SA (Sport Assistant)

Our innovative app synchronizes your music to your heartbeat, creating a perfect tempo whether you are exercising or just going about by your daily business.

We use the Apple Watch API to get a live feed of your heart rate and use our algorithm to pick a new track from the Spotify library when your heartbeat rate changes significantly. The algorithm is also adaptive to purpose the user plays music for. We have designed a UI for Apple Watch and iPhone which is comfortable to use in any scenarios (exercising and otherwise).

During the training tracks automatically change after 3 pulse measuring since it started playing and if your pulse had changed over than 5 BPM.

Repository

https://github.com/diakov2100/SA

Team members

Maksim Diakov (team leader, idea creator, server developer)
Alexander Naumov (designer, UI developer)
Ilia Polunin (mobile app developer)
Alexander Prosolkin (watch app developer)

List of classes

Name	Description	Path
SA Web API		
Startup	When an application starts, ASP.NET searches the primary assembly for a class named Startup and runs methods inside it	\Server\SA\src\SA\Startup.cs
Program	This class configures .NET Web Server to expose itself to Port that was specified in host.json, runs connection to DB server	\Server\SA\src\SA\Program.cs
SARepository	The repository contains logic for retrieving and mapping data to an entity model	\Server\SA\src\SA\Models\SARepository.cs
request	POST and PUT request structure	\Server\SA\src\SA\Models\request.cs
ISARepository	This interface defines basic CRUD operations.	\Server\SA\src\SA\Models\ISARepository.cs
Database	This class implements connection to DB server	\Server\SA\src\SA\Models\Database.cs
Artists	Artist DB structure	\Server\SA\src\SA\Models\SAModels\Artists.cs
Track	Track DB structure	\Server\SA\src\SA\Models\SAModels\Track.cs

Training	Training DB structure	\Server\SA\src\SA\Models\SAModels\Training.cs
User	User DB structure	\Server\SA\src\SA\Models\SAModels\User.cs
LoggingEvents	This class contains information about Event's IDs	\Server\SA\src\SA\Logs\LoggingEvents.cs
SAController	Controller for request processing	\Server\SA\src\SA\Controllers\SAController.cs
SA_DB (program + server)		
main	Main module that runs collecting data and db filling	\Server\SA_DB\main.py
rserver	Server for handling spotify authorization redirect	\Server\SA_DB\rserver.py
spotify_requests	Module that contains spotify authorization and requests processing	\Server\SA_DB\spotify_requests.py
filling_db	Module that places input data on Mongo DB server	\Server\SA_DB\filling_db.py
Mobile App		
NewWorkoutViewController	Class containing segues to player with different styles of work out	\Spotify\Spotify\NewWorkoutViewController.swift
AppDelegate	Class with implementation of authorization in app and it saves information about current user	\Spotify\Spotify\AppDelegate.swift
WelcomeViewController	Class which contains logic of “swipe” on the first page	\Spotify\Spotify\WelcomeViewController.swift
ViewController	Class for controller with authorization and moreover we set up player	\Spotify\Spotify\ViewController.swift
WatchViewController	we get pulse from watch app in this class	\Spotify\Spotify\WatchViewController.swift
PlayerViewController	here I create session which connect mobile app and watch app.This class contains logic of player and logic of alamofire requests on server	\Spotify\Spotify\PlayerViewViewController.swift
UI		
Main	Class containing all controllers for mobile app	\Spotify\Spotify\Main.storyboard
Interface	Class containing all controllers for watch app	\Spotify\SpotifyWatchApp\Interface.storyboard
Watch App		
InterfaceController	Class containing main logic of watch app : work out session, session with mobile app,receiving a request for pulse	\Spotify\SpotifyWatchAppExtension\InterfaceController

Program interface

First entry controllers:

1. Welcome Controller

    - Label “Welcome” which welcomes you with the first entrance;
    - Label “slide to continue” which helps to make the right move to segue to the next controller;

2. Authentification Controller

    - Label “Authentification” which describes you the main function of the screen;
    - Button “Spotify” which moves you to the sign up/sign on page on the Spotify’s website;

