|  |  |  |
| --- | --- | --- |
| **Wersja** | **Data modyfikacji** | **Najważniejsze zmiany** |
| 1.0 | 19.03.2023 | Stworzenie dokumentu; dodanie wymagań użytkownika. |
| 2.0 | 01.04.2023 | Uzupełnienie dokumentacji, dodanie diagramu przypadków użycia. |
| 3.0 | 06.05.2023 | Uzupełnienie przypadków użycia; Uzupełnienie diagramu klas. |
| 4.0 | 20.05.2023 | Uzupełnienie dokumentacji, dodanie elementów GUI. |
| 5.0 | 07.06.2023 | Uzupełnienie dokumentacji BYT |



1. **Wstęp**



**1.1 Cel**

Celem projektu jest opracowanie i wdrożenie systemu zarządzania nieruchomościami, który umożliwi efektywne śledzenie i zarządzanie informacjami związanymi z budynkami (biurowymi i magazynowymi), firmami, usługodawcami oraz wykonawcami. System ma ułatwić procesy wynajmu, świadczenia usług i budowy, a także zapewnić różnym użytkownikom odpowiednie narzędzia do zarządzania danymi.

**1.2 Zakres**

1. Ewidencja budynków: Moduł odpowiedzialny za przechowywanie informacji o budynkach, takich jak adres, powierzchnia, wysokość i status. Umożliwia dodawanie, aktualizowanie i przeglądanie informacji na temat budynków biurowych i magazynowych.

Rejestr firm: Moduł śledzący informacje o firmach, takie jak adres, nazwa, NIP, REGON, data rozpoczęcia i zakończenia współpracy. Umożliwia zarządzanie informacjami o firmach w różnych rolach, takich jak najemca, usługodawca lub wykonawca.

Zarządzanie wynajmem: Moduł odpowiedzialny za przechowywanie informacji o wynajmie budynków, takich jak firma wynajmująca, termin i cena wynajmu. Umożliwia śledzenie i aktualizowanie informacji dotyczących wynajmu.

Ewidencja usługodawców: Moduł śledzący informacje o usługodawcach, ich dostarczanych usługach, cenach oraz statusie aktywności. Umożliwia zarządzanie informacjami na temat usługodawców oraz usług przez nich oferowanych.

Rejestr wykonawców: Moduł przechowujący informacje o wykonawcach biorących udział w budowie, takich jak nazwa, adres i inne dane kontaktowe. Umożliwia zarządzanie informacjami o wykonawcach oraz ich ocenami.

**1.3 Definicje, akronimy i skróty:**

* **CRM** (Customer Relationship Management) - System zarządzania relacjami z klientami, służący do organizacji, automatyzacji i synchronizacji procesów biznesowych związanych z obsługą klienta.
* **ERP** (Enterprise Resource Planning) - System planowania zasobów przedsiębiorstwa, który integruje zarządzanie kluczowymi aspektami działalności, takimi jak finanse, logistyka, produkcja czy kadry.
* **API** (Application Programming Interface) - Interfejs programowania aplikacji, zestaw reguł i protokołów komunikacji między różnymi oprogramowaniami, umożliwiający ich współpracę.
* **REST** (Representational State Transfer) - Architektura oprogramowania stosowana w systemach rozproszonych, oparta na protokole HTTP i często używana w przypadku usług sieciowych.
* **SOAP** (Simple Object Access Protocol) - Protokół komunikacji między aplikacjami oparty na wymianie wiadomości XML, używany głównie w usługach sieciowych.
* **gRPC** (gRPC Remote Procedure Call) - System otwartego protokołu zdalnego wywoływania procedur, oparty na HTTP/2 i protokole Protocol Buffers, zaprojektowany dla szybkiej i wydajnej komunikacji między usługami.
* **NIP** (Numer Identyfikacji Podatkowej) - Numer służący do identyfikacji podatników w Polsce, wydawany przez organy podatkowe.
* **REGON** (Rejestr Gospodarki Narodowej) - Numer służący do identyfikacji podmiotów gospodarczych w Polsce, nadawany przez Główny Urząd Statystyczny.
* **Kontekst projektu** - Opisujący czynniki, które skłoniły do opracowania systemu zarządzania nieruchomościami, takie jak rosnące zapotrzebowanie na efektywne zarządzanie nieruchomościam
* **Charakterystyka użytkowników** - Opisująca różne role użytkowników w systemie, takie jak administratorzy, menedżerowie nieruchomości, najemcy itp.
* **Założenia i zależności** - Określające kluczowe wymagania techniczne, technologie używane w projekcie oraz ewentualne integracje z istniejącymi systemami.
* **Wymagania niefunkcjonalne** - Precyzujące wymagania dotyczące wydajności, skalowalności, bezpieczeństwa, dostępności itp. Wymagania powinny być jasno zdefiniowane, a także opisane wraz z metodami weryfikacji, takimi jak metryki czy testy.

**1.4 Odsyłacze**



Poniżej znajduje się lista źródeł i odsyłaczy do innych dokumentów, które mogą być przydatne podczas realizacji projektu oraz zapewniają dodatkowe informacje na temat technologii, standardów i praktyk:

* Dokumentacja API systemu CRM - szczegółowy opis interfejsów API dostępnych w systemie CRM, umożliwiających integrację z systemem zarządzania nieruchomościami.
* Dokumentacja API systemu finansowego - opis interfejsów API systemu finansowego, które umożliwiają integrację z systemem zarządzania nieruchomościami w celu automatyzacji procesów finansowych.
* Dokumentacja API systemu ERP - źródło informacji na temat interfejsów API dostępnych w systemie ERP, które mogą być użyte do integracji z systemem zarządzania nieruchomościami.
* Specyfikacja protokołu REST - oficjalna dokumentacja i wytyczne dotyczące architektury REST, stosowanej w przypadku usług sieciowych opartych na protokole HTTP.
* Specyfikacja protokołu SOAP - opis i wytyczne dotyczące protokołu SOAP, używanego w komunikacji między aplikacjami opartą na wymianie wiadomości XML.

**1.5 Przegląd**



Dokumentacja systemu zarządzania nieruchomościami została zorganizowana w sposób uporządkowany i hierarchiczny, aby ułatwić czytelnikom zrozumienie struktury projektu oraz dostęp do informacji na temat poszczególnych aspektów systemu.

