest) ResponseEntity<Object>
handleIllegalArgument(IllegalArgumentException) ResponseEntity<Object>
handleNullPointer(NullPointerException) ResponseEntity<Object>
handleNullPointer(UserAlreadyExistsException) ResponseEntity<Object>
handleNullPointer(WrongPasswordException) ResponseEntity<Object>
TextChannelService
TextChannelService()
createTextChannel(TextChannel) TextChannel
deleteTextChannel(Long) Unit
editTextChannel(TextChannel) TextChannel
getAllTextChannelsFromUser(Long) List<TextChannel>
getTextChannel(Long) TextChannel
getUser(Long) UserProfile
getUser(String) UserProfile
textChannelRepository TextChannelRepository
userRepository UserRepository
WebSocketChatService
WebSocketChatService()
chat(Long, Message) Message
getTextChannel(Long) TextChannel
getUser(Long) UserProfile
join(Long, Message) Message
leave(Long, Message) Message
textChannelRepository TextChannelRepository
userRepository UserRepository
UserController
UserController()
addFriend(String, String) ResponseEntity<String>
createUser(UserProfile) ResponseEntity<String>
deleteUser(String) ResponseEntity<String>
editUser(UserProfile, String) ResponseEntity<UserProfile>
getFriends(String) ResponseEntity<String>
login(UserProfile) ResponseEntity<UserProfile>
logout(String) ResponseEntity<String>
removeFriend(String, String) ResponseEntity<String>
userService UserService
TextChannelController
TextChannelController()
createChannel(TextChannel) ResponseEntity<TextChannel>
deleteChannel(Long) ResponseEntity<String>
editChannel(TextChannel) ResponseEntity<TextChannel>
getAllTextChannelsFromUser(Long) ResponseEntity<List<TextChannel>>
getTextChannel(Long) ResponseEntity<TextChannel>
textChannelService TextChannelService
WebSocketChatController
WebSocketChatController()
sendMessage(Long, Message) Message
testGreeting(String) String
webSocketChatService WebSocketChatService
WebSocketConfig
WebSocketConfig()
configureMessageBroker(MessageBrokerRegistry) Unit
registerStompEndpoints(StompEndpointRegistry) Unit
logoutController
logoutController()
logout() ResponseEntity<String>
KotlinBackendApplicationKt
main(String[]) Unit
ChannelNotFoundException
ChannelNotFoundException()
FriendNotFoundException
FriendNotFoundException()
KotlinBackendApplication
KotlinBackendApplication()
Powered by yFiles

Pattern implementation

- Use the builder pattern
  - named arguments in Kotlin
  - Default values in classes
- Easy to see what gets created
- Easily implement new representations without changing the object

- Covered Patterns by Frameworks:
  - Singleton Pattern
  - MVC Pattern
  - Proxy Pattern
  - …

DB ER-Model



message	
id	bigint(20)
content	varchar(255)
sender	varchar(255)
sender_id	bigint(20)
type	int(11)

text_channel	
id	bigint(20)
name	varchar(255)

user_profile	
id	bigint(20)
birthdate	varchar(255)
description	varchar(255)
email	varchar(255)
password	varchar(255)
username	varchar(255)

user_session	
session_id	varchar(255)
user_id	bigint(20)

text_channel_messages	
text_channel_id	bigint(20)
messages_id	bigint(20)

text_channel_users_name	
text_channel_id	bigint(20)
users_name	varchar(255)

user_profile_text_channels	
user_profile_id	bigint(20)
text_channels_id	bigint(20)

hibernate_sequence	
next_not_cached_value	bigint(21)
minimum_value	bigint(21)
maximum_value	bigint(21)
start_value	bigint(21)
increment	bigint(21)
cache_size	bigint(21) unsigned
cycle_option	tinyint(1) unsigned
cycle_count	bigint(21)

friend	
id	bigint(20)
nickname	varchar(255)
root_user_id	bigint(20)
userid	bigint(20)

Live presentation

Thank you for your attention!

Links to all documents/reports/etc. can be found in our final blogpost on 29.06.2021