

Sights - Tourist Attractions

Gruppenamn 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

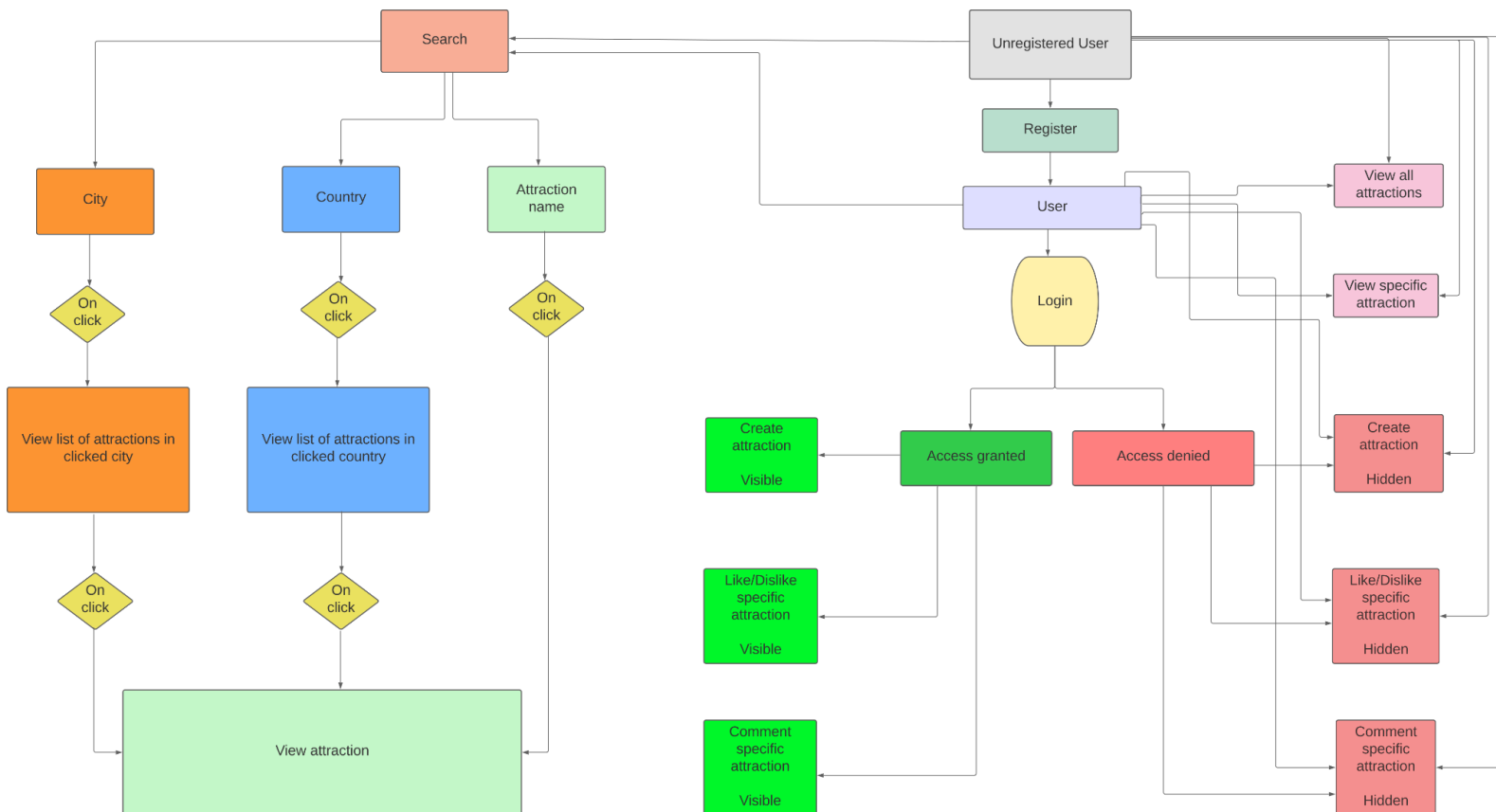
Filstruktur

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 API:ets struktur

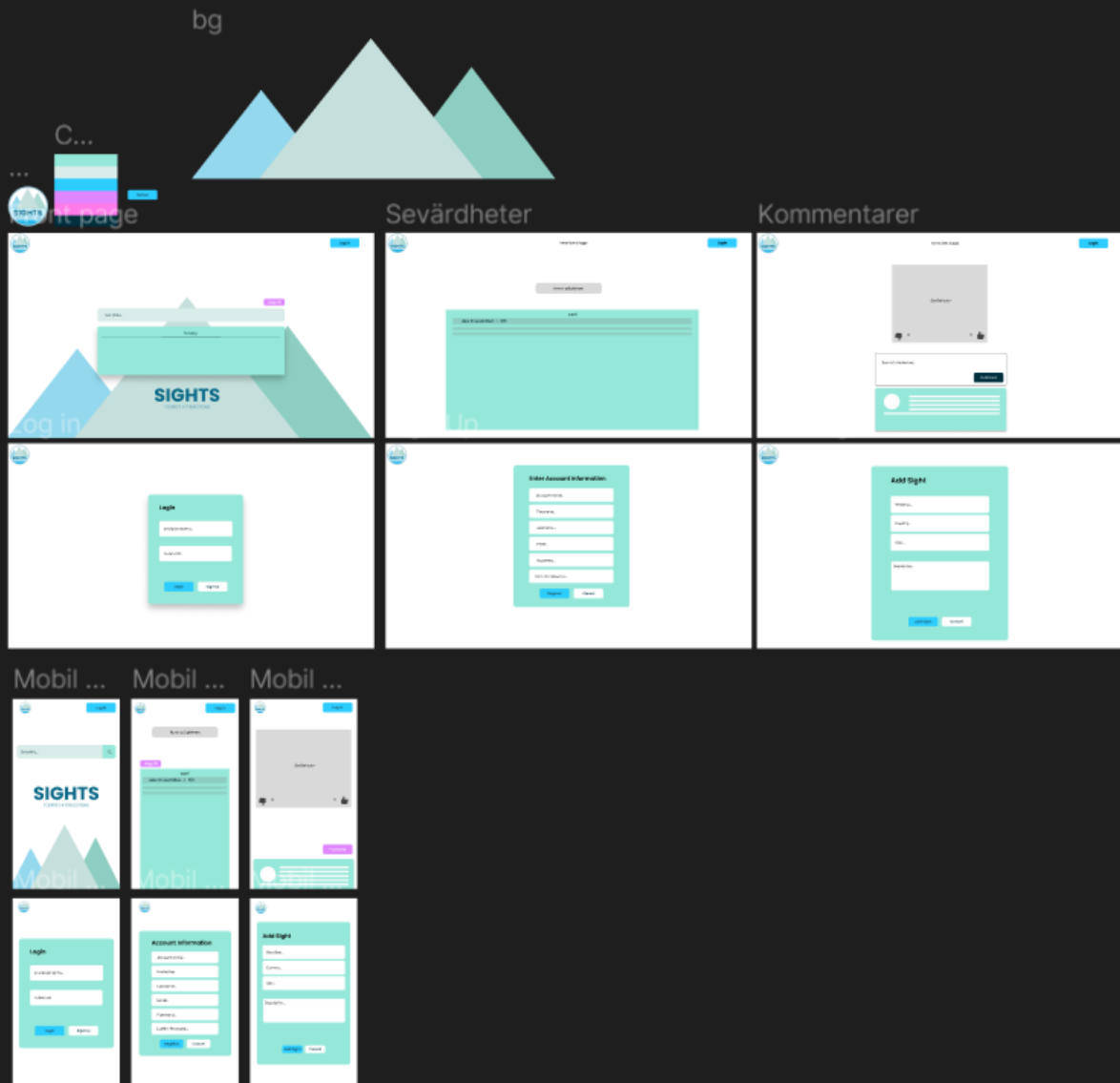
Controller Attraction										
Purpose	Http verb	Url	Uri Param	Query Param	Comment	Request Type	Comment	Response	Response Type	Comment
Read all attractions	GET	api/Attraction		attractionId, title	Read all attractions			200	IEnumerable<attraction>	See all sights
								404	ActionResult	NotFound
Read specific attraction	GET	api/Attraction/{id}	attraction id	attractionId	Shows the clicked attraction			200	Sight	Show specific sight with comments
								400	ActionResult	Bad request: Attraction id format error
Create attraction	POST	api/Attraction		Title, Description, userId, City Name, Country Name	Add attraction to DB	Attraction	The new attraction	201	ActionResult	New attraction added. Allow/deny access based on login?
								400	ActionResult	Bad request: No attraction data ID already existing
								200	IEnumerable<attraction>	Could not create
Get attractions sorted by likes	GET	api/Attraction/ByLikes			Sorted by likes			404	ActionResult	NotFound
Controller Comment										
Purpose	Http verb	Url	Uri Param	Query Param	Comment	Request Type	Comment	Response	Response Type	Comment
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	New comment added. Allow/deny access based on login?
								400	ActionResult	Bad request: UserId format error content format error attraction id format error
Controller Like										
Purpose	Http verb	Url	Uri Param	Query Param	Comment	Request Type	Comment	Response	Response Type	Comment
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	attraction id, user id	attractionId, userId	1 = like, 0 = dislike	Like	The new like	201	ActionResult	NoContentResult
