

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна інженерія»

Група: 2БКС-26

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**здобувача освіти денної форми навчання
БКС.26.08.000.КРБ**

***МАРЧЕНКА
ОЛЕКСІЯ
ВІТАЛІЙОВИЧА***

**м. Одеса
2022 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **123 «Комп'ютерна інженерія»**

Освітня програма: **«Комп'ютерна інженерія»**

Група: **2БКС-26**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційній роботі бакалавра на тему: _____

«Дослідження та впровадження програмної системи планування ресурсів на підприємстві»

Проектний матеріал складається з пояснювальної записки на _____ сторінках та графічного (презентаційного) матеріалу на _____ аркушах (слайдах).

Виконавець _____ (Марченко О.В.)

Керівник _____ (Краснієнко Н.В.)

Консультанти:

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Завідувачка кафедри _____ (Іванова Л.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « ____ » _____ 2022 р. Протокол ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР _____

“ _____ ” _____ 2022 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

Здобувачеві (здобувачці) освіти Марченку Олексію Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Дослідження та впровадження програмної системи планування

ресурсів на підприємстві

30 грудня

1

306-А2-ОД

затверджена наказом по коледжу від “ _____ ” _____ 202 _____ р. № _____

2. Термін здачі кваліфікаційної роботи _____

3. Вихідні данні до проекту (роботи) _____

Об'єкт аналізу – системи планування ресурсів на підприємстві. Приклади реалізації ПЗ

на підприємстві

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1 Вступ. 1. Технологічний розділ. 3. Охорона праці. Висновки. Перелік
використаних джерел. Додаток

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Презентація (10 слайдів)

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний	Краснієнко Н.В..		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Скорнякова О.В.		

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Робота над Вступом	27.05.2022 р	
	Робота з літературою		
2	Аналіз технологій планування ресурсів		
	на підприємстві	02.06.2022 р	
4	Впровадження програмних засобів		
	для побудови систем ERP	04.06.2022 р.	
5	Виконання розділу «Охорона праці»	08.06.2022 р.	
6	Виконання графічної частини роботи	13.06.2022 р.	
7	Чистове оформлення пояснювальної		
	записки кваліфікаційної роботи	15.06.2022 р.	
8	Підготовка доповіді та презентації до захисту	17.06.2022 р.	
9	Отримання рецензії, відповіді на		
	зауваження рецензента	21.06.2022 р.	
10	Захист роботи	24.06.2022 р.	

Виконавець _____

(підпис)

Керівник _____

(підпис)

ЗМІСТ

ВСТУП.....	7
1 АНАЛІТИЧНИЙ РОЗДІЛ.....	8
1.1 Аналіз технічного завдання.....	8
1.1.1 Найменування програми.....	8
1.1.2 Призначення і область застосування.....	8
1.2 Огляд існуючих рішень ERP – систем.....	10
2 ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	21
2.1 Фаза проектування архітектур баз даних.....	21
2.2 Обґрунтування вибору мови програмування інтерфейсу користувача.....	29
2.3 Підключення бази даних до сайту.....	30
2.4 Архітектура розроблюваної ERP- системи.....	32
3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ.....	37
3.1 Призначення програми.....	37
3.2 Реалізація програмного інтерфейсу.....	37
5 ОХОРОНА ПРАЦІ	55
5.1 Вступ.....	55
5.2 Коротка характеристика та аналіз умов раці на робочому місці програміста.....	55
5.3 Пожежна безпека.....	60
ВИСНОВКИ.....	61
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62

					БКС 26.08.000 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У наше століття обмін інформацією немислимий без сучасних технологій, що підтримують інформаційні процеси: процеси пошуку, збору, передачі, збереження, накопичення, тиражування інформації та процедури доступу до неї.

Впровадження автоматизованих систем управління на підприємстві спрямоване на вирішення найбільш наболілих проблем: організацію обліку, управління виробництвом і т.ін. В даний час у зв'язку з ускладненням процесу прийняття рішень в сучасному бізнесі успіх підприємства безпосередньо залежить від того, як швидко і злагоджено взаємодіють його структури.

З розвитком комп'ютерної техніки з'яляється все більше і більше систем, які потребують від програміста швидкої розробки більш вдосконалених засобів обробки даних, а від користувача – вміння управляти цими даними. Інтегровані рішення за назвою "комплексне планування ресурсів підприємства ERP" покращують якість прийняття рішень, продуктивність і прибутковість. ERP-системи дозволяють ефективно планувати комерційну та виробничу діяльність підприємства. Процес пошуку та корегування інформації передбачає взаємодію у режимі «запит – відповідь» користувача та інформаційної системи дистанційного доступу. Тому у аналітичній частині дипломного проекту я проводжу аналіз існуючих способів побудови сучасних ERP - систем на основі реляційної моделі. У конструкторській частині дипломного проекту я обґрунтовую вибір мови PHP та бази даних MySQL з метою розробки модулів системи з дистанційним доступом. У технологічній частині комплексної роботи я описую розробку модулів системи документообігу складу. У розділі охорона праці описані заходи щодо охорони праці і техніки безпеки користувача ПК.

					БКС 26.08.000 КРБ ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Аналіз технічного завдання

Технічне завдання для програми має містити:

- 1) найменування програми;
- 2) функціональне призначення (область застосування);
- 3) функціональні вимоги до програми;
- 4) вимоги до надійності;
- 5) вимоги до складу і параметрів технічних засобів;
- 6) вимоги до програмної сумісності;
- 7) спеціальні вимоги;
- 8) вимоги до програмної документації;
- 9) технічне завдання на розробку програми.

1.1.1 Найменування програми

Найменування програми: "ERP- система документообігу підприємства"

1.1.2. Призначення і область застосування

Вимоги до програми

- Вимоги до функціональних характеристик

Програма повинна забезпечувати можливість виконання перерахованих нижче функцій:

- 1) користувачі підключаються через інтерфейс;
- 2) можливість пошуку (фільтрації) по базі даних;
- 3) можливість пошуку (фільтрації) по базі даних інформації;
- 4) можливість вносити зміни.

Вимоги до надійності

Вимоги до забезпечення надійного функціонування програми.

					БКС 26.08.001 КРБ ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1) Надійне функціонування програми має бути забезпечене виконанням користувачем ПЗ сукупності організаційно-технічних заходів, перелік яких наведено нижче:

а) організацією безперервного живлення технічних засобів;

б) використанням ліцензійного програмного забезпечення

2) Захист інформації. Випробування програмних засобів на наявність

комп'ютерних вірусів.

3) Відмови через некоректні дії користувачів системи:

Відмови програми внаслідок некоректних дій користувача при взаємодії з програмою через інтерфейс неприпустимі.

Вимоги до складу і параметрів технічних засобів

До складу технічних засобів повинен входити IBM-сумісний персональний комп'ютер (ПЕОМ), що включає в себе:

а) процесор Core i5 та більше;

б) оперативну пам'ять обсягом, 8 Гбайт, не менше;

в) HDD, 512 Гбайт, не менше;

г) операційну систему MS Windows 10.

Дана програма повинна запускатися як безпосередньо з компакт диску, так і при копіюванні на жорсткий диск.

Програма повинна працювати із роздільністю від 1024x768 і вище в повноекранному режимі з автоматичним масштабуванням по висоті і збереженні пропорцій без погіршення якості зображення.

Вимоги до захисту інформації та програм

Особливі вимоги до захисту інформації та програм не пред'являються.

Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

1) технічне завдання;

2) програму і методики випробувань;

					БКС 26.08.001 КРБ ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

3) керівництво оператора.

1.2 Огляд існуючих рішень ERP-систем

Досліджуючи завдання на дипломне проектування я проаналізував сучасні інтегровані інформаційні системи управління, що дають змогу створити єдине інформаційне середовище для автоматизації планування, обліку, контролю, управління й аналізу всіх основних господарських процесів підприємства, що реалізовує концепцію ERP.

ERP - термін, введений дослідницькою фірмою Gartner Group для опису систем управління. Ці системи мають забезпечувати автоматизацію процесів планування, прогнозування й управління фінансами, виробництвом, матеріально-технічним постачанням і збутом, бухгалтерським обліком, а також проектування продукції і розробку технологічних процесів тощо. ERP є світовим стандартом управління, запропонованим Американським співтовариством управління виробництвом і запасами.

Відповідно до словника APICS, термін ERP-система використовується у двох значеннях:

1) це інформаційна система для ідентифікації і планування всіх ресурсів підприємства, що необхідні для здійснення продажу, виробництва, закупівель і обліку в процесі виконання клієнтських замовлень;

2) це методологія ефективного планування й управління всіма ресурсами підприємства, які необхідні для здійснення продажу, виробництва, закупівель і обліку при виконанні замовлень клієнтів у сферах виробництва, дистрибуції і надання послуг.

Абревіатура ERP використовується для позначення комплексних систем управління підприємством. Ключовий термін ERP є підприємство, і тільки потім - планування ресурсів. Дійсне призначення ERP - в інтеграції всіх відділів і функцій компанії в єдиний інформаційний простір завдяки комп'ютерній мережі, що зможе обслужити всі специфічні потреби окремих

					БКС 26.08.001 КРБ ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

підрозділів. Крім того, для ERP-систем, практично, обов'язковою є наявність можливості електронного обміну даними з іншими застосуваннями, а також моделювання ситуацій, пов'язаних з плануванням і прогнозуванням.

Типова ERP-система включає такі підсистеми:

- 1) виробництво;
- 2) постачання і збут;
- 3) управління запасами;
- 4) післяпродажне обслуговування виробленої продукції;
- 5) кадри;
- 6) наукові дослідження і конструкторські розробки;
- 7) фінанси.

