

Inżynieria oprogramowania

Bohdan Szymczak

Projekt

v. 5.2

Opis zadań do realizacji w ramach projektu

Symbol: IO52_P

Spis treści

Przedmiot projektu.....	3
Warunki zaliczenia.....	3
Opcje realizacji projektu.....	3
Produkty projektu.....	3
Bibliografia.....	5
Wymagane narzędzia	5
Przygotowanie do wykonania zadań Projektu	5
Opis Projektu	6
1 Wstęp – źródła danych.....	7
1.1 Wiedza ogólna.....	7
1.2 Opis wymagań do oferty na wykonanie oprogramowania	7
1.3 Wywiad analityka z użytkownikiem	7
2 Analiza	11
2.1 Interpretacja i doprecyzowanie wymagań w fazie uzgodnień.....	11
2.2 Specyfikacja wymagań	11
2.3 Model logiczny bazy danych	12
2.4 Hierarchia funkcji	13
2.5 Prototyp interfejsu użytkownika	13
3 Projekt.....	14
3.1 Projektowanie bazy danych	14
3.2 Projektowanie interfejsu użytkownika	14
4 Implementacja.....	15
5 Testy.....	16
5.1 Planowanie testów.....	16
5.2 Przygotowanie testów.....	17
6 Ocena projektu.....	18

Przedmiot projektu

Wykonanie analizy wymagań użytkownika, projektu technicznego i prototypu oraz przygotowanie testów aplikacji bazodanowej pracującej w środowisku okienkowym systemu Windows.

Warunki zaliczenia

Zaliczenie projektu wymaga uzyskania minimum 60 punktów.

Projekt powinien być złożony do oceny najpóźniej do wskazanej na platformie daty. Prace przekazane po terminie będą miały obniżoną ocenę. Po dacie granicznej, która przypada na tydzień przed rozpoczęciem sesji egzaminacyjnej platforma nie przyjmie pracy i nie będzie ona oceniana.

Prace można składać wyłącznie przez platformę. Prace przesłane e-mailem nie będą oceniane.

Opcje realizacji projektu

1. Realizacja indywidualna – projekt jest realizowany przez **pojedynczą osobę**
2. Realizacja zespołowa – projekt jest realizowany przez **zespół maksymalnie czteroosobowy**. W tym przypadku:
 - a) Sprawozdanie i pozostałe składniki projektu muszą być sygnowane danymi wszystkich członków zespołu – w odpowiednich miejscach muszą znaleźć się dane wszystkich osób realizujących projekt
 - b) Każdy z członków zespołu przesyła na platformę wykonany projekt oznaczając plik przekazania indywidualnymi danymi (nazwa pliku .zip zawiera dane pojedynczej osoby)
 - c) Zawartość przekazanego przez każdego członka zespołu projektu jest identyczna
 - d) Każdy z członków zespołu musi zrealizować część każdego punktu, należy podzielić się pracą w ramach każdej części projektu np. każda osoba opisuje rozdzielną część encji, osobne gałęzie hierarchii funkcji itp.
 - e) Każdy z członków zespołu zna cały projekt i potrafi wyjaśnić szczegółowo każdą jego część.

Uwaga

Rekomendowaną opcją realizacji projektu jest wykonanie go przez zespół.

Wykonanie grupowe powinno obejmować dyskusję członków zespołu nad projektem, ustalenie szczegółów realizacyjnych oraz wewnętrzny podział pracy.

Studenci samodzielnie ustalają skład grupy oraz organizują własną pracę.

Produkty projektu

Produktami zadania są:

1. Dokument **PROJEKT.PDF** – jest to sprawozdanie z realizacji projektu, które musi zawierać opis wykonania poszczególnych zadań. Szczegółowe wymagania dla poszczególnych punktów znajdują się w niniejszej instrukcji. Dokument zwany dalej sprawozdaniem utwórz w dowolnym edytorze tekstu np. z pakietu LibreOffice, a gotową wersję przekonwertuj na format **PDF**. Dokument musi posiadać następującą strukturę:
 - a) Analiza
 - Slajdy z prezentacji