**2. Ogólny opis**

**2.1 Kontekst**



System zarządzania nieruchomościami ma na celu śledzenie i zarządzanie informacjami związanymi z budynkami (biurowymi i magazynowymi), firmami, usługodawcami oraz wykonawcami. Umożliwia on użytkownikom rejestrowanie, aktualizowanie i zarządzanie informacjami dotyczącymi nieruchomości, firm, wynajmu, usług i budowy. System jest dostępny dla różnych rodzajów użytkowników, takich jak administrator aplikacji, usługodawca i wykonawca, każdy z odpowiednimi uprawnieniami i możliwościami.

**Systemy nadrzędne/zewnętrzne:**

* System zarządzania klientami (CRM) - współpraca z systemem CRM może pozwolić na lepsze zarządzanie relacjami z klientami, takimi jak firmy, usługodawcy i wykonawcy.
* System finansowy - integracja z systemem finansowym umożliwia automatyczne generowanie faktur i zarządzanie płatnościami związanych z wynajmem oraz usługami.
* System planowania zasobów przedsiębiorstwa (ERP) - współpraca z systemem ERP może pozwolić na lepsze zarządzanie zasobami, takimi jak wyposażenie i infrastruktura.

**Zasady współdziałania:**

* Współpraca z innymi systemami powinna być możliwa poprzez API lub protokoły komunikacji, takie jak REST, SOAP lub gRPC.
* System powinien obsługiwać różne poziomy dostępu dla różnych użytkowników, umożliwiając odpowiednie uprawnienia do zarządzania danymi.
* System powinien być skalowalny i elastyczny, umożliwiając łatwe rozbudowywanie funkcjonalności oraz integracje z innymi systemami.
* Bezpieczeństwo danych powinno być priorytetem. System powinien stosować odpowiednie zabezpieczenia, takie jak szyfrowanie danych, uwierzytelnianie i kontrola dostępu.
* System powinien być łatwy w obsłudze i posiadają intuicyjny interfejs użytkownika, umożliwiający zarządzanie danymi przez osoby o różnym poziomie doświadczenia.

**2.2 Wymagania użytkownika**

1. System ma przechowywać informacje o budynkach (adres, powierzchnia, wysokość, status {w budowie, dostępny, zajęty})
2. Budynki dzielą się na biurowe lub magazynowe. Budynki biurowe mają dodatkowo posiadać informację o ilości pięter.
3. Budynek magazynowy może mieć dodatkowo dobudówkę w postaci małego budynku biurowego. Musi przechowywać informację o posiadanych atestach.
4. System ma przechowywać dane o firmach (adres, nazwa, NIP, REGON, Data rozpoczęcia współpracy, data zakończenia współpracy (jeśli dotyczy)). Firma może występować w roli najemcy usługodawcy, wykonawcy oraz agenta nieruchomości.
5. System musi przechowywać informację o wynajmach budynków (firma wynajmująca, termin, cena wynajmu)
6. Usługodawcy powinni uwzględniać informacje o dostarczanych usługach i ich cenie, dodatkowo przy usługodawcy powinniśmy uwzględnić, czy jest aktywny {tak/nie}(usługodawca zostaje automatycznie oznaczony jako nie aktywny po pól roku nie świadczenia usług).
7. Najemca musi przechowywać informacje o rodzaju najmu.
8. Wykonawca przechowuję informacje o ewentualnych pod wykonawcach.
9. Dla każdego budynku należy przechowywać informacje o wykonanych usługach (data rozpoczęcia, cena, data zakończenia (jeśli dotyczy) , usługodawca)
10. Dla każdego budynku powinniśmy przechowywać informację o budowie (data rozpoczęcia, data zakończenia, opcjonalny opis, lista wykonawców). Dodatkowo dla danej budowy chcemy zapamiętać ocenę wykonawcy (w formie opisowej)
11. System powinien umożliwiać użytkownikom rejestrowanie i zarządzanie informacjami takimi jak lokalizacja, wyposażenie, infrastruktura oraz zasoby.
12. 12 .Użytkownikami systemu mogą być administrator aplikacji, usługodawca, wykonawca.
13. Wykonawca w systemie może wprowadzać informację o budowie oraz je modyfikować.
14. Usługodawca w systemie może wprowadzać informację o usłudze oraz modyfikować jej szczegóły.
15. Po półrocznym braku aktywności usługodawcy jego konto wchodzi w stan dezaktywacji.

System pozwala na wykonanie następujących czynności z podziałek na aktorów systemu:

1. Wykonawca:
   1. Wprowadź informacje o budowie
   2. Modyfikuj informacje o budowie
   3. Sporządź listę wykonanych budynków
   4. Dodaj budynek
   5. Modyfikuj status budynki
2. Usługodawca:
   1. Sporządź listę wykonanych usług
   2. Wprowadź informacje o usłudze
   3. Modyfikuj szczegóły usługi
   4. Modyfikuj status usługodawcy
3. Administrator:
   1. Wszystkie powyższe czynności
   2. Dodaj firmę
   3. Sporządź listę budynków
   4. Sporządź listę wykonanych usług dla danego budynku
4. Czas:
   1. Oznacz nieaktywnych usługodawców

**2.3 Charakterystyka użytkowników**

System zarządzania nieruchomościami będzie obsługiwany przez różne grupy użytkowników, każda z nich z własnym zestawem potrzeb, umiejętności i uprawnień. Poniżej przedstawiono ogólny opis użytkowników oraz występujące między nimi zależności:

* Administrator aplikacji - Osoba odpowiedzialna za konfigurację, zarządzanie i utrzymanie systemu. Administrator ma najwyższy poziom uprawnień i dostępu do wszystkich funkcji systemu, takich jak dodawanie i usuwanie kont użytkowników, zarządzanie danymi, przeglądanie raportów oraz monitorowanie bezpieczeństwa systemu.
* Usługodawca - Firma lub osoba świadcząca usługi związane z nieruchomościami, takie jak utrzymanie, sprzątanie, zarządzanie energią czy inne usługi techniczne. Usługodawca ma dostęp do informacji o swoich usługach, cenach oraz statusie aktywności. Może także przeglądać i aktualizować informacje dotyczące budynków, dla których świadczy usługi.
* Wykonawca - Firma lub osoba odpowiedzialna za prace budowlane, remontowe czy renowacyjne. Wykonawca ma dostęp do informacji na temat budynków, dla których realizuje projekty, oraz danych dotyczących budowy, takich jak terminy, opisy, oceny czy lista wykonawców.

