Dokument wymagań projektowych

Nazwa projektu: LogInt

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

Data: 7.06.2024

0. Wersje dokumentu

15.04.2024 - Wersja 1.0

24.05.2024 - Wersja 2.0

7.06.2024 - Wersja 2.1

1. Elementy składowe projektu (produkty projektu)

Semestr 1:

1. Dokumentacja:

- Dokument wymagan projektowych
- Dokumentacja testowa
- Dokumentacja technologiczna
- Dokumentacja wdrożeniowa
- Diagramy:
    - Diagram klas
- Diagram przypadków użycia
- Diagram aktywności
- Diagram wdrożeniowy
- Protokół z wdrożenia funkcjonalności na koniec I semestru

2. MVP (Minimum Viable Product):

- Interfejs użytkownika przy użyciu Django, JavaScript, CSS, HTML, wdrożono przy użyciu Dockera
- Aplikacja LogIntTester (imitująca aplikacje kontrahentów)
- MVP dla LogIntTester

- MVP dla aplikacji określonej przez klienta

- Modele w bazie PostgreSQL (jak na diagramie klas)

Semestr 2:

1. Moduł autoryzacji:

    - Mechanizm uwierzytelniania dla administratora.

2. MMP (Minimum Marketable Product):

    - Integracja produktu z resztą aplikacji zdefiniowanych przez klienta.

    - Integracja z raportami

3. Stworzenie MLP (Minimum Lovable Product):

    - Dodatkowe funkcjonalności, tak aby praca dla administratora była jak najprostsza oraz intuicyjna: filtrowanie, sortowanie

4. Protokół z wdrożenia funkcjonalności na koniec I semestru

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, co może stanowić granicę funkcjonalności.
- 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 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.
- Dostarczamy pokazowe środowisko egzekucyjne docelowo klient będzie musiał zapewnić urządzenie mobilne lub emulator dostosowany do jego osobistych potrzeb

3. Lista wymagań funkcjonalnych (Cechy wykrywane przez użytkownika)

1. System autoryzacji:

- Administrator loguje się do systemu za pomocą loginu i hasła lub konta google

2. Zarządzanie integracjami:

- Dodawanie, edycja i usuwanie integracji z aplikacjami kontrahentów.

- Dodawanie, edycja i usuwanie kierowców pod daną integracją

- Wybór danych do przekazania oraz sposobu ich wprowadzania do aplikacji kontrahentów za pomocą zakładki “Emulator”

4. Generowanie raportów:

- Raporty z wykonanych integracji, zawierające informacje o statusie procesu integracyjnego oraz błędach (które dane nie zostały uzupełnione i z powodu jakiego błędu) .

5. Monitorowanie historii integracji:

- Podgląd wykonanych integracji i statusy, które z nich wykonały się poprawnie.

4. Lista wymagań niefunkcjonalnych (Ograniczenia systemowe)

1. Dostępność systemu:

- Gwarancja dostępności i systemu dla użytkownika (administratora)

- Minimalny poziom dostępności 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 sekund, aby pobierać nowe rekordy.

3. Elastyczność systemu:

- Dodanie nowej aplikacji, nie określonej wcześniej przez klienta.

4. Zgodność z przeglądarkami:

- Kompatybilość środowiska kreacji z przeglądarką: Google Chrome

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

- Dostarczenie pełnej dokumentacji zdefiniowanej w 1.1

Oczekiwane kryteria akceptacji:

- Wdrożenie dodatkowych funkcjonalności: generowanie raportów, sprawdzanie historii i zarządzanie istniejącymi integracjami

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 wymagan 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. Planowana liczba wszystkich integracji to 600.

- Otrzymanie opinii od klienta. Wskaźnik będzie w skali od 1 do 6 na podstawie pisemnej opinii otrzymanej na koniec 2 semestru od klienta.

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

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.

Oczekiwane kryteria akceptacji:

- Sortowanie i filtrowanie w zakładkach raporty i historia.

- Optymalizacja wydajności - skrócenie czasu działania pojedynczej egzekucji

- Dokumentacja końcowa. (Dokumentacja z Semestr 1 punkt 1. + Dokument z Semestr 4 punkt 4)

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.

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: Średnia

    - Stopień wpływu: Średni

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

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 LogIntTestere

Termin: 24.05.2024

Opis: Na stworzonej przez nasz zespół aplikacji, zostaną przetestowane wszystkie funkcjonalności strony.

3. Wydanie MVP (Minimum Viable Product) - w pełni zintegrowany system dla aplikacji 1 aplikacji wyznaczonej przez klienta

Termin: 15.06.2024

Opis: Wydanie w pełni działającej strony, umożliwiającej korzystanie ze wszystkich wymagan funkcjonalnych dla aplikacji klienta.

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'u 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.