

Introduction

*This is the documentation for the WEA5 project hand-in: Its structure and features will be described. The bus stop project was applied to a sort of event management system for a Catholic Dorm in Linz (KHG, Katholische Hochschulgemeinde): the idea is to actually test run the system in this community, filling a real need (making events happening *right now* visible).*

Handed in by Jack Heseltine, s2110307016



Next Stop KHG

Contents

- Introduction
- Structure of the App
- Navigation Paths
- Test Runs with Screenshots
- Q&A
- Setup
- Libraries and Components
- Not Implemented Features
- +3rd Party Resoureces
- ++Own Learnings

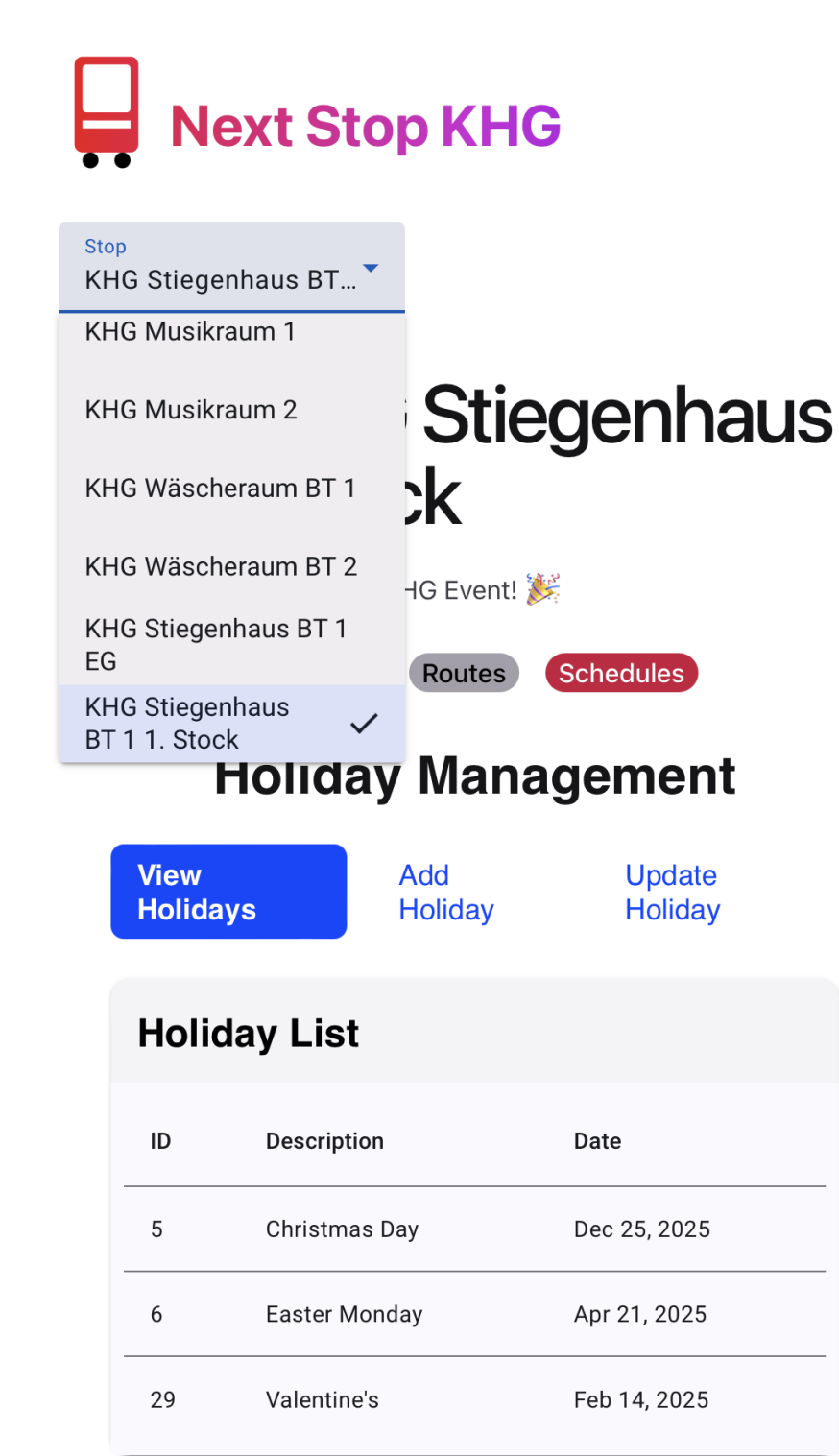
Structure of the App

One main component, one folder for services. 4 modules: holidays (management), stops, routes, schedules. These are only accessible if logged in (admin level), otherwise guarded against. The main component functions as a bus stop sign. Checks every 30 seconds if there are any connections available and flashes red if the next connection is leaving within 5 minutes.

```
*
|- (main) app
|---- holidays
|---- stops
|---- routes
|---- schedules
```

```
| - /services  
| - /guards  
| - /environments  
*
```

The actual stop the bus stop sign is showing for can be selected via drop down, or also query parameter, like <http://localhost:4200/holidays/list?stopId=21>. Corresponds to:



No route stop schedules available.

Navigation Paths

The above structure shows the main navigation paths, starting from the main app. The sub navigation further allows for selecting ...

```
*  
|- holidays|stops|routes|schedules  
|---- list view  
|---- create view  
|---- udate view  
*
```

These are the same for each component.



Next Stop KHG

Stop
KHG Kellerbar

Hello, KHG Stiegenhaus BT 1 1. Stock

Get ready to go to the next KHG Event! 🎉



Holidays

Stops

Routes

Schedules

Schedule Management

View
Schedules

Add
Schedule

Update
Schedule

Update Schedule

Select Schedule

Route ID*

Date*
01/02/2025

Validity Start*
01/02/2025



No route stop
schedules available.

Everything is display within one page = SPA.

Test Runs

1. general user: bus arrives and leaves



Next Stop KHG

Stop

KHG Stiegenhaus BT...

Hello, KHG
Stiegenhaus BT 1 EG

Get ready to go to the next KHG Event! 🎉



Upcoming Schedule ...

Bierlauf BT 1

Stop | time to leave:

KHG Stiegenhaus BT 1 EG
at 15:44

KHG Stiegenhaus BT 1 1.
Stock at 15:45

KHG Stiegenhaus BT 1 2.
Stock at 15:46

KHG Stiegenhaus BT 1 3.
Stock at 15:47

KHG Stiegenhaus BT 1 4.
Stock at 15:48

KHG Stiegenhaus BT 1 5.
Stock at 15:49



Next Stop KHG

Stop

KHG Stiegenhaus BT...

Hello, KHG Stiegenhaus
BT 1 EG



Upcoming Schedule ...

