



# Sprawozdanie końcowe

Grzybek

Opracował(a): L05 158841

## Spis treści

Praca w zespole .....	3
Za co odpowiadałem w projekcie.....	3
jaki był mój okres aktywności w projekcie .....	3
Co wykonałem w projekcie .....	4
Zespół, a praca .....	10
Jak przebiegała praca w zespole .....	10
za co w projekcie odpowiadali inni .....	10
za co w projekcie odpowiadali Project Manager'zy .....	10
za co w projekcie powinni odpowiadać Project Manager'zy .....	10
za co w projekcie odpowiadali interesariusze.....	10
za co w projekcie powinni odpowiadać interesariusze .....	11

## Praca w zespole

### ZA CO ODPOWIADAŁEM W PROJEKCIE

W ramach projektu byłem odpowiedzialny za

- Ogólne prace projektowe
- Utworzenie zarysu aplikacji
- Współudział w projektowaniu widoków aplikacji
- Przygotowanie wspólnego repozytorium
- Odpowiednie przydzielanie zadań
- Implementacja logowania przez system CAS

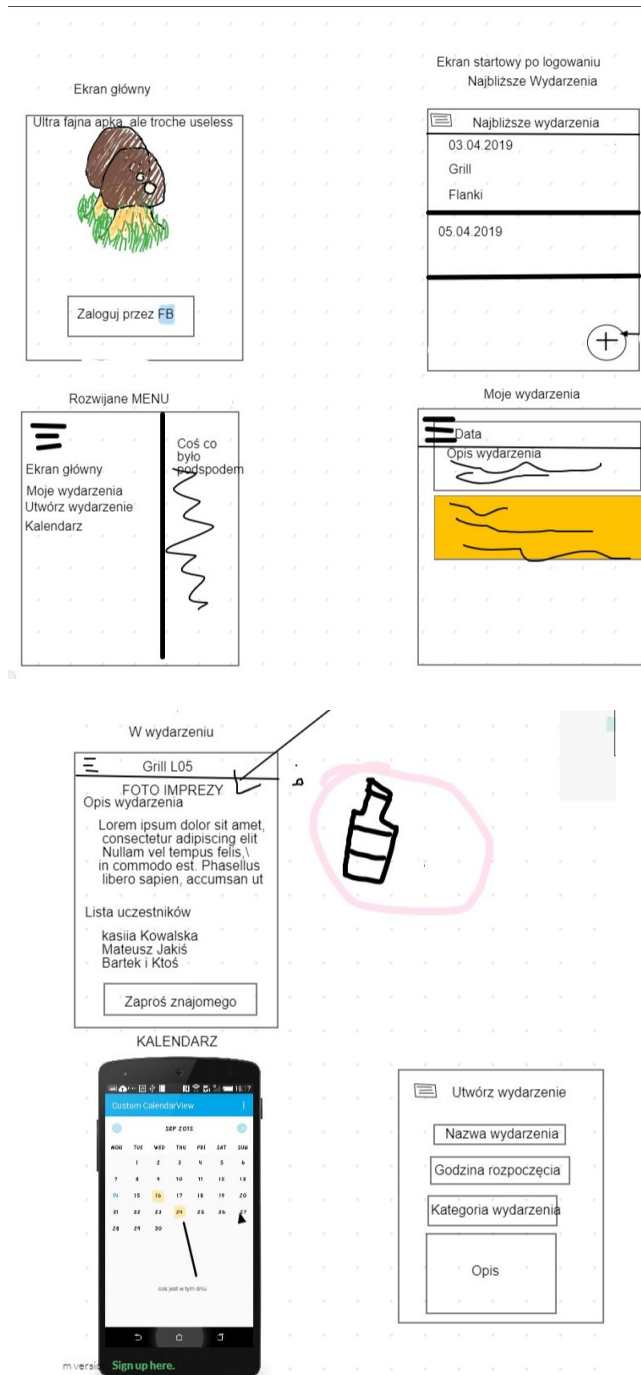
### JAKI BYŁ MÓJ OKRES AKTYWNOŚCI W PROJEKCIE

Moja aktywność w projekcie była stała, od początku do czasu ostatnich poprawek developerskich.

