RunForIT

Research

Table Of Contents

Introduction

The goal of this project research is to find the answers for the questions and problems that may arise during the planning,research or development of the project. This document will also be used to determine the most suitable development tools for the project.

The questions that need to be answered are:

- Is there a space in the market for Running Assistant app?
- What similar applications are out there ?
- Android Native, Web App or Hybrid ?
- Which frontend tool will be used during the development ?
- Which backend service will be used ?
- How the communication between the app and the server will be established ?
- Which hosting service will be used for the data storage ?
- How some of the most important functionalities will be implemented ?
  - Spotify API
  - Text to Voice conversion
  - Speed calculation

Zombies, Run!

Is a hybrid app available on both iOS and Android, which takes a new look on running assistant apps.
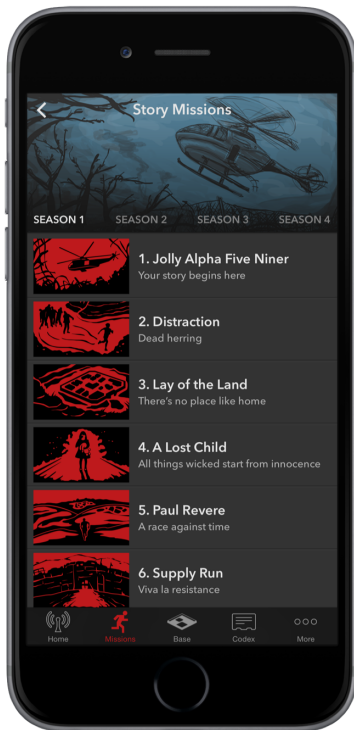
About Zombies,Run!

Story driven app

Targets gamers

500,000 downloads

In-app purchases are essential to achieve certain tasks

Monthly subscription, which enables additional missions and new game modes

Runtastic

is an Android free GPS running and fitness app, that support tracking form multiple
activities eg. running, biking or walking.

About Runtastic

Integrated with Google Fit

Integrated music player

Premium Membership, which unlock features like:
Personal training plans,
Advanced Stats & Analysis,
Weekly Fitness Report.

626,000 Downloads on Play Store.

In-app purchases

Running Trainer

Is a free android GPS tracking app, with no subscription based membership or adds.
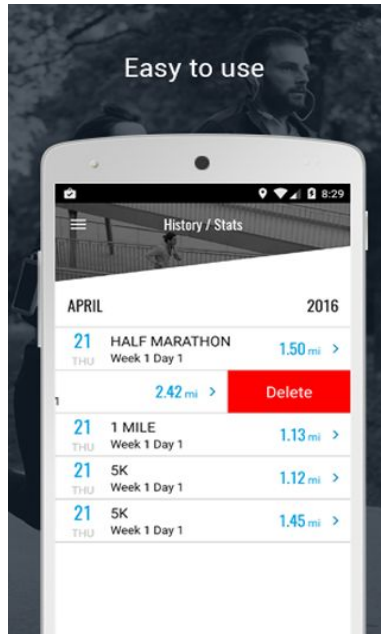
About Running Trainer

Professionally designed training programs

Accurate calorie counting

Facebook and email result sharing

Voice encouragements and statistics are updated live

In-app purchases

100,000+ Installs

Running Tracker

Lightweight and easy to use free GPS tracker developed for Android phones.

About Running Tracker
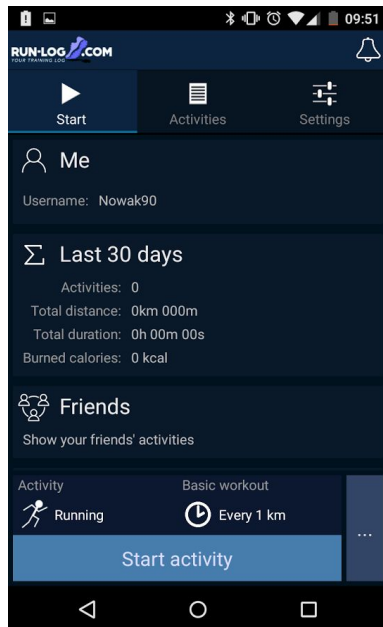
Integrated with Bluetooth Smart heart rate monitors

Integrated with Smartwatches, for increased accessibility

Share, Rate and Watch workouts of other users

500,000+ Installs

User targeted ads

Similar applications conclusion

During my research I found out that, currently there are many running assistant/GPS tracking apps out there, but most of them are unusable due to poor localisation, week UI and/or missing functionalities. The apps that I have decided to list in this document share many functionalities but at the same time each app has some unique features, eg.:

- **-Zombies, Run!**
  - Story driven gameplay increases the potential of the users to stay with the app, and use it on regular basis.
- **-Runtastic**
  - Integrated music player which combines functionalities of different apps into one,which allows the users to access everything they would need during the run within a single activity.
- **-Running Trainer**
  - Is the most interesting app that I have tested, it allows the user to share their results with their friends, additionally voice encouragement is a unique and interesting functionality which may be a significant factor while choosing the app.
- **-Running Tracker**
  - Due to the support of smart heart rate monitors and smartwatches it enables new area of features not seen in the apps listed above. Some of the new features are:
    - Heart pulse monitoring
    - Body temperature monitoring

Some of the researched features that I consider to use in my app are:
- Achievements/Milestones
- Integrated music player(Spotify API)
- Social login/Result sharing
- Voice encouragements

Frontend

Android Visual Studio is an IDE developed by Microsoft for Android development. Apps developed in Android Visual Studio are Native, which means that these apps can only be ran on Android devices. In order to run the app on a different platform(eg. iOS) all of the code would have to be rewritten using IDE which support iOS development, which makes the multi-platform development more costly and time consuming.

Some of the advantages of Android Visual Studio:
- Gradle
- App Emulation

Ionic is an open source IDE targeted at hybrid development. It supports iOS 6+ and Android 4.1+, support for Windows phone is planned. Ionic offers a library of mobile-optimized HTML, CSS and JS components, gestures, and tools for building highly interactive apps. Constructed with Sass and optimized for AngularJS[1].

Some of the advantages of Ionic:
- Support for gestures
- Deployed using Cordova, which allows for many pre built plugins
- Detailed Documentation

Xamarin[2] is an open source IDE for hybrid development, also can be used inside of Visual Studio IDE. The apps are written in c# and it uses mono framework compiler. Hybrid apps are developed for both Android and iOS platforms. The advantage of hybrid development is that the same code can be used on both platforms, which can reduce the development time significantly.

Some of the advantages of Xamarin:
- Uses Native SDK for both Android and iOS calls
- Cost effective
- Can be used with Visual Studio

OnSen UI is an another example of open source IDE used in hybrid development, it supports Android 2.3+, iOS 6+ , Firefox OS, Chrome, and Safari. OnSen UI supports multiple frameworks(AngularJS, Angular2, React, Vue.js and Meteor), which can be selected by the preference of the developer.[3]

Some of the advantages of OnSen UI:
- Most important tools are stored in the cloud(Debugger, Cordova plugins, etc.)
- Performance optimized for mobile
- Small learning curve

Backend

Skygear[4] is a subscription based all-in one type of a backend(BAAS), also offering development account with all the features available for free + a simple server to deploy your backend.

> Server Specifications

- 2GB file and data storage
- Unlimited Push Notification
- 10GB Data Transfer
- No SSD Storage
- No High Availability
- Technical support via Forum
- Sleeps after 30 mins of inactivity

Sample code written in JS[5]

//Saving files to the server using Skygear

```
skygear.publicDB.save(new Note({
  'content': 'Hello World!'
})).then((record) => {
  console.log(record);
}, (error) => {
  console.error(error);
});
```

Advantages of Skygear:

- Detailed documentation
- Many tutorials available
- JS, iOS and Android SDK available
- Supports social login for Facebook, Twitter, Google Plus, Pinterest and Tumblr platform

BaasBox is a BAAS type of a backend, this is the only backend researched in this document that I have used before the start of this project. BaasBox does not come as a package with the server, which will require me to install it on my private server. Bassbox can be set up very easily, all it requires is a server with a running linux system. Documentation provided by the developers in the most cases is not usable (eg. According to BaasDoc, social login implementation can be done in two different ways but neither of them is fully functional).

Sample Code[6]
//Saving a document on the BaasBox server

```
var post = new Object();
post.title = "My new post title";
post.body = "Body of my post.";
BaasBox.save(post, "posts")
  .done(function(res) {
    console.log("res ", res);
  })
```

Advantages of BaasBox:
- Easy to set up
- Server side scripting available
- Uses REST calls
- OrientDB

Firebase is a simple cloud based platform which can be integrated into real time apps on any web and mobile platform. It enables apps to write data, and get updates about changes to the database, instantly. Firebase also supports offline capability and your data is synchronized instantly as soon as the app regains network connectivity. Firebase stores your data as JSON documents.