								400	ActionResult	NoContentResult
								204	ActionResult	NoContentResult
Remove like/dislike for attraction	DELETE	api/Like/{id}	attraction id, user id	attractionId, userId	User un-likes/un-dislikes attraction			400	ActionResult	Bad request: attraction id format error user id format error
Controller User										
Purpose	Http verb	Url	Uri Param	Query Param	Comment	Request Type	Comment	Response	Response Type	Comment
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	Register user
Login	GET	api/User/Login		username, password	Log in a user and return token		Token with username included	200	JwtUserToken	Success
								400	ActionResult	Bad request: user is null username is null password is null
OldLogin	GET	api/User/OldLogin		username, password	Log in a user and return the user		The logged in user	200	User	Success
								404	ActionResult	NotFound
Controller City										
Purpose	Http verb	Url	Uri Param	Query Param	Comment	Request Type	Comment	Response	Response Type	Comment
Get all cities	GET	api/City						200	IEnumerable<City>	
								404	ActionResult	NotFound: Context was null
Controller Country										
Purpose	Http verb	Url	Uri Param	Query Param	Comment	Request Type	Comment	Response	Response Type	Comment
Get all countries	GET	api/Country						200	IEnumerable<Country>	
								404	ActionResult	NotFound: Context was null
Controller SubComment										
Purpose	Http verb	Url	Uri Param	Query Param	Comment	Request Type	Comment	Response	Response Type	Comment
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	NoContentResult
								400	ActionResult	Bad request: comment id format error user id format error content format error
								401	ActionResult	Unauthorized
								404	ActionResult	

