MaristenPlaner

Ein Projekt der Q11/Q12 Informatikklasse
2021/2022

Planungsphase

11.5.2021

Probleme:

-Zeitaufwand

-App Priorisierung

-Aufgabenverteilung

    -Maxi & Kayra müssen andere Aufgaben annehmen (z.B. Design)

-Datenschutz (Art des Logins, Speicherung der Daten, Server Standort etc.)

    -Stand Alone System (Ohne Microsoft's Active Directory/API oder andere Third-Party)

Anforderungen:

-API

-App/Programm/Website

-Server

-Vielleicht Database

-GUIs (z.B. NetBeans)

Aufgaben der Software:

-Liveverfolgung der Stunden- & Vertretungspläne

-Anzeigen des aktuellen Mensaplans für die aktuelle & nächste Woche

-Spezielles Interface für das Sekretariat

-Verschiedene Funktionen für verschiedene Rollen (Schüler, Lehrer etc.)

<u>Testen erfolgt während jedem Schritt</u>

13.5.2021

Grundsätzliche Aufgabenverteilung:

-Leo: Server, Server API (Mithilfe), Website, App

-Simon: Server API, Server, Programm (Mithilfe)

-Visar: Server API (Mithilfe), App (Mithilfe), Programm, Website (Mithilfe)

-Kayra: Dokumentation, Tests & Feedback, GUI Design

-Maxi: GUI Design, Tests & Feedback, Backup für Dokumentation

Implementier Reihenfolge

16.5.2021

Bei jeder Implementierung (Siehe Reihenfolge der Implementierungen: INF1 11.05.2021):

Zunächst bilden wir die konkrete Softwarestruktur (Klassen, Interfaces usw.) Dann implementieren wir die Softwarestruktur und testen sie letztendlich. Falls es Verbesserungsvorschläge gibt, fangen wir mit dem Prozess wieder an, bis wir mit dem Ergebnis zufrieden sind.

11.5.2021

1. Server & API
2. Design/GUI
3. App
4. Website App
5. Programm

Probleme aufgetreten während der Implementierung

18.5.2021

- IOS App nicht möglich
    - Zeitmangel
    - Implementierung nur über Apple Gerät möglich
    - Wird höchst wahrscheinlich nicht im App Store verfügbar zu machen sein

15.6.2021

- Microsoft Azure Active Directory funktioniert nicht

8.7.2021

- Distribution
    - Play Store verlangt 25 $ für eine Developer-Lizenz um die App online zu stellen
    - App Store verlangt $100 pro Jahr für eine Developer-Lizenz um die App online zu stellen
- getSchedule ist nicht kompatibel zu der Datenbank
    - getSchedule musste in Android App, Sekretariatsapp und Server neu geschrieben werden

Implementierung Fortschritt

Anmeldung

15.6.2021
-19.6.2021

Dadurch dass die Implementierung der Anmeldung über Azure nicht klappt, benutzen wir stattdessen einen E-Mail Bot, um die angemeldeten Personen zu verifizieren
Funktionalität:

1. Zunächst melden sich die Benutzer mit ihrer Schulemail an
2. Dann erhalten sie von unserem Bot eine Nachricht, welche einen 6-stelligen Code beinhaltet
3. Dieser Code muss in der App innerhalb von 15 Minuten angegeben werden, um sich in der App zu verifizieren

Dazu benötigt:

- No Reply E-Mail Adresse : noreply.maristenkolleg@gmail.com
- Implementierung mit Mail.java (generiert automatische E-Mail):

  Hallo Visar Lumi,
  Dein Code lautet 201022.
  Mit freundlichen Grüßen,
  Ihr Maristenplaner-Team

- Umwandlung Mail.java zu Kotlin
  - Bessere Performance
  - Bessere Codeintegration

6.7.2021

Server/ API

Implementierung über:

Ktor (Framework) und Kotlin (Programmiersprache)

Der Server bzw. die API fungiert als Schnittstelle zwischen dem client, der Android App oder der Sekretariatsapp, und der Datenbank. Dafür gibt es verschiedene Requests

- /schedule [Tag] [Klasse] [Fächer] = Eingabefunktion für Stundenpläne
  - Antwort: Stundenplan an dem eingegebenen Tag und Speicherung in der Datenbank
- /classes
  - Antwort: alle Schulklassen und Fächer
    - Grund: Die App braucht die Auswahl damit sie die Daten von der Datenbank abrufen kann.
- /register [Vorname] [Nachnahme] [E-Mail] = Registrierung in der App + Überprüfung, ob es eine Schulemailadresse ist.
  - Antwort Code wird an gültige E-Mail-Adresse geschickt
  - Code wir im Cache gespeichert und nach 15 Minuten gelöscht-> Registrierungscode hat somit eine Gültigkeit von 15 min
- /login [E-Mail] [Code] = 2. Stufe der Registrierung in der App
  - Nach richtiger Eingabe wird vom Server eine eigene Id zugewiesen. Diese Id wird gespeichert und dient zur künftigen Anmeldung, ohne alle Daten neu eingeben zu müssen

Datenbank

Implementierung über:

Mysql (Open-Source Datenbank-Service/Programm) mit SQL (Programmiersprache)

Kotllin (Programmiersprache) für Server/API und App;

JDBC Java (Programmiertyp/-sprache) für Maristenkonsole.

Hierarchie in der Datenbank:

App, API -> können nur Abrufen haben keine Änderungsbrechtigung

Datenbank

Maristenkonsole -> kann alle Daten ändern neu eingeben etc.

Datenbank- Daten

Eine 2. Manuelle Eingabe aller Stunden, Lehrer etc. wäre nicht sinnvoll.

Die Daten werden direkt aus Willy2, der Stundenplanerstellungssoftware, welche zurzeit am Maristenkolleg benutzt wird, importiert. Willy exportiert eine Exel Datei, welche wiederum dann in eine CSV Datei konvertiert werden kann und diese kann dann benutzt bzw. übersetzt werden.

Aktuell werden aus datenschutzrechtlichen Gründen nur Demo- Lehrer und Stunden von Willy2 benutzt.

