

# The Fantastic Race

Erik Westervind

Examensarbete

Mobilapputvecklare MAP19

**Handledare:** Francisco Andreasson

**Datum:** 2021-05-23

# Sammanfattning

Under pandemin har behovet av att träffas och umgås utomhus ökat. Många människor vill även minska sin skärmtid och upplever att telefonen tar upp för stor del av deras liv. Samtidigt erbjuder tekniken stora möjligheter.

Målet var att ta fram en app där vänner och bekanta kan ses utomhus och tävla mot varandra i en form av geocaching där man tar sig mellan olika platser. Inspirationen kommer från det amerikanska TV-programmet The Amazing Race.

Appen skulle utvecklas för iOS 14 med hjälp av Apples relativt nya ramverk SwiftUI. En frågeställning var därför om det skulle innebära problem att använda sig av ett så nytt ramverk för den här typen av app. Detta visade sig till största delen inte stämma.

Vissa av de ursprungliga funktioner fick på grund av omfattningen utelämnas. Resultatet blev en app som på många sätt påminner om TV-programmet the Amazing Race och som kan vidareutvecklas.

## Abstract

During the pandemic the need for meeting friends and family outdoors have increased. Many people also want to cut down on the amount of time they spend in front of the mobile screen. But at the same time the technique offers endless possibilities.

The goal was to develop an app where friends and family could meet outside and compete against each other in a type of geocaching game where you go to different spots. The app is inspired by the American TV show The Amazing Race.

The app was going to be developed for iOS 14 with Apple's new framework SwiftUI. A question was therefore if this could mean problems, working with a new framework for this kind of app. This turned out wasn't the case.

Some of the original functions of the app had to be cut out of the final product due to lack of time. The result was an app that in many ways is similar to the TV show The Amazing Race and that can continue to be improved with new features.

# Innehållsförteckning

<b>1. Inledning</b>	<b>4</b>
1.1 Bakgrund	4
1.2 Syfte	4
1.3 Frågeställningar	4
<b>2. Metod och material</b>	<b>5</b>
2.1 Metod	5
2.2 Material	5
<b>3. Utveckling</b>	<b>5</b>
3.1 Tidsplan	5
3.2 Funktionalitet	5
3.2.1 Startside	6
3.2.2 Skapa egna race	6
3.2.2.1 Allmänna inställningar	6
3.2.2.2 Välja platser	6
3.2.2.3 Bjuda in användare	6
3.2.3 Tävla	6
3.3 Design och Navigation	7
3.3.1 Färger	7
3.3.2 Navigation	7
3.4 Databas	8
<b>4. Resultat</b>	<b>10</b>
4.1 Färdigt resultat	10
4.2 Problem och utmaningar	12
4.2.1 Förbättringar av designen	12
4.2.2 Bakgrundsuppdateringar av data med Cloud Functions	12
4.2.3 Notiser	12
4.2.4 Unik design på pins och annoteringar på kartan	12
4.2.5 Inlogg med Google-konto	12
4.2.6 Testning	13
<b>5. Diskussion och slutsatser</b>	<b>13</b>
5.1 Framtiden för appen	13
5.2 Slutsats	13
<b>6. Källor</b>	<b>14</b>

# 1. Inledning

## 1.1 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.

## 1.2 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.

## 1.3 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<sup>1</sup>. Skulle det vara möjligt att bygga appen med hjälp av just SwiftUI? MapKit är ännu inte helt integrerat med SwiftUI, skulle det innebära att projektet blev svårare eller omöjligt att genomföra?

---

<sup>1</sup> Benjamin Mayo, 9to5mac.com, <https://9to5mac.com/2019/06/03/apple-announces-swiftui-a-modern-declarative-user-interface-framework-for-apple-platforms/>, hämtad 12 maj 2021

## 2. Metod och material

### 2.1 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.

### 2.2 Material

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

## 3. Utveckling

### 3.1 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

### 3.2 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. I The Amazing

Race förekommer en del moment som inte passar eller skulle tillföra något i en app. Appens funktion fokuserade därför på huvudkonceptet, att ta sig till en plats för att därifrån ta sig vidare till nästa plats.