Sights flödesschema



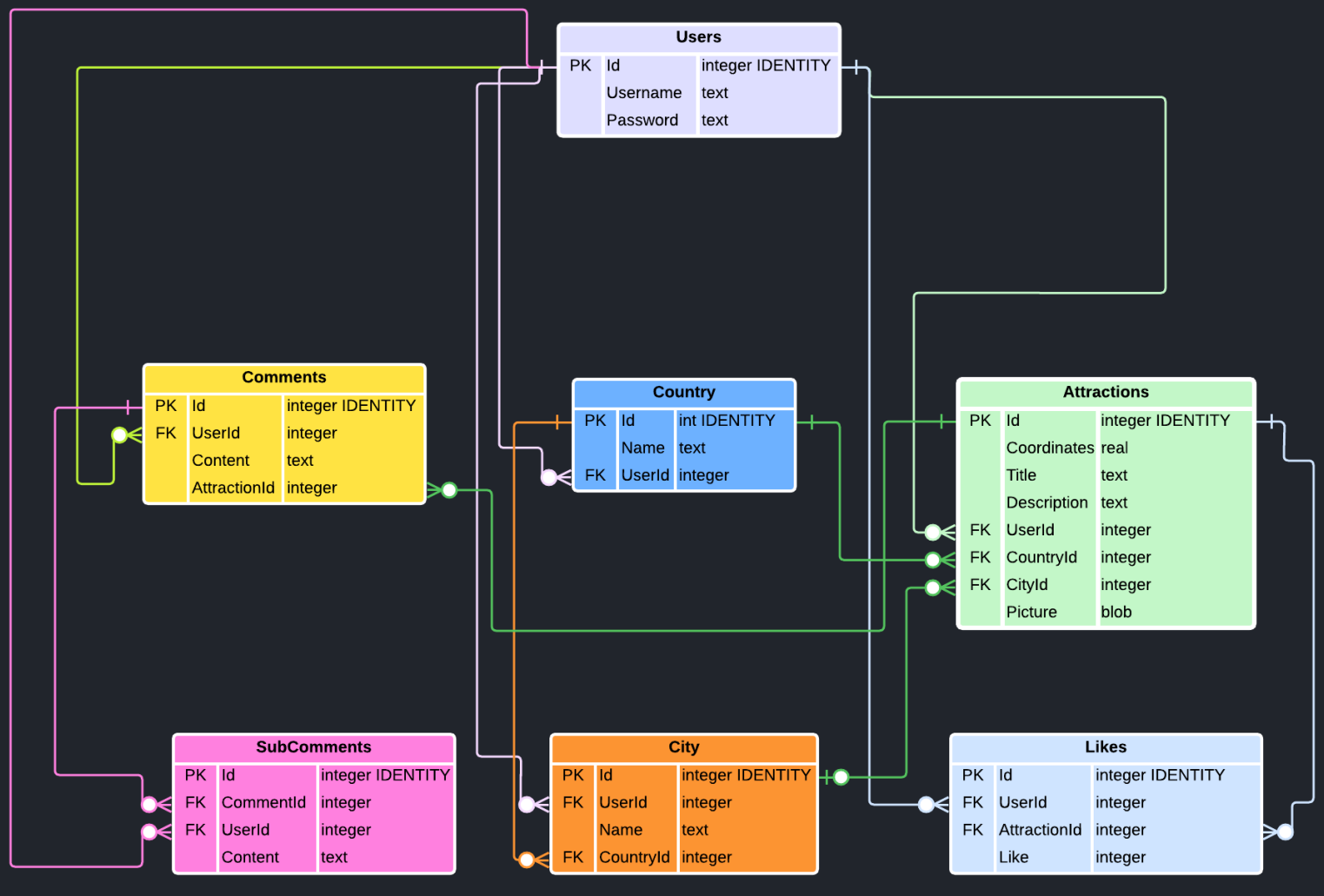
Figma för frontend

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.



Implementation av JWT

Vi har provat att implementera JWT på API:et men inte på frontend-sidan. Därför har vi två login-metoder: Login med JWT och OldLogin utan JWT. Båda dessa ligger i UserController.cs. Därefter har vi lagt in JWT till PostSubComment i SubComment.cs. Denna endpoint har inte använts någonstans men vi kan ändå visa att vi lyckats applicera JWT. Om vi hade haft mer tid hade vi gärna implementerat JWT på flera ställen.