**1.** **Dokładny opis implementacyjny systemu (na podstawie dokumentacji)**

- W projekcie systemu zostały zaimplementowane funkcjonalności, na potrzebę przedstawionych w opisie firmy w dokumentacji projektowej.

- Zgodnie z diagramem przypadków użycia w systemie znajdują się interfejsy odpowiadające danym funkcjonalnościom.

- Nasz system informatyczny wdraża dokładne kroki podane w scenariuszach powiązanych z DPU. Nowy klient wchodzący na stronę z zaimplementowanym systemem ma dostęp każdej ze stref zdefiniowanych w opisie, które umożliwiają mu wykonanie powyższych funkcjonalności takich jak zakupienie biletu czy realizacja płatności.

- System umożliwia przejście przez zawarte w scenariuszach przebiegi alternatywne takie jak na przykład wyświetlenie komunikatów o błędzie w momencie wprowadzenia błędnych danych do systemu.

- Dodatkową funkcjonalnością również zaimplementowaną w systemie jest możliwość złożenia reklamacji przez klienta i obsługa tej reklamacji przez użytkownika z uprawnieniami kierowniczymi w systemie.

- W systemie zostały zawarte wszystkie klasy wymienione w obiektowym modelu danych, między którymi funkcjonują określone wcześniej w dokumentacji związki i atrybuty. Funkcjonuje on w sposób, który umożliwia przeprowadzenie dokładnego przypadku zilustrowanego na diagramie obiektów.

- Do systemu zaimplementowany został projekt strony głównej, który zawiera w sobie podstawowe informacje podane w opisie firmy oraz możliwość przejścia do poszczególnych projektów interfejsów znajdujących się na kolejnych podstronach.

- Wszystkie interfejsy i ich funkcjonalności są zgodne z funkcjami wymienionymi w diagramie FHD dla poszczególnych użytkowników systemu. Powyższe funkcjonalności są zamieszczone w dedykowanych dla określonej grupy uprawnień użytkowników w postaci panelu użytkownika. Każdy z użytkowników z uprawnieniami wyższymi niż podstawowe ma dostęp do innego rodzaju funkcjonalności pokrywających się z diagramem FHD.

- Interfejs jest zgodny z wcześniej założonymi wymaganiami tj. przydatność, łatwość, estetyka, bezpieczeństwo i stabilność.

- Wszelkie wymagania zostały podstawowo przetestowane, a ich różnice od założonych będą opisane dokładniej w kolejnej części dokumentacji.

**2. Zmiany w implementacji względem założeń projektowych**

- Szata graficzna projektów została zmieniona, ponieważ projekty zawarte w dokumentacji były jedynie szkicami i zamysłami samych interfejsów. Z tymi zawartymi w dokumentacji zgadza się struktura i zawarte informacje, lecz są one rozszerzone o dodatkowe pola oraz odpowiednio ostylowane.

- Nie wykonaliśmy testów wymagań metrycznych zawartych w dokumentacji, ponieważ nie mamy na to wystarczających środków. Testy zostałyby wykonane w momencie zakupu systemu przez firmę i odpowiedniego dofinansowania. Wykonaliśmy natomiast podstawowe testy. Odnosząc się do bezpieczeństwa sprawdziliśmy czy strona jest odporna na podstawowe i proste ataki (SQL Injection, Blind SQL).

- Resztę testów metrycznych wykonaliśmy bez zatrudniania dodatkowych osób. Przetestowaliśmy je wśród znajomych, a stabilność systemu sprawdziliśmy za pomocą maszyny do DDOS.

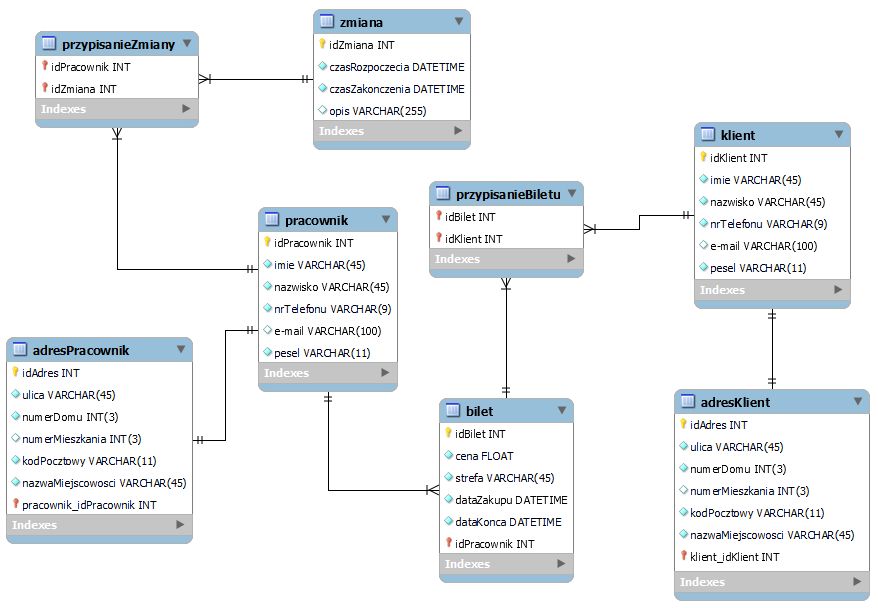
- Nie umieściliśmy wybranego w dokumentacji rozwiązania dla osoby niepełnosprawnej. Nie byliśmy w stanie zaimplementować wystarczających udogodnień dla osoby niedowidzącej w określonym czasie do oddania systemu.