- Specyfikacja wymagań
- Zestawienie encji
- Hierarchia funkcji
- Prototyp interfejsu użytkownika
- b) Projekt
 - Diagram ERD
 - Zestawienie formatek
- c) Implementacja
 - Zestawienie nazw plików SQL
- d) Testy
 - Scenariusze testów
 - Skrypty generujące dane
 - Wyniki działania poleceń SQL
- 2. Plik **.mwb** zawierający diagram ERD wykonany przy użyciu **MySQL Workbench**.
- 3. Plik zawierający skrypt **SQL** tworzący wszystkie potrzebne do działania aplikacji struktury bazodanowe
- 4. Pliki zawierające skrypty **SQL** tworzące dane testowe

Dla ułatwienia, informacje, które powinny znaleźć się w sprawozdaniu zostały wyróżnione w tekście. W każdym punkcie znajdują się instrukcje, jak należy go wykonać, np. polecenia: przeanalizuj, zaproponuj, zidentyfikuj itp. To są odpowiedzi dla Ciebie, które nie wymagają pisemnej relacji w sprawozdaniu. Jedynie polecenia wyróżnione należy opracować w sprawozdaniu.

Uwaga

1. Sprawozdanie musi być opracowane w sposób przejrzysty i czytelny. Forma dokumentacji projektowej jest równie ważna jak jego treść. Wydziel w dokumencie poszczególne części, numerując je zgodnie z treścią wymagań projektu. Stosuj w dokumencie formatowanie ułatwiające percepcję treści w postaci wyróżnień, wyliczeń, tabel itp. Na pierwszej stronie podaj swoje dane: Imię, Nazwisko, nr indeksu, wydział, grupa, rok akademicki, semestr. Do tego celu wykorzystaj szablon znajdujący się w dokumencie **Opracowanie zadań – instrukcja.pdf**
2. Opracuj starannie skrypty i zamieść w nim wszystkie wymagane elementy opisane w dokumencie „**Opracowanie zadań - instrukcja.pdf**”.
3. Rygorystycznie przestrzegaj specyfikacji wymagań, określonej w treści zadań.
4. Nazywaj czytelnie encje, atrybuty, tabele i kolumny.

*Wszystkie pliki przed wysłaniem do sprawdzenia należy spakować do jednego archiwum **zip** i oznaczyć specyficzną nazwą. Szczegółowe wytyczne znajdują się w dokumencie **Opracowanie zadań – instrukcja.pdf***

Uwagi

- Pamiętaj, aby program (skrypt) miał wszystkie standardowe elementy, które są wymienione w dokumencie „**Opracowanie zadań – instrukcja.pdf**”.
- Pamiętaj, żeby opracowanie było zgodne z szablonem umieszczonym w dokumencie „**Opracowanie zadań – instrukcja.pdf**”.
- Wykonany projekt opracuj przy użyciu narzędzi z dowolnego pakietu biurowego, a następnie skonsoliduj w jednym dokumencie (typu doc, docx) i skonwertuj na format **pdf**.
- Dokument **pdf** możesz wygenerować bezpośrednio z darmowego programu LibreOffice Writer.
- Nie dołączaj do przekazania plików, które nie są wymagane

Bibliografia

1. Kamil Stokwiszewski, Projektowanie systemów i inżynieria oprogramowania (Biblioteka PUW).
2. Krzysztof Sacha, Inżynieria oprogramowania, PWN, 2021
3. Ian Sommerville, Inżynieria oprogramowania, PWN, 2020
4. https://wiki.lazarus.freepascal.org/Lazarus_Documentation
5. <https://dev.mysql.com/doc/>

Wymagane narzędzia

1. Motor bazy danych MySQL – oprogramowanie dostępne na zasadach GNU General Public License (GPL) do pobrania ze strony <https://dev.mysql.com/downloads/mysql/>
2. Narzędzie do modelowania danych MySQL Workbench - oprogramowanie dostępne na zasadach GNU General Public License (GPL) do pobrania ze strony <https://dev.mysql.com/downloads/workbench/>
3. Narzędzie do tworzenia aplikacji typu RAD, Lazarus - oprogramowanie dostępne na zasadach GNU General Public License (GPL) do pobrania ze strony <https://www.lazarus-ide.org/> (opcjonalne)
4. **LibreOffice**, lub ekwiwalentny pakiet oprogramowania biurowego.