### 3.2.1 Start sida

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. Listelementet skulle bland annat visa tid för start, antal deltagare och vem som hade skapat tävlingen och uppdateras i realtid från databasen.

### 3.2.2 Skapa egna race

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

#### 3.2.2.1 Allmänna inställningar

Detta var 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.

#### 3.2.2.2 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.

#### 3.2.2.3 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. Sökningen av användare sköts på ett enkelt sätt via Firebase queries.

### 3.2.3 Tävla

Detta var den huvudsakliga funktionen i appen. Deltagarna ska, när tävlingen ha startat, ansluta och påbörja racet.

I TV-programmet The Amazing Race får de tävlande instruktioner eller ledtrådar om vart de ska ta sig i världen<sup>2</sup>. För att göra appen roligare och mer intuitiv får appanvändaren en karta till sin hjälp. Användaren kan 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.

---

<sup>2</sup> The Amazing Race - Wikipedia.org,  
[https://en.wikipedia.org/wiki/The\\_Amazing\\_Race\\_\(American\\_TV\\_series\)#Concept](https://en.wikipedia.org/wiki/The_Amazing_Race_(American_TV_series)#Concept), hämtad 12 maj 2021

TV-programmet Amazing race har olika deltävlingar och olika speciella moment som inte finns med i appen. Detta då det skulle bli för krångligt att implementera och inte skulle tillföra appen nämnvärt mycket.

Förutom placeringen av var nästa stopp befinner sig, kan man även se de andra deltagarnas position på kartan. Detta förutsatt att spelledaren har ställt in att den funktionen ska vara tillgänglig.

Ett mål var även att användaren skulle få en notis när den kom innanför geofencet och därmed inte behöva ha appen igång. Detta implementerades men under testningen upptäcktes att vissa anpassningar kan behöva göras för att det ska fungera optimalt. I vissa fall kunde användaren få upp en notis trots att de bara var i närheten av platsen. Detta beror troligtvis på att positionsinhämtningen är lite mer flexibel och mindre exakt när appen är i bakgrundsäge.

För att hantera kartor, geofence och användarpositioner användes guider och inspiration från RayWenderlich<sup>3</sup> och Hacking With Swift<sup>4</sup>. Apples kartor är ännu inte helt anpassade för att fungera native med SwiftUI. För att kunna använda geofences mm används därför en så kallad UIViewRepresentable som gör att man kan använda UIKit i SwiftUI.

## 3.3 Design och Navigation

En stor utmaning var att få till appens UI och UX på ett sätt som gjorde den både enkel att använda och att utveckla för. SwiftUI är relativt nytt så det fanns en risk att vissa funktioner skulle ta lite tid att bygga och att anpassningar och avkall på designen skulle behöva göras för att åstadkomma dessa inom en rimlig utvecklingstid. Det bestämdes därför att de designskisser som hade tagits fram skulle fungera som riktlinjer men att dessa kunde frångås om det skulle innebära en smidigare utveckling.

### 3.3.1 Färger

Appens design bestämdes till att vara generellt avskalad, och använda sig av ett analogt färgschema (Analogous color scheme<sup>5</sup>) med lila som primär färg. Lila symboliserar enligt den klassiska färgläran t.ex. fantasi och mysterium. Den sekundära färgen som skulle användas mest var en turkos nyans som gick bra ihop med den lila

### 3.3.2 Navigation

En hypotes var att en klar majoritet av användarna endast skulle använda appen för att delta i en tävling. De skulle förmodligen aldrig eller sällan skapa egna race. Tanken var därför att det skulle gå snabbt att logga in och komma igång. Varken de vanliga navigationselementen *Tab bar* eller en hamburgermeny