- **System płatności (Doinformuj się)!**

Nie podłączyliśmy zewnętrznego systemu realizacji płatności do naszego systemu, ponieważ zostawiliśmy wybór współpracy dla firmy/klienta.

**3. Schemat zaimplementowanej bazy danych**

W systemie został zastosowany schemat bazy danych na podstawie schematu uwzględnionego wcześniej w dokumentacji. Struktury i relacje podanych danych pokrywają się z wersją pierwotną z odpowiednim zastosowaniem kluczy obcych i ich referencyjności.



**4. Specyfikacja środowiska realizacji i struktury aplikacji**

**Wybrane środowisko:**

- PyCharm  
- Visual Studio Code

**Backend:**

- Python

**Frameworki:**

- Django

**Frontend:**

- HTML

- CSS

- JavaScript

**Baza Danych:**

- SQLite

**System kontroli wersji:**

- GIT

**Repozytorium:**

- Github

Zdecydowaliśmy się na umieszczenie całego projektu na repozytorium github, ponieważ oferuje łatwą i szybką obsługę i wiele możliwości pracy zespołowej. Do zmian projektowych używamy systemu kontroli wersji GIT przy pomocy CLI jak i wbudowanych funkcji używanych w IDE. Jako, że w projekcie zdecydowaliśmy się na utworzenie backendu za pomocą Pythona i frameworka Django to wzorzec projektowy w jakim pracujemy to MVT (Model-View-Template)

W momencie gdy użytkownik żąda danego zasobu ze strony, Django działa jak kontroler i sprawdza dostępne zasoby dla podanego adresu URL. Wszystkie żądania aplikacji są przetwarzane na podstawie pliku konfiguracyjnego urls.py . Następnie framework zajmuje się parsowaniem adresu URL przychodzącego żądania i przekazuje dane do różnych modułów. W tym modułach znajdują się modele, widoki i szablony. Jeśli URL jest dostępny to wywoływany jest widok, który prowadzi interakcję z odpowiednim modelem i szablonem. Właśnie wtedy renderowany jest szablon, a następnie Django odpowiada użytkownikowi i wysyła dany szablon jako odpowiedź. W naszym przypadku szablony znajdują się w katalogu templates w postaci plików HTML i są one połączone z frontendem, który znajduje się w katalogu static. Do frontendu używamy jedynie CSS i JavaScript. Nie używamy żadnych dodatkowych frameworków. Nasz system komunikuje się z bazą danych SQLite za pomocą modeli, które są reprezentacją bazy danych w kodzie. Ich komunikacja polega na migracji danych z modelu do bazy, w której przechowywane są rekordy.

**5. Jak SI działa**

Uruchamiane moduły

Aaaaaa

Wykorzystywane biblioteki

Bbbbbb

Komunikacja z bazą danych

Cccccc

**6. Umiejscowienie aplikacji**

Nasza aplikacja jest umiejscowiona lokalnie. Poniżej znajdzie się instrukcja instalacji i uruchomienia aplikacji:

1. Na początku trzeba udać się na repozytorium github, na którym znajduje się aplikacja (<https://github.com/Vex0on/ICC_15_00/tree/main>) i pobrać aplikację. Istnieją na to dwie możliwości:
2. Bezpośrednio z strony githuba pobrać zawartość gałęzi main jako ZIP. Aby to zrobić wystarczy wcisnąć zielony przycisk (<> CODE), a następnie po rozwinięciu menu wybrać opcję (Download ZIP)
3. Druga opcja, która jest przez nas rekomendowana to pobranie zawartości repozytorium za pomocą systemu kontroli wersji GIT. Aby to zrobić wystarczy wcisnąć zielony przycisk (<> CODE), a następnie skopiowanie linku do lokalizacji wyświetlającej się w zakładce HTTPS.