Przygotowanie do wykonania zadań Projektu

1. Zainstaluj motor bazy danych MySQL.
2. Zainstaluj MySQL Workbench
3. Zainstaluj oprogramowanie Lazarus (opcjonalne)

Opis Projektu

W kolejnych punktach znajduje się szczegółowy opis projektu i zadań do wykonania. Zapoznaj się z jego treścią.

Część wstępna zawiera informacje, jakie analityk uzyskał z różnych źródeł, lub posiada w ramach tzw. wiedzy powszechnej oraz zapis jego wywiadu z użytkownikiem. Ta wiedza jest podstawą do wykonania analizy będącej początkiem procesu tworzenia systemu informatycznego **metodą strukturalną**.

Twoim zadaniem jest przeprowadzenie procesu produkcji aplikacji zgodnie z zasadami inżynierii oprogramowania przy wykorzystaniu **metody strukturalnej**. Musisz wykonać następujące kroki:

1. Analizę wymagań użytkownika
2. Prototyp interfejsu użytkownika
3. Projekt bazy danych i aplikacji
4. Opracowanie scenariuszy testowych
5. Opracowanie skryptów generujących obiekty bazodanowe i dane testowe
6. Przygotowanie bazy danych do testów

Poszczególne kroki do wykonania są opisane w kolejnych punktach niniejszej instrukcji.

Uwaga

*Celem uniknięcia problemów interpretacyjnych przy ocenie projektu należy posługiwać się terminologią umieszczoną w danych źródłowych i konsekwentnie stosować ją we wszystkich częściach projektu. **Oznacza to obowiązek używania nazw wymagań, encji, atrybutów, związków, tabel itp. w języku polskim zarówno w projekcie, jak i w skryptach SQL.***

1 Wstęp – źródła danych

1.1 Wiedza ogólna

Korzystanie z usług hotelarskich jest powszechne. Najczęściej stosowaną metodą rezerwacji pokoju jest usługa internetowa lub bezpośredni kontakt telefoniczny.

1.2 Opis wymagań do oferty na wykonanie oprogramowania

Właścicielka niewielkiego pensjonatu położonego w turystycznej miejscowości nad morzem wynajmuje pokoje gościnne. Dotychczas nie posiadała żadnego systemu obsługi rezerwacji. Posługiwała się klasycznym kalendarzem papierowym. Ten sposób jest podatny na błędy.

Analogicznie, w papierowej księdze gości, rejestrowała pobyty osób wynajmujących pokoje.

Obecnie postanowiła zamówić system informatyczny, którego celem ma być usprawnienie procesu rezerwacji, oraz rejestracji pobyków. Po konsultacji z informatykiem spisała swoje oczekiwania w formie Specyfikacji Istotnych Warunków Zamówienia.

W kolejnym kroku analityk przeprowadził wywiad w celu zebrania szczegółowych informacji potrzebnych do wyspecyfikowania wymagań.

Wyciąg ze Specyfikacji Istotnych Warunków Zamówienia:

1. System musi wspierać proces rezerwacji pokoi
2. System musi prowadzić rejestr pobyków gości
3. System rezerwacji będzie obsługiwany przez użytkownika na podstawie komunikacji z wynajmującym poprzez e-mail lub telefon
4. Dane muszą być składowane w relacyjnej bazie danych
5. Interfejs użytkownika powinien być wykonany w postaci aplikacji okienkowej systemu Windows

1.3 Wywiad analityka z użytkownikiem

Analitik:

- Jak wynika z dokumentu SIWZ naszym zadaniem jest zbudowanie systemu informatycznego typu call centre obsługiwanego przez operatorów przyjmujących rezerwacje drogą telefoniczną lub poprzez e-mail. Na wstępie dobrze byłoby rozstrzygnąć czy serwery będą lokalne, czy skorzystamy z usługi chmury obliczeniowej.

