

# Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **System nadzorujący pracę lotniska w Goleniowie**  
Autorzy: **Maciej Doktor i Filip Antkowiak**  
Grupa: **I1-21A**  
Kierunek: **informatyka**  
Rok akademicki: **2019/20**  
Poziom i semestr: **I/4**  
Tryb studiów: **stacjonarne**

## Harmonogram prac nad prototypem:

<b>1</b>	02.03.2020	tworzenie planu
<b>2</b>	09.03.2020	tworzenie planu
<b>3</b>	23.03.2020	4
<b>4</b>	30.03.2020	5
<b>5</b>	06.04.2020	5
<b>6</b>	20.04.2020	6
<b>7</b>	27.04.2020	3 i 7
<b>8</b>	04.05.2020	8
<b>9</b>	11.05.2020	9
<b>10</b>	18.05.2020	10 i 11
<b>11</b>	25.05.2020	implementacja
<b>12</b>	01.06.2020	implementacja
<b>13</b>	15.06.2020	implementacja
<b>14</b>	22.06.2020	Prezentacja prac

# 1 Spis treści

2	Odnośniki do innych źródeł.....	3
3	Słownik pojęć .....	4
4	Wprowadzenie .....	5
4.1	Cel dokumentacji.....	5
4.2	Przeznaczenie dokumentacji .....	5
4.3	Opis organizacji lub analiza rynku .....	5
4.4	Analiza SWOT organizacji.....	5
5	Specyfikacja wymagań .....	5
5.1	Charakterystyka ogólna .....	6
5.2	Wymagania funkcjonalne .....	6
5.3	Wymagania niefunkcjonalne .....	12
6	Zarządzanie projektem .....	18
6.1	Zasoby ludzkie .....	19
6.2	Harmonogram prac .....	19
6.3	Etapy/kamienie milowe projektu.....	19
7	Zarządzanie ryzykiem .....	20
7.1	Lista czynników ryzyka.....	20
7.2	Ocena ryzyka.....	20
7.3	Plan reakcji na ryzyko .....	20
8	Zarządzanie jakością .....	21
8.1	Scenariusze i przypadki testowe.....	21
9	Projekt techniczny .....	22
9.1	Opis architektury systemu .....	25
9.2	Technologie implementacji systemu .....	25
9.3	Diagramy UML .....	25
9.4	Charakterystyka zastosowanych wzorców projektowych .....	28
9.5	Projekt bazy danych .....	29
9.6	Projekt interfejsu użytkownika.....	30
9.7	Procedura wdrożenia .....	30
10	Podsumowanie.....	35
10.1	Szczegółowe nakłady projektowe członków zespołu.....	35

## **2 Odnośniki do innych źródeł**

- Zarządzania projektem – sugerowane JazzHub
- Wersjonowanie kodu – sugerowany Git (hosting np. na Bitbucket lub Github), ew. SVN
- System obsługi defektów – np. Bitbucket, JazzHub

### 3 Słownik pojęć

Słowo	Pojęcie
Pracownicy Portu Lotniczego	Obsługa informująca, sprawdzająca bilety
Załoga samolotów	Piloci, stewardessy
protokół HTTP	Przekazywanie informacji między serwerem, a komputerem
MD5()	Algorytm mieszający, który szyfruje dane przekazane do bazy danych
Użytkownicy	Administracja lotniska, obsługa odpowiedzialna za obsługę bagażu, sprawdzanie biletów oraz odprawę oraz pasażerowie
int	Jest to typ danych, który dotyczy liczb całkowitych
varchar	Jest to typ danych, który przyjmuje tekst do bazy danych
GUI	Interfejs użytkownika

## 4 Wprowadzenie

### 4.1 Cel dokumentacji

Dokumentacja jest przeznaczona do poprawnej obsługi systemu. Dokumentacja zawiera opis oraz diagramy poszczególnych podsystemów. Opisuje dla kogo i w jakim stopniu jest przeznaczona oraz ewentualne scenariusze jakie ma dany podsystem.

### 4.2 Przeznaczenie dokumentacji

Dokumentacja przeznaczona jest dla przyszłych użytkowników systemu, aby był prawidłowo oraz w pełni użytkowany.

### 4.3 Opis organizacji lub analiza rynku

Organizacją dla której przeznaczony będzie ten system jest Port Lotniczy Szczecin-Goleniów im. NSZZ „Solidarność”. Organizacja ta zajmuje się transportem powietrznym towarów oraz ludzi. Planowana data wdrożenia systemu wyznaczona jest na drugi tydzień września (przybliżona dokładna data 10.09.2020). Głównie te lotnisko świadczy usługi publiczne niż prywatne przez co znajduje się w większym zasięgu zainteresowania ludzi niż jakichkolwiek firm. Lotnisko te było remontowane w 2016 roku, gdzie remont obejmował nawierzchnie płyty lotniska.

### 4.4 Analiza SWOT organizacji

Silne strony organizacji	Słabe strony organizacji
<ul style="list-style-type: none"><li>Największy obiekt tego typu w regionie</li><li>Stale rosnąca liczba pasażerów</li><li>Bezpośrednie połączenie do Szczecina</li></ul>	<ul style="list-style-type: none"><li>Mały rozmiar obiektu w porównaniu z konkurencją</li><li>Przestarzała infrastruktura lotniska</li></ul>
Szanse organizacji	Zagrożenia organizacji
<ul style="list-style-type: none"><li>Kontynuacja podtrzymania rosnącej liczby pasażerów lotniska</li><li>Zwiększenie liczby pasażerów międzynarodowych</li></ul>	<ul style="list-style-type: none"><li>Rozprzestrzenianie się koronawirusa (COVID-19) i związane z nim zmniejszenie się liczby podróżujących</li></ul>

## **5 Specyfikacja wymagań**

### **5.1 Charakterystyka ogólna**

#### **5.1.1 Definicja produktu**

System nadzoruje operacje, które wykonywane są na lotnisku. Produkt ten ma usprawnić pracę lotniska po przez dynamiczniejszą infrastrukturę.

#### **5.1.2 Podstawowe założenia**

System będzie przekazywał informację na tablicach przylotów/odlotów. Umożliwia łatwy sposób zidentyfikowania pasażerów oraz weryfikacje biletów. Bieżąca aktualizacja stanu lotów oraz przekazywanie najpotrzebniejszych informacji dla załogi pokładu danego lotu.

#### **5.1.3 Cel biznesowy**

Organizacja docelowa chce usprawnić pracę na lotnisku po przez wdrażanie nowego systemu, która pozwoli na łatwiejsze rozszerzanie funkcjonalności w przyszłości.

#### **5.1.4 Użytkownicy**

- Pracownicy Portu Lotniczego Szczecin-Goleniów im. NSZZ „Solidarność”
- Pasażerowie
- Załoga samolotów

#### **5.1.5 Korzyści z systemu**

Administracja lotniska(U.01):

- uproszczenie obsługi lotniska(K.01)
- większa kontrola nad funkcjonowaniem lotniska(K.02)
- możliwość zmniejszenia kadry potrzebnej do obsługi lotniska(K.03)

Pasażerowie(U.02):

- ułatwienie podróży(K.04)
- wygodny dostęp do usług i informacji(K.05)

#### **5.1.6 Ograniczenia projektowe i wdrożeniowe**

Serwer z bazą danych oraz komputery z określonym zastosowaniem w zależności od danego pracownika będą oparte na Windows 10. Wybraliśmy ten system ze względu na jego popularność, wydajność oraz powszechność. Jest to system, który używa większa część ludzi więc nie będzie problem z jego obsługą w zaleźny sposób. Dane, które będą przekazywane między sobą za pomocą protokołu HTTP, który będzie przekazywał informację między komputerem a serwerem. Dyski we wszystkich komputerach będą szyfrowane za pomocą programu BitLocker, który nie pozwoli na kradzież danych przez osoby trzecie. Dane przekazywane do bazy danych będą hashowane za pomocą algorytmów mieszających MD5() przez co dane nie będą jawnie przekazywane między sobą.