**Występujące zależności między użytkownikami:**

* Administrator aplikacji musi współpracować z usługodawcami i wykonawcami w celu utrzymania aktualnych danych na temat budynków, usług i budów.
* Usługodawcy i wykonawcy są zależni od administratora aplikacji w zakresie uzyskiwania odpowiednich uprawnień oraz dostępu do niezbędnych informacji, które umożliwią im efektywne zarządzanie swoimi usługami i projektami.
* Usługodawcy mogą współpracować z wykonawcami, gdy ich usługi są ściśle powiązane z pracami budowlanymi lub remontowymi. W takich przypadkach obie strony muszą wspólnie aktualizować informacje na temat budynków i usług, aby zapewnić spójność danych.

**2.4 Wymagania niefunkcjonalne (ograniczenia)**

* Organizacyjne: System musi być zintegrowany z istniejącymi systemami CRM, ERP i finansowymi, aby umożliwić efektywną wymianę danych między różnymi modułami.
* Prawne: System musi przestrzegać obowiązujących przepisów dotyczących ochrony danych osobowych, takich jak RODO, oraz spełniać wymogi dotyczące przechowywania i przetwarzania informacji podatkowych i finansowych.
* Architektoniczne: System powinien być oparty na modularnej i elastycznej architekturze, umożliwiającej łatwe wprowadzanie zmian, rozszerzeń oraz integracji z innymi systemami.
* Softwareowe: System powinien być zbudowany z wykorzystaniem nowoczesnych technologii i standardów, takich jak REST API, gRPC czy SSL, aby zapewnić efektywną komunikację między modułami i usługami.
* Sprzętowe: System musi być zoptymalizowany pod kątem wykorzystania zasobów sprzętowych, takich jak CPU, pamięć RAM i przestrzeń dyskowa, aby zapewnić odpowiednią wydajność i skalowalność.
* Bezpieczeństwo: System musi oferować zaawansowane mechanizmy uwierzytelniania, autoryzacji oraz szyfrowania danych, aby chronić informacje przed nieautoryzowanym dostępem, utratą czy modyfikacją.
* Wydajność: System powinien być w stanie obsługiwać dużą liczbę równoczesnych użytkowników i zapytań, zapewniając odpowiedni czas reakcji i minimalizując opóźnienia.
* Niezawodność: System musi gwarantować wysoki poziom dostępności i odporności na awarie, z automatycznymi mechanizmami redundancji, kopii zapasowych i przywracania danych.
* Łatwość użytkowania: Interfejs użytkownika systemu powinien być intuicyjny, ergonomiczny i dostosowany do potrzeb różnych grup użytkowników, z uwzględnieniem dostępności dla osób niepełnosprawnych.
* Standardy: System musi być zgodny z obowiązującymi standardami branżowymi i technologicznymi.

**2.5 Założenia i zależności**

Założenia i zależności mają istotny wpływ na wymagania systemu zarządzania nieruchomościami. Zmiany w założeniach lub zależnościach mogą prowadzić do konieczności modyfikacji wymagań lub przeglądu istniejących rozwiązań. Poniżej przedstawiono przykłady takich sytuacji:

* Zmiany w zakresie platformy programistycznej: Jeśli platforma programistyczna ulegnie zmianie, na przykład związanej z aktualizacją technologii lub migracją na nowy system, może to wpłynąć na wymagania dotyczące integracji, wydajności czy bezpieczeństwa. W takim przypadku konieczne będzie dostosowanie systemu do nowego środowiska oraz ewentualne przeprojektowanie części architektury.
* Zmiany prawa: Jeśli obowiązujące przepisy prawne zostaną zmienione, na przykład w zakresie ochrony danych osobowych, podatków czy norm budowlanych, może to wymagać wprowadzenia nowych funkcjonalności do systemu lub modyfikacji istniejących. W takiej sytuacji konieczne będzie przeglądanie i aktualizowanie wymagań prawnych oraz dostosowanie systemu do nowych regulacji.
* Potrzeby wprowadzenia nowego wymagania użytkownika: W przypadku gdy użytkownicy zgłoszą potrzebę wprowadzenia nowego wymagania lub funkcjonalności, takie jak na przykład narzędzie do zarządzania energią w budynkach, może to wpłynąć na wymagania systemu oraz architekturę. W takiej sytuacji konieczne będzie ocena możliwości wprowadzenia nowej funkcjonalności oraz analiza wpływu na istniejące wymagania i moduły systemu.

Analiza wpływu zmian w założeniach i zależnościach na wymagania systemu pozwala na lepsze zrozumienie ryzyka związanego z projektem oraz elastyczność w dostosowywaniu systemu do zmieniających się warunków i potrzeb użytkowników. Regularne przeglądy założeń i zależności pomagają w utrzymaniu aktualności wymagań i minimalizacji ryzyka związanego z nieoczekiwanymi zmianami w otoczeniu projektu.

**3. Model analityczny**

**3.1 Diagram przypadków użycia**



**3.2 Dokumentacja przypadków użycia:**

|  |  |
| --- | --- |
| **Imię i nazwisko** | **Przypadek użycia** |
| Kamil Paczyński | Wprowadź informacje o budowie, Sporządź listę budynków |
| Kamil Wojas | Wprowadź informację o usłudze, Modyfikuj status usługodawcy |
| Mateusz Laskowski | Modyfikuj status budynku,  Sporządź listę wykonanych budynków |
| Jakub Wrzesień | Dodaj budynek, Dodaj firmę |

