

ВСТУП

На сьогоднішній день ефективна автоматизація будь-якого виду діяльності підприємства може здійснюватися тільки на основі застосування інформаційних систем (ІС). У випадку ВЦ, вона надає співробітникам оперативну інформацію про хід виконання робіт, зайнятість працівників та грошовий оборот. Ця інформація необхідна для правильного прийняття рішень у процесі роботи підприємства.

Об'єктом автоматизації виступив бізнес-процес, невеликого видавничого центру Державного університету телекомунікацій. Чисельність працівників фірми нараховує десять чоловік, з них четверо є обслуговуючим персоналом (бухгалтер, 2 менеджера, Оператор-адміністратор), директор і п'ять інженерів-майстрів. Організація надає свої послуги фізичним особам як на території замовника так і в стаціонарних умовах.

На поточний момент облік робіт по друку поліграфії проводиться програмним забезпеченням на базі застарілих технологій та обмеженого функціоналу. У наслідок цього суттєво збільшується час на обслуговування клієнтів та неможливість оперативного отримання даних про хід робіт.

Для покращення ситуації було вирішено оновитися до більш сучасної та продуктивнішої. Впровадження нової ІС дозволить підвищити продуктивність праці співробітників видавничого центру, якість і швидкість обслуговування клієнтів, за рахунок більш якісного та швидкого обслуговування клієнтів, оперативного доступу до інформації про хід робіт і скорочення часу на аналіз даних про виконані роботи за рахунок зручних звітів.

Метою даної роботи є розробка інформаційної системи підтримки діяльності видавничого центру Державного університету телекомунікацій.

Для досягнення цієї мети треба вирішити наступні задачі:

- провести аналіз діяльності видавничого центру;
- провести огляд існуючих програмних засобів підтримки діяльності видавничих центрів;

- визначити вимоги до інформаційної системи та виконати планування робіт;
- змодельовати процеси діяльності видавничого центру;
- спроектувати базу даних інформаційної системи підтримки діяльності видавничого центру;
- реалізувати інформаційну систему пі підтримки діяльності видавничого центру та провести її тестування.

Об’єктом дослідження є область прикладного та веб-програмування.

Предметом дослідження є інформаційна система по обробці та оптимізації виконання замовлень видавничого центру.

Методи дослідження в роботі використані такі: пошуковий по наявній методичній та науковій літературі із аналізом знайденого матеріалу, проектування, теоретичне моделювання, систематизація, тестування, аналіз документації та результатів діяльності дослідників з проблеми проведеного дослідження.

Джерельна база дослідження. Робота ґрунтується на аналізі законодавства України та зарубіжних країн, науково-методичної літератури, методичних посібників, наукових статей, періодичних видань та напрацювань сучасних програмістів та вчених і дослідників в галузі створення веб-додатків.

Теоретична та практична цінність роботи полягає в наявності теоретичного матеріалу по дослідженню, відсіяного з-поміж іншого в процесі пошуку інформації по темі, та в систематизації матеріалу напрямку дослідження. Проведене дослідження має більш глибокий ступінь аналізу напрямку дослідження, спираючись на попередні дослідження вчених, дисертантів та дослідників напрямку дослідження. В проектній частині створено структуру бази даних інформаційної системи підтримки діяльності видавничого центру та розроблено програмний код. Розроблено технічне завдання на розробку інформаційної системи.

Структура роботи. Робота складається з 70 листів друкованого тексту, 26 рисунків, 4 таблиць та 3 додатків і налічує 72 джерела використаної літератури.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний аналіз роботи видавничого центру Державного університету телекомунікацій

Видавничі центри надають широкий спектр послуг з видавничої підтримки і друку поліграфії та іншої продукції. Об'єктом автоматизації став видавничий центр Державного університету телекомунікацій, де, у результаті збільшення клієнтів і, як наслідок навантаження на персонал, виникла необхідність модернізації інформаційного забезпечення процесу обслуговування клієнтів.

Зараз, для підтримки діяльності використовується класичний спосіб організації та обліку виконання робочого процесу – на папері. У видавничому центрі Державного університету телекомунікацій є особа відповідальна за ведення обліку всіх заявок від отримання заявки до її закриття.

Тому обставини склались таким чином, що вже не розглядається питання: чи потрібно оновлювати існуючу систему обліку замовлень до сучасних вимог, приймаючи цю необхідність як даність. Наразі питання змістилося в іншу площину, якими програмними засобами її можливо модернізувати. Проаналізуємо існуючі програмні засоби для підтримки діяльності видавничих центрів, опираючись на які можна виділити переваги кожного з них та сформулювати вимоги для проектування власного додатку підтримки роботи видавничого центру Державного університету телекомунікацій. Зразки документів для ведення робочого процесу наведені в Додатку Г.

1.2 Огляд існуючих програмних засобів для підтримки діяльності видавничих центрів

Програмне забезпечення для видавничих центрів має ключову роль в оптимізації процесів управління, тому його вибір має бути виважений та обґрунтований.

На сьогоднішній день існує значна кількість варіантів інформаційних систем для видавничих центрів. Умовно їх можна поділити на такі типи:

- прикладні програми;
- веб-додатки;
- побудовані на базі програм бухгалтерського обліку (1С та інших).

Для аналізу були обрані найпопулярніші представники у своїх категоріях.

Сервіс-Центр

На даний час у компанії використовується ІС на базі Microsoft Access 2003. Вона була розроблена у 2003 році силами компанії. За час використання з 2003 року було виявлено ряд недоліків:

- мала продуктивність;
- неможливість доступу до ІС не з внутрішньої мережі;
- відсутність розділення прав доступу;
- необхідність встановлення MS Access 2003 для роботи з ІС.

До переваг використання існуючої ІС можливо віднести:

- добре налаштований функціонал до потреб ВЦ;
- безпека зберігання даних клієнтів.



Рисунок 1.1 – Головна сторінка ІС «Сервис-Центр»

Процес-Онлайн [3]

Програма для різних типів видавничих бізнесів, що займаються друком і обслуговуванням будь-яких виробів: від велосипедів до комп'ютерів, від побутової техніки до смартфонів. Дозволяє обробляти замовлення, контролювати співробітників, вести складський облік, друкувати документи і звіти. Автоматично відправляє SMS клієнтові про завершення робіт над замовленням, а менеджера повідомляє по email про хід роботи над замовленням.

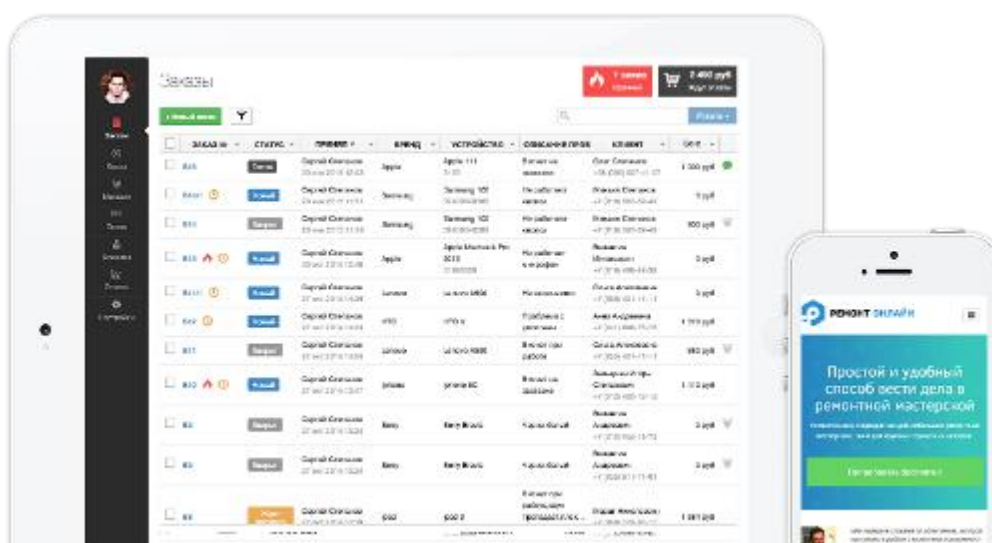


Рисунок 1.2 – Головна сторінка ІС «Процес-Онлайн»

До переваг даної програми можна винести:

- велика кількість вбудованих модулів (sms-інформування, складський облік та інші).
- доступність онлайн 24 / 7;
- постійні оновлення продукту;
- мобільний додаток.

Слабкими сторонами є:

- щомісячна абонплата;
- неможливість гарантувати безпеку персональних даних клієнтів;
- можливість втрати дані у наслідок припинення підтримки проекту.

