

## **1.Wstęp, wymagania funkcjonalne i нефункционалне**

### **Wstęp:**

Prężnie rozwijające się przedsiębiorstwo "JanuszWare" prowadzone przez Janusza Pracza które zajmuje się naprawą komputerów, poprosiło naszą grupę o napisanie aplikacji wspomagającej zarządzanie pracą serwisantów. Poprzez natłok zgłoszeń o psujących się komputerach i zwiększenie kadry pracowniczej, organizacja w firmie znacząco zmalała. Nasz program ma za zadanie poprawić organizację pracy w przedsiębiorstwie i usprawnić zarządzanie firmą.

### **Wymagania funkcjonalne:**

- Logowanie.
- <cru> pracowników.
- <cru> naprawy
- google maps api do lokalizacji klientów.
- wyliczanie premii.
- <crud> magazynu części komputerowych.
- <cru> klientów.
- autouzupełnianie pól podczas dodawania naprawy.
- wyświetlanie braków w magazynie.
- generowanie raportów dnia

### **Wymagania нефункционалне:**

- Hasło użytkowników musi być odpowiednio silne (8 znaków, 1 duża litera, jeden znaczek alfanumeryczny)
- niezawodność.
- system działa na środowisku Windows.
- Interfejs musi być prosty i intuicyjny

## 2.Opis aktorów

**Administrator** - Osoba odpowiedzialna za zarządzanie bazą danych (dodawanie nowych pracowników itp.)

**Sekretarka** - Osoba odpowiedzialna za dodawanie zgłoszeń do systemu, ma również możliwość modyfikacji danych klienta.

**Serwisant** - Jest to osoba, która odbiera dodane zgłoszenia i naprawia komputery. Ma możliwość wyjmowania produktów do magazynu i wkładania produktów do magazynu. System udostępnia dla serwisanta lokalizację klienta na mapie.

**Szef** - Jest to przedsiębiorca zarządzający firmą, ma takie same możliwości co serwisant, ale dodatkowo może zażądać magazynem.

**System** - generuje raport dnia i wylicza premie.

### **3.Opis przypadków użycia**

#### **Opis przypadku użycia "Logowanie":**

1. Programista: Damian Bajno

2. Aktorzy:

- Szef
- Serwisant
- Sekretarka
- Administrator

3. Krótki opis:

- Logowanie się do systemu.

4. Warunki wstępne:

- Osoba musi istnieć w systemie

5. Warunki końcowe:

- System odnotowuje osobę jako zalogowaną.

6. Główny przepływ zdarzeń:

- A) Osoba wypełnia formularz danymi niezbędnymi do logowania.
- B) Osoba akceptuje formularz.
- C) Poprawnie zalogowano do systemu, osoba zostanie przeniesiona do panelu odpowiadającego jej uprawnieniom.

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o błędnych danych.
- Osoba anuluje logowanie.
- Błąd podczas logowania.

### **Opis przypadku użycia "Dodaj zgłoszenie":**

1. Programista: Damian Bajno

2. Aktorzy:

- Sekretarka

3. Krótki opis:

- Dodawanie zgłoszenia do systemu.

4. Warunki wstępne:

- Zgłoszenie awarii przez klienta telefonicznie lub w biurze.

5. Warunki końcowe:

- System zapisuje zgłoszenie w systemie.

6. Główny przepływ zdarzeń:

A) Sekretarka wypełnia formularz danymi potrzebnymi do dodania zgłoszenia.

B) Sekretarka akceptuje wypełniony formularz.

C) Zgłoszenie zostaje poprawnie dodane do systemu, sekretarka zostanie przeniesiona do listy zgłoszeń.

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o błędnych danych.
- Sekretarka anuluje dodawanie zgłoszenia.
- Błąd podczas dodawania zgłoszenia.

### **Opis przypadku użycia "Zakończ zgłoszenie":**

1. Programista: Damian Bajno

2. Aktorzy:

- Serwisant

3. Krótki opis:

- Status zgłoszenia zmienia się na "ukończone".

4. Warunki wstępne:

- Zgłoszenie musi istnieć.
- Zgłoszenie musi być wykonane.

5. Warunki końcowe:

- System odnotowuje zmianę statusu zgłoszenia.

6. Główny przepływ zdarzeń:

- A) Serwisant wybiera zgłoszenie z listy odebranych przez niego zgłoszeń i zatwierdza przyciskiem jego ukończenie
- B) Pojawia się formularz będący raportem, który należy uzupełnić by zakończyć zgłoszenie.
- C) Poprawnie zmieniono status zgłoszenia, zmiana zostaje odnotowana w systemie.

7. Alternatywny przepływ zdarzeń:

- Osoba cofa zmianę statusu zgłoszenia.
- Osoba wprowadziła błędne dane o czym informuje komunikat.
- Błąd podczas zmiany statusu zgłoszenia.