Użytkownik:

- Obydwa rozwiązania są przewymiarowane i niepotrzebne, przynajmniej na tym etapie. Mój pensjonat liczy obecnie osiemnaście pokoi. Wystarczy program na pojedynczym komputerze.

Analitik:

- Czy to oznacza, że system będzie obsługiwany jednocześnie tylko przez jednego użytkownika?

Użytkownik:

- Tak.

Analitik:

- Proszę opisać proces rezerwacji pokoju.

Użytkownik:

- Najczęściej telefonuje wynajmujący i pyta o dostępność pokoju dla określonej liczby osób w interesującym go terminie. Wtedy sprawdzam, czy dysponuję wolnym pokojem o oczekiwanych parametrach i jeśli taki istnieje, to po potwierdzeniu ceny dokonuję wstępnej rezerwacji, która

jest ważna określony czas. W tym czasie wynajmujący musi przelać zaliczkę. Kiedy potwierdzi wpływ pieniędzy, dokonuje rezerwacji obowiązującej.

Analitik:

- A co się dzieje, jeśli zaliczka nie wpłynie.

Użytkownik:

- Rezerwacja jest anulowana.

Analitik:

- Jak wygląda rezerwacja e-mailem?

Użytkownik:

- Tak samo, tyle że uzgodnienia są prowadzone w formie korespondencji.

Analitik:

- W jaki sposób następuje wybór pokoju do rezerwacji?

Użytkownik:

- Pokoje mają różne parametry, które decydują o cenie wynajmu. Jest to położenie i liczba miejsc noclegowych.

Analitik:

- Co oznacza położenie?

Użytkownik:

- Niektóre pokoje są z widokiem na morze. Również ważne jest piętro.

Analitik:

- Czy to jest wszystko, co decyduje o cenie wynajmu?

Użytkownik:

- Nie. Cena zależy od terminu – w sezonie ceny są oczywiście wyższe. Natomiast jeśli w pokoju będzie mieszkać mniej osób, niż jest miejsc noclegowych, cena będzie zmniejszona. Również stali klienci otrzymują zniżki.

Analitik:

- Jaka jest definicja stałego klienta?

Użytkownik:

- To osoba, która co najmniej raz wynajmowała wcześniej pokój.

Analitik:

- Jak jest identyfikowany taki klient?

Użytkownik:

- Klienci są rejestrowani w zeszycie. Zapisuję tam ich dane oraz historię wynajmu pokoi.

Analitik:

- Jak rozumiem, pokój wraca do puli wolnych pokoi kiedy wynajmujący opuści pensjonat?

Użytkownik:

- Tak jest na ogół. Zdarza się jednak, że pomimo rezerwacji klient nie przyjeżdża. W takim przypadku pokój pozostaje zarezerwowany przez jedną dobę, koszt pokrywa zaliczka. Od następnego dnia może być wynajęty innej osobie.

Analitik:

- Myślę, że proces rezerwacji jest dla mnie jasny. Jakie są oczekiwania wobec systemu informatycznego? Jak funkcjonalność powinna być zaimplementowana.

Użytkownik:

- Oczekuję ułatwienia procesu rezerwacji. System powinien pokazywać wolne pokoje o określonych parametrach we wskazanym przedziale czasu. System powinien rejestrować rezerwacje i korygować je w przypadku niewpłacenia zaliczki. Ważne jest również dla mnie, żeby System zapamiętywał informacje o wynajętych pokojach, daty przyjazdu i odjazdu gości, pobrane opłaty za pobyt.

Analitik:

- Czyli, oprócz obsługi rezerwacji System musi zapamiętywać dane klientów i historię ich pobyków.

Użytkownik:

- Tak.

Analitik:

- Czy pokój może być wynajęty bez wcześniejszej rezerwacji?

Użytkownik:

- Tak. Zdarza się, że do pensjonatu zgłasza się gość, jak to mówimy „z ulicy”, i w przypadku, gdy jest odpowiedni dla niego pokój wynajmuje go „od ręki”.

Analitik:

- Jak przebiega wtedy proces najmu?

Użytkownik:

- Bardzo prosto. Zapisuję dane klienta w zeszycie, zapisuję czas pobytu i pobieram opłatę.