**3.2.1. Dodaj budynek**

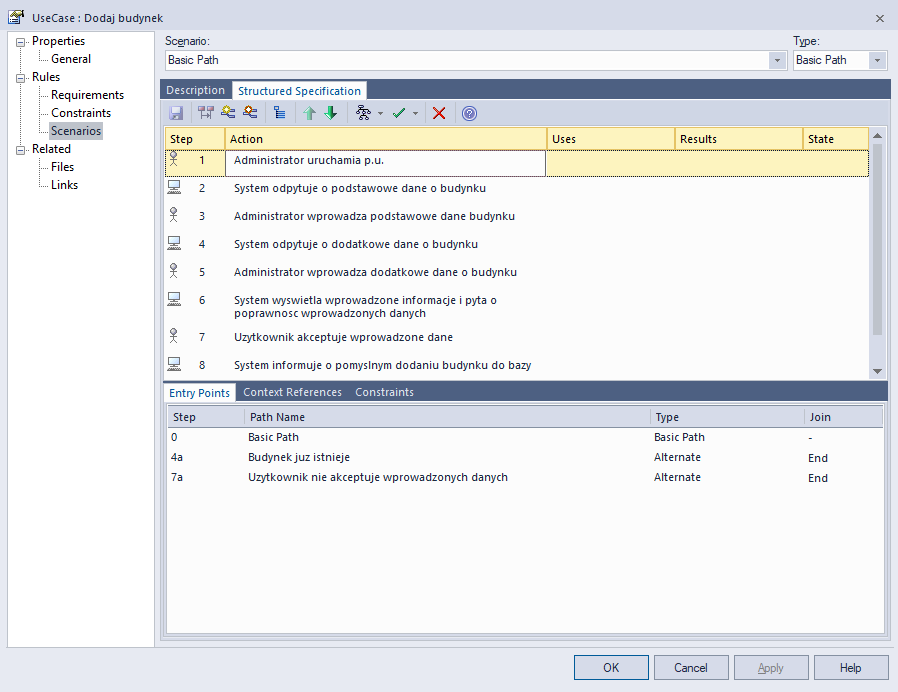
1. Aktorzy: Użytkownik systemu (Wykonawca bądź Administrator).
2. Opis funkcjonalności podstawowej: Użytkownik systemu może dodać nowy budynek do bazy danych aplikacji, który zawiera informacje takie jak: adres, powierzchnia, wysokość, status.
3. Warunki wstępne: Użytkownik systemu musi być zalogowany i mieć uprawnienia do modyfikowania danych budowlanych.
4. Warunki końcowe: Rekord budynku zostaje zapisany w bazie danych aplikacji.
5. Zakończenie: Przypadek użycia kończy się, gdy użytkownik zatwierdzi wprowadzone dane lub anuluje proces wprowadzania nowego budynku.

Wyjątki:

* Dodawany budynek istnieje już w bazie danych aplikacji. Użytkownik jest o tym informowany.
* Wystąpił błąd podczas walidacji informacji - wyświetlany jest komunikat o błędzie.



**Przebieg główny “Dodaj budynek”**



**3.2.2. Dodaj firmę**

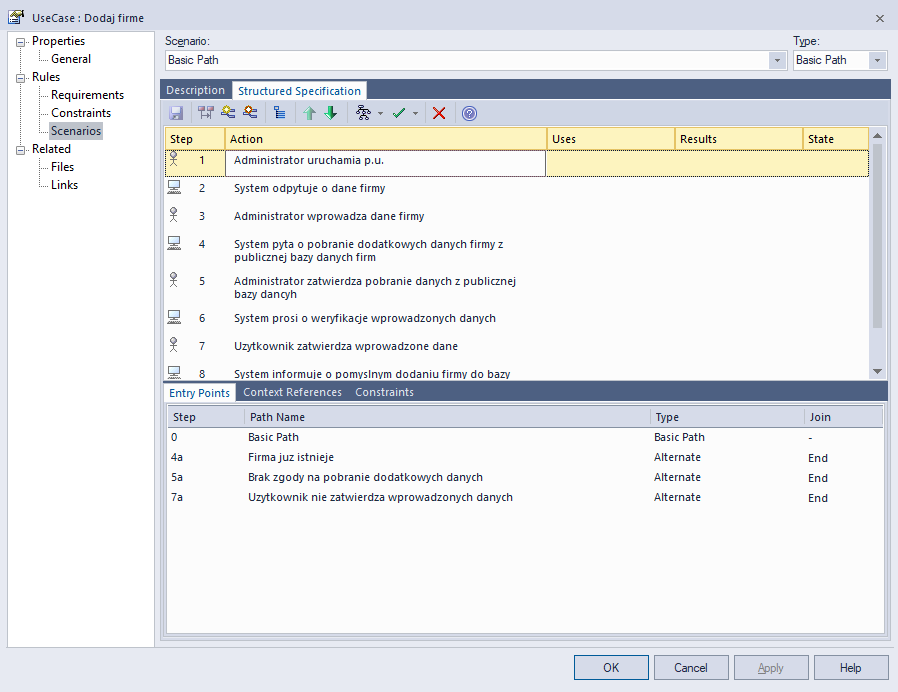
1. Aktorzy: Użytkownik systemu (Administrator)
2. Opis funkcjonalności podstawowej: Administrator może dodać nową firmę do bazy danych aplikacji, która zawiera informacje takie jak: nazwa, NIP, REGON, data początku i zakończenia współpracy.
3. Opis funkcjonalności dodatkowej: W trakcje dodawania nowej firmy można pobrać dodatkowe dane z zewnętrznej publicznej bazy danych o firmach.
4. Warunki wstępne: Użytkownik systemu musi być zalogowany jako Administrator
5. Warunki końcowe: Rekord z danymi nowej firmy zostaje zapisany w bazie danych aplikacji.
6. Zakończenie: Przypadek użycia kończy się, gdy użytkownik zatwierdzi wprowadzone dane lub anuluje proces wprowadzania nowej firmy.

Wyjątki:

* Dodawana firma istnieje już w bazie danych aplikacji. Użytkownik jest o tym informowany.
* Wystąpił błąd podczas walidacji informacji - wyświetlany jest komunikat o błędzie.



**Przebieg główny “Dodaj firmę”**



**3.2.3. Modyfikuj informacje o budowie**

**2a)**

Ogólny opis p.u.:

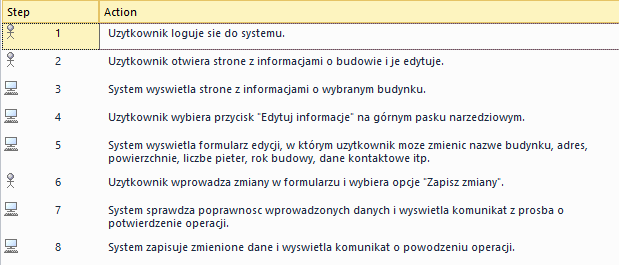
1. Aktorzy: Użytkownik systemu (osoba odpowiedzialna za zarządzanie danymi budowlanymi)
2. Opis funkcjonalności podstawowej: Użytkownik systemu może modyfikować informacje dotyczące budowy, takie jak: nazwa, adres, powierzchnia, liczba pięter itp.
3. Opis funkcjonalności opcjonalnej: Użytkownik systemu może dodać lub usunąć zdjęcia budynku, pliki projektowe, dane techniczne itp.
4. Warunki wstępne: Użytkownik systemu musi być zalogowany i mieć uprawnienia do modyfikowania danych budowlanych.
5. Warunki końcowe: Zmienione informacje o budowie zostaną zapisane w systemie.
6. Zakończenie: Przypadek użycia kończy się, gdy użytkownik zatwierdzi zmiany lub anuluje modyfikacje.

