

# Sights - Tourist Attractions

## Gruppnamn och gruppmedlemmar

Annika Sörenstam - Jenny Dyews, Louise Littecke, Fredric Lundgren, Teodor Fredriksson, Alexandra Salén Sammueang, Maria Kilsved och Frans Nilsson Lidström

## Repository

<https://github.com/MariaKilsved/sights.git>

## Hur man bygger och startar

Bygg backend med VisualStudio. Kör frontend med Live Server i VisualStudio Code.  
Backend kan testas separat från frontend genom Swagger.

För att få god översikt över databasen rekommenderar vi DB Browser for SQLite:

<https://sqlitebrowser.org/>

## Installationer

Lägg till följande user secrets:

```
{  
  "JsonWebTokenKeys": {  
    "ValidateIssuerSigningKey": true,  
    "IssuerSigningKey": "64A63153-11C1-4919-9133-EFAF99A9B456",  
    "ValidateIssuer": true,  
    "ValidIssuer": "https://localhost:7249",  
    "ValidateAudience": true,  
    "ValidAudience": "https://localhost:7249",  
    "RequireExpirationTime": true,  
    "ValidateLifetime": true  
  }  
}
```

## Övergripande mjukvaruarkitektur

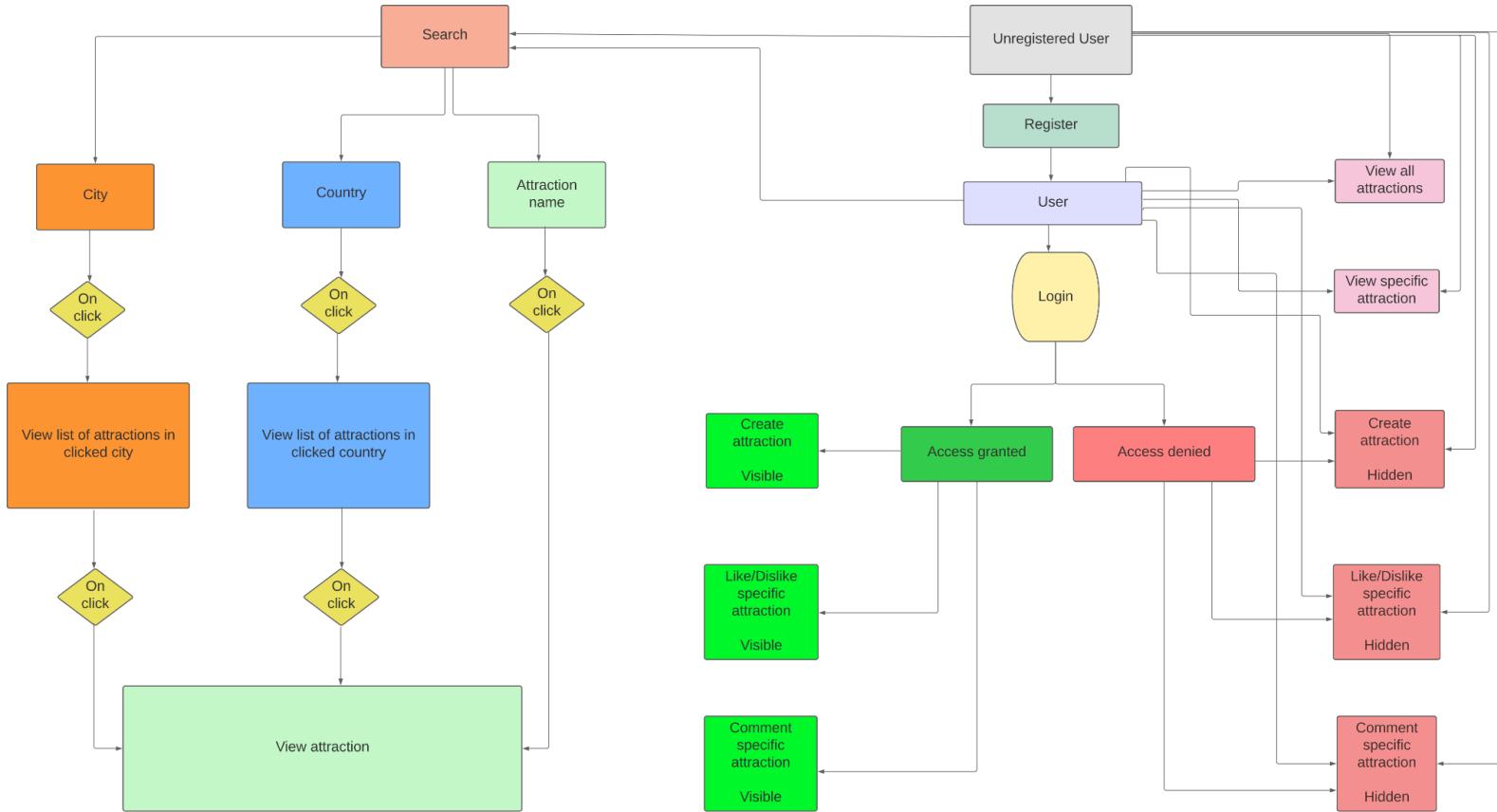
### Filstruktur

Vi har lagt upp databasen, API:et och webbsidan i samma GitHub repo. API:et är uppdelat i olika controllers och models. Webbsidan är uppdelad i återanvändbara komponenter.

