# The Guessing App

Examensarbete av Carl Schlyter

för Medieinstitutet

”Webbutvecklare inom e-handel”

2020-04-16

# Dokumentation för applikationen

Innehåll

[The Guessing App 1](#_Toc37938182)

[Dokumentation för applikationen 1](#_Toc37938183)

[1. Idén med The Guessing App 1](#_Toc37938184)

[1.1 Idén 1](#_Toc37938185)

[2. Applikationens delar 1](#_Toc37938186)

[2.1 Databasen 1](#_Toc37938187)

[2.2 Apiet 2](#_Toc37938188)

[2.3 Front end 2](#_Toc37938189)

## Idén med The Guessing App

1.1 Idén  
Idén med The Guessing App handlar om att göra det möjligt för en reklambyrå att aktivera sina kunder genom en rolig gissningstävling. The Guessing App anpassas då varje gång för varje enskild kund, med relevant layout, logga mm och det bestäms med kunden vad gissningstävlingen ska handla om. Det kan vara nästan vad som helst, kommande väder, finalister i Melodifestivalen, antal medaljer vid ett mästerskap eller som i det exempel som just nu finns, matchresultat i ett fotbollsmästerskap.

## Applikationens delar

2.1 Databasen

Databasen i MySQL består i exemplet av fyra tabeller: Users, Betters, Matches och Bets. Alla tabeller har automatiska primary-key-id-fält (UUID) och lösenordet i Users är hashat. Jag har just nu, för att skala ner det här examensarbetet, valt att inte använda lösenordet eller login i applikationen och man kan hävda att det inte heller är helt nödvändigt med en sådan här typ av tjänst.

Jag har använt MySQL-Triggers i Users- respektive i Matches-tabellerna för att automatisk uppdatera andra tabeller givet vissa händelser:

**Users –** Efter ”insert” skapas automatiskt rader i Betters där användaren är anonymiserad.

**Matches** – Efter ”update” uppdateras automatiskt Bets med faktiskt matchresultat och med poäng beroende på förhållandet matchresultat och registrerad gissning i Bets.

Matches-tabellen är tänkt att uppdateras av en administratör, men det finns inget gränssnitt för det just nu, utan uppdateringen får göras direkt i databasen för att trigga den händelsen.

2.2 Apiet

Apiets aktiva models/endpoints är i exemplet:

**Game/read –** För att lista de aktuella matcherna med matchdatum senare än dagens datum.

**User/create** – För att skapa en användare i databasen.

**User/read** – För att lista en topplista på användare med deras poäng från gissningar på spelade matcher.

**Bet/read** – För att lista genomsnittliga gissningar av andra gissare på kommande matcher.

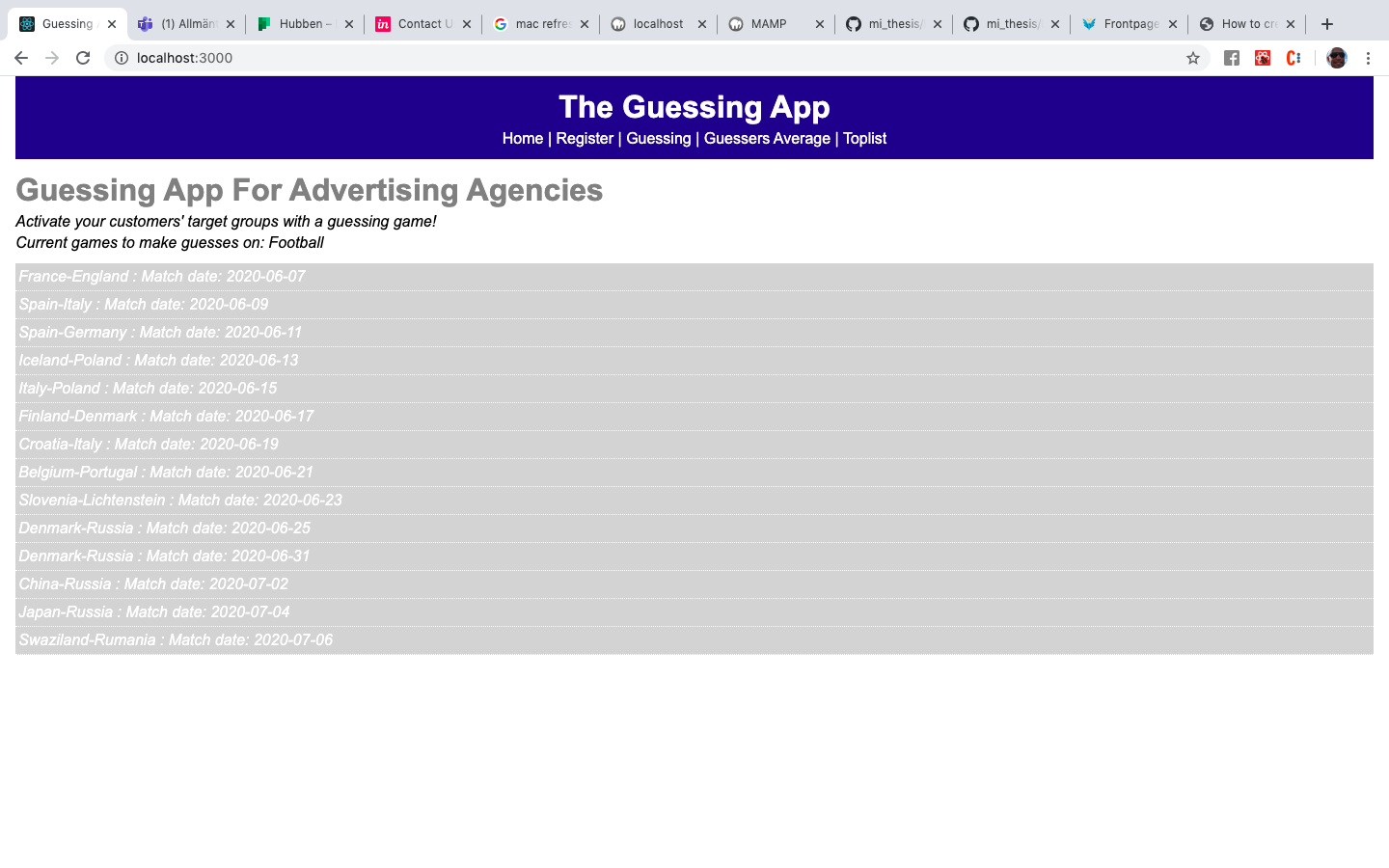
**Bet/create** – För att skapa en gissning på en kommande match.

2.3 Front end

Front end är utvecklat i REACT med Node Package Manager / Create React App med följande gränssnitt och funktioner:

***2.3.1 ”Home” / Förstasidan***

Startsidan med listan över aktuella matcher att gissa resultatet på:



***2.3.2 ”Register” / Registreringssidan***

Med ett formulär för registrering av en användare:

A screenshot of a social media post

Description automatically generated

***2.3.3 ”Guessing” / Gissningssidan***

Med ett formulär för registrering av en gissning:

**A screenshot of a social media post

Description automatically generated**

***2.3.4 ”Guessers Average” /Genomsnittliga gissningar***

Med en lista på kommande matcher som någon gissat resultatet på och genomsnittsgissningarna på de matcherna:

**A screenshot of a social media post

Description automatically generated**

***2.3.5 ”Toplist” / Topplista***

Med en lista på användare som gissat på färdigspelade matcher och rangordnade efter användarnas samlade poäng.

A screenshot of a social media post

Description automatically generated

\*\*\*\*\*\*\*