## CO WYKONAŁEM W PROJEKCIE

### WSPÓLNA PRACA NAD MAKIETAMI WIDOKÓW:

Po dostaniu podstawowych wytycznych od PM'ów prowadziłem rozmowy z członkami zespołu w celu ustalenia podstawowego wyglądu aplikacji, a także użycia odpowiednich narzędzi. Dodatkowo zorganizowaliśmy spotkanie online w czasie, którego wspólnie ustaliliśmy podstawowy wygląd aplikacji.



Rysunki wyżej pokazują pierwsze etapy pracy na projektem wyglądu aplikacji. Od lewej od góry: ekran logowania, ekran startowy, rozwijane menu, ekran wydarzeń użytkownika, ekran szczegółów wydarzenia, ekran kalendarza, ekran tworzenia wydarzenia.

Od Project Managerów, nie dostaliśmy jakichkolwiek makiet. Po przedstawieniu naszych pierwszych zarysów zgodzili się na zaproponowany wygląd.

Finalnie aplikacja wygląda tak jak na przedstawionych zrzutach ekranu:



W kolejności od lewej od góry: ekran logowania, ekran startowy, lista wydarzeń użytkownika, ekran tworzenia wydarzenia, ekran kalendarza, ekran szczegółów danego wydarzenia.

## Utworzenie zarysu aplikacji

Prowadziłem poszukiwanie odpowiednich narzędzi do wykonania projektu. W tym celu wystosowałem zapytanie do doświadczonych osób na stronie pełnej profesjonalistów (<https://www.wykop.pl/mikroblog/>):

The screenshot shows a discussion thread on the Wykop.pl forum. The thread is titled "Czołem Mireczki <3 Nawet nie wiecie jak się cieszę, że Was mam." and is posted by user "Caishen" 2 months ago. The thread contains several replies from other users, including "plan\_9", "PieceOfShit", "WydajnaJednostkaIndywidualna", "tptak", and "Caishen" again. The replies discuss various options for hosting the backend, such as Heroku, Firebase, AWS, and Docker. The thread also includes hashtags like #androiddev, #naukaprogramowania, and #sysadmin. The thread is marked as "Zakończony" (Closed) and has a "Zakończony" (Closed) status.

**Caishen** 2 mies. temu +2

Czołem Mireczki <3 Nawet nie wiecie jak się cieszę, że Was mam.

Na studia mamy przygotować aplikację na androida + backend (baza danych i całe zaplecze projektu). Projekt robimy w parę osób i nie do końca wiem co powinienem użyć żeby postawić backend. Gdybym to robił sam to prawdopodobnie byłoby to wirtualka, na której stałoby wszystko. Jednak to rozwiązanie odpada bo byłoby dostępne tylko lokalnie.

Są jakieś darmowe rozwiązania tak aby można było testować backend z frontendem, tak aby każdy miał tą samą wersję backendu?

Myslałem nad kupieniem jakiegoś najtańszego vps'a i tam postawić zaplecze. Macie lepszą propozycję?

Planujemy pisać front i back end w Javie.

#androiddev #naukaprogramowania #sysadmin

+1: spierdelolada, tomuho

**plan\_9** 2 mies. temu +4

@Caishen: Heroku

+1: spierdelolada, Caishen +2 innych

**PieceOfShit** 2 mies. temu +9

Jeżeli nacisk jest kładziony na samą apkę, to backend możecie łatwo i za darmo zrobić w Firebase

+1: puszczalka, spierdelolada +7 innych

**WydajnaJednostkaIndywidualna** 2 mies. temu +3

@Caishen: <https://education.github.com/pack>

+1: Whipie, Caishen +1 inny

**Caishen** 2 mies. temu 0

@WydajnaJednostkaIndywidualna: @PieceOfShit: @plan\_9: dziękuję :) Zabieram się do czytania i wybiorę coś spoko

**plan\_9** 2 mies. temu +1

@Caishen: AWS w ramach free tier można mieć przez rok za darmo ale do takiego studenckiego projektu to będzie raczej overkill, na Heroku postawisz to łatwiej i szybciej.