Analitik:

- Proszę przypomnieć, ile jest pokoi?

Użytkownik:

- Na obecną chwilę osiemnaście, ale w przyszłości może to się zmienić. Nie wykluczam przebudowy bądź rozbudowy budynku.

Analitik:

- Rozumiem, ale obecnie wszystkie osiemnaście jest w użyciu?

Użytkownik:

- To też ulega zmianie. Okresowo niektóre pokoje nie są dostępne, np. z powodu remontu.

Analitik:

- Podsumuję. System ma służyć rejestracji rezerwacji i wynajmu. Czy to się zgadza?

Użytkownik:

- Tak.

Analitik:

- Czy istnieją jeszcze jakieś szczególne wymagania?

Użytkownik:

- Tak. Proces rezerwacji i uzgodnień terminu najmu powinien być bardzo sprawny. Rozmowa przez telefon powinna być krótka, więc potrzebne informacje muszę otrzymywać bardzo efektywnie. Często jest tak, że nie można znaleźć terminu oczekiwanego przez klienta, gdyż są już inne rezerwacje. Wtedy muszę znaleźć i zaproponować termin zbliżony lub inny, ale droższy pokój.

Analitik:

- Podsumuję. Należy wykonać system, który gromadzi dane o pokojach i klientach, i w oparciu o kalendarz wspiera proces rezerwacji i najmu pokoi. Szczególne znaczenie ma ergonomia i efektywność działania interfejsu użytkownika w procesie znajdowania odpowiedniego pokoju o wskazanych parametrach dla określonego przedziału czasu. Czy to się zgadza?

Użytkownik:

- Tak. Na dziś jest to wszystko czego oczekuję. W przyszłości system będzie rozbudowany tak, żeby współpracował z innymi zewnętrznymi systemami np. rezerwacji internetowej i z systemem księgowym wystawiającym rachunki za pobyt. Ale to plany na przyszłość.

Analitik:

- Kto będzie obsługiwał system?

Użytkownik:

- Głównie ja, ale również inni pracownicy.

Analitik:

- Czy potrzebna jest rejestracja, który użytkownik dokonał zapisów w systemie, np. przeprowadził rejestrację?

Użytkownik:

- Nie jest to konieczne, ale może być przydatne w przyszłości, jeśli rozszerzę działalność.

Analitik:

- Rozumiem. Na tym kończę wywiad.

2 Analiza

2.1 Interpretacja i doprecyzowanie wymagań w fazie uzgodnień

Na podstawie informacji z punktu 1. zidentyfikuj wymagania użytkownika. Zaproponuj architekturę systemu i zdefiniuj wymaganą funkcjonalność. Określ funkcjonalności wymagane i opcjonalne.

Wskaż możliwe rozwiązania dla przyszłej rozbudowy systemu.

Rozważ wymagania pozafunkcjonalne dotyczące np. ochrony danych.

Zwróć uwagę na fakt, że użytkownik opisał dość ogólnie swoje wymagania, zatem oczekuje się, że analityk przygotuje rozwinięcie wymagań i zaproponuje własne rozwiązania, które w procesie dalszych uzgodnień zostaną zaakceptowane przez użytkownika.

Znajdź zagadnienia, które nie zostały precyzyjnie określone w SIWZ i w wywiadzie. Umieść je w prezentacji w formie pytania do użytkownika, które ma wyjaśnić szczegóły tych zagadnień.

Wykonaj przy pomocy narzędzia do prezentacji np. LibreOffice Impress lub Microsoft PowerPoint prezentację dla użytkownika, która będzie stanowić formę weryfikacji poziomu zrozumienia systemu przez strony. Prezentacja powinna zawierać elementy opisane powyżej, zagadnienia do wyjaśnienia, propozycje potencjalnych rozszerzeń i krótką prezentację realizatorów (element marketingowy).

