

Besprechungsprotokoll 12.02.2025

Milad Moradi

Protokoll des Gesprächs mit Herrn Professor Bauer und Herrn Professor Stütz

Table 1. Teilnehmer

Anwesend	Verteiler
Thomas Stütz	
Petter Bauer	
Amir Mohamadi	
Milad Moradi	
Obaidullah Mustaq	
Kyaw Thet Htun	

Table 2. Ort und Zeit

Ort	HTL-Leonding
Von-Bis	Mittwoch . 12.02.2025,
Dauer	100 min

Besprochene Punkte

1. Musik bei Mitmachgeschichten

Musik soll parallel zur Mitmachgeschichte abgespielt werden.

Abstimmung der Musik mit den Geschichten.

Technische Umsetzung und Steuerung.

2. Gästeverwaltung

Liste aller Gäste, die ausgelesen werden kann.

Steuerungsmöglichkeiten für die Gäste.

3. Bewegungssteuerung

Eigene Bewegungen für die Roboter definieren.

Eine Person aus der Gruppe kümmert sich darum.

4. Inhalte auf beiden Robotern

Alle Inhalte sollen auf beiden Robotern laufen.


Technische Voraussetzungen zur Umsetzung.

Anhänge

Mitmachgeschichten-Auswahl

Mitmachgeschichten

Nach Geschichte filtern:

	Zirkusbesuch		
	Wanderung		
	Bergwanderung		
	Besuch Alpaka		
	Wildalpen		
	Sommertag		
	Handbewegungen		
	Test		

Neue Geschichte

Mitmachgeschichten-Editieren

Titel der Geschichte

Zirkusbesuch



Abbrechen

Hinzufügen

Sprachausgabe	Bewegung	Dauer(sec)	Bild hochladen
<p>↓</p> <p>Während die Zirkusvorstellung beginnt, winken wir begeistert den talentierten Artisten zu. Die Arme bewegen sich fröhlich im Takt der Musik, und wir freuen uns Teil des</p>	Winken	15	
<p>↑</p> <p>↓</p> <p>Mit einem lauten "Hurra" applaudieren wir am Ende der Zirkusvorstellung. Die freudige Bewegung der Hände begleitet ihren Applaus, und wir jubeln den Künstlern zu.</p>	Hurra	5	
<p>↑</p> <p>Nach der aufregenden Vorstellung nehmen wir Platz und essen gemeinsam Popcorn. Abschließend fahren wir dann wieder nach Hause.</p>	Essen	5	

Initialisierung der Datenbank

```
using Microsoft.EntityFrameworkCore;
var MyAllowSpecificOrigins = "_myAllowSpecificOrigins";
var connectionString="Server=databasepepper;Database=db_pepper;User=sa;Password=Password1234!;TrustServerCertificate=True;";
var builder = WebApplication.CreateBuilder(args);

builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen(options =>
{
    options.SwaggerDoc("v1", new Microsoft.OpenApi.Models.OpenApiInfo { Title = "Pepper in Elderly Care API Documentation ", Version = "v1" });
});

builder.Services.AddCors(options =>
{
    options.AddPolicy(name: MyAllowSpecificOrigins,
        policy =>
        {
            policy.AllowAnyOrigin()
                .AllowAnyHeader()
                .AllowAnyMethod();
        });
});

builder.Services.AddControllers()
    .AddJsonOptions(options =>
    {
        options.JsonSerializerOptions.PropertyNameCaseInsensitive = true;
    });

builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(connectionString));
var app = builder.Build();
app.UseSwagger();
app.UseSwaggerUI();
app.UseCors(MyAllowSpecificOrigins);
```

