

Dokument wymagań projektowych

Nazwa projektu: LogInt

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

Data: 15.04.2024

0. Wersje dokumentu

15.04.2024 - Wersja 1.0

1. Elementy składowe projektu (produkty projektu)

Semestr 1:

1. Moduł autoryzacji:

- Implementacja mechanizmu uwierzytelniania dla administratora.
- Zapewnienie bezpiecznego dostępu do funkcji administracyjnych.

2. Specyfikacja funkcjonalna:

- Dokument opisujący szczegółowo funkcjonalności projektu.
- Określenie wymagań użytkownika oraz sposobu ich realizacji.

3. Prototyp interfejsu użytkownika:

- Wykonanie prototypu interfejsu użytkownika przy użyciu Django.
- Przedstawienie wstępnej wizualizacji aplikacji serwerowej oraz panelu administracyjnego.

4. Analiza technologiczna:

- Przeprowadzenie analizy porównawczej wydajności technologii.
- Uzasadnienie wyboru technologii Python i Django do implementacji aplikacji serwerowej.

Semestr 2:

1. Aplikacja serwerowa w języku Python oparta o framework Django:

- Implementacja ogólnej logiki projektu.
- Przechwytywanie odpowiedzi API, mapowanie danych oraz ustawienie warunków czasowych
- Instancja serwera bazy danych oparta o silnik MSSQL.

2. Konfiguracja serwera bazy danych.

- Tworzenie zapytań SQL umożliwiających efektywne zarządzanie danymi.

3. Dokumentacja użytkownika:

- Opracowanie dokumentacji użytkownika w formie dokumentu oraz systemu pomocy.
- Przygotowanie samouczka dla administratorów aplikacji.

4. Prezentacja informacji o funkcjonalnościach i korzyściach wynikających z użycia aplikacji.

- Raporty z wykonanych testów użyteczności.

5. Przeprowadzenie testów użyteczności aplikacji.

6. Dokumentacja wyników testów i wniosków dotyczących poprawek.

2. Granice projektu

- Funkcjonalności w zakresie zarządzania flotą pojazdów: Chociaż projekt ma na celu usprawnienie zarządzania informacjami dotyczącymi kierowców i pojazdów, nie obejmuje pełnego zakresu funkcji związanych z zarządzaniem flotą. Na przykład, nie będzie zapewniona funkcjonalność śledzenia zużycia paliwa, planowania tras czy monitorowania stanu technicznego pojazdów.
- Integracja z wszystkimi aplikacjami kontrahentów: Projekt zakłada integrację z aplikacjami kontrahentów, ale nie gwarantuje możliwości integracji ze wszystkimi aplikacjami dostępnymi na rynku. Istnieje możliwość, że niektóre aplikacje kontrahentów nie będą kompatybilne z proponowanym rozwiązaniem, co może stanowić granicę funkcjonalności.
- Pełna automatyzacja procesu integracji: Pomimo planowanej automatyzacji procesu wypełniania danych i integracji z aplikacjami kontrahentów, istnieje ryzyko, że nie wszystkie procesy będą w pełni zautomatyzowane. Na przykład, niektóre aplikacje kontrahentów mogą wymagać interwencji manualnej w przypadku nieoczekiwanych sytuacji lub problemów związanych z integracją.
- Bezpieczeństwo danych: Chociaż projekt uwzględni moduł autoryzacji 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.
- Wsparcie techniczne: Projekt zakłada, że administrator będzie odpowiedzialny za zarządzanie integracjami, ale nie obejmuje pełnego zakresu wsparcia technicznego dla klientów. Nie będzie dostępne 24/7 wsparcie techniczne, a klient będzie musiał polegać na dokumentacji oraz ograniczonym wsparciu telefonicznym w przypadku problemów.
- Ograniczenia technologiczne: Choć projekt zakłada wykorzystanie narzędzi takich jak Appium/Selenium/Emulator telefonu do integracji z aplikacjami mobilnymi, istnieją pewne ograniczenia technologiczne, które mogą wpłynąć na skuteczność integracji. Na przykład, niektóre aplikacje mogą być chronione przed automatycznym wypełnianiem danych, co może uniemożliwić pełną automatyzację procesu.

3. Lista wymagań funkcjonalnych

1. System autoryzacji:

- Umożliwienie administratorowi logowania się do systemu za pomocą unikalnego identyfikatora i hasła.
- Zapewnienie bezpiecznego dostępu do funkcji administracyjnych poprzez autoryzację użytkownika.