+1: Caishen

**tptak** 2 mies. temu +1

@Caishen: a czemu by nie na przykład tworzyć dockerowy obraz i z niego korzystać? Chyba że poprzez tę samą wersję backendu chcesz powiedzieć tę samą instancję, to wtedy rozumiem.

+1: Caishen

**Caishen** 2 mies. temu 0

@WydajnaJednostkaIndywidualna: @PieceOfShit: @plan\_9: Z tego co widzę Heroku prezentuje się super ale jest jeden szkopuł. Nasz grupa liczy 7 osób. Dwie za dużo żeby korzystać z darmowej grupy do kooperacji. Na szczęście @WydajnaJednostkaIndywidualna podrzucił paczkę dla studentów od GitHuba i partnerów. Jest tam rozszerzona wersja Heroku i z tego co rozumiem grupa współpracowników może być większa.

Firebase do backendu może używać tylko NodeJS. I tak musimy się uczyć od podstaw więc najwyżej wyjdzie Java + NodeJS.

AWS raczej za duży kombajn

@tptak: docker odpada bo chciałbym żeby to była jednak jedna instancja.

**PieceOfShit** 2 mies. temu via Wykop Mobilny (Android) 0

@Caishen: jeżeli ten wasz backend to tylko CRUD, to żadnego nodejs nie musisz znać, używając Firebase Realtime Database np.

**Caishen** 2 mies. temu 0

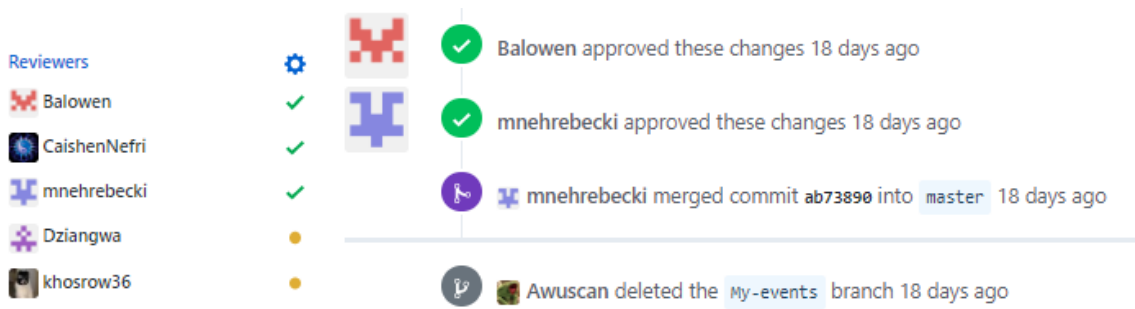
@PieceOfShit: logika musi być po za aplikacją mobilną. W aplikacji user będzie wynajmować miejsce na daną godzinę określonego dnia. Nie bardzo pasuje sprawdzanie po stronie użytkownika czy termin jest wolny.

Po analizie potrzeb naszego projektu, grupowo ustaliliśmy technologie, które zostaną użyte.

## PRZYGOTOWANIE WSPÓLNEGO REPOZYTORIUM NA PLATFORMIE GITHUB

Zaraz po otrzymaniu wytycznych dotyczących projektu należało utworzyć wspólne repozytorium na platformie GitHub.

