

Київський національний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

НА ТЕМУ:

**Розробка програмного забезпечення провайдерської
компанії «Версус»**

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

підпис студента

Гудименко Артема
Леонідовича

Науковий керівник
кандидат технічних наук,
доцент

підпис керівника

Сашнєова Мар'яна
Василівна

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

КИЇВ – 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки

Криворучко О. В.

"20" жовтня 2020 р.

Завдання

на випускню кваліфікаційну роботу студентів

Гудименку Артему Леонідовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи Розробка програмного забезпечення провайдерської компанії «Версус»

Затверджена наказом ректора від "30" жовтня 2020 р. № 3225

2. Строк здачі студентом закінченої роботи _____

3. Цільова установка та вихідні дані до роботи

Мета роботи – розробка функціоналу білінгової системи для першої лінії телефонної підтримки та відділу підключення нових клієнтів, враховуючи сучасні потреби ринку

Об'єкт дослідження – компанія «Версус», котра надає телекомунікаційні послуги

Предмет дослідження – аналіз та вивчення методів створення сучасних білінгових систем, їх структуризація

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО БІЛІНГОВІ СИСТЕМИ

Поняття білінгової системи, аналіз існуючих сучасних білінгових систем

Мета розробки білінгової системи

Технічне завдання

Висновок до розділу 1

РОЗДІЛ 2 АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ ТА ХАРАКТЕРИСТИКА
БАЗОВОЇ ОРГАНІЗАЦІЇ КОМПАНІЇ «ВЕРСУС»

Моделювання аналізу та методології проектування

Модель взаємовідносин(модель ER)

Методики розробки білінгових систем

Алгоритмізація розробки проекту

Опис алгоритму розробки

Діаграми для моделювання білінгової системи

Система контролю і управління доступом

Висновок до розділу 2

РОЗДІЛ 3 РОЗРОБКА ВЕБ ДОДАТКУ УПРАВЛІННЯ БІЛІНГОВОЮ СИСТЕМОЮ
КОМПАНІЇ «ВЕРСУС»

Методика випробувань і результати експериментальної перевірки

Опис ключових етапів розробки інформаційної системи

Інтерфейс картки абонента

Принцип роботи

Плагін

Висновки до розділу 3

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	21.09.2020	21.09.2020
2.	<i>Вступ та перелік літературних джерел</i>	14.12.2020	14.12.2020
3.	<i>Розділ 1. «ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО БІЛІНГОВІ СИСТЕМИ»</i>	19.02.2021	19.02.2021
4.	<i>Розділ 2. «АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ ТА ХАРАКТЕРИСТИКА БАЗОВОЇ ОРГАНІЗАЦІЇ КОМПАНІЇ «ВЕРСУС»»</i>	05.03.2021	05.03.2021
5.	<i>Розділ 3. «РОЗРОБКА ВЕБ ДОДАТКУ УПРАВЛІННЯ БІЛІНГОВОЮ СИСТЕМОЮ КОМПАНІЇ «ВЕРСУС»»</i>	10.04.2021	10.04.2021
6.	<i>Висновки</i>	24.04.2021	24.04.2021
7.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	14.05.2021	14.05.2021
8.	<i>Підготовка автореферату та презентації доповіді</i>	17.05.2021	17.05.2021
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>	18.05.2021 - 21.05.2021	
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.06.2021	
11.	<i>Здача прошого випускної кваліфікаційної роботи на кафедру</i>	02.06.2021	
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «20» жовтня 2020 р.

8. Науковий керівник випускної кваліфікаційної роботи
Сашньова М.В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Цензура М.О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Гудименко А. Л.

11. Відгук керівника випускної кваліфікаційної роботи

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Науковий керівник випускної кваліфікаційної роботи

(*nidnuc*, *data*)

Відмітка про попередній захист Сашньова М. В.

(ПИБ, підпис, дата)

12. Висновок про випускню кваліфікаційну роботу

Випускна кваліфікаційна робота студента Гудименко А. Л.

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми

Цензура М. О.

(прізвище, ініціали, підпис)

Завідувач кафедри

Криворучко О. В.

(підпис, прізвище, ініціали)

« » 20 p

АНОТАЦІЯ

Дипломна робота на тему «Розробка програмного забезпечення провайдерської компанії «Версус»» містить 64 сторінок, 11 рисунків та 1 додаток. Перелік посилань налічує 7 найменувань.

Мета дослідження: розробка функціоналу білінгової системи для першої лінії телефонної підтримки та відділу підключення нових клієнтів, враховуючи сучасні потреби ринку.

Об'єкт дослідження: компанія «Версус», котра надає телекомунікаційні послуги.

Предметом дослідження: аналіз та вивчення методів створення сучасних білінгових систем, їх структуризація.

У першому розділі проаналізовано існуючі у світі білінгові системи для компаній, що надають телекомунікаційні послуги, складено технічне завдання, де визначено мету, призначення та передумови створення білінгової системи, описано вимоги до розробки програмного продукту.

У другому розділі ми ознайомившись зі всіма аспектами розробки білінгових систем, описаних вище, ми почали моделювати архітектуру білінгової системи. Використовуючи різні підходи та моделюючи їх ми зупинились на варіанті використання CRM для забезпечення сумісності всіх компонентів, які потрібні для забезпечення функціонування білінгової системи. Було створено декілька діаграм для графічного зображення структури білінгової системи для полегшення розробки окремих модулів та зв'язків між ними.

В третьому розділі ми ознайомились з інструментом Wordpress та його принципом роботи щодо створення нових записів у базі даних. Було продемонстровану базу даних, її структуру, зв'язки між таблицями та поля які було створено в процесі роботи. Розроблено веб-інтерфейс білінгової системи та було забезпечено отримання необхідної інформації з бази даних. А також забезпечена можливість додавання нових записів до бази даних.

ANNOTATION

Thesis on "Software development of the provider company" Versus "" contains 64 pages, 11 figures and 1 appendices. The list of links includes 7 items.

The purpose of the study: to develop the functionality of the billing system for the first line of telephone support and the department of connecting new customers, taking into account the current needs of the market.

Object of research: Versus, a company that provides telecommunications services.

Subject of research: analysis and study of methods of creating modern billing systems, their structuring.

The first section analyzes the existing billing systems in the world for companies providing telecommunications services, a technical task, which defines the purpose, purpose and prerequisites for creating a billing system, describes the requirements for software development.

In the second section, we got acquainted with all aspects of the development of billing systems described above, we began to model the architecture of the billing system. Using different approaches and modeling them, we focused on the option of using CRM to ensure the compatibility of all components required to ensure the functioning of the billing system. Several charts have been created to graphically represent the structure of a billing system to facilitate the development of individual modules and the relationships between them.

In the third section, we got acquainted with the Wordpress tool and its principle of work on creating new records in the database. The database, its structure, relationships between tables and fields that were created in the process of work were demonstrated. The web interface of the billing system was developed and the necessary information from the database was obtained. It is also possible to add new records to the database.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО БІЛІНГОВІ СИСТЕМИ ТА ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
1.1. Загальні відомості про білінгові системи	6
1.2. Системне вивчення та аналіз сучасних білінгових систем.....	10
1.3. Системні вимоги до білінгових систем	12
1.4. Вимоги до дизайну білінгових систем.....	15
1.5. Технічне завдання на розробку програмного забезпечення.....	16
1.6. Висновки до розділу 1	17
РОЗДІЛ 2 АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ ТА ХАРАКТЕРИСТИКА БАЗОВОЇ ОРГАНІЗАЦІЇ КОМПАНІЇ «ВЕРСУС»	18
2.1. Моделювання аналізу та методології проектування	18
2.2. Експлуатаційна доцільність використання білінгової системи:	19
2.3. Модель взаємовідносин(модель ER) для оптимізації БД:	19
2.4. Методика розробки проєкту білінгової системи	21
2.5. Алгоритмізація розробки робочої частини проєкту.....	22
2.6. Опис алгоритму розробки робочої частини проєкту	23
2.7. Діаграми для моделювання білінгової системи.....	24
2.8. Система контролю і управління доступом	33

					КНТЕУ 121 07-08.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка програмного забезпечення провайдерської компанії "Версус"	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Вступ	2	64
Керівник	Сашньова М.В.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Гудименко А.Л.							
					Вступ			

2.9. Висновок до розділу 2	35
РОЗДІЛ 3 РОЗРОБКА ВЕБ ДОДАТКУ УПРАВЛІННЯ БІЛІНГОВОЮ СИСТЕМОЮ КОМПАНІЇ «ВЕРСУС»	36
3.1. Методика випробувань і результати експериментальної перевірки ...	36
3.2. Опис ключових етапів розробки інформаційної системи.....	39
3.3 Інтерфейс картки абонента	40
3.4. Принцип роботи	41
3.5. Плагін	44
3.6. Висновки до розділу 3	45
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТКИ.....	48
Додаток А.....	48

					КНТЕУ 121 07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		3

ВСТУП

Технології не стоять на місці, прогрес йде вперед з шаленою швидкістю, і ми вже не уявляємо собі нашого життя без автоматизації різноманітних процесів як в повсякденному житті, так і під час роботи.

Сучасні тенденції розвитку ринку телекомунікацій, пов'язані з постійно і швидко зростаючим різноманіттям видів послуг, що ставлять перед операторами зв'язку все більш складні завдання по організації процесу надання цих послуг своїм клієнтам. До таких завдань відносяться забезпечення гнучкої системи обліку і тарифікації послуг, що надаються, виставлення рахунків і обліку оплати (що становить основу білінгових систем або АСР - автоматизованих систем розрахунків); забезпечення врахування потреб клієнтів до різного виду послуг; підтримка різних способів оплати (передоплати в різних формах, оплати за виставленими рахунками тощо), та ін.

Актуальність теми напряду пов'язана з вже існуючою кількістю послуг, що можуть надавати провайдерські компанії, новими послугами, що будуть впроваджуватися, а також необхідним часом для вирішення питань, з якими можуть звертатися клієнти; полегшення взаємодії з великою кількістю інформації, що вигідно впливатиме на ефективність роботи, що в свою чергу буде забезпечувати конкурентоспроможність компанії на ринку.

					КНТЕУ 121 07-08.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка програмного забезпечення провайдерської компанії "Версус"	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Вступ	4	64
Керівник	Сашньова М.В.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Гудименко А.П.							
					Вступ			

Об'єктом дослідження є компанія «Версус», котра надає телекомунікаційні послуги.

Предметом дослідження є аналіз та вивчення методів створення сучасних білінгових систем, їх структуризація; аналіз існуючих систем керування базами даних (далі СУБД) з подальшим вибором найбільш підходящою для цілей, що переслідує створення білінгу.

В компанії регулярно зі значною та структурованою інформацією в базі даних працює саме перша лінія телефонної підтримки та відділ підключення нових клієнтів, тому метою дослідження є розробка функціоналу білінгової системи для першої лінії телефонної підтримки та відділу підключення нових клієнтів, враховуючи сучасні потреби ринку.

Для досягнення мети були поставлені такі завдання:

- Охарактеризувати існуючі білінгові системи;
- Розробити технічне завдання для білінгової системи «Версус»;
- Забезпечити незалежність роботи білінгової системи від ОС (Операційної системи);
- Створити зручний інтерфейс програмного забезпечення провайдерської компанії «Версус».