Відповідно до сучасних вимог APICS, ERP-система крім ядра, що реалізовує стандарт MRP II, має включати такі модулі:

- 1) управління логістичними ланцюжками SCM;
- 2) планування і складання виробничих графіків APS (Advanced Planning and Scheduling);
- 3) управління відносинами з клієнтами CRM;
- 4) електронної комерції (Electronic Commerce);
- 5) управління даними про виріб - PDM (Product Data Management);
- 6) платформу Business Intelligence, що включає рішення на основі технологій OLAP і DSS (Decision Support Systems);
- 7) модуль, що відповідає за конфігурацію системи (Stand Alone Configuration Engine, SCE);
- 8) планування ресурсів FRP (Finite Resource Planning).

На рисунку 1.1 показано основні функціональні модулі, що входять до складу ERP-систем.

					БКС 26.08.001 КРБ ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

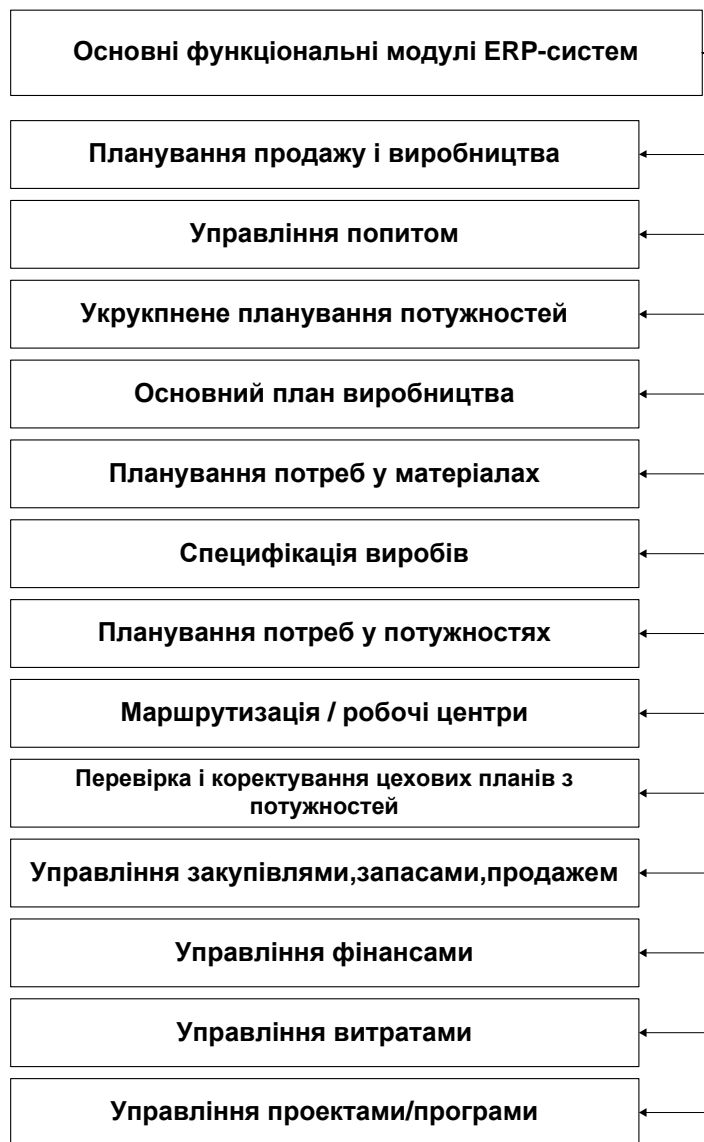


Рисунок 1.1 – Основні функціональні модулі ERP-систем.

Основні функції ERP-систем:

- 1) ведення конструкторських і технологічних специфікацій, що визначають склад виробів, а також матеріальні ресурси й операції, необхідні для його виготовлення;
- 2) формування планів продажу і виробництва;
- 3) планування потреб у матеріалах і комплектуючих, термінів і обсягів поставок для виконання плану виробництва продукції;

- 4) управління запасами і закупівлями: ведення договорів, реалізація централізованих закупівель, забезпечення обліку та оптимізації складських і цехових запасів;
- 5) планування виробничих потужностей від укрупненого планування до використання окремих верстатів і устаткування;
- 6) оперативне управління фінансами, включаючи складання фінансового плану і здійснення контролю за його виконанням, фінансовий та управлінський облік;
- 7) управління проектами, включаючи планування етапів і ресурсів, необхідних для їх реалізації.

Основними відмінностями ERP-систем від MRP II-систем можна вважати:

- 1) більшу кількість типів виробництв і видів діяльності підприємств та організацій;
- 2) планування ресурсів за різними напрямками діяльності;
- 3) можливість управління групою автономно працюючих підприємств, корпоративними структурами;
- 4) більше уваги приділено підсистемам фінансового планування й управління; наявність функцій управління транснаціональними корпораціями, включаючи підтримку кількох часових поясів, мов, валют, систем бухгалтерського обліку;
- 5) більше уваги приділяється створенню інформаційної інфраструктури підприємства, гнучкості, надійності, сумісності з різними програмними платформами;
- 6) інтегрованість із застосуваннями та іншими системами, що використовуються підприємством, такими як системи автоматизованого проектування, автоматизації управління технологічними процесами, електронного документообігу, електронної комерції;

					БКС 26.08.001 КРБ ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

- 7) наявність у системі або інтеграція з програмними засобами підтримки прийняття рішень;
- 8) наявність розвинених засобів конфігурації апаратних і програмних засобів.

Більше уваги надано засобам інтеграції зі сховищами даних. У деяких ERP-системах розроблено засоби інтеграції та адаптації з іншими програмними продуктами і системами (див. рисунок 1.2).

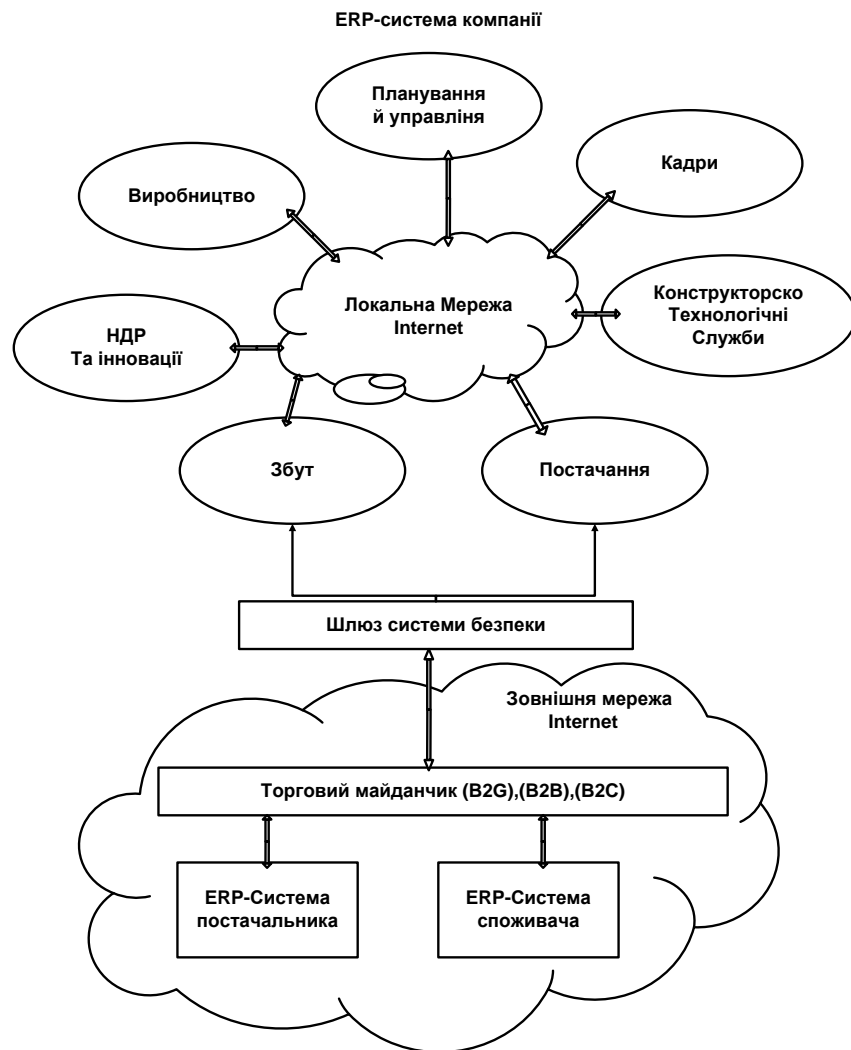


Рисунок 1.2 – Зв'язок різних ERP-систем

Традиційно ERP-системи будуються на базі трирівневої архітектури клієнт - сервер, коли між сервером бази даних і клієнтами знаходиться сервер застосувань, що відповідає за здійснення всієї бізнес-логіки системи. Процедура впровадження ERP-системи включає створення робочої групи

щодо впровадження ІС, виявлення інформаційних потоків на підприємстві, формування нормативно-довідкової бази документів, побудову концептуальної, логічної і фізичної моделей діяльності підприємства, опис оптимізації бізнес-процесів підприємства, виконання пілотного проекту, тобто етап попереднього впровадження. На цій стадії повністю моделюється вся діяльність підприємства, в окремих підрозділах у систему вводяться фактичні дані і послідовно тестуються бізнес-функції шляхом моделювання реальних ситуацій, відпрацьовуються взаємні роботи підрозділів на основі тестових прикладів. Протягом адаптації системи на підприємстві здійснюється налагодження системи відповідно до проекту впровадження і тестування окремих модулів у групі впроваджень. Кінцеві користувачі при цьому навчаються працювати з настроєною системою на своїх робочих місцях.

У процесі дослідної експлуатації системи перевіряють повністю функціональність усіх модулів і вводять систему у промислову експлуатацію.

Висновок. В Україні підприємств, які впровадили ERP-системи, невелика кількість, що пов'язано з відсутністю коштів на такі дорогі проекти, неефективністю впровадження (70% проектів впровадження ERP-систем завершуються невдало), певною складністю ефективної інтеграції ERP-систем з іншими застосуваннями, у тому числі при інтеграції із системами електронного бізнесу (B2B, B2C), обмеженими аналітичними можливостями ERP-систем і недостатньою підтримкою процесів прийняття рішень, прямих комунікацій між інформаційними системами замовника і постачальника.

