

Dokument wymagań projektowych

Nazwa projektu: LogInt

Autorzy: Daria Dworzyńska, Jakub Paszke, Miłosz Rolewski, Michał Wujec

Data: 24.05.2024

0. Wersje dokumentu

15.04.2024 - Wersja 1.0

24.05.2024 - Wersja 2.0

0.1 Słownik pojęć

1. Elementy składowe projektu (produkty projektu)

Semestr 1:

1. Dokumentacja:

- Stworzenie dokumentu opisującego funkcjonalności projektu.
- Sporządzenie dokumentacji dotyczącej przeprowadzonych testów, w tym szczegółowy opis metod testowania oraz wyników testów.
- Stworzenie Manuala (Instrukcji obsługi)

2. Stworzenie MVP (Minimum Viable Product):

- Wykonanie prototypu interfejsu użytkownika przy użyciu Django.
- Stworzenie prototypu dla aplikacji LogIntTester (napisanej przez nas), która będzie stanowić podstawę pod aplikację klienta ("szkielet" projektu).
- Stworzenie MVP dla aplikacji "Drive" (od DB Schenker).
- Konfiguracja baz danych – stworzenie modeli w bazie PostgreSQL
- Konfiguracja przykładowego środowiska egzekucji – emulatora android studio

3. Wdrożenie MVP u klienta

Semestr 2:

1. Moduł autoryzacji:

- Implementacja mechanizmu uwierzytelniania dla administratora.

2. Stworzenie MMP (Minimum Marketable Product):

- Zintegrowanie naszej strony z resztą aplikacji zdefiniowanych przez klienta.

3. Stworzenie MLP (Minimum Lovable Product):

- Dodanie dodatkowych funkcjonalności, tak aby praca dla administratora była jak najprostsza oraz intuicyjna – takie opcje jak filtrowanie, sortowanie, konkretne raporty do pobrania.

4. Wdrożenie MMP i MLP u klienta

2. Granice projektu

- Nie będą zapewnione funkcjonalności zarządzania pojazdami: śledzenia zużycia paliwa, planowanie tras, śledzenie rodzaju przewożonego towaru czy monitorowanie stanu technicznego pojazdów.
- Niektóre aplikacje kontrahentów mogą nie być kompatybilne z zaproponowanym przez nas systemem.
- Nie wszystkie procesy będą w pełni zautomatyzowane - niektóre aplikacje kontrahentów mogą wymagać interwencji manualnej w przypadku nieoczekiwanych sytuacji lub problemów związanych z integracją. Niektóre aplikacje (np. wymagające CAPTCHA) mogą być chronione przed automatycznym wypełnianiem danych, co może uniemożliwić pełną automatyzację procesu.
- Bezpieczeństwo danych: Projekt zawiera autoryzację dla administratora i zapewnia bezpieczny dostęp do funkcji administracyjnych, nie obejmuje pełnego zakresu funkcji związanych z bezpieczeństwem danych. W szczególności nie będzie wdrożona pełna polityka bezpieczeństwa informacji, w tym szyfrowanie danych czy zaawansowane mechanizmy kontroli dostępu.
- Przyszłe aktualizacje aplikacji nie są objęte naszym zakresem obowiązków. Za przeprowadzanie i zarządzanie aktualizacjami będzie odpowiedzialny administrator systemu.
- Skrypt jest ściśle związany ze środowiskiem wykonywalnym. Nie zapewniamy uniwersalizmu skryptów.
- Dostarczamy pokazowe środowisko egzekucyjne docelowo klient będzie musiał zapewnić urządzenie mobilne lub emulator dostosowany do jego osobistych potrzeb (oraz własnoręcznie go skonfigurować)
- Zakładka Source przyjmie dane tylko w formacie JSON.

