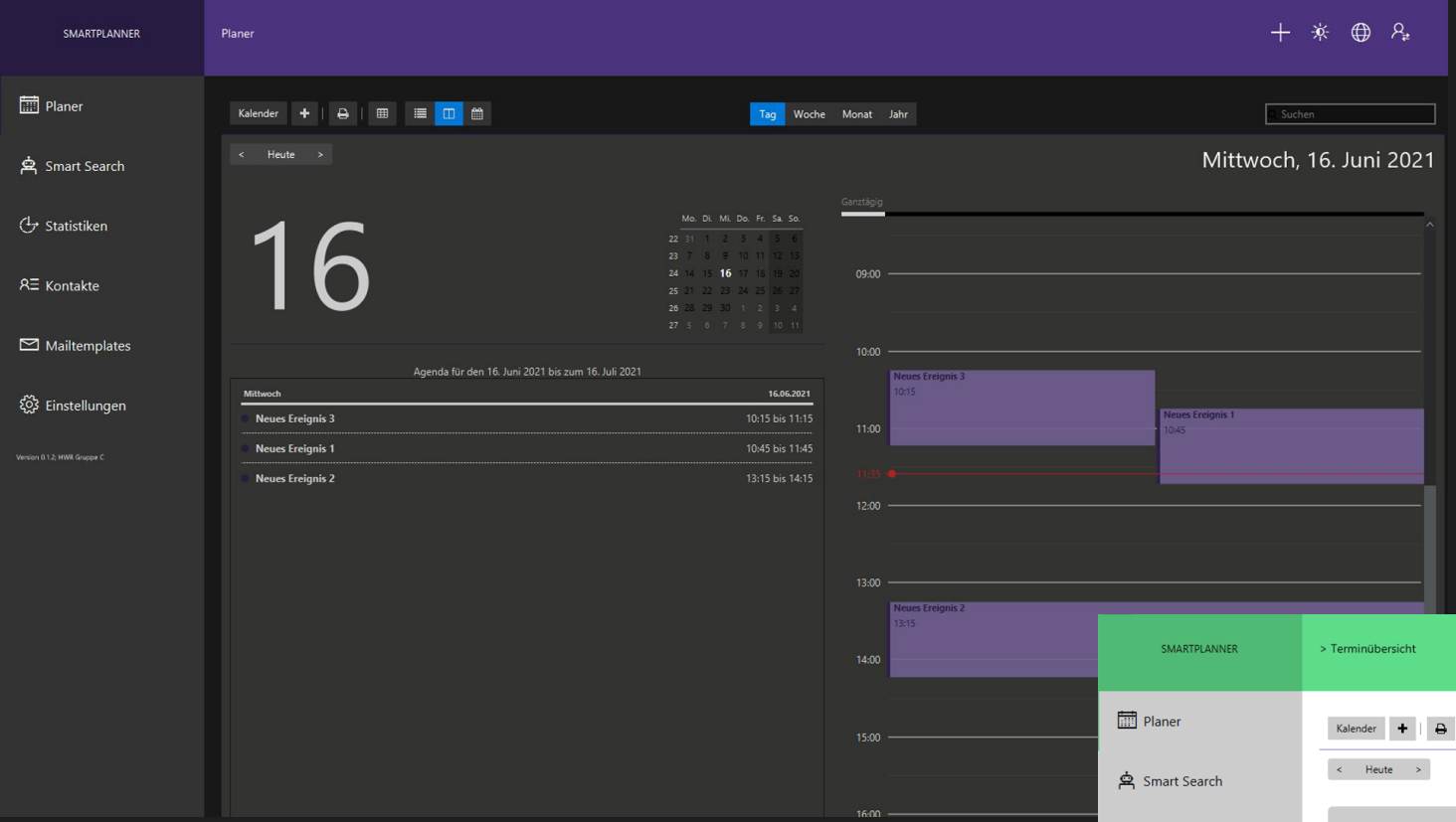
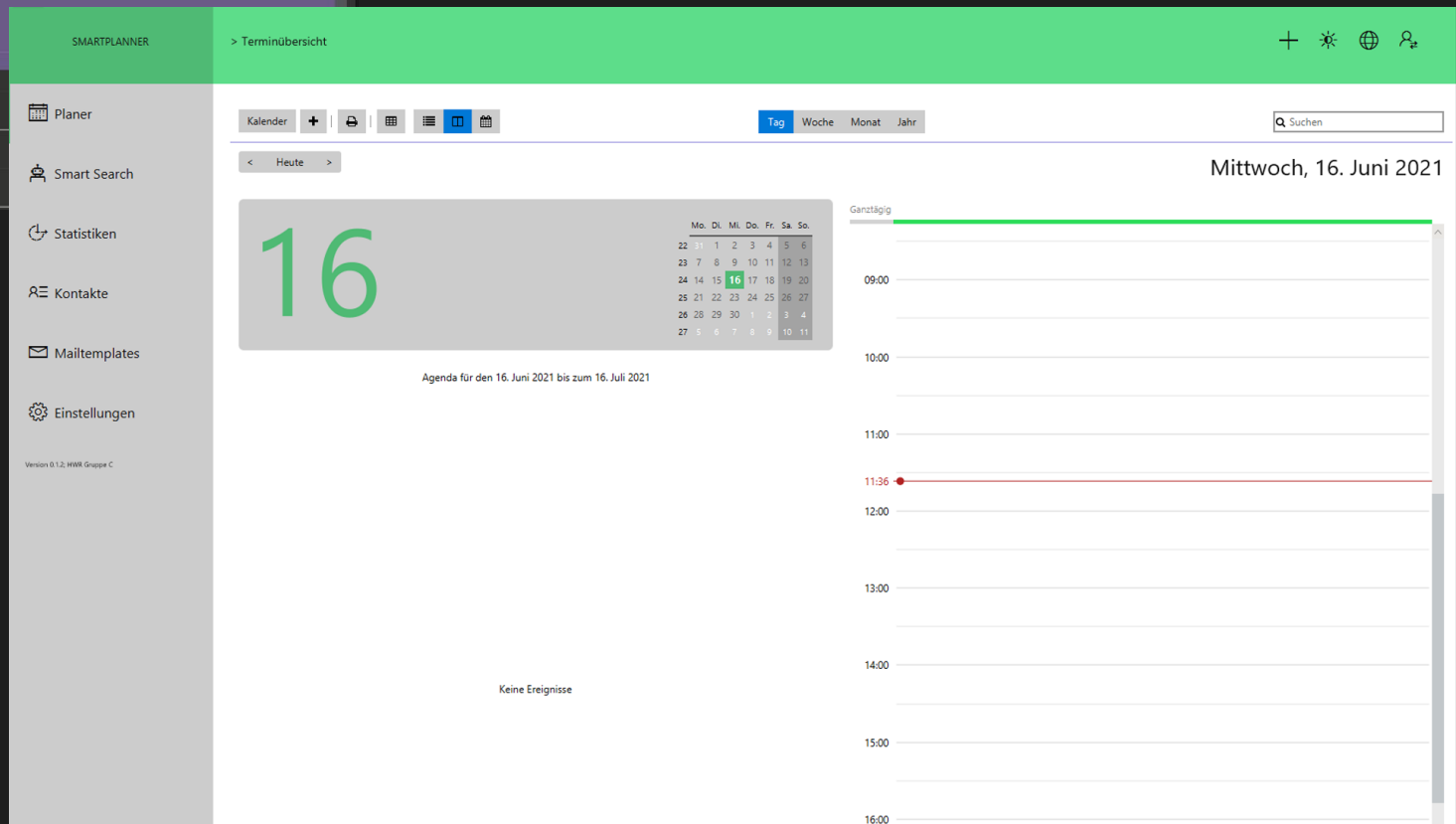