Переваги впровадження ERP-систем полягають у зниженні собівартості за рахунок підвищення продуктивності; зменшенні часу виходу товарів і послуг на ринок; скороченні кількості виробничих операцій; зменшенні складських запасів; підвищенні якості продуктів; зростанні інформованості керівництва; підвищенні якості прогнозу і планування; формалізації бізнес-процесів компанії; інтеграції всіх підрозділів компанії.

Є кілька шляхів упровадження ERP: власними силами, завдяки адаптації інтегрованих ERP-систем або використання комерційного ПЗ. Відомі ERP-

					БКС 26.08.001 КРБ ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

системи різних компаній, зокрема SAP, Oracle, Ахарта тощо, характеристику яких буде подано нижче.

Характеристика System21

System21 - система, що повністю відповідає поняттю Enterprise Resource Planning (ERP) і Supply Chain Management (SCM). System21 протягом 20 років працює більше ніж у 2000 компаніях світу і лідирує серед ERP-систем на базі платформи iSeries (AS/400) у секторі середнього бізнесу. Її основні модулі: "Головна книга", "Виробництво", "Логістика", "Фінанси", "Дебітори", "Кредитори", "Матеріальний облік", "Постачання", "Збут", "Планування".

Характеристика iRenaissance

Базисом *iRenaissance* компанії ROSS Systems є концепція управління ресурсами підприємства ERP (Enterprise Resource Planning) й управління ланцюжками постачань SCM (Supply Chain Management). Вони дають змогу раціонально управляти підприємством і звести до мінімуму витрати виробництва.

Використання IC *iRenaissance* корпорації ROSS Systems допомагає досягти найбільш високих показників при використанні її на середніх і великих підприємствах з процесним типом виробництва, до якого можна віднести такі галузі промисловості: харчову, хімічну, металургійну, нафтопереробну, целюлозно-паперову, фармацевтичну тощо.

Крім того, *iRenaissance* використовується для автоматизації управління в медичних, адміністративних та освітніх установах, у бібліотеках, у сфері збуту.

Система *iRenaissance* має такі властивості:

- 1) швидке впровадження - власна методологія FrontRun-ner дає змогу скоротити терміни впровадження системи до 3- 6 місяців;
- 2) відносно низька вартість експлуатації;
- 3) легкість адміністрування системи.

Основні завдання, вирішенню яких сприяє *iRenaissance*:

- 1) планування обсягів виробництва;

					БКС 26.08.001 КРБ ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

- 2) облік готової продукції;
- 3) зберігання і пошук управлінської інформації (постачальники, матеріали, устаткування);
- 4) розрахунок витрат, калькуляція собівартості продукції;
- 5) організація і планування виробничого процесу;
- 6) управління фінансами і контроль взаєморозрахунків з постачальниками і споживачами;
- 7) бухгалтерський облік на підприємстві і в його підрозділах;
- 8) аналіз і прогноз продажу;
- 9) контроль за використанням кредитів;
- 10) планування й аналіз бюджету підприємства та його підрозділів;
- 11) розрахунок, аналіз, узагальнення і прогноз фінансових показників підприємства;
- 12) розрахунок грошових потоків на основі бухгалтерських звітів;
- 13) управління перевезеннями та оптимізація транспортних витрат;
- 14) управління експлуатацією устаткування;
- 15) організація електронного документообігу;
- 16) ведення електронного архіву підприємства;
- 17) облік кадрів, управління персоналом;
- 18) облік тру до витрат і розрахунок заробітної плати;
- 19) управління проектами.

Система складається з: виробничого, фінансового модулів, модулів управління персоналом і розрахунку зарплати тощо. Ці модулі можуть використовуватися незалежно один від одного для автоматизації окремих напрямів діяльності підприємства. Впровадження iRenaissance дає змогу оптимізувати роботу підприємства - позбавитися непотрібних і дубльованих операцій, запасів, тобто скоротити витрати і збільшити прибуток.

Систему iRenaissance легко налагодити відповідно до офіційних вимог щодо документообігу і звітності різних країн. Вона має вбудований репозиторій знань (бізнес-процедур), в арсеналі якого зберігаються різні

					БКС 26.08.001 КРБ ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

бізнес-процедур. iRenaissance повністю локалізована, що підтверджується успішно завершеними проектами з упровадження, й успішною експлуатацією системи на російських підприємствах.

Система підтримує кишенькові ПК і пристрої для зчитування штрихкодів. Клієнт-серверна архітектура iRenaissance повністю відповідає таким, що пред'являється до вимог КІС з надійності і безпеки. Крім того, вона є мас штабова пою, тобто їй властиве поступове нарощування конфігурації у міру підключення нових, призначених для користувача, робочих місць. Вся корпоративна інформація зберігається в єдиній базі даних системи, що дає змогу організувати сховище даних та аналізувати накопичену інформацію за допомогою засобів підтримки прийняття рішень, наприклад за допомогою DSSuite, власної системи ROSS Systems.

iRenaissance є відкритою системою. Це означає, що користувач може додавати до неї необхідні функції, використовуючи вбудовані CASE-засоби і мову четвертого покоління (4GL).

Основні модулі iRenaissance:

- 1) iRenaissance Connect - засіб швидкої розробки Web-інструментарію, що забезпечує зв'язок з ERP-платформою підприємства;
- 2) iRenaissance Financial - управління фінансами компанії;
- 3) iRenaissance Distribution - управління продажем;
- 4) iRenaissance Manufacturing - управління виробництвом;
- 5) iRenaissance Maintenance - управління експлуатацією
- 6) майна;
- 7) iRenaissance Human Resources/Payroll - управління персоналом і розрахунок заробітної плати;
- 8) iRenaissance CS Materials Management - управління матеріалами;
- 9) iRenaissance Transportation Manager - управління перевезеннями і транспортними витратами;
- 10) iRenaissance DSSuite - засіб підтримки прийняття рішень;
- 11) iRenaissance SAM - засіб моделювання й опису бізнес-процесів.

					БКС 26.08.001 КРБ ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Система iRenaissance підтримує платформи Windows, Unix, OPENVMS, СУБД Oracle, MS SQL Server, Sybase SQL Server, RDB.

Характеристика Microsoft Dynamics AX

Microsoft Dynamics AX - комплексне ERP-рішення, створене спеціально для середніх і великих компаній, яке допомагає розширити свої можливості та отримати нові конкурентні переваги.

Характеристика Microsoft Axapta

Microsoft Axapta має такі переваги: це система, що дає змогу вести бізнес саме так, як клієнт вважає за потрібне; покращує взаємодію з клієнтами, діловими партнерами і співробітниками; надає потужну функціональність в єдиній інтегрованій системі; надає можливості для швидкого зростання і розвитку бізнесу.

Microsoft Axapta охоплює всі сфери бізнесу, включаючи виробництво і дистрибуцію, управління ланцюжками постачань і проектами, фінансовий менеджмент і засоби бізнес-аналізу, управління відносинами з клієнтами й управління персоналом. Система відповідає всім вимогам міжнародних стандартів бухгалтерського обліку і законодавства, може працювати з багатьма мовами і з різними валютами. В основі Microsoft Business Solutions- Axapta лежать найсучасніші новітні технології управління і високотехнологічні рішення, що дозволяють ефективно управляти підприємством. Система підходить для автоматизації бізнес-процесів у межах управлінського обліку переважно для середніх і великих підприємств різних сфер господарської діяльності.

Axapta - це ERP-система, що працює в середовищі електронного бізнесу. Унікальність системи Axapta полягає в тому, що її сучасна технологія забезпечує єдиний інформаційний простір підприємства, в якому бек-офіс і фронт-офіс працюють як єдине ціле. Axapta пропонує набір можливостей для бізнес-аналізу, що удосконалює процес прийняття рішень і комплексне управління відносинами з клієнтами (CRM).

					БКС 26.08.001 КРБ ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Основними модулями системи Ахарта є фінанси; торгівля і логістика; виробництво; електронна комерція; управління персоналом; проекти; управління відносинами з клієнтами; управління знаннями; управління логістичними ланцюжками. Великий набір функціональних можливостей системи Ахарта дозволяє одержати низку певних переваг: менші витрати на створення і підтримку системи; легкість в оновленні за-стосунків; баланс надмірної інформації; повна інтеграція бізнес-процесів.

Характеристика Microsoft XAL

Microsoft XAL - це інтегрована автоматизована система з універсальним набором функцій для управління фінансово-господарською діяльністю підприємств середнього і великого бізнесу. Система охоплює практично всі функціональні ділянки підприємства і складається з таких модулів: головна книга, дебітори, кредитори, продаж, закупівля, склад, основні засоби, зарплата, управління кадрами, проекти і планування/ управління виробництвом.

Крім стандартних модулів, є різні доповнення, які розширюють можливості наявних модулів системи. У системі є розвинені засоби планування, аналізу діяльності та оперативного управління підприємством.

Microsoft XAL - багатовалютна система з можливістю отримання управлінської, фінансової і податкової звітності та ведення обліку за міжнародними стандартами GAAP, IAS.

Продукт базується на клієнт-серверних технологіях, у нього вбудовано апаратно-незалежну мову четвертого покоління XAL (extended Application Language), що забезпечує повне масштабування і можливість розвитку системи відповідно до вимог бізнесу, що змінюються. Гнучкість, налагоджуваність і масштабування системи забезпечуються можливістю маніпулювання великою кількістю параметрів, а також наявністю вбудованих інструментів налагоджування.