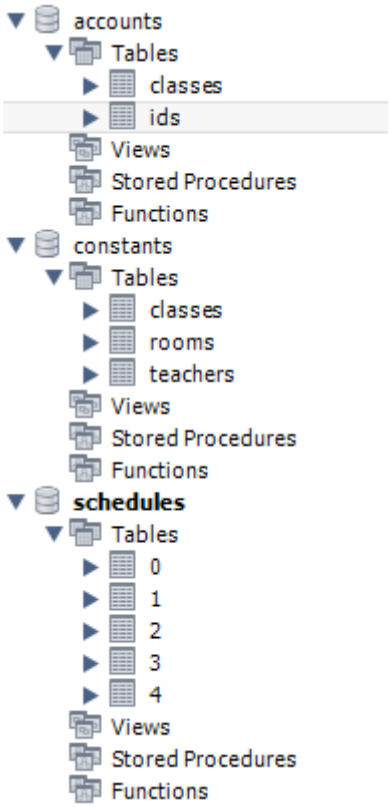
Datenbank - Struktur

Die Datenbank ist unterteilt in mehrere „Unterdatenbanken“, die „Schemas“.

In den „Schemas“ sind „Tabellen“ gespeichert, die aber keine Zeilen besitzen, sondern nur Spalten.

Die Maristenplaner Datenbank besitzt drei schemes, in denen verschiedene Daten gespeichert werden:

- accounts: Die Klassen und Fächerauswahl der Schüler [classes] und die Anmelde Id´s [ids].
- constants: Konstanten die sich nicht groß verändern,
    - Schulklassen [classes]: Klassen und Fächer
    - Räume [rooms]: Kürzel der Räume und Raumname
    - Lehrer [teachers]: Kürzel, Vorname und Nachname
- schedules: Alle Wochentage werden gespeichert dafür steht Montag für 0 und Freitag für 4 etc.



Datenbank - Schwierigkeiten bei der Speicherung der Daten

1.Problem: Mehrere Informationen müssen gespeichert werden

Lösung: Die Datenbank wird in 3 „Schemas“ aufgeteilt: accounts; constants; schedules

2.Problem: Aufrufen der Stundenpläne: Nicht für jeden Schüler kann ein eigener Stundenplan gespeichert werden, da der Speicherplatz dann ineffizient genutzt ist.

Lösungen: 1. Stundenplan aller Klassen oder 2. Den Lehrerstundenplan

1. Den Stundenplan der Klasse einzeln speichern

- Nicht möglich da es zu „Doppelbelegung“ in den Zellen kommen kann und es zusätzlich viel Speicher verbrauchen würde.

  Beispiel: Latein/Französisch

2. Den Lehrerstundenplan als Grundlage nehmen und daraus den Klassenstundenplan zusammenstellen

- Funktioniert, da jeder Lehrer nur eine Klasse pro Stunde unterrichten kann.

3.Problem: Der klassische Stundenplan ist nicht speicherbar, da es keine Reihen gibt, und zusätzlich haben mehrere Lehrer zur gleichen Zeit Unterricht.

Lösung: Jeder Tag bekommt eine eigene Tabelle [schedules>tables>0-4]. In den Spalten stehen die Lehrerkürzel, hier A, Al, Au und B, und die Stunde [hour].

hour	A	Al	Au	B
1	NULL	NULL	NULL	NULL
2	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL
4	NULL	NULL	NULL	{"class"...
5	NULL	NULL	NULL	{"class"...
6	NULL	NULL	NULL	{"class"...
MP	NULL	NULL	NULL	NULL
7	NULL	NULL	NULL	NULL
8	NULL	NULL	NULL	NULL
9	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL

Die Zeile Stunde [hour], dient als eine Art Suchleiste, somit kann man nach dem Kürzel des Lehrers und der Stunde suchen.

Beispiel:

```
1 •   SELECT BeM FROM schedules.`0` WHERE hour='1';
```

Suche Lehrerkürzel [BeM] aus Tabelle Montag [0] in der Stunde [1]

6.Problem: Vertretungen

Die Vertretungen werden in den Stundenplan mit eingebaut. Sollte die Stunde vertreten werden wird gleich der Vertretungslehrer und Raum mit ausgegeben.

Boolean substituted überprüft, ob es nicht stattfindet;

String substituted_teacher gibt den Lehrer aus;

String subsititude_room gibt den Raum aus.

```java
public Lesson(int day, String hour, String teacher, String[] clazz, String subject, int course, String room, boolean substituted, String substitute_teacher, String substitute_room) {
    this.day = day;
    this.hour = hour;
    this.teacher = teacher;
    this.clazz = clazz;
    this.subject = subject;
    this.course = course;
    this.room = room;
    this.substituted = substituted;
    this.substitute_teacher = substitute_teacher;
    this.substitute_room = substitute_room;
}
```

Android App Entwicklung

26.6.2021
-XX.7.2021

Implementierung über:

Flutter (Framework) und Dart (Programmiersprache)

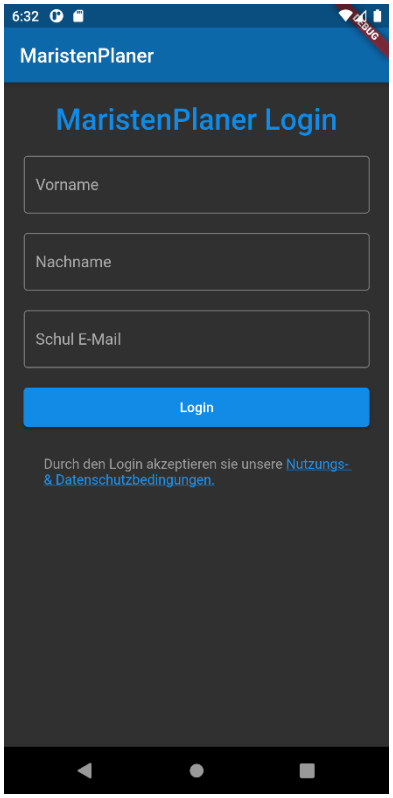
Funktionen:

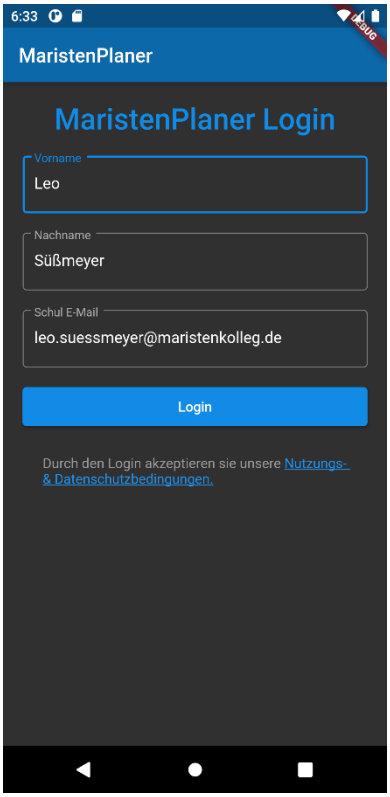
- Vertretungsplan ansehen
- Stundenplan ansehen
- Mensa Menüplan

Schritte:

- Stundenplan API Request — 26.6.2021
  - Die App stellt eine Request an den Server, der der App dann den Stundenplan übermittelt
- Navigation Bar — 29.6.2021
  - Design und Funktion der Navigation Bar hinzugefügt
- About „App“ Bereich mit Lizenzen — 30.6.2021
  - Beinhaltet alle Lizenzen
- Server Fix wegen API request — 2.7.2021
  - Bugfix
- Dark Mode
  - Design; Kompatibilität zum Dark Mode
- Stundenplan schöner machen
  - Design fix
- Einbau erster Settings — 4.7.2021
  - Hinzufügen eines Tabs „Settings“ mit noch nicht vielen Funktionen
- Login und Authentication Screen
  - Implementierung des Login- und Authentificationscreens
  - System siehe „Anmeldung“
- Refresh Funktion — 6.7.2021
  - Neu laden des, bis jetzt, Stundenplans
- Sharred preferences
  - Hinzufügen, dass man sich nicht bei jedem öffnen der App neu einloggen muss (App resettet die Nutzerdaten nicht)
- getSchedule Fix — 17.9.2021
  - s.o. Probleme
- Anmeldung statt Name, Nachname wir jetzt auch E-Mail als Anmeldemittel verwendet

Funktionsweise:

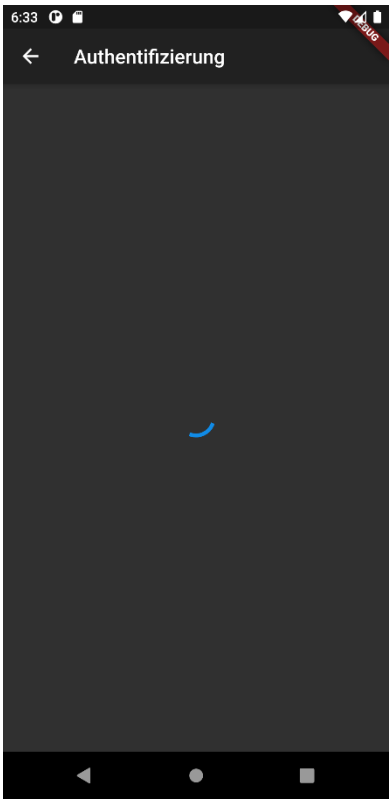


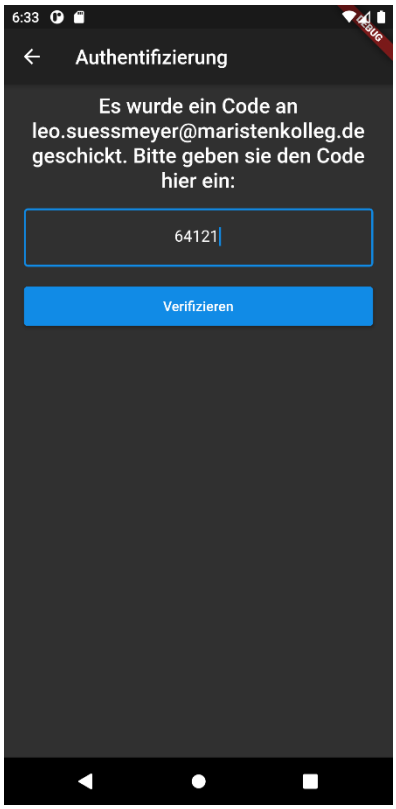


Bei Start wird die Login-page angezeigt.

Sobald der Name eingegeben ist, wird unten automatisch die E-Mail generiert, wobei auch Sonderzeichen etc. berücksichtigt werden (ü = ue).

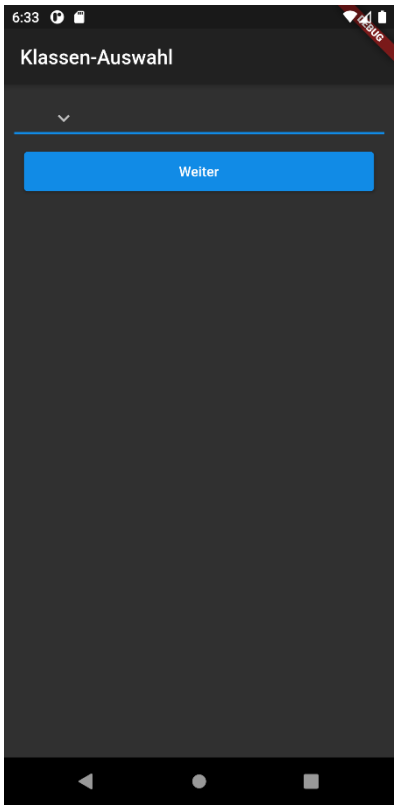
Die E-Mail wir geschickt so bald auf „Login“ getippt wurde, es öffnet sich das „Authentifizierung“ Fenster.