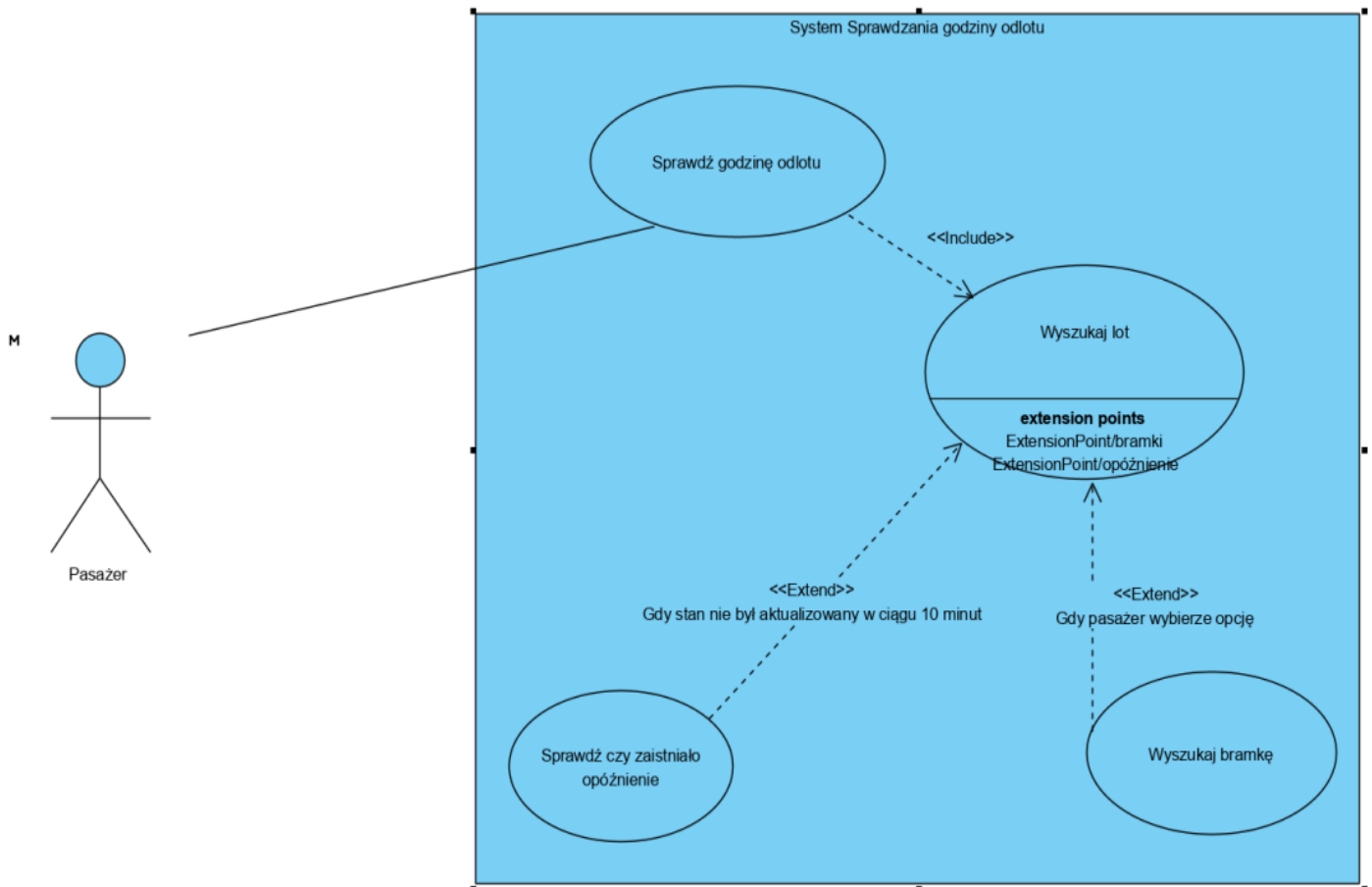
## 5.2 Wymagania funkcjonalne

### 5.2.1 Lista wymagań

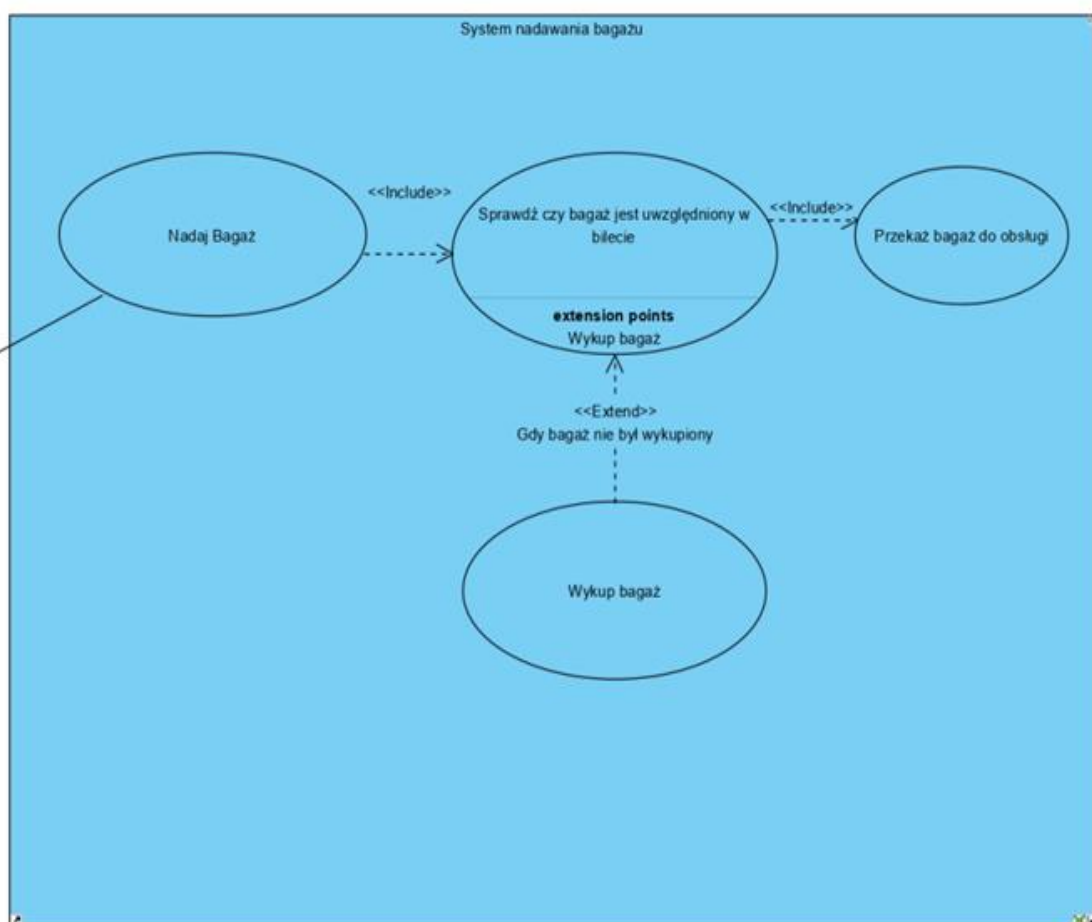
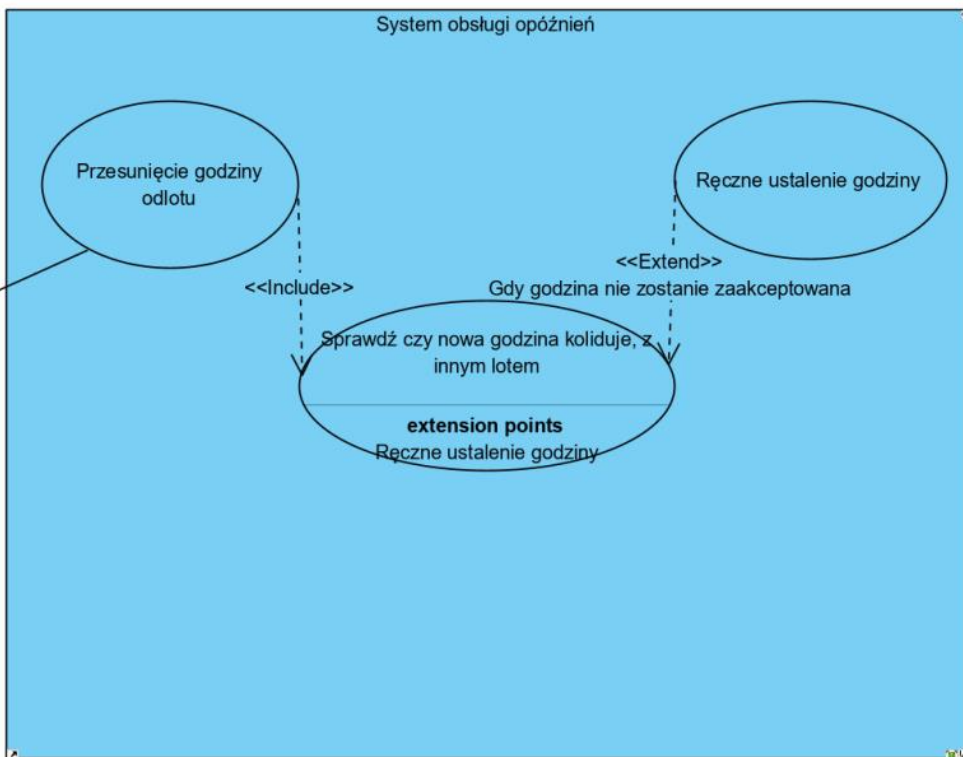
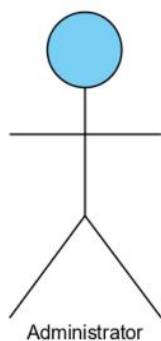
- Obsługa pasażera (obsługa lotniska)
- Obsługa bagażu (obsługa lotniska)
- Załoga lotnicza (załoga samolotu)
- Nadawanie bagażu (klient)
- Godziny odlotu (klient)
- Obsługa opóźnień (administrator)
- Płyta lotniska (administrator)