2. Zarządzanie integracjami:

- Możliwość tworzenia, edycji i usuwania integracji z aplikacjami kontrahentów.
- Zapewnienie interfejsu graficznego umożliwiającego konfigurację źródła danych, typu aplikacji, harmonogramu działania integracji oraz mapowanie danych.

3. Automatyczne wypełnianie danych:

- Realizacja mechanizmu automatycznego wypełniania pól w aplikacjach kontrahentów na podstawie danych zgromadzonych przez klienta.
- Umożliwienie administratorowi określenia kryteriów wyboru danych do przekazania oraz sposobu ich wprowadzania do aplikacji kontrahentów.

4. Generowanie raportów:

- Możliwość generowania raportów z wykonanych integracji, zawierających informacje o statusie procesu integracyjnego oraz ewentualnych błędach.
- Zapewnienie funkcji eksportu raportów w różnych formatach (np. PDF, CSV) dla potrzeb analizy i archiwizacji danych.

5. Monitorowanie historii integracji:

- Zapewnienie możliwości przeglądania historii wykonanych integracji wraz z szczegółowymi informacjami dotyczącymi wysłanych danych i rezultatów integracji.
- Umożliwienie administratorowi śledzenia postępów procesu integracyjnego oraz identyfikowanie ewentualnych problemów.

6. Interfejs użytkownika:

- Stworzenie intuicyjnego interfejsu użytkownika, umożliwiającego łatwą nawigację i wykonywanie wszystkich funkcji systemu.
- Zapewnienie responsywności interfejsu, aby umożliwić korzystanie z systemu na różnych urządzeniach, w tym także na urządzeniach mobilnych.

7. Zabezpieczenie danych:

- Zapewnienie bezpiecznego przechowywania danych klienta oraz danych uwierzytelniających.
- Implementacja mechanizmów szyfrowania danych oraz kontroli dostępu, aby zapobiec nieautoryzowanemu dostępowi do informacji.

8. Wsparcie techniczne:

- Udostępnienie dokumentacji technicznej oraz materiałów szkoleniowych dla administratorów systemu.
- Zapewnienie ograniczonego wsparcia technicznego dla klientów w przypadku problemów związanych z korzystaniem z systemu.

4. Lista wymagań niefunkcjonalnych

Dostępność systemu:

- Wymóg zapewnienia wysokiej dostępności systemu, minimalizujący czas przestoju i gwarantujący dostępność usług dla użytkowników.
- Minimalny poziom dostępności systemu powinien wynosić 99,5% w ciągu każdego miesiąca kalendarzowego.

Wydajność systemu:

- Wymóg zapewnienia wysokiej wydajności systemu, umożliwiającej obsługę dużej liczby użytkowników i integracji równocześnie.
- Czas odpowiedzi systemu na żądania użytkowników nie powinien przekraczać 1 sekundy dla większości operacji.

Skalowalność systemu:

- Wymóg zapewnienia możliwości skalowania systemu w razie wzrostu obciążenia lub liczby użytkowników.
- System powinien być elastyczny i łatwo rozszerzalny, umożliwiając dynamiczne dostosowanie zasobów w zależności od potrzeb.

Interfejs użytkownika:

- Wymóg zapewnienia intuicyjnego i responsywnego interfejsu użytkownika, umożliwiającego łatwą nawigację i korzystanie z systemu na różnych urządzeniach.
- Interfejs powinien być przyjazny dla użytkownika, zrozumiały nawet dla osób nieobeznanych z technologią.

Zgodność z przeglądarkami:

- Wymóg zapewnienia kompatybilności systemu z popularnymi przeglądarkami internetowymi, takimi jak Google Chrome, Mozilla Firefox, Safari oraz Microsoft Edge.
- System powinien być testowany i zoptymalizowany pod kątem różnych przeglądarek, aby zapewnić spójne działanie na wszystkich platformach.

Wsparcie techniczne:

- Wymóg zapewnienia dokumentacji technicznej oraz materiałów szkoleniowych dla administratorów systemu.
- Ograniczone wsparcie techniczne dla klientów w przypadku problemów związanych z korzystaniem z systemu, w tym możliwość zgłaszania błędów i prośba o pomoc.