1С: Предприятие

Рішення призначене для автоматизації діяльності підприємств, що надають послуги різного характеру [4].

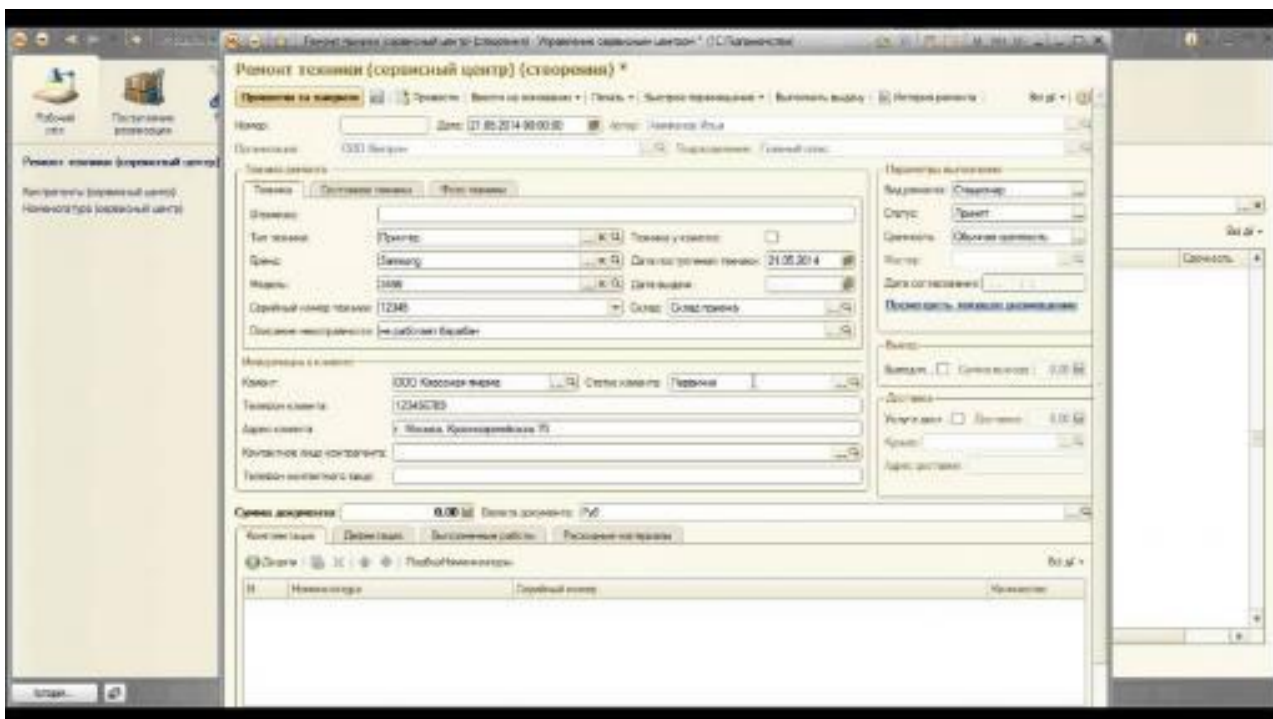


Рисунок 1.3 – Головна сторінка ІС «1С: Підприємство»

Конфігурація «Управление центром» розроблена на основі типової конфігурації «Управление небольшой фирмой», редакція 1.4 системи програм «1С: Підприємство» зі збереженням усіх можливостей і механізмів типового рішення.

Дозволяє організувати єдину інформаційну систему для управління різними аспектами його діяльності:

- маркетингом і продажами;
- постачанням і закупівлями, взаємодією з виробниками;
- складом і виробництвом;
- підмінним фондом;
- взаєморозрахунками і доставками;
- плануванням, підготовкою та обліком діяльності;
- роздрібними продажами;
- грошовими коштами;
- зарплатою і персоналом;
- майном;

- фінансами.

До недоліків даної ІС можна віднести:

- надмірність і громіздкість;
- велика ціна;
- Указ Президента України №133 / 2017, «Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)» який ускладнює використання продуктів на базі ІС.

Для більш наглядного порівняння вищеописаних ІС була побудована таблиця 1.1.

Таблиця 1.1

Порівняння розглянутих ІС

	Сервис-Центр	Процес-Онлайн	ІС: Підприємство
Інтеграції з іншим ПЗ	-	-	+
Безпека даних	+	-	+
Online-доступ	-	+	+
Юридичні аспекти	+	-	-
Продуктивність	-	+	-
Ціна	+	-	-

За результатами порівняльного аналізу можна зробити висновок, що кожен з них має як певні плюси так і мінуси. Також був розглянутий варіант розробки власного веб-додатку. Шляхом аналізу узагальнених даних була побудована наступна порівняльна таблиця.

Таблиця 1.2

Порівняння типів ПЗ

Функції	Тип інформаційної системи			
	Прикладна програма	Сторонній веб-додаток	Власний веб-додаток	На базі ІС
Інтеграції з іншим ПЗ	Ні	Ні	Ні	Так
Масштабування	Ні	Ні	Так	Так
Безпека даних	Так	Ні	Так	Так
Ціна	Так	Так	Так	Ні
Online-доступ	Ні	Так	Так	Так
Додатковий функціонал	Ні	Так	Ні	Так

Для оцінки правильності вибору типу ПЗ були опитані співробітники видавничого центру. В результаті, на основі математично обґрунтованого методу аналізу ієрархій Томаса Саати [5], був проведений аналіз отриманих даних. Були побудовані ієрархії відносин та прораховані вагові коефіцієнти за допомогою матриць парних порівнянь. Розрахунки проводились за допомогою системи підтримки прийняття рішень (СППР) «Вибір» [6].

СППР «Вибір» – аналітична система, заснована на методі аналізу ієрархій (МАІ), є простим і зручним засобом, яке допоможе оцінити альтернативи по кожному з факторів.

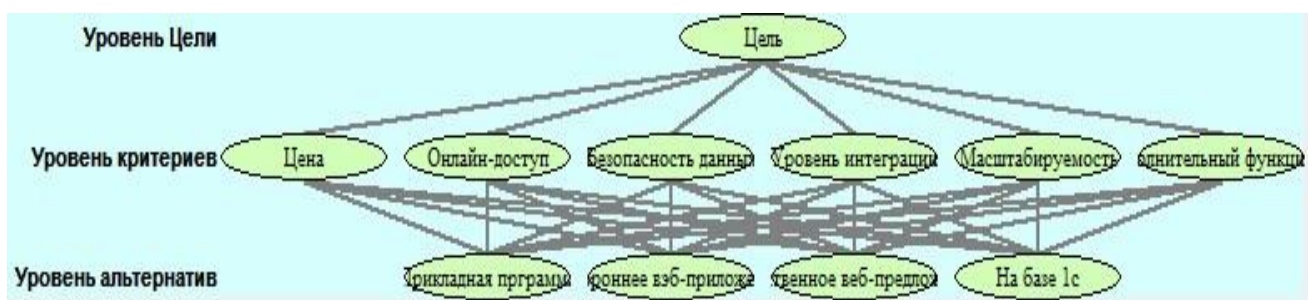


Рисунок 1.4 – Входні дані СППР «Вибір»

Після розрахунків програми було отримано результат, котрий обґрунтовує доцільність розробки власного веб-додатку. Одними з найвпливовіших вагових коефіцієнтів виявились ціна та безпека даних. Вони зіграли ключову роль у виборі типу програмного забезпечення. Результат у вигляді кругової діаграми представлено на рис. 1.5. Звіт роботи програми СППР «Вибір» розміщено у Додатку В.