---

<sup>3</sup> SwiftUI Maps & Location Fundamentals. (2020, september 29). Raywenderlich. <https://www.raywenderlich.com/14255236-swiftui-maps-location-fundamentals/>, hämtad 23 maj 2021

<sup>4</sup> Hudson, P. (2019, december 2). *Advanced MKMapView with SwiftUI*. Hacking With Swift. <https://www.hackingwithswift.com/books/ios-swiftui/advanced-mkmapview-with-swiftui>, hämtad 23 maj 2021

<sup>5</sup> Tigercolor.com, <https://www.tigercolor.com/color-lab/color-theory/color-harmonies.htm>, hämtad 12 maj 2021

användes. Istället kom användaren direkt till en lista av race så att man snabbt skulle kunna starta den aktuella tävlingen.

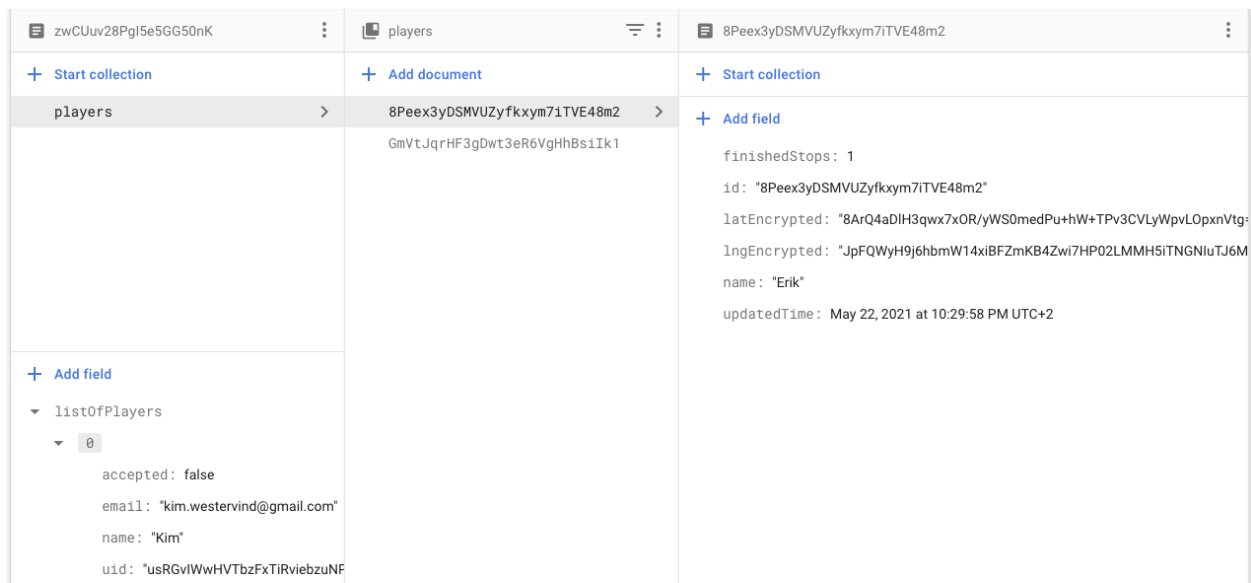
Väl inne i tävlingen möts man av en karta med instruktioner och olika tabs där man kan välja byta vy för att se resultattavlan och en vy för överblick över platserna.

De olika vyerna öppnas sedan med SwiftUI:s `fullScreenCover` som liknar sheets men fyller ut hela skärmen. Detta passade bra eftersom det inte skulle vara någon djupare navigering inne i de olika vyerna. För ikoner användes dels Apples SF-symbols<sup>6</sup>, men även ikoner from Flaticon. De anpassades då för att fungera både för Light- och Dark mode.