Diagramy

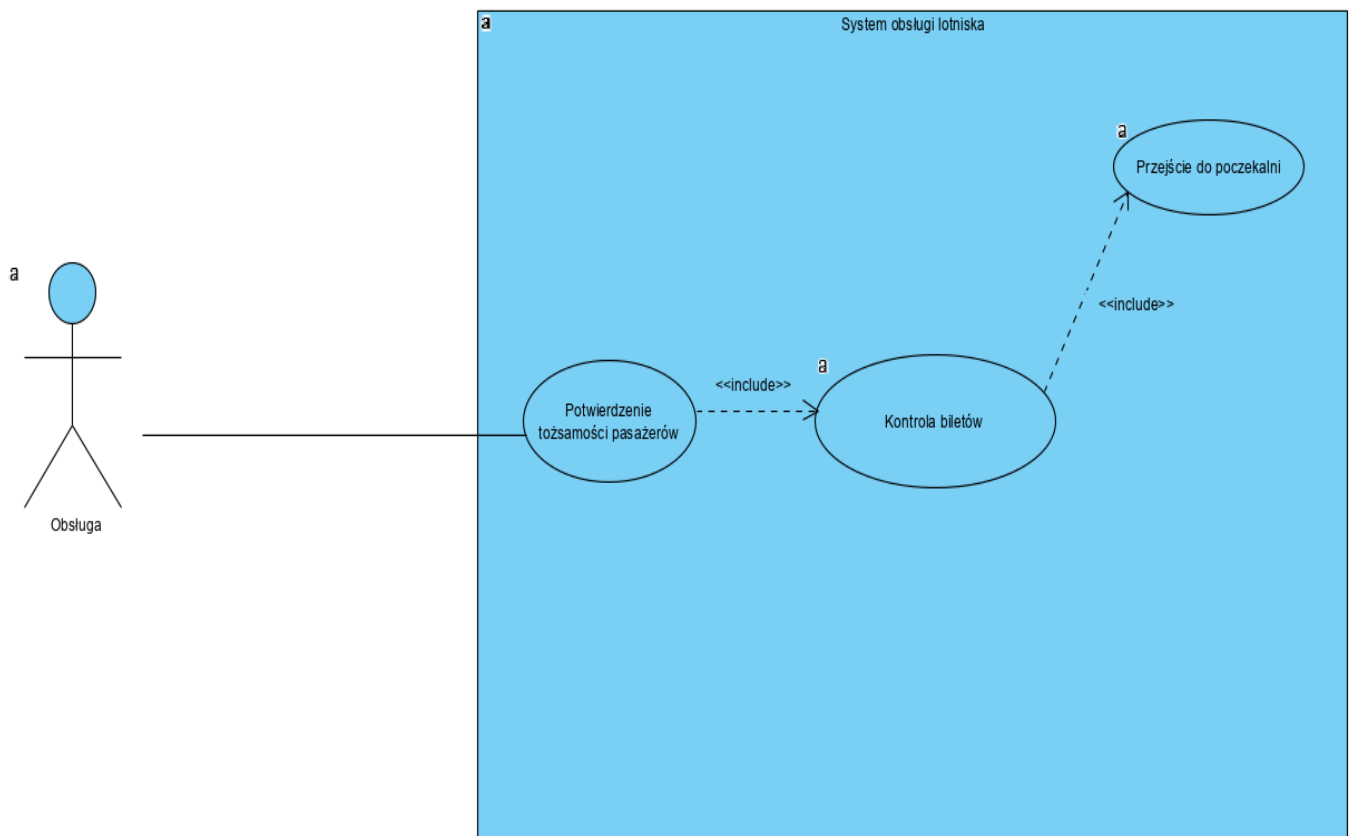
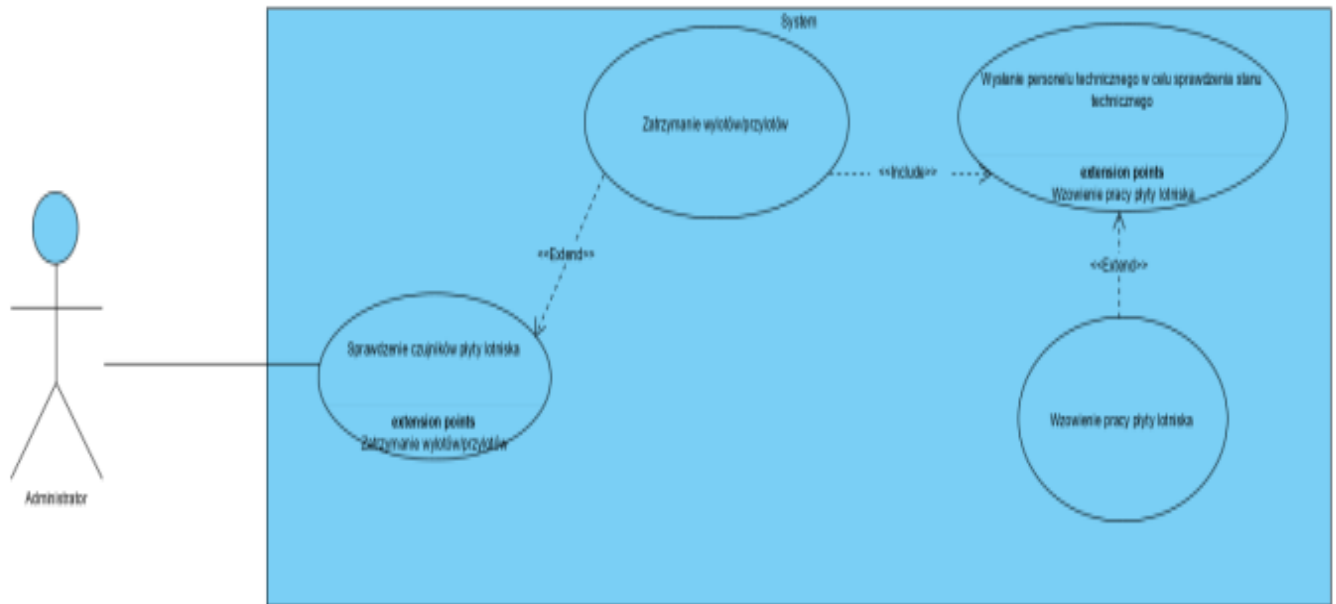
### 5.2.2 Diagramy przypadków użycia

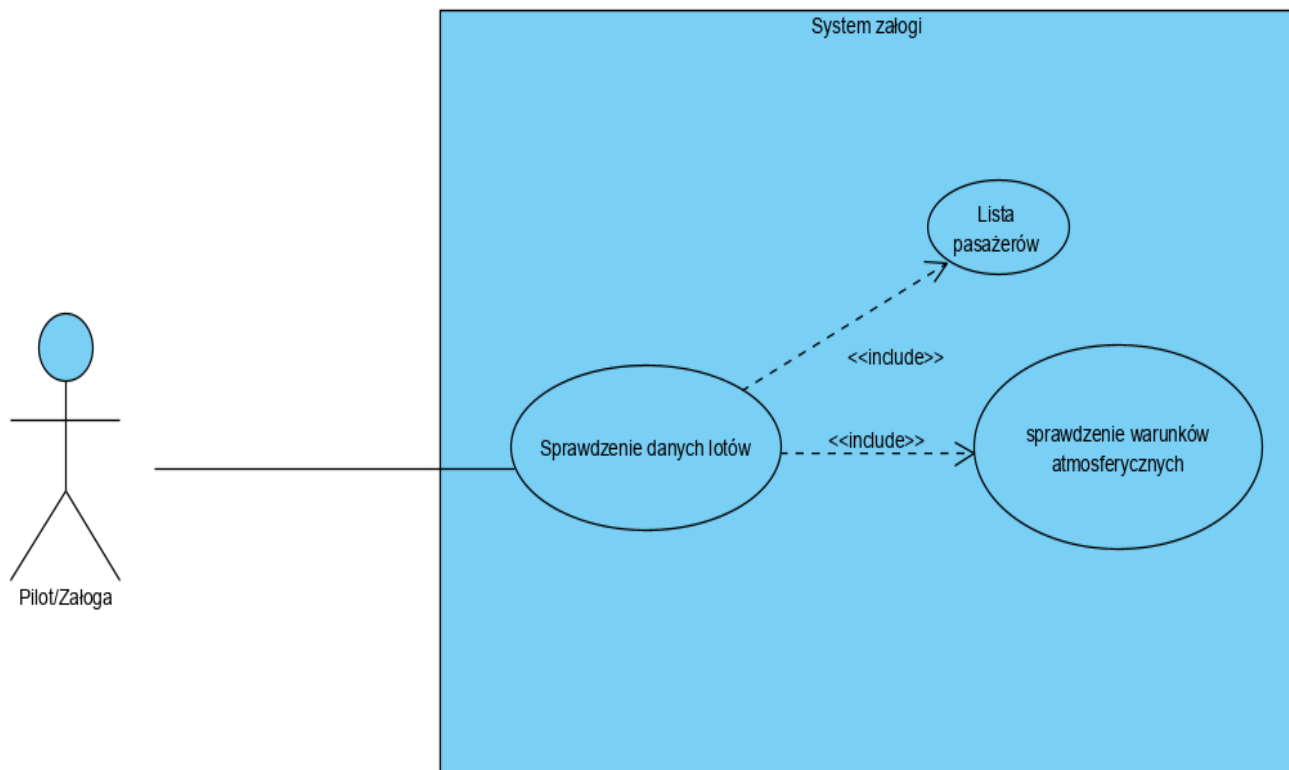
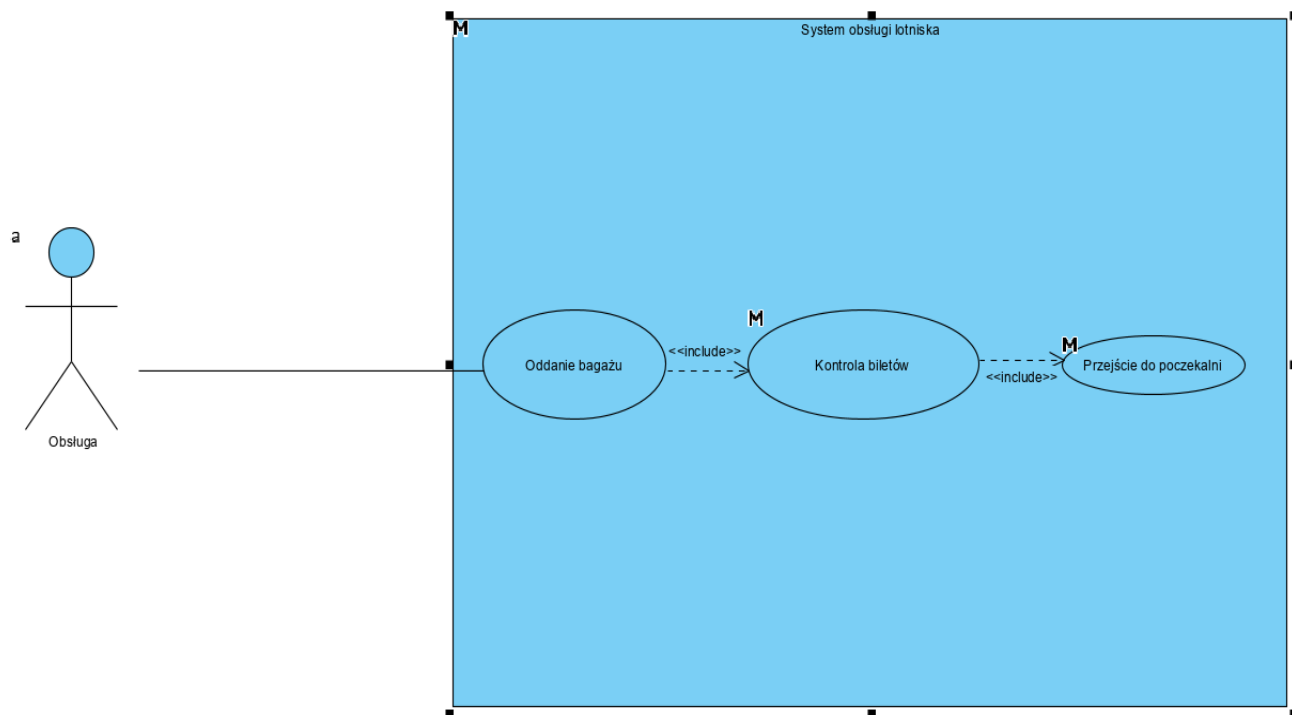


