

*** Database ***

Peter Schneider-Kamp

petersk@imada.sdu.dk

<http://imada.sdu.dk/~petersk/>

EXAM PROJECT



- Delete Instagram, Twitter, Facebook & Co!
- **Hoply** – your very own social media platform:
 - *Follow your fellows!*
 - *Miff them with messages!*
 - *Collaborate and chill!*

The (Not So) Small Print

- Three types of features:
 - a) Basic features (user creation, following, messaging)
 - b) Advanced features (location sharing, group messages, picture sharing)
 - c) Fun features (your fantasy is the limit 😊)
- Groups of sizes 1–3
- Minimal expectations (and grades) depend on group size
 - 1) 3x basic
 - 2) 3x basic & 1x advanced/fun
 - 3) 3x basic & 2x advanced/fun

Possible Platforms

Mobile app development frameworks:

- **Android Studio** (Java, Kotlin, Scala)
- Android NDK (C/C++)
- iOS SDK (Objective-C/Swift)

Cross-platform frameworks:

- Ionic, Xamarin, PhoneGap, Flutter, React Native, ...

Web-based frameworks:

- Ruby on Rails, Django, Laravel, JavaBean, ...
- **Support ONLY for Android with Java!**

REMOTE DATABASE

Remote Database Design

- Three schemata for basic features:
 - users(id text, name text, stamp timestamp)
 - follows(follower text, followee text, stamp timestamp)
 - messages(id serial, sender text, receiver text, body text, stamp timestamp)
- Foreign key constraints for
follower, followee, sender, receiver
- Will be extended for advanced features.
- For "Fun features", you as a group need to request (reasonable) extensions!

Remote Database Access

- RESTful web service using PostGREST
- Base URI is also Swagger documentation:
 - <http://caracal.imada.sdu.dk/app2019/>
- Support all CRUD operations:
 - Selection using GET
 - Insertion using POST
 - Deleting using DELETE

Remote Database Selection 1

- To select all rows from the users table, perform a GET to the following URL:
- <http://caracal.imada.sdu.dk/app2019/users>
- To select only the row belonging to the user with id "killme2", perform a GET to the following URL:
- <http://caracal.imada.sdu.dk/app2019/users?id=eq.killme2>

Remote Database Selection 2

- To select only users created after a certain time stamp:
- <http://caracal.imada.sdu.dk/app2019/users?stamp=gt.2019-04-01T11:31:47%2B02:00>
- Note how the “+” sign is encoded as “%2B”!

Remote Database Insertion

- To insert a message, perform a POST to the following URI:
- <http://caracal.imada.sdu.dk/app2019/messages>
- Header:
Content-Type: application/json
- Payload:

```
{"sender":"killme2","receiver":"killme1","body":"you suck, too"}
```

Remote Database Deletion

- To delete all users, perform a DELETE to the following URI:
- <http://caracal.imada.sdu.dk/app2019/users>

Remote Database Authorization

- Read access possible anonymously (for now)
- Write access requires authorization!
- Based on a “bearer” token
- Header:

Authorization: Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyYb2xlIjoYXBwMjAxOSJ9.3MGDqJYkivAsiMOXwvoPTD6_LTCWkP3RvI2zpzoB1XE

BASIC FEATURES

User Creation

1. Ask the user for a user id
2. Check that the user id is available
3. Ask the user for their full name
4. Insert the new user into the users table
5. Synchronize with the remote database

Reuse the user id in all features, when that user is logged into the the app.

Following

1. Reuse the user id stored in the app
2. Let the user specify a user to follow
3. Insert the fact into the follows table
4. Synchronize with the remote database

Following is not necessarily symmetric!

Messaging

1. Reuse the user id stored in the app
2. Let the user specify a user to message
3. Let the user write a message
4. Store the message in the local database
5. Synchronize with the remote database

In the basic feature, no images and no group messages.