## 3.4 Databas

Till projektet användes Firebase Cloud Firestore som databas. Eftersom databasen behövde spara användares positioner lades vikt vid att skapa databasens regler så att endast de som var behöriga skulle kunna läsa av informationen. En utmaning var då att samtidigt bygga databasstrukturen så att den skulle vara logiskt och enkelt att arbeta med när man skulle göra anrop mot den. Användarnas positioner krypterades även för att man inte skulle kunna se dennes position om man tittade i databasen.

Under projektets gång behövde strukturen av databasen uppdateras ett antal gånger för att hitta en logik som passade appen.



<sup>6</sup> Apple SF symbols, <https://developer.apple.com/design/human-interface-guidelines/sf-symbols/overview/>, hämtad 12 maj 2021



```

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

    //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.invites
    }

    //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
    }
  }
}

```

*En tidig version av reglerna för databasen*

Att integrera Firebase ihop med SwiftUI krävde en del funderingar då Googles egna dokumentation inte var uppdaterade. Valet gjordes att följa en guide av Stewart Lynch<sup>7</sup> då integrationen var smidig och tydlig. På köpet kunde användarna logga in i appen med hjälp av sitt iCloud-konto.

---

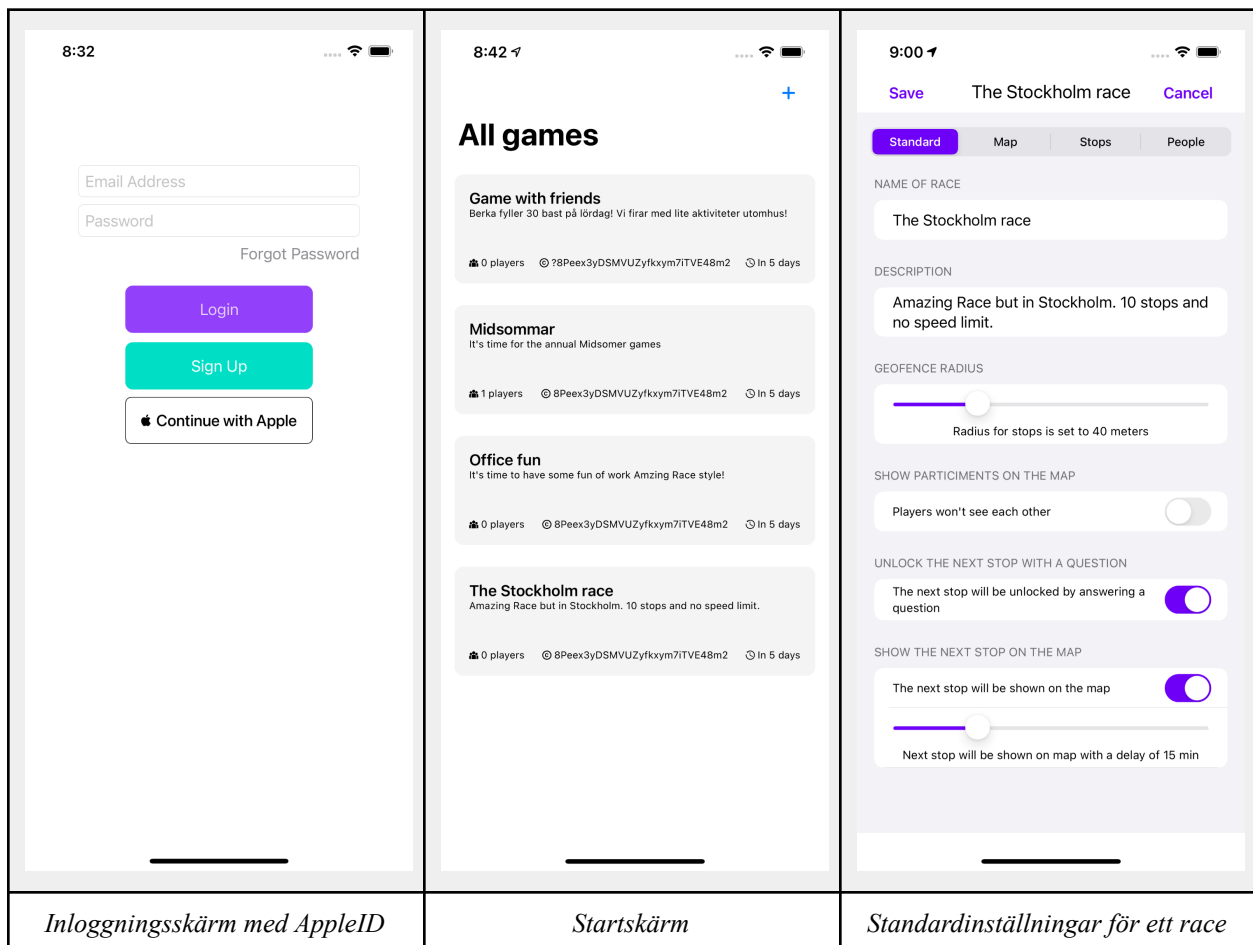
<sup>7</sup> Stewart Lynch,  
[https://www.youtube.com/watch?v=r0fSiNaYjPI&list=PLBn01m5Vbs4B79bOml3FL\\_MFxiXVuDrma](https://www.youtube.com/watch?v=r0fSiNaYjPI&list=PLBn01m5Vbs4B79bOml3FL_MFxiXVuDrma),  
 hämtad 12 maj 2021