Wyjątki:

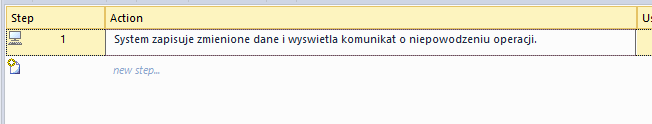
* Użytkownik nie ma uprawnień do modyfikowania danych budowlanych - wyświetlany jest komunikat o braku uprawnień.
* Wystąpił błąd podczas zapisywania zmian - wyświetlany jest komunikat o błędzie.

**2b)**

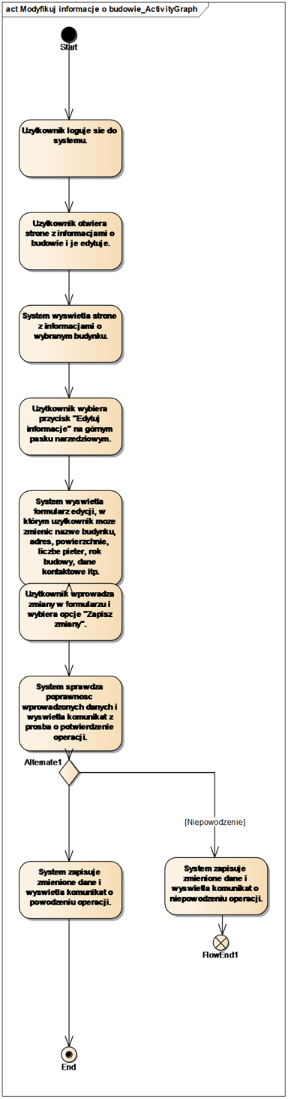
**Przebieg główny “Modyfikuj informacje o budowie”**



**Przebieg alternatywny “System zapisuje dane”**



**2c)**



**3.2.4. Sporządź listę budynków**

**a)** Ogólny opis przypadku użycia:

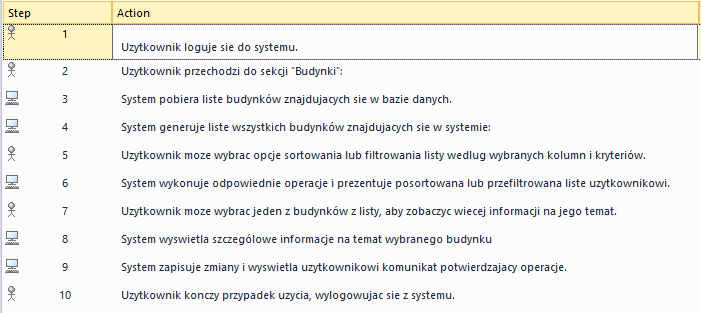
1. Aktorzy: Administrator
2. Do czego służy: Umożliwia użytkownikowi wygenerowanie listy budynków znajdujących się w systemie.
3. Opis funkcjonalności podstawowej: Użytkownik wybiera opcję "Generuj listę budynków" i system generuje listę wszystkich budynków z informacjami takimi jak nazwa, adres, powierzchnia, liczba pięter itp.
4. Opis funkcjonalności opcjonalnej: Użytkownik może sortować listę według wybranej kolumny (np. nazwa, adres) oraz może filtrować listę według wybranych kryteriów (np. powierzchnia większa niż X metrów kwadratowych).
5. Warunki wstępne: Użytkownik musi być zalogowany do systemu.
6. Warunki końcowe: System wyświetla listę budynków lub informację o braku budynków w systemie.
7. Zakończenie: Użytkownik kończy przypadek użycia po wygenerowaniu listy budynków.

Wyjątki:

* Brak budynków w systemie - system wyświetla informację o braku budynków.
* Błędne dane wejściowe - system wyświetla komunikat z informacją o błędzie i prosi użytkownika o wprowadzenie poprawnych danych.

**b)**

**Przebieg główny “Sporządź liste budynków”**



**Przebieg alternatywny “System wyświetla komunikat o błędzie”**

Inserting image...