a









### 5.2.3 Szczegółowy opis wymagań

- ID = 1
- Sprawdzenie godzin odlotu przez klienta
- ułatwienie podróży(K.04), wygodny dostęp do usług i informacji(K.05)
- Pasażerowie(U.02)
- Scenariusze, dla każdego z nich:
  - Pasażer posiada ważny bilet
  - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
    - .1. Wybranie odpowiedniej opcji w menu**
    - .2. Zeskanowanie biletu**
      - **Gdy zeskanowany bilet widnieje w bazie ważnych biletów następuje przejście do następnego kroku**
      - **Gdy bilet nie widnieje w bazie ważnych biletów podróżujący zostaje skierowany do informacji.**
    - .3. Wyświetlenie godziny odlotu oraz bramki(System)**
  - Efekty – Pasażer wie kiedy i do której bramki powinien się udać
  - Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
  - Częstotliwość - 5
  - Istotność – 2

- ID = 2
- Nadanie bagażu (klient)
- ułatwienie podróży(K.04), wygodny dostęp do usług i informacji(K.05)
- Pasażerowie(U.02)
- Scenariusze, dla każdego z nich:
  - Pasażer posiada ważny bilet oraz bagaż
  - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
    - .1. Pasażer udaje się do punktu nadawania bagażu**
    - .2. Zeskanowanie biletu**
      - **Możliwość wykupienia bagażu gdy ten nie jest zawarty w bilecie**
      - **Gdy bilet uwzględnia bagaż nadawany przejście dalej**
    - .3. Zważenie bagażu**
      - **Gdy waga bagażu jest niższa niż 20 kg następuje przejście do następnego kroku**
      - **Gdy waga bagażu przekroczy 20 kg przejście do następnego kroku będzie zależało od uiszczenia odpowiedniej opłaty.**
    - .4. Przekazanie bagażu obsłudze**
      - Efekty – Pasażer nadał bagaż
      - Wymagania нефункционалне – szczegółowe wobec poszczególnych wymagań funkcjonalnych
      - Częstotliwość - 5
      - Istotność – 4

- ID = 3
- Nazwa - Obsługa opóźnień(reorganizacja wylotów) –
- Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w
  - . (id i treść elementu, do którego się odwołujemy)
    - .1. możliwość zmniejszenia kadry potrzebnej do obsługi lotniska(K.03)
    - .2. uproszczenie obsługi lotniska(K.01)
    - .3. większa kontrola nad funkcjonowaniem lotniska(K.02)
- Użytkownicy Administrator
- Scenariusze, dla każdego z nich:
  - Warunki początkowe – wystąpiło opóźnienie
  - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
    - .1. Sprawdzenie czy zgłoszono opóźnienie(system)**
    - .2. Wygenerowanie nowej godziny odlotu**
    - .3. Sprawdzenie czy nowa godzina koliduje z innym odlotem**
      - Gdy godzina koliduje z innym odlotem następuje zgłoszenie do administracji oraz wygenerowanie godziny wylotu w następnym dniu.
      - Gdy godzina nie koliduje z innymi odlotami następuje przejście do następnego działania
    - .4. Przekazanie nowej godziny do akceptacji przez administrację**
      - Gdy administracja nie zaakceptuje wygenerowanej godziny, należy wpisać ją ręcznie lub wygenerować ją w późniejszej dacie.
      - Przejście do następnego działania gdy administracja zaakceptuje wygenerowaną godzinę
    - .5. Zaktualizowanie godziny odlotu**
  - Efekty – Wygenerowanie nowej godziny odlotu w przypadku opóźnienia
  - Wymagania нефunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
  - Częstotliwość - 1
  - Istotność – inaczej: zależność krytyczna, znaczenie - 5

- ID = 4
- Nazwa - Nadzór stanu płyty lotniska
- Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w
  - . (id i treść elementu, do którego się odwołujemy)
    - .1. uproszczenie obsługi lotniska(K.01)
    - .2. większa kontrola nad funkcjonowaniem lotniska(K.02)
    - .3. możliwość zmniejszenia kadry potrzebnej do obsługi lotniska(K.03)
- Użytkownicy Administrator
- Scenariusze, dla każdego z nich:
  - Warunki początkowe
  - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
    - .1. Sprawdzenie czujników**
      - Gdy czujniki nie zawiadomią o awarii następne kroki nie są wykonywane, następuje ponowne sprawdzenie czujników
      - Gdy czujniki wskażą wystąpienie awarii następuje przejście do następnej czynności
    - .2. Powiadomienie administracji o wystąpieniu awarii**
      - Gdy administracja zdecyduje o kontynuacji normalnej pracy lotniska następuje zresetowanie czujników i powrót o czynności nr 1.
      - Gdy administracja zdecyduje o kontynuacji pracy lotniska, ale chce sprawdzić źródło potencjalnej awarii następuje pominięcie kroku nr 3.
      - Gdy administracja zdecyduje o zatrzymaniu pracy lotniska następuje przejście do następnego punktu.
    - .3. Wysłanie komunikatu o zatrzymaniu pracy płyty lotniska**
    - .4. Wysłanie personelu technicznego na obszar płyty który zgłosił awarię**
  - Efekty – Wygenerowanie nowej godziny odlotu w przypadku opóźnienia
  - Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
  - Częstotliwość - 1

Istotność – inaczej: zależność krytyczna, znaczenie – 5

- ID = 5
- Kontrola biletów
- Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w □. (id i treść elementu, do którego się odwołujemy)
- Obsługa na lotnisku
- Scenariusze, dla każdego z nich:
  - Pasażer posiada ważny dokument tożsamości oraz bilet
  - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
    - 1. Przekazanie dokumentu tożsamości**
    - 2. Sprawdzenie danego biletu**
    - 3. Oczekiwanie na wybrany lot**
  - Efekty – szybka identyfikacja pasażera oraz przydzielenie go do odpowiedniego lotu
  - Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
  - Częstotliwość - 5
  - Istotność – 4

- ID = 6
- Przekazanie bagażu
- Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w □. (id i treść elementu, do którego się odwołujemy)
- Obsługa na lotnisku
- Scenariusze, dla każdego z nich:
  - Naklejenie numeru identyfikacyjnego na bagaż
  - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
    - 1. Przekazanie bagażu obsłudze**
    - 2. Nadanie identyfikatora danemu bagażu**
    - 3. Umieszczenie bagażu na taśmie i przekazanie do odpowiedniego samolotu.**
  - Efekty – identyfikacja bagażu do odpowiedniego pasażera
  - Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
  - Częstotliwość - 5
  - Istotność – 3