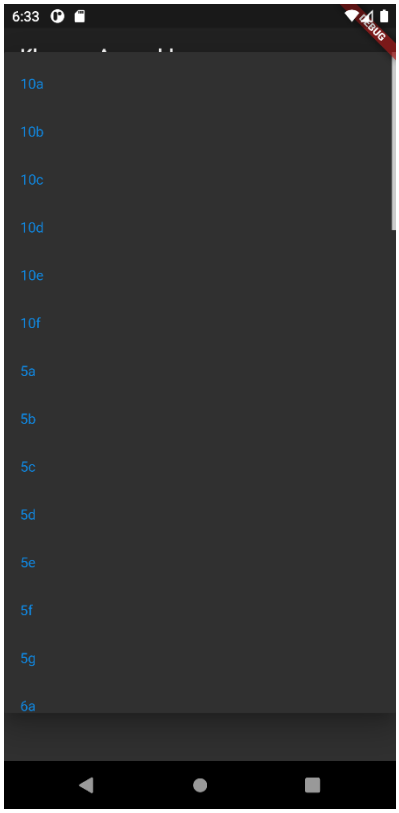


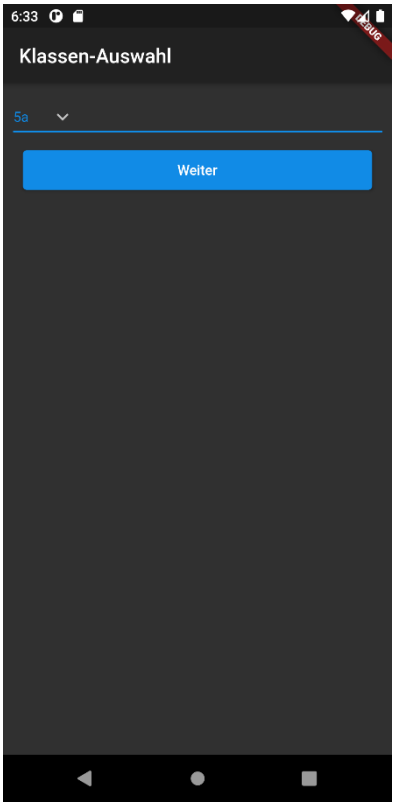


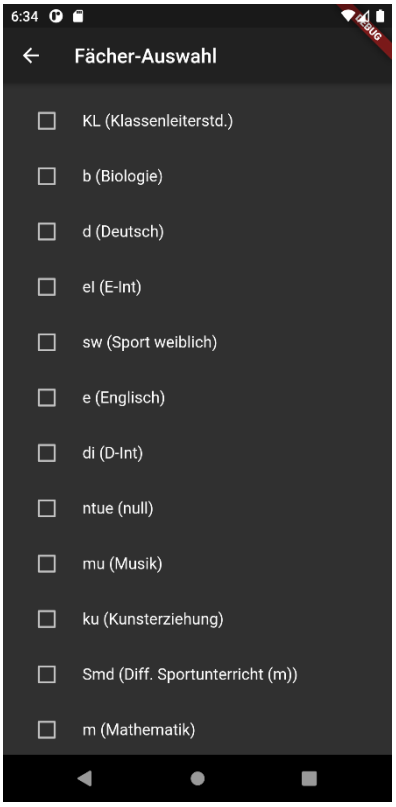
Im Authentifizierungsfenster muss nur der Code eingegeben und auf den Button „Verifizieren“ getippt werden.

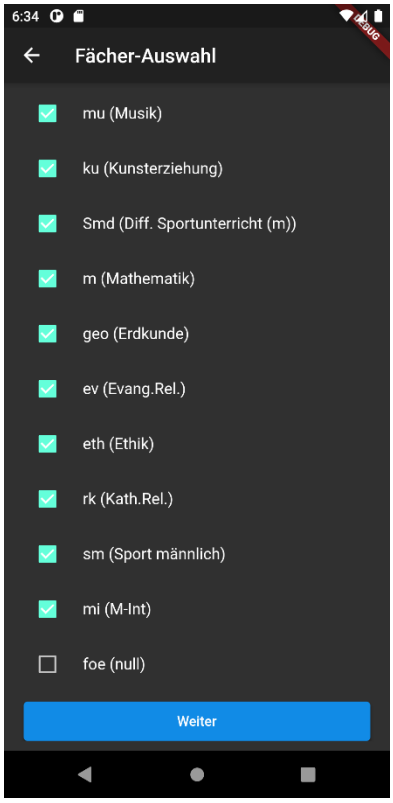
Danach öffnet sich die Klassen- und Fächerauswahl.



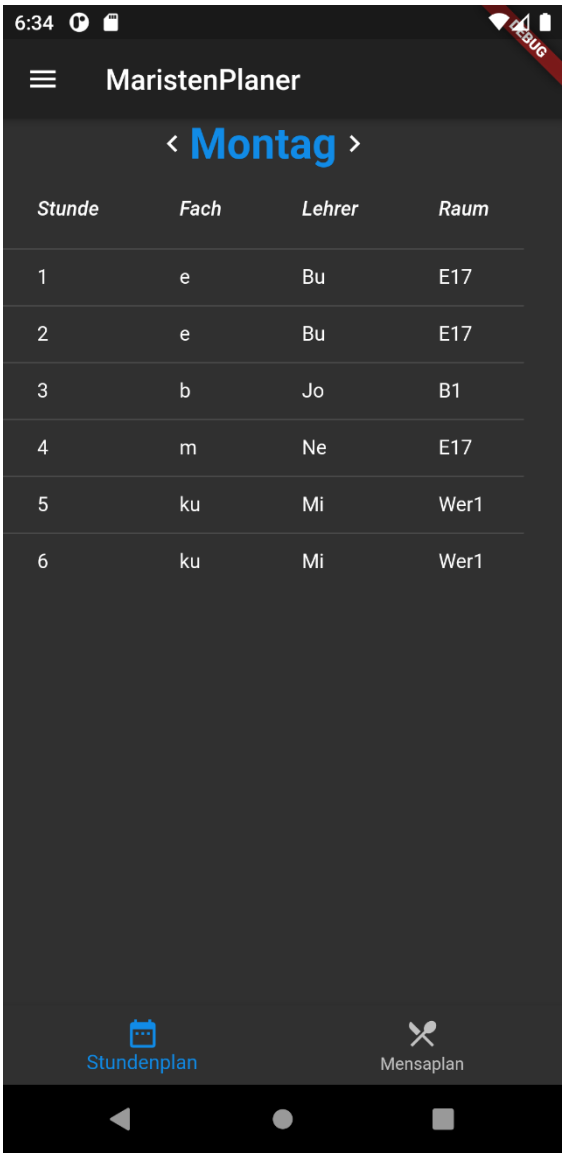








Nachdem die Klasse und die Fächer ausgewählt wurden, wird die Startseite der App geöffnet- der Stundenplan



Der Stundenplan wird jedem Schüler je nach Wochentag angezeigt. Wenn man auf „Menaplan“ tippt, wird die aktuelle Mensaplan-pdf angezeigt.