Main controllers:

3. New Work out Controller

    - Header “New Work out” which describes you the main function of the screen and serves as the main reference point to create constraints for the other objects;
    - Button “Just run” which moves you to the player screen;
    - Button “Max. rhythm” which moves you to the player screen (in the future it will change parameters of your running values) ;
    - Button “Max. distance” which moves you to the player screen (in the future it will change parameters of your running values);
    - Button “Add mine” which moves you to the player screen (in the future it will change parameters of your running values which you will create by settings);

4. Player Controller

    Header:

    - Label “Rhythm” with a beat indicator which shows your pulse in bpm (min. 60 bpm/starts with this value);
    - Buttons “+/-” which rises and reduces your pulse value, if you want to do this by yourself (tracks automatically changes after 3 pulse measuring since it started playing and if your pulse had changed over than 5 bpm);

    Player:

    - Button “Play/Stop” which starts/stops play music depending on your heartbeat; shows your pulse in bpm (min. 60 bpm/starts with this value);
    - Button “Next” which changes the music on the music with the same beat;
    - Button “End Work Out” which overs training and returns you back onto New Work out Controller;

Apple Watch controller:

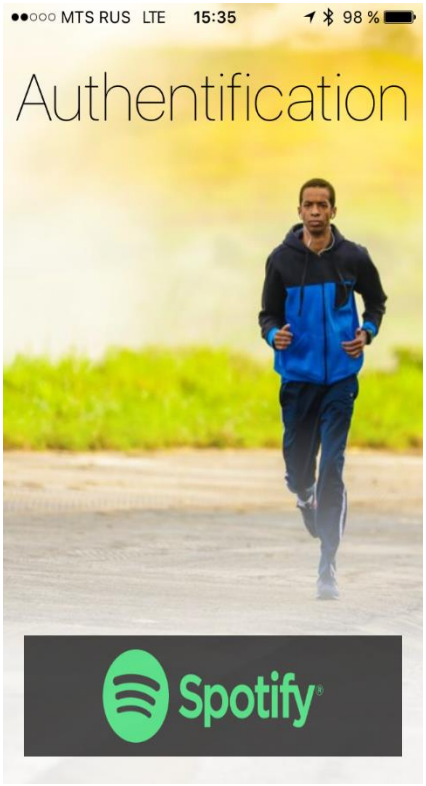
5. Start&Stop Controller

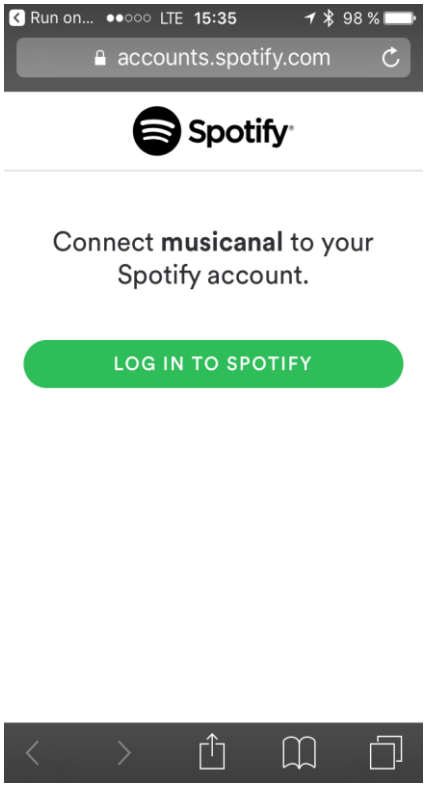
    - Button “Start/Stop” which starts/stops to measure your pulse and to send it to your iPhone.