Практична значущість, як згадувалось вище, забезпечується великою кількістю інформації, що має бути структурованою та легкодоступною. А також відсутністю альтернативних підходів для вирішення цього питання.

					КНТЕУ 121 07-08.БР	Аркуш
						5
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 1

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО БІЛІНГОВІ СИСТЕМИ ТА ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Загальні відомості про білінгові системи

Білінгова система – це програмне забезпечення, розроблене спеціально для операторів (провайдерів). Слово білінг походить від англійського bill (рахунок), тобто білінгова система дозволяє тарифікувати надані послуги доступу [1].

Білінгова система враховує весь трафік, що проходить через комп'ютер. Завдяки спеціальному драйверу, враховується трафік, що проходить через мережеві інтерфейси, NAT, проксі-сервер і поштовий шлюз програми. Кожен мережевий пакет аналізується і підраховується з точністю до байта. Облік трафіку чітко розділяється на «зовнішній» і «внутрішній». Зовнішній - це трафік, отриманий від вищого провайдера, внутрішній – трафік, що надано в користування абонентам. Зовнішній трафік може бути детально розділений по підключеннях, провайдерам, серверів, мереж і протоколів - тобто можна завжди отримати детальну картину загального споживання трафіку або створити окремі лічильники трафіку з попередженнями і блокуваннями при досягненні певного ліміту.

Окремо можна задати знижки і вказати безкоштовний трафік. Перерахунок даних про баланс відбувається по кожному пакету, так що блокування по

					КНТЕУ 121 07-08.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка програмного забезпечення провайдерської компанії "Версус"	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Р1	6	64
Керівник	Сашньова М.В.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Гудименко А.Л.							
					Основні теоретичні відомості про білінгові системи			

перевищенню спрацьовує практично миттєво, виключаючи перевитрата трафіку. Стан особового рахунку, оплати, причини блокування, своє споживання трафіку і мережеву статистику абоненти можуть подивитися в своєму особистому кабінеті.

Білінгові системи давно не просто ведуть баланс клієнта і виставляють рахунки. Вони глибоко інтегровані в бізнес: забезпечують контроль інтернет трафіку, керують процедурою надання доступу до послуг, які надає компанія та якими користується клієнт. Надають клієнтові звіт про витрати в різних розрізах - за місяць, за рік, з моменту останнього платежу. Плюс білінгова система, як правило, повинна підтримувати безліч балансів користувача.

У невеликій кількості білінгових систем є, наприклад, окремий модуль продакт-каталог, який дозволяє легко комбінувати послуги в пакети і налаштовувати акції. Параметри якими він оперує можуть бути як послуги, так і окремі параметри такі як: хвилини розмови, пакети трафіку або послуги, пропонувані партнерами (так звані послуги третіх осіб), наприклад, квитки в кіно або поїздки на таксі. Є думка, що продакт-каталог - це взагалі елемент CRM-системи (система керування вмістом) [2], а не білінгу.

Також у випадку коли компанія складається з декількох філій, кожна з яких має власне програмне забезпечення, тобто свою білінгову систему, то не буде забезпечена можливість, наприклад, запускати масштабні акції відразу по всій країні, оскільки не буде єдиного інструменту для управління різним програмним забезпеченням. Страждає звітність та аналітика: зібрати дані по підключеннях і користування послугами з усіх цих розрізнених систем просто неможливо. Часткове вирішення проблеми - зонтична CRM, в якій є продакт-каталог. Деякі компанії так і роблять - торгують з CRM, а тарифікують в білінгу. Для цього в компанії повинен бути дійсно просунутий IT-відділ.

					КНТЕУ 121 07-08.БР	Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата		

Інтегрувати написані на різних мовах і за різними принципами білінгові системи з єдиною CRM - непросте завдання.

Підведемо підсумок. Важливим завданням для телекомунікаційної компанії є організація сервісного клієнтського обслуговування. У більшості телекомпаній єдиною точкою входу для клієнтів є call-центр. Використання CRM-системи дозволяє значно спростити роботу оператора сервісного центру за рахунок можливості:

- Управляти повним набором даних про клієнтів (картка клієнта);
- Управляти послугами клієнта (обробляти заявки, переглядати довідники тарифних планів, особових рахунків і послуг, використовувати калькулятор тарифних планів, тощо);
- Управляти зверненнями клієнта (фіксувати звернення клієнтів, класифікувати їх за темами, типу, статусам, переглядати історію взаємодії за зверненнями, управляти маршрутами обробки звернень, тощо);
- Управляти базою знань (керувати інформаційними статтями, які зберігають інформацію з відповідями по різних питань клієнтів і користувачів, структурувати відомості про рішення запитів за темами).

Крім того, CRM-система разом із системою телефонії (call-центр) дозволяє автоматизувати дії оператора в частині реєстрації вхідних і вихідних дзвінків, проведення телефонного анкетування клієнтів. Також CRM надає засоби моніторингу та аналізу обслуговування клієнтів та аналізу ефективності роботи самих операторів call-центру.

Підсумовуючи вищесказане, перерахуємо, які переваги дає використання CRM-системи телекомунікаційної компанії:

- Спрощення процесу продажів:
 - Зменшення часу доступу до інформації про клієнта;
 - Розробка механізмів крос-продажів і допродаж супутніх послуг;

					КНТЕУ 121 07-08.БР	Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

- Поява шаблонів (сценаріїв) продажів новому абоненту і обслуговування існуючих;
 - Зберігання історії про взаємодії з клієнтом до продажів.
- Скорочення часу обслуговування абонентів:
 - Формування бази знань - швидкий пошук відповідей на типові запитання;
 - Надання інформації про клієнта в «єдиному» вікні;
 - Контроль обробки заявок на обслуговування клієнтів.
 - Скорочення часу і ресурсів для побудови аналітичних звітів
 - Розробка системи звітності для аналізу поведінки клієнтів.
 - Отримання розширених засобів і аналітичних вимірювань для аналізу клієнтів:
 - Єдина, актуальна і достовірна база клієнтів;
 - Формування історії взаємин з клієнтами;
 - Збір і аналіз додаткових аналітичних ознак для аналізу клієнтів.
 - Отримання інструменту для сегментації клієнтської бази за різними критеріями:
 - Формування цільових груп і підгруп клієнтів;
 - Автоматизація процесу формування «профілю клієнта»;
 - Централізоване зберігання інформації для подальшого розвитку алгоритмів аналізу сегментів і обробки даних;
 - Використання цільових груп клієнтів для проведення маркетингових та інших «адресних» активностей.

Ефективність від використання CRM-системи досягається багато в чому завдяки її розширеній інтеграції з іншими ІТ-системами телекомунікаційної

					КНТЕУ 121 07-08.БР	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

компанії. Як приклад: якщо не реалізована інтеграція CRM з білінгом, оперативний доступ оператора call-центру на момент дзвінка до інформації про стан рахунку клієнта утруднений. Відповідно, пошук необхідних даних призводить до зниження швидкості обслуговування клієнтських дзвінків.

1.2. Системне вивчення та аналіз сучасних білінгових систем.

Системний аналіз - це процес збору та інтерпретації фактів, діагностування проблем та інформації, щоб рекомендувати вдосконалення системи. Це діяльність з вирішення проблем, яка вимагає інтенсивного спілкування між користувачами системи та розробниками системи. Завдання дослідження та аналізу системи полягає у проведенні досліджень та аналізів операційного та технологічного характеру, а також сприянні обміну та розробці методів та інструментів для оперативного аналізу в застосуванні до оборонних проблем.

Системний аналіз або дослідження є важливою фазою будь-якого процесу розробки системи. Існуюча система ретельно вивчається та визначаються проблемні області.

Попереднє дослідження - це процес збору та інтерпретації фактів, використовуючи інформацію для подальших досліджень системи. Попереднє дослідження - це діяльність з вирішення проблем, яка вимагає інтенсивного спілкування між користувачами системи та розробниками системи. Вона проводить різні техніко-економічні обґрунтування.

У цих дослідженнях можна отримати приблизну цифру системної діяльності, з якої можна приймати рішення щодо стратегій, яких слід дотримуватися для ефективного системного вивчення та аналізу.

					КНТЕУ 121 07-08.БР	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

Всі існуючі на даний момент часу на ринку комунальних послуг білінгові системи умовно можна розділити на два типи:

- «Готові» системи, написані вузькоспеціалізованими компаніями-розробниками під конкретні види завдань, що відповідають конкретним законодавчим вимогам певних держав.
- Так звані «важкі», «універсальні» для всіх видів бізнесу системи комплексної обробки даних ERP-системи. [3]

Природно, що під поняттям «готові системи» мається на увазі саме те, що ці системи практично відразу ж після покупки готові до експлуатації, тобто їх не треба переписувати, щоб усунути будь-які недоліки Проектування або принципові помилки ведення бухгалтерського обліку.

Якщо порівнювати «готові», некоректно працюють системи, яких більш ніж достатньо на ринку України з «важкими» системами, то останні Вам дадуть певні гарантії відсутності грубих помилок Проектування. Однак, 100% гарантії відсутності принципових помилок все одно ніхто не дасть. При покупці готової системи Ви бачите що купуєте, а не те, «як це планується зробити в майбутньому». Тобто, необхідно розуміти, що купуючи «важку» систему, Ви отримуєте не готове рішення, а інструментальне засіб для його створення, універсальну заготовку. Зрозуміло, чим більше універсальність заготовки, тим більше роботи по її настройці. Якщо проводити аналогію з будівництвом будинку, то замість готового житла Ви отримуєте ескіз будинку, фундамент, цегла та кельму. Жити в такому будинку, природно, неможливо. Пройде кілька років, перш ніж Ви налаштуєте свою «важку» систему. Природно, це буде нелегко з різних причин. Наприклад, як показує практичний досвід, в двох

					КНТЕУ 121 07-08.БР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

випадках з трьох, відстрочка в кілька років до отримання готового продукту (якщо він взагалі коли-небудь буде) зазвичай погано закінчується для кар'єри чиновників, які беруть участь у процес прийняття рішення щодо закупівлі «Важкою» системи, так як «гроші витрачені, а результату немає». Уразливий стан для функціонера зберігається протягом усього періоду настройки і впровадження «важкої» системи.

1.3. Системні вимоги до білінгових систем

Існує безліч вимог до білінгових систем. Однією з ключових вимог є простота використання. Вона має бути розроблена таким чином, що навіть людина з базовими знаннями про комп'ютери зможе експлуатувати її з відносною легкістю. Також система має бути:

- Самостійною;
- Незалежною;
- Може впоратися з великою базою клієнтів
- Незалежною від бази даних;
- Незалежною від операційної системи;
- Її можна розширювати в майбутньому.

Для різних користувачів для білінгової системи необхідно мати систему входу. Паролі повинні бути зашифровані при передачі від клієнта до сервера і навпаки. Білінгова система повинна бути доступною за допомогою браузера. Доступ до інформації про клієнта повинен бути легким. Одночасне використання цієї системи декількома суб'єктами не повинно пошкодити дані і вся інформація повинна бути точно збережена, оскільки особливість системи така, що деякі параметри як звіти, зміни працівника, тощо, потрібно редагувати. Білінгова система повинна працювати з різних місць, та підключатися до однієї

					КНТЕУ 121 07-08.БР	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

бази даних. Система повинна бути оцінена спочатку з технічної точки зору. Оцінка цієї доцільності повинна базуватися на системних вимогах з точки зору введення, результату, програм та процедур.