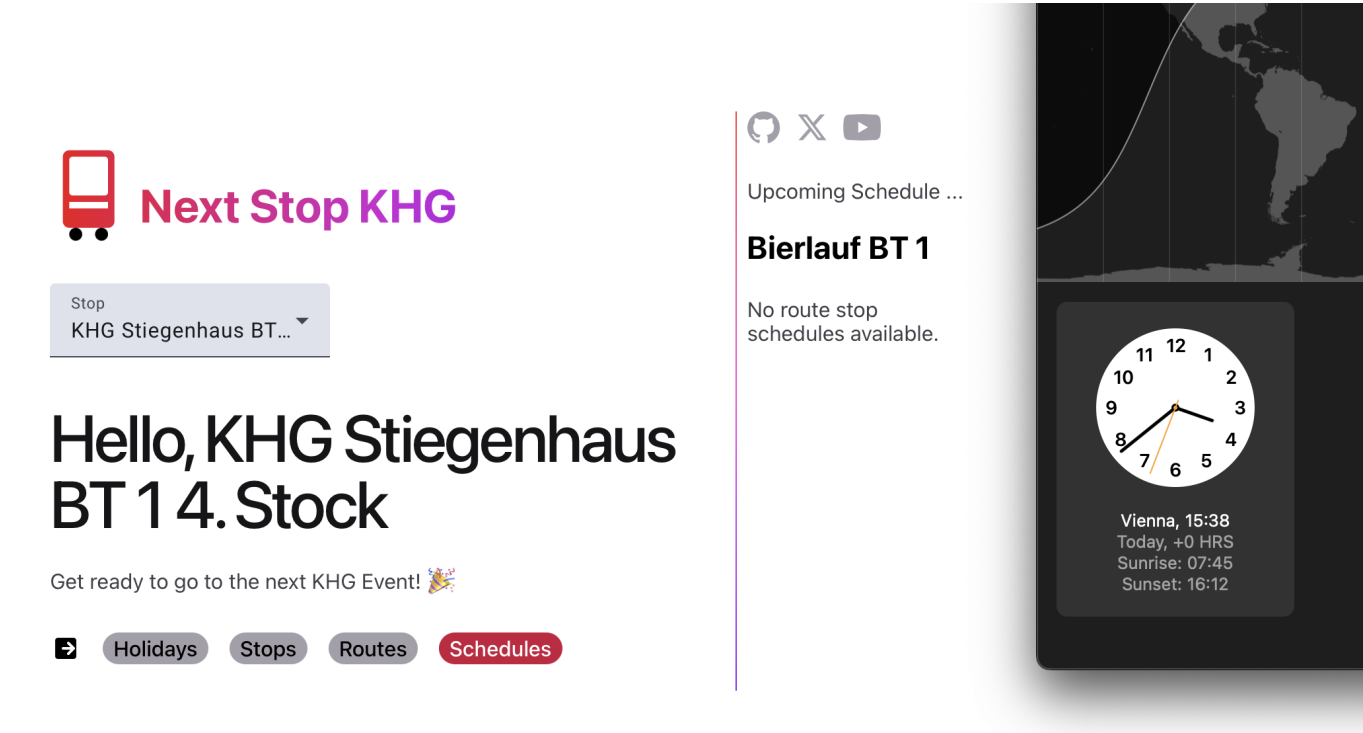