5. Kryteria akceptacji projektu dla I semestru prac

Wymagane kryteria akceptacji:

- Dostarczenie prototypu interfejsu użytkownika: Wymagane jest przedstawienie prototypu interfejsu użytkownika, który będzie prezentował wygląd i funkcjonalności planowanej aplikacji. Prototyp powinien zostać zaakceptowany przez klienta.
- Zaimplementowanie podstawowej logiki systemu: Wymagane jest zaimplementowanie podstawowej logiki systemu, umożliwiającej przetwarzanie danych oraz zarządzanie integracjami.
- Przeprowadzenie testów funkcjonalnych: Wymagane jest przeprowadzenie testów funkcjonalnych, które potwierdzą poprawne działanie podstawowych funkcji systemu. Testy powinny obejmować główne scenariusze użycia.
- Uzyskanie akceptacji od opiekuna projektu: Wymagane jest uzyskanie akceptacji od opiekuna projektu, który potwierdzi, że wykonane prace są zgodne z założeniami i oczekiwaniami projektu.

Oczekiwane kryteria akceptacji:

- Integracja z wewnętrznym systemem autoryzacji klienta: Oczekiwane jest zaimplementowanie integracji z wewnętrznym systemem autoryzacji klienta, umożliwiającej logowanie użytkowników za pomocą istniejących danych uwierzytelniających.
- Wdrożenie dodatkowych funkcjonalności: Oczekiwane jest wdrożenie dodatkowych funkcjonalności, takich jak generowanie raportów czy zarządzanie użytkownikami, które zwiększą użyteczność systemu.
- Zakończenie testów obciążeniowych: Oczekiwane jest zakończenie testów obciążeniowych, które pozwolą ocenić wydajność i skalowalność systemu podczas pracy w rzeczywistych warunkach.
- Zrozumienie procesów zachodzących u klienta: Oczekiwane jest zrozumienie procesów biznesowych klienta oraz dostosowanie projektu do ich potrzeb i wymagań, co pozwoli na efektywne wsparcie działań firmy.

6. Mierzalne wskaźniki wdrożeniowe

- Ukończenie wszystkich zaplanowanych funkcjonalności: Ocena stopnia ukończenia wszystkich zaplanowanych funkcjonalności projektu. Wskaźnik ten może być wyrażony jako procent zrealizowanych funkcji w stosunku do zaplanowanych.
- Pozytywne wyniki testów akceptacyjnych: Ocena wyników testów akceptacyjnych, które potwierdzą zgodność działania systemu z oczekiwaniami klienta. Wskaźnik ten może być wyrażony jako procent testów zakończonych pomyślnie.
- Zakończenie integracji z systemami kontrahentów: Ocena zakończenia procesu integracji z aplikacjami kontrahentów i sprawdzenie poprawności działania automatycznego wypełniania danych. Wskaźnik ten może być wyrażony jako procent zakończonych integracji.
- Otrzymanie pozytywnej opinii od klienta: Ocena zadowolenia klienta z gotowego produktu, wyrażona poprzez uzyskanie pozytywnej opinii lub akceptacji produktu do użytku. Wskaźnik

ten może być wyrażony jako procent klientów, którzy wyrazili zadowolenie z gotowego produktu.

- Przygotowanie dokumentacji użytkowej: Ocena przygotowania kompletnych i zrozumiałych materiałów dla użytkowników końcowych, umożliwiających efektywne korzystanie z systemu. Wskaźnik ten może być wyrażony jako procent ukończenia dokumentacji użytkowej.

- Zakończenie szkolenia dla administratorów systemu: Ocena ukończenia szkolenia dla administratorów systemu, które umożliwi im skuteczne zarządzanie i utrzymanie działania systemu. Wskaźnik ten może być wyrażony jako procent ukończenia szkolenia przez wszystkich administratorów.

7. Kryteria akceptacji projektu dla II semestru prac

Wymagane kryteria akceptacji:

- Zakończenie wszystkich zaplanowanych funkcjonalności: Wymagane jest ukończenie wszystkich zaplanowanych funkcjonalności projektu, jak określono w fazie planowania.
- Pozytywne wyniki testów końcowych: Wymagane jest uzyskanie pozytywnych wyników testów końcowych, które potwierdzą, że system spełnia założenia i wymagania klienta.
- Uzyskanie akceptacji od klienta: Wymagane jest uzyskanie akceptacji od klienta dla dostarczonych produktów, co oznacza, że są one zgodne z oczekiwaniami i gotowe do użytku.