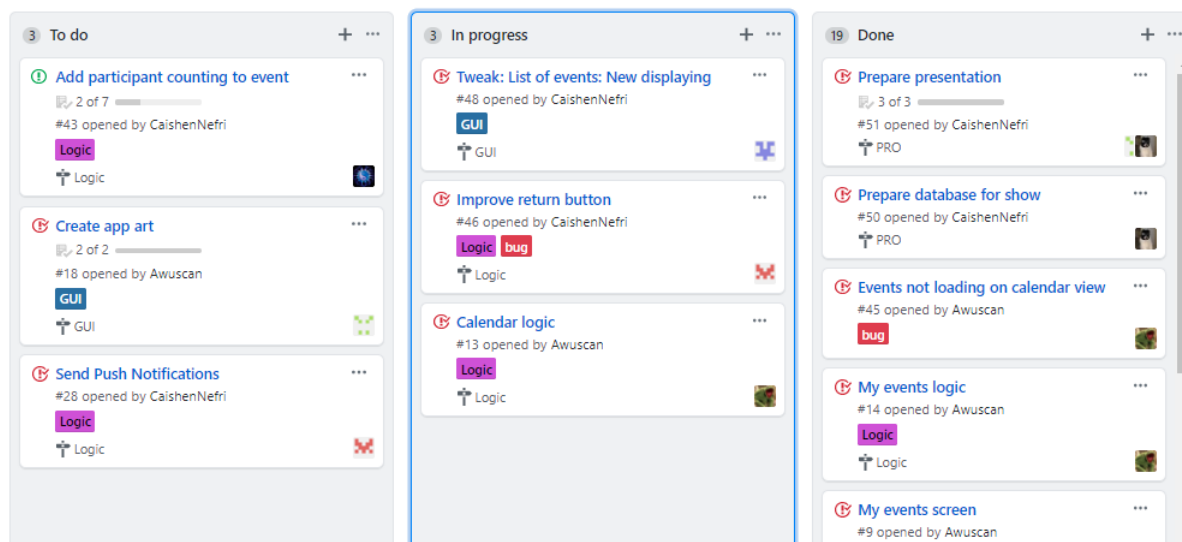
Początkowo dodałem kolaborantów projektu, czyli członków zespołu projektowego, a także ustaliłem ograniczenia. Ustalenie ograniczeń było konieczne, ponieważ każdy z pracujących osób używał GitHub'a pierwszy raz. Ograniczenia zostały wprowadzone w celu zabezpieczenia naszej pracy. Polegały one na tym, że do głównej gałęzi projektu mergować można było dopiero wtedy gdy conajmniej trzy osoby zaakceptują proponowane zmiany.



Dodatkowo oprócz kontroli wersji platforma GitHub umożliwia w łatwy sposób wprowadzenia metodologii Kanban. W łatwy sposób przypisywało się osoby odpowiedzialne za poszczególne zadania. Dodatkowo zadania powiązane były z „Issues” i po mergowaniu były automatycznie uznawane jako wykonane i przenoszone do odpowiedniej kolumny.

Mateusz Moch przygotował pierwszy zestaw zadań do wykonania, a także odpowiednie kategorie. Następnie ja odpowiadałem za przypominanie o nich oraz dodawanie nowych i dbałem o czystość na tablicy.

PRz Project Grzybek   
Updated 1 hour ago



---

## LOGOWANIE PRZEZ CAS PO STRONIE FIREBASE

Po otrzymaniu danych użytkownika od aplikacji moduł autentykuję użytkownika i odsyła token potwierdzający poprawność operacji, który jest niezbędny do zalogowania użytkownika w aplikacji.

„getCustomToken” to główna funkcja przyjmująca base64 w zapytaniu i zwracająca omawiany token.

```
exports.getCustomToken = functions.https.onCall(data => {
  // Read 64base from APP
  var base64FromApp = data.response;
  console.log("Data from APP:", JSON.stringify(data));

  const myJSON = baseToJSON(base64FromApp);
  return createFirebaseAccountCAS(myJSON);
});
```