1)

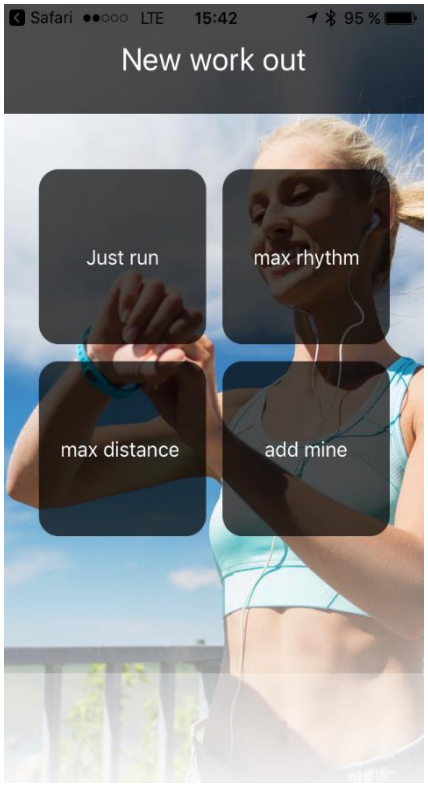

MTS RUS LTE 15:35 98 %
Welcome!
Swipe to continue >>

2)

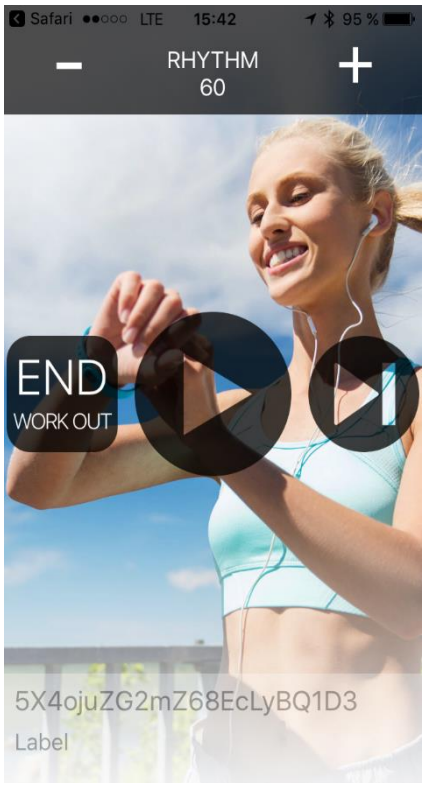

MTS RUS LTE 15:35 98 %
Authentification
Spotify


Run on... LTE 15:35 98 %
accounts.spotify.com
Spotify
Connect musicanal to your Spotify account.
LOG IN TO SPOTIFY

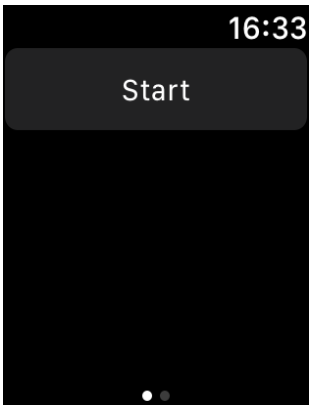
3)

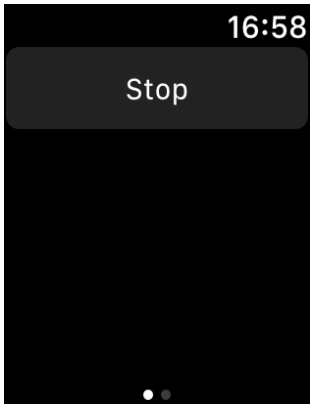

Safari LTE 15:42 95 %
New work out
Just run
max rhythm
max distance
add mine

4)


Safari LTE 15:42 95 %
RHYTHM
60
END
WORK OUT
5X4ojuZG2mZ68EcLyBQ1D3
Label

5)


16:33
Start


16:58
Stop

Challenges we ran into

Managing time, fine tuning the algorithm to produce a good heart rate to BPS adjustment and integrating Apple Watch with an iPhone in terms of interface.

What we learned

Analyzing body metrics with Apple Watch API, using Spotify SDK, integrating several devices in one app, building own web api, hosting server.

What's next for SA

Raising funds with Kickstarter to bring our app to other platforms. Life-testing.