



Inżynieria Oprogramowania & Bazy Danych - Projekt

3. Pierwszy sprint

Opracował: Maciej Penar

Spis treści

1. Fabric – dla mobilek – 1 osoba	3
2. Info – dla sklepu z notatkami – 0 osób	3
3. Sprint numer (chyba) 1	3
4. Kontrola Sprintu	3
4. Moje spostrzeżenia	4
L6 - Terminarz	4
L6 - Sklep	4
L5 - Paliwo	4
L5 – Projekt Grzybek	4
Jest jeszcze ta czwarta opcja:	7

1. Fabric – dla mobilek – 1 osoba

Podpiąć się pod znane i lubiane narzędzie do wyłapywania bugów oraz badania użyteczności aplikacji mobilnych – Fabric.

Założyć konto: <https://fabric.io>

Podpiąć swoje adresy mailowe oraz mój adres: mpenar@kia.prz.edu.pl jako betatesterów.

*Na ogół bardzo kiepskim pomysłem jest commitowanie na publiczny repozytorium **api key** – just sayin’.*

2. Info – dla sklepu z notatkami – 0 osób

Tylko info tu zaparkuję:

W pewnym bliżej nieokreślonym momencie sprawdzę jak wygląda aplikacja którą macie na adresie:

<https://projekcik-prz.azurewebsites.net/>

Jeśli adres ulegnie zmianie to pisać lub zostawić notkę na repo (czyt. Sekcję „Moje spostrzeżenia”)

3. Sprint numer (chyba) 1

Pewnie w tym lub kolejnym tygodniu PM’i powiedzą: działać.

Na tym etapie już powinniście móc coś zakodzić – jeśli harmonogramy + makiety + specka pozwalają. Jak nie - to musicie być bardziej napastliwi wobec PM’ów i wydębić od nich materiały.

Oto tabelka dni kiedy popatrzę na Wasze aplikacje mobilne (via Fabric) / strony webowe:

Projekt	Termin
L6 – Terminarz	16.04.2019
L6 – Sklep	15.04.2019
L5 – Paliwo	15.04.2019
L5 – Projekt Grzybek	17.04.2019

4. Kontrola Sprintu

W ramach podsumowania pracy przed przerwą świąteczną chciałbym zobaczyć jak harmonogramy przystają do stopnia zaawansowania. Czyli klasyczne Expectations vs Reality ([link](#)).

Proszę o podesłanie na maila:

- Odpowiednich dostępuów (Fabric’owcy) do aplikacji
- Aktualnej wersji harmonogramu

4. Moje spostrzeżenia

L6 - TERMINARZ

Na razie śpię spokojnie. Dopytać PM'ów odnośnie harmonogramu – albo nie dostałem/albo gdzieś mi uciekł/albo już o nim zapomniałem.

Jeśli dobrze pamiętam to nie ma kont użytkowników – na razie można zostawić w tej postaci. Byłyby po nic – skoro i tak jest to terminarz osobisty. Lepiej jest backupować poprzez Google Drive czy Dropbox.

Nie pamiętam na czym stanęło.

L6 - SKLEP

Odniosłem wrażenie na spotkaniu że PM'i chcą się Was pozbyć i nie chce im się prowadzić tego projektu... i siedzieć na spotkaniu. Przynajmniej harmonogram się nadaje (nawet) do pracy. Nie wiem czy został ustalony format danych jakie przechowujecie tj. materiały jako obrazy/pdf/docx i w jaki sposób jest autoryzowany dostęp do tych zasobów (skoro są płatne).

Poza tym jeśli chcielibyście się przenieść na localhost to polecam tunelowanie via Ngrok:

<https://ngrok.com/>

Z IIS'em trzeba się trochę pomęczyć: <https://www.twilio.com/docs/usage/tutorials/how-use-ngrok-windows-and-visual-studio-test-webhooks>

L5 - PALIWO

Bardzo mnie cieszy że PM'i i zespół developerów stwierdził że sterowanie głosem to feature aplikacji. IMO: w pierwszej kolejności oprogramujcie „palcówkowe” (będę się potem z tego tłumaczył) działanie programu. Jakbym ja to pisał to bym nad każdą aktywnością posadził jakąś **klasę abstrakcyjną** która potem posłużyłaby za przechwytywanie głosu.

Dorzucam jeszcze jedno info na które PM'i nie znaleźli odpowiedzi:

- <https://stackoverflow.com/questions/45612505/find-nearest-gas-station-using-google-maps-v2-in-android-and-want-to-display-the>
- <https://developers.google.com/maps/documentation/javascript/get-api-key>

Poza tym tu nie mam komentarzy – przycisnąć PM'ów żeby zdali harmonogram i zacząć przepisywać specke na apkę.