Funkcja „createFirebaseAccountCAS” uaktualnia lub tworzy encję użytkownika w bazie FireStore.

```
async function createFirebaseAccountCAS(userJSON) {
  var ref = userJSON["cas:serviceResponse"]["cas:authenticationSuccess"];
  // The UID we'll assign to the user
  const uid = ref["cas:attributes"]["cas:uid"]["_text"];
  console.log("User UID: ", uid);

  var info = {
    displayName: ref["cas:user"]["_text"],
    emailVerified: true,
    email: ref["cas:attributes"]["cas:mail"]["_text"],
    disabled: false
  };
  console.log("UserINFO:", info);

  // Create or update the user account.
  const userCreationTask = admin.auth()
    .updateUser(uid, info)
    .catch(error => {
      // If user does not exist we create it.
      if (error.code === 'auth/user-not-found') {
        info.uid = uid;
        return admin.auth().createUser(info);
      }
      throw error;
    });

  // Information about user which can't be put into admin.auth()
  var additionalInfo = {
    usos_id: ref["cas:attributes"]["cas:usos_id"]["_text"],
    departmentnumber: ref["cas:attributes"]["cas:departmentnumber"]["_text"],
    employeetype: ref["cas:attributes"]["cas:employeetype"]["_text"]
  };

  // Adding additional information to database
  const databaseTask = admin.firestore().doc(`/Users/${uid}`).set(additionalInfo);

  // Wait for all async task to complete then generate and return a custom auth token.
  await Promise.all([userCreationTask, databaseTask]);
  // Create a Firebase custom auth token.
  const token = await admin.auth().createCustomToken(uid);
  console.log('Created Custom token for UID "', uid, '" Token:', token);
  return token;
}
```



Funkcja „baseToJSON” konwertuje 64Base do normlanej postaci XML, a następnie do końcowej postaci JSON.

```
function baseToJSON(response) {
  const convert = require('xml-js');

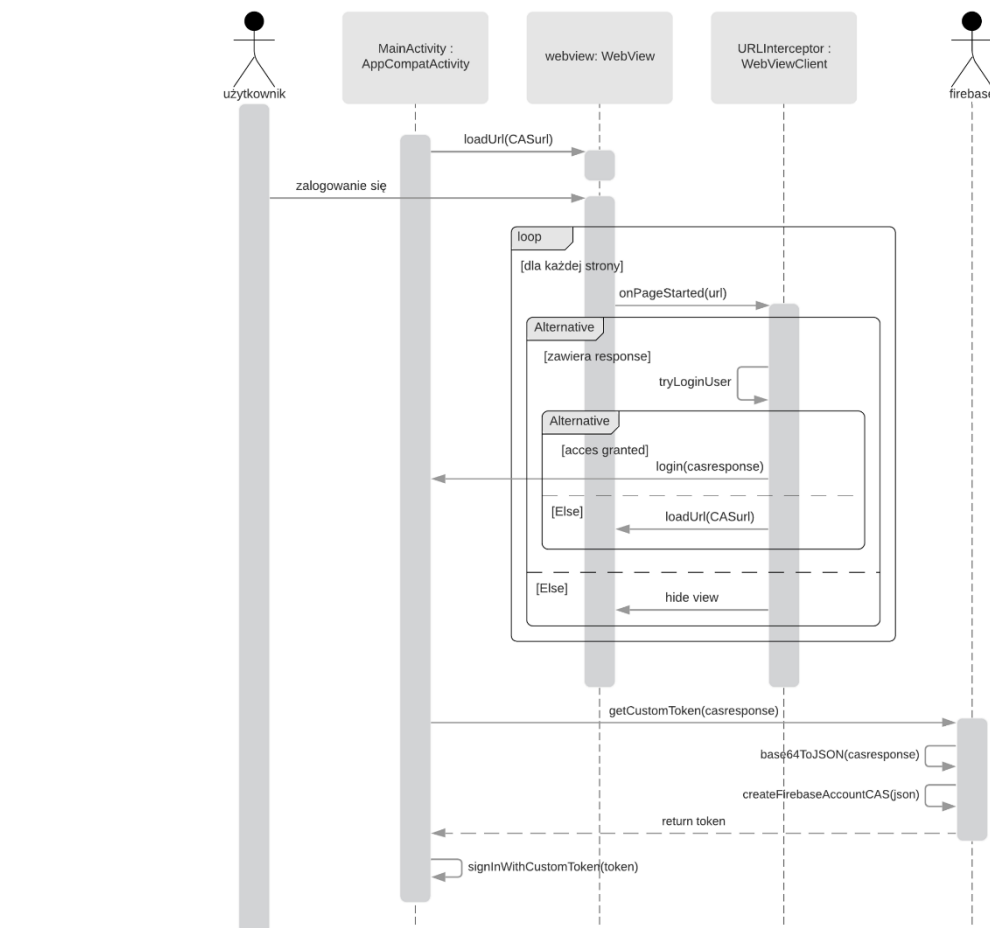
  // Parspe response (base64) to string (XML)
  const decoded = Buffer.from(response, 'base64').toString('ascii');
  console.log("Decoded base64 to string:", decoded);

  // Parse XML to JS Object
  const options = {compact:true, spaces: 4}
  const result = convert.xml2js(decoded, options);

  console.log("Convert XML to JSON: ", JSON.stringify(result, null, 4));

  return result;
}
```