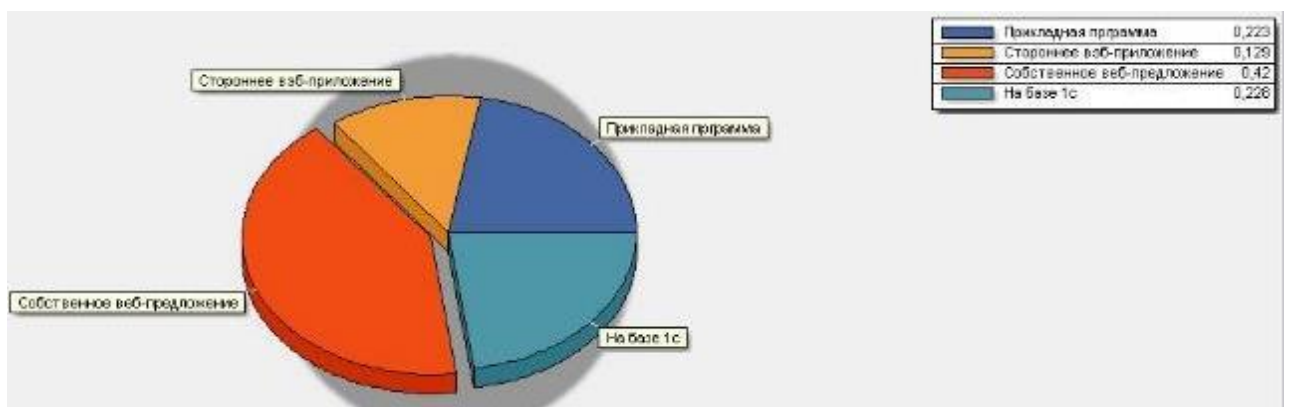


Рисунок 1.5 – Результат аналізу аних за допомогою СППР «Вибір»

За результатами проведеного аналізу видно, що поточна ІС безнадійно застаріла і необхідна міграція на більш сучасну. Серед розглянутих

альтернатив найбільш оптимальним є розробка власного додатку, оскільки лише він дозволить реалізувати більшість вимог, буде дешевший у реалізації і гарантуватиме приватність даних клієнтів.

2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі проектування

Інформаційна система повинна забезпечувати автоматизацію роботи видавничого центру, а саме вирішувати задачі прийому заявок, ведення обліку заявок, розрахунок зароблених коштів, формування спеціалізованих звітів.

Використання інформаційної системи підтримки діяльності видавничого центру спрямоване на підвищення продуктивності праці співробітників видавничого центру, якості та швидкості обслуговування клієнтів.

Впровадження інформаційної системи передбачає такі результати:

- введення і зберігання інформації по заявках, в дану видавничу організацію;
- редагування інформації про замовлення;
- пошук інформації по певним користувачам та критеріям;
- додавання і коректування даних про клієнтів і співробітників ВЦ;
- отримання звіту по заявках певного клієнта.

Також було сформоване та затверджене технічне завдання, яке було розміщене у додатку А.

2.2 Вибір методів реалізації інформаційної системи обробки замовлень видавничого центру Державного університету телекомунікацій

Для реалізації програмної частини існує велика кількість мов програмування, СУБД, веб-серверів та фреймворків на яких можна створити веб-додаток будь-якої складності. Важливо грамотно обрати засоби для реалізації ІС. Враховуючи поставлені вище завдання та попередній досвід найбільш оптимальним засобом для розробки такого продукту є застосування мови розмітки документів у Всесвітній павутині – HTML [7], формальної

мови опису зовнішнього вигляду документа, написаного з використанням мови розмітки – CSS [8], прототипно-орієнтованої сценарного типу мови програмування JavaScript [9], скриптової мови програмування загального призначення PHP [10] з використанням CodeIgniter Framework [11], веб-сервера APACHE [12] та баз даних MySQL [13].

HTML – мова розмітки гіпертекстових документів. HTML-документи лежать в основі Веб, і відображаються із допомогою веб-браузерів. Разом із видимою інформацією, HTML-документи містять додаткові метадані, такі як, наприклад, мова тексту, автор документа, стислий підсумок [7].

Каскадні таблиці стилів (CSS) є потужним інструментом для керування зовнішнім виглядом вашого веб-сайту, впливає на уявлення документа або набору документів.

PHP – мова програмування, створена для генерування HTML-сторінок на веб-сервері і роботі з базами даних [10].

В даний час, підтримується переважною більшістю хостинг-провайдерів. В області програмування для веб, PHP – одна з популярних скриптових мов завдяки своїй простоті, швидкості виконання, багатій функціональності і великій кількості супровідної документації.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

CodeIgniter – популярний MVC фреймворк з відкритим вихідним кодом, написаний на мові програмування PHP, для розробки повноцінних веб-систем і додатків. Даний фреймворк має такі можливості:

- висока продуктивність;
- патерн Модель-вид-контролер;

- якісна і повна документація з прикладами, а також велика спільнота і Wiki;
- безліч відео-уроків, які можна знайти як на офіційному сайті, так і на сторонніх ресурсах;
- фреймворк дає свободу програмісту, не створюючи будь-яких структурних обмежень і конвенцій;
- програмісту не потрібно вчитися користуватися генераторами коду з командного рядка;
- CodeIgniter працює практично на будь-якому хостинговому плані, який має підтримку PHP версії 5.1 і вище [11].

CodeIgniter вважається одним з найшвидших і не вимогливих до ресурсів фреймворків.

JavaScript (JS) – динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [9].

MySQL – вільна система керування реляційними базами даних. Зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування [13].

Apache HTTP-сервер – відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем.

Web-сервер Apache є самостійним, некомерційним, вільно розповсюджуваним продуктом. Продукт підтримує безліч можливостей, багато з яких реалізовані як скомпільовані модулі, які розширюють основні функціональні можливості. Вони різняться від серверної підтримки мов програмування до схем аутентифікації. Існують інтерфейси для підтримки мов програмування Perl, Python, Tcl і PHP [12].

Популярні методи стискування на Apache включають зовнішній модуль `mod_gzip`, створений для зменшення розміру веб-сторінок, переданих по HTTP.

Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ВИДАВНИЧОГО ЦЕНТРУ ДЕРЖАВНОГО УНІВЕРСИТЕТУ ТЕЛЕКОМУНІКАЦІЙ

3.1 Структурно-функціональне моделювання діяльності видавничого центру

Функціональна модель представляє з необхідним ступенем деталізації систему функцій, які в свою чергу відображають свої взаємовідносини через об'єкти системи. Вона являє собою ієрархію взаємопов'язаних діаграм, кожна з яких представляє підсистему або її окрему компоненту. Вершина цієї структури містить загальний опис системи, який деталізується на наступних рівнях декомпозиції.

Для створення функціональної моделі використано програмний продукт компанії Computer Associates – Erwin Process Modeler.

Розглянемо головний напрямок діяльності видавничого центру Державного університету телекомунікацій – друк поліграфії. Контекстну діаграму to-be даного процесу представлено на рис. 3.1.



Рисунок 3.1 – Контекстна діаграма to-be

Кожна діаграма складається з блоків діяльності, що відображають одну чи кілька функцій та стрілок, що являють собою ресурси, потрібні для забезпечення діяльності. Діяльність (activity) є поіменованим процесом, функцією або задачею, що виконується протягом певного інтервалу часу та

формує результати своєї роботи. Діяльність зображається прямокутником, у середині якого подається назва у формі дієслова. Блоки на діаграмі розташовують у порядку домінування з позицій автора діаграми. Найбільш домінуючий блок розташовують у верхньому лівому, а найменш домінуючий – в правому нижньому куті. Таким чином, структура діаграми показує, які функції здійснюють вплив на інші. Всі блоки нумеруються в порядку домінування. Стрілки (arrow) або дуги зображують об'єкти, зв'язують блоки та відображають їх взаємодію. Дуги зображаються лініями зі стрілками на кінцях. Імена стрілок можуть приєднуватись до них за допомогою зигзагів (squiggle). Між об'єктами (дугами) та функціями (діяльностями) можливі відношення: вхід (input), контроль (control), вихід (output) та механізм (mechanism). Кожне з названих відношень зображається дугою, зв'язаною з певною стороною блока: ліва сторона призначена для вхідних дуг, верхня – для дуг контролю, права – для вихідних дуг, нижня – для дуг механізмів.

Вхідними стрілками до функції «Виконання заявок» є:

- клієнт;
- раявки.

На виході будуть виконана заявка та звіт.

Будь-яка діяльність діаграми може бути деталізована на діаграмі декомпозиції, яка є дочірньою щодо батьківської діаграми. На діаграмі декомпозиції зображаються блоки діяльності, що представляють функції, які є складовими батьківської діяльності у процесі детального аналізу. Всі дуги батьківської діяльності автоматично переносяться на дочірню діаграму під час її створення. Для зв'язку батьківської та дочірньої діаграм користуються S-номерами, що дають можливість виключити неоднозначність зв'язку між діаграмами.

На першому рівні деталізації моделі головна батьківська діаграма декомпозиціонується на наступні блоки представлено на рис. 3.2.

- створення заявки;
- друк;

– видача пристрою.