SMART
PLANNER
(Arbeitstitel)

DevTeam:

Kamilla
Danny
Corentin
Nico R.
Robert
Nico S.
Hannes



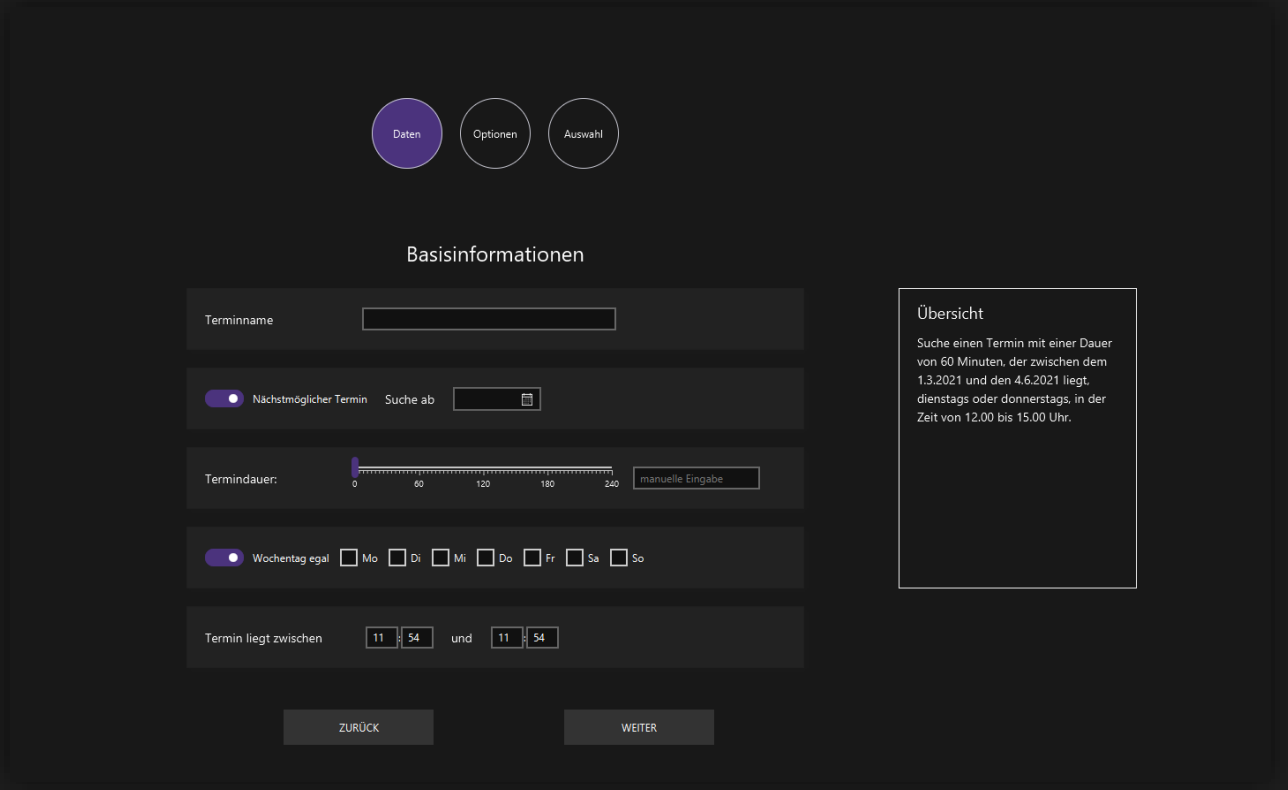
Gliederung

<> Analysephase (Robert): Gesamtübersicht der Usecases als Usecasediagramm, „Hauptusecase“ Smarte Suche als Aktivitätsdiagramm