3. Lista wymagań funkcjonalnych (Cechy wykrywane przez użytkownika)

1. System autoryzacji:

- administrator loguje się do systemu za pomocą loginu i hasła

2. Zarządzanie integracjami:

- Możliwość tworzenia, edycji i usuwania integracji z aplikacjami kontrahentów.
- Możliwość tworzenia, edycji i usuwania źródeł danych z aplikacjami kontrahentów.
- Określenie kryterii wyboru danych do przekazania oraz sposobu ich wprowadzania do aplikacji kontrahentów za pomocą zakładki "Emulator" lub własnoręcznej weryfikacji (działania aplikacji)

4. Generowanie raportów:

- Logi z wykonanych egzekucji

5. Monitorowanie historii integracji:

- Umożliwienie przeglądania historii wykonanych operacji (integracji, kont(Driver), source'ówi) sprawdzenia, które z nich wykonały się poprawnie.

4. Lista wymagań niefunkcjonalnych (Ograniczenia systemowe)

1. Poprawność systemu:

- Gwarancja dostępności i systemu dla użytkownika (administratora)
- Minimalny poziom poprawności systemu powinien wynosić 90% w ciągu każdego miesiąca kalendarzowego. Poziom ten będzie mierzony ilością pozytywnie wykonanych integracji przez ilość wszystkich integracji

2. Wydajność systemu:

- Wysoka wydajność systemu integracji, umożliwiająca obsługę dużej liczby aplikacji i integracji równocześnie.
- Trigger będzie uruchamiany co 10 minut aby pobierać nowe rekordy.

3. Elastyczność systemu:

- Możliwość dodania nowej aplikacji, nie określonej wcześniej przez klienta.

4. Zgodność z przeglądarkami:

- Kompatybilność programu z przeglądarką: Google Chrome, Firefox

5. Kryteria akceptacji projektu dla I semestru prac

Wymagane kryteria akceptacji:

- Dostarczenie prototypu interfejsu użytkownika - działający szkielet programu na naszej własnej aplikacji - LogIntTester
- Dostarczenie MVP – nasz system zostanie zintegrowany z aplikacją "Driver" od DB Schenker

- Przeprowadzenie testów: Każdy z task'ów zostanie przetestowany według konkretnych testów, które znajdują się w jego opisie na JIRA. Wszystkie testy zostaną spisane i udokumentowane w osobnym dokumencie umieszczonym na Githubie.
- Uzyskanie akceptacji od firmy, która potwierdzi, że wykonane prace są zgodne z założeniami i oczekiwaniami projektu.
- Zarządzanie istniejącymi integracjami

Oczekiwane kryteria akceptacji:

- Wdrożenie dodatkowych funkcjonalności: generowanie raportów i sprawdzanie historii.

6. Mierzalne wskaźniki wdrożeniowe

- Ukończenie wszystkich zaplanowanych funkcjonalności: Wskaźnik zostanie wyrażony jako procent zrealizowanych funkcjonalności w stosunku do wszystkich zaplanowanych. Jako funkcjonalności rozumie się listy wymagań w punkcie 3 oraz 4.
- Pozytywne wyniki przeprowadzonych testów. Wskaźnik ten będzie wyrażony jako procent testów zakończonych pomyślnie w stosunku do wszystkich wykonanych testów. Zostanie on wyznaczony Na koniec 1 semestru oraz na koniec 2 semestru na podstawie dokumentu na Githubie, gdzie będą udokumentowane wszystkie przeprowadzone testy.
- Sprawdzenie poprawności działania automatycznego wypełniania danych. Wskaźnik będzie wyrażony jako procent poprawnie wykonujących się integracji (czyli tych w których w kolumnie w historii będzie "available data") do wszystkich integracji.
- Otrzymanie opinii od klienta. Wskaźnik będzie w skali od 1 do 6 na podstawie pisemnej opinii otrzymanej na koniec 1 i 2 semestru od klienta. Finalny wynik będzie średnią arytmetyczną tych 2 wartości.
- Przygotowanie dokumentacji. Jako pełną dokumentację rozumie się: dokument wizji projektu, historię dokumentów wymagań projektowych, dokument przeprowadzonych testów, kod na Githubie, Manual.