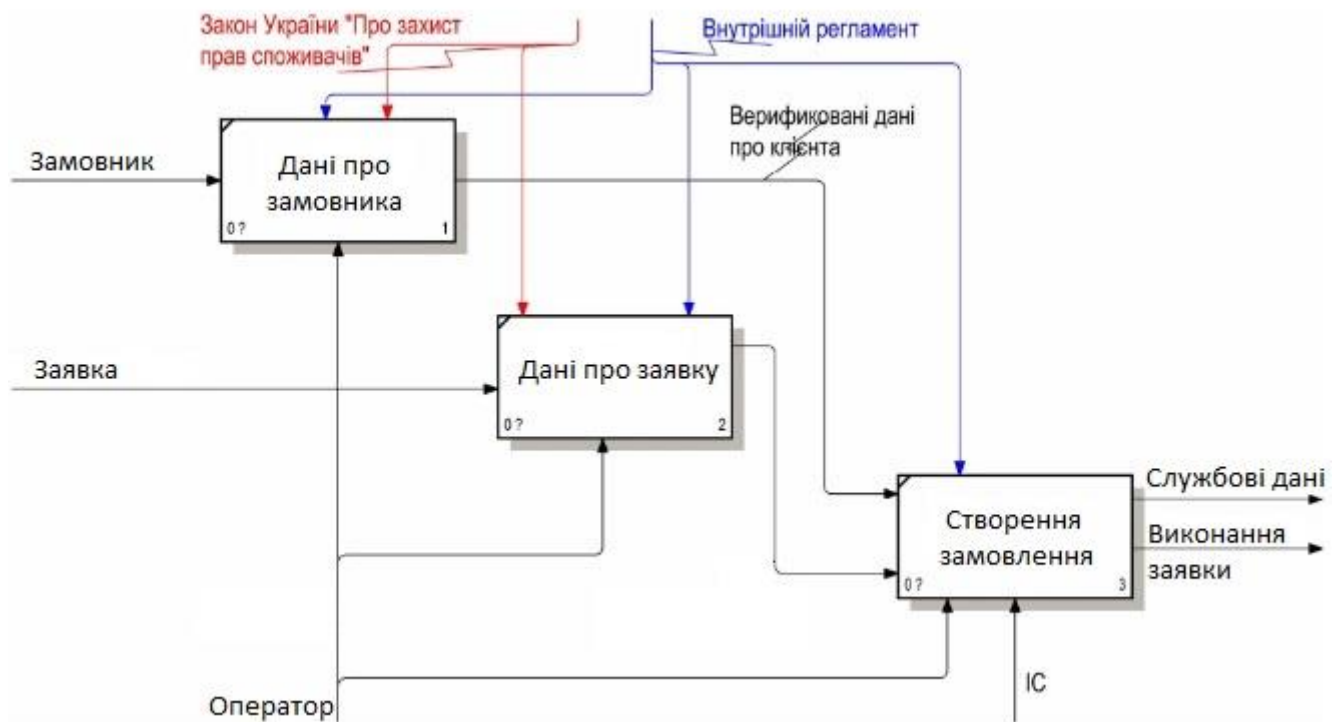


Рисунок 3.2 – Діаграма декомпозиції IDEF0 «Створення заявки»

Вхідними стрілками до діяльності «Отримати дані про клієнта» є «Клієнт (данні про клієнта)»; вихідними – «Верифіковані данні клієнта»; стрілками контролю – «Внутрішній регламент» та «Закон України «Про захист прав споживачів»»; стрілками механізмів – «Оператор».

Вхідними стрілками до діяльності «Дані про заявку» є «Дані про замовлення»; вихідними – стрілками контролю – «Внутрішній регламент» та «Закон України «Про захист прав споживачів»»; стрілками механізмів – «Оператор».

Вхідними стрілками до діяльності «Створення заявки» є «Верифіковані данні клієнта»; вихідними – «Службові дані» та «Виконання заявки»; стрілками контролю – «Внутрішній регламент» та «Закон України «Про захист прав споживачів»»; стрілками механізмів – «Оператор».

3.2 Моделювання бази даних інформаційної системи підтримки діяльності видавничого центру

Реляційна модель даних – це модель даних, де текстова чи числова інформація, що зображується за допомогою таблиць. Кожна таблиця, яка називається відношенням, складається з рядків, які називаються кортежами, та стовпчиків, які називаються атрибутами. Реляційна модель визначає представлення даних (структура), захищеність від некоректних змін (цілісність) та операції, що можуть бути виконані з даними (операції з даними).

ER-діаграми зручні тим, що процес виділення сутностей, атрибутів і зв'язків є ітераційним. Розробивши перший наближений варіант діаграм, вони уточнюються, опитуючи експертів предметної області.

При розробленні інформаційної моделі в термінах ER-моделей ми повинні отримати таку інформацію про предметну область:

- список сутностей предметної області;
- список атрибутів сутностей;
- опис взаємозв'язків між сутностями.

У першу чергу повинна бути вивчена предметна область і процеси, що відбуваються в ній. Для цього були опитані співробітники фірми, прочитана документація, вивчені форми замовлень, накладних і т. п. Тобто проводимо аналіз предметної області.

У ході бесіди з менеджером у продажах, з'ясовується перелік функцій, які проектувана система повинна виконувати:

- зберігати інформацію про клієнтів;
- зберігати інформацію про персонал;
- зберігати інформацію про пристрої;
- зберігати інформацію про проведені роботи.

Виділимо всі іменники в цих реченнях – це будуть потенційні кандидати на сутності й атрибути, і проаналізуємо їх (незрозумілі терміни будемо виділяти знаком питання):

- клієнт (Client) – явний кандидат на сутність;
- працівник (Employee) – явний кандидат на сутність;

- замовлення (Order) – явний кандидат на сутність;
- Виконання заявки (service) – явний кандидат на сутність;
- (?) посада співробітника (position) – це швидше за все атрибут, але атрибут невідомо якої сутності?

Відразу виникає очевидний зв'язок між сутностями – «клієнти можуть мати багато замолень». Перший варіант діаграми має такий вигляд:

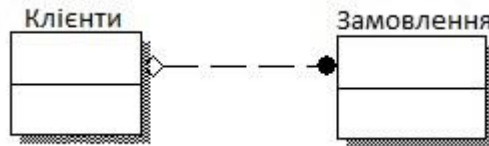


Рисунок 3.4 – Зв'язок між сутностями Client та Order

Також виникає очевидний зв'язок між сутностями – «багато замовлень може бути на виконанні». Другий варіант діаграми має такий вигляд:

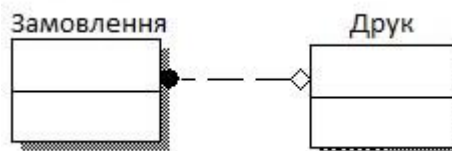


Рисунок 3.5 – Зв'язок між сутностями Order та Service

Припустимо, що клієнт замовляє друк фото, котрий обслуговує спеціаліст за спеціалізацією. Різні посади з певним окладом, можуть обіймати декілька людей. Таким чином, після уточнення діаграма буде мати такий вигляд:

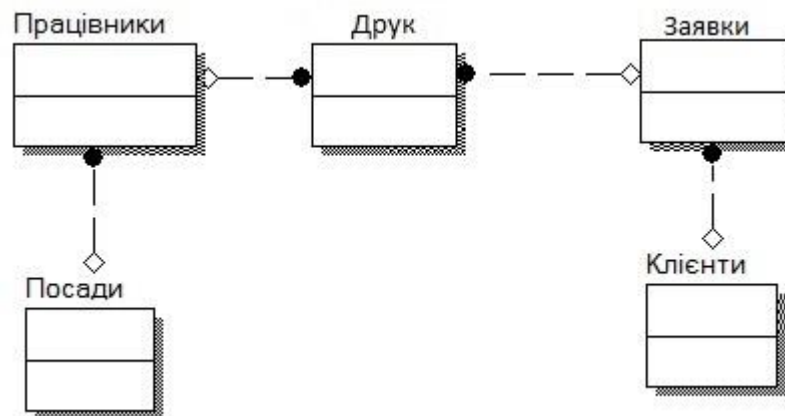


Рисунок 3.6 – Відображення зв'язків між виділеними сутностями

Розглянемо властивості (кандидати в атрибути) сутностей:

Клієнт

Клієнт має атрибути: номер клієнта, прізвище, ім'я, по батькові, адреса, телефон, Номер паспорту.

Працівник

Спеціаліст має атрибути: номер працівника, прізвище, ім'я, по батькові, номер посади, телефон, адреса, пошта.

Замовлення

Замовлення має атрибути: номер заявки, дата подачі, ціна, розмір фото, кількість фото, установлена якість, номер клієнта (володаря).