MaristenConsole

Funktionen:

- Vertretungen eintragen
- Lehrerstundenpläne einsehen/ändern
- Lehrerliste einsehen/ändern
- Klassenlisten einsehen/ändern
- Raumplan einsehen/ändern

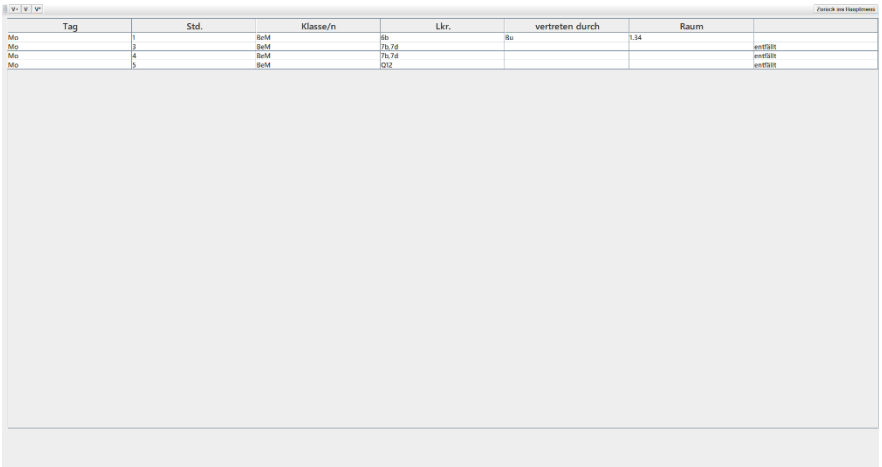
Implementierung über:

IntelliJ in Java

Funktionen und Erklärung

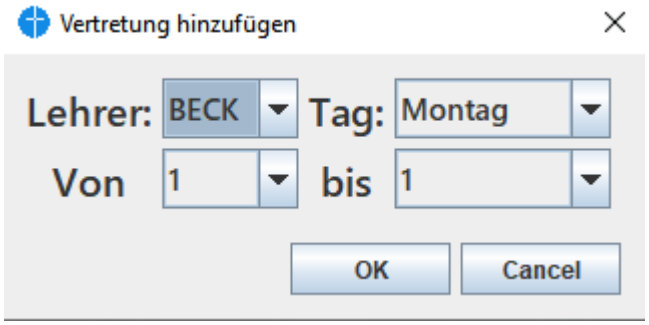
- Erstellung eines Designs
    - Über Figma.com

- Vertretungen

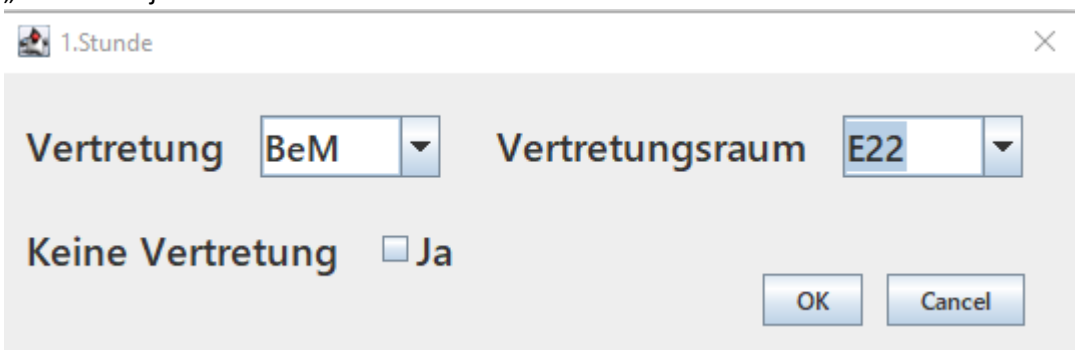


Tag	Std.	Klasse/n	Lkr.	vertreten durch	Raum	
Mo	1	8eM	6b	Su	1.34	
Mo	3	8eM	7b.7d			entfällt
Mo	4	8eM	7b.7d			entfällt
Mo	5	8eM	Q12			entfällt

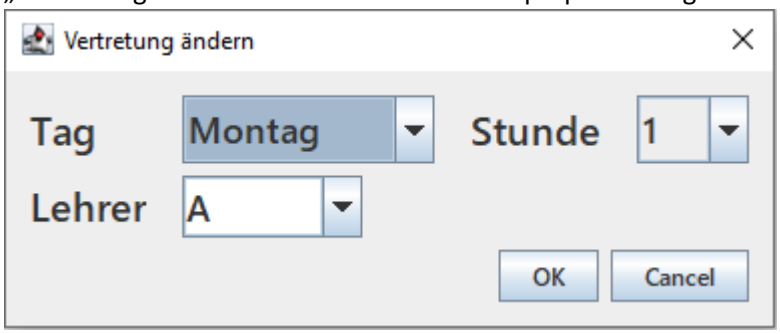
- „+“ Pop-up Fenster wird geöffnet (JDialog)



- „OK“ öffnet jeweils ein weiteres Fenster



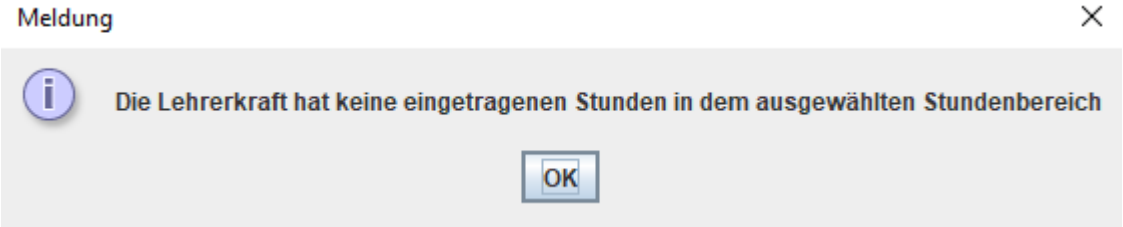
- Jede Stunde des Lehrers wird durchgegangen und muss einzeln bestätigt werden
- Jede Änderung wird sofort in der Datenbank gespeichert
- „Vertretungen ändern“ wird ein anderes Pop-up Fenster geöffnet



