

## STATUT PROJEKTU

---

### CEL

Celem projektu Concerto jest stworzenie systemu, który zwiększy skuteczność akcji reklamowych wykorzystujących plakaty poprzez:

- możliwość łatwego kupna biletu na wydarzenie od organizatora wydarzenia,
- szybki sposób dodania informacji o wydarzeniu do kalendarza osoby bezpośrednio zainteresowanej,
- wygodne przekazywanie zaproszeń na wydarzenie pomiędzy znajomymi.

### PRODUKTY

System Concerto składa się z trzech komponentów: aplikacji mobilnej, serwera obsługi zapytań oraz API dla organizatorów.

#### Aplikacja mobilna

Aplikacja na urządzenia mobilne kierowana jest do adresatów akcji reklamowych. Będzie służyła do zamawiania biletów na reklamowane wydarzenie, oraz przekazywanie informacji do znajomych. Aplikacja wysyła zrobione przez użytkownika zdjęcia do Serwera obsługi zapytań.

#### Serwer obsługi zapytań

Serwer obsługi zapytań porównuje przesłane przez użytkownika zdjęcie, ze zdjęciami znajdującymi się w bazie danych i zwraca informacje przypisane do plakatu.

#### API dla organizatorów

API udostępnia narzędzia pozwalające organizatorom wydarzeń na przysyłanie plakatów wydarzeń wraz z opisem, datą i miejscem wydarzenia. Określenie API pozwala na łatwe wdrożenie systemu u nowych partnerów biznesowych.

#### Artefakty

##### Dokumentacja techniczna serwera

Dokumentacja techniczna serwera jest to dokument opisujący implementację i sposób działania serwera obsługi zapytań.

##### Dokumentacja techniczna API

Dokumentacja techniczna API opisuje sposób implementacji i korzystania z API dla organizatorów.

##### Dokumentacja procesu testowania

Dokumentacja procesu testowania to dokumenty przedstawiające proces przebiegu testowania.

#### Statut projektu

Statut projektu opisuje najważniejsze zagadnienia występujące w projekcie, opisuje jego cel i założenia.

### WYZNACZNIKI SUKCESU

Projekt musi spełnić szereg wymagań, aby można go było uznać za sukces. Powodzenie projektu zależy zarówno on samej implementacji, jak i nastawienia użytkowników oraz organizatorów.

Oczekujemy że projekt osiągnie następujące rezultaty:

- Dla organizatorów – zwiększenie skuteczności akcji reklamowych.
- Dla firmy Concerto – przyznanie części przychodu ze sprzedaży dokonanych dzięki aplikacji mobilnej.

### **Wsparcie ze strony organizatorów**

Projekt odniesie sukces jedynie w przypadku, gdy znaczny procent organizatorów koncertów będzie korzystać z API, co przeniesie się bezpośrednio na wysokie pokrycie bazy.

### **Responsywność systemu**

Konieczne jest aby system cechował się wysoką responsywnością, z uwagi na to, że użytkownik końcowy korzysta z niego za pomocą aplikacji mobilnej i operacje na niej są wykonywane niejako "w biegu".

### **Intuicyjność**

Podobnie rzecz ma się z intuicyjnością aplikacji, użytkownik smartfona nie może zastanawiać się nad obsługą programu.

### **Bezawaryjność**

Wymagana jest również wysoka bezawaryjność tj. aplikacja mobilna nie może mieć negatywnego wpływu na system operacyjny telefonu, oraz sama nie ulegać awariom, podobnie serwer musi gwarantować ciągłość pracy.

## **KOMUNIKACJA**

Zespół ma możliwość komunikowania się na wielu płaszczyznach, zarówno bezpośrednio jak i za pośrednictwem Internetu.

### **Regularne spotkania**

W określonych odstępach czasu spotyka się cały zespół lub wydzielona grupa celem omówienia postępów lub kolejnych działań.

### **Notatki ze spotkań**

Na każdym oficjalnym spotkaniu sporządzana jest notatka zawierająca poruszone tematy oraz postanowienia.

### **Grupa dyskusyjna**

Do dyspozycji całego zespołu jest oddana również grupa dyskusyjna, gdzie mogą być omawiane bieżące problemy czy wątpliwości, oraz uzgadniane terminy kolejnych spotkań.

### **Wiki**

Każdy moduł projektu posiada stronę wiki na repozytorium, gdzie będzie można znaleźć wszystkie niezbędne informacje dla zespołu.

### **Protokół komunikacyjny dla wiadomości Git**

Kolejną formą komunikacji między członkami jest pole wiadomości przy każdej zmianie w repozytorium. To stąd zespół może czerpać wiedzę o bieżących zmianach.

## **KONTROLA & QA**

Ważnym aspektem realizacji projektu jest kontrola i zapewnienie jakości. Zagwarantowana jest poprzez tworzenie dokumentacji, testowanie, raportowanie błędów ujawnionych podczas testowania i kontrolę postępu. Ze strony każdego programisty oczekuje się dodatkowo, że będzie trzymał się wybranej notacji, korzystał ze wzorców projektowych oraz wersjonował swój kod.

### **Tworzenie dokumentacji**

Tworzenie dokumentacji ma na celu usystematyzowanie i jednoznaczne określenie projektu. Dokumentacja szczegółowo opisuje każdy element systemu.

### **Testowanie**

Kolejnym ważnym elementem jest testowanie. Każdy programista sam testuje kod, dodatkowo testy odbywają się w ramach zespołów.

### **Raportowanie błędów**

Jeśli programista lub inny członek zespołu natrafi w czasie testowania na błąd, zobowiązany jest opisać go w raporcie. Raporty o błędach pozwalają na szybką ich eliminację, oraz ich łatwiejsze odtworzenie.