<> Entwurfsphase (Nico): UML-Klassendiagramm und Sequenzdiagramm der Smarten Suche

<> Architektur (Danny): Nutzung des MVC Modells, abstrahierte Darstellung der Zusammenhänge der Funktionen

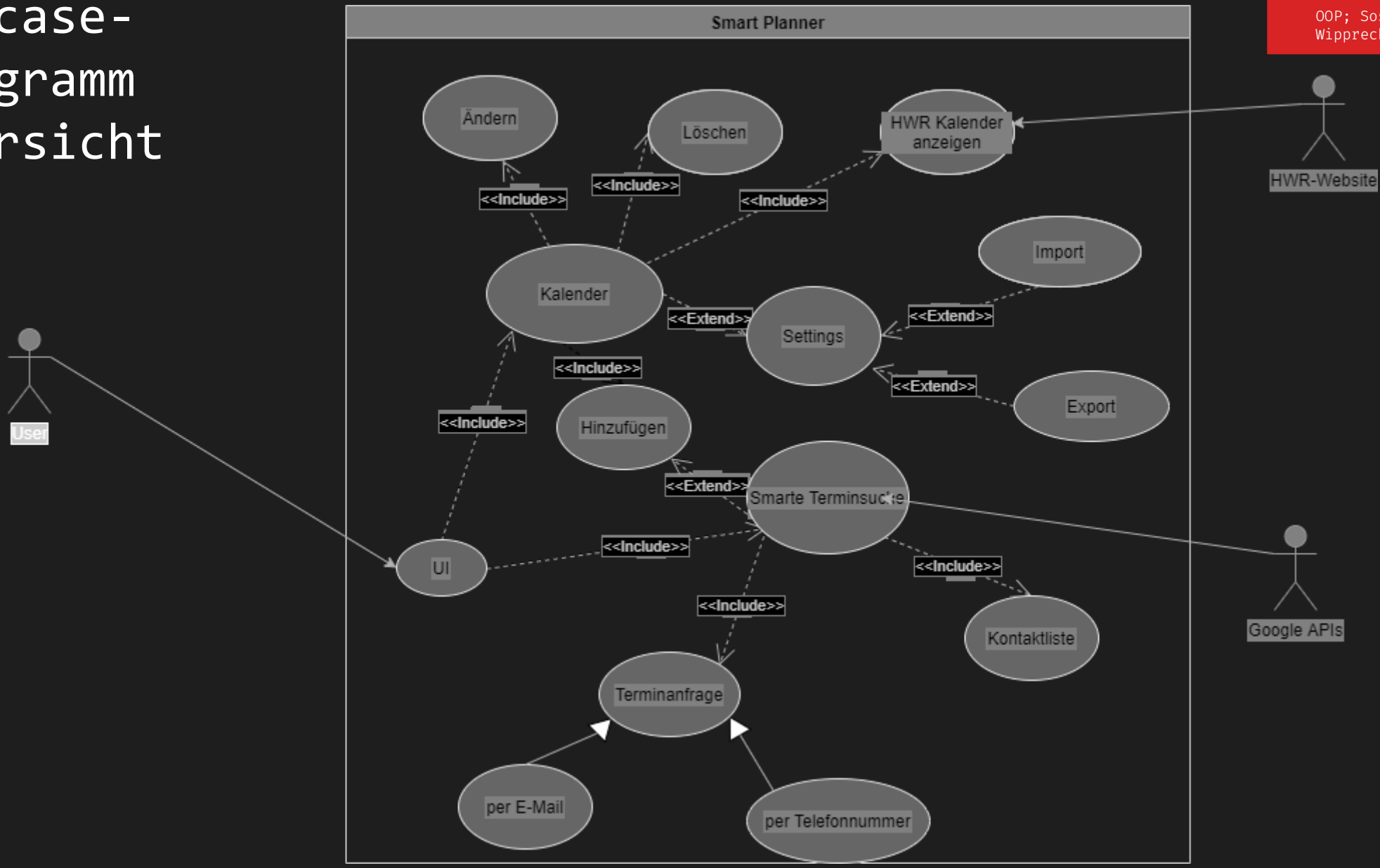
<> Architektur (Hannes): genutzte Patterns und Programmfluss, Module und externe Abhängigkeiten