- ID = 7
- System załogi
- Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w □. (id i treść elementu, do którego się odwołujemy)
- Załoga lotnicza
- Scenariusze, dla każdego z nich:
  - Wybranie danego lotu
  - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
    - 1. Załogowanie się do odpowiedniego lotu(system)**
    - 2. Sprawdzenie ilości pasażerów**
    - 3. sprawdzenie pogody/ warunków atmosferycznych**
  - Efekty – dostarczenie najpotrzebniejszych informacji dla całej załogi samolotu
  - Wymagania нефункционалне – szczegółowe wobec poszczególnych wymagań funkcjonalnych
  - Częstotliwość - 5
  - Istotność – 4

### 5.3 Wymagania niefunkcjonalne

1. **Wydajność** – w odniesieniu do konkretnych sytuacji – funkcji system  
Funkcje systemu będą umożliwiać korzystanie z nich przez 1000 użytkowników jednocześnie  
System oraz bazy danych nie będą zajmować więcej niż 2 TB pamięci.
2. **Bezpieczeństwo** – utrata, zniszczenie danych, zniszczenie innego systemu przez nasz  
– wraz z działaniami zapobiegawczymi i ograniczającymi skutki

Weryfikacja użytkownika w przypadku pasażerów wykonywana za pomocą unikalnego numeru biletu, pracownicy lotniska(w tym załoga samolotów) otrzymają unikalne ID oraz hasło, a także kartę magnetyczną, po których wprowadzeniu otrzymają dostęp do odpowiednich elementów systemu. Zawartość dysków w komputerach obsługujących system oraz serwerze głównym będzie zapisywana w cyklicznych odstępach czasu (1 dzień) na dyskach w RAID 6 umieszczonych przy serwerze głównym.

3. **Zabezpieczenia** - dostęp do systemu będzie możliwy wyłącznie z terenu lotniska. Do danych krytycznych dostęp będą miał tylko upoważniony personel po wcześniejszym potwierdzeniu tożsamości. Zawartość dysków w komputerach obsługujących system będzie szyfrowana za pomocą programu BitLocker. Połączenia pomiędzy serwerem głównym a komputerami obsługi będą szyfrowane przy pomocy technologii MD5
4. **Inne cechy jakości** – najlepiej ilościowo, żeby można było zweryfikować (zmierzyć)  
– adaptowalność, dostępność, poprawność, elastyczność, łatwość konserwacji, przenośność, awaryjność, testowalność, użyteczność

Cały system funkcjonować będzie na łatwo dostępnych podzespołach(komputerach stacjonarnych wyposażonych w kartę sieciową pozwalającą na podłączenie komputera do sieci LAN oraz serwerze główny opartym na serwerach HPE ProLiant DL360 Gen10), zapewniając odpowiednią skalowalność systemu. W związku z wykorzystaniem dostępnych na rynku komputerów(HP ProDesk 400 G4) do łączenia się z serwerem również konserwacja nie będzie skomplikowana.

## 6 Zarządzanie projektem

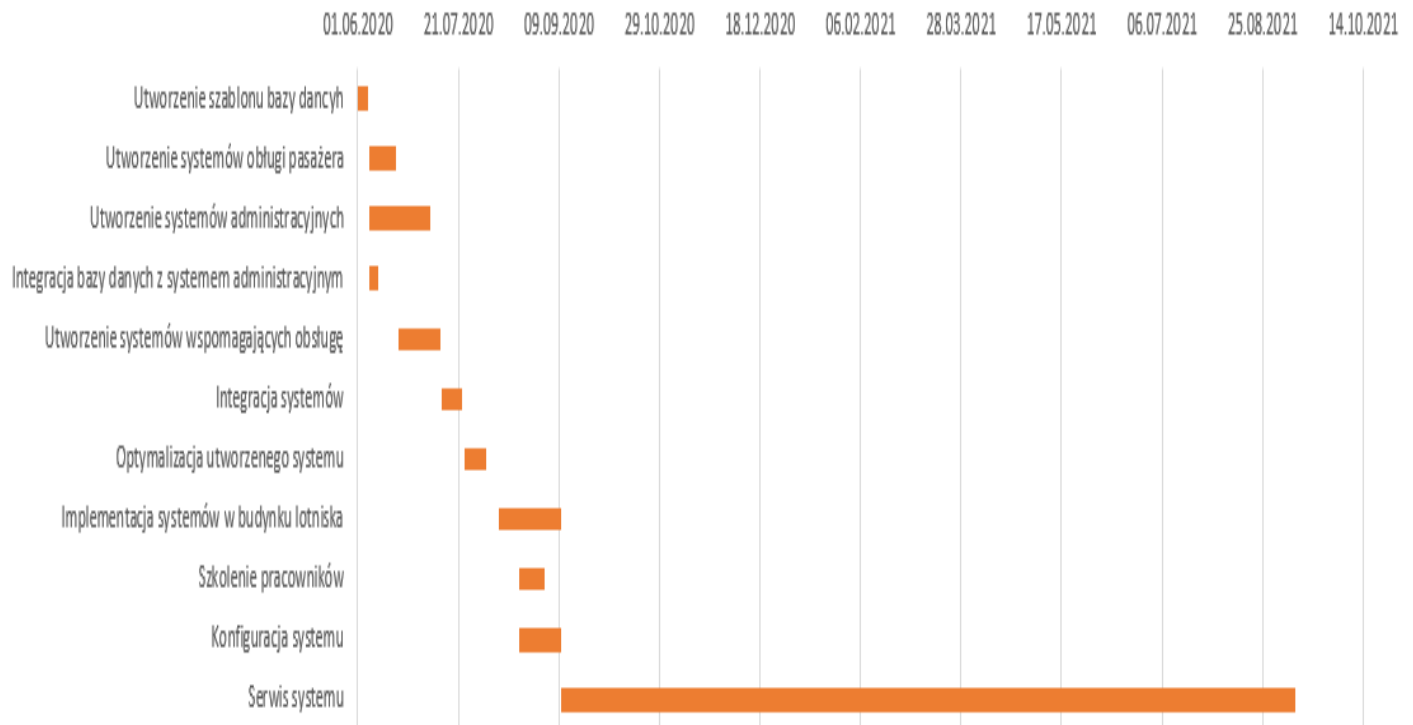
### 6.1 Zasoby ludzkie

3 - programistów

6 – osób wprowadzających system

### 6.2 Harmonogram prac

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta. Obejmuje również harmonogram wdrożenia projektu – np. szkolenie, rozruch, konfiguracja, serwis – może obejmować różne wydania (tj. o różnej funkcjonalności – personal, professional, enterprise) i wersje (1.0, 1.5, itd.)



### 6.3 Etapy/kamienie milowe projektu

1. Integracja systemów
2. Implementacja systemów w budynku lotniska
3. Konfiguracja systemu

## **7 Zarządzanie ryzykiem**

### **7.1 Lista czynników ryzyka**

- 1) Brak prądu na lotnisku
- 2) Ograniczony dostęp do Internetu niewielkie
- 3) Awaria systemu
- 4) Uszkodzenie bazy danych
- 5) Uszkodzenie okablowania transmisji danych

### **7.2 Ocena ryzyka**

- 1) nieznaczne, zatrzymanie pracy lotniska i systemu
- 2) nieznaczne, praca systemu w trybie offline
- 3) niewielkie
- 4) niewielkie, utrata danych
- 5) małe, chwilowe zatrzymanie pracy terminalu połączonego uszkodzonym kablem

### **7.3 Plan reakcji na ryzyko**

- 1) Brak prądu na lotnisku – zastosowanie agregatów, które pozwolą na pracę najpotrzebniejszych podsystemów
- 2) Ręczne wprowadzanie danych do systemu/ użycie danych z przed utraty Internetu
- 3) Automatyczne diagnozowanie problemu, wysłanie ekipy serwisowej
- 4) Odbudowanie bazy danych z kopi zapasowej
- 5) Wykrycie uszkodzonego kabla, wysłanie ekipy serwisowej

## 8 Zarządzanie jakością

### 8.1 Scenariusze i przypadki testowe

ID – 3

Nazwa scenariusza – Test systemu obsługi opóźnień

Kategoria – testy jednostkowe

Opis – test poprawnej reorganizacji wylotów

Tester – tester oprogramowania

Termin – **TODO – po wytworzeniu modułu**

Narzędzia wspomagające – brak

Przebieg działań:

Lp.	tester	system
1.	Oflagowanie lotu jako opóźnionego	Wyszukanie lotu oflagowanego, oraz lotów następujących po nim
2.	Wpisanie oczekiwanej godziny przylotu samolotu opóźnionego	Wygenerowanie nowej godziny wylotu po uwzględnieniu czasu potrzebnego na wykonanie procedur związanych z przylotem, oraz odprawą
3.	-	Wypisanie proponowanej godziny odlotu
4.	Sprawdzenie czy podane godziny odlotu są poprawne, i ich akceptacja(Gdy są poprawne)	Przekazanie nowych godzin odlotu do systemów informacyjnych, pozytywne zakończenie testu
5.	-	Negatywne rozpatrzenie testu.

założenia, środowisko, warunki wstępne, dane wejściowe

- założenia – poprawność wykazywanych godzin opóźnień
- środowisko – Python 3.8.2
- warunki wstępne – system obsługi opóźnień połączony z systemami informacyjnymi
- dane wejściowe – aktualna baza danych z przylotami i wylotami.

zestaw danych testowych

Z1: Baza wylotów i przylotów z obecnym oknem na wylot - 1

Zwracana informacja : nowa godzina odlotu - 4

Z2: Baza wylotów i przylotów bez możliwego okna wylotu - 1

Zwracana informacja: „Brak możliwej godziny odlotu” - 5

ID – 4

Nazwa scenariusza – Nadzór stanu płyty lotniska

Kategoria – testy jednostkowe

Opis – test poprawnej działania w przypadku wykrycia uszkodzenia nawierzchni płyty lotniska lub awarii systemów znajdujących się na niej.