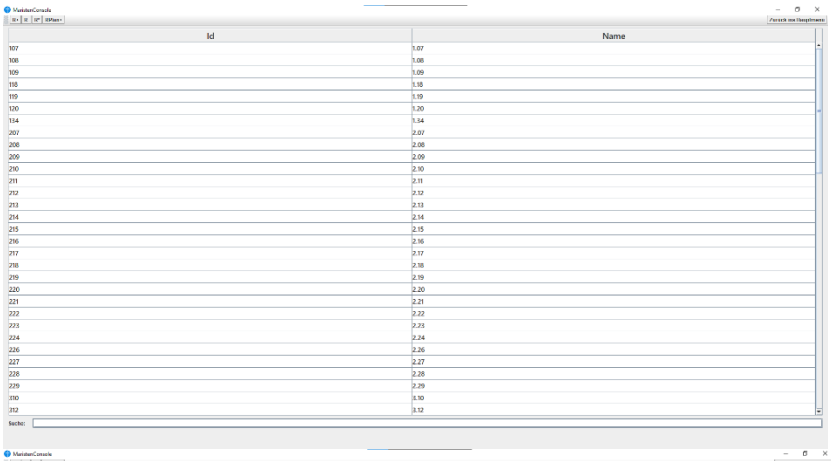
- Fehlermeldung Pop-up Fenster

  Die Fehlermeldung Pop-up Fenster öffnen sich sobald ein Fehler jeglicher Art erkannt wird.

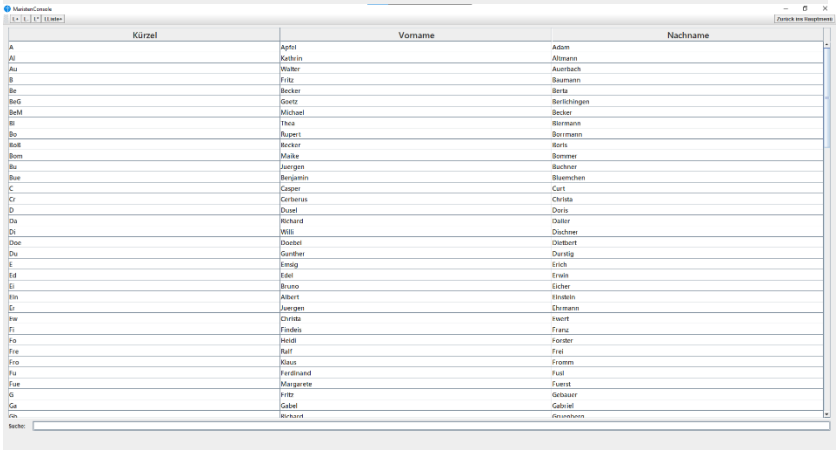
  Beispiel:



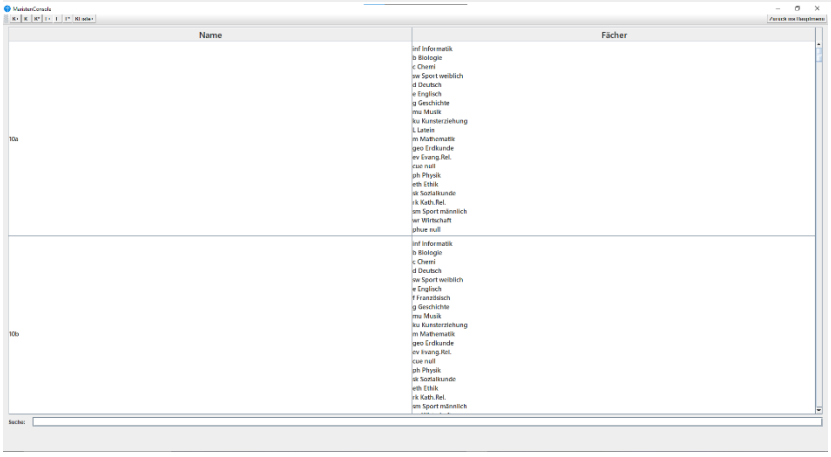
- Lehrerliste/Raumplan/Klassenliste (funktionieren auf die gleiche Weiße)



Id	Name
107	1.07
108	1.08
109	1.09
118	1.18
119	1.19
120	1.20
134	1.34
207	2.07
208	2.08
209	2.09
210	2.10
211	2.11
212	2.12
213	2.13
214	2.14
215	2.15
216	2.16
217	2.17
218	2.18
219	2.19
220	2.20
221	2.21
222	2.22
223	2.23
224	2.24
226	2.26
227	2.27
228	2.28
229	2.29
110	1.10
112	1.12



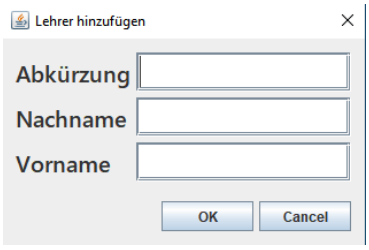
Kürzel	Vorname	Nachname
A	Apfel	Adam
Al	Kathrin	Altmann
Au	Walter	Auerbach
B	Fritz	Baumann
Be	Becker	Berta
BeG	Goetz	Berlichingen
BeM	Michael	Becker
Bl	Thea	Biermann
Bo	Rupert	Borrmann
Bod	Becker	Boris
Bom	Maike	Bommer
Bu	Juergen	Buchner
Bue	Benjamin	Bluemchen
C	Casper	Curt
Cr	Cerberus	Christa
D	Dusel	Doris
Da	Richard	Dalter
Di	Willi	Dischner
Doe	Doerbel	Dietbert
Du	Gunther	Durstig
E	Emsig	Erich
Ed	Edel	Erwin
Ei	Bruno	Eicher
Ein	Albert	Einstein
Er	Juergen	Ehrmann
Ew	Christa	Ewert
Fi	Findeis	Franz
Fo	Heidi	Forster
Fre	Ralf	Frei
Fro	Klaus	Fromm
Fu	Ferdinand	Fusl
Fue	Margarete	Fuerst
G	Fritz	Gebauer
Ga	Gabel	Gabriel
Ge	Richard	Gesundheit