Механізм відстежування призначених для користувача змін зберігає початкові версії змінних елементів і забезпечує безпеку для розробників,

					БКС 26.08.001 КРБ ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

оскільки завжди можна відновити функціональність, що існувала в початковій системі. При зіставленні з Microsoft XAL/ERP-систем, близьких за функціональністю тиражованих програмних рішень для управління бізнесом, таких як SunSystems, Platinum, Exact, Microsoft XAL відрізняється мінімальними інвестиціями, необхідними для установки і супроводу системи.

Оскільки система Microsoft XAL побудована відповідно до правил ведення бізнесу на західних підприємствах, для підприємств вона може слугувати не тільки засобом автоматизації, а й засобом передачі західних стандартів ведення бізнесу. Конструкція Microsoft XAL орієнтована на те, щоб продукт можна було легко і швидко доопрацювати з урахуванням галузевих та індивідуальних вимог конкретного замовника. У Microsoft XAL реалізовано практично всі основні методи, за допомогою яких можна розраховувати обсяги виробництва на основі одержаних замовлень і плану реалізації, визначати параметри завантаження виробничих потужностей, приймати рішення щодо використання наявних ресурсів тощо.

Висновок. У аналітичному розділі докладно розглянуто особливості ERP-систем та способи розробки програмного забезпечення для їх реалізації.

					БКС 26.08.001 КРБ ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

2.1 Фаза проектування архітектури баз даних

Для розгляду способів організації баз даних потрібно визначити декілька понять. Ядро БД відповідає за управління даними в зовнішній пам'яті, управління буферами оперативної пам'яті, управління транзакціями і журналізацію. Відповідно, можна виділити такі компоненти ядра, як менеджер даних, менеджер буферів, менеджер транзакцій.

Ядро БД є основною резидентною частиною СУБД. При використанні архітектури "клієнт-сервер" ядро є основною складовою серверної частини системи.

Основною функцією компілятора мови БД є компіляція операторів мови БД в деяку виконувану програму.

У окремі утиліти БД зазвичай виділяють такі процедури, які дуже не вигідно виконувати з використанням мови БД, наприклад, завантаження і вивантаження БД, збір статистики, глобальна перевірка цілісності БД і т.ін. Утиліти програмуються з використанням інтерфейсу ядра БД, а іноді навіть з проникненням всередину ядра. Загальний склад засобів, необхідних для роботи готового застосування з БД, показаний на рисунку 2.1.

Згідно цій загальній схемі, ми маємо ланцюжок Додаток —> Ядро БД —> база даних.

У структурі додатку є ланцюжок Невізуальні компоненти —> Візуальні компоненти. Невізуальні компоненти надають програмістові деякі функції по управлінню ядром бази даних, а також самими даними.

					БКС 26.08.002 ДП ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.1 – Загальний склад засобів для роботи готового додатку з БД.

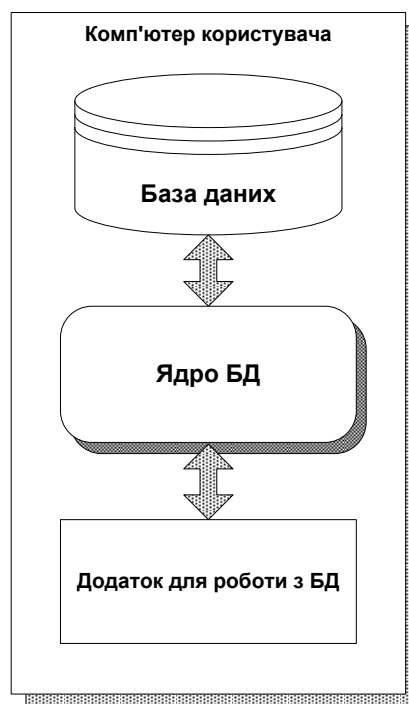


Рисунок 2.2 – Однокористувацька архітектура при роботі з локальними БД.

За допомогою візуальних компонент дані відображаються на екрані (таблиці, списки, випадні списки, графіки).

Місцеположення ядра БД і самих баз даних в цьому ланцюжку не відображені. Місцеположення Ядра БД і баз даних залежить від використовуваної архітектури.

Є чотири різновиди архітектури баз даних:

- а) локальні бази даних;
- б) архітектура "файл-сервер";
- в) архітектура "клієнт-сервер";
- г) багатоланкова архітектура.

Використання тієї або іншої архітектури накладає сильний відбиток на загальну ідеологію роботи додатку, на програмний код в додатку, на склад компонентів для роботи з БД, використовуваних в додатку [4].

Особливістю локальної бази даних і архітектури "файл-сервер" є те, що при роботі з локальними базами даних самі БД розташовані на тому ж комп'ютері, що і додатки, що здійснюють доступ до них. Робота з БД відбувається в одно користувацькому режимі.

Ядро БД знаходиться на комп'ютері користувача. Додаток відповідальний за підтримку цілісності БД і за виконання запитів до БД. Загальна схема однокористувацької архітектури показана на рисунку 2.2. При роботі в архітектурі "файл-сервер" БД і додаток розташовані на файловому сервері мережі (наприклад, Novell NetWare). Можлива багатокористувацька робота з однією і тією ж БД, коли кожен користувач з свого комп'ютера запускає додаток, розташований на мережевому сервері. Тоді на комп'ютері користувача запускається копія додатку. По кожному запиту до БД з додатку дані з таблиць БД переганяються на комп'ютер користувача, незалежно від того, скільки реально потрібно даних для виконання запиту. Після цього виконується запит. Кожен користувач має на своєму комп'ютері локальну копію даних, що час від часу оновлюються з реальної БД, розташованої на мережевому сервері.

При цьому зміни, які кожен користувач вносить до БД, можуть бути до певного моменту невідомі іншим користувачам, що робить актуальним завдання систематичного оновлення даних на комп'ютері користувача з реальної БД.

Іншим актуальним завданням є блокування записів, які змінюються одним з користувачів: це необхідно для того, щоб в цей час інший користувач не вніс змін до тих же даних.

У архітектурі "файл-сервер" вся тяжкість виконання запитів до БД і управління цілісністю БД лягає на додаток користувача. БД на сервері є пасивним джерелом даних. Загальна схема архітектури "файл-сервер" показана на рисунку 2.3. Кардинальних відмінностей з погляду архітектури між однокористувацькою архітектурою і архітектурою "файл-сервер" немає. І

					БКС 26.08.002 ДП ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

в тому і в іншому випадку як СУБД застосовуються так звані "персональні" (або "локальні") СУБД такі як Paradox, dBase і ін. Сама база даних в цьому випадку є набором таблиць, індексних файлів, файлів полів коментарів (мемо-полів), що зберігаються в одному каталозі у диску у вигляді окремих файлів [4].

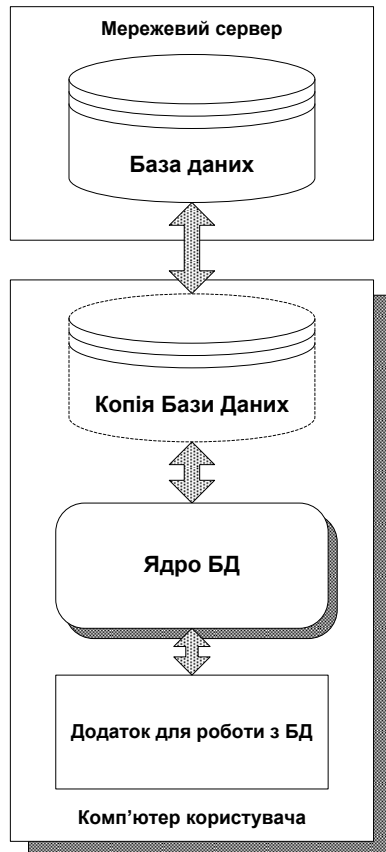


Рисунок 2.3 – Архітектура “Файл-Сервер”

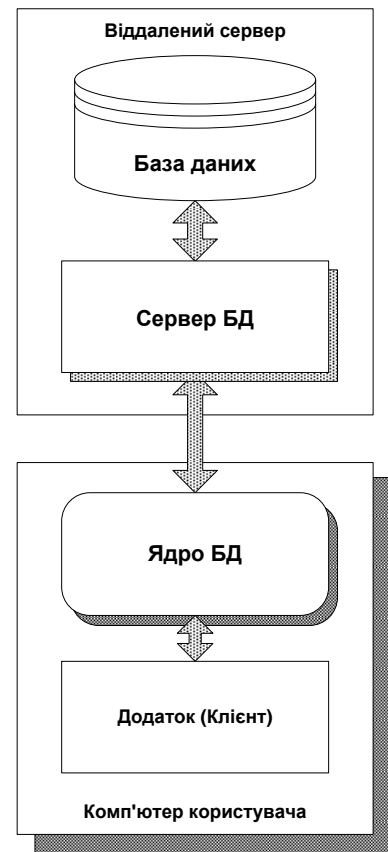


Рисунок 2.4 – Архітектура “Клієнт-Сервер”

В даному дипломному проекті використовуємо архітектуру "клієнт-сервер" яка функціонує за нижче приведеними правилами (див.рис.2.4).

Архітектура клієнт-сервер є одним із [архітектурних шаблонів програмного забезпечення](#) та є домінуючою концепцією у створенні

розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними.

Вона передбачає такі основні компоненти:

- 1) набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- 2) набір клієнтів, які використовують сервіси, що надаються серверами;
- 3) мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Клієнт та сервер. Загальноприйнятим є положення, що клієнти та сервери — це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми — і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Обов'язки та взаємодія. Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- 1) рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- 2) прикладний рівень, який реалізує основну логіку додатку і на якому здійснюється необхідна обробка інформації;
- 3) рівень управління даними, який забезпечує зберігання даних та доступ

					БКС 26.08.002 ДП ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

ДО НИХ.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів — клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- 1) модель тонкого клієнта, в рамках якої вся логіка додатку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- 2) модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача. Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузерами (веб-оглядачами). Найпоширеніші браузери: Google Chrome, Internet Explorer, Firefox, Safari і Opera. Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме — автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляда веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляда сторінок і збору інформації про них.

