



Sprawozdanie końcowe

Projekt grzybek

Opracował: Bartłomiej Nawój L05 156314

Spis treści

Praca w zespole	3
Za co odpowiadałem w projekcie.....	3
jaki był mój okres aktywności w projekcie	3
Co wykonałem w projekcie	3
Zespół, a praca	5
Jak przebiegała praca w zespole	5
za co w projekcie odpowiadali inni	6
za co w projekcie odpowiadali Project Manager'zy	6
za co w projekcie powinni odpowiadać Project Manager'zy	6
za co w projekcie odpowiadali interesariusze.....	6
za co w projekcie powinni odpowiadać interesariusze	6

Praca w zespole

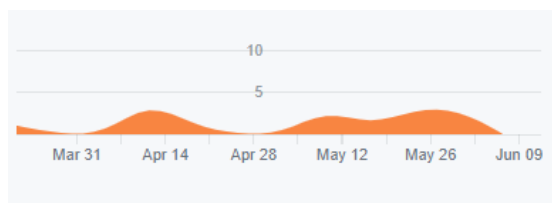
ZA CO ODPOWIADAŁEM W PROJEKCIE

Oprócz zaangażowania w tworzenie aplikacji, pełniłem również rolę zastępcy lidera projektu. Zastępowałem go, gdy z przyczyn osobistych nie mógł być dostępny. Pomagałem również na przestrzeni komunikacyjnej z menadżerami projektu, biorąc czynny udział we wszystkich spotkaniach dotyczących projektu, czy też rozmawiając z PM'ami zdalnie. Co do wytwarzania oprogramowania, byłem w drużynie odpowiedzialnej za back-end aplikacji.

JAKI BYŁ MÓJ OKRES AKTYWNOŚCI W PROJEKCIE

Moja aktywność w projekcie zaczęła się jeszcze przed przypisaniem pm'ów do projektu. W etapie przedprodukcyjnym brałem udział w szukaniu odpowiedniej technologii, tworząc Proof-of-concept dla technologii Django. Końcowo, po rozmowach z zespołem, wybrano tworzenie aplikacji dla systemu Android w języku Kotlin.

Wraz z przypisaniem studentkom zarządzania roli menadżerów projektu, zaczęła się moja aktywność jako zastępcy lidera projektu. Moją aktywność w procesie tworzenia aplikacji przedstawia wykres commitów do repozytorium projektu:



Jak widać, moje prace nad projektem zaczęły się na przełomie marca i kwietnia, a zakończyły wraz z prezentacją projektu w czerwcu.

CO WYKONAŁEM W PROJEKCIE

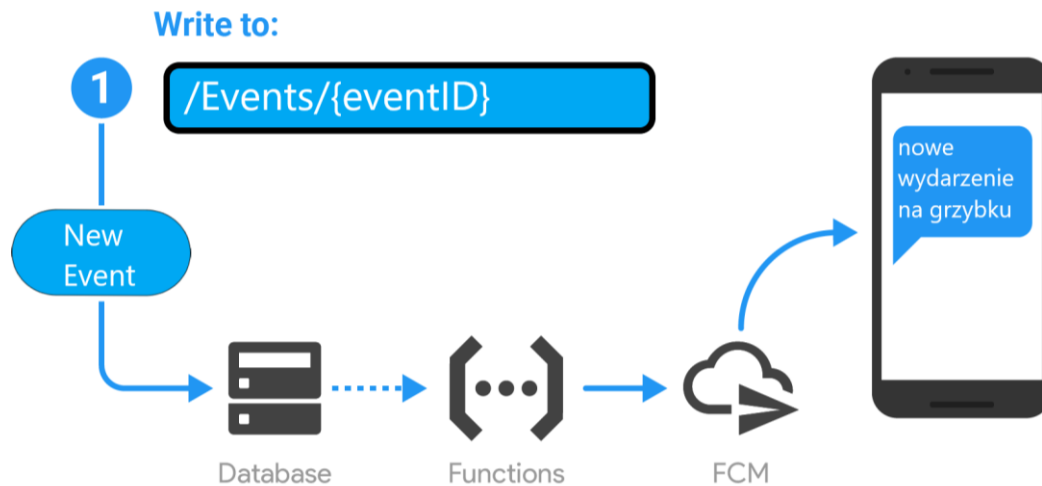
Na samym początku prac nad projektem zająłem się połączeniem projektu z platformą Firebase. Było to kluczowe, aby umożliwić dalszy rozwój aplikacji. Firebase to darmowy BaaS (Backend as Service) od Google, który zawiera wiele funkcjonalności pomagających programistom w tworzeniu i zarządzaniu aplikacją. Zająłem się integracją z projektem „Grzybek” następujących narzędzi wchodzących w skład panelu Firebase:

- Crashlytics – rozpoznawanie przyczyn awarii aplikacji, poprzez raporty „na żywo” przychodzące na maile developerów
- Cloud Firestore – baza danych NoSQL
- Cloud Messaging – rozwiązanie umożliwiające wysyłanie powiadomień do użytkowników.