Diagram przygotowany przez Mateusza Mocha zawiera też przepływ w działaniu omawianego modułu:



## Zespół, a praca

### JAK PRZEBIEGAŁA PRACA W ZESPOLE

Praca w zespole przebiegała bardzo dobrze. Każdy z członków zespołu ochotczo i odpowiedzialnie podszedł do wyznaczonego mu zadania.

### ZA CO W PROJEKCIE ODPOWIADALI INNI

W ramach projektu każdy z członków zespołu miał odpowiednio przydzielone zadanie.

- Artur Przysaś
  - Kontakt z PM
  - Back-end
  - Reprezentowanie projektu
- Mateusz Moch
  - Rozpoznanie używanej technologii
  - Przydzielenie podstawowych zadań
  - Back-end
- Rafał Piszko
  - Front-end
  - Back-end
- Bartłomiej Nawój
  - Kontakt z PM
  - Back-end
  - Reprezentowanie projektu
- Mateusz Nehrebecki
  - Front-end
- Maciej Miśkowiec
  - Projekty graficzne

Dodatkowo każdy z członków brał udział w projektowaniu widoków aplikacji oraz analizie w jaki sposób powinna ona działać.

### ZA CO W PROJEKCIE ODPOWIADALI PROJECT MANAGER'ZY

Project manager'zy byli odpowiedzialni za przedstawienie początkowych wymagań jakie ma spełniać aplikacja oraz jakiego powinna być typu (mobilna, strona www, itp.). Dodatkowo służyli oni radą podczas zmian funkcjonalności aplikacji.

### ZA CO W PROJEKCIE POWINNI ODPOWIADAĆ PROJECT MANAGER'ZY

Według zdobytej wiedzy podczas zajęć project manager'zy powinni bardziej odpowiadać za stronę kliencką aplikacji. Odnoszę wrażenie, że projekt widoków, zachowanie aplikacji, ograniczenia powinny być konsultowane i uzgadnianie razem z nimi. Dodatkowo powinni wywierać presję, pilnować terminów, sprawdzać postępy.

### ZA CO W PROJEKCIE ODPOWIADALI INTERESARIUSZE

Podczas projektu nie mieliśmy styczności z interesariuszem ze strony PM'ów.

Pan Magister Maciej Penar w ramach prowadzonych zajęć analizował działanie aplikacji, funkcje jakie spełnia, a także stronę wizualną. Dodatkowo podczas pojawiających się problemów technicznych służył radą nawet w środku nocy.

#### ZA CO W PROJEKCIE POWINNI ODPOWIADAĆ INTERESARIUSZE

Według nabytej wiedzy podczas zajęć interesariusz ze strony informatyków interesował się aż nadto w projekt.

Interesariusz od strony PM'ów, moim zdaniem powinien tak jak PM'owie trzymać pieczę nad projektem.