

The Fantastic Race

Erik Westervind

Examensarbete

Mobilapputvecklare MAP19

Handledare: Francisco Andreasson

Datum:

Sammanfattning

Innehållsförteckning

Inledning

Bakgrund

Syfte

Frågeställningar

Metod och material

Metod

Material

Utveckling

Tidsplan

I början sattes ett ganska aggressivt tidsschema för att känna att projektet kom framåt. Under arbetets gång behövde planeringen självklart uppdateras och revideras för att anpassas till olika situationer som uppstod.

Det ursprungliga schemat såg ut så här

Januari:

Bestämma layout och göra skisser i Adobe XD. Sätta upp projektet, databas och inloggning i appen

Februari:

Bygga logik för hur tävlingar skapas, delas och accepteras

Mars:

Påbörja spelfunktionalitet, addera kartvyer

April:

Färdigställa spelfunktionalitet, visa ut de tävlande på kartan

Maj:

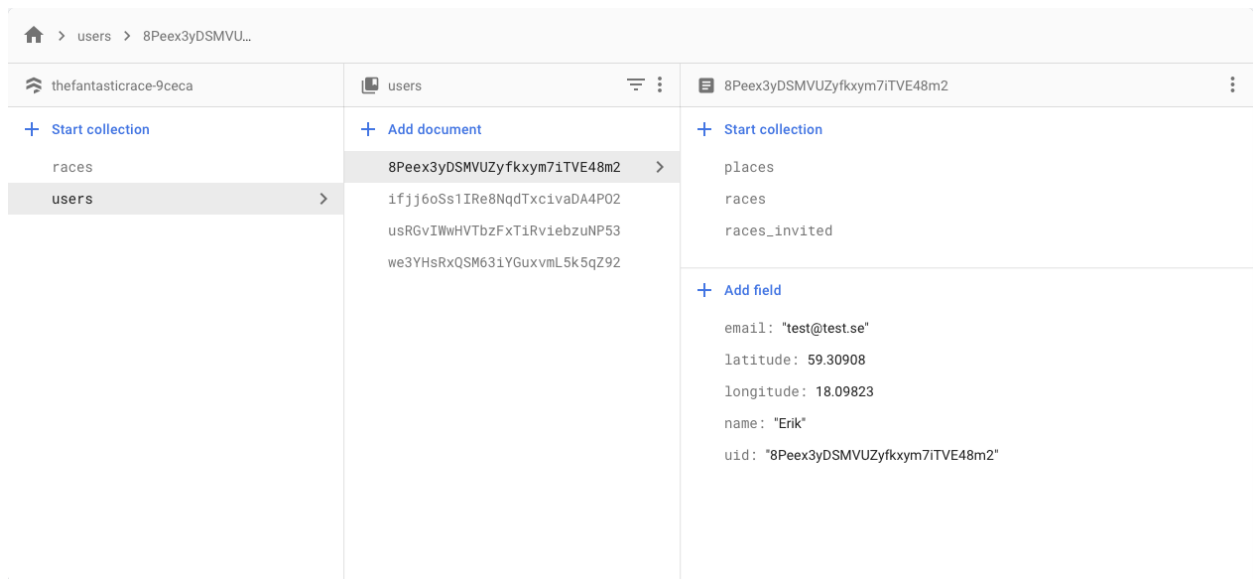
Addera notifieringar, kryptering, färdigställa appen, lansera på App Store

Funktionalitet

Design

Databas

Till projektet användes Firebase Cloud Firestore för databas. Eftersom databasen behöver spara användares positioner lades stor vikt på att sätta databasens regler så att endast de som var behöriga skulle kunna läsa av informationen. Men det skulle samtidigt vara smidigt att göra anropen mot databasen från appen. Firebase användes även för kontohantering och inloggning av användare.



```

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    //Rule to access races collection:
    match /races/{documentID} {
      //allow read: if request.auth.uid != null in get(/databases/{database}/documents/races/{documentID})
      //allow read: if exists(/databases/{database}/documents/races/{documentID}/users/{request.auth.uid})
      allow read: if request.auth.uid != null
    }

    //Rule to access list of users in a race:
    match /races/{documentID}/users/{userID} {
      allow read: if request.auth.uid in get(/databases/{database}/documents/races/{documentID}).data.invited
      allow create, update, delete: if request.auth.uid == userID &&
        request.auth.uid in get(/databases/{database}/documents/races/{documentID}).data.invited
    }

    //Rule to access the stops of a parent race in order for the user to create their own data:
    match /races/{documentID}/stops/{stopID} {
      allow read: if request.auth.uid in get(/databases/{database}/documents/races/{documentID}).data.invited
    }

    //Rule to access user document:
    match /users/{userID} {
      allow read: if request.auth != null
      allow create, update, delete, write: if request.auth.uid == userID
    }

    //Rule to access users stops:
    match /users/{userID}/places/{placeID=**} {
      allow read, update, delete, create: if request.auth.uid == userID
    }

    //Rule to access users collection of invited games:
    match /users/{userID}/races_invited/{raceID=**} {
      allow read, update, delete, create: if request.auth.uid == userID
    }
  }
}

```

Resultat