**Przebieg alternatywny “System wyświetla informacje o braku budynków”**



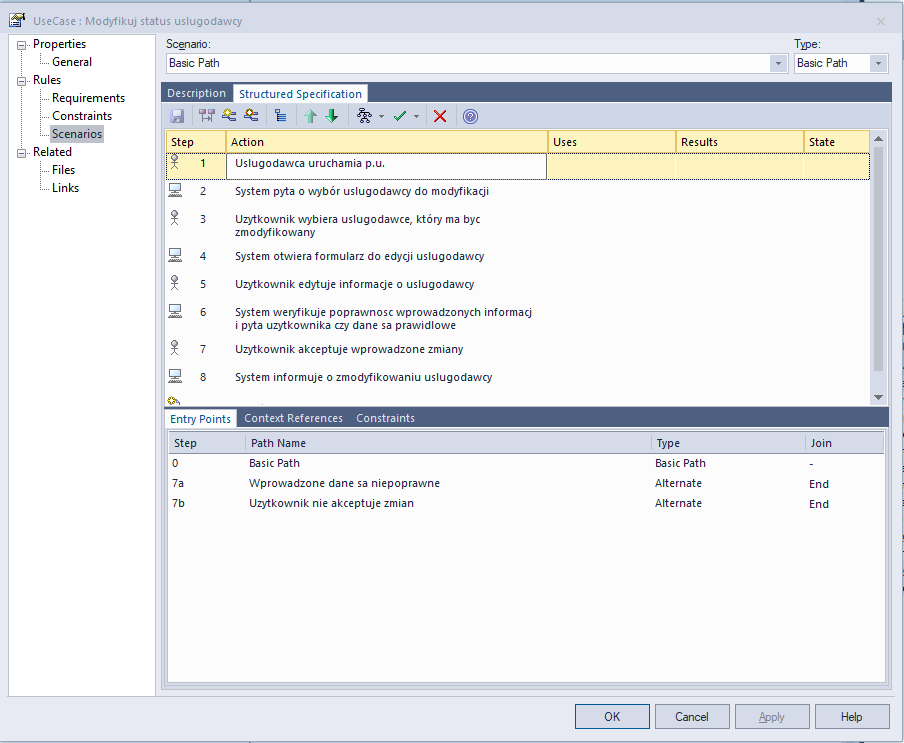
**c)**

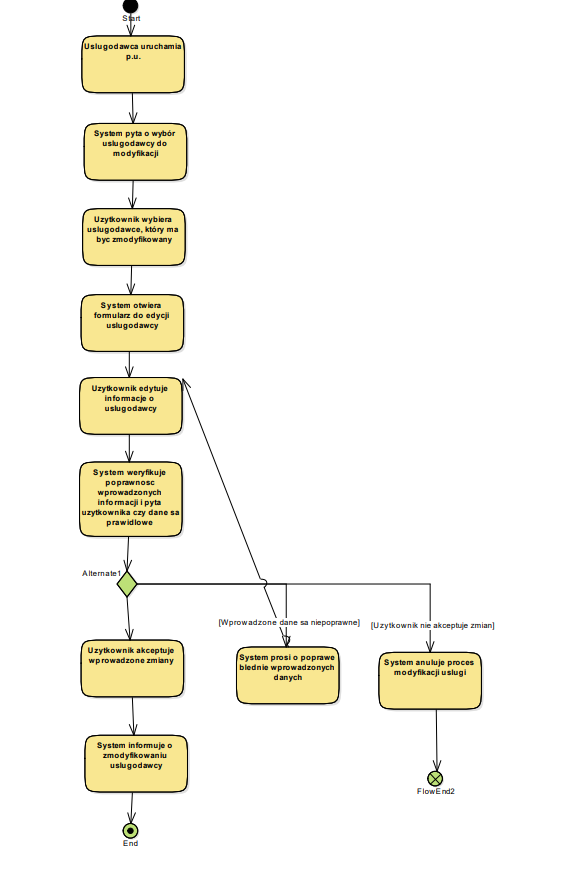


**3.2.5 Modyfikacja status uslugodawcy**

1. Ogólny opis przypadku użycia
   * Aktorzy – Administrator
   * Do czego służy - Służy do umozliwienia użytkownikowi zmiany informacji o usługodawcy, takich jak nazwa, adres czy nr telefonu.
   * Opis funkcjonalności podstawowej - Użytkownik może wybrać usługodawcę z listy, których informacje chce zmodyfikować. System otwiera formularz, w którym użytkownik może dokonać edycji informacji na temat wybranego usługodawcy. Po wprowadzeniu zmian, system weryfikuje poprawność wprowadzonych informacji I pyta użytkownika, czy chce zachować wprowadzone zmiany. Jeśli uytkownik akceptuje zmiany, system aktualizuje informacje o usługodawcy I informuje użytkownika o powodzeniu operacji.
   * Opis funkcjonalności opcjonalnej – W momncie, gdy użytkownik wprowadza zmiany w danych usługodawcy, system może porównać je z dotychczasowymi danymi i wyświetlić ostrzeżenie, jeśli wprowadzone zmiany są zbyt duże lub istnieją konflikty.
   * Warunki wstępne - Dostęp do formularza edycji. Wybur usługodawcy, którego chcemy zmodyfikować. Dostęp do systemu, który obsługuje dane usługodawcy. Użytkownik musi być zalogowany.
   * Warunek końcowy - Zapisanie zmodyfikowanych informacji o usługodawcy i poinformowanie użytkownika o usługodawcy i poinformowanie użytkownika o poprawnym wykonaniu operacji.
   * Zakończenie - Przypadek użycia kończy się w momencie, gdy użytkownik zaakceptuje wprowadzone zmiany, a system poinformuje go o zmodyfikowaniu usługodawcy.
   * Wyjątki - Błąd wczytywania listy dostępnych usługodawców. Wybór nieprawidłowego usługodawcy. Błąd podczas zapisywania zmian. Błąd weryfikacji wprowadzonych danych.

**b)** Przebieg glowny – Modyfikacji statusu uslugodawcy





**3.2.6 Modyfikuj status budynku**

a) Ogólny opis przypadku użycia "Modifikuj status budynku":

Aktorzy:

* Administrator systemu
* Użytkownik systemu

Funkcjonalność podstawowa:

* Zmiana statusu budynku (np. aktywny, nieaktywny, w trakcie remontu)

Funkcjonalność opcjonalna:

* Dodawanie komentarzy dotyczących zmiany statusu
* Wysyłanie powiadomień do zainteresowanych stron o zmianie statusu

Warunki wstępne:

* Użytkownik musi być zalogowany do systemu EA Sparks
* Użytkownik musi mieć uprawnienia do modyfikacji statusu budynku

Warunki końcowe:

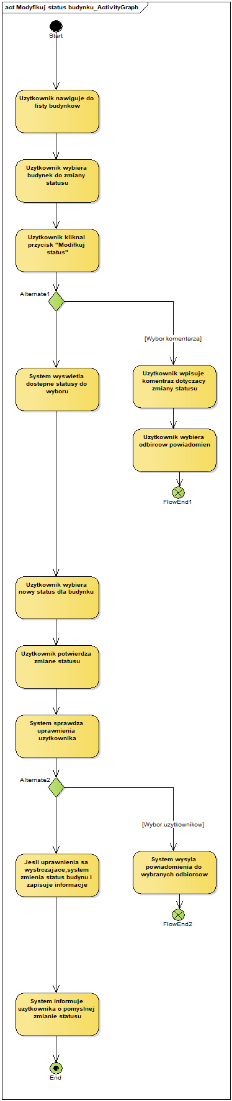
* Status budynku zostaje zmieniony
* Opcjonalnie: Komentarz do zmiany statusu zostaje dodany
* Opcjonalnie: Powiadomienia zostają wysłane do zainteresowanych stron

Zakończenie:

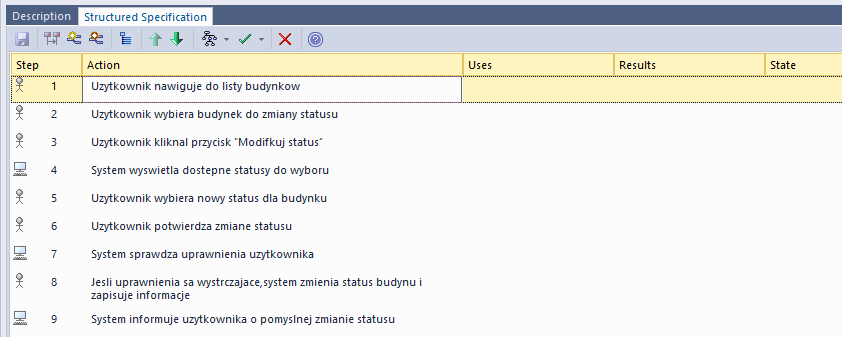
* W dowolnym momencie (przykładowo, gdy użytkownik zdecyduje się anulować zmianę statusu)
* W punktach określonych w scenariuszu (np. po potwierdzeniu zmiany statusu)