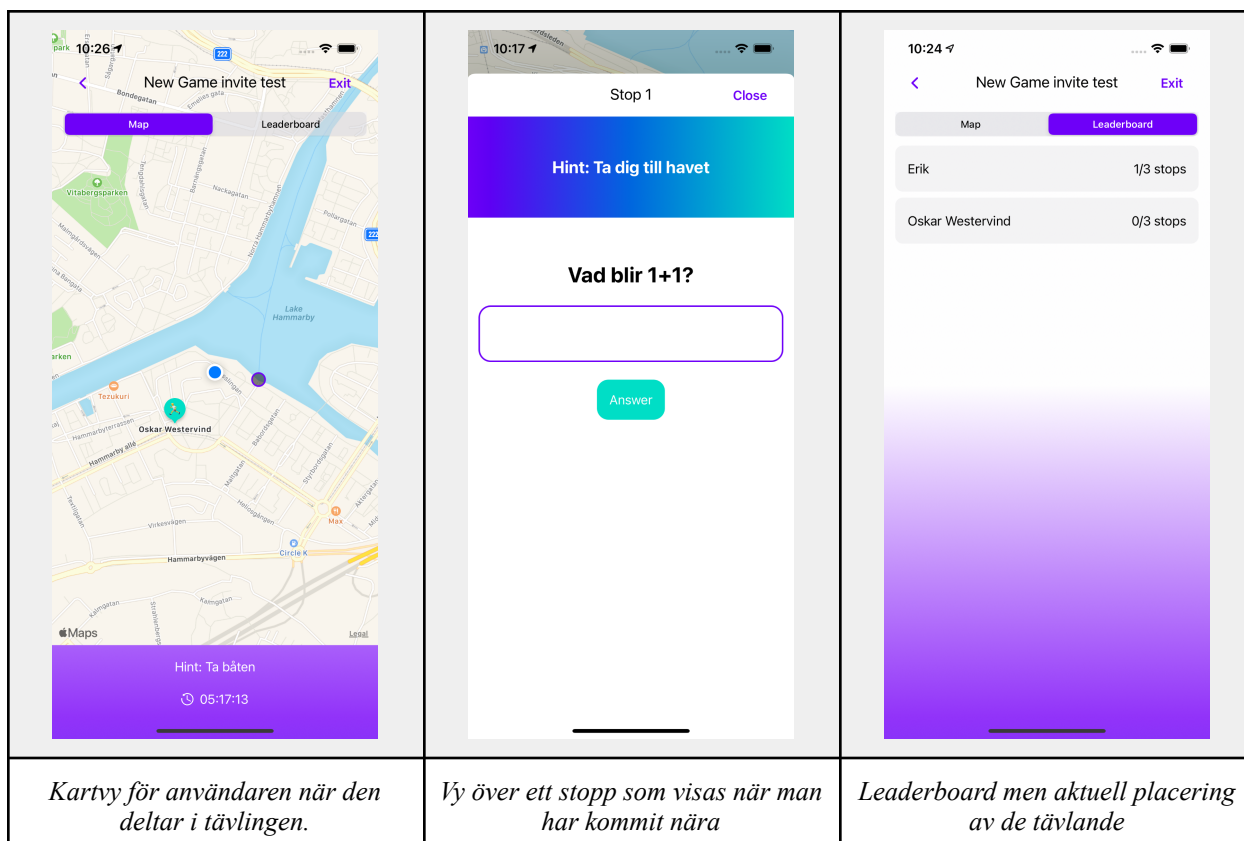
