





Vision

Übersichtlicher und nutzerfreundlicher Eventplaner, um besonders im privaten Bereich möglichst viele Anwender zu erreichen



Was wir bisher erreicht haben

- UI mit JavaFX
- Account System
- REST API
- MariaDB Datenbank
- Event System

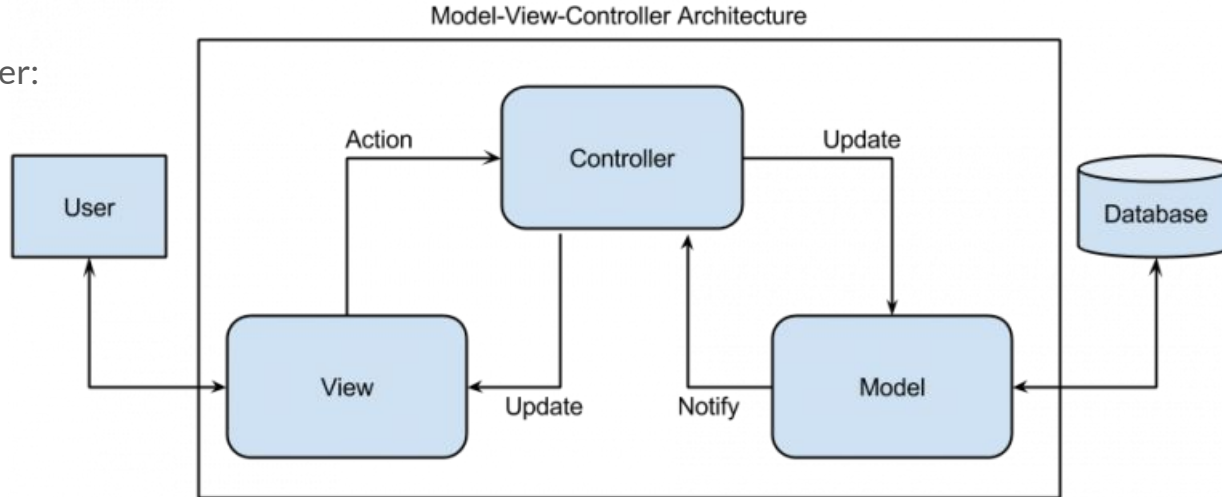


Verworfenene Ideen

- Kalender auf der Startseite
- Gruppen
- Bilder für Benutzer, Events und Gruppen
- Chat
- Notifications

Architekturentscheidungen

MVC Muster:



<https://www.methodpark.de/blog/wp-content/uploads/2018/07/Model-View-Controller-High-Level-Diagram-768x339.png>























Aber wieso?

- Trennung von Komponenten
 - > Erlaubt an einer Komponente zu arbeiten, ohne die anderen zu berühren

REST API

- Aufgrund von Vertrautheit des Teams

 /api/group/ [GET]	GroupController
 /api/group/ [POST]	GroupController
 /api/group/{id} [GET]	GroupController
 /api/group/{id} [PATCH]	GroupController
 /api/group/{id} [DELETE]	GroupController
 /api/group/{id}/add [POST]	GroupController
 /api/group/{id}/remove [POST]	GroupController
 /api/auth/delete [DELETE]	AuthController
 /api/auth/login [GET]	AuthController
 /api/auth/me [GET]	AuthController
 /api/auth/signup [POST]	AuthController
 /api/event/ [GET]	EventController
 /api/event/ [POST]	EventController
 /api/event/{id} [GET]	EventController
 /api/event/{id} [PATCH]	EventController
 /api/event/{id} [DELETE]	EventController
 /api/event/{id}/add [POST]	EventController
 /api/event/{id}/remove [POST]	EventController
 /api/user/ [GET]	UserController
 /api/user/{ids} [GET]	UserController