Посада

Посада має такі атрибути: номер посади, назва посади, ставка.

Виконання заявки

Виконання заявки має атрибути: номер заявки, тип гарантії, заявлену кількість і якість фото, дата прийому, дату виконання, дата видачі, код спеціаліста, ціну роботи.

Доповнимо ER-діаграму атрибутами сутностей (див. рис. 4), що буде остаточним варіантом інформаційної моделі предметної області. Остання є вхідними даними для розроблення логічної моделі.

Всі відношення відповідають вимогам реляційної БД до нормальних форм, що дозволяє завершити етап проектування логічної моделі і надати матеріал для подальшого етапу проектування БД – розроблення фізичної моделі БД

3.3 Моделювання варіантів використання інформаційної системи підтримки діяльності видавничого центру

Для того, щоб точніше зрозуміти як повинна працювати система, було описано функціональності системи через варіанти використання (Use Case або прецеденти).

Варіанти використання – це опис послідовності дій, які може здійснювати система у відповідь на зовнішні дії користувачів або інших

програмних систем. Варіанти використання відображають функціональність системи.

Діаграми варіантів використання описують функціональне призначення системи або те, що система повинна робити. До складу моделі варіантів використання входить опис акторів, варіантів використання (ВВ) програмного продукту (ПП) та діаграми варіантів використання (рис. 3.8).

Адміністратор-Оператор – співробітник, котрий створює заявки в ІС при зверненні клієнту до ВЦ;

Спеціаліст – співробітник, котрий обслуговує заявки в ІС.

Менеджер – співробітник, котрий керує хід виконання заявок та оброблює звіти.

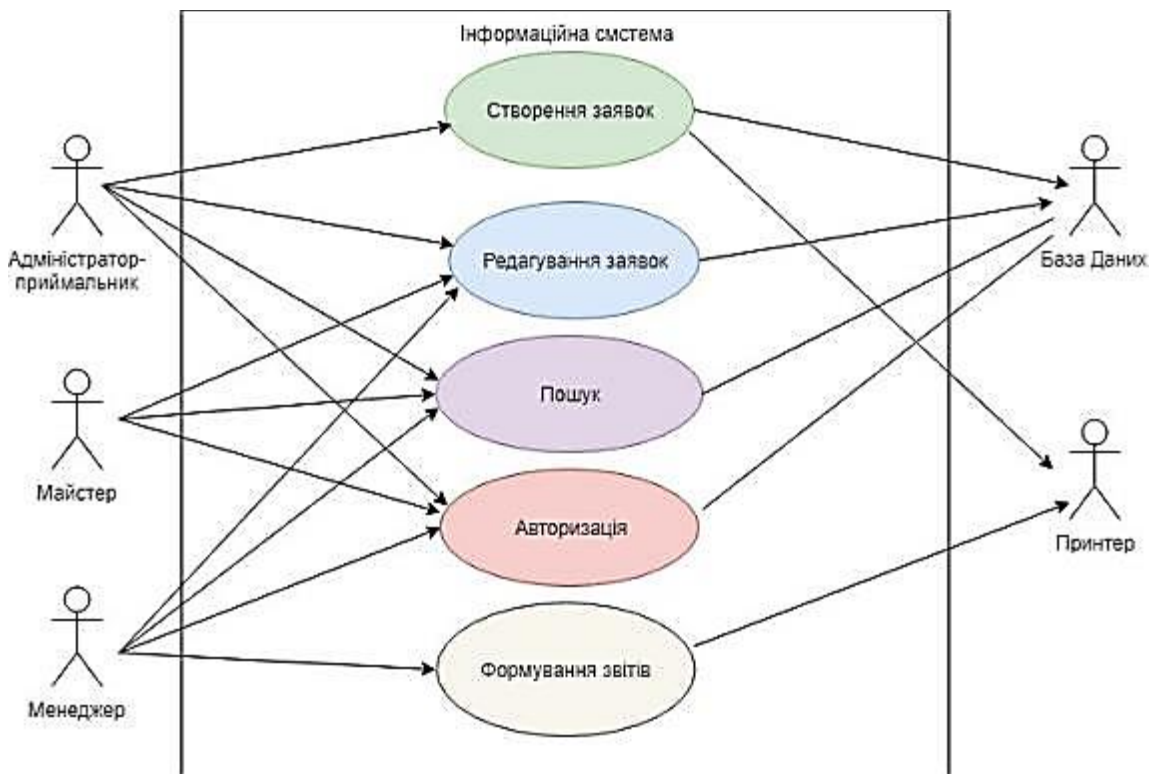


Рисунок 3.7 – Діаграма варіантів використання

Варіанти використання:

а) ВВ Авторизація – ВВ надає права роботи в ІС з певним рівнем доступу.

б) ВВ Створення заявок – ВВ дозволяє адміністратору-Оператору створювати нові заявки на друку поліграфії.

в) ВВ Редагування заявок – ВВ дозволяє адміністратору-Оператору, спеціалістам та менеджерам редагувати заявки.

г) ВВ Пошук – ВВ дозволяє адміністратору-Оператору, спеціалістам та менеджерам шукати заявки по критеріям.

д) ВВ Формування звітів – ВВ дозволяє менеджерам формувати та друкувати звіти.

4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Тестування та запуск компонентів ІС

Web-додаток вирішено розробити як web-сайт з використанням CodeIgniter framework. CodeIgniter – дуже легка, добре працююча система. Він ідеальний як і для новачка (тому що легкий для освоєння), так і ідеально підходить для великих і вимогливих веб-додатків. CodeIgniter активно розвивається і має повну, легку для розуміння документацію.

CodeIgniter framework – це каркас, призначений для створення динамічних веб-сайтів, мережевих додатків, сервісів або ресурсів. Також, це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми.

CodeIgniter використовує підхід Модель-Вид-контролер (MVC), що забезпечує поділ між логікою і презентацією. Це особливо добре для проектів де дизайнер працює з файлами шаблонів, де присутність коду зведено до мінімуму. Структура Моделі MVC представлена на рис. 4.1.

Головна мета застосування цієї концепції полягає в відділенні бізнес-логіки (моделі) від її візуалізації (представлення, виду). За рахунок такого поділу підвищується можливість повторного використання коду. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах і/ або з різних точок зору. Зокрема, виконуються наступні завдання:

- до однієї моделі можна приєднати кілька видів, при цьому не зачіпаючи реалізацію моделі. Наприклад, деякі дані можуть бути одночасно представлені у вигляді електронної таблиці, гістограми і кругової діаграми;
- не змінюючи реалізацію видів, можна змінити реакції на дії користувача (натискання мишею на кнопки, введення даних) – для цього досить використовувати інший контролер;

- ряд розробників спеціалізується тільки в одній з областей: або розробляють графічний інтерфейс, або розробляють бізнес-логіку. Тому можливо добитися того, що програмісти, які займаються розробкою бізнес-логіки (моделі), взагалі не будуть обізнані про те, яке уявлення буде використовуватися.



Рисунок 4.1 – Структура MVC

Модель (Model) надає дані і реагує на команди контролера, змінюючи свій стан.

Вид (View) відповідає за відображення даних моделі користувачеві, реагуючи на зміни моделі.

Контролер (Controller) інтерпретує дії користувача, сповіщаючи модель про необхідність змін.

CodeIgniter написаний на PHP, а сервером бази даних є MySQL. На рис. 4.2 представлена структура файлів та папок Фреймворку. У теці application міститься кілька каталогів:

- / Config – тут зберігаються всі налаштування для поточного проекту.

- / Controllers – в цю папку потрібно зберігати всі створені вами контролери.

- / Error – шаблони для сторінок-помилки.

- / Helpers – папка призначена для зберігання створених вами хелперів.

- / Hooks – директорія для «пасток».

/ Language – папка для зберігання мовних файлів.

/ Libraries – все створені вами бібліотеки повинні бути тут.

