Mobile BuN

Kalender-App

Wintersemester 24/25

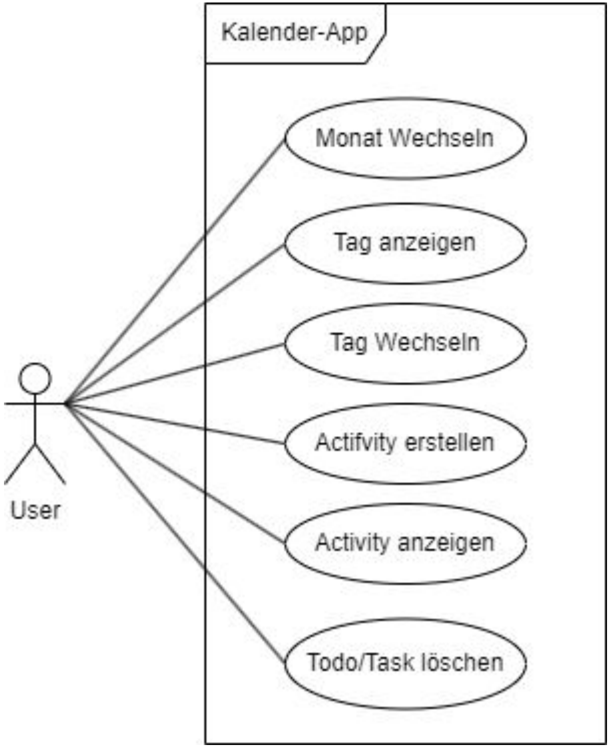
Sean Bien (589477)

Hochschule für Technik
und Wirtschaft Berlin

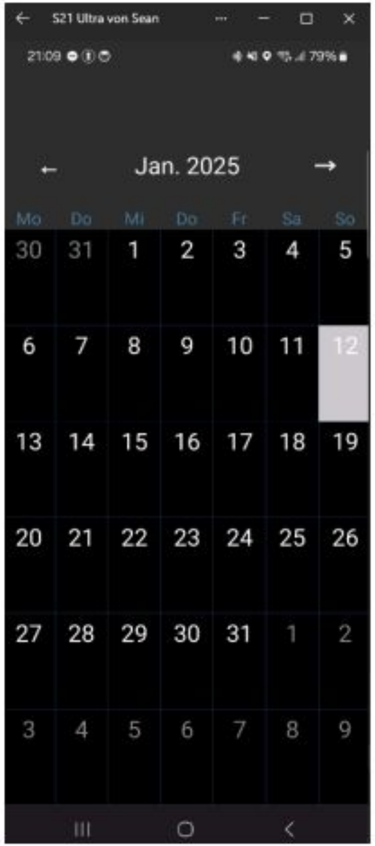
University of Applied Sciences

Was ist die Kalender-App?

- Navigation druch Monat/Tage
- Erstellen, anzeigen und löschen von Aktivitäten
- Besonderheit:
  - Einbindung von OSMdroid,
  - Nutzung von Recycler- und Listviews für die erstellten Tasks und die Erstellung des Kalenders