Tester – tester oprogramowania

Termin – **TODO – po wytworzeniu modułu**

Narzędzia wspomagające – brak

Przebieg działań:

Lp.	tester	system
1.	Wysłanie potwierdzenia o poprawnym stanie płyty lotniska	System nie podejmuje działań (Gdy system podejmie działania test zostaje rozpatrzony negatywnie)
2.	Wysłanie informacji o błędzie z losowego czujnika	Wysłanie powiadomienia o awarii
3.	-	Sprawdzenie ID czujnika zgłaszającego awarię, przekazanie lokacji awarii na terminal(w wypadku testowy testerowi)
4.	Sprawdzenie poprawnej lokacji wykrycia awarii	Pozytywne rozpatrzenie testu – gdy tester potwierdzi zgodność
5.	-	Negatywne rozpatrzenie testu

założenia, środowisko, warunki wstępne, dane wejściowe

- założenia – poprawność synchronizacji modułów sprawdzających stan płyty lotniska
- środowisko – Python 3.8.2
- warunki wstępne – Symulacja połączenia z czujnikami.
- dane wejściowe –zestaw danych testowych

Z1: Informacja o poprawnym stanie lotniska - 1

Zwracana informacja : brak

Z2: Informacja o awarii losowo wybranego czujnika- 2

Zwracana informacja: ID czujnika oraz jego dokładna lokacja – 3

ID – 5

Nazwa scenariusza – Kontrola biletów

Kategoria – testy jednostkowe

Opis – test poprawnego zweryfikowania tożsamości oraz biletu pasażera

Tester – tester oprogramowania

Termin – **TODO – po wytworzeniu modułu**

Narzędzia wspomagające – brak

Przebieg działań:

Lp.	tester	system
1.	Podaje dokument tożsamości weryfikujący pasażera	Sprawdza czy pasażer znajduje się w systemie
2.	-	Weryfikuje czy dane na bilecie oraz dokumencie są poprawne
3.	Udaje się do poczekalni (po potwierdzeniu wszystkich danych)	Potwierdza udanie się do poczekalni na dany lot przy poprawnych danych lub blokuje, jeżeli dane się nie zgadzają w jakimś stopni i ponownie są weryfikowane.

założenia, środowisko, warunki wstępne, dane wejściowe

- założenia – Porównanie danych na dokumencie oraz na bilecie
- środowisko – Python 3.8.2
- warunki wstępne – System kontroli biletów połączony z bazą danych
- dane wejściowe – informacje o pasażerze

Z1: Weryfikacja dostarczonych danych o pasażerze - 2

Zwracana informacja : Potwierdzenie poprawności lub ponowna weryfikacja - 3

ID – 6

Nazwa scenariusza – System przekazania bagażu

Kategoria – testy jednostkowe

Opis – test poprawnego działania przekazaniu bagażu i przypisania do pasażera danego lotu

Tester – tester oprogramowania

Termin – **TODO – po wytworzeniu modułu**

Narzędzia wspomagające – brak

Przebieg działań:

Lp.	tester	system
1.	Przekazanie bagażu do miejsca z przekazywanymi bagażami	-
2.	Sprawdzenie biletów danego pasażera	Nadanie ID bagażu do danego pasażera
3.	-	Po poprawnej weryfikacji biletów przekazanie bagażu na dany lot.

założenia, środowisko, warunki wstępne, dane wejściowe

- założenia – Sprawdzenie przypisanego bagażu do pasażera
- środowisko – Python 3.8.2
- warunki wstępne – Połączenie z systemem Kontroli biletów
- dane wejściowe – Pasażer musi posiadać bilet na lot

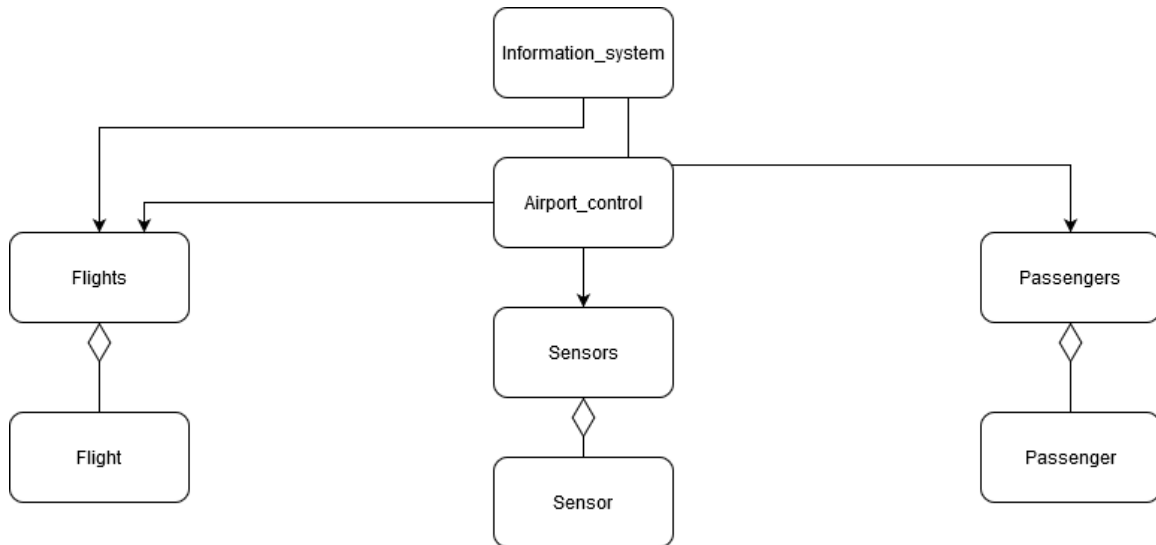
Z1: Informacja o weryfikacji pasażera do bagażu - 2

Zwracana informacja : brak



## 9 Projekt techniczny

### 9.1 Opis architektury systemu



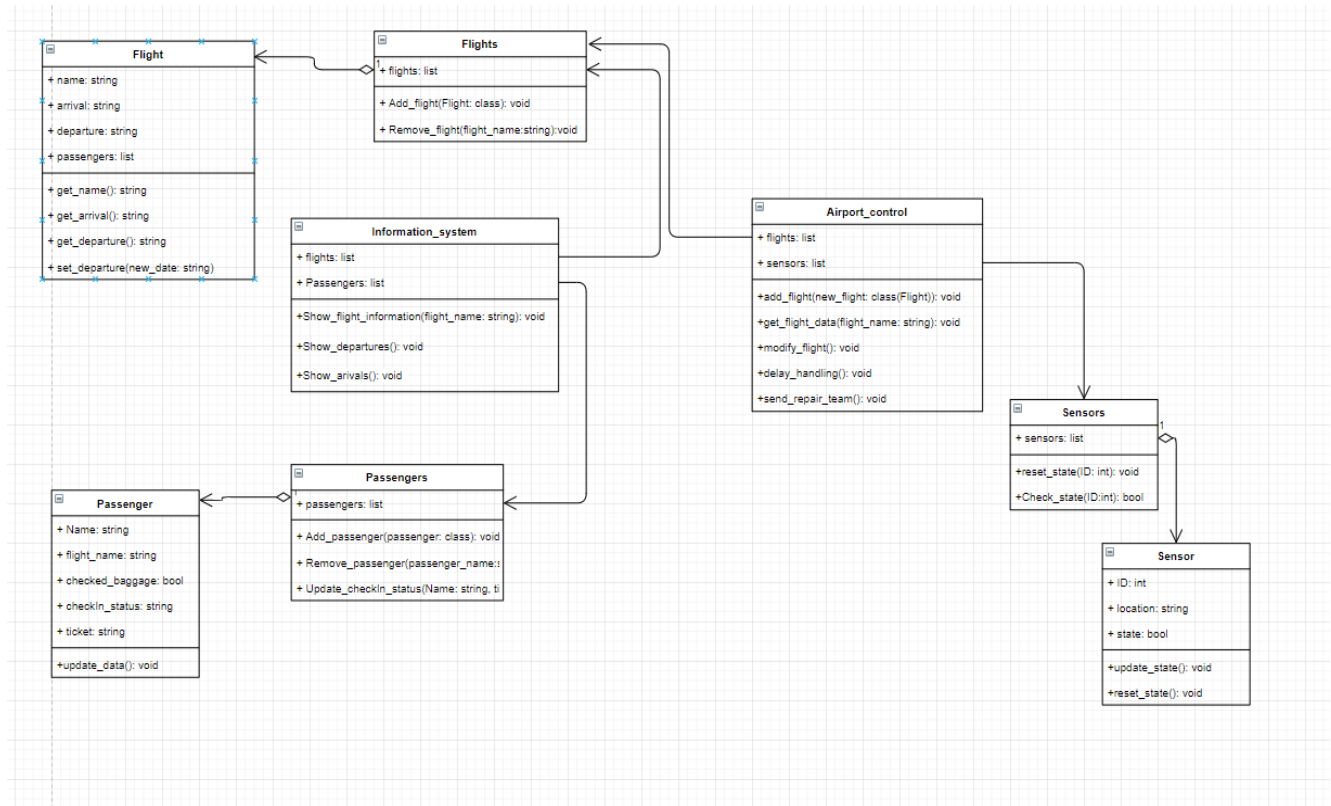
Główną klasą jest `Information_system`, które zajmuje się pozostałymi klasami, które są ze sobą powiązane. W tej klasie znajdują się informacje o wszystkich lotach oraz pasażerach. Klasy `Flights` oraz `Passengers` zawierają informacje odpowiednio o lotach oraz pasażerach. W tych klasach można odpowiednio dodawać lub usuwać loty/pasażerów. Pod klasy `Flight` oraz `Passenger` zawierają informacje o pojedynczym pasażerze/ locie, które są dodawane. Klasa `Airport_control` zajmuje się nadzorem lotniska oraz (czynnościami administracyjnymi), którymi będą zajmowali się pracownicy. W tej klasie dodaje się oraz usuwa się loty itp. Klasa `Sensors` zajmuje się czujnikami, które kontrolują stan płyty lotniska. Pod klasa `Sensor` zajmuje się pojedynczym czujnikiem na płycie lotniska.