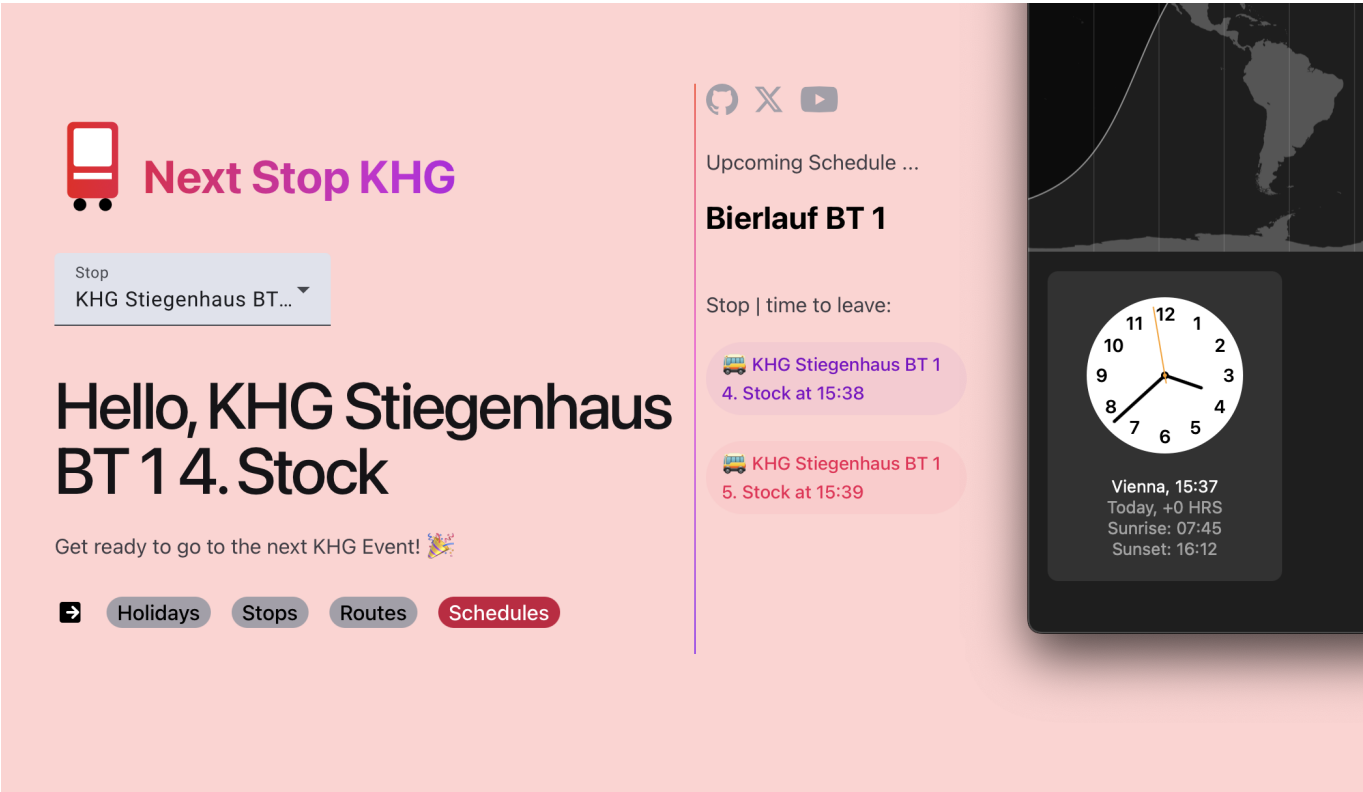
Bierlauf BT 1