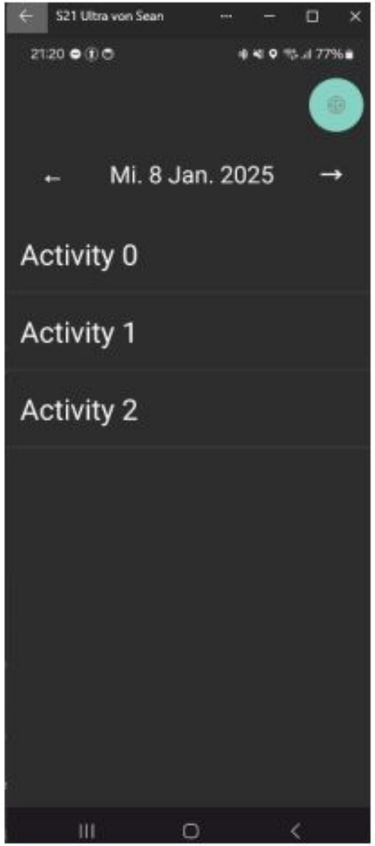
GUI



Jan. 2025

Mo	Do	Mi	Do	Fr	Sa	So
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

CalenderView



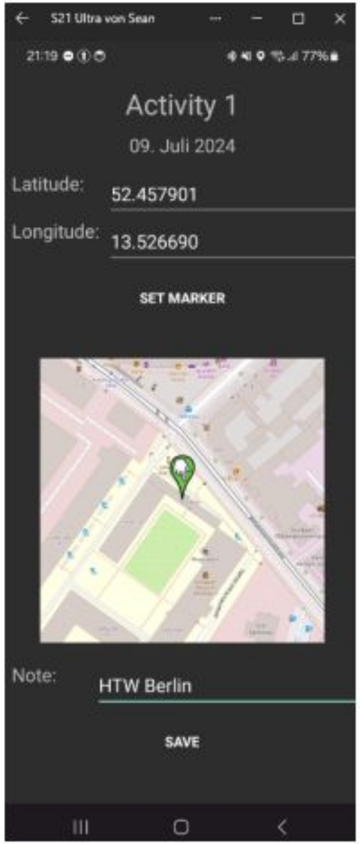
Mi. 8 Jan. 2025

Activity 0

Activity 1

Activity 2

DayView



Activity 1

09. Juli 2024

Latitude: 52.457901

Longitude: 13.526690

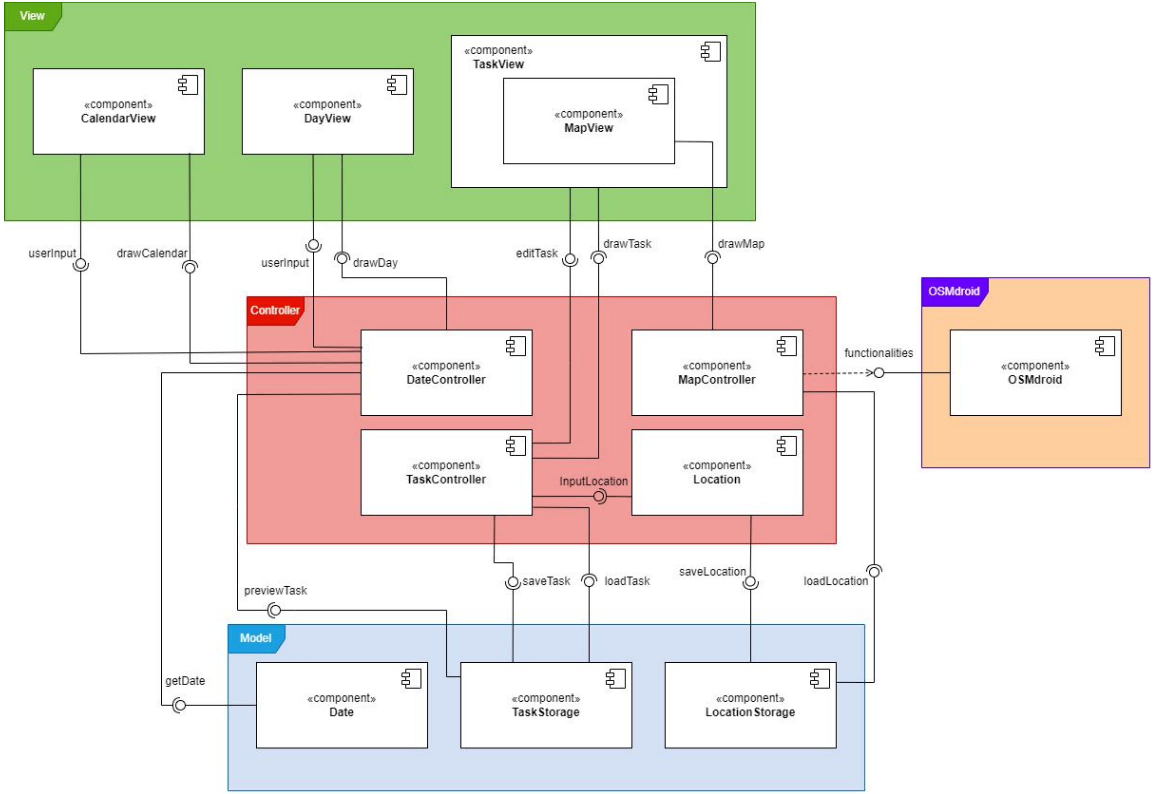
Note: HTW Berlin

ActivityView

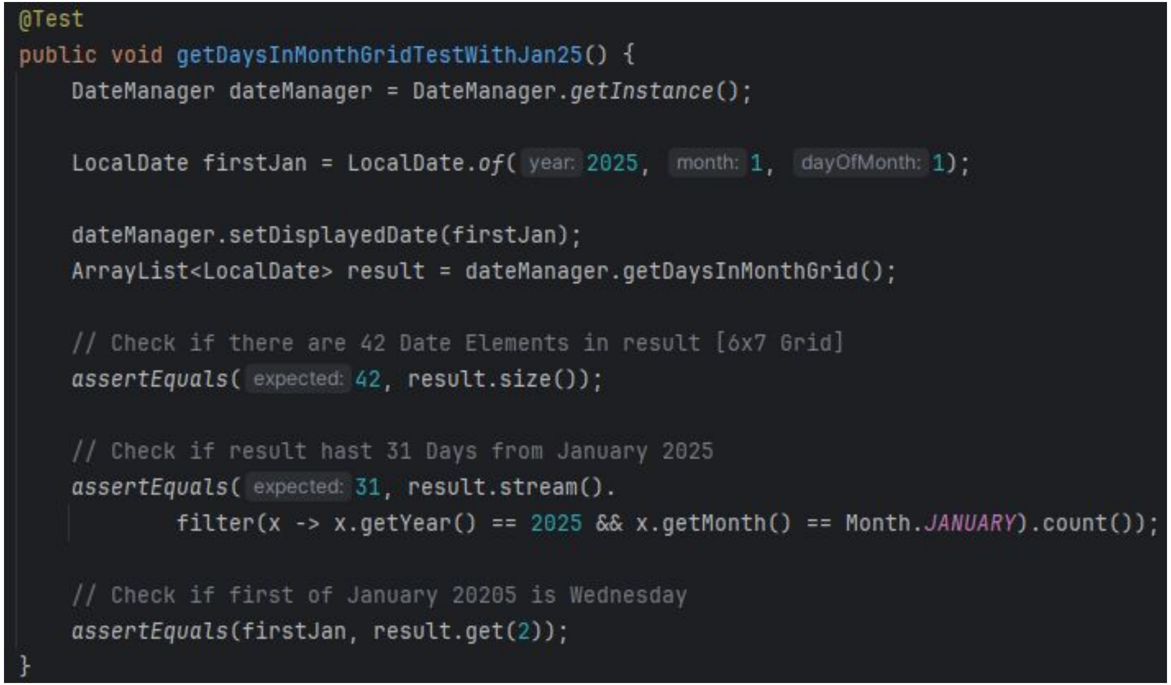
Demonstration