7. Kryteria akceptacji projektu dla II semestru prac

Wymagane kryteria akceptacji:

- Zakończenie wszystkich zaplanowanych funkcjonalności, tak jak zaznaczono w elementach składowych projektu przewidzianych dla semestru 2.
- Konsultacja z firmą, która potwierdzi, że system spełnia założenia i wymagania klienta.

Oczekiwane kryteria akceptacji:

- Zaimplementowanie dodatkowych funkcjonalności: takich jak sortowanie i filtrowanie w zakładkach raporty i historia.
- Optymalizacja wydajności, tak aby integracje zajmowały jak najmniej czasu.

- Przygotowanie dokumentacji końcowej: Jako pełną dokumentację rozumie się: dokument wizji projektu, historię dokumentów wymaga projektowych, dokument przeprowadzonych testów, kod na Githubie, Manual.

8. Organizacja pracy zespołu

Zakres prac poszczególnych członków zespołu:

- Daria Dworzyńska: Jej zadaniem jest rozwój logiki biznesowej systemu oraz implementacja backendu aplikacji. Ponadto, zajmuje się integracją zewnętrznych systemów.
- Jakub Paszke: Jest odpowiedzialny za analizę wymagań klienta oraz projektowanie interfejsu użytkownika. Ponadto, zajmuje się programowaniem części frontendowej aplikacji.
- Miłosz Rolewski: Jego zadaniem będzie przygotowanie automatycznego systemu mapowania danych do środowiska mobilnego zawierającego aplikacje kontrahentów. Pomaga również w przygotowaniu dokumentacji technicznej.
- Michał Wujec: Jest odpowiedzialny za zarządzanie bazą danych oraz optymalizację jej wydajności. Pomaga w implementacji funkcjonalności wymagających pracy z danymi.

Role projektowe i deweloperskie:

- Daria Dworzyńska: Programista backendu, integrator systemów.
- Jakub Paszke: Projektant interfejsu użytkownika, programista frontendowy.
- Miłosz Rolewski: programista ds. systemów mobilnych, specjalista ds. jakości kodu
- Michał Wujec: Administrator bazy danych, specjalista ds. optymalizacji.

Zarządzanie komunikacją z klientem:

- Komunikację z klientem zarządza osoba pełniąca rolę Product Ownera, czyli Jakub Paszke. Komunikacja odbywa się głównie poprzez regularne spotkania, e-maile oraz platformy do zarządzania projektami, takie jak np. Jira.

Metodyka pracy:

- Zespół przyjął metodykę Agile, a dokładniej SCRUM. Wybór tej metodyki wynikał z jej elastyczności, umożliwiającej adaptację do zmieniających się wymagań klienta oraz umożliwiającej częste iteracje i szybkie dostarczanie wartościowych produktów. Zespół spotykał się 3 razy w tygodniu na Videochat'cie na platformie DISCORD.

Narzędzia wspomagające prace projektowe:

- Do zarządzania kodami źródłowymi zespół wykorzystuje system kontroli wersji Git, a kod przechowywany jest na platformie GitHub.
- Do zarządzania przebiegiem projektu wykorzystywana jest JIRA.

Diagram cyklu życia zadania w projekcie:

- Powstanie zadania: Zadanie pojawia się na tablicy zadań, po czym zostaje przypisane do odpowiedniego członka zespołu.
- Analiza wymagań: Członek zespołu dokonuje analizy wymagań i określa zakres oraz priorytet zadania.
- Implementacja: Zadanie jest implementowane przez odpowiedniego programistę, który tworzy kod i przeprowadza testy jednostkowe.
- Testowanie: Po zakończeniu implementacji zadanie przechodzi przez etap testowania, gdzie sprawdzana jest jego funkcjonalność.
- Akceptacja: Po pomyślnym przejściu testów zadanie zostaje zaakceptowane przez Product Ownera i uznane za wykonane.
- Wdrożenie: Wykonane zadanie zostaje wdrożone na serwer produkcyjny i udostępnione użytkownikom.