### 9.2 Technologie implementacji systemu

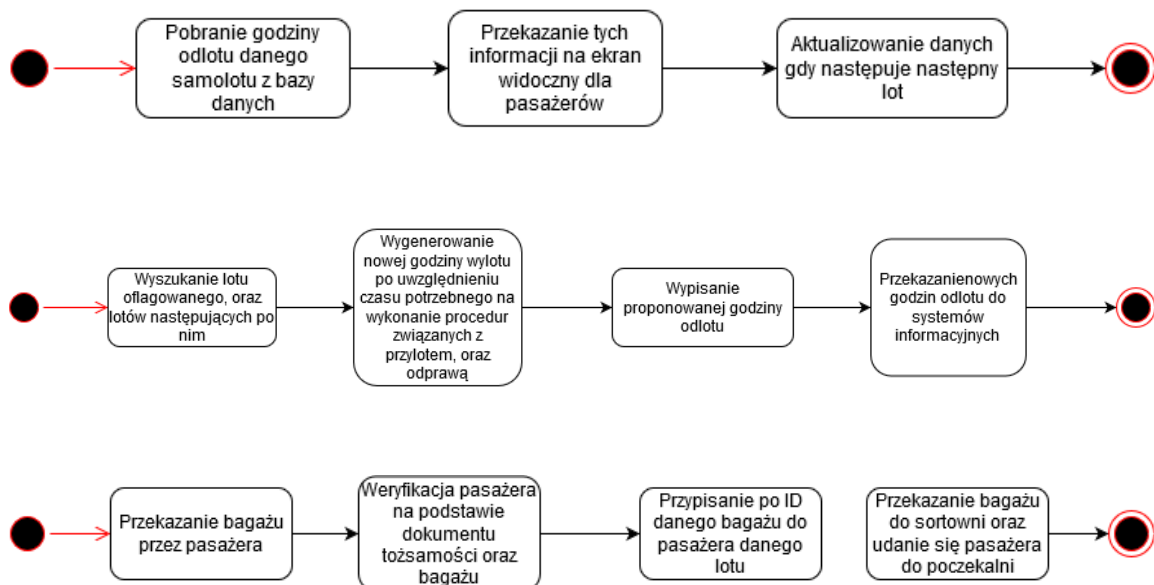
Technologia	Uzasadnienie
Python 3.8.2	Najbardziej rozwojowy język, który pozwala na łatwe rozszerzanie funkcjonalności
MySQL 8.0.16	Najpopularniejszy model baz danych
MD5()	Szyfrowanie danych przekazanych do bazy

## 9.3 Diagramy UML

### 9.3.1 Diagram(-y) klas

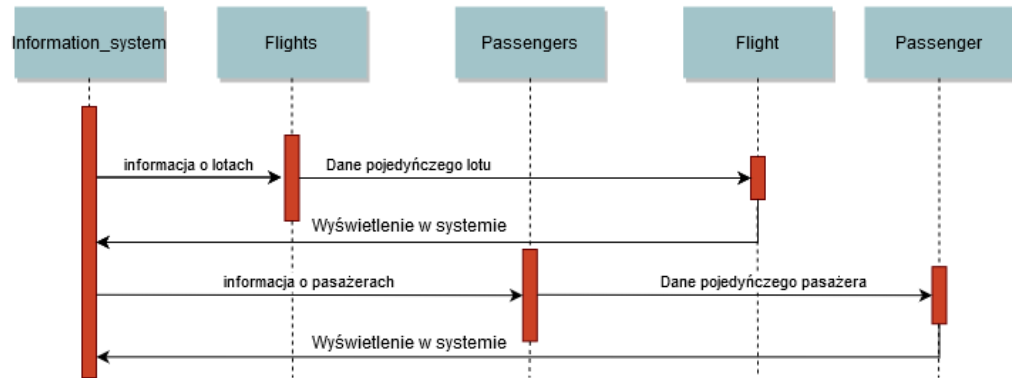


### 9.3.2 Diagram(-y) czynności



### 9.3.3 Diagramy sekwencji

Diagram sekwencyjny



### 9.3.4 Inne diagramy

Diagram komponentów(Kontrola biletów):

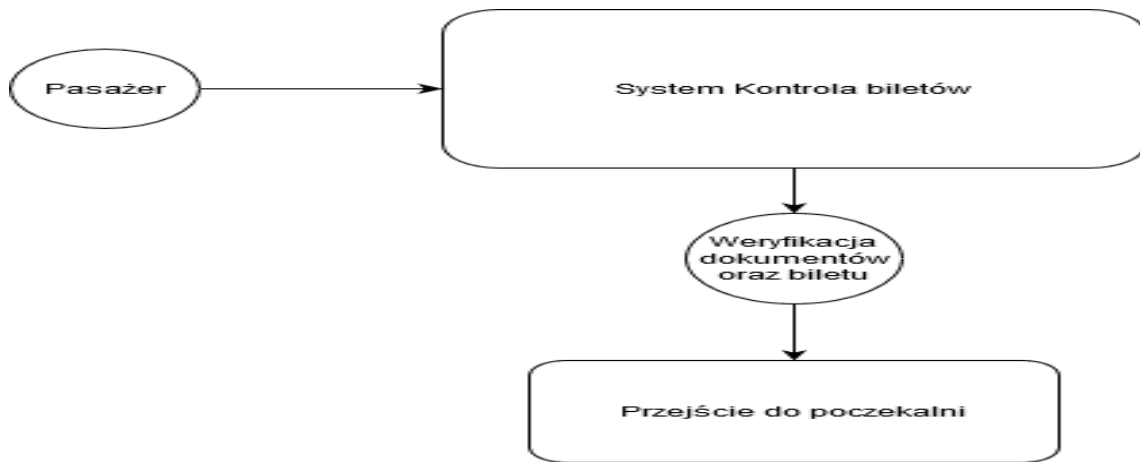


Diagram rozmieszczenia:

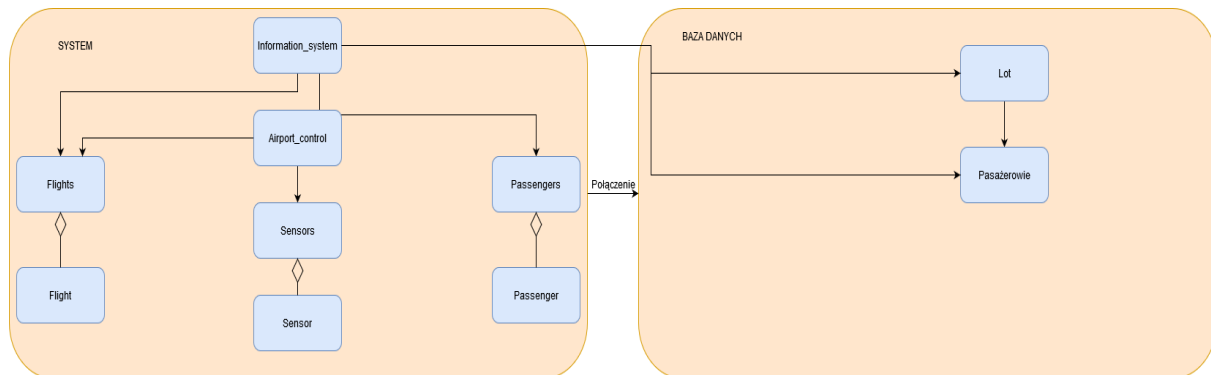
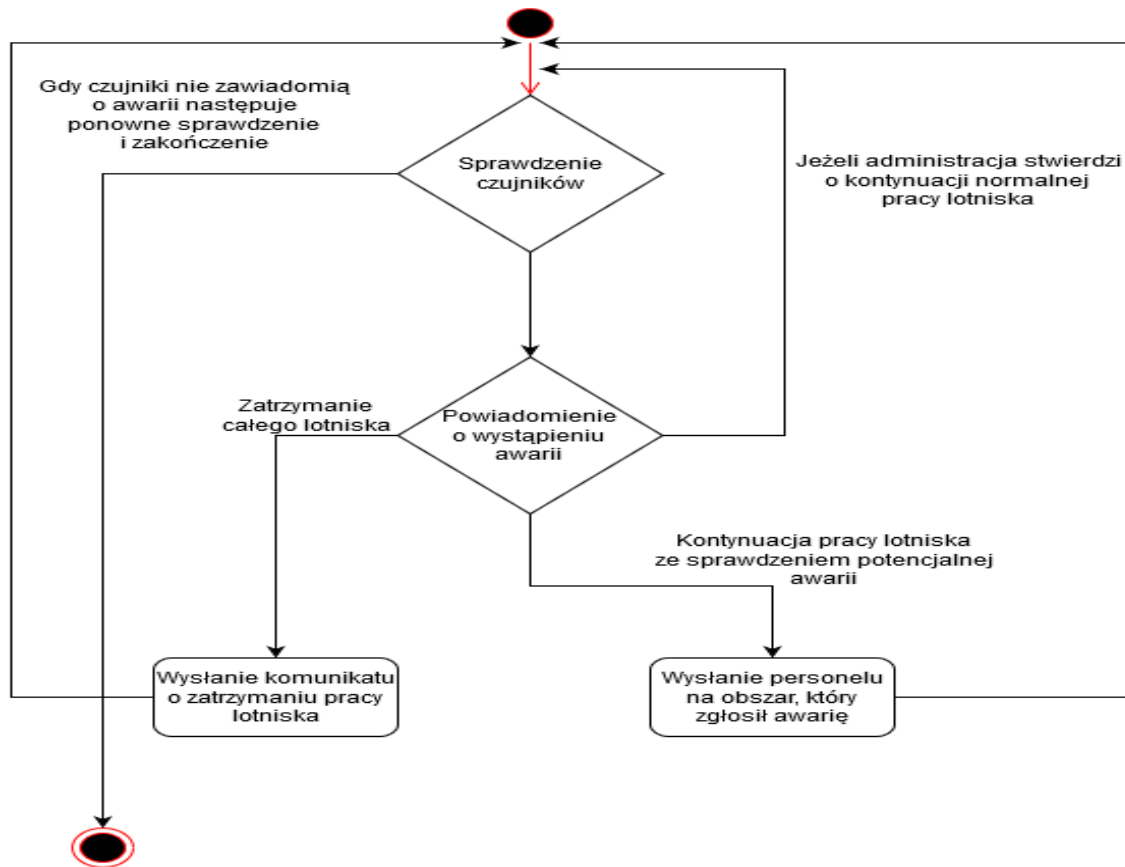


Diagram maszyny stanowej (Nadzór stanu płyty lotniska):



#### 9.4 Charakterystyka zastosowanych wzorców projektowych

Główną rzeczą, którą będzie widział użytkownik jest GUI w którego skład wchodzi Information\_system na którym widać wszystkie dane na temat lotów oraz pasażerów. Klasa Flights oraz Passengers wraz z pod klasami Flight oraz Passenger łączy się z bazą danych, w której znajdują się tabele Lot oraz Pasażer, gdzie widnieją wszystkie informacje, które przekazywane są do głównej klasy Information\_system. Jest również klasa „techniczna” Airport\_control, która zajmuje się nadzorowaniem płyty lotniska za pomocą czujników które zawarte są w klasie Sensors, a pojedynczy czujnik w klasie Sensor. Jest to klasa przeznaczona dla części administracyjnej lotniska.

## 9.5 Projekt bazy danych

### 9.5.1 Schemat

Pasażer						
Numer Biletu	Imię	Nazwisko	Nazwa Lotu	Typ Bagażu	Status Odprawy	Numer fotela
PK			FK1			

Lot						
Nazwa Lotu	Numer Rejestracyjny	Model Samolotu	Godzina odlotu	Godzina przylotu	Bramka	Destynacja
PK						

Baza danych lotnisko składa się z dwóch tabel Pasażer oraz Lot. Kluczami głównymi obu tabel są Numer Biletu oraz Nazwa Lotu na podstawie których będą weryfikowane poszczególni pasażerowie oraz loty. Tabela Lot jest powiązana z tabelą Pasażer kluczem obcym Nazwa Lotu. Oznacza to, że pasażera nie doda się do bazy danych, jeżeli nie widnieje w tabeli Lot dany kurs do którego pasażer miałby być przypisany. Aby pasażer mógł zostać dodany musi być przypisany do lotu, który istnieje (w tabeli Lot). Wszystkie pola w tabeli Lot są varchar o zadanej długości w zależności od kolumny. W tabeli Pasażer również wszystkie pola są typu varchar o zadanej długości w zależności od kolumny oprócz kolumny Numer Biletu oraz Numer fotela, które są int o danych długościach,

## 9.6 Projekt interfejsu użytkownika

### 9.6.1 Lista głównych elementów interfejsu

- Wyświetlanie odlotów
- Dodanie Pasażera
- Dodanie Lotu
- Wyświetlenie informacji o wybranym locie
- System obsługi opóźnień

### 9.6.2 Przejścia między głównymi elementami

### 9.6.3 Projekty szczegółowe poszczególnych elementów

- numer – 1
- nazwa – Wyświetlanie odlotów
- projekt graficzny –

# Odloty

Nazwa lotu	Destynacja	Wylot	Przylot	Bramka
RY0001	Warszawa	2020-06-08 12:00:00	2020-06-08 15:00:00	A10
RY0002	Warszawa	2020-06-08 13:00:00	2020-06-08 16:00:00	A10
RY0003	Warszawa	2020-06-08 14:00:00	2020-06-08 17:00:00	A10

Cofnij

Odśwież

- wykorzystane dane – Tabela lotów z bazy danych

- numer – 2
- nazwa – Formularz dodania lotu
- projekt graficzny –

## Dodaj Lot

Nazwa Lotu:	A1234
Numer rej:	A1
Model samolotu:	Boeing 777
Wylot:	6/12/20 10:00
Przylot:	6/12/20 12:00
Bramka	A21
Destynacja	Madryt

Anuluj

Dodaj

- numer – 3
- nazwa – Formularz dodania Pasażera
- projekt graficzny –

## Dodaj Pasażera

Numer Biletu	1000
Imię:	Jan
Nazwisko:	Kowalski
Nazwa lotu	AB0001
Rodzaj bagażu	rejestrowany
Status odprawy	odprawiony
Fotel	1

Anuluj

Dodaj



- numer – 4
- nazwa – System obsługi opóźnień
- projekt graficzny –

## Obsługa opóźnień

Opóźniony lot:

Opóźnienie

Znaleziono nową optymalną datę wylotu:

22.06.2020 20:45

- Opis – System po podaniu danych wyszukuje wolną datę w której mógłby odbyć się wylot

### ***9.7 Procedura wdrożenia***

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

<b>Zadanie</b>	<b>Start</b>	<b>Koniec</b>	<b>Czas trwania</b>
Utworzenie szablonu bazy danych	01.06.2020	06.06.2020	5
Utworzenie systemów obsługi pasażera	07.06.2020	20.06.2020	13
Utworzenie systemów administracyjnych	07.06.2020	07.07.2020	30
Integracja bazy danych z systemem administracyjnym	07.06.2020	11.06.2020	4
Utworzenie systemów wspomagających obsługę	21.06.2020	12.07.2020	21
Integracja systemów	13.07.2020	23.07.2020	10
Optymalizacja utworzonego systemu	24.07.2020	04.08.2020	11
Implementacja systemów w budynku lotniska	10.08.2020	10.09.2020	31
Szkolenie pracowników	20.08.2020	02.09.2020	13
Konfiguracja systemu	20.08.2020	10.09.2020	21
Serwis systemu	10.09.2020	10.09.2021	365

## 10 Podsumowanie

### 10.1 Szczegółowe nakłady projektowe członków zespołu

Czynność	Maciej	Filip
Tworzenie planu	02.03.20 – 09.03.20 (50%)	02.03.20 – 09.03.20 (50%)
Opracowanie wszystkich punktów	23.03.20 – 18.05.20 (50%)	23.03.20 – 18.05.20 (50%)
Tworzenie prototypu (implementacja) oraz GUI	25.05.20 – 07.06.20 (25%)	25.05.20 – 07.06.20 (75%)
Kończenie oraz poprawnie dokumentacji	07.06.20 (75%)	07.06.20 (25%)
Podsumowanie:	Oceńam, że nakład na projekt lotniska w Goleniowie został wykorzystany maksymalnie. Czas oraz nasze umiejętności poświęcone na stworzenie dokumentacji oraz implementacji prototypu zostały wykorzystane jak najlepiej oraz rzetelnie.	Uważam, że udział w pracach nad projektem był porównywalny, w dalszej części projektu skupiliśmy się na częściach zgodnie z naszymi preferencjami, zachowując jednak równomierny rozkład prac.

Instrukcja jak uruchomić prototyp:

- Proszę pobrać aplikację obsługującą serwer lokalny (np.XAMPP)
- Po przejściu przez wszystkie kroki instalator, uruchomić aplikację
- Kliknąć Start na Apache oraz MySQL
- Uruchomić przeglądarkę i wpisać adres: <http://localhost/phpmyadmin/>
- Utworzyć nową, pustą bazę danych o nazwie: lotnisko
- Potem należy kliknąć Import i wybrać plik załączony w paczce o nazwie: lotnisko.sql
- Następnie po poprawnym zaimportowaniu bazy danych uruchomić plik: GUI.py
- Skompilować kod i otworzy się prototyp.