Add-Get-TagAlongStories

```
//Add step to TagAlongStory
app.MapPost("/tagalongstories/{id}/steps", async (ApplicationDbContext context, int id, StepDTO stepToAdd) => {
    var tagAlongStory = await context.TagAlongStories.Include(tas => tas.Steps).Where(tas => tas.Id == id).FirstOrDefaultAsync();
    if (tagAlongStory == null)
    {
        return Results.NotFound("No tag along story with id " + id + " found");
    }
    if (stepToAdd == null)
    {
        return Results.BadRequest("Provided step is null");
    }
    Step step = await Converter.ConvertToStepAsync(stepToAdd, context, tagAlongStory.Steps.Count);
    tagAlongStory.Steps.Add(step);
    context.Update(tagAlongStory);
    await context.SaveChangesAsync();
    return Results.Ok(tagAlongStory);
})
.WithSummary("Add a step to the tag along story")
.WithOpenApi();

//Get step from TagAlongStory
app.MapGet("/tagalongstories/{id}/steps", async (ApplicationDbContext context, int id) => {
    var tagAlongStory = await context.TagAlongStories.Include(tas => tas.Steps).Where(tas => tas.Id == id).FirstOrDefaultAsync();
    if (tagAlongStory == null)
    {
        return Results.NotFound("No tag along story with id " + id + " found");
    }
    return Results.Ok(tagAlongStory.Steps);
})
.WithSummary("Get all steps of a tag along story")
.WithOpenApi();

#endregion

app.Run();
```

Get-All-TagAlongStories

```
#region "Tag Along Story Endpoints"
// Get all TagAlongStories
app.MapGet("tagalongstories", async (ApplicationDbContext context, bool? withoutDisabled) => {
    IQueryable<TagAlongStory> query = context.TagAlongStories.Include(tas => tas.Steps);
    if (withoutDisabled.HasValue && withoutDisabled.Value)
    {
        query = query.Where(tas => tas.IsEnabled);
    }
    var tagAlongStories = await query.ToListAsync();

    var tagAlongStoriesDTO = new List<TagAlongStoryDTO>();

    foreach (var tagalongstory in tagAlongStories)
    {
        var storyIconLink = $"http://152.67.78.190:8080/tagalongstories/{tagalongstory.Id}/picture";

        tagAlongStoriesDTO.Add(new TagAlongStoryDTO
        {
            Id = tagalongstory.Id,
            Name = tagalongstory.Name,
            StoryIcon = storyIconLink,
            Steps = [],
            IsEnabled = tagalongstory.IsEnabled
        });
    }

    if (tagAlongStoriesDTO.Count != 0)
    {
        return Results.Ok(tagAlongStoriesDTO);
    }
    return Results.NotFound("No tag along stories found");
})
.WithSummary("Get all TagAlongStories")
.WithOpenApi();
#endregion
```

Delete-Update-TagAlongStories

```

// Update an existing TagAlongStory
app.MapPut("/tagalongstories/{id}", async (ApplicationDbContext context,int id,TagAlongStoryDTO tagAlongStoryDTO) =>
{
    var tagAlongStory = await context.TagAlongStories.Include(tas=>tas.Steps).Where(tas => tas.Id == id).FirstOrDefaultAsync();

    if (tagAlongStory == null)
    {
        return Results.NotFound("No Tag Along Story found");
    }
    if(tagAlongStoryDTO==null)
    {
        return Results.BadRequest("Provided tag along story is null");
    }
    if(tagAlongStoryDTO.Id!=id)
    {
        return Results.BadRequest("Dto id doesnt match requested update id");
    }
    tagAlongStory.Name=tagAlongStoryDTO.Name;
    tagAlongStory.StoryIcon=Crypt.decryptByteArray(tagAlongStoryDTO.StoryIcon);
    tagAlongStory.IsEnabled=tagAlongStoryDTO.IsEnabled;
    tagAlongStory.Steps=await Converter.ConvertToStepList(tagAlongStoryDTO.Steps,context);
    context.TagAlongStories.Update(tagAlongStory);
    await context.SaveChangesAsync();

    return Results.Ok("Updated tag along story");
})
.WithSummary("Update an existing TagAlongStory")
.WithOpenApi();

// Delete a TagAlongStory
app.MapDelete("/tagalongstories/{id}", async (ApplicationDbContext context,int id) =>
{
    var tagAlongStory = await context.TagAlongStories.Include(tas=>tas.Steps).Where(tas => tas.Id == id).FirstOrDefaultAsync();

    if (tagAlongStory == null)
    {
        return Results.NotFound("No tag along story with id "+ id+" found!");
    }

    context.TagAlongStories.Remove(tagAlongStory);
    await context.SaveChangesAsync();
    return Results.Ok(tagAlongStory);
})
.WithSummary("Delete a TagAlongStory")
.WithOpenApi();

```