					БКС 26.08.002 ДП ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

Для запиту до [веб-сервера](#) клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева архітектура. Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка додатку. Програми проміжного рівня можуть функціювати під управлінням спеціальних серверів додатків, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Веб-користувач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних. Найчастіше [веб-сервер](#) і серверні [модулі](#) проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі. На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв [PHP](#), а для управління даними — [СУБД MySQL](#). Таким чином, зв'язок PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі.. Так, для створення серверних [застосунків](#), крім [PHP](#), широко застосовуються

					БКС 26.08.002 ДП ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

[Java](#), [Perl](#), [Python](#). Взагалі, технології створення розподілених, зокрема веб-застосунків, стрімко розвиваються. Слід згадати про технології [EJB \(Enterprise Java Beans\)](#), [CORBA](#), а також про [.NET](#) — порівняно нову ініціативу компанії [Microsoft](#). Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки [XML \(Extensible Markup Language\)](#).

Розроблювана система відноситься саме до інтерактивних веб-сайтів, що розроблено на мову серверних сценаріїв [PHP](#), а для управління даними використовується [СУБД MySQL](#).

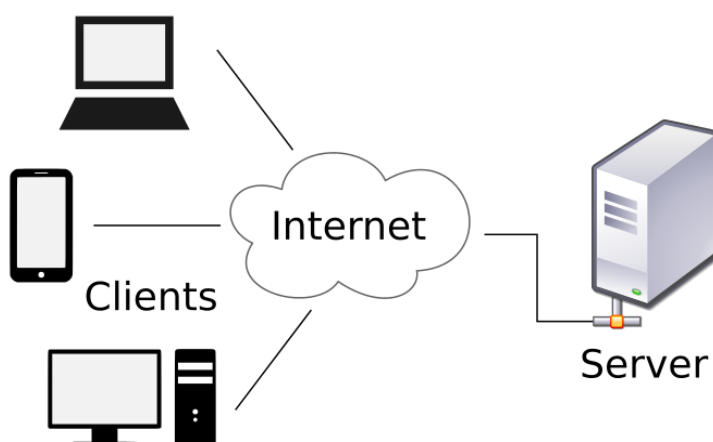


Рисунок 2.5 – Модель взаємодії клієнт-серверної архітектури.

2.2 Обґрунтування вибору мови програмування інтерфейсу користувача

Мова програмування PHP. PHP (англ. PHP: Hypertext Preprocessor - «PHP: препроцесор гіпертексту»; спочатку Personal Home Page Tools [3] - «Інструменти для створення персональних веб-сторінок») - скриптова мова програмування загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов програмування, що застосовуються для створення динамічних веб-сайтів. У 1994 році данський програміст Расмус Лердорф створив набір скриптів на Perl / CGI для висновку і обліку відвідувачів його онлайн-резюме, що обробляє шаблони HTML-

					БКС 26.08.002 ДП ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

документів. Лерддорф назвав набір Personal Home Page (Особиста Домашня Сторінка).

Сьогодні в області програмування для мережі Інтернет PHP - один з популярних скриптових мов (разом з JSP, Perl і мовами, використовуваними в ASP.NET) завдяки своїй простоті, швидкості виконання, багатій функціональності, платформ і розповсюдженню початкових кодів на основі ліцензії PHP.

Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів для розробки веб-додатків. Основні з них:

- 1) автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
- 2) взаємодія з великою кількістю різних систем управління базами даних; автоматизована відправка HTTP-заголовків; робота з HTTP-авторизацією;
- 3) робота з cookies і сесіями; робота з локальними і віддаленими файлами, сокетами;
- 4) обробка файлів, що завантажуються на сервер; робота з XForms.

В даний час PHP використовується сотнями тисяч розробників.

2.3 Підключення бази даних до сайту

База даних сайту - це механізм, що забезпечує зберігання, доступ, первинну обробку й пошук інформації. Будь-який [динамічний сайт](#), що містить структуровані дані, має необхідність в базі даних.

Проектування бази даних для сайту - це один із ключових моментів ранньої стадії розробки. Цей процес необхідно проводити під контролем висококваліфікованого фахівця. Саме на даному етапі закладаються основи, які надалі впливають на швидкість функціонування й складність розробки всього проекту. Для малого та середнього обсягу даних MySQL—оптимальний

					БКС 26.08.002 ДП ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

вибір.

Переваги MySQL:

- 1) найкраща швидкість обробки даних на обсязі до 500000 записів;
- 2) безкоштовні відкриті ліцензії;
- 3) простота використання;
- 4) підтримка більшістю хостингових компаній;
- 5) можливість використання на різних платформах (Unix, Windows, ін.).

Є кілька способів роботи з MySQL в PHP. На даний момент найпопулярніший спосіб - це об'єктів-орієнтована модель спілкування з MySQL. Саме з використанням цього способу ми і будемо працювати з базами даних у розроблюваному проекті автоматизованої системи.

Алгоритм підключення БД наступний:

- 1) Підключення.
- 2) Відправлення запитів і отримання результату.
- 3) Закриття з'єднання.

Підключитися до бази даних через PHP можна наступним чином:

```
<? Php $ mysqli = new mysqli ( 'localhost', 'Admin', 'pass', 'mybase');  
?>
```

Тут ми створюємо екземпляр об'єкта MySQLI, передаючи конструктору наступні параметри:

- 1) Ім'я хоста, на якому працює MySQL.
- 2) Ім'я користувача.
- 3) Пароль.
- 4) Ім'я бази даних, з якою ми хочемо працювати.

Якщо які-небудь дані будуть невірні, то, відповідно, конструктор поверне помилку, і підключення не буде.

Але, якщо буде помилка підключення, то виконання скрипта не буде зупинений. В результаті, він почне далі виконувати наш код. У більшості випадків, при помилці підключення потрібно припинити виконання скрипта,

					БКС 26.08.002 ДП ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

тому пишуть так:

```
<? Php $ mysqli = @new mysqli ( 'localhost', 'Admin', 'pass', 'mybase'); if  
(mysqli_connect_error ())  
{Echo "Підключення неможливо:" .mysqli_connect_error ();}>
```

Тут ми перевіряємо: якщо були якісь помилки при підключенні, то виводимо їх і закінчуємо виконання скрипта.

Далі виконаємо закриття підключення. У прикладі нижче ми підключаємося до бази даних, а після перевірки на успішність підключення, закриваємо цю сполуку:

```
<? Php $ mysqli = @new mysqli ( 'localhost', 'Admin', 'pass', 'mybase'); if  
(mysqli_connect_errno ())  
{Echo "Підключення неможливо:" .mysqli_connect_error (); } $ mysqli-  
> Close ();?>
```

2.4 Архітектура розроблюваної ERP системи

Всі ERM-системи, незалежно від того, хто їх розробник, об'єднує загальна архітектура, яку можна описати таким чином:

Платформа. Базові можливості і середовище для роботи модулів і компонентів. У код платформи зміни може вносити тільки розробник. Користувачі і фахівці з впровадження не мають доступу до цього програмного коду.

До складу платформи входять:

Ядро. Програмне середовище, в якій буде проводитися робота, для якої можна писати якісь надбудови і компоненти.

Базовий функціонал. Перелік довідників і функцій, без яких не може працювати жодна компанія. Це довідник користувачів з правами доступу, довідник клієнтів, довідник товарів / послуг і т.ін. Цей функціонал вбудований в платформу, на відміну від модулів він не може відключатися.

Управління даними. База даних, в тому числі, зберігання і методи обробки (інтерпретації) даних. У цю категорію входять сховище даних на

					БКС 26.08.002 ДП ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

сервері, програмне забезпечення для роботи з базами даних (SQL або будь-яка альтернатива), інструменти для інтерпретації та обробки даних і відправки їх в програмні модулі.

Модулі. Компоненти, які підключаються до платформи в міру необхідності. Всі вони працюють з єдиною базою даних і застосовують базовий функціонал (в міру необхідності). В іншому модулі працюють незалежно один від одного, можуть «бесшовно» підключатися і без проблем відключатися, якщо потреба в них зникла. Така модульна структура - важлива відмінна риса ERP-систем. Модулі діляться, в свою чергу, на кілька типів:

Модулі внутрішнього використання. Цей рівень - Plug-in, які використовуються співробітниками компанії. Це управління складом, виробництво, бухгалтерія, CRM та ін. Модулі можна підключати, відключати, налаштовувати силами фахівців з впровадження. У стандартний набір зазвичай входять - MRP, HR, CRM, Управління постачанням і закупівлями.

Модулі роботи з зовнішніми користувачами. Цей шар містить в собі модулі, необхідні для взаємодії із зовнішніми користувачами, потенційними і реальними клієнтами компанії, партнерами, користувачами продукції, постачальниками і покупцями. Це може бути інтернет-магазин, особисті кабінети для постачальників і покупців на корпоративному сайті і тому подібні рішення.

Деякі ERP-системи містять в собі готові CMS-системи для створення інтернет-магазину або корпоративного сайту з нуля, інші пропонують тільки окремі інструменти «надбудови» до сайту і / або клієнтські програми (для мобільних і планшетів).

Конектори - готові рішення для зв'язку зі сторонніми додатками. Найчастіше використовують API з ядра платформи.

Дозволяють інтегрувати телефонію, налаштувати обмін даними з сайтом або будь-якими програмними продуктами і системами. Конектори призначені тільки для обміну даними і зазвичай використовуються для обміну

					БКС 26.08.002 ДП ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

даними з EDI, CMS, CAD, BI, OLAP і ін. Тобто з тими системами які не входять до ERP, але використовуються в компанії.

У деяких систем немає яскраво вираженої модульності, вони все вже вбудовані в програму, але використовувати їх можна окремо один від одного в міру необхідності.