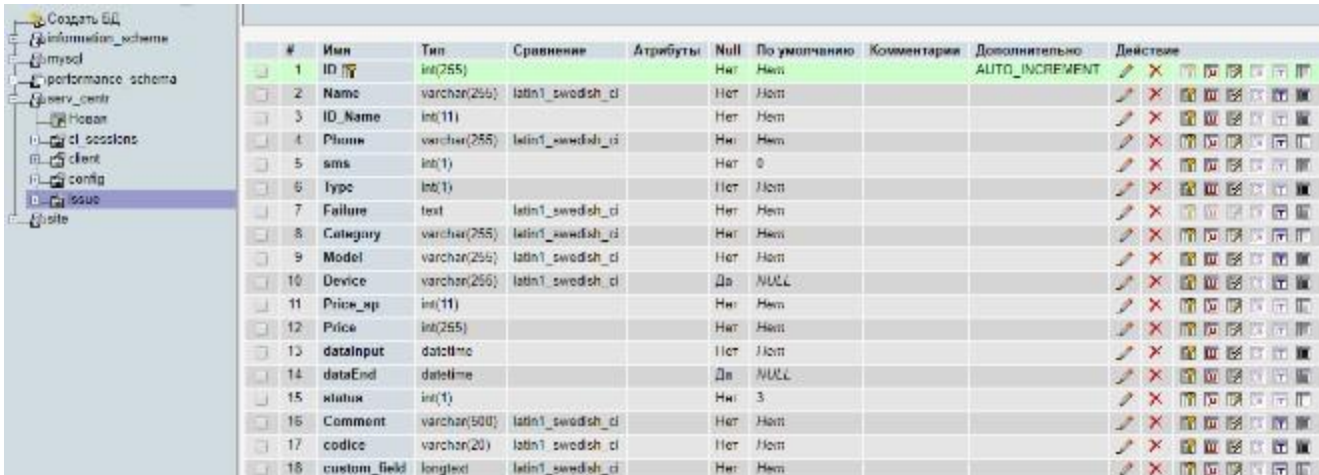
/ Models – папка з моделями даних.

/ Views – директорія для зберігання уявлень вашого проекту.

Для того щоб створити web-додаток спочатку потрібно спочатку встановити всю необхідну оснастку. Для цього знадобиться віртуальний хостинг або LAMP (Linux, Apache, MySQL, PHP) на який потім необхідно завантажити файли і папки CodeIgniter.

Наступним кроком є створення бази даних. Вона була створена засобами MySQL Workbench. Це інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення й експлуатацію БД в єдине безкоштовне оточення для системи баз даних MySQL.

Потім вона була завантажена на сервер з СУБД, котрий працює під управлінням phpMyAdmin. Він дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Назва створеної бази даних – serv_centр. На рис. 4.3. зображена створена база даних.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	ID	int(255)			Нет	Нет		AUTO_INCREMENT	
2	Name	varchar(255)	latin1_swedish_ci		Нет	Нет			
3	ID_Name	int(11)			Нет	Нет			
4	Phone	varchar(255)	latin1_swedish_ci		Нет	Нет			
5	sms	int(1)			Нет	0			
6	Type	int(1)			Нет	Нет			
7	Failure	text	latin1_swedish_ci		Нет	Нет			
8	Category	varchar(255)	latin1_swedish_ci		Нет	Нет			
9	Model	varchar(255)	latin1_swedish_ci		Нет	Нет			
10	Device	varchar(255)	latin1_swedish_ci		Да	NULL			
11	Price_up	int(11)			Нет	Нет			
12	Price	int(255)			Нет	Нет			
13	dataInput	datetime			Нет	Нет			
14	dataEnd	datetime			Да	NULL			
15	status	int(1)			Нет	3			
16	Comment	varchar(500)	latin1_swedish_ci		Нет	Нет			
17	codice	varchar(20)	latin1_swedish_ci		Нет	Нет			
18	custom_field	longtext	latin1_swedish_ci		Нет	Нет			

Рисунок 4.2 – База даних в phpMyAdmin

Щоб встановити CodeIgniter в корінь сайту потрібно перенести весь зміст попередньо розархівованого фреймворку в кореневу папку веб-серверу.

Після цього можливо перейти за посиланням `site / index.php` та переконатися що все зроблено вірно. Цей процес представлено на рис 4.3.



Рисунок 4.3 – Головна сторінка `site / index.php`.

Для підключення створеної БД до CodeIgniter необхідно за допомогою текстового редактору в файл `application / config / database.php` вписати налаштування підключення до бази даних (рис. 4.4).

```
|
| The $active_group variable lets you choose which connection group to
| make active. By default there is only one group (the "default" group).
|
| The $active_record variables lets you determine whether or not to load
| the active record class
**/

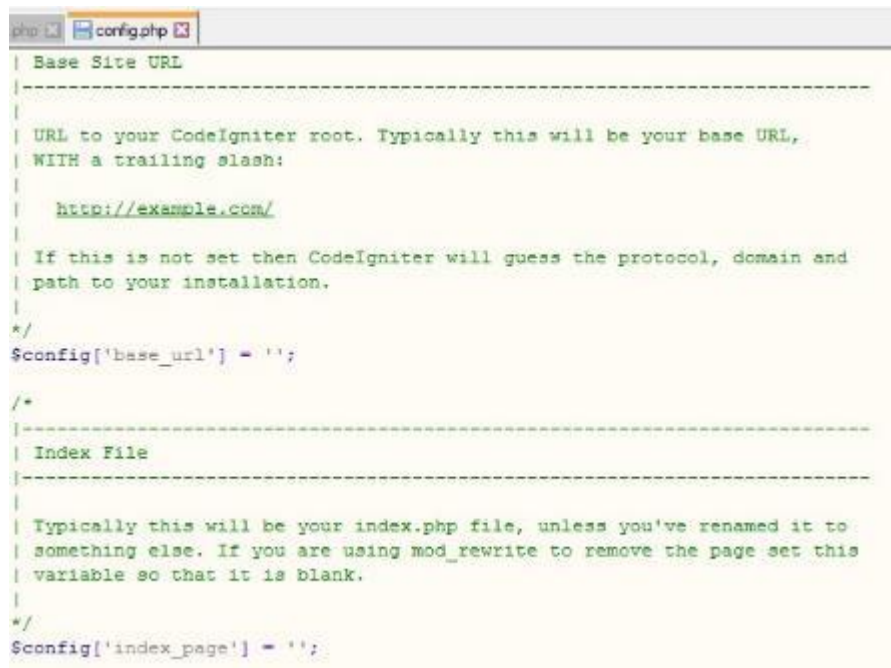
// The following values will probably need to be changed.
$db['default']['username'] = "root";
$db['default']['password'] = "";
$db['default']['database'] = "service";

// The following values can probably stay the same.
$db['default']['hostname'] = "localhost";
$db['default']['dbdriver'] = "mysqli"; //Updated to latest driver.
$db['default']['dbprefix'] = "";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = "utf8";
$db['default']['dbcollat'] = "utf8_general_ci";

$active_group = "default";
$active_record = TRUE;
```

Рисунок 4.4 – Зміст файлу `database.php`

Після цього необхідно за допомогою текстового редактору в файл `application / config / config. php` вказати веб-адресу нашого сайту, як на рис. 4.5.



```
php | config.php |
-----
| Base Site URL
|-----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|   http://example.com/
|
| If this is not set then CodeIgniter will guess the protocol, domain and
| path to your installation.
|
| */
| $config['base_url'] = '';
|
| /*
|-----
| Index File
|-----
|
| Typically this will be your index.php file, unless you've renamed it to
| something else. If you are using mod_rewrite to remove the page set this
| variable so that it is blank.
|
| */
| $config['index_page'] = '';
```

Рисунок 4.5 – Зміст файлу config. php

Клієнтську частину web-сайту було розроблено за допомогою таких інструментів як CSS, HTML, Bootstrap та JavaScript.

Для створення гнучкого інтерфейсу був використаний Bootstrap (фреймворк). Bootstrap – це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері.

Bootstrap має модульну структуру і складається переважно з наборів таблиць стилів LESS, які реалізують різні компоненти цього набору інструментів.

Основні інструменти Bootstrap:

- Сітки (grid) – наперед задані, готові до використання колонки
- Шаблони (template) – фіксовані чи адаптивні шаблони сторінок
- Типографіка (typography) – опис та визначення класів для шрифтів, таких як шрифти для коду, цитат тощо
- Мультимедіа (media) – засоби управління зображеннями та відео

- Таблиці (table) – засоби оформлення таблиць, які зокрема забезпечують сортування
- Форми (form) – класи для оформлення як форм, так і деяких подій
- Навігація (nav, navbar) – класи для оформлення вкладок, сторінок, меню і панелей навігації
- Сповіщення (alert) – класи для оформлення діалогових вікон, підказок і спливаючих вікон
- Іконочний шрифт (icon font) – набір іконок у вигляді шрифту, складається майже з 500 компонентів.

У результаті був реалізований адаптивний інтерфейс під різні розширення екранів чи смартфонів.

4.2 Планування змісту структури робіт IT-проекту (WBS).