Для розробки проекту необхідна мова програмування вищого рівня, наприклад, РНР. Для цілей бази даних потрібна сумісна з програмуванням база

даних, така як MySQL, а для звітності ми використовуємо Data report. Для використання цієї системи нам потрібен комп'ютер з простою конфігурацією:

1. Процесор
 - Процесор Intel Pentium N5000
 - Продуктивність процесору 2596
 - Кількість ядер 4 ядра
 - Тактова частота 1.1–2.7 ГГц
 - Кількість потоків - 4 потоки
 - Об'єм кеша L3 4 Мб
2. Материнська плата
 - Максимальна кількість слотів пам'яті 1 слот
 - Максимальний обсяг пам'яті 8 Гб
3. Відеокарта
 - Продуктивність відеокарти 332
 - Відеокарта Intel UHD Graphics 600
 - Тип Інтегрована
4. Внутрішній накопичувач ??????
5. Оперативна пам'ять:
 - Обсяг 4 Гб
6. Додатково:

					КНТЕУ 121 07-08.БР	Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата		

- Слот для 2.5 HDD/SSD
- Слот для M2 SSD
- Блок живлення 40 Вт

Проект слід розробляти таким чином, щоб необхідні функції та ефективність були досягнуті. Проект розроблений в рамках новітніх технологій. Система була розроблена з використанням PHP.

Вимоги до операційної системи. Операційна система є важливим чинником при розробці будь-якого програмного забезпечення. Адже саме від операційної системи залежить який інструментарій будуть мати можливість використовувати розробники. А також від яких вразливостей операційної системи слід буде передбачити засоби захисту.

В реаліях сьогодення білінгові системи, що розроблені виключно під конкретні операційні системи, значною мірою програють кросплатформним, від них зазвичай відмовляються. Оскільки компанії-замовники зазвичай вже мають комп'ютерні станції з вже встановленою операційною системою та базою даних, то більш універсальним рішенням буде створення з боку розробників ПЗ продукту що буде працювати в умовах різних операційних систем.

Економічна доступність білінгових систем. Система, що розвивається, повинна бути обґрунтована витратами та вигодами. Отже, якщо ми розробимо систему, інвестуючи наш ресурс, включаючи час, гроші та інтелектуальність, тоді зусилля компанії будуть зменшені. Таким чином, ми можемо назвати нематеріальну вигоду. Це очікуваний результат цієї системи і наскільки це можливо, економічно оптимізований.

Критерії забезпечення того, що зусилля зосереджуються на проєкті, який дасть найкращі результати, повертаються якнайшвидше. Одним із факторів, що впливає на розробку нової системи, є витрати, які вона потребуватиме.

					КНТЕУ 121 07-08.БР	Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

Оскільки система розроблена в рамках проектної роботи, витрати на запропоновану систему не передбачаються вручну. Також усі ресурси вже доступні, це вказує на те, що система економічно можлива для розвитку.

1.4. Вимоги до дизайну білінгових систем

Дизайн - це перший крок на етапі розробки будь-якого розробленого продукту або системи. Дизайн - це творчий процес. Хороший дизайн - запорука ефективної системи. Термін "Проектування" визначається як "процес застосування різних методів і принципів з метою визначення процесу або системи з достатньою деталізацією, щоб дозволити її фізичну реалізацію". Це може бути визначено як процес застосування різних технік і принципів з метою визначення пристрою, процесу або системи з достатньою деталізацією, щоб дозволити його фізичну реалізацію. Розробка програмного забезпечення відповідає технічному ядру процесу інженерії програмного забезпечення та застосовується незалежно від використовуваної парадигми розробки. Дизайн системи розробляє архітектурні деталі, необхідні для побудови системи або продукту. Як і у випадку з будь-яким систематичним підходом, це програмне забезпечення також пройшло максимально можливу фазу проектування, точно налаштувавши всі рівні ефективності, продуктивності та точності. проектування системи проходить дві фази розробки: логічне та фізичне проектування.

Логічний дизайн. Логічна структура інтерфейсу та визначення меж системи. Він включає наступні кроки:

- Переглядає поточну фізичну систему - це потоки даних, вміст файлів, частоти обсягу тощо.
- Готує технічні характеристики - тобто визначає формат, зміст та частоту звітів.

					КНТЕУ 121 07-08.БР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

- Готує вихідні специфікації, формат, вміст та більшість функцій введення.
- Готує специфікації редагування, захисту та контролю.
- Вказує план реалізації.
- Готує логічний Проектний похід через інформаційний потік, вихід, контроль введення та план впровадження.
- Переглядає переваги, витрати, цільові дати та системні обмеження.

1.5. Технічне завдання на розробку програмного забезпечення

- ❖ Найменування системи
 - Білінгова система для операторів першої лінії
- ❖ БС1
 - Початок робіт 14.12.2020
 - Завершення робіт 24.04.2021
- ❖ Призначення і цілі створення системи
 - Розробка функціоналу білінгової системи
- ❖ Характеристики об'єкта автоматизації
 - Самостійне з'єднання з базою даних
 - Самостійне внесення змін до бази даних
 - Автоматичне отримання оновленої інформації
- ❖ Вимоги до системи
 - Багатоплатформність (забезпечується тим що це веб додаток)
 - Простота інтерфейсу
 - Забезпечення необхідним функціоналом
 - Отримання інформаційної картки клієнта

					КНТЕУ 121 07-08.БР	Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата		

- Внесення змін до інформації з попереднього пункту
 - ❖ Вимоги до надійності
- Обмеження прав користувачів білінгу
- Шифрування інформації(SSH шифрування)
 - ❖ Вимоги до видів забезпечення
- Браузер(Chrome або Mozilla firefox)
 - ❖ Склад і зміст робіт
- Отримання вимог від замовника
- Планування необхідних функцій
- Створення дизайну
- Реалізація Проєкту
- ❖ Порядок контролю і приймання системи
 - Розміщення програмного додатку на сервері
 - Маніпуляція з базою даних через білінгову систему

1.6. Висновки до розділу 1

В даному розділі ми ознайомилися з поняттям білінгової системи, вимогами та основними підходами до розробки таких систем. Завдяки системному аналізу було охарактеризовано системні вимоги для забезпечення стабільної роботи білінгової системи. Відповідно до поставленого завдання створено технічне завдання, умови якого мають бути виконані в повному обсягу. Ознайомились з таким поняттям як дизайн, а також його підвидом – логічний дизайн. Ознайомились з сучасною технологією: CRM системою та яким чином вона може позитивно вплинути на роботу білінгової системи.

					КНТЕУ 121 07-08.БР	Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ ТА ХАРАКТЕРИСТИКА БАЗОВОЇ ОРГАНІЗАЦІЇ КОМПАНІЇ «ВЕРСУС»

2.1. Моделювання аналізу та методології проектування

На технічному рівні програмна інженерія починається з ряду завдань моделювання, які ведуть до повної специфікації вимог. Моделювання аналізу використовує поєднання тексту та діаграмних форм, щоб зобразити вимоги до даних, функцій та поведінки таким чином, що є відносно простим для розуміння та перевірки на правильність, повноту та послідовність. Модель аналізу є першим технічним поданням системи.

Існує кілька методів моделювання аналізу, але лише дві з цих моделей широко використовуються. Вони являють собою структурований аналіз та об'єктно-орієнтований аналіз (ООА). У цьому проекті використана модель структурованого аналізу. Оскільки ООА використовується, коли існує багато потоків трансформації, існує велика ймовірність пошкодження даних. Оскільки в цій системі мало потоків трансформації та багато потоків транзакцій, було обрано структурований аналіз.

Методологія проектування відноситься до розробки системи або методу для унікальної ситуації. Сьогодні цей термін найчастіше застосовується до технологічних галузей стосовно веб-дизайну, програмного забезпечення або проектування інформаційних систем.

					КНТЕУ 121 07-08.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка програмного забезпечення провайдерської компанії "Версус"	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					P2	18	64
Керівник	Сашньова М.В.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Гудименко А.Л.				Аналіз предмета дослідження та характеристика базової організації компанії «Версус»			

Ключовим для методології проектування є пошук найкращого рішення для кожної ситуації проектування, будь то промисловий дизайн, архітектура чи технологія. Методологія проектування наголошує на використанні методів мозкового штурму для заохочення інноваційних ідей та спільного мислення для реалізації кожної запропонованої ідеї та пошуку найкращого рішення. Задоволення потреб і потреб кінцевого споживача є найважливішим питанням.

У методології проектування також використовуються основні методи дослідження, такі як аналіз та тестування.

2.2. Експлуатаційна доцільність використання білінгвової системи:

Оскільки ми використовуємо мову програмування високого рівня, існує безліч можливостей зробити нашу систему зручною для користувача за допомогою текстового поля, кнопок тощо.

Запропонована система буде бажаною в рамках існуючого способу управління. Оператору стане дуже просто керувати системою. Ця система підвищить продуктивність всієї компанії.

2.3. Модель взаємовідносин(модель ER) для оптимізації БД:

У програмній інженерії модель сутність-взаємозв'язок (модель ER) - це модель даних для опису даних або інформаційних аспектів ділового домену або вимог до його процесу абстрактним способом, що дозволяє в кінцевому підсумку реалізуватися в базі даних, наприклад реляційна база даних. Основними компонентами моделей ER є сутності (речі) та взаємозв'язки, які можуть існувати між ними.

При моделюванні ER структура бази даних зображається у вигляді діаграми, яка називається діаграмою взаємозв'язку сутності (або діаграмою ER), яка нагадує графічний розподіл речення на його граматичні частини. Суб'єкти відображаються як точки, багатокутники, кола або овали.

					КНТЕУ 121 07-08.БР	Аркуш
						19
Зм.	Аркуш	№ докум	Підпис	Дата		

Відносини зображуються як лінії, що з'єднують точки, багатокутники, кола чи овали. Будь-яка діаграма ER має еквівалентну реляційну таблицю, а будь-яка реляційна таблиця має еквівалентну діаграму ER. Діаграмування ER - неоціненна допомога інженерам у Проектуванні, оптимізації та налагодженні програм баз даних.

У логічному сенсі сутності є еквівалентом граматичних іменників, таких як працівники, відділи, товари чи мережі. Сутність може бути визначена за допомогою її властивостей, які називаються атрибутами. Відносини - це еквівалент дієслів або асоціацій, таких як акт покупки, акт ремонту, членство в групі або керівництво відділом. Відносини можна визначити відповідно до кількості сутностей, пов'язаних з нею, відомих як ступінь.