- Przygotuj prezentację na zajęcia. Bądź gotów/gotowa do osobistego przeprowadzenia prezentacji oraz odpowiadania na pytania prowadzącego i grupy studenckiej
- Umieść w dokumencie sprawozdania poszczególne strony prezentacji

2.2 Specyfikacja wymagań

Na podstawie informacji z punktu 1 i prezentacji systemu sporządź formalną specyfikację wymagań funkcjonalnych. Użyj w tym celu wzorca specyfikacji podanego na wykładzie. Poszczególne wymagania szczegółowe opisz w formie poniższej tabeli:

Numer wymagania	
Nazwa wymagania	
Opis wymagania	
Dane wejściowe	
Źródło danych	
Opis funkcji	
Opis bazy danych	
Wymagania pozafunkcjonalne	

Zamieść w sprawozdaniu formalną specyfikację wymagań.

- Na podstawie opisu specyfikacji podanej na wykładzie i literatury wykonaj kompletną specyfikację zawierającą: wstęp, opis ogólny, wymagania szczegółowe, słownik pojęć.

2.3 Model logiczny bazy danych

Zidentyfikuj encje i związki między encjami. Zidentyfikuj identyfikatory poszczególnych encji, a jeśli nie istnieje identyfikator naturalny, to zaznacz, że encja będzie miała identyfikator sztuczny.

Każda encja powinna zawierać następujące elementy:

- 1) Nazwa encji
- 2) Identyfikator encji
- 3) Główne atrybuty i ich charakterystyka.:
 - nazwa
 - dziedzina:
 - typ danych i maksymalny rozmiar
 - zbiór dozwolonych wartości
 - zakres dozwolonych wartości
 - obowiązkowość (czy dozwolona wartość pusta)
 - unikalność
- 4) Opis encji (krótki komentarz charakteryzujący encję)
- 5) Związki (relacje encji z innymi encjami)

Przykład encji OSOBA

Encja	OSOBA						
Opis	Osoba fizyczna zarejestrowana w bazie PESEL						
	Nazwa	Dziedzina			Obowi ązkow ość	Unikal ność	Uwagi
		Typ danych i max.rozmiar	Zbiór dozwolonych wartości	Zakres dozwolonych wartości			
Identyfikat or	PESEL	Liczba całkowita jedenastocyfrowa	Warunek walidacji: <ul style="list-style-type: none"> • 11 cyfr • Poprawna suma kontrolna (algorytm) 	00000000000- 99999999999	Tak	Tak	
Atrybuty	Nazwisko	Łańcuch znaków, 30	Litery łacińskie i polskie znaki diakrytyczne	Od 'a' do 'ż...ż'	Tak	nie	
	Imię_1	Łańcuch znaków, 20	Litery łacińskie i polskie znaki diakrytyczne	Od 'a' do 'ż...ż'	Tak	Nie	
	Imię_2	Łańcuch znaków, 20	Litery łacińskie i polskie znaki diakrytyczne	Od 'a' do 'ż...ż'	Nie	Nie	
	Data_ur	Data	Data	Od 01-01-1900 do data bieżąca	Tak	nie	
Związki	Zamieszka nie	Zamieszkuje w MIEJSCOWOŚĆ					

Umieść w sprawozdaniu tabelaryczne zestawienie wszystkich encji wzorując się na powyżej zamieszczonym przykładzie.

Uwaga

Ponieważ w projekcie nie jest używane żadne zaawansowane narzędzie umożliwiające transformację modelu logicznego na model fizyczny, celem uniknięcia powtarzania pracy, możesz pominąć w zestawieniu encji wszystkie atrybuty, dla których reguły poprawności wynikają wprost z deklaracji typu w bazie danych.

Wszystkie atrybuty muszą być poprawnie opisane w definicji tabeli w bazie danych.

Opis encji i jej identyfikatora (klucza głównego) są obowiązkowe.

Opis związków jest obowiązkowy; nie jest wymagane opisywanie relacji na diagramie ERD.

2.4 Hierarchia funkcji

Na podstawie specyfikacji wymagań zidentyfikuj funkcje elementarne i złożone.

Umieść w sprawozdaniu *hierarchię funkcji* – może być w postaci diagramu albo listy zagnieżdżonej

2.5 Prototyp interfejsu użytkownika

Interfejs użytkownika jest w projektowanym systemie kluczowy dla efektywnej pracy. Zaprojektuj ekran roboczy rezerwacji pokoju, jako formatkę okienkową systemu Windows. Formatka ma zapewnić efektywny sposób znalezienia odpowiedniego pokoju i jego rezerwację podczas rozmowy z klientem.