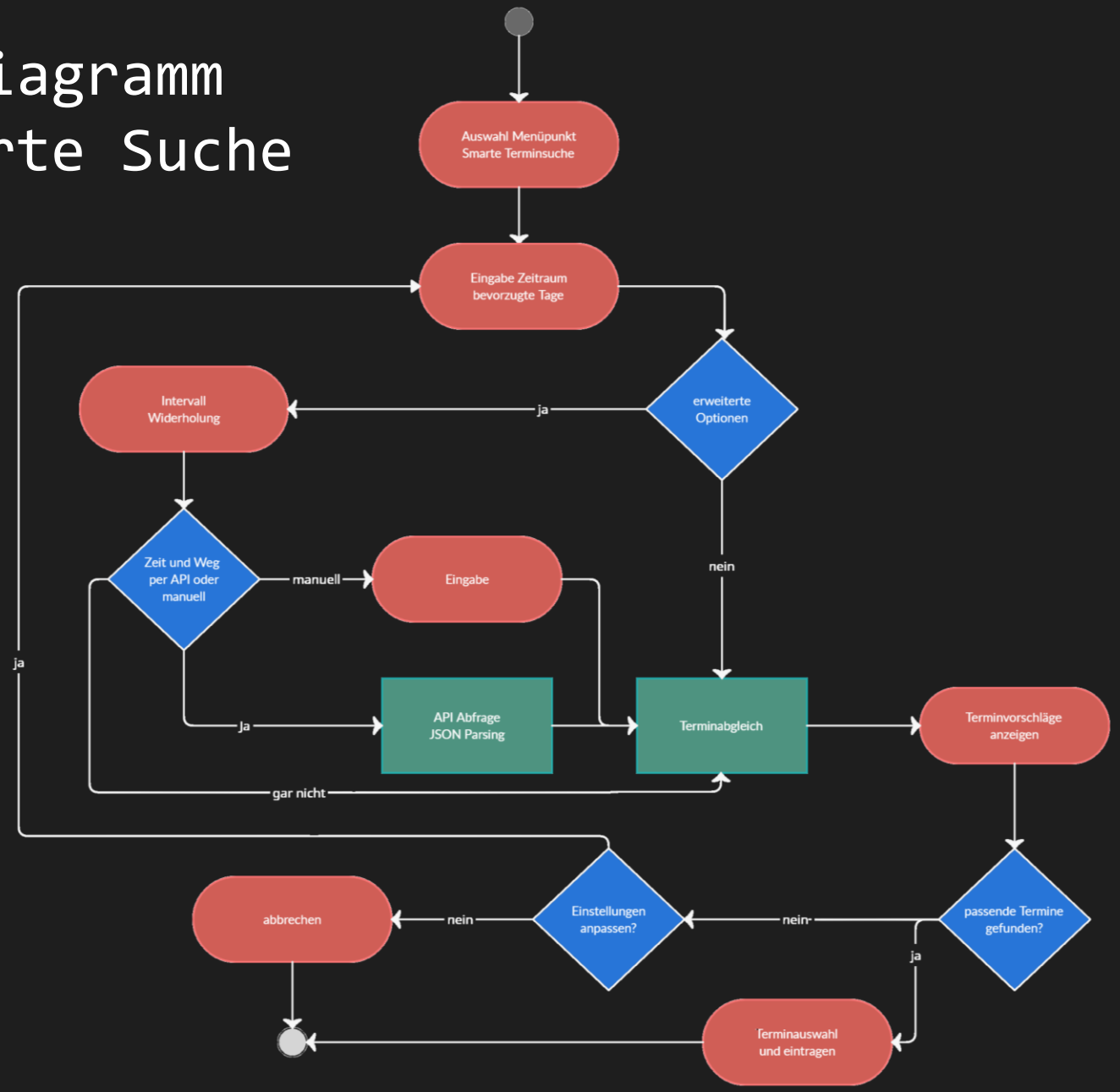
Motivation und Zielstellung

Motivation	Zielstellungen/Hauptanforderungen
vereinfachter Abruf des HWR Kalenders, der sich ständig ändert	eigene Kalenderapp, die regelmäßig die Änderungen crawlt (Intervall oder bei Änderung) und darstellt
Erinnerung und Buchung regelmäßiger Termine (Zahnarzt), die oft vergessen werden	**dynamische Suche nach freien Terminen anhand vorhandener Termine und Vereinbarung des Termins per Mailanfrage (automatische Generierung der Mail)**
Terminsuche oft schwierig bzw. manuell langwierig	**smarte Vorschlagsfunktion die z.B. auch Fahrtzeiten mit einbezieht und automatisch Termine an freien Slots vorschlägt bzw. direkt einträgt**
verteilte Kalender nerven und sind in der UX nicht konsistent (Google, HWR, privat-mobile, Arbeit-Outlook)	durch Import eigener und des HWR Kalenders entsteht eine zentrale Verwaltung der Termine. Fremde Kalender können auch in der smarten Suche genutzt werden und anschließend exportiert werden.