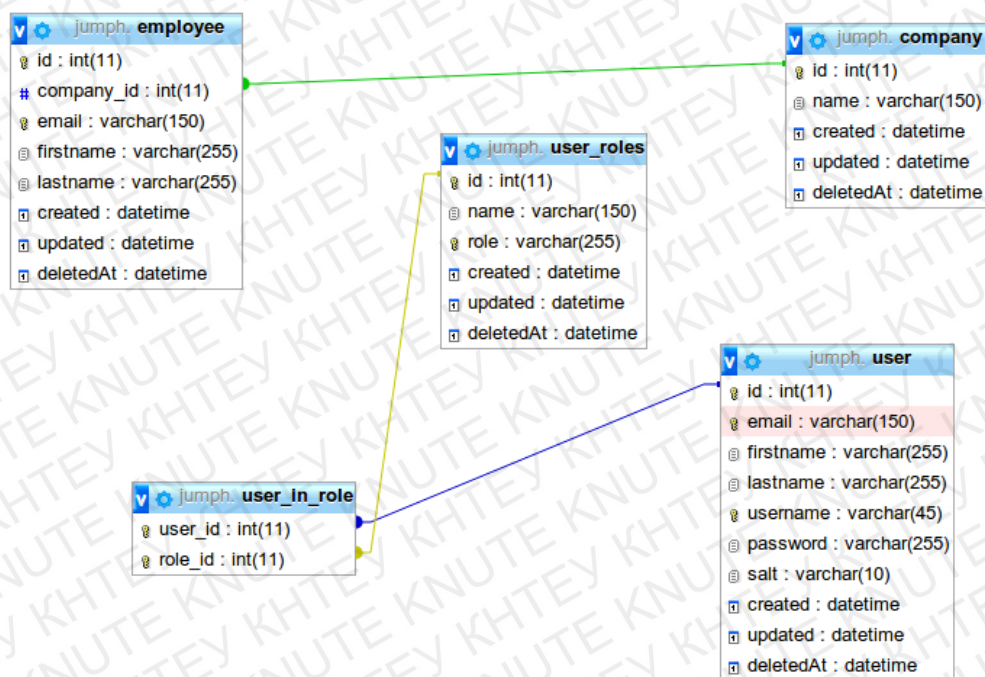


Рис. 2.1 Зображення ER моделі прав доступу користувачів

					КНТЕУ 121 07-08.БР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

2.4. Методика розробки проєкту білінгової системи

1. Збір інформації

Під збором інформації мається на увазі отримання необхідної інформації від замовника, а також з додаткових джерел.

2. Наступним етапом розробки була селекція і компоновка всієї зібраної інформації. Всі отримані дані перетворювалися за допомогою логічної вибірки, в набір впорядкованої за категоріями інформації. Вся інформація була чітко відсортована по категоріях, після чого була перетворена в чітку, ієрархічну структуру за категоріями.

3. Далі були створені напрацювання готових програмних шаблонів, що містять інтерактивні компоненти.

Однією з основних і найбільш значущих завдань при розробці білінгу є створення оптимального дизайну. Для досягнення потрібного ефекту, було розроблено кілька графічних дизайн - шаблонів зовнішнього інтерфейсу. Далі, після узгодження з замовником, був відібраний найбільш оптимальний варіант.

4. Не повинно бути зайвої інформації, щоб інтерфейс не виглядав перевантаженим, і завантаження картки абонента відбувалася в максимально короткий час. У зв'язку з цим була обрана найбільш проста, але максимально функціональний варіант зовнішнього інтерфейсу.

5. Система розрахована на будь-якого користувача. Навіть недосвідчені користувачі можуть з успіхом орієнтуватися та використовувати цей інструмент.

6. Наступним етапом розробки було створення пробних експериментальних програмних модулів програмної основи.

Особливістю даного етапу розробки є активне використання досвіду інших розробників. Для цього була використана інформація, а також готові демонстраційні програмні компоненти, отримані з мережі Інтернет, шляхом

					КНТЕУ 121 07-08.БР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

пошуку потрібної інформації на сайтах розробників любителів і розробників професіоналів.

7. Далі була також проведена селекція отриманих даних і відбір найбільш підходящих з них, після чого, були уважно вивчені методи і принципи, що лежать в основі програмних компонентів створеними незалежними розробниками. Після детального вивчення принципів роботи і застосовуваних методик програмування, були створені програмні шаблони, і процедури для складання робочої частини програмного продукту.

8. Далі йде етап розробки, в ході якого проводиться тестування і налагодження окремих програмних компонентів і блоків, призначених для збирання готового проекту.

9. Завершальним етапом розробки є складання і тестування готового проекту. Методикою роботи на даному етапі є багато платформене тестування готового програмного продукту. Готовий проект тестувався на працездатність на локальному комп'ютері розробника, а також на різних операційних системах з встановленим набором програмних компонентів.

2.5. Алгоритмізація розробки робочої частини проекту

Основним завданням при створенні і розробки робочої частини проекту є створення повноцінного функціонуючого програмного коду із зовнішньою графічною надбудовою (інтерфейсом користувача). Дане завдання поділяється на кілька етапів:

1. Створення функціонального HTML коду для відображення зовнішніх форм і елементів управління.
2. Створення активних елементів дизайну, які базуються на об'єктно-орієнтованих мовах HTML.

Простіше кажучи, алгоритмізація завдання зводиться до вирішення двох основних завдань:

					КНТЕУ 121 07-08.БР	Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата		

- Розробка функціонального зовнішнього інтерфейсу користувача, на мові HTML

- Розробка внутрішнього, серверного компонента, що базується на технології ASP, для функціонування інтерактивних елементів програмного продукту, що спеціалізуються на роботі з базою даних.

2.6. Опис алгоритму розробки робочої частини проекту

У розгорнутому, докладному вигляді алгоритм задачі являє собою наступне:

1. Створення шаблону зовнішнього інтерфейсу з застосуванням технології HTML, розробка дизайну візуальних компонентів: зовнішній вигляд Інтернет - сторінки, активна анімація, елементи управління.

2. Створення процедур захисту від несанкціонованого доступу і несанкціонованої реєстрації.

3. Створення програмного ASP модуля Інтернет форуму.

4. Створення обробника всіх можливих помилок.

5.Тестування і налагодження.

Переваги запропонованої нашої системи:

- Обробка всієї інформації про клієнтів і правильне створення рахунків-фактури.
- Безпека даних.
- Зменшити введення даних вручну.
- Легко отримати картку клієнтів.
- Доступні функціональні можливості для виконання різних завдань.
- Легко надавати клієнтам інформацію про рахунки.
- Мінімальне споживання часу.

					КНТЕУ 121 07-08.БР	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

Запропонована система зручна для користувача, оскільки пошук та зберігання даних відбувається швидко і дані підтримуються ефективно.

2.7. Діаграми для моделювання білінгової системи

UML - це інструментарій для моделювання, який ви можете використовувати для побудови своїх діаграм. Варіанти використання представлені позначеною овальною формою. Фігури на паличці представляють акторів у процесі, а участь актора в системі моделюється з допомогою межі між актором та варіантом використання. Щоб зобразити межу системи, намалюйте поле навколо самого випадку використання.

Для відображення станів при роботі оператора з білінговою системою ми використовували діаграму станів наведену на рис. 2.2. Також часто діаграма станів використовується аналітиками для опису послідовності переходів об'єкта з одного стану в інший. Діаграма станів покаже нам всі можливі стани, в яких може перебувати об'єкт, а також процес зміни станів в результаті зовнішнього впливу.

Основними елементами діаграми станів є «Стан» і «Перехід». Діаграма станів має схожу семантику з діаграмою діяльності, тільки діяльність тут замінена станом, переходи символізують дії. Таким чином, якщо для діаграми діяльності відмінність між поняттями «Діяльність» і «Дія» полягає в можливості подальшої декомпозиції, то на діаграмі станів діяльність символізує стан, в якому об'єкт перебуває тривалий кількість часу, в той час як дія моментально.

					КНТЕУ 121 07-08.БР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

Стан може містити тільки ім'я або ім'я та додатково список внутрішніх дій. Список внутрішніх дій містить перелік дій, які виконуються під час знаходження об'єкта в даному стані. Даний список фіксований. Список основних дій включає наступні значення:

entry - дія, яка виконується в момент входу в даний стан (вхідний дію);

exit - дія, яка виконується в момент виходу з цього стану (вихідна дія);

do - виконується діяльність ("do activity") протягом усього часу, поки об'єкт знаходиться в даному стані

defer - подія, обробка якого пропонується в іншому стані, але після того, як всі операції в поточному будуть завершені.[\[4\]](#)

					КНТЕУ 121 07-08.БР	Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		

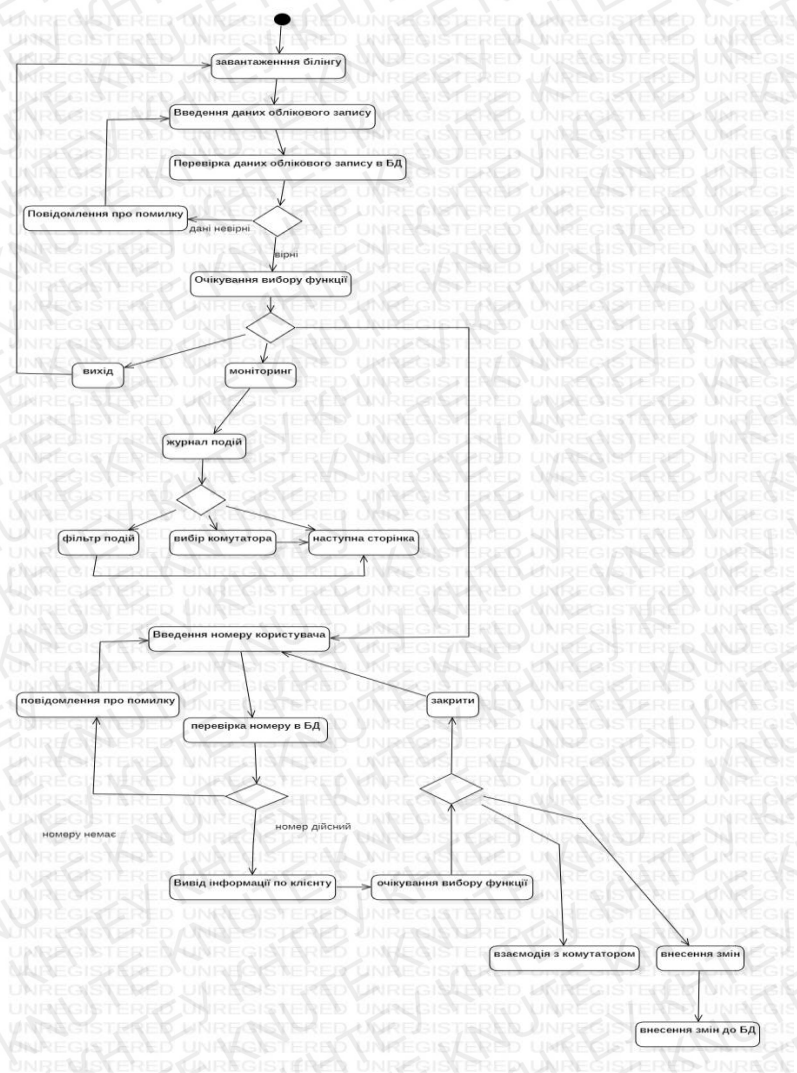


Рис. 2.2 Діаграма станів при роботі оператора з білінговою системою

					Аркуш
					26
Зм.	Аркуш	№ докум	Підпис	Дата	

КНТЕУ 121 07-08.БР

Правильна діаграма Usecase, зображена на рис. 2.3. відображає огляд високого рівня взаємозв'язку між випадками використання, акторами та системами. Експерти рекомендують використовувати схеми випадків використання, щоб доповнити більш описовий варіант використання тексту.