# Web A:ets struktur

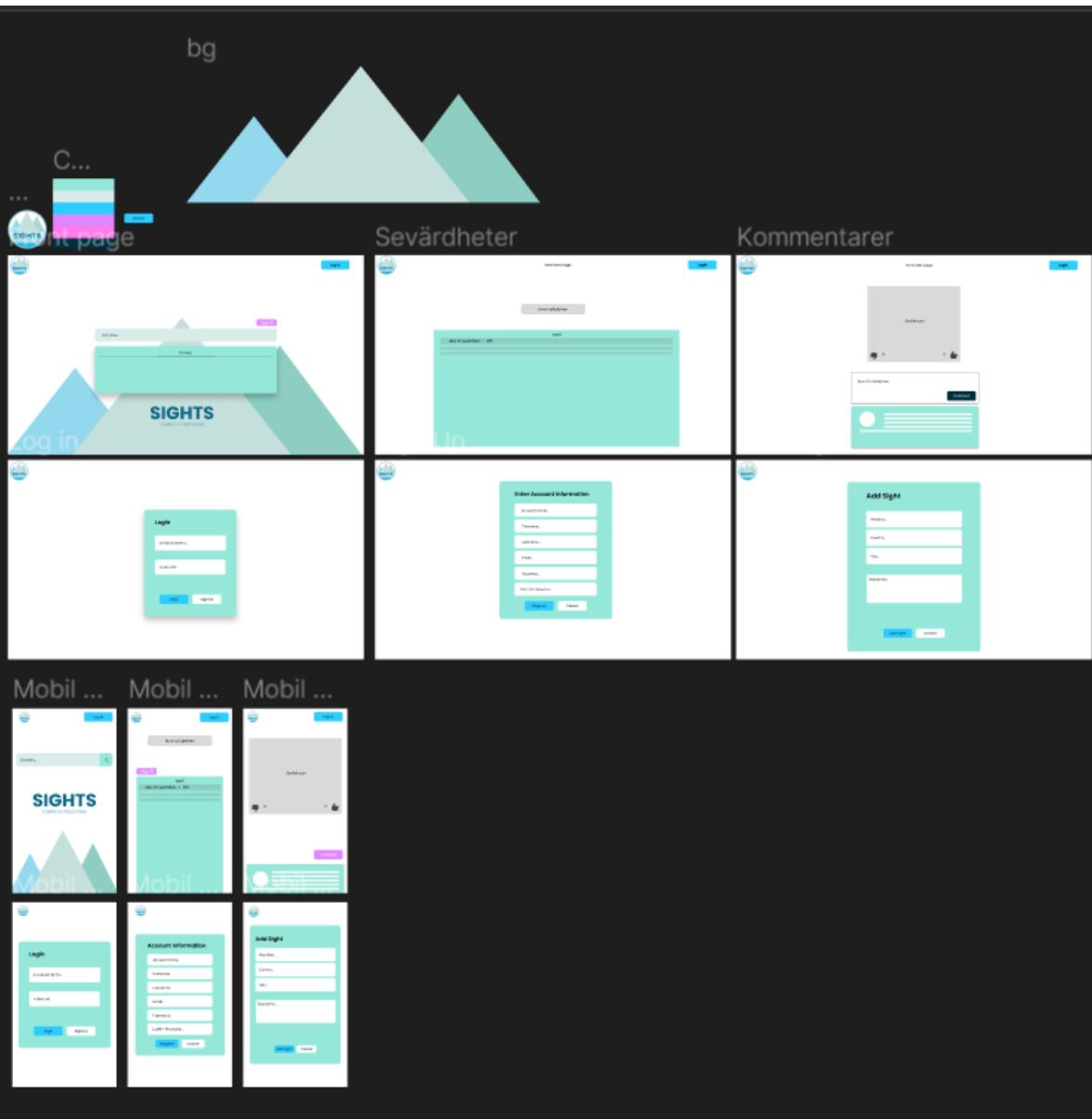
Controller Attraction								
Purpose	Http verb	Url	Url Param	Query Param	Comment	Request Type	Comment	Response Response Type
Read all attractions	GET	api/Attraction		attractionId, title	Read all attractions			200 IEnumerable<attraction> 404 ActionResult
Read specific attraction	GET	api/Attraction/{id}	attraction id	attractionId	Shows the clicked attraction			200 Sight 400 ActionResult
Create attraction	POST	api/Attraction		Title, Description, userId, City Name, Country Name	Add attraction to DB	Attraction	The new attraction	201 ActionResult 400 ActionResult
Get attractions sorted by likes	GET	api/Attraction/ByLikes			Sorted by likes			200 IEnumerable<attraction> 404 ActionResult
Controller Comment								
Purpose	Http verb	Url	Url Param	Query Param	Comment	Request Type	Comment	Response Response Type
Get comments by attraction	GET	api/Comment/ByAttraction		attractionId	Get comments on specific attraction			201 List<CommentUser> Comment + Username
Create comment	POST	api/Comment		userId, content, attractionId	Comment specific attraction	Comment	The new comment	201 ActionResult 400 ActionResult
Controller Like								
Purpose	Http verb	Url	Url Param	Query Param	Comment	Request Type	Comment	Response Response Type
Get likes	GET	api/Like			Get all likes			200 IEnumerable<Like>
Get like by id	GET	api/Like/{id}	like id	like id	Get a specific like			200 IEnumerable<Like>
Like/dislike attraction	POST	api/Like	attractionId, user id	attractionId, userId	1 = like, 0 = dislike	Like	The new like	201 ActionResult 400 ActionResult
Remove like/dislike for attraction	DELETE	api/Like/{id}	attraction id, user id	attractionId, userId	User un-likes/un-dislikes attraction			204 ActionResult 400 ActionResult