Структура декомпозиції робіт (WBS) у проектному менеджменті є орієнтованою на доконане виконання проекту декомпозицією проекту на менші частки. Структура декомпозиції робіт є ключовою часткою робіт по проекту, яка організовує командну роботу по проекту у керовані частини.

WBS є ієрархічною декомпозицією проекту у фази, кінцеві результати та пакети робіт. Вона є ієрархічною структурою, що показує подальший розподіл необхідних для виконання мети зусиль; наприклад, програма, проект чи договір. У проекті чи договорі, розробка WBS відбувається, починаючи з кінцевих цілей та успішного розподілу її у керовані частини, що можуть бути оцінені за критеріями розміру, тривалості та відповідальностей (наприклад, системи, підсистеми, компоненти, задачі, підзадачі та пакети робіт) та включають усі необхідні для досягнення мети проекту кроки.

Система декомпозиції робіт надає загальний каркас для природнього розвитку загального планування та контролю договору і є базисом для розподілу роботи на частини, що можуть бути визначеними, та з яких може бути зроблене Технічне Завдання і установлені звіти по технічним даним, графікам, вартостям, робочим годинам.

Структура декомпозиції робіт дозволяє зібрати до купи підлеглі витрати по задачах, матеріалах тощо на вищій рівень «батьківських» задач, матеріалів тощо. Для кожного елементу структури декомпозиції робіт генерується опис задачі, що має бути виконаною. Ця техніка (іноді називається структурою декомпозиції системи використовується для визначення і налагодження сумарних рамок проекту.

WBS організовується навколо ключових продуктів проекту (чи запланованих результатів), а не необхідних робіт для випуску продукту (заплановані дії). Так як заплановані результати є бажаним завершенням проекту, вони формують відносно стабільний набір категорій, у яких ціни запланованих для їх досягнення необхідних дій можуть бути зібрані до купи. Добре розроблена WBS робить легко досяжним призначення кожної діяльності проекту до виключно однієї термінальної події у WBS. Додатково до її функцій у обліку витрат WBS також допомагає співвіднести вимоги одного рівня системних специфікацій до іншого, наприклад, відповідність матриці вимог перехресних посилань до функціональних вимог на вищій чи нижчій рівні документації.

Розробка WBS зазвичай має відбуватися на початку проекту і перед детальним плануванням проекту і задач. WBS є попереднім етапом, основою для розробки мережових і календарних планів, які потребують повного переліку всіх робіт за проектом, які можна отримати, маючи пакети робіт. WBS наочно демонструє весь обсяг робіт і місце окремих виконавців.

Основні етапи розробки WBS:

- визначення ступеня деталізації проектних робіт (так, щоб вони піддавалися оцінці);
- визначення кількості рівнів (як правило, три-чотири, для сучасних компаній чотири – оптимально);
- розробка структури кожного рівня (формуються горизонтальні рівні);
- підготовка опису елементів WBS (коротка назва кожної складової WBS);

- формування системи кодування (коднуються всі блоки);
- проведення зворотних обчислень (витрати знизу догори за принципом: відділ локалізації – субпідрядник).

WBS може застосовуватися для об'єднання робіт, які необхідно виконати, організаційних структур і відповідальності за роботу з підсистемами планування, оцінки, розподілу витрат і ресурсів, аналізу, контролю та обліку в єдину взаємопов'язану інтегровану систему управління проектом. Будуємо таблицю WBS (рис. Б. 1)

Організаційна структура проекту (OBS). Ця структура стосується тільки внутрішньої організаційної структури проекту і не зачіпає відносин проектних груп або учасників з батьківськими організаціями. Будується OBS аналогічно робочій структурі, а саме:

- на першому рівні відображається організаційна структура як єдиний елемент;
- на другому і нижчих рівнях ділиться нижча частина структури на основні організаційні елементи.

Цей процес повторюється до найбільш низького рівня – базових робочих груп (змішаних цільових або функціональних), а при реалізації малих проектів – до окремих виконавців.

Обсяг робіт для цих найнижчих організаційних рівнів являє собою найбільш низькі елементи WBS, кожен з яких можна планувати і контролювати як окремі одиниці. Саме таке правило діє для створення OBS. Кількість рівнів залежить від розміру проекту.

Об'єднання робочої та організаційної структури дає можливість інтегрувати, планувати і контролювати роботу та порівнювати її виконання по підрозділах і організації взагалі. Кожен менеджер в цій ієрархії має свій набір планів і звітів за своїми сферами відповідальності. Як вже підкреслювалося, розподіл WBS здійснюється до робочого пакету, який виконується окремою групою. OBS, в свою чергу, розбивається до рівня груп, які виконують найнижчий рівень робіт в WBS. Таким чином, роботи

найнижчого рівня WBS притаманні як WBS, так і OBS, тобто це – фундаментальні блоки обох структур. Будуємо таблицю OBS (рис. Б. 2).

Побудова календарного графіка виконання ІТ-проекту. Для того, щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткових мережевих моделей, а також, проекту в цілому з урахуванням вихідних і святкових днів, будують календарний графік робіт.

Він є реальним розподілом робіт по пакету по календарними датами, тобто своєрідним розкладом виконання робіт. Діаграма Ганта є досить зручним для користування. Діаграма Ганта представляє собою відрізки, які розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, як складова плану, розміщуються по вертикалі. Початок, кінець і довжина відрізків на шкалі часу відповідають початку, кінця і тривалості завдання.

За допомогою програми Microsoft Project Professional побудовано діаграму Ганта (рис. Б. 3).

Управління ризиками. До основних ризиків розробки Web-додатку для інформаційної системи підтримки діяльності видавничого центру Державного університету телекомунікацій є:

- неглибоке вивчення предметної області;
- збільшення навантаження під час реалізації проекту;
- зміна цілей у ході реалізації проекту;
- людський фактор;
- відмова обладнання;
- відсутність кваліфікованого програміста;
- зміна строків виконання роботи;
- незнаходження спільної мови між керівником і виконавцем;
- зростання вимог до проекту.

Таблиця 4.1

Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Відсутність досвіду	3	2
2	Збільшення навантаження під час реалізації проекту	3	2
3	Зміна цілей у ході реалізації проекту	4	4
4	Людський фактор	2	3
5	Відмова обладнання	2	4
6	Відсутність кваліфікованого програміста	2	5
7	Зміна строків виконання роботи	2	4
8	Складнощі при впровадженні на місці	3	3
9	Зростання вимог до проекту	3	3

Аналізуючи ризики за ймовірністю їх виникнення, можемо їх розділити на:

- Ігноровані
 - Відсутні
- Незначні
 - Збільшення навантаження під час реалізації проекту
 - Людський фактор
 - Відмова обладнання
 - Зміна строків виконання роботи
- Помірні
 - Відсутність кваліфікованого програміста
 - Зростання вимог до проекту
 - Відсутність досвіду
 - Складнощі при впровадженні на місці
- Істотні
 - Зміна цілей у ході реалізації проекту
- Критичні
 - Відсутні

Таблиця 4.2

Матриця «Ймовірність – Втрати»

Ймовірність	-	5	5	10	15	20	25
	Зміна цілей у ході реалізації проекту	4	4	8	12	16	20
	Збільшення навантаження під час реалізації проекту Зростання вимог до проекту	3	3	6	9	12	15

	Відсутність досвіду Людський фактор Відмова обладнання Відсутність кваліфікованого програміста Зміна строків виконання роботи Складнощі при впровадженні на місці	2	2	4	6	8	10	
	-	1	1	2	3	4	5	
			1	2	3	4	5	
			-	під час при впровадженні на місці	ходи реалізації проекту Відмова обладнання кваліфіковано ГО			
			Втрати					

Класифікація ризиків за рівнем впливу:

- Прийнятні
 - Відсутні
- Виправдані
 - Неглибоке вивчення предметної області
 - Збільшення навантаження під час реалізації проекту
 - Людський фактор
 - Відмова обладнання
 - Відсутність кваліфікованого програміста
 - Зміна строків виконання роботи
 - Не знаходження спільної мови між керівником і виконавцем
 - Зростання вимог до проекту
- Неприпустимі
 - Зміна цілей у ході реалізації проекту.

4.3 Результат реалізації web-додатку

При вводі адреси ІС в браузері користувач потрапляє на сторінку авторизації. На ній можливо ввести логін та пароль для входу до ІС або перевірити поточний стан заявки по коду заявки, яку надає адміністратор-Оператор при прийомі заявки.