Діаграми випадків використання UML ідеально підходять для:

- Представлення цілей взаємодії системи та користувача
- Визначення та організація функціональних вимог у системі
- Визначення контексту та вимог системи
- Моделювання основного потоку подій у випадку використання

Схема взаємодії з білінговою системою

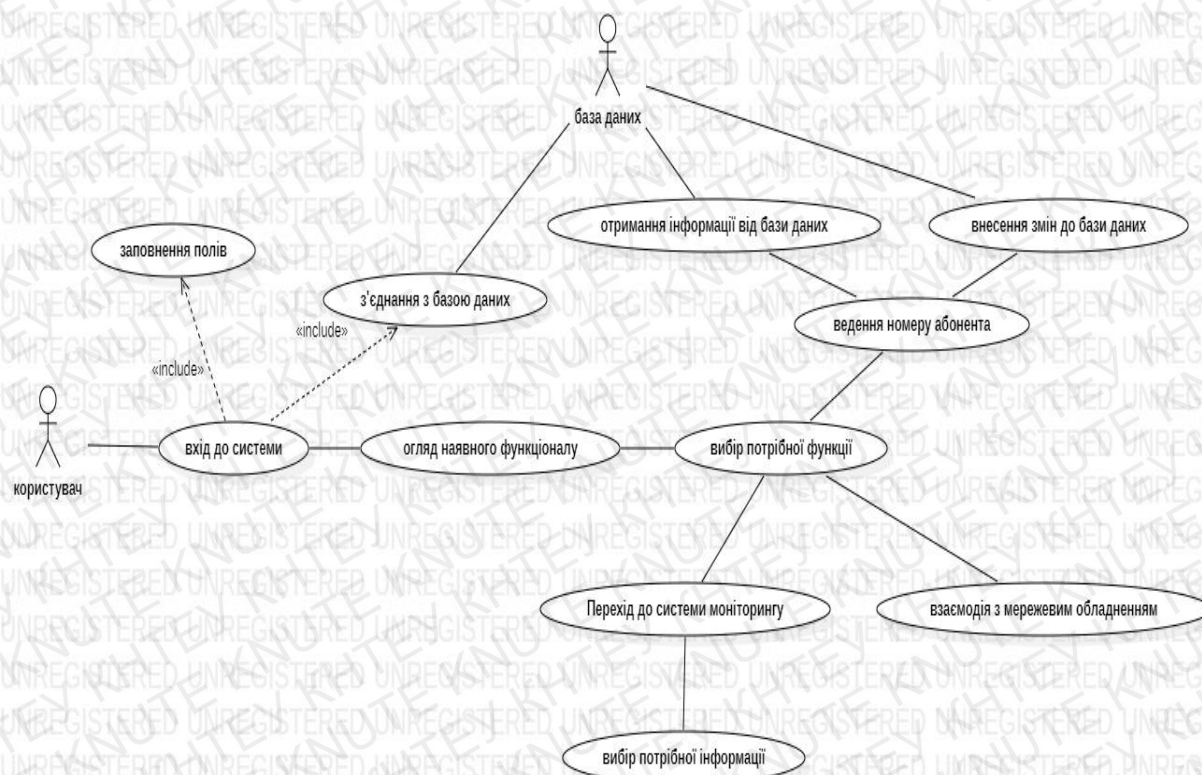


Рис. 2.3 Usecase діаграма для відображення взаємодії з білінговою системою

					КНТЕУ 121 07-08.БР	Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата		

Діаграма діяльності

Діаграма діяльності, зображена на Рис. 2.4, визначається як діаграма UML, яка фокусується на виконанні і потоці поведінки системи замість реалізації. Це також називається об'єктно-орієнтованої блок-схемою. Діаграми дій складаються з дій, які складаються з дій, які застосовуються до технології поведінкового моделювання.

Діаграма діяльності візуально представляє ряд дій, управління в системі, подібний до блок-схеми або схеми потоку даних. Діаграми діяльності часто використовуються при моделюванні бізнес-процесів. Вони також можуть описати кроки на схемі використання. Модельовані дії можуть бути послідовними та одночасними. В обох випадках діаграма діяльності матиме початок (початковий стан) і кінець (кінцевий стан).

Між ними є способи зобразити діяльність, потоки, рішення, охорону, злиття та часові події тощо. Дізнайтеся про символи діаграми діяльності нижче:

Початковий стан

Маленьке заповнене коло, за яким слідує стрілка, представляє початковий стан дії або початкову точку для будь-якої діаграми діяльності. Для діаграми діяльності з використанням плавців переконайтеся, що початкова точка розміщена у верхньому лівому куті першої колонки.

Стан дії представляє безперерйну дію об'єктів.

					КНТЕУ 121 07-08.БР	Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата		

Об'єктний потік

Потік об'єктів відноситься до створення та модифікації об'єктів за видами діяльності. Стрілка потоку об'єкта від дії до об'єкта означає, що дія створює об'єкт або впливає на нього. Стрілка потоку об'єктів від об'єкта до дії вказує на те, що стан дії використовує об'єкт.

Переваги діаграм діяльності

Діаграми діяльності представляють ряд переваг для користувачів. Подумайте про створення діаграми діяльності, щоб:

- Продемонструйте логіку алгоритму.
- Опишіть кроки, виконані у випадку використання UML.
- Ілюструйте бізнес-процес або робочий процес між користувачами та системою.
- Спростіть та вдосконалить будь-який процес, пояснивши складні аспекти використання.
- Моделюйте елементи програмної архітектури, такі як метод, функція та операція.[\[5\]](#)

					КНТЕУ 121 07-08.БР	Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата		

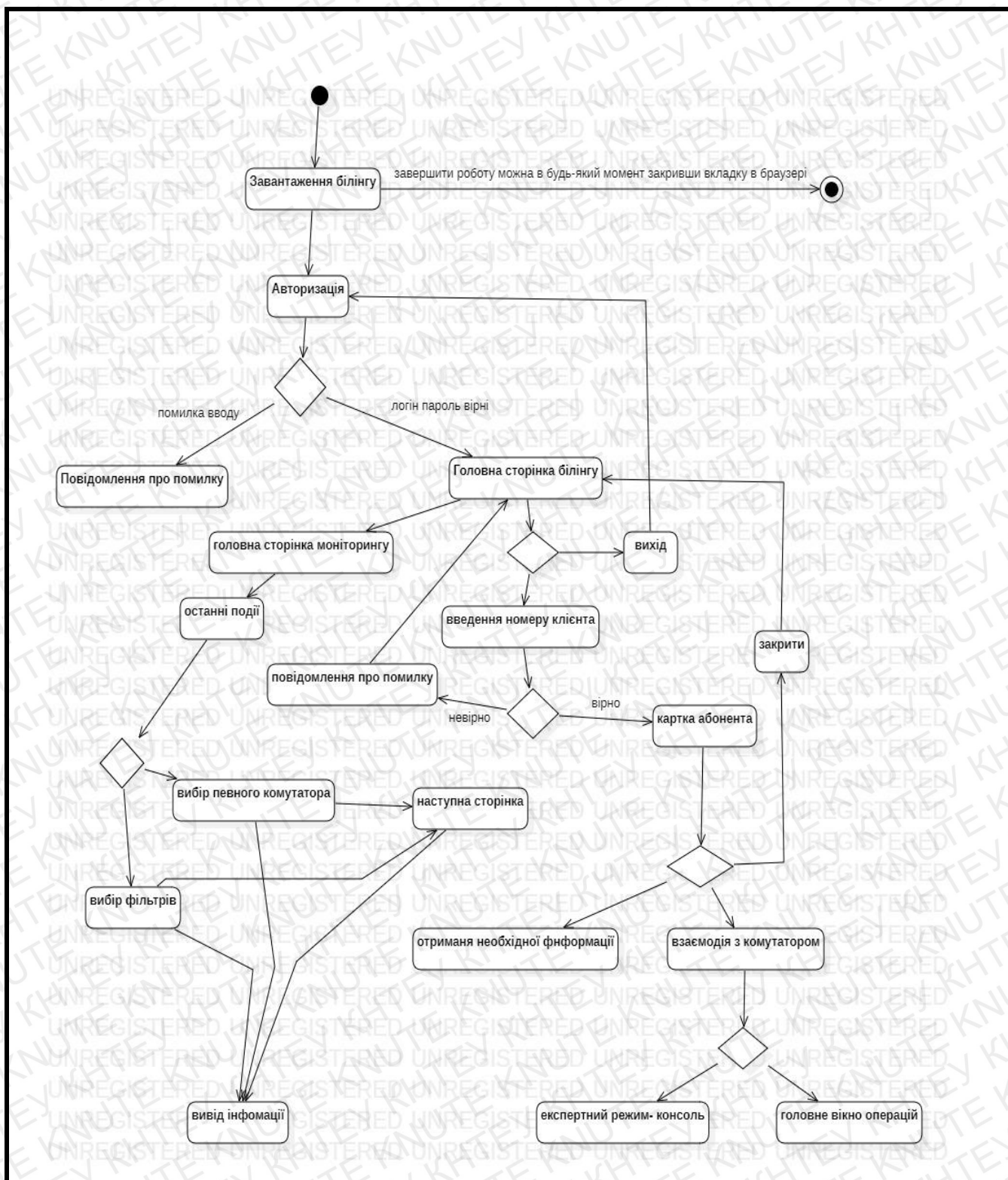


Рис. 2.4 Діаграми діяльності використовуються при моделюванні процесів під час роботи з білінговою системою

					КНТЕУ 121 07-08.БР	Аркуш
						30
Зм.	Аркуш	№ докум	Підпис	Дата		

Діаграма класів

Діаграма класів, наведена на Рис. 2.5, демонструє класи системи, їх атрибути, методи і взаємозв'язку між ними. Діаграма класів розроблюваної системи представлена рисунку нижче. На представленій діаграмі знаходиться п'ять класів.

Діаграма класів моделює статичну структуру системи. Вона показує зв'язки між класами, об'єктами, атрибутами та операціями.

Класи представляють абстракцію сутностей із загальними характеристиками. Асоціації представляють відносини між класами.