Na komputerze tworzymy katalog, który będzie naszym lokalnym repozytorium i za pomocą wiersza poleceń inicjalizujemy repozytorium komendą:

**git init**

Następnie łączymy się z repozytorium lokalnym za pomocą komendy:

**git remote add origin <** **https://github.com/Vex0on/ICC\_15\_00.git >**

Następnie ściągamy zawartość repozytorium zdalnego do lokalnego poleceniem:

**git pull origin main**

**Informacje dostępu:**

Użytkownik:  
Recepcjonistka:  
Księgowa:  
Kierownik:  
Admin:  
Email firmy:

**7. Instrukcja Obsługi**

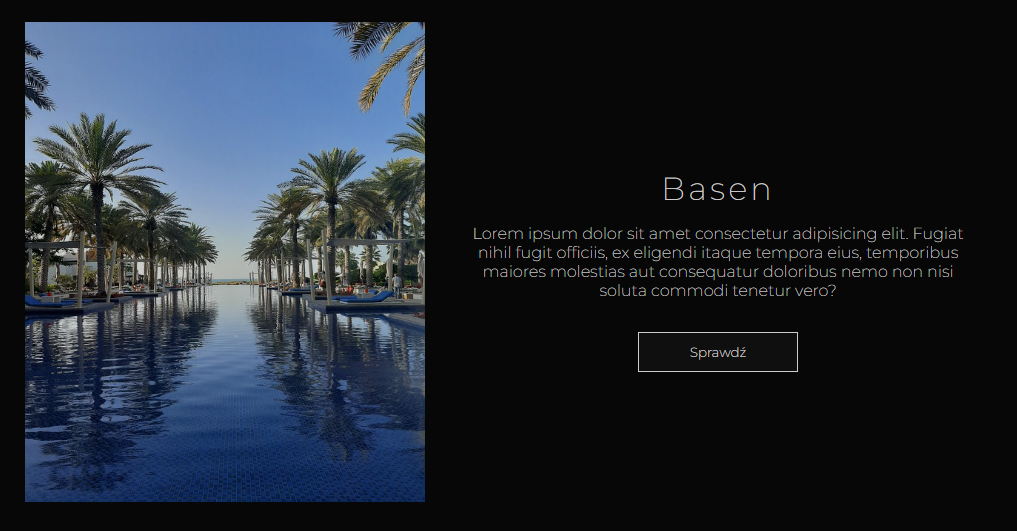
Po wejściu na stronę ukazuje nam się strona główna aplikacji. Na środku jest widoczna nazwa firmy, a na górze belka nawigacyjna.



Strona główna to rodzaj strony one-page. Oznacza to, że opcje dostępne w belce nie przeniosą użytkownika do oddzielnej podstrony, a jedynie zostanie on przeniesiony niżej do wybranej sekcji na tej samej stronie. Wyjątkiem jest „Zaloguj się”, które przenosi do osobnej podstrony z formularzem logowania oraz „Panel”, który jest dostępny jedynie dla zalogowanych użytkowników

Belka nawigacyjna:

1. **Basen/Siłownia/Spa** – Po kliknięciu użytkownik zostaje przekierowany do wybranej sekcji na stronie, która zawiera opis, zdjęcie obiektu i przycisk umożliwiający przejście do podstrony, na której znajduje się możliwość zakupienia biletu przypisanego do tej strefy. Aby zakupić bilet użytkownik musi wcisnąć przycisk „Sprawdź/Kup Bilet”.



1. **Kupowanie Biletu** – Po przejściu do podstrony danej sekcji trafiamy na formularz, który użytkownik musi wypełnić swoimi danymi, aby móc zakupić bilet. Poniżej danych znajduje się możliwość dostosowania swojego zakupu czyli wybór rodzaju biletu. Dla każdej strefy dostępny jest bilet normalny jak i zniżka ulgowa lub grupowa. Do tego użytkownik może wybrać datę i godzinę o której bilet będzie aktywny. Jako, że bilet jest całodobowy to nie ma potrzeby wyboru daty zakończenia. Po wypełnieniu formularza użytkownikowi pozostaje wciśnięcie przycisku „Kup Bilet”

Obraz zawierający tekst, monitor, czarny, ekran

Opis wygenerowany automatycznie

1. **O nas** - Po kliknięciu użytkownik zostaje przekierowany do sekcji zawierającej krótki opis oferty firmy.
2. **Kontakt** – Po kliknięciu użytkownik zostaje przekierowany do sekcji kontaktowej podzielonej na dwie części. W lewej części znajduje się formularz kontaktowy z polami „Imię i nazwisko”, „Email” oraz „Wiadomość”. Po ich wypełnieniu i wciśnięciu przycisku „Wyślij” użytkownik przesyła wiadomość na maila firmowego.

W prawej części znajdują się informacje kontaktowe firmy takie jak adres, numer telefonu czy adres email.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Zaloguj się –

Rejestracja –

Zapomniałeś hasła –

Reklamacja –

Panel użytkownika:

Użytkownik

Recepcjonistka

Księgowa

Kierownik

Admin