Demonstration der App

Aufbau der Komponenten nach MVC

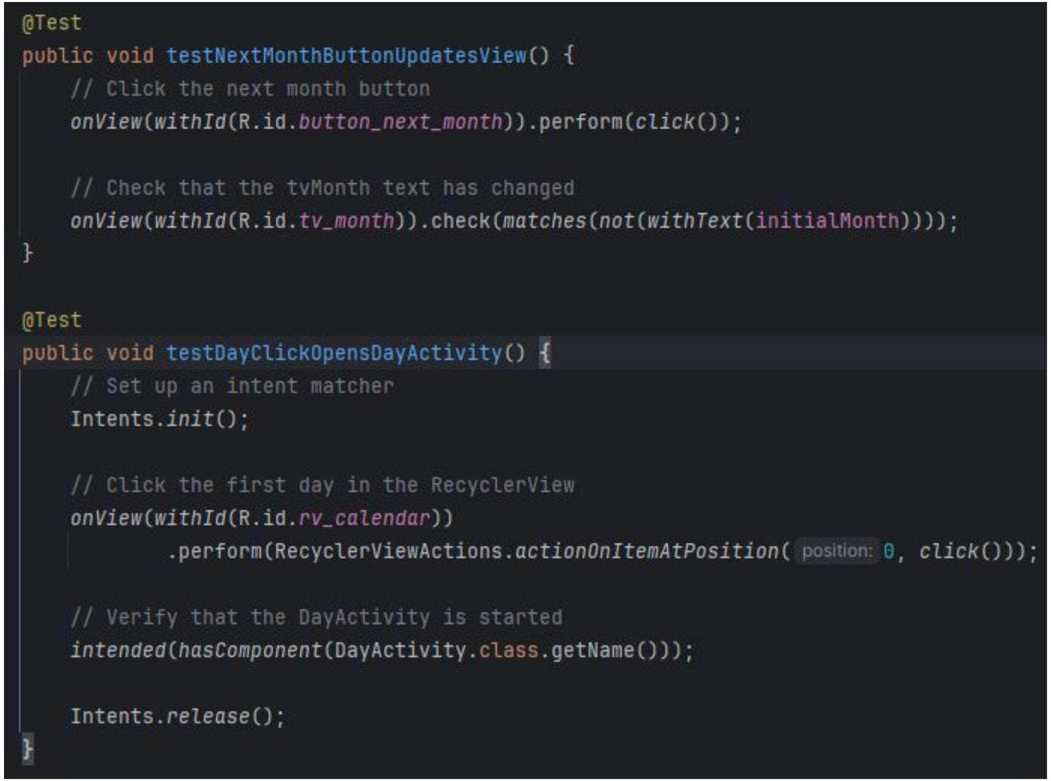


JUnit Test für das Date Model



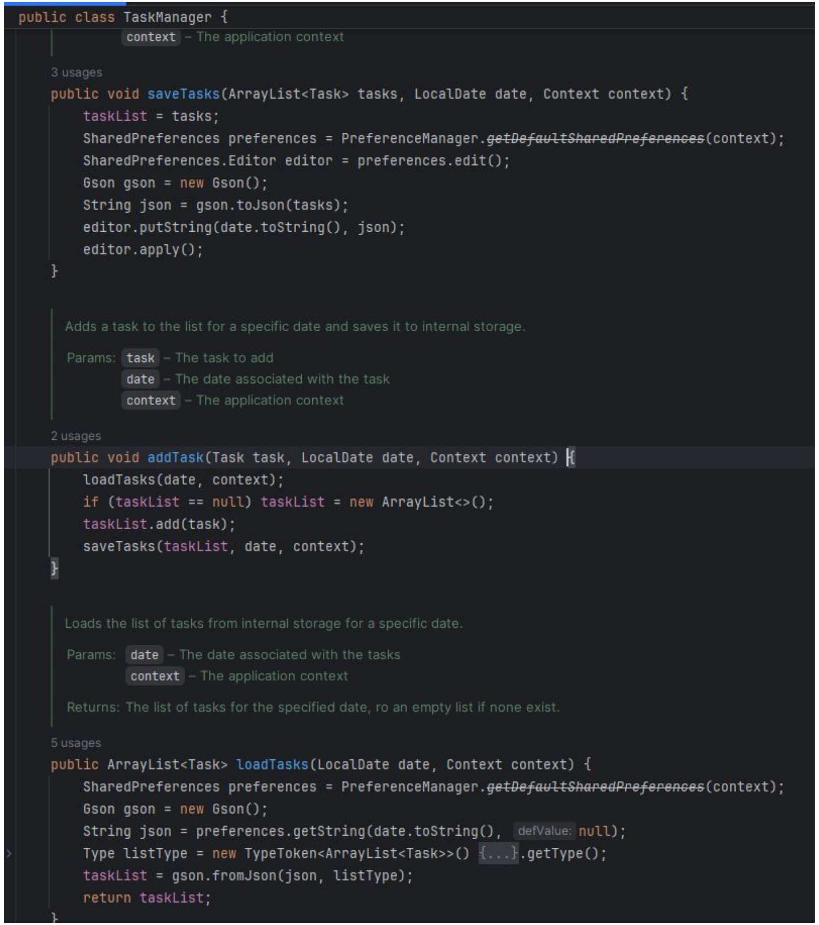
```
@Test
public void getDaysInMonthGridTestWithJan25() {
    DateManager dateManager = DateManager.getInstance();

    LocalDate firstJan = LocalDate.of( year: 2025,  month: 1,  dayOfMonth: 1);

    dateManager.setDisplayedDate(firstJan);
    ArrayList<LocalDate> result = dateManager.getDaysInMonthGrid();

    // Check if there are 42 Date Elements in result [6x7 Grid]
    assertEquals( expected: 42, result.size());

    // Check if result hast 31 Days from January 2025
    assertEquals( expected: 31, result.stream().
            filter(x -> x.getYear() == 2025 && x.getMonth() == Month.JANUARY).count());

    // Check if first of January 20205 is Wednesday
    assertEquals(firstJan, result.get(2));
}
```