Зображуйте класи прямокутниками, розділеними на відсіки. Помістіть ім'я класу в перший розділ (відцентрований, напівжирний та з великої літери), перелічіть атрибути у другому розділі (вирівнювання за лівим краєм, не виділено жирним шрифтом та малими літерами) та запишіть операції у третій.[\[6\]](#)

					КНТЕУ 121 07-08.БР	Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		

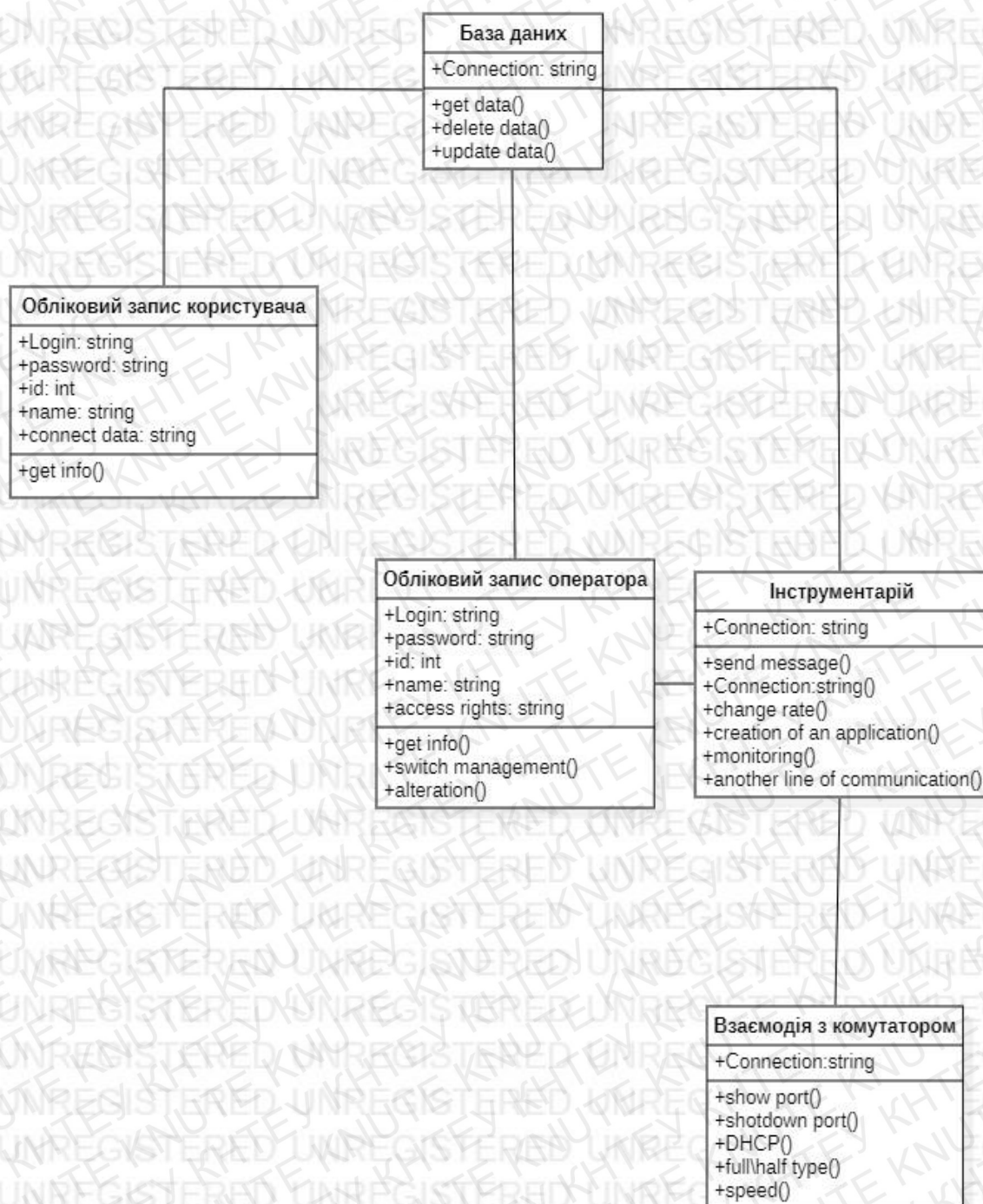


Рис. 2.5 Діаграма класів демонструє класи системи, їх атрибути, методи і взаємозв'язку між ними в білінгві системі

					КНТЕУ 121 07-08.БР	Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата		

2.8. Система контролю і управління доступом

Програмна СКУД. Технології ХХІ століття дозволили створити новий вид СКУД, якісно відрізнявся від своїх попередників. Програмна СКУД - це сукупність програмно-технічних засобів по автоматизації контролю переміщення співробітників по території підприємства, обліку використання робочого часу, що дозволяють аналізувати і оптимізувати роботу підприємства за допомогою складання звітів і графіків в режимі реального часу (on-line режим). Модель програмної СКУД виглядає наступним чином.

Відповідно, при реалізації програмної СКУД такий проміжний етап як контролер виключається, і все обладнання підключається безпосередньо до комп'ютера, що дозволяє миттєво отримувати і обробляти інформацію, складаючи різного роду звіти в режимі реального часу.

Крім того, в програмній СКУД користувач може самостійно налаштовувати систему зі свого робочого місця. У разі пред'явлення специфічних вимог до роботи системи (наприклад, підтвердження проходу, використання нестандартного обладнання, зокрема, ваг з метою їх розміщення на автомобільній прохідній, фотографування персон / автомобілів в момент проходу / проїзду і ін.) Програмна СКУД надає кошти для їх реалізації.

Отже, головною відмінністю вищезгаданих видів СКУД є наступне: в контролерних СКУД інформація про призначені картах і проходах спочатку закладається в контролер, і тільки потім переноситься на комп'ютер. Як правило, це відбувається вже *post factum*. А це, в свою чергу, означає, що управління на турнікет йде саме з контролера: саме він приймає рішення, пропустити чи ні даної людини в дану двері. У програмних же СКУД безпосередній зв'язок обладнання з

					КНТЕУ 121 07-08.БР	Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата		

комп'ютером дає можливість отримувати і обробляти інформацію відразу ж в момент її надходження.

У разі реалізації контролерної СКУД всі зареєстровані події, всі коди ідентифікаторів зі списком прав кожного з них зберігаються в пам'яті контролера. При цьому необхідно враховувати, що пам'ять контролера, незважаючи на сучасні технології, обмежена, час читання подій з пам'яті контролера при відновленні комплексного режиму може бути досить тривалим. При установці програмної СКУД вся інформація зберігається в пам'яті, відповідно, комп'ютера, і процедура внесення змін до системи набагато легше здійснення.

Оскільки контролер є окремим компонентом системи, то при виході його з ладу порушується безпека всієї системи, що створює додатковий фактор ризику для СКУД.

Розглянемо також питання про електроживленні СКУД. Професійні контролери, як правило, мають власний акумулятор, який підтримує працездатність контролера від декількох годин до декількох діб. При використанні ж простого автономного контролера без власного блоку живлення, необхідно купувати джерело безперебійного живлення, спеціально для цього призначений.

Таким чином, в разі відключення електрики контролера потрібне додаткове харчування - або контролер повинен мати акумулятор, що збільшує його вартість, або потрібно окреме джерело безперебійного живлення для контролера, що збільшує вартість всієї системи. У той час як наявність джерела безперебійного живлення для комп'ютера - це само собою зрозумілий факт.

					КНТЕУ 121 07-08.БР	Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата		

Підведемо підсумки. Програмна СКУД - це:

- миттєва реєстрація входу / виходу співробітника або відвідувача з виведенням інформації на комп'ютер охоронця
- організація контролю доступу співробітників при змінному графіку роботи;
- можливість складання звітів і графіків (табелі обліку використання робочого часу, діаграм, зведених таблиць);
- оперативність при складанні звітів;
- можливість розширення системи;
- можливість сполучення з системами відеоспостереження;
- можливість використання біометричних ідентифікаторів;
- облік і забезпечення безпеки автотранспорту;
- інтеграція з бухгалтерськими та фінансовими системами.

2.9. Висновок до розділу 2

Ознайомившись зі всіма аспектами розробки білінгових систем, описаних вище, ми почали моделювати архітектуру білінгової системи. Використовуючи різні підходи та моделюючи їх ми зупинились на варіанті використання CRM для забезпечення сумісності всіх компонентів, які потрібні для забезпечення функціонування білінгової системи. Було створено декілька діаграм для графічного зображення структури білінгової системи для полегшення розробки окремих модулів та зв'язків між ними

					КНТЕУ 121 07-08.БР	Аркуш
						35
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБКА ВЕБ ДОДАТКУ УПРАВЛІННЯ БІЛІНГОВОЮ СИСТЕМОЮ КОМПАНІЇ «ВЕРСУС»

3.1. Методика випробувань і результати експериментальної перевірки

Загальна мета проведення випробувань - визначення ступеня відповідності створеного програмного продукту (ПП) технічним завданням.

В життєвому циклі ПП можна виділити наступні види випробувань:

- випробування дослідного зразка ПП на повну відповідність вимогам технічного завдання;
- випробування робочої версії ПП, адаптованої до умов конкретного застосування;
- випробування версії модернізованого ПП при супроводі.

Найбільш повним і різнобічним випробуванням піддається ПП при первинній перевірці досвідченого зразка.

Для забезпечення повноти випробувань дослідного зразка ПП доцільно виділяти такі цілі випробувань і відповідні їм категорії тестування: [\[8\]](#)

- функціональне тестування - для перевірки повноти і коректності вирішення основних завдань при типових умовах;
- стресове тестування - випробування програм при граничних і критичних значеннях параметрів і умов експлуатації;

					КНТЕУ 121 07-08.БР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри	Криворучко О.В.				Розробка програмного забезпечення провайдерської компанії "Версус"	Стадія	Аркуш	Аркушів
Керівник	Сашньова М.В.					РЗ	36	64
Гарант	Цензура М.О.					Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.	Гудименко А.Л.							

- тестування використання ресурсів ЕОМ - для перевірки коректності розподілу пам'яті і продуктивності;
- ергономічне тестування - для перевірки зручності і простоти інтерфейсу програми;
- тестування документації - для перевірки повноти та відповідності внутрішнім і зовнішнім стандартам по структурі, оформленню.

Функціональне тестування - більш широке і найважче систематизована.

Набір випробувальних тестів повністю визначається функціональними завданнями і складністю ПП. Ці тести повинні забезпечувати перевірку і демонстрацію замовнику-користувачеві якості вирішення функціональних завдань, сформульованих в технічному завданні і конкретизованих в документації.

Оскільки вичерпне тестування для складних ПП неможливо, велике значення має уточнення областей варіювання тестових даних. Крім того, для випробувань стійкості функціонування при аномаліях на вході програми тестуються при значеннях змінних, які неможливі при нормальному функціонуванні джерел вихідних даних.

Функціональне тестування полягає в перевірці правильності виконання відповідних команд на відповідних формах. Оцінка правильності виконання команд здійснюється візуально і за відповідними повідомленнями.

В першу чергу нами було перевірено створення нових записів в БД, тобто нових абонентів. Результати того, що перевірка була успішно пройдена зображено на Рис. 3.3.

					КНТЕУ 121 07-08.БР	Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата		

Наступним кроком тестування була перевірка спроможності білінгової системи отримувати потрібну інформацію з БД.

WordPress надає можливість зручно маніпулювати своєю базою даних за рахунок php класу wpdb.

PHP клас - відноситься до ООП (об'єктно-орієнтованого програмування) і в традиційному своєму розумінні являє собою повністю самодостатній код, який повинен виконувати певну функцію. Наприклад, у випадку класу wpdb ми можемо робити різні операції з базою даних WordPress знаючи всього кілька "важелів" керування класом, які називаються методами.

Звертатися до методів класу wpdb потрібно обов'язково через глобальну змінну \$wpdb (це екземпляр класу wpdb). Також потрібно пам'ятати, що всередині звичайних функцій потрібно обов'язково глобалізувати \$wpdb, інакше вона буде простою змінною всередині функції, робиться це так:

```
global $wpdb;
```

З допомогою методів \$wpdb можна управляти довільними таблицями в базі даних, не обов'язково тільки тими, які були створені WordPress. Припустимо, серед інших таблиць WP, є таблиця newtable і нам потрібно вибрати всі поля id з неї. Реалізувати це можна таким SQL запитом використовуючи \$wpdb:

```
$Newtable = $wpdb->get_results ( "SELECT id FROM newtable");
```

Створювати свої таблиці рекомендується за допомогою функції dbDelta (). Або можна використовувати звичайний SQL запит і метод \$wpdb->query ('CREATE TABLE ...');

Для плагінів створення нових таблиць зазвичай вішається на хук активації плагіна, см. Register_activation_hook ().