9. Ryzyka projektowe

1. Opóźnienia w integracji nowych aplikacji

- Opis: Integracja z nową aplikacją klienta może być bardziej skomplikowana niż pierwotnie zakładano ze względu na różnice w strukturze danych i ich przepływie.
- Szansa zaistnienia: Średnia
- Stopień wpływu: Wysoki
- Postępowanie: Regularna komunikacja z kontrahentami, wcześniejsze dostosowywanie integracji do ich wymagań.

2. Ryzyko: Błędy w bezpieczeństwie danych

- Opis: Istnieje ryzyko wystąpienia luk w zabezpieczeniach, co będzie prowadzić do wycieku danych.
- Szansa zaistnienia: Niska
- Stopień wpływu: Bardzo wysoki
- Postępowanie: Regularne przeglądy kodu, szybkie reagowanie na wszelkie znalezione luki w zabezpieczeniach.

3. Ryzyko: Niezgodność z oczekiwaniami klienta

- Opis: Dostarczone rozwiązanie nie będzie spełniać oczekiwań klienta pod względem funkcjonalności lub wydajności.
- Szansa zaistnienia: Wysoka
- Stopień wpływu: Wysoki

- Postępowanie: Regularne prezentacje, MVP i MMP dla klienta, częste spotkania w celu weryfikacji postępów, wczesne identyfikowanie ewentualnych niezgodności i szybkie dostosowywanie się do nowych wymagań.

4. Ryzyko: Problemy z wydajnością aplikacji

- Opis: Istnieje ryzyko, że aplikacja będzie działać wolno lub niestabilnie, szczególnie podczas przepływu większej ilości danych.

- Szansa zaistnienia: Niski

- Stopień wpływu: Wysoki

- Postępowanie: Regularne testy wydajnościowe, optymalizacja kodu i multiprocessing.

5. Ryzyko: Niezgodność technologii z oczekiwaniami klienta

- Opis: Wybrane technologie nie spełniają oczekiwań klienta pod względem wydajności, skalowalności lub bezpieczeństwa.

- Szansa zaistnienia: Niska

- Stopień wpływu: Średni

- Postępowanie: Regularna konsultacja z klientem na temat wykorzystywanych technologii.

10. Kamienie milowe

1. Pierwsze wydanie prototypu interfejsu użytkownika (UI)

Termin: 10.05.2024

Opis: Opracowanie i dostarczenie pierwszej wersji prototypu interfejsu użytkownika, umożliwiającej klientowi zapoznanie się z planowanym wyglądem i funkcjonalnościami aplikacji.

2. Wydanie prototypu, przetestowanego na aplikacji LogIntTester

Termin: 24.05.2024

Opis: Na stworzonej przez nasz zespół aplikacji, zostaną przetestowane wszystkie funkcjonalności strony zaplanowane na 1 semestr.

3. Wydanie MVP (Minimum Viable Product) - w pełni zintegrowany system dla aplikacji "Driver" od DB Schenker

Termin: 30.06.2024

Opis: Wydanie w pełni działającej strony, umożliwiającej korzystanie ze wszystkich wymaganych funkcjonalnych dla aplikacji "Driver".

4. Publikacja MMP (Minimum Marketable Product) - w pełni zintegrowany system dla pozostałych aplikacji klienta

Termin: 30.09.2024

Opis: Udostępnienie wersji MMP do testów końcowych klienta w celu zbierania opinii, identyfikacji ewentualnych błędów oraz sugestii dotyczących poprawy interfejsu i funkcjonalności.

5. Publikacja MLP (Minimum Lovable Product)

Termin: 30.10.2024

Opis: Po feedbacku od klienta dodanie funkcjonalności tak aby nasz produkt był bardziej intuicyjny i prostszy w użytkowaniu

5. Finalizacja projektu

Termin: 1.12.2024

Opis: Zakończenie wszystkich testów i wprowadzenie ostatnich poprawek zgodnie z opiniami klienta oraz wynikami testów. Finalizacja wersji produkcyjnej gotowej do wdrożenia.