

# The Fantastic Race

Erik Westervind

Examensarbete

Mobilapputvecklare MAP19

**Handledare:** Francisco Andreasson

**Datum:**

# Sammanfattning

## Innehållsförteckning

## Inledning

### Bakgrund

The Amazing race är ett amerikanskt TV-format där lag tävlar mot varandra om att ta sig till olika platser runt om i världen. Det lag som kommer fram till slutmålet först vinner en miljon dollar. Programmet har fått något av en kultstatus inom vissa kretsar men det finns trots detta ingen app för den som vill skapa egna tävlingar med vänner och bekanta. Trots programmets popularitet finns ingen app där användare kan skapa egna liknande tävlingar med sina vänner och bekanta.

### Syfte

Syftet är att bygga en applikation som möjliggör för användare att skapa egna 'Amazing races' och låta vänner och kollegor tävla i dessa under t.ex. en eftermiddag eller helg. En deltagare ska då ta sig till en plats som beskrivs eller visas på kartan, när person tagit sig dit låses nästa plats upp, och så vidare. Möjligheten ska även finnas att se hur långt de andra tävlande har kommit och deras position på kartan.

Man ska även kunna skapa publika tävlingar där vem som helst kan tävla och se sin ranking mot andra.

### Frågeställningar

Går det att skapa en app som efterliknar tv-programmet 'The Amazing Race' som både är enkel, rolig och säker att använda? Appen kommer behöva spåra användarnas positioner, vilka möjligheter finns för att göra detta på ett så säkert och anonymt sätt som möjligt?

SwiftUI är fortfarande ett relativt nytt gränssnitt för att bygga iOS-appar med. Skulle det trots detta vara möjligt att bygga appen med hjälp av just SwiftUI?

## Metod och material

### Metod

Plattformen som valdes var Apples iOS ihop med det deklarativa ramverket SwiftUI 2.0. Informationen som krävdes för att utveckla appen har främst hämtats från Apples egen

dokumentation, Raywenderlich.com, Stackoverflow.com och olika Youtube-kanaler. Även kunskaper som inhämtats från utbildningens kurser har kommit till användning. Främst då från iOS-kursen och API- och backend-kursen.

## Material

Projektet är utvecklat på en Macbook Pro med macOS Catalina i Xcode 12.4.

## 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

Eftersom idén för appen kommer från ett existerande tv-koncept skulle funktionerna i appen efterlikna det som händer i tv-programmet men anpassas för att passa en app och vanliga människor.

### Skapa egna race

En användare kommer ha rollen som lekledare och vara den som skapar upp "reiset" och bjuder in sina vänner till att tävla. Appen kommer behöva dessa vyer för att skapa upp tävlingarna:

## Allmänna inställningar

Detta skulle vara en vy för att göra grundinställningar för tävlingen, till exempel namn och beskrivning. Men skulle även ha möjlighet att ställa in regler för tävlingen så som om den tävlande behöver svara på en fråga för att komma vidare, hur stort geofencet ska vara och starttid.

## Välja platser

Eftersom en tävling går ut på att ta sig mellan olika platser behöver man kunna välja ut dessa platser på en karta. När man har valt placeringen ska man sedan fylla i ytterligare information om platsen, till exempel fråga och svar för att komma vidare. Det behöver även vara enkelt att sortera om ordningen på platserna så man får till ett optimal rutt för de tävlandes.

## Bjuda in användare

För att det ska bli en tävling krävs det att några är med och tävlar. Det behövs därför en vy där man kan söka efter och bjuda in andra användare att delta i tävlingen. De ska då få en pushnotis och kunna välja om de vill delta eller ej. I förlängningen skulle man kunna tänka sig att man kan skapa publika tävlingar som vem som helst kan delta i och tävla på tid.

## Startsida

När användaren har loggat in i appen ska denne se en lista över alla tävlingar, både sådana som användaren själv har skapat, och de som användaren är inbjuden till.

## Tävla

Detta är den huvudsakliga funktionen i appen. Deltagarna ska, när tävlingen ha startat, ansluta och påbörja rejset. Användaren ska då se sin position på en karta och var nästa plats ligger som användaren ska ta sig till. När användaren har tagit sig till positionen ska dennes telefon genom positionering känna av det och nästa plats på kartan då visas ut. Och så vidare tills man kommer till den sista positionen, det vill säga går i mål. Först i mål vinner.

Ett mål var även att användaren skulle få en notis när den kom innanför geofencet och därmed inte behöva ha appen igång.

## Design

## Databas

Till projektet användes Firebase Cloud Firestore för databas. Eftersom databasen behöver spara användares positioner las stor vikt vid 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.

<div> <div>🏠</div> <div>&gt; users &gt; 8Peex3yDSMVU...</div> </div>	<div> <div>📁</div> <div>users</div> <div>☰ ⋮</div> </div>	<div> <div>📁</div> <div>8Peex3yDSMVUZyfkxym7iTVE48m2</div> <div>⋮</div> </div>
<div> <div>+</div> <div>Start collection</div> </div>	<div> <div>+</div> <div>Add document</div> </div>	<div> <div>+</div> <div>Start collection</div> </div>
<div> <div>races</div> <div>users &gt;</div> </div>	<div> <div>8Peex3yDSMVUZyfkxym7iTVE48m2 &gt;</div> <div>ifjj6oSs1IRe8NqDTxcivaDA4P02</div> <div>usRGvIWwHVTbzFXt1RviebzuNP53</div> <div>we3YHsRxQSM63iYGuxvml5k5qZ92</div> </div>	<div> <div>places</div> <div>races</div> <div>races_invited</div> </div>
		<div> <div>+</div> <div>Add field</div> <div>email: "test@test.se"</div> <div>latitude: 59.30908</div> <div>longitude: 18.09823</div> <div>name: "Erik"</div> <div>uid: "8Peex3yDSMVUZyfkxym7iTVE48m2"</div> </div>

```
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.invitedUsers
      allow create, update, delete: if request.auth.uid == userID &&
        request.auth.uid in get(/databases/{database}/documents/races/{documentID}).data.invitedUsers
    }

    //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.invitedUsers
    }

    //Rule to access user document:
    match /users/{userID} {
      allow read: if request.auth.uid == userID
      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
    }
  }
}
```

Att integrera Firebase med ihop med SwiftUI krävde en del funderingar då Googles egna dokumentation ännu inte är uppdaterad. Valet gjordes att följa en guide av Stewart Lynch då integrationen var smidig och tydlig.

## Resultat