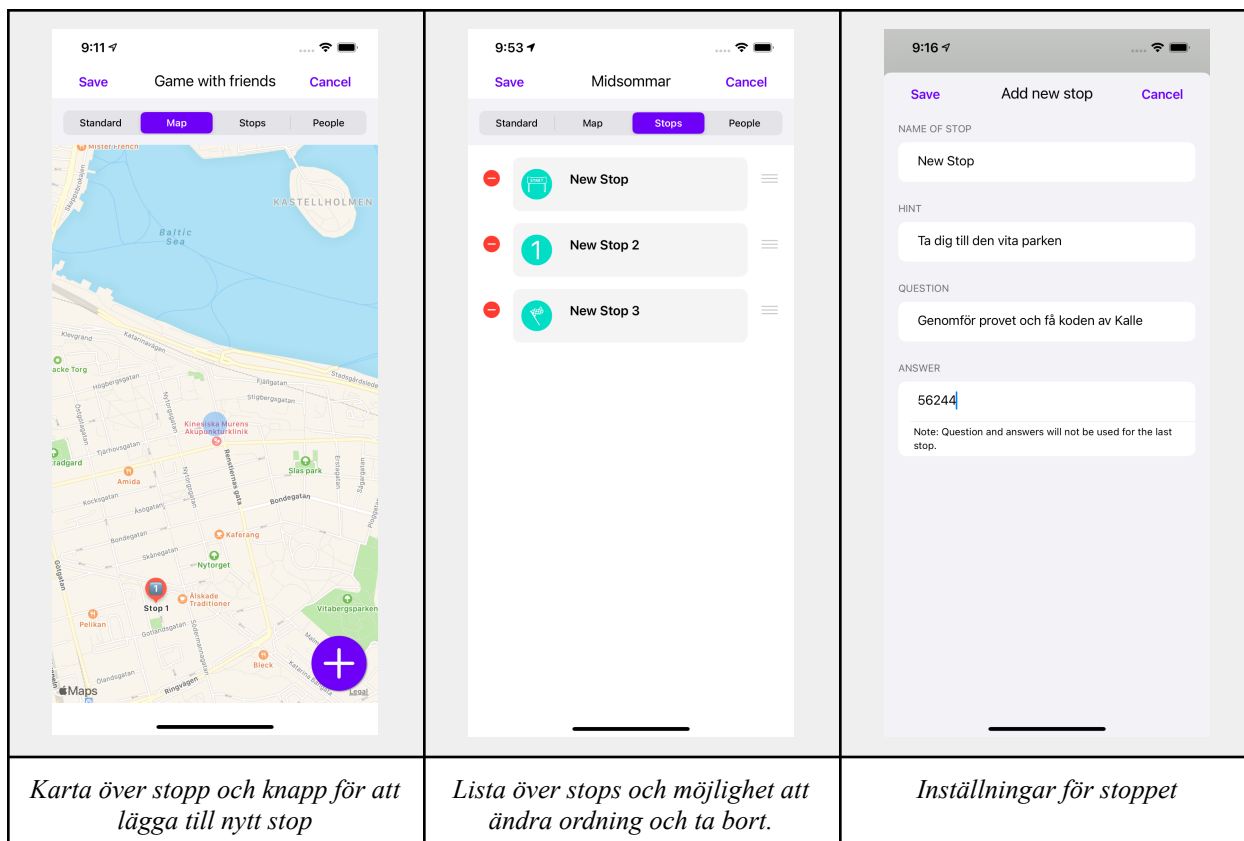
## 4. Resultat