Aby możliwe było rozprowadzanie wersji beta naszej aplikacji do testerów, po uzyskaniu porady od prowadzącego, podłączyłem projekt do narzędzia Fabric – ‘Beta by Crashlytics’. Rozsyłaniem najnowszych wersji aplikacji zajmowałem się osobiście.

Zaimplementowałem logikę wysyłania i odbierania powiadomień o nowo utworzonych wydarzeniach. Aby użytkownik mógł otrzymywać powiadomienia, tuż po zalogowaniu jest zapisywany do subskrybentów tematu „notifications”. Założenia projektu przewidywały wysyłanie powiadomień do każdego użytkownika o każdym wydarzeniu, a takie rozwiązanie właśnie to umożliwia.

Firebase Cloud Functions umożliwia pisanie funkcji w javascript i hostowanie ich w chmurze Google Cloud. Funkcje te mogą być później wyzwalane przez różne 'eventy', w tym przypadku jest to dodanie nowego wydarzenia do bazy danych. Napisano funkcję w node.js nasłuchującą, czy zostało dodane nowe wydarzenie do bazy. Cały proces mechanizmu działania powiadomień prezentuje poniższy rysunek:



Wspomniana funkcja wygląda następująco:

```

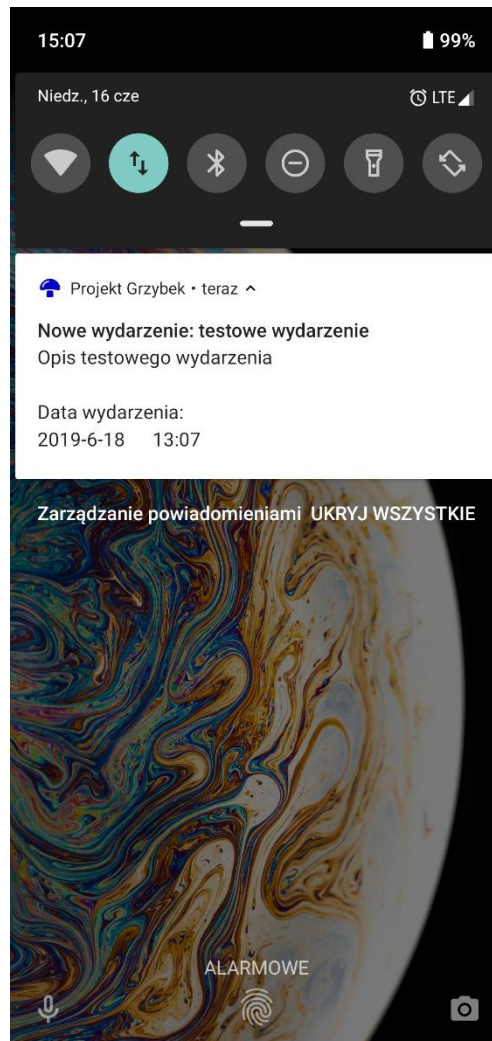
exports.notifyNewEvent = functions.firestore
  .document('/Events/{eventID}')
  .onCreate((snap, context) => {
    const newEvent = snap.data();
    const name = newEvent['Name'];
    const notificationBody = newEvent['Desc'];
    const owner = newEvent['Owner'];
    const eventDate = newEvent['DateStart'].toDate();
    var time = eventDate.toLocaleTimeString().substr(0,5);

    const payload = {
      notification: {
        title: "Nowe wydarzenie: " + name,
        body: notificationBody + "\n\n" + "Data wydarzenia:\n" +
          eventDate.toLocaleDateString() + " " + time,
      },
      topic: "notifications"
    };
    admin.messaging().send(payload)
      .then((response) => {
        // Response is a message ID string.
        console.log('Successfully sent message:', response);
      })
      .catch((error) => {
        console.log('Error sending message:', error);
      });
  })
  });
  