Controller User								
Purpose	Http verb	Url	Url Param	Query Param	Comment	Request Type	Comment	Response Response Type
Get user by id	GET	api/User/{id}	User id	User id	Get the user with a certain id			200 User
Create user	POST	api/User		username, password	Register a user	User	The new user	201 ActionResult
Login	GET	api/User/Login		username, password	Log in a user and return token		Token with username included	200 JwtUserToken
OldLogin	GET	api/User/OldLogin		username, password	Log in a user and return the user		The logged in user	200 User 404 ActionResult
Controller City								
Purpose	Http verb	Url	Url Param	Query Param	Comment	Request Type	Comment	Response Response Type
Get all cities	GET	api/City						200 IEnumerable<City> 404 ActionResult
Controller Country								
Purpose	Http verb	Url	Url Param	Query Param	Comment	Request Type	Comment	Response Response Type
Get all countries	GET	api/Country						200 IEnumerable<Country> 404 ActionResult
Controller SubComment								
Purpose	Http verb	Url	Url Param	Query Param	Comment	Request Type	Comment	Response Response Type
Get SubComment	GET	api/SubComment/{id}	subcomment id	subcomment id	Necessary for post to api/SubComment/{id}			200 SubComment
Create SubComment	POST	api/SubComment	comment id	commentId, userId	Example of JWT in use	SubComment	The new subcomment	201 ActionResult 400 ActionResult 401 ActionResult 404 ActionResult

## Sights flödesschema



## Figma för frontend

Figma användes för att på ett illustrativt sätt ta fram design samt struktur på hemsidan



## Databasen arkitektur

Databasen är gjord i Sqlite. Detta betyder att det finns ett begränsat antal datatyper att välja på.

