VIA UNIVERSITY COLLEGE
ICT ENGINEERING

Project Description

SportLook App

Ibrahim Yildirim (166802)
Juraj Petrik (166740)
Alexandru Vasile (168134)

Supervisors
Ib Havn
Jan Petersen

February 2015

Project Description

Background description

Fitness and sport have been a part of human history for such a long time, and with the developing trend for healthy lifestyle, doing sports is an essential part of that lifestyle. However today the possibilities for socializing over sport events is only common if you are a member of a club (i.e. football, basket club etc.). With the success of many apps such as gomore etc. shows sign of people interested in doing activities together with other people, even if they don't know them, but as long as it is doing something that benefits each part.

A healthy lifestyle can save the government money in the long term, as you will use less as their resources (hospital, caring etc.), and sport event would be a good opportunity for the local communities around to find other people sharing the same interests as you.

Purpose

The purpose of this project is to create a client(app)/server system, where users can create and join sport events.

Problem formulation

- How to design and develop the UI of the mobile platform, in order for the users to feel completely comfortable using the system?
- What would be the best approach to develop this platform? Native-approach or hybrid-approach?
- Should we develop for both Android and iOS platforms or should we just focus on one of the platforms?
- What percent of the mobile platform(s) should we support in terms of operating system versions?
- How can we ensure the security of the platform?'
- What technology should we use as backend?
- How can we adapt the UI so that it supports multiple screen dimensions?
- What open source libraries could we use to speed up the development?
- Would it be better to develop a responsive website than developing native app(s)?
- What data should we cache?
- How can we ensure proper error handling when using the Facebook SDK (for log in, sharing etc.) ?

Delimitation

- The system will consist of a client (app) communicating with a server (back-end). For the end users, only the client will be accessible.
- The focus will be the functionality of the system and the user experience and UI design

will have a lower priority.

- There will be no support for operating system versions below iOS 7.0.
- The backend will not have any user interface, as it should not be accessible by end users.
- Scalability problems of the mobile platform will not be taken into consideration.
- There will be no cache support in the mobile app(s), the app relying on an internet connection for being functional.

Choice of model and method

What Partial problem	**Why** Why study this problem?	**Which** Which models/theories are expected to be used to solve the problem?
How to design and develop the UI of the mobile platform, in order for the users to feel completely comfortable using the system?	*It's important to have both a functional and user friendly platform in order to have repeat-users and provide a good experience. This is a must to ensure the success of the service.*	*Research on practices applied by other sharing economy platforms and also study user experience best practices. There is also a need for researching the current and latest technologies, to ensure a performant platform.*
What would be the best approach to develop this platform? Native-approach or hybrid-approach?	*In the mobile development community, there is a big discussion on what is the best approach in developing, native or hybrid. It's important to get accustomed with latest theories on this subject.*	*Research on what are the advantages and disadvantages of each approach, while having in mind our mobile platform needs.*
Should we develop for both Android and iOS platforms or should we just focus on one of the platforms?	*This is important because it's a common decision that needs to be taken when launching a new service for mobile.*	*Research on whether it would be better to focus on a specific operating system (Android, iOS) or try to support a large part of the mobile market.*
What percent of the mobile platform(s) should we support in terms of operating system versions?	*After the platform(s) to develop for are chosen, it's also important to decide what operating system versions should be supported, since this affects the availability of the service on the mobile market.*	*Research on what would be the main difference in development regarding supporting different operating system versions. Study if supporting certain versions would significantly influence the development time.*
How can we deal with user payments?	*This is a very sensitive topic, since money is involved and the users must be protected and secured.*	*Research on best practices used by other sharing economy platforms. Consider using mobile payments platforms as a service. (for example BrainTree)*

How can we ensure the security of the platform?	*This is yet another sensitive topic, since leaking of sensitive user data will cause the downfall of the platform, users not trusting it anymore.*	*Research regarding common threats and best practices. Testing if there are security issues after the platform is developed.*
What technology should we use as backend?	*This is important since the platform needs high availability and performance. The performance must be assured also for the backend, not only for the client side.*	*Research on current available backend technologies and consider using a backend as a service (BaaS) such as Parse .*
How can we adapt the UI so that it supports multiple screen dimensions?	*Due to the need of covering a big percent of the mobile market, this is an essential problem.*	*Research on how to develop and design the UI, so that as many as possible screen sizes will be adequate for having a good user experience.*
What open source libraries could we use to speed up the development?	*This should be considered since there are standard development problems, which the development community has already resolved, so there is no need to reinvent the wheel.*	*Research on most popular open source libraries, according to the platform's needs.*
What data should we cache?	*Storing all data would make the app slower, and take space on the users phone*	*Research and decide what is most essential for the app to run smoothly to create a good user experience.*

Time schedule

Phase	Target tasks	Start date	End date
Project idea	Define the project idea.	15.10.2014	10.11.2014
Project description	Describe the idea through a project description.	10.11.2014	18.11.2014
Final Project description	Improve the project description, after receiving feedback.	18.11.2014	1.12.2014
Project planning	Overall work distribution plan, definition of requirements and functionality span of the project.	1.12.2014	18.12.2014
Backlog definition	Write the backlog and assign priorities to the tasks.	18.12.2014	15.01.2015

Sprints Definition	Define the separate sprints and plan the distribution of the workload between developers.		15.02.2015
Project development	Development start of the project	1.02.2015	1.06.2015

References and expected sources

http://developer.android.com/about/dashboards/index.html (versions of Android)

https://medium.com/ios-apprentice/9-time-saving-ios-7-libraries-43e943e6627 (iOS libraries)

https://www.infinum.co/the-capsized-eight/articles/top-5-android-libraries-every-android-developer-should-know-about (popular Android open source libraries)

http://developer.android.com/training/multiscreen/screensizes.html (supporting different screen sizes in Android)

https://gigaom.com/2014/09/13/how-will-the-new-iphone-screen-sizes-affect-ios-developers/ (supporting different screen sizes in iOS)

https://parse.com/docs (documentation for Parse - backend as a service)