Wyjątki:

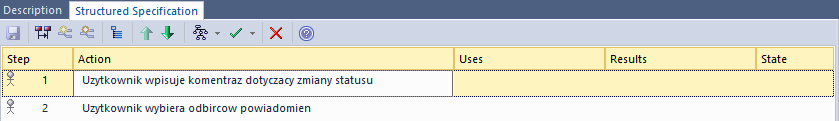
* Błąd systemu
* Brak uprawnień użytkownika do zmiany statusu budynku



**Przebieg glowny “Modifikuj status budynku”**



**Przebieg alternatywny “Wybor komentarza”**



**Przebieg alternatywny ”Wybor uzytkownikow”**



**3.2.7 Sporzad liste wykonanych budynkow**

Aktorzy:

* Administrator systemu
* Użytkownik systemu

Funkcjonalność podstawowa:

* Wyświetlanie listy wykonanych budynków (np. ukończonych, oddanych do użytku)

Funkcjonalność opcjonalna:

* Filtrowanie listy według kryteriów (np. lokalizacja, data ukończenia, inwestor)
* Eksportowanie listy do pliku (np. CSV, PDF)

Warunki wstępne:

* Użytkownik musi być zalogowany do systemu EA Sparks
* Użytkownik musi mieć uprawnienia do przeglądania listy wykonanych budynków

Warunki końcowe:

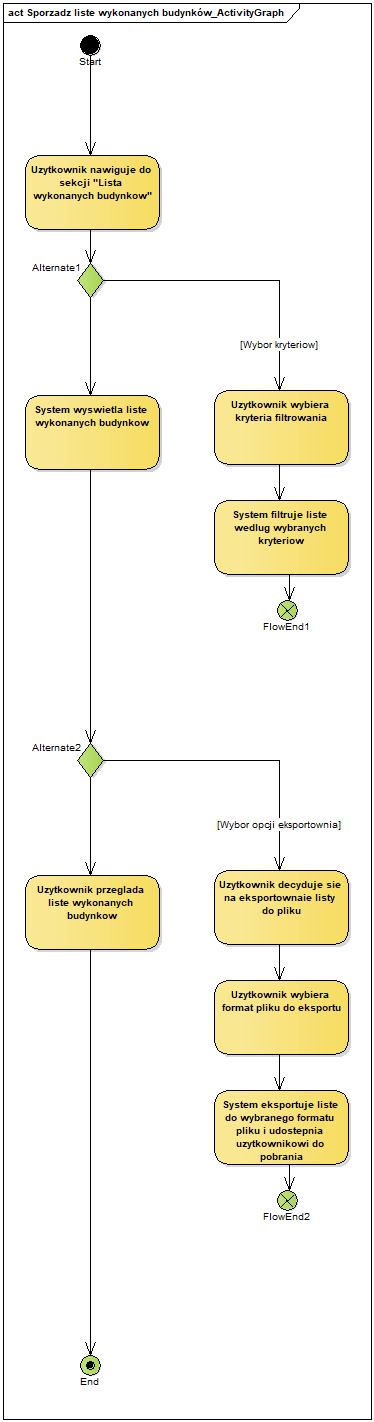
* Lista wykonanych budynków jest wyświetlana użytkownikowi
* Opcjonalnie: Lista jest filtrowana według kryteriów
* Opcjonalnie: Lista jest eksportowana do pliku

Zakończenie:

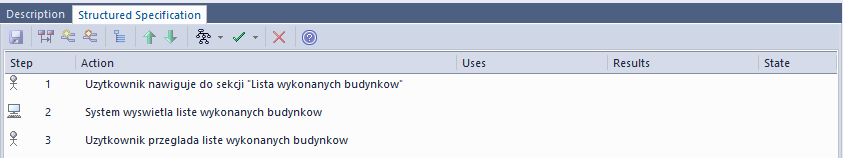
* W dowolnym momencie (np. gdy użytkownik opuści stronę z listą)
* W punktach określonych w scenariuszu (np. po zakończeniu przeglądania listy)

Wyjątki:

* Błąd systemu
* Brak uprawnień użytkownika do przeglądania listy wykonanych budynków



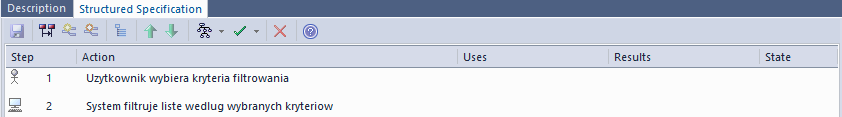
**Przebieg glowny “Sporzad liste wykonanych budynkow”**



**Przebieg alternatywny “wybor kryteriow”**



**Przebieg alternatywny “Wybor opcji eksportowania”**



**3.3**

Charakterystyka ograniczenia, opis sposobu weryfikacji - najlepiej z wykorzystaniem metryki (definicja - opis wymagania niefunkcjonalnego, sposób przeprowadzenia pomiaru lub oszacowania, jednostka pomiaru, interpretacja pomiaru) .

**Organizacyjne**:

* Charakterystyka ograniczenia: System musi być zintegrowany z istniejącymi systemami CRM, ERP i finansowymi, aby umożliwić efektywną wymianę danych między różnymi modułami.
* Sposoby weryfikacji: Przeprowadzanie testów integracyjnych, weryfikacja poprawności i kompletności wymienianych danych między systemami. Metryka może być oparta na liczbie przesłanych i odebranych komunikatów, czasie reakcji systemu zapytania, itd.

**Prawne**:

* Charakterystyka ograniczania: System musi przestrzegać obowiązujących przepisów dotyczących ochrony danych osobowych , takich jak RODO, oraz spełniać wymogi dotyczące przechowywania i przetwarzania informacji podatkowych i finansowych.
* Sposób weryfikacji: Przeprowadzanie audytu zgodności systemu z obowiązującymi przepisami oraz wdrożenie mechanizmów ochrony danych osobowych, takich jak szyfrowanie, uwierzytelnianie, autoryzacje, itp. Metryka może być oparta na liczbie błędów lub naruszen, które zostały wykryte podczas audytu lub kontroli.