No route stop schedules
available.

Get ready to go to the next KHG Event! 🎉



Another Run with Clock (logged in as admin this time)



2. admin user: click through admin menu

Schedule Management

[View Schedules](#)

[Add Schedule](#)

[Update Schedule](#)

Schedules List

Schedule ID	Route ID	Date	Stops
5	1	Jan 1, 2025	<ul style="list-style-type: none">Stop ID: 20 - Sequence: 2 - Time: 23:57Stop ID: 21 - Sequence: 3 - Time: 23:55Stop ID: 22 - Sequence: 4 - Time: 23:59Stop ID: 23 - Sequence: 5 - Time: 23:59Stop ID: 24 - Sequence: 6 - Time: 23:56Stop ID: 25 - Sequence: 7 - Time: 23:58
12	1	Jan 2, 2025	<ul style="list-style-type: none">Stop ID: 20 - Sequence: 2 - Time: 15:35Stop ID: 21 - Sequence: 3 - Time: 15:36Stop ID: 22 - Sequence: 4 - Time: 15:37Stop ID: 23 - Sequence: 5 - Time: 12:30Stop ID: 24 - Sequence: 6 - Time: 15:38Stop ID: 25 - Sequence: 7 - Time: 15:39

Time*

12:30 PM

Stop ID

24

Sequence Number

6

Time*

12:30 PM

Stop ID

25

Sequence Number

7

Time*

12:30 PM

Update Schedule



Stop

KHG Kellerbar

Hello, KHG Stiegenhaus
BT 1 4. Stock

Get ready to go to the next KHG Event! 🎉

- ➔
- Holidays
- Stops
- Routes
- Schedules

Schedule Management

- View Schedules
- Add Schedule
- Update Schedule

Create a New Schedule

Route ID*

1

Date*

01/03/2025

Validity Start*

01/03/2025

Validity Stop*

01/03/2025

Route Stop Schedules

Stop ID

20

Upcoming Schedule ...

Bierlauf BT 1

No route stop
schedules available.



Stop

KHG Kellerbar

Hello, KHG Stiegenhaus
BT 1 4. Stock

Get ready to go to the next KHG Event! 🎉

- ➔
- Holidays
- Stops
- Routes
- Schedules

Route Management

- View Routes
- Add Route
- Update Route

Route List			
ID	Route Number	Validity Period	Day Validity
3	KHG Stiegenhaus BT 2 5. Stock zum Gottesdienst	2024-01-01 to 2025-12-31	Weekends
2	Bierlauf BT 2	2024-01-01 to 2025-12-31	Weekends
4	KHG Stiegenhaus BT 2 5. Stock zur Mensa	2024-01-01 to 2025-12-31	Weekends
20	string1	string1	string
26	Test	Test	Test
1	Bierlauf BT 1	2024-01-01 to 2025-12-31	Weekends



Upcoming Schedule ...

Bierlauf BT 1

No route stop
schedules available.