					КНТЕУ 121 07-08.БР	Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата		

Створення окремого підключення до бази даних

Потрібно розуміти, що один об'єкт класу wpdb {} працює з однією БД - з поточної базою даних WordPress. Якщо потрібно працювати паралельно з будь-якої іншої БД, то потрібно створити ще один об'єкт класу wpdb із зазначенням нових параметрів з'єднання, відмінних від тих що вказані в wp-config.php. Робиться це так:

```
global $wpdb2;
```

```
$wpdb2 = new wpdb ( 'ім'я_юзера', 'пароль', 'назва_БД', 'localhost');
```

Якщо не вдалося підключитися, і потрібно обірвати PHP з повідомленням про цю помилку

```
if (! empty ($wpdb2-> error))
```

```
wp_die ($wpdb2-> error);
```

Готово, тепер використовуємо функції класу wpdb

```
$Results = $wpdb2-> get_results ( "SELECT * FROM table");
```

Для перевірки того, що наша база отримує зміни достатньо зайти в базу даних та побачити там внесену інформації, що продемонстровано на Рис. 7.

3.2. Опис ключових етапів розробки інформаційної системи

Перше з чим ми зіткнулись це необхідність додати до стандартний записів додаткові необхідні нам поля , такі як баланс, заборгованість, бонуси та інше.

Ми не просто так вибрали CMS Wordpress адже там такий інструментарій передбачено, називається він довільні поля. Приклад довільного поля наведено на Рис.3.1

					КНТЕУ 121 07-08.БР	Аркуш
						39
Зм.	Аркуш	№ докум	Підпис	Дата		

Користувачі поля

Додати нове користувацьке поле:

Назва	Значення
баланс	0

Додати користувацьке поле

Користувачі поля можуть бути використані щоби додавати додаткові метадані до запису, ви можете [використовувати їх у своїй темі](#).

Рис. 3.1 Довільні поля

Завдяки цьому інструменту додавати нові записи до БД, тобто нових клієнтів, можна прямо з під адміністративної панелі. Далі це було реалізовано за допомогою створеного нами плагіну.

Наступним етапом було створення веб-інтерфейсу для білінгової системи та створення зв'язку з базою даних.

Наступним кроком стало тестування білінгової системи на сервері.

3.3 Інтерфейс картки абонента

На Рис. 3.2 наведено приклад як буде виглядати інтерфейс картки абонента.

Ми розробили максимально простий для сприйняття та інформативний інтерфейс зі всім необхідним для вирішення елементарних завдань в повсякденній роботі оператора.

					КНТЕУ 121 07-08.БР	Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата		

Switchdown

Особистий кабінет

Баланс

перевір діалог

книгу слухавку

додати скаргу

Номер договору: 0000000001

Статус договору: Дійсний

Тип договору: звичайний

Ім'я: Гудименко Артем Леонідович

тел. 067-791-31-66

Баланс: 109.60 грн.

Заборгованість: 0.00 грн.

Бонуси: 22

стаж: 112 місяць(ів)

Адрес: м. Київ, вул. Драгоманова 14 , кв. 153

Паспортні дані: ТТ 943343

Заявки

Створити заявку

+ Заявка №1

+ Скарга №1

+ Скарга №2

Проблеми на лінії

Довідник команд

Довідник телефонів

Додати бінд

Біндинг

IP user 89.34.76.16 | netmask 255.255.255.0 | gateway 56.35.23.86

р 89.34.76.16 | mac 25:A6:88:7B:75:DE | switch 178.65.34.23 | port 5 | DHCP

Тарифний план – 100Мбит / с – 200грн.

Рис. 3.2 Інтерфейс картки абонента

3.4. Принцип роботи

Для забезпечення роботи картки абонента ми створили шаблон для записів Wordpress в який підтягується інформація з бази даних, в результаті вся інформація набуває привабливого вигляду за короткий період часу а головне все це автоматично.

Далі наведено приклад як метадані зберігає інформацію в базі даних.

					КНТЕУ 121 07-08.БР	Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

Недавнее	Избранное		meta_id	post_id	meta_key	meta_value
		☐	1455	214	_elementor_page_settings	a:1:{s:18:"eael_ext_toc_title";s:17:"Table of Cont...
		☐	1456	214	_elementor_data	[[{"id":"e91ae3e","elType":"section","settings":{"s...
		☐	1457	214	_elementor_css	a:6:{s:4:"time";i:1620855521;s:5:"fonts";a:0:{}s:5...
		☐	1458	215	_elementor_edit_mode	builder
		☐	1459	215	_elementor_template_type	wp-page
		☐	1460	215	_elementor_version	3.2.3
		☐	1461	215	_wp_page_template	elementor_canvas
		☐	1462	215	_elementor_page_settings	a:1:{s:18:"eael_ext_toc_title";s:17:"Table of Cont...
		☐	1463	215	_elementor_data	[[{"id":"e91ae3e","elType":"section","settings":{"s...
		☐	1464	215	_elementor_css	a:6:{s:4:"time";i:1620855521;s:5:"fonts";a:0:{}s:5...
		☐	1465	11	_elementor_controls_usage	a:9:{s:6:"button";a:3:{s:5:"count";i:10;s:15:"cont...
		☐	1466	11	_elementor_css	a:6:{s:4:"time";i:1620855645;s:5:"fonts";a:0:{}s:5...
		☐	1467	216	_edit_lock	1620887331:1
		☐	1468	217	_edit_lock	1620887481:1
		☐	1469	218	_edit_lock	1620887475:1
		☐	1470	220	_edit_last	1
		☐	1471	220	_edit_lock	1620889245:1
		☒	1472	220	Номер договору	0000000001
		☒	1474	220	Тип договору	звичайний
		☒	1475	220	Заборгованість	0
		☒	1476	220	стаж	112
		☒	1477	220	Адрес	м. Київ, вул. Драгоманова 14 , кв. 153
		☒	1478	220	Паспортні дані	ТТ 943343

Рис. 3.3 Нові записи в базі даних

					КНТЕУ 121 07-08.БР	Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата		

Для розуміння роботи БД в Wordpress на Рис. 3.4 наведена логічна структура бази даних

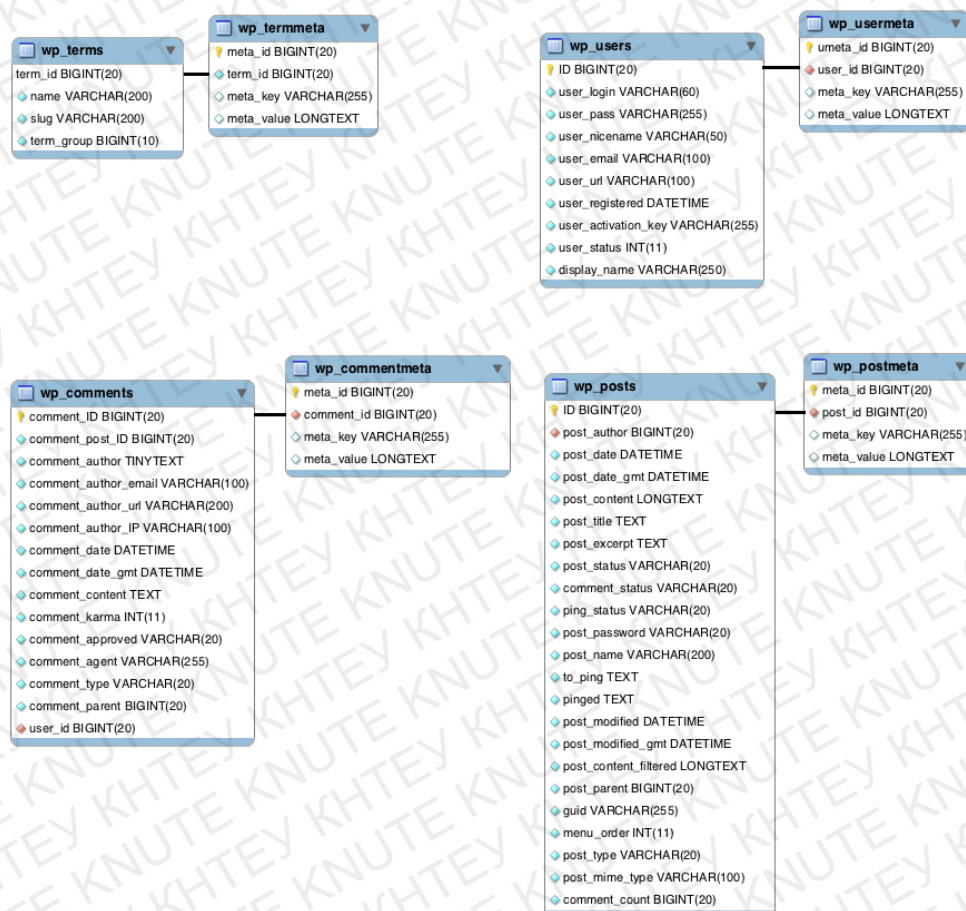


Рис. 3.4 Логічна структура бази даних wordpress

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-08.БР	43

3.5. Плагін

Наступним кроком було створення плагіну, оскільки на нашу думку внесення нових записів засобами Wordpress незручно. Плагін називається Artem.

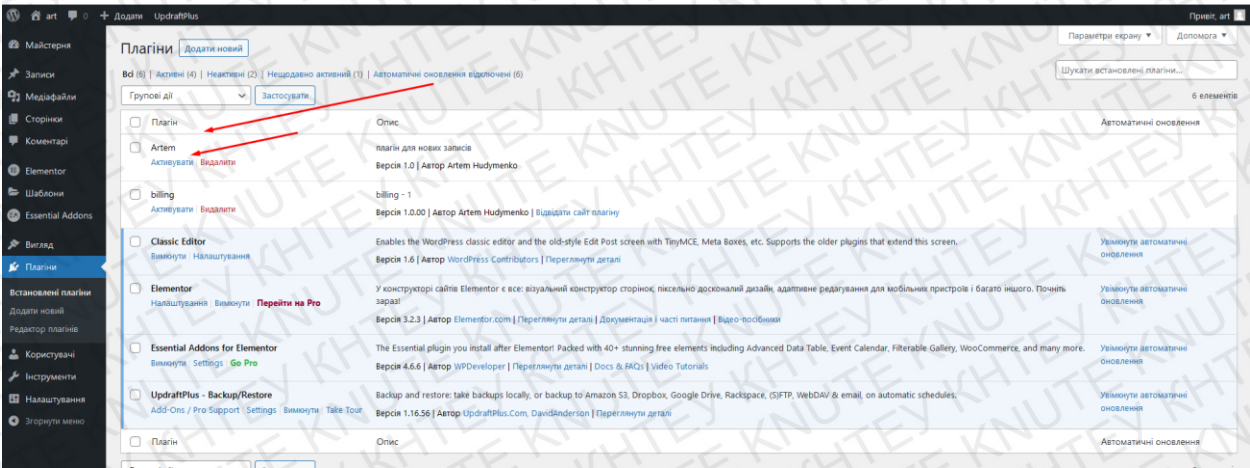


Рис. 3.5 Розташування плагіну

					КНТЕУ 121 07-08.БР	Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата		

На Рис. 3.6 наведено приклад інтерфейсу плагіну, за допомогою якого можна додавати нові записи до бази даних.