### **Opis przypadku użycia "Odbierz wiadomość":**

1. Programista: Damian Bajno

2. Aktorzy:

- Szef

3. Krótki opis:

- Odbieranie wiadomości od użytkowników systemu.

4. Warunki wstępne:

- Skrzynka pocztowa nie może być pusta.

5. Warunki końcowe:

- Odczyt wiadomości przesłanych przez użytkowników.

6. Główny przepływ zdarzeń:

- A) Szef klika przycisk "Skrzynka pocztowa".
- B) Szef dokonuje wyboru wiadomości, które chce odczytać poprzez kliknięcie w nią.
- C) Wiadomość zostaje oznaczona jako przeczytana, a informacja o tym zapisana w systemie.

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o braku nowych wiadomości.
- Błąd podczas wczytywania wiadomości.

### **Opis przypadku użycia "Wyświetl braki":**

1. Programista: Damian Bajno

2. Aktorzy:

- Szef
- Serwisant

3. Krótki opis:

- System wyświetla części, których brakuje w magazynie.

4. Warunki wstępne:

- Brak danej części w magazynie.

5. Warunki końcowe:

- System reaguje na polecenie i wyświetla wynik rządanej operacji.

6. Główny przepływ zdarzeń:

A) Osoba klika przycisk w panelu "Wyświetl braki"

B) System w wyniku tej operacji zwraca okno z częściami, których brak w magazynie.

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o braku braków.
- Błąd systemu podczas wykonywania żądanych operacji .

### **Opis przypadku użycia "Wypłać premie":**

1. Programista: Damian Bajno

2. Aktorzy:

- Szef

3. Krótki opis:

- Przyznanie dodatkowego wynagrodzenia do stałej pensji pracownika.

4. Warunki wstępne:

- Minimalna ilość ukończonych zgłoszeń musi wynosić 45.

5. Warunki końcowe:

- System zapisuje informacje o przyznanej premii.

6. Główny przepływ zdarzeń:

A) Szef klika przycisk "Przyznaj premie", system wyświetla komunikat "Czy chcesz potwierdzić operację?" oczekując na zatwierdzenie.

B) Premia jest wyliczana przez system na podstawie algorytmu.

C) System dolicza premie do wypłaty Serwisanta.

7. Alternatywny przepływ zdarzeń:

- Szef odrzuca komunikat generowany przez system i anuluje przyznanie premii.
- Błąd podczas wykonywania algorytmu o czym poinformuje komunikat.
- Błąd w trakcie egzekwowania operacji.



### **Opis przypadku użycia "Sprawdź magazyn":**

1. Programista: Damian Bajno

2. Aktorzy:

- Szef
- Serwisant

3. Krótki opis:

- Wyświetlenie aktualnego stanu magazynu.

4. Warunki wstępne:

- Magazyn musi istnieć.

5. Warunki końcowe:

- System egzekwuje polecenie i zwraca jego wynik na ekran.

6. Główny przepływ zdarzeń:

A) Osoba zatwierdza przycisk "Sprawdź magazyn" kliknięciem.

B) System wyświetla w oknie części znajdujące się w magazynie oraz ich liczbę.

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o pustym magazynie.
- Błąd podczas wykonywania operacji.

### **Opis przypadku użycia "Dodaj klienta":**

1. Programista: Damian Bajno

2. Aktorzy:

- Sekretarka

3. Krótki opis:

- Dodawanie klienta do systemu.

4. Warunki wstępne:

- Klient posiadać wymagane dane.

5. Warunki końcowe:

- System dodaje dane klienta do bazy danych.

6. Główny przepływ zdarzeń:

A) Sekretarka wypełnia formularz danymi uzyskanymi od klienta i potrzebnymi do jego rejestracji.

B) Sekretarka akceptuje formularz.

C) Poprawnie dodano do systemu, Sekretarka zostanie przeniesiona do listy klientów.

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o błędnych danych.
- Sekretarka anuluje dodawanie klienta.
- Błąd podczas dodawania nowego klienta do bazy danych.

### **Opis przypadku użycia "Wyślij wiadomość":**

1. Programista: Patryk Basa

2. Aktorzy:

- Serwisant
- Sekretarka

3. Krótki opis:

- Wysyłanie wiadomości do szefa.

4. Warunki wstępne:

- Osoba musi być zalogowana w systemie.

5. Warunki końcowe:

- Informacja zostanie wysłana do szefa.

6. Główny przepływ zdarzeń:

A) Pracownik uzupełnia pole wiadomości.

B) Po akceptacji formularza wyświetli się wiadomość o poprawnym wysłaniu wiadomości

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o błędzie podczas wysyłania wiadomości.
- Pracownik anulował wysyłanie wiadomości.
- Wskoczył błąd o próbie wysłania pustej wiadomości.