Інші називають модулями підсистеми, що відключаються. Частина ERP-систем виділяють все модулі дійсно в окремі продукти. І пропонують розроблення ядра, а до нього - перелік модулів на вибір з можливістю розширювання системи.

Переваги модульної структури ERP. Важливою перевагою ERP систем є можливість підключати і застосовувати будь-який з модулів (внутрішній або зовнішній) в стислі терміни.

Причому, можливості, які таким чином підключаються до ERP, додаються в систему абсолютно «бесшовно». В цьому полягає важлива відмінність ERP від інтеграції декількох програмних продуктів між собою або від системи, яка виросла зі спеціалізованої за рахунок численних доробок і надбудов. Кожен з модулів ERP-системи працює незалежно від інших, він може бути підключений або відключений в будь-який момент часу, може просто не використовуватися, при цьому інші модулі зможуть продовжувати роботу. А для підключення того чи іншого модуля немає необхідності вносити зміни в програмний код ядра, інших модулів.

Розроблювана система функціонує на базі нижче перерахованих принципів:

- 1) Принцип єдиної бази даних: контроль, управління, точність і оперативність.
- 2) Гнучкість в роботі компанії з урахуванням змін ринку.

На рисунку 2.6 представлено структурну схему розроблюваної системи.

					БКС 26.08.002 ДП ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

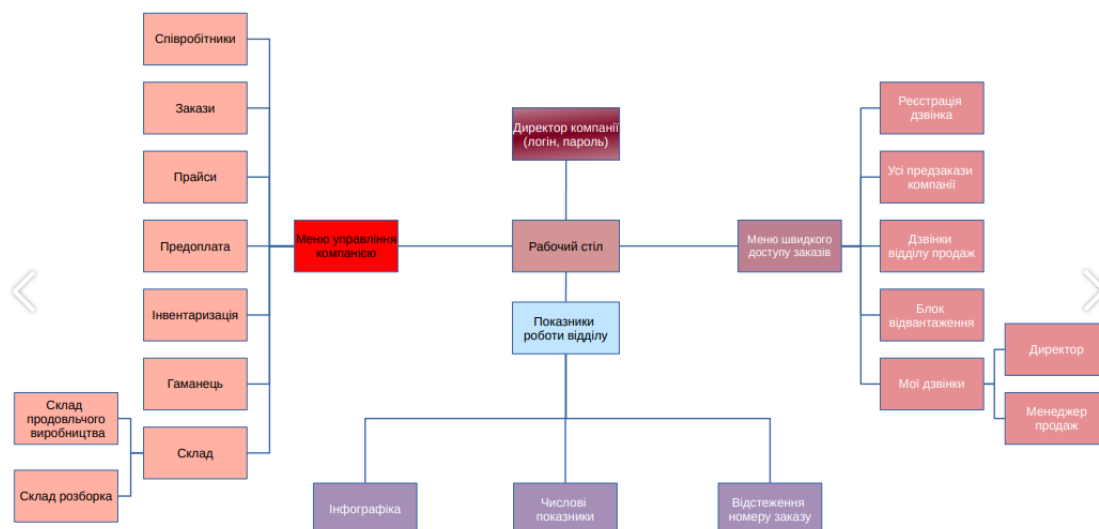


Рисунок 2.6 – Структурна схема системи

Результати від впровадження ERP системи

ERP - це, перш за все, можливість об'єднати всі бізнес-процеси в одній потужній і зручній системі.

- 1) Виділемо найбільш значущі:
 - Доступність даних. Один раз внесені дані стають доступні в рамках цілої системи, при цьому не потрібні звірки, додаткові узгодження і перевірки.
- 2) Узгодженість даних. Застосування загальної бази даних дозволяє уникнути етапів звірки та погодження даних.
- 3) Контроль роботи співробітників. У разі, коли один з підрозділів вносить дані, наприклад, про витрату (переміщенні) товару, то інший підрозділ відразу ж отримує про це відомості, і після реального отримання перерахованих позицій ставить його собі на прихід. Розбіжність у цифрах в цьому випадку практично неможливо, загальна база даних виключає ймовірність багатьох зловживань, а директор в режимі реального часу може виявити будь-які розбіжності і їх причини.
- 4) Значне зниження числа помилок, пов'язаних з людським фактором. Єдина система не дозволить списати товар не з того складу, так як загальна база даних вкаже, що товару там немає. Інформація про оплати

					БКС 26.08.002 ДП ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

буде передаватися в бухгалтерію на основі даних з банку, що також виключає помилки.

- 5) Готовий набір об'єднаних між собою інструментів. Наприклад, якщо відділ продажів створює рахунок-фактуру, то вона є підставою для автоматичного створення бухгалтерських документів, а після оплати - видаткових документів зі складу.

Висновок. У конструкторському розділі обґрунтовано та розроблено модульну структуру ERP системи.

					БКС 26.08.002 ДП ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

3.1 Призначення програми

Результат виконаної роботи - динамічний сайт, з підключеною до нього базою даних, можливістю управління користувачами, авторизацією і реєстрацією користувачів. А також корегування та зчитування інформації з бази даних.

Так як програма використовується багатьма посадовими особами компанії, для кожного співробітника була створена версія програми з різними рівнями доступу та функціями. Наприклад, менеджер з продажу не має можливості нараховувати зарплати, на відміну від директора, а бухгалтер може тільки переглядати деякі документи, що створені директором без права їх змінювати.

Для кожної людини, що користується програмою, є свій логін і пароль, відомий тільки цій людині. При переході на сайт співробітник потрапляє на сторінку авторизації, з якої і починається його взаємодія з програмою. У системному коді програми є кілька груп користувачів: менеджер з продажу, бухгалтер, програміст, директор, керуючий цехів, керуючий складу, робочий. Для кожної групи свої права (принцип ООП - успадкування). Кожна сторінка сайту розділена на блоки, що не змінюються в залежності від посади співробітника.

3.2 Реалізація програмного інтерфейсу

Інтерфейс програми досить простий в освоєнні. Не дивлячись на складність, що здається, через декілька хвилин роботи з програмою, призначення всіх елементів управління стає ясным. Програма реалізована за допомогою модуля головного вікна програми.

					БКС 26.08.004 КРБ ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Блок-схема алгоритму початку роботи з програмою представлена на рисунку 3.1, вид головного меню на етапі авторизації представлений на рисунку 3.2.

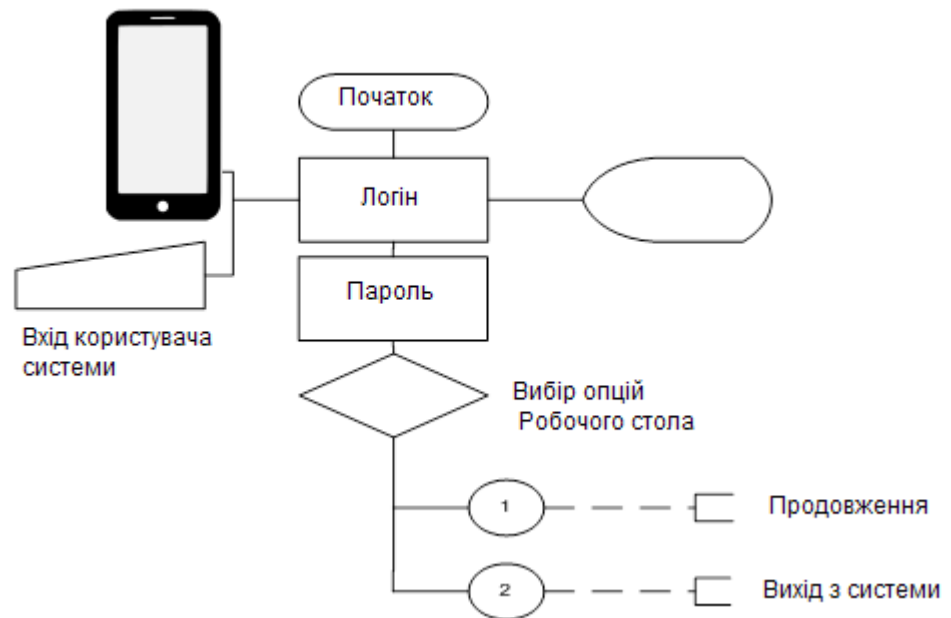


Рисунок 3.1 – Блок-схема алгоритму початку роботи з програмою

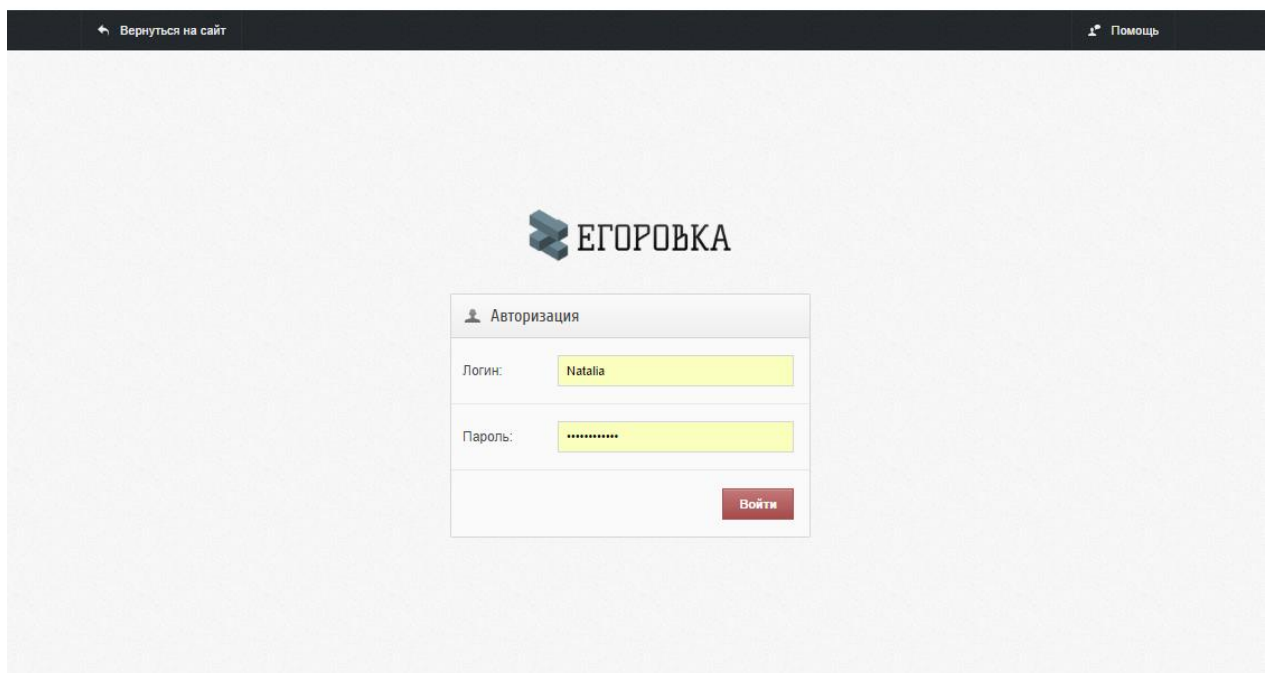


Рисунок 3.2 – Вид головного меню програми (етап авторизації)