**Architektoniczne**:

* Charakterystyka ograniczenia: System powinien być oparty na modularnej i elastycznej architekturze, umożliwiającej łatwe wprowadzanie zmian, rozszerzen oraz integracji z innymi systemami.
* Sposób weryfikacji: Przeprowadzenie analizy architektury systemu, testów jednostkowych, integracyjnych i wydajnościowych oraz weryfikacja zgodności systemu z obowiązującymi standardami branżowymi i technologicznymi. Metryka może być oparta na czasie wprowadzania zmian w systemie, liczbie błędów w czasie wprowadzania zmian oraz na liczbie integracji z innymi systemami.

**Softwareowe**:

* Charakterystyka ograniczenia: System powinien być zbudowany z wykorzystaniem nowoczesnych technologii i standardów, takich jak REST API, gRPC czy SSL, aby zapewnić efektowną komunikację między modułami i usługami.
* Sposoby weryfikacji: Analiza kodu źródłowego, testy jednostkowe i integracyjne, weryfikacje zgodności systemu z obowiązującymi standardami technicznymi. Metryka może być oparta na liczbie błędów w kodzie, czasie odpowiedzi systemu na zapytania lub czasie przetwarzania danych.

**Sprzętowe**:

* Charakterystyka ograniczenia: System musi być zoptymalizowany pod kątem wykorzystania zasobów sprzętowych, takich jak CPU, pamięć RAM i przestrzen dyskowa, aby zapewnić odpowiednią wydajność i skalowalność.
* Opis sposobu weryfikacji: Weryfikacja spełnienia wymaga sprzętowych może być przeprowadzona za pomocą narzędzi diagnostycznych, takich jak oprogramowanie do monitorowania wykorzystania zasobów systemowych. Jednostką pomiaru może być zużycie zasobów CPU, zużycie pamięci RAM lub wykorzystanie przestrzeni dyskowej w systemie.
* W przypadku testów wydajnościowych, jednostką pomiaru może być np. liczba zapytan obsłużonych przez system w jednostce czasu lub czas odpowiedzi systemu na konkretne żądania.
* Interpretacja wyników pomiaru powinna obejmować ocenę, czy system spełnia wymagania sprzętowe, czy też nie. W przypadku niespełnienia wymagan, należy określić poziom dla użytkownika systemu, takich jak spowolnienie lub niestabilność systemu, oraz zidentyfikować przyczyny niedoskonałości, takie jak niedostateczna ilość pamięci RAM lub nieodpowiedni typ dysku twardego.
* Sposób weryfikacji: Aby sprawdzić, czy system spełnia wymagania dotyczące bezpieczenstwa, należy przeprowadzić testy penetracyjne, aby wykryć ewentualne luki w zabezpieczeniach. Należy również przeprowadzić testy obciążeniowe, aby sprawdzić, czy system jest odporny na ataki typu DoS i czy jest w stanie obsłużyć dużą liczbę równoczesnych zapytan.
* Metryka: W przypadku testów penetracyjnych, jednostką pomiaru może być liczba wykrytych luk w zabezpieczeniach na jednostkę czasu lub jednostkę objętości kodu. W przypadku testów obciążeniowych, jednostką pomiaru może być średni czas odpowiedzi systemu na jednostkę zapyta lub liczba zapyta obsłużonych na jednostkę czasu przy określonym obciążeniu. Interpretacja wyników powinna polegać na określeniu, czy wyniki spełniają wymagania określone dla danego wymagania niefunkcjonalnego.
* Łatwość użytkowania: Aby zweryfikować, czy system jest łatwy w obsłudze, należy przeprowadzić testy użyteczności, w których użytkownicy będą musieli wykonywać zadania związane z danymi funkcjonalnościami systemu. Należy także zbierać opinie i sugestie od użytkowników w celu oceny ergonomii interfejsu.
* Metryka: Jednostką pomiaru w testach użyteczności może być czas wykonania zadania, liczba popełnionych błędów, liczba kliknięć, liczba zmian ekranu itp. Można też wykorzystać skalę oceny zadowolenia użytkowników z systemu, taką jak System Usability Scale. Interpretacja wyników powinna polegać na określeniu, czy wyniki spełniają wymagania określone dla danego wymagania niefunkcjonalnego.
* Standardy: Aby sprawdzić, czy system jest zgodny z obowiązującymi standardami branżowymi i technologicznymi, należy dokonać weryfikacji na podstawie dokumentacji technicznej i przepisów prawnych. Można również przeprowadzić testy, aby upewnić się, że system spełnia wymagania stawiane przez standardy.
* Metryka: W przypadku weryfikacji na podstawie dokumentacji, jednostką pomiaru może być liczba odniesie do konkretnych wymagan standardów lub liczba stwierdzonych niezgodności z danym standardem. W przypadku testów, jednostką pomiaru może być liczba testów spełniających wymagania standardów lub liczba testów, w których stwierdzono niezgodność z danym standardem. Interpretacja wyników powinna polegać na porównaniu uzyskanych wyników z wymaganiami określonymi w standardach oraz na ocenie, czy system spełnia te wymagania. Przykładowo, w przypadku testów wydajnościowych, jednostką pomiaru może być liczba zapyta obsłużonych w określonym czasie przez system przy użyciu określonej ilości zasobów sprzętowych. Można również mierzyć zużycie pamięci, CPU i przestrzeni dyskowej przez system podczas wykonywania różnych zadan. Interpretacja wyników może polegać na porównaniu uzyskanych wyników z minimalnymi wymaganiami sprzętowymi oraz na ocenie, czy system jest zoptymalizowany i działa z odpowiednią wydajnością przy minimalnym zużyciu zasobów.

**3.4 Diagram klas**



**4 Model GUI**

**4.1 Projekt GUI ekranu startowego aplikacji**

****

**4.1.1 Projekty ekranów głównych dla wszystkich aktorów osobowych**

**Wykonawca**

****

**Administrator**

****

**Usługodawca**

****

**4.2 Wykonanie projektu GUI dla wybranego jednego z przypadku użycia**

**4.2.1. Dodaj budynek**



**4.2.2. Dodaj firmę**



**4.2.3. Modyfikuj informacje o budowie4.2.4 Sporządzanie listy budynków**

**4.2.5.Modyfikuj status usługodawcy**

**4.2.6. Modyfikuj status budynku**

**4.2.7. Sporządź listę wykonanych budynków**

**4.2.8 Informacje o usługach**