Usecase-Diagramm Übersicht



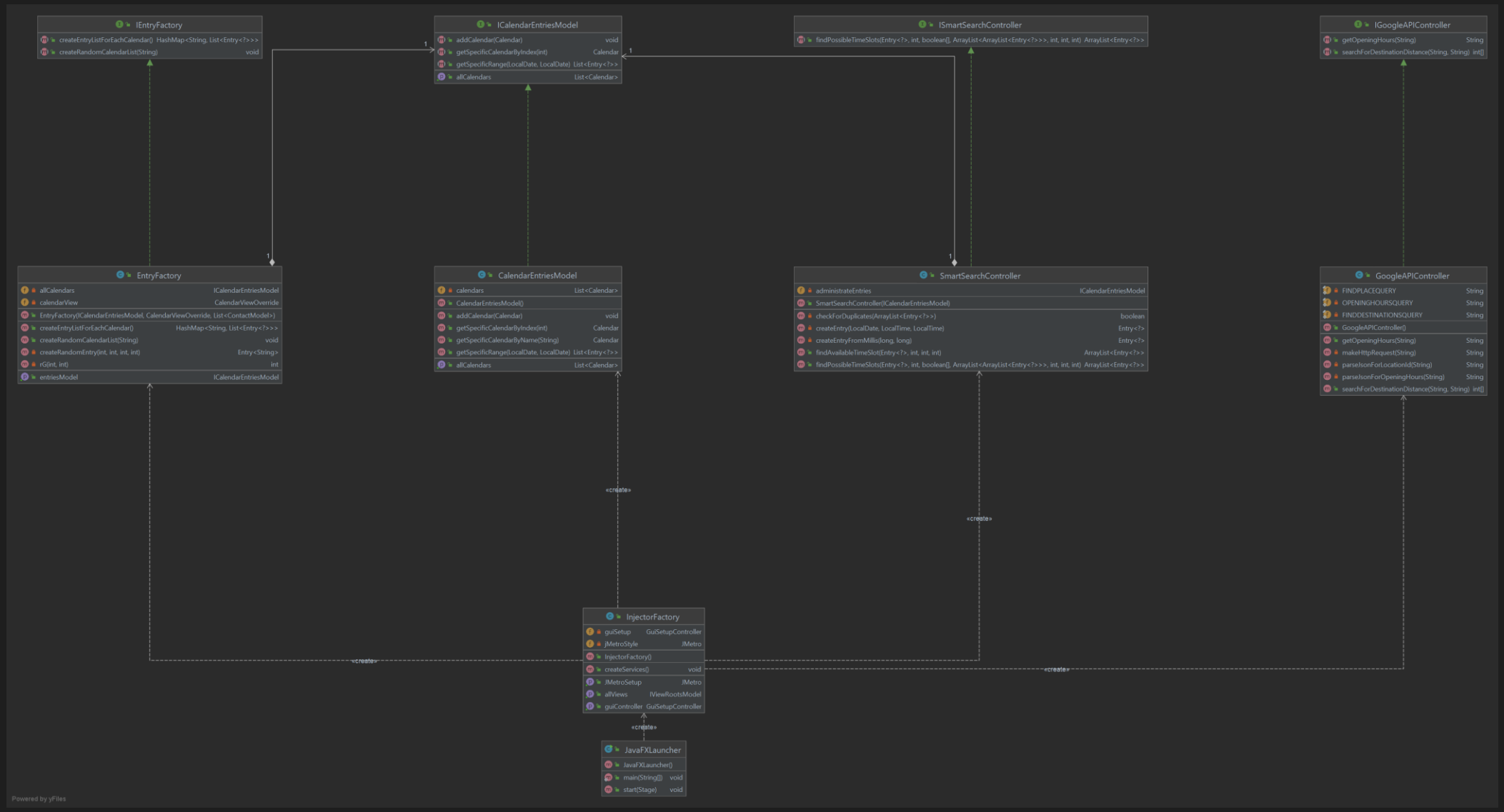
Aktivitätsdiagramm