### 4.1 Färdigt resultat

Under projektets gång visade det sig att farhågorna kring att det skulle vara svårt att utveckla appen för SwiftUI till stor del var överdrivna. Många enklare uppgifter som till exempel att visa upp listor och ändra färger på ett element vägde upp det merjobb som och kunskapsluckor som fanns kring de andra mer avancerade delarna i projektet som involverade gps och kartor.

Det färdiga resultatet med exempel på de viktigaste vyerna.





## 4.2 Problem och utmaningar

Vissa av de ursprungliga funktionerna fick under projektets gång prioriteras bort. Vissa delar av appen blev inte heller lika genomarbetade som det från början var tänkt. Anledningen var främst att del av grundfunktionaliteten tog längre tid än planerat att utveckla och att det därför behövde prioriteras.

Funktioner som inte kom med eller blev helt färdigställda:

### 4.2.1 Förbättringar av designen

När projektet startades fanns det en initial tanke med hur UI och UX skulle se ut och fungera, samt vilka färger som skulle användas. Eftersom det fanns en del osäkerhet i hur vissa lösningar enklast kunde implementeras med SwiftUI las medvetet mindre tid på design. Bland annat är vissa av appens färger och ikoner ännu inte uppdaterade för att fungera bra i iOS Dark mode-läge.

### 4.2.2 Bakgrundsuppdateringar av data med Cloud Functions

För att minska logiken i själva appen för framförallt vän-inbjudningarna användes Firestore Cloud Functions. Detta skulle även krävas för att kunna använda pushnotiser i appen. På grund av bristande kunskap är funktionerna enkelt uppsatta och det finns en del förbättrings- och optimeringspotential.

### 4.2.3 Notiser

En feature som från början var påtänkt var möjligheten att skicka pushnotiser till de tävlande. Till exempel skulle den tävlande kunna få en notis om en motståndare kom fram till ett stopp före och på så sätt stressa och sporra den tävlande att skynda sig. Pushnotiser var något som inte blev klart i denna version.

Däremot finns en funktion för lokala notiser. Så om man har skärmen släckt eller är inne i en annan app och kommer fram till ett stopp, får man en notis om att man är framme.

### 4.2.4 Unik design på pins och annoteringar på kartan

Då det initialt fanns en osäkerhet i hur vissa tekniska funktioner bäst skulle lösas i appen lades fokus på just de tekniska bitarna före design av kartan och elementen som visas ut. En lösning som inte implementerades var möjligheten att visa ut en bilder på de tävlande i annoteringarna på kartan. Detta skulle dock, förutom anpassningar i kartvyn, kräva ytterligare utveckling och vyer för att möjliggöra för användarna att ladda upp en profilbild. Det prioriterades därför bort i den här versionen.

### 4.2.5 Inlogg med Google-konto

En fördel med att använda Firebase som autentisering är att de har färdiga lösningar för att autentisera användaren med hjälp av olika tjänster. Eftersom detta är en iOS-app var det naturligt att möjliggöra inloggning med hjälp av användarens iCloud-konto. Det fanns ett mål om att lägga till fler inloggningstjänster, framförallt via ett Google-konto, men på grund av andra mer prioriterade delar i appen gjordes det aldrig.