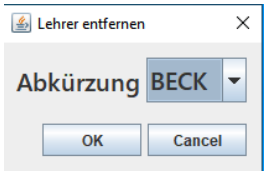
Name	Fächer
10a	inf Informatik b Biologie c Chemi sw Sport weiblich d Deutsch e Englisch g Geschichte mu Musik ku Kunsterziehung L Latein m Mathematik geo Erdkunde ev Evang.Rel. cue null ph Physik eth Ethik sk Sozialkunde k Kath.Rel. sm Sport männlich w Wirtschaft phue null
10b	inf Informatik b Biologie c Chemi d Deutsch sw Sport weiblich e Englisch f Französisch g Geschichte mu Musik ku Kunsterziehung m Mathematik geo Erdkunde ev Evang.Rel. cue null ph Physik sk Sozialkunde eth Ethik k Kath.Rel. sm Sport männlich

  - Jeweils id/ Kürzel/ Name der Klasse
    und Raum Name/ Nachname, Vorname/ Fächer

    Beispiel Lehrerliste:
  - „L+“ Fenster wird geöffnet

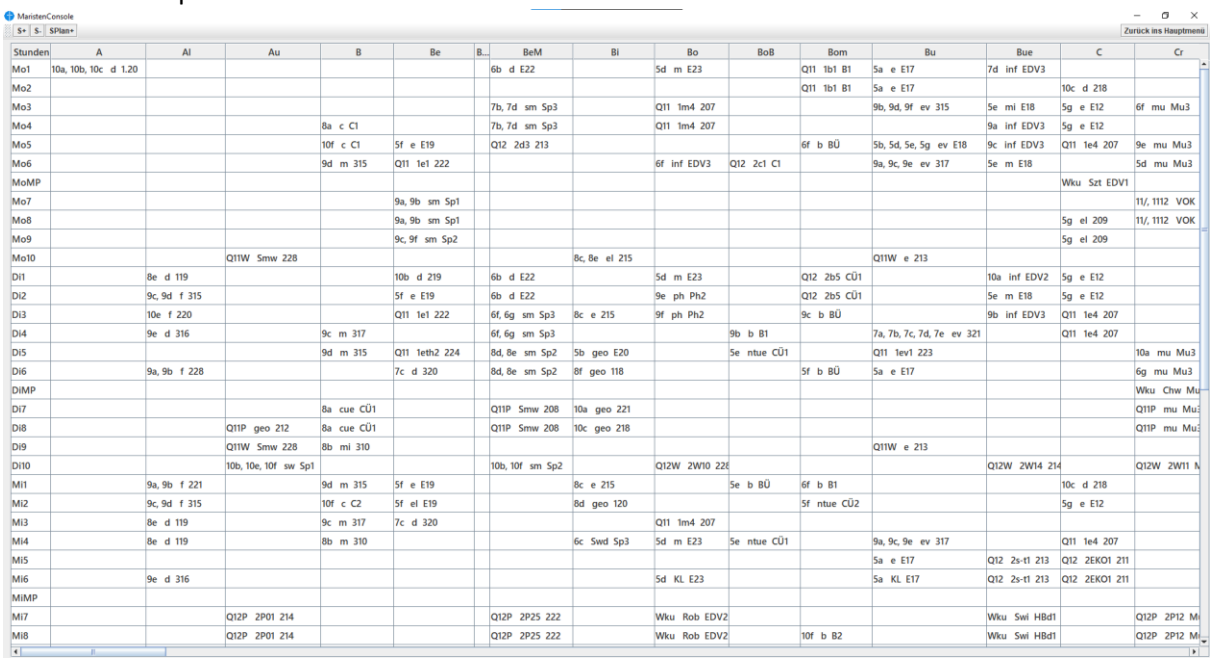


- Wird sofort in der Datenbank gespeichert und Tabelle wird aktualisiert
- „L– „Fenster wird geöffnet



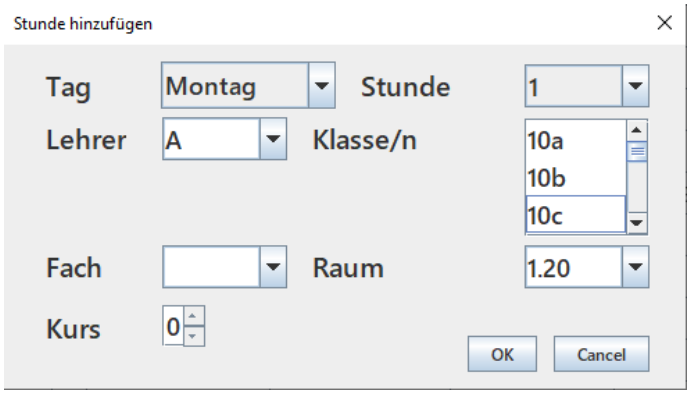
- Wird sofort in der Datenbank gespeichert und Tabelle wird aktualisiert