```

Gdy do tabeli 'Events' trafia nowe wydarzenie, tworzona jest migawka (eng. Snapshot) danych o wydarzeniu. Utworzenie zwykłych zmiennych tekstowych odpowiadających polom tekstowym w bazie jest proste. Przypisanie tekstu do wartości **const** jest wystarczające, aby można było ten tekst wysłać w powiadomieniu. Drobny problem sprawiło wyświetlanie daty wydarzenia. Okazało się, że w naszej bazie danych data wydarzenia jest przetrzymywana jako **TimeStamp**. Skonwertowanie tego typu do lokalnego odpowiednika wyświetlania czasu, a

następnie tej wartości do typu **String** okazało się wystarczające by takie pole przesłać w powiadomieniu. W aplikacji „Projekt grzybek” skorzystano z natywnych powiadomień systemu Android, czyli powiadomień bez dodatkowych danych (eng. Payload). Zdecydowano się na takie rozwiązanie, aby uprościć implementację tej funkcjonalności, ale również dlatego, że założenia projektowe nie mówiły inaczej. Dalsze linijki to obsługa błędów. Jeśli taki nastąpi, w konsoli Firebase można sprawdzić powód nie wysłania powiadomienia.

W momencie, w którym utworzone zostało nowe wydarzenie na grzybku, każdy użytkownik aplikacji otrzymuje powiadomienie wyglądające w następujący sposób:



Zespół, a praca

JAK PRZEBIEGAŁA PRACA W ZESPOLE

Praca w zespole przebiegała bez większych problemów, można by wręcz powiedzieć wzorowo. Dzięki zakładce 'projects' na repozytorium projektu podzielono aplikację na mniejsze problemy i przyporządkowano każdemu kilka zadań do wykonania. Każdy wykonywał przydzielone sobie zadania, a w razie problemów reszta grupy służyła pomocą. Nikt nie odwlekał swoich obowiązków na później, dzięki czemu aplikacja została utworzona na czas, a proces jej tworzenia był – w miarę możliwości – równomiernie rozłożony w czasie.

ZA CO W PROJEKCIE ODPOWIADALI INNI

Całą grupę podzielono na mniejsze zespoły odpowiedzialne za:

- Front-end
 - Mateusz Nehrebecki
 - Rafał Piszko
- Back-end
 - Mateusz Moch
 - Marcin Pazowski
 - Artur Przysaś
- Przygotowanie plików graficznych
 - Maciej Miśkowiec
- Lider zespołu i komunikacja z PM'ami
 - Artur Przysaś

ZA CO W PROJEKCIE ODPOWIADALI PROJECT MANAGER'ZY

- Przedstawili ogólny pomysł na aplikację (który wymagał poprawek)
- Sprawdzali etapowo wykonaną przez nas pracę
- Sprawdzali działanie coraz to nowszych wersji aplikacji
- Wymagali dotrzymywania terminów
- Oczekiwali raportów z pracy

ZA CO W PROJEKCIE POWINNI ODPOWIADAĆ PROJECT MANAGER'ZY

Moim zdaniem, project manager'zy powinni przygotować gruntowny plan stworzenia aplikacji, po uprzednim upewnieniu się, że ich pomysł ma sens i czy jest lepszy od innych – gotowych już - rozwiązań. Plan powinien zawierać potrzebne makiety aplikacji, schematy działania, projekt wyglądu poszczególnych widoków. Niestety, wkład zespołu z kierunku zarządzania ograniczał się do rozliczania naszej pracy. Początkowy pomysł na aplikację wymagał wielu przemyśleń i usprawnień, którymi zdawało by się przejmowaliśmy się tylko my.

ZA CO W PROJEKCIE ODPOWIADALI INTERESARIUSZE

Prowadzący przedmiot po stronie naszego kierunku udzielił wielu informacji i porad niezbędnych do wykonania niektórych z funkcjonalności. Bez jego pomocy, zaimplementowanie autoryzacji przez CAS byłoby niemożliwe. Przez cały proces tworzenia aplikacji pomagał w jej testowaniu i usprawnianiu.

Nie posiadam informacji o tym, za co odpowiadał prowadzący przedmiot po stronie kierunku zarządzania.

ZA CO W PROJEKCIE POWINNI ODPOWIADAĆ INTERESARIUSZE

Przyglądając się na projekt z boku, można dojść do wniosku, że mógłby się odbywać bez uczestnictwa studentów zarządzania. Ciężko stwierdzić, za co powinien odpowiadać interesariusz z wydziału zarządzania. Moim zdaniem, interesariusz z wydziału elektroniki i zarządzania całkowicie wywiązał się ze swoich obowiązków, a nawet pomagał z nawiązką (poprawa pomysłu na aplikację, wygląd aplikacji).