Espresso Test für den CalendarView



```
@Test
public void testNextMonthButtonUpdatesView() {
    // Click the next month button
    onView(withId(R.id.button_next_month)).perform(click());

    // Check that the tvMonth text has changed
    onView(withId(R.id.tv_month)).check(matches(not(withText(initialMonth))));
}

@Test
public void testDayClickOpensDayActivity() {
    // Set up an intent matcher
    Intents.init();

    // Click the first day in the RecyclerView
    onView(withId(R.id.rv_calendar))
            .perform(RecyclerViewActions.actionOnItemAtPosition( position: 0, click()));

    // Verify that the DayActivity is started
    intended(hasComponent(DayActivity.class.getName()));

    Intents.release();
}
```

Speichern/Laden



```
public class TaskManager {
            context – The application context

    3 usages
    public void saveTasks(ArrayList<Task> tasks, LocalDate date, Context context) {
        taskList = tasks;
        SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(context);
        SharedPreferences.Editor editor = preferences.edit();
        Gson gson = new Gson();
        String json = gson.toJson(tasks);
        editor.putString(date.toString(), json);
        editor.apply();
    }

     Adds a task to the list for a specific date and saves it to internal storage.

     Params: task – The task to add
             date – The date associated with the task
             context – The application context

    2 usages
    public void addTask(Task task, LocalDate date, Context context) {
        loadTasks(date, context);
        if (taskList == null) taskList = new ArrayList<>();
        taskList.add(task);
        saveTasks(taskList, date, context);
    }

     Loads the list of tasks from internal storage for a specific date.

     Params: date – The date associated with the tasks
             context – The application context

     Returns: The list of tasks for the specified date, ro an empty list if none exist.

    5 usages
    public ArrayList<Task> loadTasks(LocalDate date, Context context) {
        SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(context);
        Gson gson = new Gson();
        String json = preferences.getString(date.toString(), defValue: null);
        Type listType = new TypeToken<ArrayList<Task>>() {...}.getType();
        taskList = gson.fromJson(json, listType);
        return taskList;
    }
```

Fehler/ Gewünschte Features

- Stärkeres Einhalten von MVC bzw. Komponentendiagramms
- Früheres Anfangen
- Anzeigen der Aktivitäten funktioniert nicht richtig
- Bluetooth Funktion war eigentlich geplant