Usecase Smarte Suche



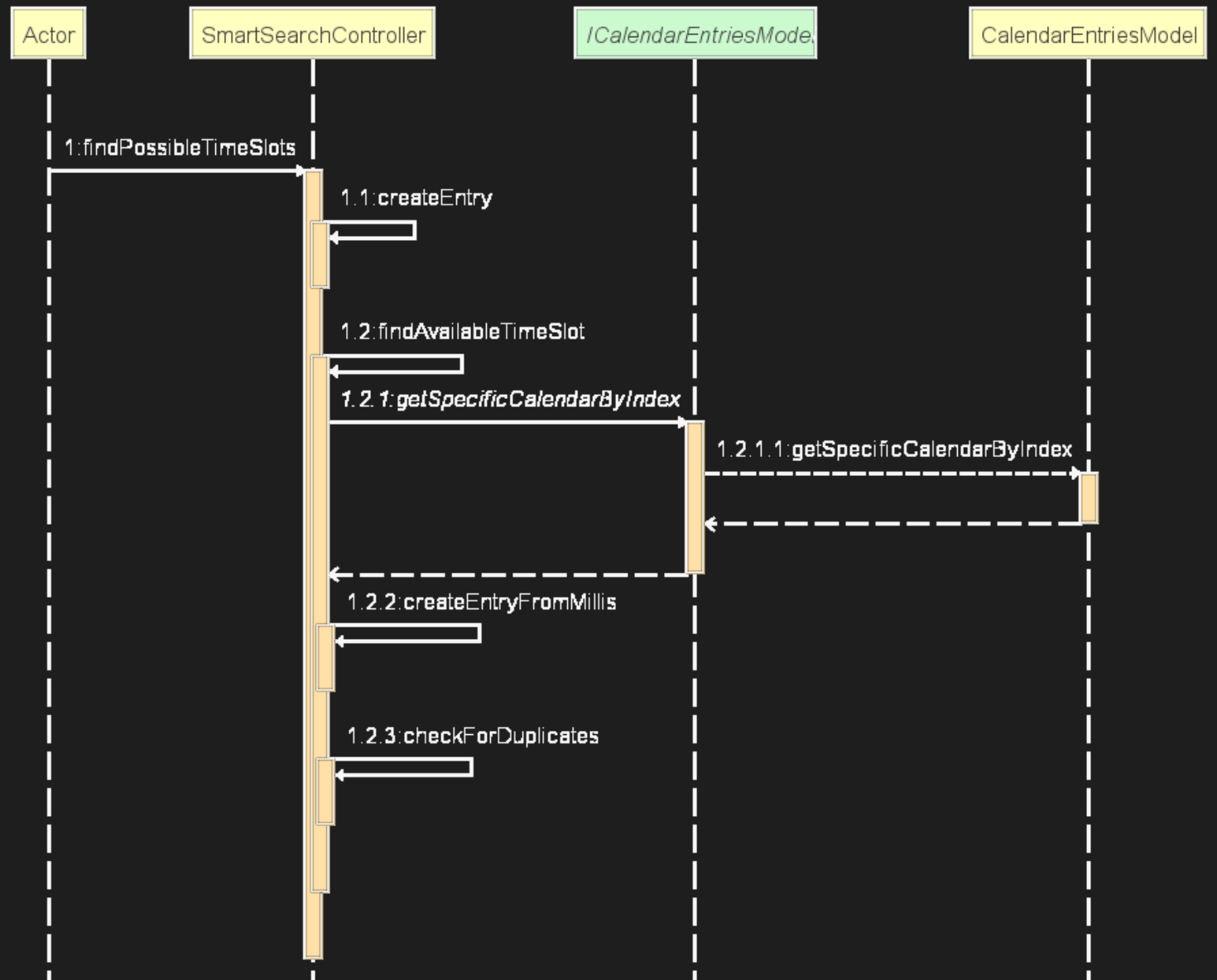
UML – Klassendiagramm Usecase Smarte Suche

