

Programowanie zespołowe

KARTA PROJEKTU

Nazwa	Aplikacja darowizn
Kierownik	Jakub Regeńczuk
Skład grupy projektowej	Marcin Głaba Katarzyna Lebioda Michał Swat

Specyfikacja wymagań	<p>Implementacja aplikacji obsługującej system darowizn oparty o kontakt zbierającego z darczyńcą.</p> <ol style="list-style-type: none">Wygodna i prosta w obsłudze aplikacja mobilna<ul style="list-style-type: none">Przejrzysty interfejs zgodny z obecnymi zasadamiWyświetlanie konta użytkownika
-----------------------------	---

	<ol style="list-style-type: none"> 2. Strona serwera <ul style="list-style-type: none"> • REST API dla pobierania danych z bazy • Możliwość rejestracji użytkownika i zalogowania go do bezpiecznej sesji • Renderowanie aplikacji webowej 3. Baza danych <ul style="list-style-type: none"> • Przechowywanie informacji na temat użytkowników, transakcji oraz powiadomień • Bezpieczne przechowywanie haseł i danych użytkowników 4. Klient <ul style="list-style-type: none"> • REST API do wysyłania zapytań do serwera • tworzenie i wysyłanie zapytań w formacie zrozumiałym dla serwera 5. Aplikacja webowa <ul style="list-style-type: none"> • Umożliwia podgląd platformy z przeglądarki 6. Obsługa płatności <ul style="list-style-type: none"> • Wejście do biblioteki mobilnej (przelewy24) w której nastąpi wybór płatności - paypal, przelew, płatność kartą 7. System powiadomień <ul style="list-style-type: none"> • Na ekranie smartfona pojawi się informacja o poprawnej wpłacie, powrót do aplikacji.
<p>Cechy charakterystyczne i ograniczenia wybranych technologii</p>	<p>Aplikacja mobilna jest tworzona w oparciu o system Android.</p> <p>Strona i web api zbudowane są na podstawie php i javascript oraz od części graficznej standard CSS3.</p> <p>Część backendowa projektu tworzona jest w języku Python oraz frameworku Django.</p> <p>Obsługa płatności będzie opierać się na bibliotece napisanej w języku Java.</p> <p>Grupa projektowa korzysta z Trello - prostego, szybkiego oraz skutecznego narzędzia do zarządzania zadaniami.</p>

	<p>Projekty zbudowany jest w oparciu o tablice, na których to znajdują się listy zadań. Listy te pozwalają tworzyć karty (zadania), które w bardzo prosty sposób możemy sortować, czy przenosić pomiędzy różnymi listami.</p> <p>Repozytorium kodu znajduje się w serwisie github (https://github.com/RegenJ/Donation-Aplication)</p>
Struktura podziału pracy	<p>Grupa projektowa podzielona jest zgodnie z preferencjami członków. Za stronę aplikację mobilną oraz webową odpowiedzialna jest Katarzyna Lebioda oraz Marcin Głaba.</p> <p>Analizą oraz później implementacją i obsługą systemu płatności i notyfikacji zajmuje się Marcin Głaba.</p> <p>Część serwerową i bazodanową tworzy Michał Swat.</p> <p>Jakub Regeńczuk odpowiedzialny jest za implementację strony klienckiej komunikującej się z serwerem i frontem aplikacji. Zajmuje się również koordynacją pracy członków oraz tworzeniem dokumentacji.</p> <p>Podział prac na poszczególne zadania jest aktualizowany na bieżąco na tablicach na Trello.</p>
Harmonogram	<ol style="list-style-type: none"> 1. Druga iteracja (wersja beta): system z większością działających funkcjonalności <ol style="list-style-type: none"> a. Poprawki do poprzedniej wersji b. Dodawanie kolejnych, niezbędnych funkcji systemu c. Scalenie i rozwiązanie konfliktów d. Testy Termin: 12 zajęcia (15.01) 2. Ostateczna wersja systemu <ol style="list-style-type: none"> a. Końcowe poprawki i dodatkowe funkcjonalności b. Dokumentacja poszczególnych części c. Całościowe testy systemu Termin: 13 zajęcia (22.01)
Wykonane	<ul style="list-style-type: none"> <input type="checkbox"/> Stworzenie pełnej specyfikacji systemu i podział pracy zespołu. <ol style="list-style-type: none"> d. Rozeznanie w dostępnych technologiach i wybranie najodpowiedniejszych dla danych części systemu e. Założenie repozytorium oraz ustalenie kanałów i zasad komunikacji f. Postawienie serwera Termin: do 4 zajęć (6.11) <input type="checkbox"/> Projekt systemu, w tym przynajmniej diagramu klas/obiektów i diagramu przypadków użycia

- g. Analiza systemów płatności oraz wybranie najodpowiedniejszego dla aplikacji do **13.11**
- h. Rozplanowanie architektury, wyglądu i funkcji aplikacji - spotkanie zespołu **8/9.11**
- i. Stworzenie diagramów UML
 - i. Pierwsza wersja **do 13.11**
 - ii. Poprawki do **20.11**
- j. Rozpoczęcie implementacji poszczególnych części systemu
 - i. Konfiguracja środowiska dla aplikacji mobilnej, stworzenie pierwszych widoków w aplikacji - **13.11**
 - ii. Serwer: endpointy dla rejestracji użytkownika (sprawdzanie czy już taki istnieje) **13.11**
 - iii. Serwer: umożliwienie korzystania z podstawowej bazy pozostałym członkom zespołu: chmura lub skrypt stawiający i konfigurujący **13.11**
- k. Dodanie stworzonych funkcjonalności na repozytorium
 - i. Stworzenie własnych branchy przez pozostałych członków od takiego wstępnego mastera

Termin: do 7 zajęć (27.11)

- ❑ Pierwsza iteracja systemu (wersja alpha): działający prototyp z implementacją podstawowych funkcjonalności systemu
 - l. Serwer i baza danych:
 - i. Rejestracja i logowanie **4.12**
 - ii. Tworzenie zbiórki przez użytkownika i przechowywanie zbiorów użytkownika **8.12**
 - iii. Zmiana hasła i wylogowywanie **16.12**
 - m. Klient:
 - i. Dokończenie metod logowania rejestracji **4.12**
 - ii. Komunikacja z serwerem: obsługa requestów **12.12**
 - n. Aplikacja mobilna
 - i. Widoki logowania/rejestracji, **8.12**
 - ii. Tworzenia zbiorów: widok z formularzem (nazwa, opis, data, suwak z kwotą) **8.12**
 - iii. Widok zbiórki
 - o. Aplikacja webowa
 - i. Strona rejestracji i logowania **4.12**
 - ii. Tworzenie zbiórki **4.12**
 - p. Płatności

	Termin: 10 zajęcia (18.12)
--	----------------------------