#### 4.2.6 Testning

Det hade varit önskvärt att både hinna testa appen mer av både riktiga användare samt via automatiserade tester.

## 5. Diskussion och slutsatser

### 5.1 Framtiden för appen

Eftersom kunskapsnivån och tiden var begränsad finns det fortfarande en del utvecklingspotential. För att lansera appen kommer det förmodligen krävas en startguide som användaren får upp första gången för att konceptet ska bli tydligare. Innan en eventuell lansering på App Store behöver den även genomgå ett mer organiserat testande där riktiga användare testat att den fungerar när man springer runt på stan och spelar.

### 5.2 Slutsats

På grund av Covid-19 har människors behov av att träffas och umgås utomhus växt sig större. Många vill även minska sin skärmtid av olika anledningar då man upplever att telefonens appar stjälar fokus från individen istället för att vara det hjälpmedel det från början var tänkt att vara. I det här projektet får användaren möjlighet att träffas och tävla mot vänner och bekanta utomhus. Projektet är utvecklades med hjälp av SwiftUI, vilket var lätt att utveckla med. Även funktioner som ännu inte var anpassade för SwiftUI gick bra att bygga med hjälp av bakåtkompatibiliteten.

Telefonens och apparers möjlighet tas samtidigt till vara utan att det för den delen innebär ohälsosam skärmtid. Idén för The Fantastic Race kommer ursprungligen från tv-programmet The Amazing Race. Av förklarliga skäl har konceptet ändrats och anpassats för att passa app-formatet och en tävling mellan vänner. En person som har sett och uppskattar tv-programmet borde känna igen sig i appen och gilla den upplevelse som den förmedlar.

## 6. Källor

Apple inc. (n.d.). *SF Symbols*. Apple. Retrieved maj 12, 2021, from

<https://developer.apple.com/design/human-interface-guidelines/sf-symbols/overview/>

Hudson, P. (2019, december 2). *Advanced MKMapView with SwiftUI*. Hacking With Swift.

<https://www.hackingwithswift.com/books/ios-swiftui/advanced-mkmapview-with-swiftui>

Lynch, S. (2020, mars 23). *Firebase Login With SwiftUI*. Youtube.com. Retrieved maj 12, 2021, from

[https://www.youtube.com/playlist?list=PLBn01m5Vbs4B79bOmI3FL\\_MFxjXVuDrma](https://www.youtube.com/playlist?list=PLBn01m5Vbs4B79bOmI3FL_MFxjXVuDrma)

Mayo, B. (2019, juni 3). *Apple announces SwiftUI, a modern declarative user interface framework for*

*Apple platforms*. 9to5mac.com. Retrieved maj 12, 2021, from

<https://9to5mac.com/2019/06/03/apple-announces-swiftui-a-modern-declarative-user-interface-framework-for-apple-platforms/>

Naumov, A. (2019). *ViewInspector*. View Inspector. <https://github.com/nalexn/ViewInspector>

*SwiftUI Maps & Location Fundamentals*. (2020, september 29). Raywenderlich.

<https://www.raywenderlich.com/14255236-swiftui-maps-location-fundamentals/>

Tigercolor. (2015). *Color Harmonies*. Tiger Color. Retrieved maj 12, 2021, from

<https://www.tigercolor.com/color-lab/color-theory/color-harmonies.htm>

Wikipedia.org. (2007, april 9). *The Amazing Race*. Retrieved maj 12, 2021, from

[https://en.wikipedia.org/wiki/The\\_Amazing\\_Race\\_\(American\\_TV\\_series\)#Concept](https://en.wikipedia.org/wiki/The_Amazing_Race_(American_TV_series)#Concept),