Single Responsibility Principle

- Eine Klasse ist nur für eine Funktion zuständig
- Erlaubt ebenfalls das Arbeiten an einer Funktion, ohne andere zu berühren

```
public class Event { 37 usages  ⓘ Impo lex +1 *  
    //Attributes  
    private User host; 3 usages  
    private List<User> organisers = new ArrayList<>(); 2 usages  
    private List<EventMember> participants = new ArrayList<>(); 4 usages  
    private Chat chat; 3 usages  
    private String title; 3 usages  
    private String description; 3 usages  
    private String date; 3 usages  
    private String place; 3 usages  
    private ArrayList<EventViewUI> UIs = new ArrayList<>(); 5 usages
```

```
public class User { ⓘ Impo lex +1 *  
    //Attributes  
    private int userId; 3 usages  
    private String userName; 3 usages  
    private List<Event> events = new ArrayList<>(); 2 usages  
    private List<Group> groups; 2 usages  
    private Map<String, String> settings; 2 usages  
    private Notification notifications; no usages
```




Tech-Stack

- MariaDB
- Spring Boot Api
- JUnit
- JavaFx
- Maven
- GitHub



CI/CD

- Maven
- GitHub Actions
- Unit Tests, falls Änderung am Code
- Automatisch beim Merge auf main
- Benachrichtigung bei fehlgeschlagenen Tests

```
name: Java CI with Maven

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:

    runs-on: windows-latest

    steps:
      - uses: actions/checkout@v4
      - name: Set up JDK 21
        uses: actions/setup-java@v3
        with:
          java-version: '21'
          distribution: 'temurin'
          cache: maven
      - name: Build with Maven
        #run: mvn -B package --file pom.xml
        run: mvn package -f pom.xml
```



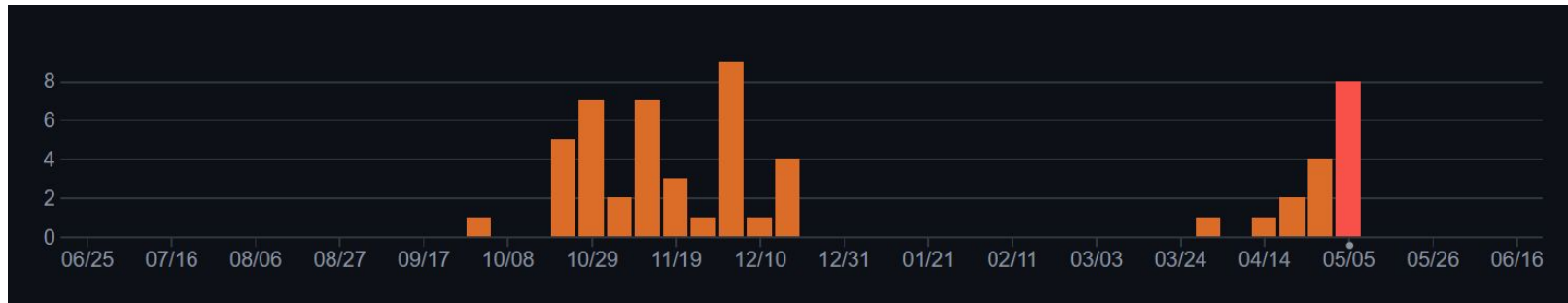
Metriken

- Commit frequency
- Depth of inheritance tree
- Coupling between objects
- Berechnet mit “Metrics Reloaded”



Commit frequency

- Aktivität am Projekt
- Misst Code und Dokumentation (dev und doc Branches)



Depth of inheritance tree

- Hoher Wert -> komplex
- Niedriger Wert -> Vererbung gering ausgenutzt

class ▲	DIT
classes.Chat	1
classes.DBConnec	1
classes.Event	1
classes.EventMen	1
classes.Group	1
classes.Message	1
classes.Notificatio	1
classes.User	1
main.Main	1
ui.EventChatView	2
ui.EventHostView	2
ui.EventParticipat	2
ui.EventspageView	2
ui.LoginView	2
ui.MainView	2
ui.StageHelper	1
ui.test	2
unittests.EventTes	1
Total	
Average	1,39

Coupling between objects

- Hoher Wert -> hohe Komplexität

interface ▲		CBO
🔒	interfaces.Event	4
🔒	interfaces.UI	3
Total		
Average		3,50



Live-Demo

Retrospektive