### **Kontrola postępu**

Kontrola postępu jest przeprowadzana w trakcie trwania projektu. Celem jest uzyskanie pewności, że prace zakończą się w wyznaczonym terminie.

### **Konsekwentna notacja**

Cały zespół korzysta z jednej wybranej notacji, dzięki czemu kod jest czytelny dla wszystkich członków.

### **Wzorce projektowe**

Zespół korzysta również ze wzorców projektowych, co również wpływa pozytywnie na czytelność kodu oraz zmniejsza ryzyko popełnienia błędu.

### **Wersjonowanie**

Wersjonowanie natomiast pomaga w łączeniu zmian dokonanych przez wiele osób. Realizowane jest przez rozproszony system kontroli wersji Git.

## **ZASOBY**

Zasoby dostępne dla zespołu w czasie realizacji projektu można podzielić na następujące grupy:

Materialne, czasowe, ludzkie, dostępne technologie i szkolenia.

### **Zasoby materialne**

#### **Serwer**

Serwer jest to komputer świadczący usługi na rzecz aplikacji mobilnej.

#### **Urządzenia mobilne**

Zespół potrzebuje również platformy testowej, w tym wypadku telefonów komórkowych, tabletów służących do testowania aplikacji mobilnej.

### **Baza danych wydarzeń kulturalnych**

Niezbędna do działania systemu jest oprócz serwera również baza zawierająca informacje o

wydarzeniach kulturalnych takie jak ich opis, data, miejsce oraz plik z plakatem.

### **Zasoby czasowe**

Zespół ma jeden rok na realizację projektu.

### **Zasoby ludzkie**

W zespole każda osoba ma określoną funkcję i możemy wyróżnić programistów, analityka oraz managera projektu. Niezbędne są również osoby z zewnątrz tj. współpracujący organizatorzy.

### **Zespoły programistyczne**

Programiści podzieleni są na zespoły osób pracujących nad implementacją aplikacji mobilnej i programów pracujących po stronie serwera.

### **Analityk**

Analityk jest to osoba odpowiedzialna za merytoryczne opracowanie zagadnień związanych z przetwarzaniem obrazów dla zespołu programistów.

### **Manager projektu**

Wszystkim zarządza manager projektu - osoba koordynująca pracę i odpowiedzialna za prawidłowy przebieg projektu.

### **Współpracujący organizatorzy**

Firmy które nawiązały współpracę biznesową w projekcie, biorące czynny udział w tworzeniu bazy wydarzeń kulturalnych są konieczne, aby zachować aktualność bazy wydarzeń.

### **Technologie**

Zespół musi wykorzystać wiele technologii, aby stworzyć działający system. Należą do nich .NET, Ruby, Ruby on Rails. Dodatkowo wykorzystywany jest LaTeX oraz Git. Dostęp do wszystkich stosowanych technologii jest darmowy, więc nie są ponoszone żadne koszty licencji.

### **Ruby**

Ruby jest to dynamiczny język programowania, posłuży m.in. do stworzenia modułu przetwarzającego obrazy.

### **Ruby on Rails**

Z Ruby związany jest również framework Ruby on Rails, który umożliwi stworzenie API dla organizatorów.

### **.NET**

W przypadku aplikacji mobilnej opiera się na .NET - Platformie programistycznej umożliwiającej tworzenie aplikacji platformę Windows Phone 7 – 8.

### **LaTeX**

Dokumentacja projektu powstaje przy wykorzystaniu LaTeX'a do składu tekstu.

### **Git**

Technologią umożliwiającą wersjonowanie kodu jest Git.

### **Szkolenia**

Z uwagi na to, że zespół nie ma doświadczenia w realizacji podobnych projektów przewidziano warsztaty z przetwarzania obrazów oraz zbudowanie przykładowej bazy danych.

## **Warsztaty z przetwarzania obrazu**

Szkolenie poszerzające wiedzę na temat technologii przetwarzania obrazów.

## **Przykładowa, kontrolowana baza danych**

Dzięki stworzeniu kontrolowanej bazy danych będzie możliwe testowanie powstającego oprogramowania.

## **RYZYO**

Projekt może nie odnieść sukcesu z wielu niezależnych przyczyn.

### **Niska jakość danych od organizatorów**

Istnieje możliwość, że organizatorzy będą wysłać niekompletne dane, bezużyteczne z punktu widzenia użytkownika aplikacji. W tym przypadku należy wysłać pisemne upomnienia, a w razie nagminności tego procederu, proponować szkolenia z obsługi formularza.

### **Za małe możliwości sprzętowe**

Przetwarzanie obrazów jest obciążającym procesem, więc platforma sprzętowa może nie obsłużyć wielu zapytań naraz. Zabezpieczyć można się przed tym opłacając odpowiednio wydajniejszy serwer.

### **Zły dobór technologii**

Windows Phone może zdobyć marginalny procent rynku smartfonów, co przełoży się na małą liczbę użytkowników. Aby wyeliminować takie ryzyko należy przygotować również aplikacje na konkurencyjne systemy tj. iOS i Android.

### **Brak doświadczenia zespołu**

zespół nie miał wcześniej do czynienia z przetwarzaniem obrazów, braki we wiedzy członków należy uzupełnić szkoleniem.

### **Brak zainteresowania produktem końcowym**

Posiadacze smartfonów mogą uznać, że oferowana funkcjonalność jest dla nich nieprzydatna.

**Zbyt wysoki koszt utrzymania** – Koszt utrzymania serwera może przewyższyć przychody z prowizji od sprzedaży biletów.