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

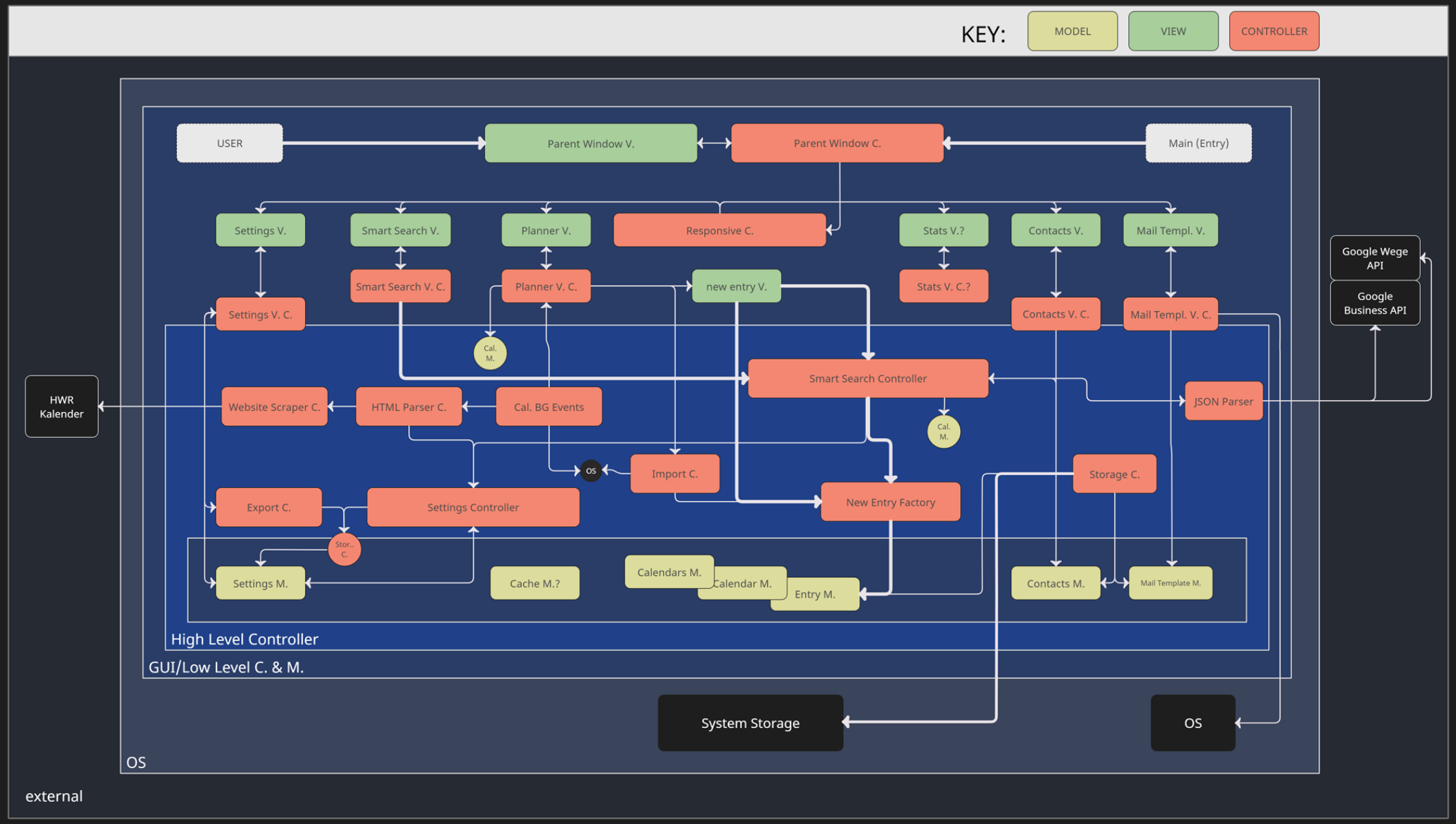
OOP; Sose 2021
Wipprecht/Dimitrov



Sequenzdiagramm Usecase Smarte Suche

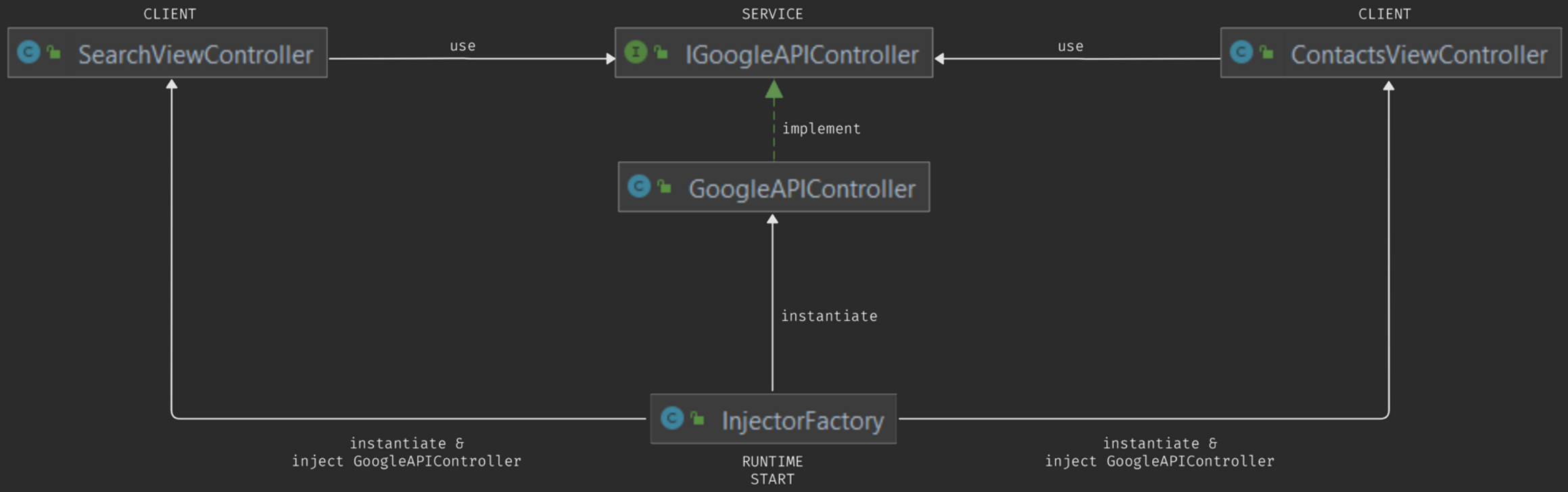


MVC – Modell und Funktionalitäten



Inversion of Control Principle (Fowler, Martin)

(hier implementiert durch Dependency Injection als Design Pattern)



```
public void createServices() throws IOException
{
    jMetroStyle = new JMetro();
    var customCalendarView = new CalendarViewOverride();
    var mailTemplates = new ArrayList<MailTemplateModel>();
    var contacts = new ArrayList<ContactModel>();
    var settings = new SettingsModel();

    IGoogleAPIController apiCt = new GoogleAPIController();
    IMailCreationController mailCreationCt = new MailCreationController(mailTemplates);
    ICalendarEntriesModel calendarEntriesModel = new CalendarEntriesModel();
    IContactFactory contactFactory = new ContactFactory(contacts);
    IWebsiteScraperController websiteCt = new WebsiteScraperController(settings);
    ISmartSearchController smartSearch = new SmartSearchController(calendarEntriesModel);
    IEntryFactory appointmentEntryCreator = new EntryFactory(calendarEntriesModel, customCalendarView, contacts);
    IIOController ioCt = new IOController(appointmentEntryCreator, contacts, settings, mailTemplates, calendarEntriesModel);