Wykonaj prototyp przy użyciu dowolnego narzędzia do modelowania lub RAD np. Lazarusa. Nie twórz żadnego kodu wykonywalnego. Umieść na formatce odpowiednie komponenty Windows.

Umieść i opisz w sprawozdaniu interfejs użytkownika.

- *Umieść w sprawozdaniu zrzut ekranu zawierający czytelny obraz prototypu.*
- *Opisz ogólną funkcjonalność formatki*
- *Opisz przeznaczenie poszczególnych komponentów*

3 Projekt

3.1 Projektowanie bazy danych

Dokonaj transformacji encji na tabele i sprowadź bazę do trzeciej postaci normalnej. Wykonaj przy użyciu MySQL Workbench diagram ERD. W szczególności:

1. Wprowadź wszystkie tabele
2. Zdefiniuj klucze główne
3. Określ typy danych w poszczególnych kolumnach
4. Zdefiniuj kolumny NOT NULL i NULL
5. Zdefiniuj relacje pomiędzy tabelami
6. Rozwiąż relacje typu wiele do wielu
7. Zdefiniuj potrzebne perspektywy (widoki) o ile takie zaprojektujesz.

Umieść w sprawozdaniu projekt bazy danych:

- Zamieść w sprawozdaniu obraz diagram ERD
- Dołącz do przekazania plik **.mwb** zawierający diagram ERD wykonany przy użyciu **MySQL Workbench**

UWAGA

Diagram musi być sporządzony przy użyciu następujących opcji MySQL Workbench:

- Model/Object Notation/Workbench (Default)
- Model/Relationship Notation/Crow's Foot (IE)
- Obraz diagramu ERD, potrzebny do umieszczenia w sprawozdaniu, wykonaj przy użyciu opcji File/Export/Export as PNG

3.2 Projektowanie interfejsu użytkownika

1. Wykonaj zestawienie wszystkich formatek (ekranów), które będą zaimplementowane w formie następującej tabeli:

Lp	Nazwa formatki	Przeznaczenie	Krótki opis działania

Umieść w sprawozdaniu tabelaryczne zestawienie formatek.

4 Implementacja

Wykonaj stronę bazodanową aplikacji w następujących krokach:

1. Wygeneruj w narzędziu MySQL Workbench skrypt tworzący w bazie danych obiekty zdefiniowane na diagramie ERD
2. Wygeneruj obiekty bazodanowe przy użyciu skryptu

Dołącz do przekazania skrypty SQL w formie pliku z rozszerzeniem .sql.

W sprawozdaniu umieść zestawienie zawierające nazwy plików i opis ich zawartości.

UWAGA

Wykonanie kompletnej aplikacji przekracza dostępne ramy czasowe przedmiotu, dlatego zostanie pominięte.

5 Testy

5.1 Planowanie testów

1. Opracuj scenariusze testowe aplikacji weryfikujące poprawność implementacji wyspecyfikowanych wymagań
2. Każdy scenariusz musi zawierać:
 - Symbol testu
 - Numer i nazwę wymagania
 - Cel testu
 - Sposób przeprowadzenia testu
 - Sposób sprawdzenia - oczekiwany rezultat
 - Dane do testu

Przykład

Symbol testu	KU-001
<u>Nr wymagania</u>	10
<u>Nazwa wymagania</u>	Edycja danych osobowych
<u>Cel testu</u>	Sprawdzenie kompletności wprowadzania danych na formatce
<u>Sposób wykonania</u>	Wypełnić pola na formatce i zatwierdzić. Pomiąć poszczególne pola.
<u>Sposób sprawdzenia</u>	Obserwacja zachowania formatki. Weryfikacja braku zapisu do tabeli OSOBY.
<u>Oczekiwane działanie</u>	Brak wymaganych danych powinien skutkować wyświetleniem komunikatu błędu. W tabeli OSOBY nie może pojawić się zapis.
<u>Dane wejściowe</u>	Nazwisko: Kowalski Imię: Jan Data urodzenia: 2001-10-28
<u>Wynik testu</u>	
<u>Uwagi testera</u>	