Нижче приведено код і лістинг програми головного меню на мові PHP.

```
<?php
    include 'include/template/header.php';
    if($_SESSION['id']== "")
        {
?>
<script type="text/javascript" language="javascript">
function auth() {
var msg = $('#auth').serialize();
$.ajax({
type: 'POST',
url: '/function/auth.php',
data: msg,
success: function(data) {
$('#result_auth').html(data);
},
error: function(xhr, str){
alert('Возникла ошибка: ' + xhr.responseCode);
}
});
}
</script>
</head>
<body>
<div id="topNav">
    <div class="fixed">
        <div class="wrapper">
            <div class="backTo"><a href="#" title=""><span>Вернуться на
сайт</span></a></div>
            <div class="userNav">
                <ul>
                    <li><a href="#" title=""><span>Помощь</span></a></li>
                </ul>
            </div>
        </div>
    </div>
</div>
<div class="loginWrapper">
    <div class="loginLogo"></div>
    <div class="loginPanel">
        <div class="head"><h5 class="iUser">Авторизация</h5></div>
        <form method="POST" id="auth" action="javascript:void(null);"
onsubmit="auth()" class="mainForm">
            <fieldset>
                <div class="loginRow noborder">
                    <label for="req1">Логин:</label>
```

```

        <div class="loginInput"><input type="text" name="login"
class="validate[required]" id="req1" /></div>
    </div>
    <div class="loginRow">
        <label for="req2">Пароль:</label>
        <div class="loginInput"><input type="password" name="password"
class="validate[required]" id="req2" /></div>
    </div>
    <div class="loginRow">
        <div class="submitForm"><input type="submit" value="Войти"
class="redBtn" /></div>
    </div>

    <div id="result_auth"></div>

</fieldset>
</form>
</div>
</div>
<?php
include 'include/template/footer.php';
}
else
{
    echo '<script language="JavaScript"> window.location.href = "/home.php"</script>';
}
}

```

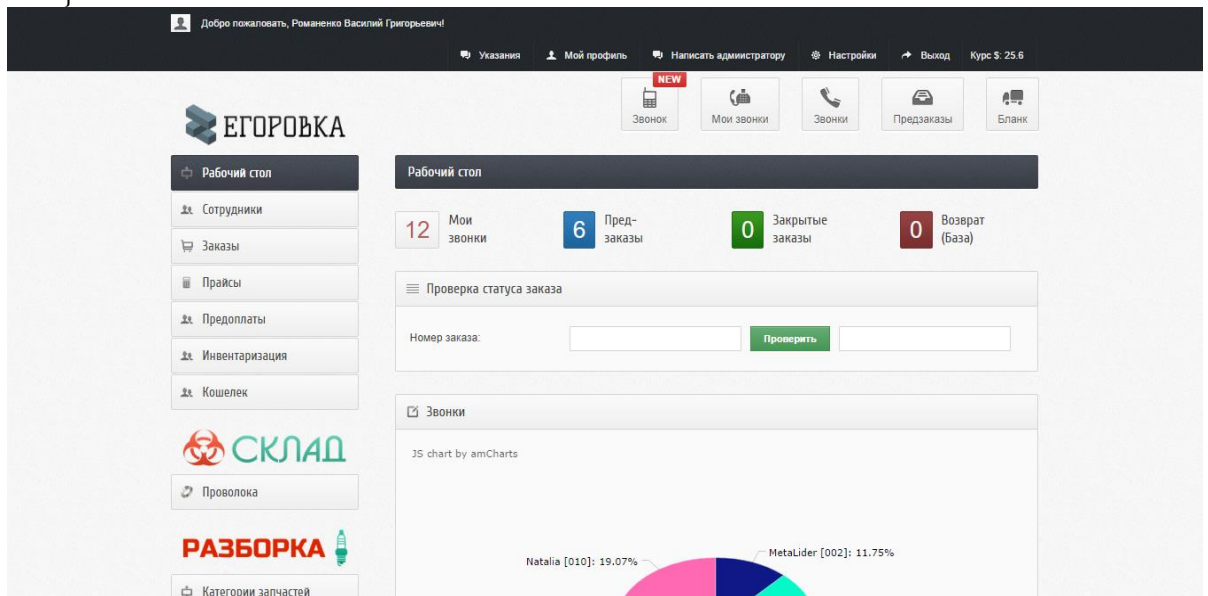


Рисунок 3.3 – Програма розробки Робочого стола користувача системи

Нижче приведено код і лістинг програми Робочого стола на мові PHP.

```

<?php
include 'include/template/header.php';
if($_SESSION['id']!="")
{
    if($_SESSION['access']=="4"){echo      '<script      language="JavaScript">
window.location.href = "/preorder_list.php"</script>';}
?>

```

									Арк.
									40
Змн.	Арк.	№ докум.	Підпис	Дата	БКС 26.08.004 КРБ ПЗ				

```

<script src="https://www.amcharts.com/lib/3/amcharts.js"></script>
<script src="https://www.amcharts.com/lib/3/pie.js"></script>
<script src="https://www.amcharts.com/lib/3/serial.js"></script>
<style>
  #chartdiv {
    width      : 100%;
    height     : 435px;
    font-size  : 11px;
  }
  #chartdiv_status {
    width      : 100%;
    height     : 435px;
    font-size  : 11px;
  }
  #chartdiv_preorder {
    width      : 100%;
    height     : 435px;
    font-size  : 11px;
  }
  #chartdiv_close {
    width      : 100%;
    height     : 435px;
    font-size  : 11px;
  }
  .amcharts-export-menu-top-right {
    top: 10px;
    right: 0;
  }
</style>
<script type="text/javascript">
  var chart = AmCharts.makeChart("chartdiv_status", {
    "type": "serial",
    "theme": "light",
    "marginRight": 70,
    <?php
    $query_call = "SELECT COUNT(*) FROM `scroll_call` WHERE `status`='2'
OR `status`='0' OR `status`='5'";
    $res_call = mysql_query($query_call) or die(mysql_error());
    $result_call = mysql_fetch_array($res_call);
    $query_preorder = "SELECT COUNT(*) FROM `scroll_call` WHERE
`status`='8'";
    $res_preorder = mysql_query($query_preorder) or die(mysql_error());
    $result_preorder = mysql_fetch_array($res_preorder);
    $query_made = "SELECT COUNT(*) FROM `scroll_call` WHERE
`status`='9'";
    $res_made = mysql_query($query_made) or die(mysql_error());
    $result_made = mysql_fetch_array($res_made);
    $query_deliv = "SELECT COUNT(*) FROM `scroll_call` WHERE
`status`='7'";
    $res_deliv = mysql_query($query_deliv) or die(mysql_error());
    $result_deliv = mysql_fetch_array($res_deliv);

```

						Арк.
					БКС 26.08.004 КРБ ПЗ	41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$query_delivery = "SELECT COUNT(*) FROM `scroll_call` WHERE
`status`='10'";
$res_delivery = mysql_query($query_delivery) or die(mysql_error());
$result_delivery = mysql_fetch_array($res_delivery);
$query_designed = "SELECT COUNT(*) FROM `scroll_call` WHERE
`status`='13'";
$res_designed = mysql_query($query_designed) or die(mysql_error());
$result_designed = mysql_fetch_array($res_designed);
$query_close = "SELECT COUNT(*) FROM `scroll_call` WHERE
`status`='11'";
$res_close = mysql_query($query_close) or die(mysql_error());
$result_close = mysql_fetch_array($res_close);
$query_back = "SELECT COUNT(*) FROM `scroll_call` WHERE
`status`='12'";
$res_back = mysql_query($query_back) or die(mysql_error());
$result_back = mysql_fetch_array($res_back);
$query_del = "SELECT COUNT(*) FROM `scroll_call` WHERE `status`='99'";
$res_del = mysql_query($query_del) or die(mysql_error());
$result_del = mysql_fetch_array($res_del);
?>
"dataProvider": [{
  "country": "Звонок",
  "visits": <?php echo $result_call[0];?>,
  "color": "#FF0F00"
}, {
  "country": "Предзаказ",
  "visits": <?php echo $result_preorder[0];?>,
  "color": "#FF6600"
}, {
  "country": "Готов",
  "visits": <?php echo $result_made[0];?>,
  "color": "#FF9E01"
}, {
  "country": "Отгрузка",
  "visits": <?php echo $result_deliv[0];?>,
  "color": "#FCD202"
}, {
  "country": "Отгружен",
  "visits": <?php echo $result_delivery[0];?>,
  "color": "#F8FF01"
}, {
  "country": "Рассчитан",
  "visits": <?php echo $result_designed[0];?>,
  "color": "#009933"
}, {
  "country": "Закрыт",
  "visits": <?php echo $result_close[0];?>,
  "color": "#B0DE09"
}, {
  "country": "Возврат",
  "visits": <?php echo $result_back[0];?>,
  "color": "#04D215"
}

```

					БКС 26.08.004 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