Oczekiwane kryteria akceptacji:

- Zaimplementowanie dodatkowych funkcjonalności: Oczekiwane jest zaimplementowanie dodatkowych funkcjonalności, które zostały zidentyfikowane w trakcie rozwoju projektu i uznane za istotne dla zwiększenia wartości produktu.
- Optymalizacja wydajności i bezpieczeństwa: Oczekiwane jest przeprowadzenie optymalizacji systemu pod kątem wydajności i bezpieczeństwa, aby zapewnić płynne działanie oraz chronić dane przed nieautoryzowanym dostępem.
- Przygotowanie dokumentacji końcowej: Oczekiwane jest przygotowanie pełnej dokumentacji końcowej, która umożliwi użytkownikom efektywne korzystanie z systemu oraz jego dalsze utrzymanie.
- Ukończenie szkoleń dla personelu klienta: Oczekiwane jest przeprowadzenie szkoleń dla personelu klienta, które umożliwią im skuteczne korzystanie z systemu i pełne wykorzystanie jego możliwości.

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 wykorzystywane są narzędzia do zarządzania projektami, takie jak 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. **Ryzyko: Opóźnienia w integracji zewnętrznych systemów**

- Opis: Integracja z zewnętrznymi systemami może być bardziej skomplikowana niż pierwotnie zakładano ze względu na różnice w strukturze danych, dostępności interfejsów lub brak dokumentacji.
- Szansa zaistnienia: Średnia
- Stopień wpływu: Wysoki
- Postępowanie: Regularna komunikacja z dostawcami zewnętrznych systemów, wcześniejsze testowanie i ewentualne dostosowywanie integracji do ich wymagań.

2. **Ryzyko: Błędy w bezpieczeństwie danych**

- Opis: Istnieje ryzyko wystąpienia luk w zabezpieczeniach, co może prowadzić do wycieku danych lub ataków na system.
- Szansa zaistnienia: Niska
- Stopień wpływu: Bardzo wysoki
- Postępowanie: Regularne przeglądy kodu, testy penetracyjne, stosowanie beweracyjnych procesów bezpieczeństwa, szybkie reagowanie na wszelkie znalezione luki w zabezpieczeniach.

3. **Ryzyko: Niezgodność z oczekiwaniami klienta**

- Opis: Istnieje możliwość, że dostarczone rozwiązanie nie będzie spełniać oczekiwań klienta pod względem funkcjonalności lub wydajności.
- Szansa zaistnienia: Średnia
- Stopień wpływu: Wysoki
- Postępowanie: Regularne prezentacje i demo 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 obciążenia większej liczby użytkowników.
- Szansa zaistnienia: Średnia
- Stopień wpływu: Średni
- Postępowanie: Regularne testy wydajnościowe, optymalizacja kodu i infrastruktury, monitorowanie aplikacji w czasie rzeczywistym w celu szybkiego reagowania na ewentualne problemy.

5. ****Ryzyko: Niezgodność technologii z oczekiwaniami klienta****

- Opis: Może się okazać, że wybrane technologie nie spełniają oczekiwań klienta pod względem wydajności, skalowalności lub bezpieczeństwa.
- Szansa zaistnienia: Niska
- Stopień wpływu: Średni
- Postępowanie: Regularna ocena wykorzystywanych technologii, konsultacje z ekspertami branżowymi, elastyczność w stosowaniu alternatywnych rozwiązań w razie potrzeby.

6. ****Ryzyko: Nieprzewidziane problemy z infrastrukturą****

- Opis: Problemy z serwerami, chmurą obliczeniową lub inną infrastrukturą mogą prowadzić do niedostępności aplikacji lub utraty danych.
- Szansa zaistnienia: Średnia
- Stopień wpływu: Wysoki
- Postępowanie: Monitorowanie infrastruktury w czasie rzeczywistym, regularne backupy danych, plan awaryjności i szybkie działanie w przypadku wystąpienia problemów.

10. Kamienie milowe

Pierwsze wydanie prototypu interfejsu użytkownika (UI)

Termin: 15.05.2024

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

Zakończenie implementacji logiki biznesowej (backend)

Termin: 30.06.2024

Opis: Zakończenie implementacji wszystkich planowanych funkcjonalności w warstwie logiki biznesowej aplikacji, umożliwiających prawidłowe działanie i obsługę danych.

Publikacja wersji beta aplikacji do testów użytkowników

Termin: 15.09.2024

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

Zakończenie testów akceptacyjnych i finalizacja wersji produkcyjnej

Termin: 30.12.2024

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