Upcoming Schedule ...



Stop

KHG Kellerbar

Hello, KHG Stiegenhaus BT 1 4. Stock

Get ready to go to the next KHG Event! 🎉

- ➔
- Holidays
- Stops**
- Routes
- Schedules

Stop Management

- View Stops
- Add Stop
- Update Stop

Stop List

ID	Description
2	KHG Foyer
3	KHG Raum der Stille
4	KHG Kinderrestaurant
5	KHG TV-Raum
6	KHG Zeitungsleseraum
7	KHG KHJ-Zimmer
8	KHG Pastoralbüro
9	KHG Garten

Bierlauf BT 1

No route stop schedules available.



Next Stop KHG

Stop

KHG Kellerbar

Hello, KHG Stiegenhaus
BT 1 4. Stock

Get ready to go to the next KHG Event! 🎉

- ➔
- Holidays
- Stops
- Routes
- Schedules

Holiday Management

View
Holidays

Add
Holiday

Update
Holiday

Holiday List

ID	Description	Date
5	Christmas Day	Dec 25, 2025
6	Easter Monday	Apr 21, 2025
29	Valentine's	Feb 14, 2025



Upcoming Schedule ...

Bierlauf BT 1

No route stop
schedules available.



Next Stop KHG

Stop
KHG Kellerbar

Hello, KHG Stiegenhaus BT 1 4. Stock

Get ready to go to the next KHG Event! 🎉



Holidays

Stops

Routes

Schedules

Holiday Management

View
Holidays

Add
Holiday

Update
Holiday

Create a New Holiday

Description*

Date*

01/02/2025

☐ Is this a school break?

Create Holiday



Upcoming Schedule ..

Bierlauf BT 1

No route stop
schedules available.



Next Stop KHG

Stop
KHG Kellerbar

Hello, KHG Stiegenhaus BT 1 4. Stock

Get ready to go to the next KHG Event! 🎉



Holidays

Stops

Routes

Schedules

Holiday Management

View
Holidays

Add
Holiday

Update
Holiday

Update Holiday

Holiday ID*



Description*

Date*

01/02/2025



Is this a school break?

Update Holiday



Upcoming Schedule ...

Bierlauf BT 1

No route stop
schedules available.

a. URLs

These work via URL router module: I tested with the Holidays module, to introduce the common type "Holidays"

- It is enough to simply change app.routes.ts to ...

```
import { Routes } from '@angular/router';
import { HolidayListComponent } from './holidays/holiday-list/holiday-list.component';

export const routes: Routes = [
  { path: 'holidays', loadChildren: () =>
import('./holidays/holidays.module').then(m => m.HolidaysModule) },
  { path: 'stops', loadChildren: () =>
import('./stops/stops.module').then(m => m.StopsModule) },
  { path: 'routes', loadChildren: () =>
import('./routes/routes.module').then(m => m.RoutesModule) },
  { path: 'schedules', loadChildren: () =>
import('./schedules/schedules.module').then(m => m.SchedulesModule) },
  { path: '**', redirectTo: '' },
];
```

b. Auth-guards to protect components component-wise

E.g. Holidays component (RedirectAuthGuard) - see also URL-interactions with the above url specs.

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HolidaysComponent } from './holidays.component';
import { HolidayListComponent } from './holiday-list/holiday-list.component';
import { HolidayCreateComponent } from './holiday-create/holiday-create.component';
import { HolidayUpdateComponent } from './holiday-update/holiday-update.component';
import { RedirectAuthGuard } from '../guards/redirect-auth.guard';

const routes: Routes = [
  {
    path: '',
    component: HolidaysComponent,
    canActivate: [RedirectAuthGuard], // Ensure authentication
    children: [
      { path: '', component: HolidayListComponent },
      { path: 'list', component: HolidayListComponent },
      { path: 'create', component: HolidayCreateComponent },
      { path: 'update', component: HolidayUpdateComponent },
    ],
  },
];
```

```

    },
  ];

  @NgModule({
    imports: [RouterModule.forChild(routes)],
    exports: [RouterModule],
  })
  export class HolidaysRoutingModule {}

```

I.e. sub-urls are determined here but protected, redirect to base app component.

c. HTML Form level validation

If the server cannot process the data, at this second-tier a message will be issued and the user informed with a pop-up. The form validation looks like this in the case of Holidays:

```

<mat-card>
  <h2>Create a New Holiday</h2>
  <form [formGroup]="holidayForm" (ngSubmit)="onSubmit()">
    <mat-form-field class="holiday-form-field" appearance="fill">
      <mat-label>Description</mat-label>
      <input matInput formControlName="description" placeholder="Enter
holiday description" />
      <mat-error *ngIf="submitted &&
holidayForm.controls['description'].invalid">
        A description is required (min 3 characters).
      </mat-error>
    </mat-form-field>

    <mat-form-field class="holiday-form-field" appearance="fill">
      <mat-label>Date</mat-label>
      <input matInput formControlName="date" placeholder="Enter holiday
date" type="date" />
      <mat-error *ngIf="submitted &&
holidayForm.controls['date'].invalid">
        A valid date is required.
      </mat-error>
    </mat-form-field>

    <mat-checkbox formControlName="isSchoolBreak">
      Is this a school break?
    </mat-checkbox>

    <button mat-raised-button class="holiday-button" color="primary"
type="submit">
      Create Holiday
    </button>
  </form>
</mat-card>

```

d. reactive server message handling in the front end

See also immediately above. If the schedule cannot be pulled for example:

```
fetchNextConnections(): void {
  this.scheduleService.getNextConnections(this.stopId).subscribe({
    next: (data) => {
      this.scheduleId = data[0]?.scheduleId;
      this.routeStopSchedules = data[0]?.routeStopSchedules || [];
      this.fetchRouteDetails(data[0]?.routeId);

      // pull out time for flashing logic
      const now = new Date();

      const providedTime = new Date(now);

      providedTime.setMinutes(data[0]?.routeStopSchedules[0].time.minute);
      providedTime.setHours(data[0]?.routeStopSchedules[0].time.hour);

      const timeDifference = providedTime.getTime() - now.getTime();

      if (timeDifference > 0 && timeDifference <= 5 * 60 * 1000) {
        this.isFlashing = true;
      } else {
        this.isFlashing = false;
      }
    },
    error: (err) => {
      console.info('Error fetching connections:', err);
      if (this.isFlashing) {
        this.isFlashing = false;
      }
      this.routeStopSchedules = [];
    },
  });
}
```

So this information is passed visually to the user. In later stages of development I largely removed console error logs since some outage is possible. This is logged at info level and the user informed in tandem, be it visually.

Setup

See readme: requires running backend and db. I might still look at online deployment of asp.net core/wea bundle.

Locally: `ng serve` in the repo.

Structure

at first two-tier authorisation was considered (special admins group), but then in accordance with bus stop info signs: the main view should be public, only one extra tier (~admins) for management and check-in.

Libraries and Components

- Angular Material Design (liked this very much, first time)
- Auth0 AuthService made Auth0 integration very straightforward
- (Native) RouterModule (super helpful, see structure above)

Not Implemented Features

These simplifications were made, Hard:

- No Checkin.
- No diagrams and delays (also not in backend - cf. SWK).

Soft:

- Full object display: update forms refer to ids but it would be better to get relevant object.

+Resources

I want to make a note of resources helpful for work on this app.

- <https://auth0.com/docs/libraries/auth0-single-page-app-sdk>
- <https://community.auth0.com/t/angular-and-net-6-integrated/124647> and https://auth0.com/docs/quickstart/spa/angular/02-calling-an-api?_gl=11ps7fdf_gcl_auMTQwODIzMDA1MC4xNzM0ODY5MTMx_gaMTY2MTE2MDI1NS4xNzM0ODY5MTMy_ga_QKMSDV5369*MTczNTQ3OTcyMi40LjEuMTczNTQ4MzYyNC42MC4wLjA. (for authorization oekn/audience setup questions)/also reference from auth0 perspective as far as flows in general <https://auth0.com/docs/get-started/authentication-and-authorization-flow/authorization-code-flow-with-pkce>

++Learnings

Not always using DTOs on client side, instead relying on form functionality to shape data here and let backend take over the validation.