Symbol testu	KU-002
<u>Nr wymagania</u>	10
<u>Nazwa wymagania</u>	Edycja danych osobowych
<u>Cel testu</u>	Sprawdzenie poprawności zapisu do bazy
<u>Sposób wykonania</u>	Wypełnić wszystkie pola na formatce. Odświeżyć zawartość komponentu na formatce.
<u>Sposób sprawdzenia</u>	Po odświeżeniu zawartości prezentowane są wprowadzone dane. Bezpośrednie zapytanie do bazy przy użyciu SQL potwierdza obecność danych w tabeli.
<u>Oczekiwane działanie</u>	Wprowadzenie poprawnych danych powinno skutkować zapisem do tabeli OSOBY
<u>Dane wejściowe</u>	Nazwisko: Kowalski Imię: Jan Data urodzenia: 2001-10-28
<u>Wynik testu</u>	
<u>Uwagi testera</u>	

Wzorując się na powyżej zamieszczonym przykładzie umieść w sprawozdaniu scenariusze testów.

Uwaga

Nie jest wymagane opracowanie wszystkich scenariuszy. Do zaliczenia tego punktu wystarczy opracować:

- **5 przykładowych scenariuszy przy realizacji projektu przez jedną osobę**
- **10 przykładowych scenariuszy przy realizacji projektu przez zespół.**

Zauważ, że do jednego wymagania można zaplanować więcej niż jeden test.

5.2 Przygotowanie testów

1. Przygotuj skrypty SQL wprowadzające do bazy danych pakiet danych testowych przy pomocy ciągu poleceń:

```
insert into TABELA
values
```

2. Wykonaj w bazie danych przygotowane skrypty
3. Sprawdź zawartość tabel, do których zostały wprowadzone dane testowe wykonując polecenie:

```
select * from TABELA;
```

Umieść w sprawozdaniu:

- *Zawartość skryptów generujących dane*
- *Wyniki działania poleceń: select * from TABELA;*

Dołącz do przekazania skrypty SQL.

UWAGA

- *Pamiętaj o właściwej kolejności wprowadzania danych do tabel testowych uwzględniając istniejące między nimi relacje.*
- *Dane testowe powinny obejmować co najmniej wprowadzenie: klientów, rezerwacji, pobyków, danych pokoi.*
- *Nie umieszczaj w sprawozdaniu zrzutów ekranów z wynikiem działania polecenia select, zamiast tego wykorzystaj np. opcję export do pliku txt. Opcja jest dostępna jako „export/import” w oknie wyjściowym „result grid” MySQL Workbench.*

6 Ocena projektu

Projekt jest oceniany punktowo. Maksymalna liczba punktów do uzyskania wynosi 100. Zaliczenie projektu wymaga uzyskania minimum 60 punktów, przy czym każda z czterech części projektu musi być wykonana. Szczegółowa punktacja umieszczona jest w poniższej tabeli:

Część	Sekcja	Punkty	Max. Punktów
Analiza	Prezentacja	10	55
	Specyfikacja wymagań - część ogólna	5	
	Specyfikacja wymagań - część szczegółowa	15	
	Encje z atrybutami	10	
	Hierarchia funkcji	5	
	Prototyp	10	
Projekt	Diagram ERD z kolumnami i więzami	12	18
	Zestawienie formatek	6	
Implementacja	Generacja obiektów bazodanowych	3	3
Testy	Planowanie testów (5 lub 10 poz.)	10	19
	Przygotowanie testów	9	
Całość	Ocena czytelności i zgodności z formalnymi wymaganiami od -5 do +5		5
		5	
		RAZEM:	100

Ocena czytelności i formy sprawozdania oraz formalnych wymagań na program i przekazanie ma rozszerzoną skalę **od -5 do +5** punktów.

Pod uwagę brana jest forma opracowania dokumentu projektowego, jego przejrzystość i czytelność, a także zgodność z formalnymi wymaganiami przekazanych plików.