Sample Code[7]
//Sending data to the Firebase server

// Base64 formatted string
var message =

'5b6p5Y+344GX44G+44GX44Gf77yB44GK44KB44Gn44Go44GG77yB';
ref.putString(message, 'base64').then(function(snapshot) {
  console.log('Uploaded a base64 string!');
});

Advantages of Firebase:
- Easy to set up
- Uses REST calls
- Cloud server comes as a package
- RESTFul API with JSON support

Apache userGrid - Is an open source backend theh offers complete SDKs for iOS, Android, HTML5/JS, Node.js, Ruby, Java, .NET, and PHP. The Usergrid SDKs work by making RESTful API calls from your application to the API. In some cases, both synchronous and asynchronous calls are supported for most methods, as in the case of the Usergrid Android SDK, while in others only asynchronous calls are supported.[8]

- Live Chat, Support forum
- Multiple SDKs available
- Java Based
- JSON calls

RESTful API using JSON

For my project I decided to go with a RESTFul API using JSON for my project as I already have some experience and would like to increase my knowledge in the area:

Advantages of using RESTful API:
- Easy to read, uses basic commands
- Uses simple methods like POST, GET, PUT and DELETE.
- Uses HTTP
- Resources are identified by URI's (Uniform resource identifiers)

Why JSON?
- Previous experience
- Lightweight
- Highly portable

Data Hosting

When it comes to hosting, during the research I found out that I have three hosting options available for this project:
- Server that came as a packet with my selected backend(eg. SkyGear)
  - Speed and functionalities may be limited
  - Once up and running, should work flawlessly
  - Quick access to support in case of an issue
- Use my private home server
  - If an issue arises, it up to me to fix it.
  - Possibility of a data loss
  - Speed limited to my home connection
  - I can control the server locally
  - System is known to me
- Rent a cheap cloud solution for the duration of the project(eg. OVH)
  - Some cloud solutions may be costly
  - Online support available
  -
  - I have some experience with using services like OVH
  - High speed/Reliable

Voice synthesis option

ResponsiveVoice.js

ResponsiveVoice is a HTML5-based Text-To-Speech library designed to add voice features to web sites and apps across all smartphone, tablet and desktop devices. It supports 51 languages through 168 voices, no dependencies[9]

Sample usage

```
<script src='https://code.responsivevoice.org/responsivevoice.js'></script>
<input onclick='responsiveVoice.speak("Hello World");' type='button' value='    Play' />
```

Mozilla Speech Synthesis API

The Web Speech API makes web apps able to handle voice data.Speech synthesis is accessed via the SpeechSynthesis interface, a text-to-speech component that allows programs to read out their text content (normally via the device's default speech synthesiser.) Different voice types are represented by SpeechSynthesisVoice objects, and different parts of text that you want to be spoken are represented by SpeechSynthesisUtteranceobjects. You can get these spoken by passing them to the SpeechSynthesis.speak() method.

Sample usage
```
<script type="text/javascript">
    var u = new SpeechSynthesisUtterance();
    u.text = 'Hello World';
    u.lang = 'en-US';
    u.rate = 1.2;
    u.onend = function(event) { alert('Finished in ' + event.elapsedTime + ' seconds.'); }
    speechSynthesis.speak(u);
 </script>
```

Native, Web App, or Hybrid ?

During my research I discovered that the best solution for my project will be going towards hybrid app development. One of the major advantages of going hybrid is that I can reach both major platforms(Android, iOS) by coding just one application and then wrapping it into a native container using Cordova (ex-PhoneGap). The amount of tools and tutorials available for the chosen app development method makes the coding more accessible for the developers who are new to the hybrid development.

Frontend

As for the frontend tool I decided to go with Ionic, which is currently huge on the market and offers everything that is needed for quick,reliable and professional development. The weighting factors for choosing Ionic over Onsen UI were the continuous development of the tool and many interesting plugins available(eg. gestures).

Backend

After researching the available backend services I decided to go with a solution that I personally haven't used before. FireBase offers integrated cloud server, Restful communication and detailed documentation on the calls and functionality.

Additional functionalities discovered

Build-in music player - I decided to go with the Spotify Api due to the huge user base and functionalities available (mood based playlist/ playlists matching your activity).
Voice encouragements - Will be a core functionality of my project, it both increases the complexity of the app and implements unique functionality.

References

Zombies, run! https://zombiesrungame.com/

Runtastic https://www.runtastic.com/en/apps/runtastic

Running Trainer https://play.google.com/store/apps/details?id=com.runningrecords.prt

Running Tracker https://play.google.com/store/apps/details?id=com.maciekcz.runlogcom

[1] http://www.gajotres.net/ionic-vs-onsenui/

[2] https://www.xamarin.com/

[3] http://www.gajotres.net/ionic-vs-onsenui/

[4] https://skygear.io/pricing/

[5] https://docs.skygear.io/js/guide/record#save

[6] http://opensource.baasbox.com/documentation/0.9.5/?javascript#create-a-document

[7] https://firebase.google.com/docs/reference/js/firebase.database.Query

[8] https://usergrid.apache.org/docs/introduction/async-vs-sync.html

[9] https://responsivevoice.org/

[10]https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API#Examples