L5 – PROJEKT GRZYBEK

Spotkanie poszło po bandzie – odniosłem wrażenie, że PM'ki chcą by aplikacja robiła:

- Albo za dużo – czyli przepisywała gro funkcjonalności sieci społecznych. Ewidentny problem z nieprzemyśleniem skąd czerpać bazę użytkowników i listę znajomych. Rozważyliśmy wariant poza-Facebookowy – i tu w specyfikacji nie ma informacji jak wygląda „socjalna” część aplikacji tj:
 - Jak dodawać znajomych

- Jak usuwać znajomych
- Jak zapraszać znajomych

Tak naprawdę modelowanie sieci społecznych to jest trudniejszy problem niż się wydaje, tj. boję się że możecie napisać rozwiązanie na małą skalę (to chyba że zaprzęgnięcie Neo4J via Docker/Vagrant).

- Albo za mało i niebezpiecznie – czyli niejasny sposób autentykacji użytkownika (autoryzację przemilczę) przez Facebook’ co tak naprawdę nie zwalnia z obowiązku z przechowywania informacji na temat konta... gdzieś... pewnie backend, ale nie ma uzasadnienia dlaczego bo **wydarzenia nie mają zdefiniowanych ograniczeń**

Na domiar złego PM’ki:

- akceptowały wszystko co wlezie, aż do momentu kiedy padła opcja zrobienia tego przez API Twittera
- próbowały mnie przekonać że opcja zakończenia projektu w 8 h jest zła
- nie wyjaśniły mi w czym ten projekt jest lepszy od grupy na facebooku reprezentującej zdarzenia na Grzybku



IM(H)O musicie panowie nacisnąć na PM’ki mocniej, ustawić je do pionu żeby sprecyzowały co aplikacja ma robić i gdzie jest **wartość** tej aplikacji.

Rozważmy opcje jakie mamy:

1. na ten moment - robić wg. niekompletnej specyfikacji sieć społeczną (znajomych) zapisywać po Waszej stronie za pomocą widoków które nie wiemy jak mają wyglądać
2. robić nadbudówkę na API Facebooka (o ile się da) tak żeby userzy logowali się via Facebooka (lub żeby Facebook pozwolił na wykonywanie żądań poprzez konto) i mieli możliwość tworzenia predefiniowanych Grzybko-zdarzeń tj. o stałej lokalizacji

3. wykorzystać Twittera który wiadomo że ma API do wysyłania / odbierania wiadomości i aplikacja korzystałaby z dedykowanego konta jako magazynu danych



JEST JESZCZE TA CZWARTA OPCJA:

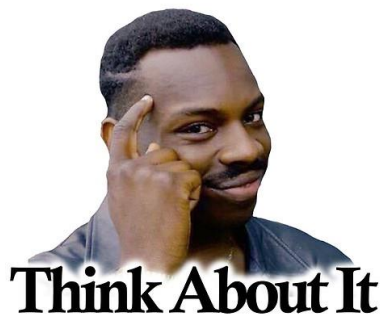
Pomyśleć jak pominąć backend tej aplikacji..... Nie wspominałem o tym, bo..... chciałem przedłużyć spotkania do 3h... ale już tłumaczę:

- imo same konta w tej aplikacji nie mają żadnego znaczenia – nie mam pojęcia po co są, chyba tylko żeby **można było odpisać twórcy zdarzenia** że ... że się będzie czy coś.
- Dodatkowo wiemy że zdarzenia nie mają ograniczeń – **więc można na nie spojrzeć jak na blok tekstu który nie trzeba przetwarzać.....**

Pytanie... jak pominąć backend...

Być może wystarczy napisać aplikację która **rozgłasza pushe** do wszystkich użytkowników na zasadzie „Best Effort” (jak się uda to ok, jak nie to nie). To można efektywnie robić poprzez nasłuchiwanie kanałów w Firebase. Ten kto odebrał pusha - zapisałby go lokalnie i wyświetlił. Do pusha możnaby dopisać adres nadawcy i w ten sposób można by realizować odpowiedź że ... się będzie....

I backend nie jest potrzebny.... Bo po co...



I teraz **pro-tip** Panowie: wbrew intuicji może się wydawać że tę decyzję trzeba podjąć szybko. To poważna decyzja i pomimo tego że jesteście powoli pod presją czasu to weźcie dzień lub dwa na przemyślenie i rozważenie za/przeciw.

Dajcie znać na której opcji stajecie.