|  |  |
| --- | --- |
| **Nazwa projektu:** | HeartBeat analyzer |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Autorzy specyfikacji:** | | | | |
|  | | | | |
|  | *imię i nazwisko* | *nr indeksu* | *grupa* | *adres email* |
|  | Wiśniewski Dawid | Inf94387 | ISWD | Wisniewski.dawid@gmail.com |
|  |  |  |  |  |

Opis projektu

Jako, że choroby serca są dość popularnymi dolegliwościami, moglibyśmy wykorzystać dostępne narzędzia do badania zdrowia. <http://abrc.snu.ac.kr/sjkwon/EMBC2011.pdf> pod zadanym adresem widnieje publikacja potwierdzająca wystarczalność akcelerometru w iPhone do analizy bicia serca. Program zbierałby informacje o tym jak działa jego serce, po czym użytkownik oznaczałby choroby, które zostały u niego zdiagnozowane. Takie paczki informacji ( zapis akcelerometru oraz informacje o chorobach ) byłyby wysyłane na zdalny serwer, który próbowałby użyć narzędzi analizy danych do „nauczenia się” pewnych chorób. Być może udałoby się wykrywać pewne odchylenia od normy i poprawnie je klasyfikować informując użytkownika o potencjalnych problemach. Projekt obejmowałby zaimplementowanie modułu zbierającego dane z akcelerometru oraz zapisującego informacje o stwierdzonych chorobach, stworzenie webservice'u na który można wysyłać tak zebrane dane oraz okresowe budowanie klasyfikatora starającego się decydować czy nowoprzybyła paczka informacji pasuje do jakiegoś wzorca choroby.

Funkcjonalność projektu

Spis wymaganych funkcjonalności z podziałem na obowiązkowe (min. 10) i opcjonalne (min. 5).

|  |  |  |  |
| --- | --- | --- | --- |
| *Lp* | *Opis funkcjonalności* | *Typ* | *Status* |
| 1 | Zbieranie informacji z akcelerometru | obowiązkowa |  |
| 2 | Implementacja wysyłania danych na serwer | obowiązkowa |  |
| 3 | Trening modeli na podstawie zebranych danych | obowiązkowa |  |
| 4 | Podejmowanie decyzji na podstawie zebranych danych | obowiązkowa |  |
| 5 | Przedstawienie informacji o najpopularniejszych chorobach | obowiązkowa |  |
| 6 | Dostarczenie intuicyjengo interfejsu użytkownika | obowiązkowa |  |
| 7 | Wykrywanie braku połączenia sieciowego | obowiązkowa |  |
| 8 | Informowanie o nauczeniu się nowych chorób | obowiązkowa |  |
| 9 | Powtarzanie badania jeśli uzyskany rezultat słabej jakości | obowiązkowa |  |
| 10 | Możliwość wybrania choroby z predefiniowanej listy | obowiązkowa |  |
| 11 | Rozszerzanie predefiniowanej listy chorób, nowe choroby byłyby automatycznie wysyłane do innych aplikacji | opcjonalna |  |
| 12 | Utworzenie hierarchii klasyfikatorów do lepszego klasyfikowania podobnych chorób | opcjonalna |  |
| 13 | Możliwość wysłania próbki danych do lekarza | opcjonalna |  |
| 14 | Umożliwienie kontaktu z osobami z tą samą chorobą | opcjonalna |  |
| 15 | Informowanie o popularnych metodach leczenia wykrytych chorób | opcjonalna |  |

Koncepcja realizacji

Aplikacja mobilna powinna działać na Smartphone'ach wyposażonych w dość czuły akcelerometr. Powinna ona dostarczać prosty interfejs użytkownika, który pozwoli zapisać próbkę danych o określonej długości i wybrać choroby ( jeśli jakieś zdiagnozowano ). Aplikacja w miarę możliwości powinna być wieloplatformowa ( PhoneGap ). Aplikacja mobilna powinna dostarczać bazę predefiniowanych chorób związanych z biciem serca oraz krótkie wyjaśnienie czym dana choroba się charakteryzuje.

Z drugiej strony powinniśmy zaimplementować webservice przyjmujący dane od użytkowników. Taki serwis pobierałby paczki danych i co jakiś czas ( określone ilość zgłoszeń ? ) próbowałby stworzyć klasyfikator, który nauczyłby się oznaczonych chorób. Nauczenie się nowej choroby ( klasyfikacja z trafnością > pewien próg ) powodowałaby rozesłanie informacji do pacjentów, którzy nie są świadomi, że mogą ją posiadać. Rozesłanie informacji powinno wystąpić po uruchomieniu aplikacji przez użytkownika, powinna być zarejestrowana ostatnia data urochomienia aplikacji, po czym jeśli od tamtego momentu doszła jakaś nowa w bazie – powinniśmy sprawdzić uprzednio wysłane dane pod kątem występowania tych nowych chorób.

Sugerowane narzędzia:

Platformy Android, iOS

PhoneGAP – tworzenie mobilnych crossplatformowych aplikacji

node.js, JavaScript, Bootstrap – aplikacja webowa

Materiały źródłowe

* http://abrc.snu.ac.kr/sjkwon/EMBC2011.pdf
* Książki dot. uczenia maszynowego
* Dokumentacja Androida/iOS
* Dokumentacja PhoneGAP