Рис. 3.6 Інтерфейсу плагіну

3.6. Висновки до розділу 3

В цьому розділі ми ознайомились з інструментом Wordpress та його принципом роботи щодо створення нових записів у базі даних. Було продемонстровану базу даних, її структуру, зв'язки між таблицями та поля які було створено в процесі роботи. Розроблено веб-інтерфейс білінгової системи та було забезпечено отримання необхідної інформації з бази даних. А також забезпечена можливість додавання нових записів до бази даних.

					КНТЕУ 121 07-08.БР	Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

При вирішенні завдання створення білінгової системи «Версус» в роботі показано весь процес створення програмного продукту. Було проведено аналіз існуючих рішень, моделювання білінгової системи для вибору оптимально шляху для вирішення поставленої задачі. Створення Інтерфейсу відповідно до вимог, продумана логіка і взаємодія компонентів шляхом роботи через одну CMS і базу даних. Проведене тестування відповідно до технічного завдання .

У процесі роботи були вивчені сучасні методи для вирішення подібних задач та задіяний самий ефективний з них. Таким чином, завдання вирішені в повному обсязі, мета досягнута.

					КНТЕУ 121 07-08.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка програмного забезпечення провайдерської компанії "Версус"	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					ВП	46	64
Керівник	Сашньова М.В.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Гудименко А.Л.				Висновки та пропозиції			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Загальні відомості про білінгові системи [Електронний ресурс] – Режим доступу до ресурсу: 1. <https://www.imprezz.in/what-is-billing-and-invoicing-software-advantages-of-billing-system-and-its-types/>.
2. Система керування вмістом [Електронний ресурс] – Режим доступу до ресурсу: <https://web24.pro/rozrobka-sajtiv-blog/systema-upravlinnya-sajtom/>.
3. ERP-системи [Електронний ресурс] – Режим доступу до ресурсу: <https://www.apptivo.com/blog/6-benefits-of-an-online-billing-system/>.
4. Діаграми для моделювання [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/9828818/>.
5. Діаграми діяльності [Електронний ресурс] – Режим доступу до ресурсу: <https://www.smartdraw.com/activity-diagram/>.
6. Діаграма класів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.smartdraw.com/class-diagram/>.
7. Тестування [Електронний ресурс] – Режим доступу до ресурсу: 7. <https://www.guru99.com/software-testing-introduction-importance.html>

					КНТЕУ 121 07-08.БР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри	Криворучко О.В.				Розробка програмного забезпечення провайдерської компанії "Версус"	Стадія	Аркуш	Аркушів
Керівник	Сашньова М.В.					СД	47	64
Гарант	Цензура М.О.					Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.	Гудименко А.Л.							
					Список використаних джерел			

ДОДАТКИ

Додаток А

код плагіну.

```
<?php
```

```
/*
```

Plugin Name: Artem

Plugin URI:

Description: плагін для нових записів

Version: 1.0

Author: Artem Hudymenko

Author URI:

License: GPLv2 or later

Text Domain: Artem

```
*/
```

```
if(!defined('ABSPATH')){
```

```
    die;
```

```
}
```

```
class Artem
```

```
{
```

```
public function register(){

    // register ost type

    add_action('init',[$this,'custom_post_type']);

    //enqueue

    add_action('admin_enqueue_scripts',[$this,'enqueue_admin']);

    add_action('wp_enqueue_scripts',[$this,'enqueue_front']);

    //Load template

    add_filter('template_include', [$this,'abon_template']);

    //Add menu admin

    add_action('admin_menu', [$this,'add_admin_menu']);

    //Add links to plugin page

    add_filter('plugin_action_links_' . plugin_basename(__FILE__),
    [$this,'add_plugin_setting_link']);

    add_action('admin_init',[$this,'settings_init']);

}
```

```
static function activation(){
```

```
    //update rewrite rules
```

```
    flush_rewrite_rules();
```

```
}
```

```
static function deactivation(){
```

```
    //update rewrite rules
```

```
    flush_rewrite_rules();
```

```
}
```

```
public function get_terms_hierarchical($tax_name,$current_term){
```

```
    $staxonomy_terms = get_terms($tax_name,['hide_empty'=>false,'parent'=>0]);
```

```
    if(!empty($staxonomy_terms)){
```

```
        foreach($staxonomy_terms as $term){
```

```
            if($current_term == $term->term_id){
```

```
                echo      '<option      value="'. $term->term_id.'"      selected>'. $term->name.'</option>';
```

```
            } else {
```

```
                echo '<option value="'. $term->term_id.'">'. $term->name.'</option>';
```

```

    }

    $child_terms =
    get_terms($tax_name,['hide_empty'=>false,'parent'=>$term->term_id]);

    if(!empty($child_terms)){
        foreach($child_terms as $child){
            echo '<option value="'. $child->term_id.'"> - '.$child->
            name.'</option>';
        }
    }
}
}
}

//Register settings

public function settings_init(){

    register_setting('booking_settings','booking_settings_options');

    add_settings_section('booking_settings_section',
    esc_html__('Settings','Artem'), [$this, 'settings_section_html'], 'Artem_settings');

```

```
add_settings_field('posts_per_page', esc_html__('Posts per page','Artem'),
[$this, 'posts_per_page_html'], 'Artem_settings', 'booking_settings_section');
```

```
add_settings_field('title_for_abons', esc_html__('Archive page title','Artem'),
[$this, 'title_for_abons_html'], 'Artem_settings', 'booking_settings_section');
```

```
}
```

```
//Settings section html
```

```
public function settings_section_html(){
```

```
    echo esc_html__("Hello, world!", 'Artem');
```

```
}
```

```
//Settings fields HTML
```

```
public function posts_per_page_html(){
```

```
    $options = get_option('booking_settings_options'); ?>
```

```
        <input type="text" name="booking_settings_options[posts_per_page]"
value="<?php echo isset($options['posts_per_page']) ? $options['posts_per_page'] :
""; ?>" />
```

```
<?php }
```

```

public function title_for_abons_html(){

    $options = get_option('booking_settings_options'); ?>

    <input      type="text"      name="booking_settings_options[title_for_abons]"
value="<?php echo isset($options['title_for_abons']) ? $options['title_for_abons'] :
""; ?>" />

```

```

<?php }

```

```

//Add settings link to plugin page

```

```

public function add_plugin_setting_link($link){

```

```

    $custom_link = '<a
href="admin.php?page=Artem_settings">'.esc_html__( 'Settings','Artem').'</a>';

```

```

    array_push($link, $custom_link);

```

```

    return $link;

```

```

}

```

```

//Add menu page

```

```

public function add_admin_menu(){

```

```

    add_menu_page(

```

```

        esc_html__( 'Artem Settings Page', 'Artem' ),

```

```

        esc_html__( 'Artem','Artem'),

```

```
'manage_options',  
  
    'Artem_settings',  
  
    [$this, 'Artem_page'],  
  
    'dashicons-admin-multisite',  
  
    100  
  
    );  
}
```

//Artem Admin HTML

```
public function Artem_page(){  
  
    require_once plugin_dir_path(__FILE__).'admin/admin.php';  
  
}
```

//Custom template for abons

```
public function abon_template($template){  
  
    if(is_post_type_archive('abon')){  
  
        $theme_files = ['archive-abon.php','Artem/archive-abon.php'];  
  
        $exist = locate_template( $theme_files, false);  
  
        if($exist != ""){  
  
            return $exist;  
  
        } else {
```

```
        return plugin_dir_path(__FILE__).'templates/archive-abon.php';

    }

}

return $template;
}

//Enqueue Admin

public function enqueue_admin(){

    wp_enqueue_style('ArtemStyle',          plugins_url('/assets/admin/styles.css',
__FILE__));

    wp_enqueue_script('ArtemScript',        plugins_url('/assets/admin/scripts.js',
__FILE__));

}

//Enqueue Front

public function enqueue_front(){

    wp_enqueue_style('ArtemStyle',          plugins_url('/assets/front/styles.css',
__FILE__));

    wp_enqueue_script('ArtemScript',        plugins_url('/assets/front/scripts.js',
__FILE__));

}

//Register CPT

public function custom_post_type(){
```

```
register_post_type('abon',  
[  
    'public' => true,  
  
    'has_archive' => true,  
  
    'rewrite' => ['slug'=>'abons'],  
  
    'label' => esc_html__( 'abon', 'Artem' ),  
  
    'supports' => ['title', 'editor', 'thumbnail']  
]  
);
```

```
$labels = array(  
  
    'name'          => _x( 'Locations', 'taxonomy general name', 'Artem' ),  
  
    'singular_name' => _x( 'Location', 'taxonomy singular name', 'Artem' ),  
  
    'search_items'  => __( 'Search Locations', 'Artem' ),  
  
    'all_items'     => __( 'All Locations', 'Artem' ),  
  
    'parent_item'   => __( 'Parent Location', 'Artem' ),  
  
    'parent_item_colon' => __( 'Parent Location:', 'Artem' ),  
  
    'edit_item'     => __( 'Edit Location', 'Artem' ),  
  
    'update_item'   => __( 'Update Location', 'Artem' ),  
  
    'add_new_item'  => __( 'Add New Location', 'Artem' ),  
  
    'new_item_name' => __( 'New Location Name', 'Artem' ),  
  
    'menu_name'     => __( 'Location', 'Artem' ),
```

```
);  
  
$args = array(  
    'hierarchical' => true,  
  
    'labels' => $labels,  
    'show_ui' => true,  
    'show_admin_column' => true,  
    'query_var' => true,  
    'rewrite' => array( 'slug' => 'abons/location' ),  
);
```

```
register_taxonomy('location','abon',$args);
```

```
$labels_type = array(  
    'name' => _x( 'Types', 'taxonomy general name', 'Artem' ),  
    'singular_name' => _x( 'Type', 'taxonomy singular name', 'Artem' ),  
    'search_items' => __( 'Search Types', 'Artem' ),  
    'all_items' => __( 'All Types', 'Artem' ),  
    'parent_item' => __( 'Parent Type', 'Artem' ),  
    'parent_item_colon' => __( 'Parent Type:', 'Artem' ),  
    'edit_item' => __( 'Edit Type', 'Artem' ),  
    'update_item' => __( 'Update Type', 'Artem' ),  
    'add_new_item' => __( 'Add New Type', 'Artem' ),  
    'new_item_name' => __( 'New Type Name', 'Artem' ),
```

```
'menu_name'      => __( 'Type', 'Artem' ),
    );

    $args_type = array(

        'hierarchical'      => false,
        'labels'            => $labels_type,
        'show_ui'           => true,
        'show_admin_column' => true,
        'query_var'         => true,
        'rewrite'           => array( 'slug' => 'abon/location' ),
    );

    register_taxonomy('type','abon',$args_type);
}

if(class_exists('Artem')){

    $Artem = new Artem();

    $Artem->register();

}

register_activation_hook( __FILE__, array( $Artem, 'activation' ) );

register_deactivation_hook( __FILE__, array( $Artem, 'deactivation' ) );
```