### **Opis przypadku użycia "Aktualizuj dane klienta":**

1. Programista: Patryk Basa

2. Aktorzy:

- Sekretarka

3. Krótki opis:

- Zmiana danych klienta w bazie danych.

4. Warunki wstępne:

- Osoba musi być zalogowana w systemie.
- W bazie danych musi istnieć klient

5. Warunki końcowe:

- Dane klienta zostaną zmienione

6. Główny przepływ zdarzeń:

A) Sekretarka wybiera Klienta którego dane chce zmienić

B) Zostanie wyświetlony formularz z starymi danymi

C) Po zatwierdzeniu formularza wyświetla się komunikat o poprawności zmianie danych

7. Alternatywny przepływ zdarzeń:

- System wyświetla komunikat o błędzie podczas zmiany danych.
- Sekretarka anulowała zmianę danych.
- Wyskoczył błąd o próbie zaakceptowania błędnie uzupełnionego formularza.

### **Opis przypadku użycia "Generuj raport dnia":**

1. Programista: Patryk Basa

2. Aktorzy:

- System

3. Krótki opis:

- Codziennie po zadanej godzinie zostanie wygenerowany raport

4. Warunki wstępne:

- Nie istnieje raport danego dnia

5. Warunki końcowe:

- Raport dnia zostaje dodany do bazy danych

6. Główny przepływ zdarzeń:

A) Zawsze po zadanej godzinie system generuje raport dnia.

7. Alternatywny przepływ zdarzeń:

- Gdy system nie będzie miał połączenia z bazą, będzie oczekiwał na generowanie raportu do ponownego połączenia się z bazą danych.

### **Opis przypadku użycia "Lokalizuj klienta":**

1. Programista: Patryk Basa

2. Aktorzy:

- Szef
- Serwisant

3. Krótki opis:

- Na mapie zostanie wyświetlona dom klienta.

4. Warunki wstępne:

- Klient musi być w systemie

5. Warunki końcowe:

- Zostanie wyświetlona lokalizacja klienta na mapie

6. Główny przepływ zdarzeń:

A) Serwisujący wybiera klienta którego chce zlokalizować

B) Wyświetla się mapa wraz z lokalizacją klienta

7. Alternatywny przepływ zdarzeń:

- Zostaje wyświetlony błąd o braku połączenia z internetem.

### **Opis przypadku użycia "Lokalizuj zgłoszenia":**

1. Programista: Patryk Basa

2. Aktorzy:

- Szef
- Serwisant

3. Krótki opis:

- Na mapie zostaną wyświetlone lokalizacje zgłoszeń.

4. Warunki wstępne:

- Musi istnieć co najmniej 1 nieodebrane zgłoszenie

5. Warunki końcowe:

- Na mapie zostaną wyświetlone lokalizacje zgłoszeń.

6. Główny przepływ zdarzeń:

A) zostanie wyświetlona mapa oraz lista zgłoszeń które będzie mógł odebrać.

7. Alternatywny przepływ zdarzeń:

- Zostaje wyświetlony błąd o braku połączenia z internetem.
- Zostaje wyświetlony błąd o braku zgłoszeń.

### **Opis przypadku użycia "Odbierz zgłoszenia":**

1. Programista: Patryk Basa

2. Aktorzy:

- Szef
- Serwisant

3. Krótki opis:

- Serwisant może wybrać zgłoszenia które chce wykonać

4. Warunki wstępne:

- Musi istnieć co najmniej 1 nieodebrane zgłoszenie

5. Warunki końcowe:

- Zgłoszenia zmieniają status na odebrane

6. Główny przepływ zdarzeń:

A) Zostaje wyświetlona lista zgłoszeń które może odebrać.

B) Gdy serwisant wybierze zgłoszenia, wyświetli się komunikat potwierdzający.

C) Wyświetli się komunikat o poprawnym odebraniu zgłoszeń

7. Alternatywny przepływ zdarzeń:

- Serwisant anuluje wybór zgłoszeń
- Zostanie wyświetlony błąd o braku zgłoszeń



### **Opis przypadku użycia "Uzupełnij pola klienta":**

1. Programista: Patryk Basa

2. Aktorzy:

- Sekretarka

3. Krótki opis:

- Zostaną przepisane dane klienta z bazy do formularza zgłoszeniowego.

4. Warunki wstępne:

- Klient musi istnieć w systemie

5. Warunki końcowe:

- Pola zostaną uzupełnione

6. Główny przepływ zdarzeń:

A) Sekretarka wpisując dane klienta dostaje podpowiedzi co do danych klienta.

B) Sekretarka może przepisać dane w formularz z bazy danych.

7. Alternatywny przepływ zdarzeń:

- Brak klienta w bazie danych.