The image shows a web application interface with two main sections. The top section, titled "АВТОРИЗАЦИЯ В ИС" (Authorization in IS), features a blue header. Below the header, there are two input fields: "Email address" containing "demo@demo.com" and "Password" containing four dots. To the right of these fields is a green "Login" button with a checkmark icon. The bottom section, titled "СТАТУС" (Status), has a white background. It includes a label "Код заявки" (Application code) and a sub-label "Введите код заявки для получения текущего статуса" (Enter application code to get current status). Below this is a text input field containing the word "code". At the bottom of this section is a blue button with a magnifying glass icon and the text "Узнать статус заявки" (Check application status).

Рисунок 4.7 – Сторінка авторизації

Для перевірки статусу заявки необхідно ввести код у відповідне поле та натиснути «Узнать статус заявки». Даний функціонал представлений на рис 4.8.

АВТОРИЗАЦИЯ В ИС

Email address:

Password:

[Login](#)

СТАТУС

Код заявки
Введите код заявки для получения текущего статуса |

[Узнать статус заявки](#)

Клиент	Татьяна Макарова	ID	5
Состояние	Выдан	Создана	2015-10-28 13:36:55
Описание	Печать фото	Категория	Photo
Качество	Глянec	Пieces	ljkijk
Предоплата	1000	€ Цена	1400
Номер телефона	0503456781	Код заявки	test

Рисунок 4.8 – Відстеження статусу заявки

Після успішної авторизації користувач потрапляє на головну сторінку, де знаходяться елементи навігації та управління ІС. Тут можливо переглянути список заявок та виконати пошук необхідної заявки.

Обравши пункт «Создать заявку» з'являється вікно створення нової заявки на друк. Приклад вікна представлено на рисунку 4.9.

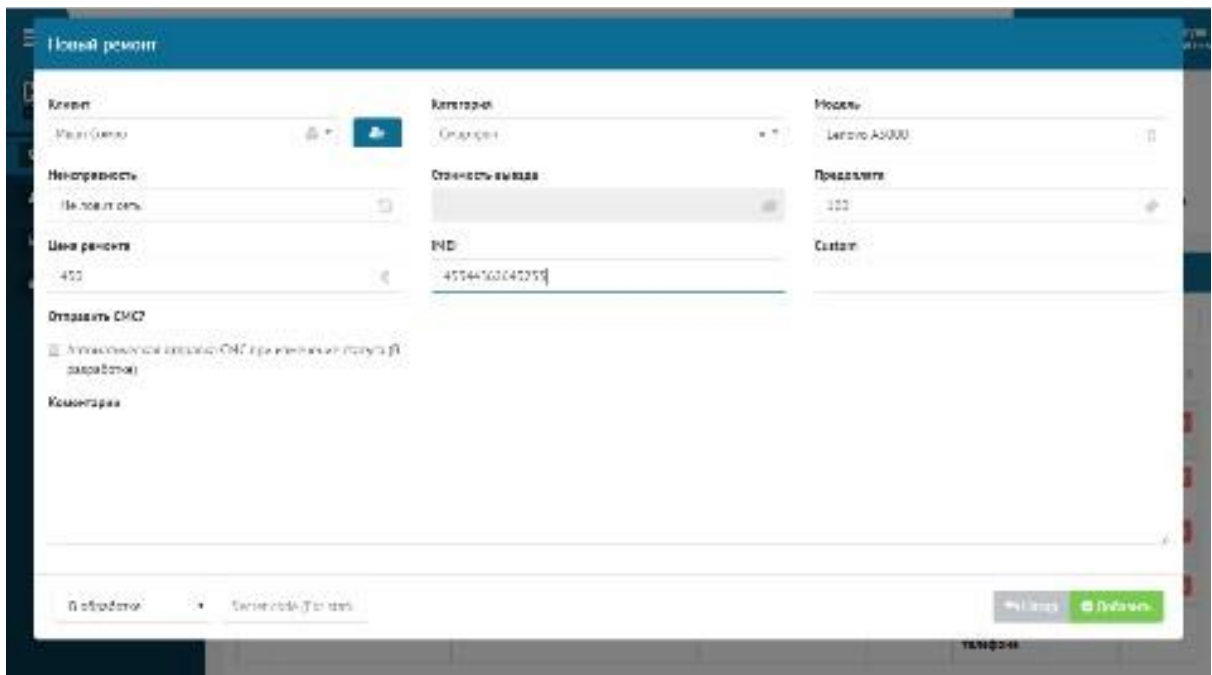


Рисунок 4.9 – «Створити заявку»

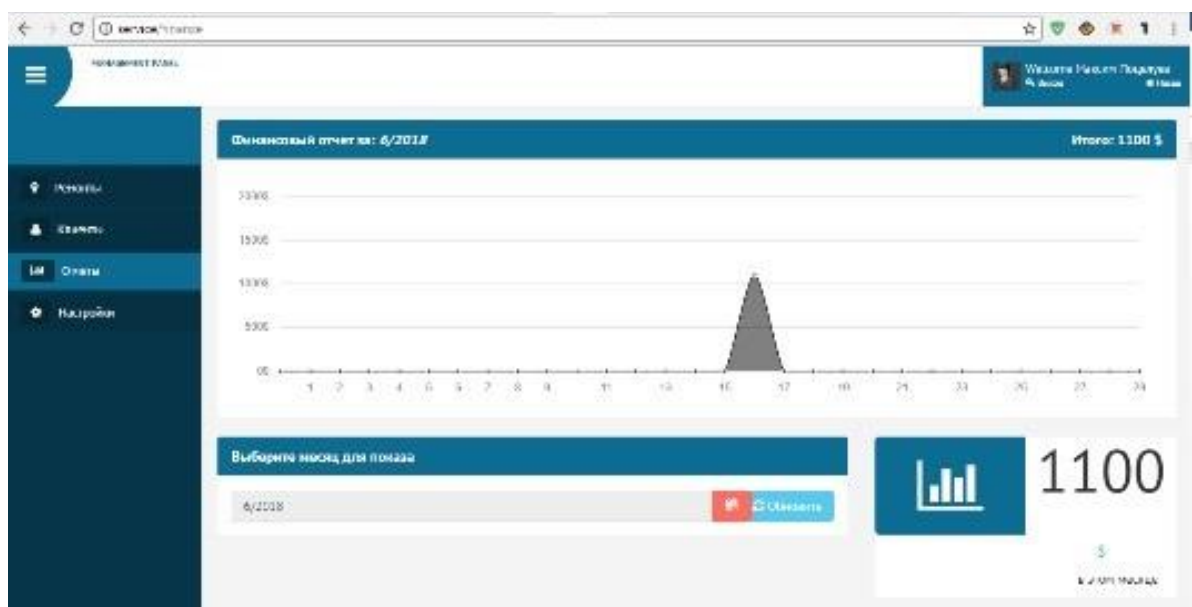


Рисунок 4.10 – Сторінка звітів

При переході до категорії звітів, відображається інформація про зароблені кошти за поточний місяць з розбиттям по датам. Інформація відображається у вигляді графіку. Існує можливість обрати необхідний місяця для аналізу.

При необхідності, більшість налаштувань можливо зробити у розділі «Настройки». Вони розбиті по категоріям, і дозволяють налаштувати дані про ІС, типи заявок та інші. Також можливо підключення зовнішніх сервісів, наприклад для відправки СМС-повідомлень.

ВИСНОВОК

У ході розробки даного роботи було проведено детальний аналіз предметної області. У результаті була виявлена необхідність модернізації процесу супроводу робіт з друку поліграфії у видавничому центрі Державного університету телекомунікацій, що спричинило за собою розробку інформаційної системи на базі web-додатку.

В аналітичній частині дипломної роботи були описані вимоги до розроблюваної системи і обґрунтований вибір програмного і технічного забезпечення для реалізації завдання.

У проектній частині роботи для проекрованої ІС були розроблені: логічна модель реляційної бази даних та побудована діаграма варіантів використання.

На основі розробленої логічної структури і програмного алгоритму, з урахуванням вимог до системи, перерахованих у ТЗ, була реалізована фізична модель БД інформаційної системи.

Були обрані інструментарій та технології, на яких була реалізована програмна частина згідно вимог технічного завдання. Обраний фреймворк CodeIgniter виявився зручним, надійним та простим у використанні. Завдяки йому ІС може бути легко модернізована згідно нових потреб в установі. Bootstrap надав можливість зручно користуватися ІС як на старих моніторах із низькою роздільною здатністю екранів так і на нових FullHD-моніторах чи смартфонах.