    IExportController exportCt = new ExportController(appointmentEntryCreator, contacts, settings, mailTemplates, calendarEntriesModel);
    IImportController importCt = new ImportController(appointmentEntryCreator, contacts, settings, mailTemplates, calendarEntriesModel);

    var settingsVCt = new SettingsViewController(settings);
    var statsVCt = new StatsViewController(contacts, calendarEntriesModel);
    var contactsVCt = new ContactsViewController(contacts, contactFactory, apiCt, ioCt);
    var mailVCt = new MailTemplateViewController(ioCt, settings, mailCreationCt, contacts, mailTemplates);
    var searchVCt = new SearchViewController(smartSearch, appointmentEntryCreator, contacts, contactFactory, mailTemplates, settings, apiCt,
    var plannerVCt = new PlannerViewController(calendarEntriesModel, appointmentEntryCreator, importCt, exportCt, customCalendarView);
    allViews = new ViewRootsModel(plannerVCt, searchVCt, statsVCt, contactsVCt, mailVCt, settingsVCt);
    guiSetup = new GuiSetupController(jMetroStyle, allViews);

    guiSetup.init();
    //ioCt.loadCalendarsFromFile();
    settings.addPropertyChangeListener(new ChangeListener());
    //websiteCt.startScraperTask();
}
```

DI am Beispiel der bei Start erzeugten Factory

letzte Folie: Gesamtarchitektur
derzeitiger Stand als extra Datei
(umlkomplett.png)