- Lehrerstundenpläne

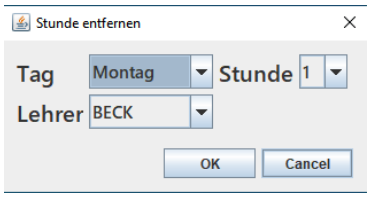


Stunden	A	Al	Au	B	Be	B...	BeM	Bi	Bo	BoB	Bom	Bu	Bue	C	Cr
Mo1	10a, 10b, 10c d 1.20						6b d E22		5d m E23		Q11 1b1 B1	5a e E17	7d inf EDV3		
Mo2											Q11 1b1 B1	5a e E17		10c d 218	
Mo3							7b, 7d sm Sp3		Q11 1m4 207			9b, 9d, 9f ev 315	5e mi E18	5g e E12	6f mu Mu3
Mo4				8a c C1			7b, 7d sm Sp3		Q11 1m4 207				9a inf EDV3	5g e E12	
Mo5				10f c C1	5f e E19		Q12 2d3 213				6f b BÜ	5b, 5d, 5e, 5g ev E18	9c inf EDV3	Q11 1e4 207	9e mu Mu3
Mo6				9d m 315	Q11 1e1 222				6f inf EDV3	Q12 2c1 C1		9a, 9c, 9e ev 317	5e m E18		5d mu Mu3
MoMP														Wku Szt EDV1	
Mo7					9a, 9b sm Sp1										11/, 1112 VOK
Mo8					9a, 9b sm Sp1									5g el 209	11/, 1112 VOK
Mo9					9c, 9f sm Sp2									5g el 209	
Mo10			Q11W Smw 228					8c, 8e el 215				Q11W e 213			
Di1		8e d 119			10b d 219		6b d E22		5d m E23		Q12 2b5 CÜ1		10a inf EDV2	5g e E12	
Di2		9c, 9d f 315			5f e E19		6b d E22		9e ph Ph2		Q12 2b5 CÜ1		5e m E18	5g e E12	
Di3		10e f 220			Q11 1e1 222		6f, 6g sm Sp3	8c e 215	9f ph Ph2		9c b BÜ		9b inf EDV3	Q11 1e4 207	
Di4		9e d 316		9c m 317			6f, 6g sm Sp3			9b b B1		7a, 7b, 7c, 7d, 7e ev 321		Q11 1e4 207	
Di5				9d m 315	Q11 1eth2 224		8d, 8e sm Sp2	5b geo E20		5e ntue CÜ1		Q11 1ev1 223			10a mu Mu3
Di6		9a, 9b f 228			7c d 320		8d, 8e sm Sp2	8f geo 118			5f b BÜ	5a e E17			6g mu Mu3
DiMP															Wku Chw Mu
Di7				8a cue CÜ1			Q11P Smw 208	10a geo 221							Q11P mu Mu
Di8			Q11P geo 212	8a cue CÜ1			Q11P Smw 208	10c geo 218							Q11P mu Mu
Di9			Q11W Smw 228	8b mi 310								Q11W e 213			
Di10			10b, 10e, 10f sw Sp1				10b, 10f sm Sp2		Q12W 2W10 22				Q12W 2W14 21		Q12W 2W11 M
Mi1		9a, 9b f 221		9d m 315	5f e E19			8c e 215		5e b BÜ	6f b B1			10c d 218	
Mi2		9c, 9d f 315		10f c C2	5f el E19			8d geo 120			5f ntue CÜ2			5g e E12	
Mi3		8e d 119		9c m 317	7c d 320				Q11 1m4 207						
Mi4		8e d 119		8b m 310				6c Swd Sp3	5d m E23	5e ntue CÜ1		9a, 9c, 9e ev 317		Q11 1e4 207	
Mi5												5a e E17	Q12 2s-t1 213	Q12 2EKO1 211	
Mi6		9e d 316							5d KL E23			5a KL E17	Q12 2s-t1 213	Q12 2EKO1 211	
MiMP															
Mi7			Q12P 2P01 214				Q12P 2P25 222		Wku Rob EDV2				Wku Swi HBd1		Q12P 2P12 M
Mi8			Q12P 2P01 214				Q12P 2P25 222		Wku Rob EDV2		10f b B2		Wku Swi HBd1		Q12P 2P12 M

- Werden als J-Table angezeigt
- „+“ Fenster wird geöffnet

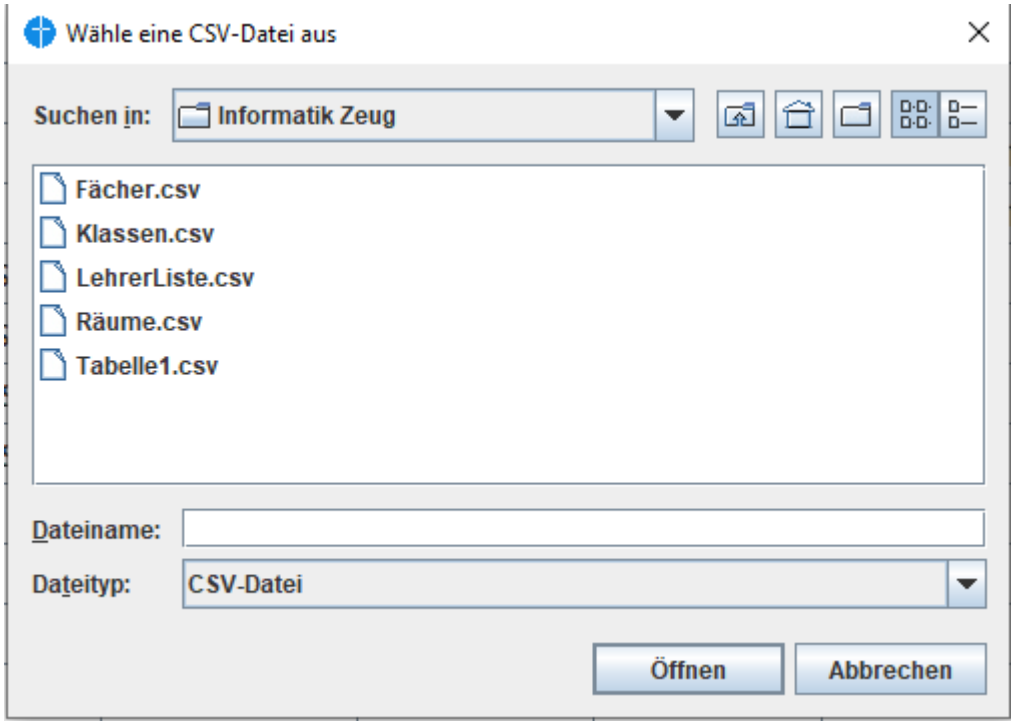


- Wird sofort in der Datenbank gespeichert und Tabelle wird aktualisiert
- „–„ Fenster wird geöffnet



- Wird sofort in der Datenbank gespeichert und Tabelle wird aktualisiert

- Import der Dateien
  - Auf SPlan+ Fenster wird geöffnet:



Für manche Pläne müssen mehrere Dateien importiert werden, Es gibt keine Gesamtdatei, die man aus Willy2 exportieren kann.

  - .csv Datei (aus Willy2 siehe oben) wir importiert und es wird eine Fortschritts-Anzeige angezeigt.