```

    }, {
      "country": "Удален",
      "visits": <?php echo $result_del[0];?>,
      "color": "#CD0D74"
    }
  ]],
  "valueAxes": [{
    "axisAlpha": 0,
    "position": "left",
    "title": "Количество заказов"
  }],
  "startDuration": 1,
  "graphs": [{
    "balloonText": "<b>[[category]]: [[value]]</b>",
    "fillColorsField": "color",
    "fillAlphas": 0.9,
    "lineAlpha": 0.2,
    "type": "column",
    "valueField": "visits"
  }],
  "chartCursor": {
    "categoryBalloonEnabled": false,
    "cursorAlpha": 0,
    "zoomable": false
  },
  "categoryField": "country",
  "categoryAxis": {
    "gridPosition": "start",
    "labelRotation": 45
  },
  "export": {
    "enabled": true
  }
}
});
</script>
<script type="text/javascript">
var chart = AmCharts.makeChart( "chartdiv", {
  "type": "pie",
  "theme": "light",
  "dataProvider": [
    <?php
    $query = "SELECT * FROM `users` WHERE `access`='2'";
    $res = mysql_query($query) or die(mysql_error());
    while ($result = mysql_fetch_array($res))
    {
      $query_count = "SELECT COUNT(*) FROM `scroll_call`
WHERE `status` IN (0,2,5) AND `manager`='".$result['id']."'";
      $res_count = mysql_query($query_count) or die(mysql_error());
      $result_count = mysql_fetch_array($res_count);
      $number=str_pad($result['id'], 3, '0', STR_PAD_LEFT);
      echo "{";
      echo "country": "'".$result['login']."' ['.$number.'],';";
      echo "value": ' '.$result_count[0].';";
    }
  ]
}
);

```

					БКС 26.08.004 КРБ ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        echo "color" : " ".$result['color']."";
        echo '},';
    }
    ?>
],
"valueField": "value",
"titleField": "country",
"colorField": "color",
"outlineAlpha": 0.4,
"depth3D": 20,
"balloonText": "[[title]]<br><span style='font-size:14px'><b>[[value]]</b>
([[percents]]%)</span>",
"angle": 30,
"export": {
    "enabled": true
}
} );
var chart = AmCharts.makeChart( "chartdiv_preorder", {
    "type": "pie",
    "theme": "light",
    "dataProvider": [
        <?php
        $query = "SELECT * FROM `users` WHERE `access`='2'";
        $res = mysql_query($query) or die(mysql_error());
        while ($result = mysql_fetch_array($res))
        {
            $query_count = "SELECT COUNT(*) FROM `scroll_call`
WHERE `status`='8' AND `manager`='". $result['id']. "'";
            $res_count = mysql_query($query_count) or die(mysql_error());
            $result_count = mysql_fetch_array($res_count);
            $number=str_pad($result['id'], 3, '0', STR_PAD_LEFT);
            echo "{ ";
            echo "country": " ".$result['login']. " [ ".$number. " ] ";
            echo "visits": " ".$result_count[0]. " ";
            echo "color" : " ".$result['color']. "" ;
            echo '},';
        }
        ?>
],
"valueField": "visits",
"titleField": "country",
"colorField": "color",
"startEffect": "elastic",
"startDuration": 2,
"labelRadius": 15,
"innerRadius": "50%",
"depth3D": 10,
"balloonText": "[[title]]<br><span style='font-size:14px'><b>[[value]]</b>
([[percents]]%)</span>",
"angle": 15,
"export": {
    "enabled": true
}
} );

```

						Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
  });
  jQuery( '.chart-input' ).off().on( 'input change', function() {
    var property = jQuery( this ).data( 'property' );
    var target = chart;
    var value = Number( this.value );
    chart.startDuration = 0;
    if ( property == 'innerRadius' ) {
      value += "%";
    }
    target[ property ] = value;
    chart.validateNow();
  });
</script>

```

```

<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = google.visualization.arrayToDataTable([
    ]);
    var options = {
      pieHole: 0.5,
      pieSliceTextStyle: {
        color: 'black',
      },
      legend: 'none'
    };
    var chart = new
google.visualization.PieChart(document.getElementById('donut_single'));
    chart.draw(data, options);
  }
</script>
</head>
<body>
<?php
include 'include/template/top_band.php';
include 'include/template/top_menu.php';
?>
<!-- Content wrapper -->
<div class="wrapper">

<?php
include 'include/template/menu.php';
?>
<!-- Content -->
<div class="content">
<div class="title"><h5>Рабочий стол</h5></div>
<div class="stats">

```

					БКС 26.08.004 КРБ ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <ul>
            <li><a href="/my_call.php" class="count grey" title=""><?php
return_status_count(0,$_SESSION['id']);?></a><span>Мои<br>звонки</span></li>
            <li><a href="/preorder_list.php" class="count blue" title=""><?php
return_status_count(8,$_SESSION['id']);?></a><span>Пред-<br>заказы</span></li>
            <li><a href="/closed_orders.php" class="count green" title=""><?php
return_status_count(11,$_SESSION['id']);?></a><span>Закрытые<br>заказы</span></li>
        >
            <li class="last"><a href="/return_orders_base_themplate.php"
class="count red" title=""><?php
return_status_count(14,$_SESSION['id']);?></a><span>Возврат<br>(База)</span></li>
        </ul>
    </div>
    <form method="POST" id="check_status" action="javascript:void(null);"
onsubmit="check_status()" class="mainForm">
        <fieldset>
            <div class="widget first">
                <div class="head"><h5 class="iList">Проверка статуса
заказа</h5></div>
                <div class="rowElem noborder"><label>Номер заказа:</label><div
class="formRight"><input type="text" style="width: 38%;" name="number"/><input
type="submit" style="margin-left: 10px;" value="Проверить" class="greenBtn" /><input
type="text" disabled style="width: 38%;margin-left: 10px;text-align: center;"
id="status"/></div></div>
                <div id="result"></div>
            </div>
        </fieldset>
    </form>
    <div class="widget first">
        <div class="head"><h5 class="iCreate">Звонки</h5></div>
        <div class="body">
            <center><div id="chartdiv"></div></center>
        </div>
    </div>
    <div class="widget">
        <div class="head"><h5 class="iCreate">Предзаказы</h5></div>
        <div class="body">
            <center><div id="chartdiv_preorder"></div></center>
        </div>
    </div>
    <div class="widget">
        <div class="head"><h5 class="iCreate">Статусы</h5></div>
        <div class="body">
            <center><div id="chartdiv_status"></div></center>
        </div>
    </div>
</div>
<?php
include 'include/template/footer.php';
}
else

```

					БКС 26.08.004 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

```

{
echo '<script language="JavaScript"> window.location.href = "/index.php"</script>';
}
?>

```

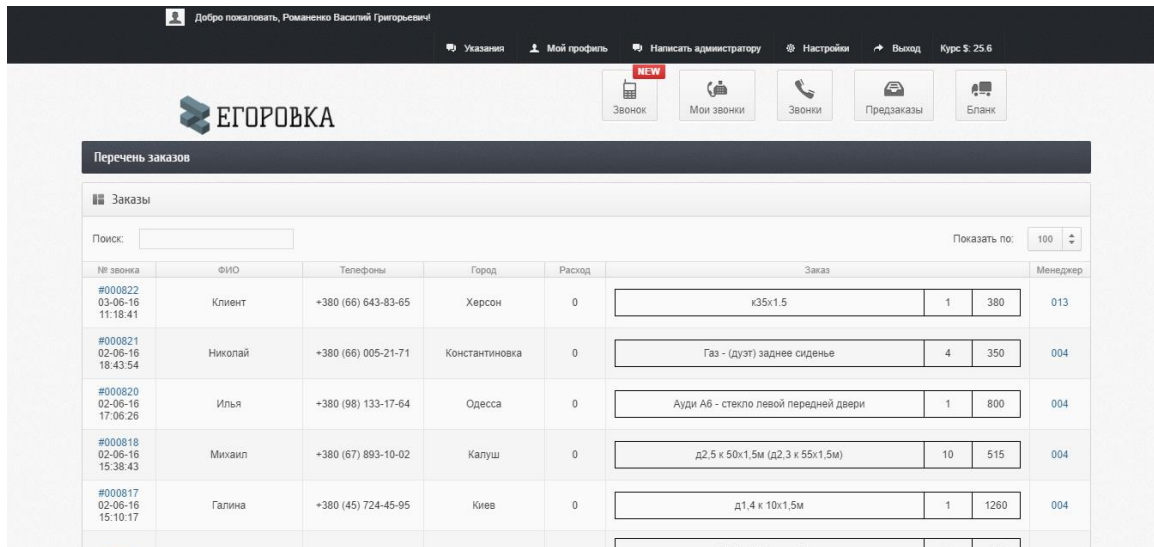


Рисунок 3.3 – Вид вікна формування столу заказів

Нижче приведено код і лістинг програми формування столу заказів на мові PHP.

```

<?php
include 'include/template/header.php';
if($_SESSION['id']!=" " && $_SESSION['access']!="3")
{
?>
</head>
<body>
<?php
include 'include/template/top_band.php';
include 'include/template/top_menu.php';
?>
<div class="wrapper">
<div class="content" style="width: 120%;margin-left: -10%;">
<div class="title"><h5>Перечень заказов</h5></div>
<div class="table" style="margin-top: 10px;">
<div class="head"><h5 class="iFrames">Заказы</h5></div>
<table cellpadding="0" cellspacing="0" border="0" class="display"
id="example" >
<thead>
<tr>
<th>№ звонка</th>
<th>ФИО</th>
<th>Телефоны</th>
<th>Город</th>
<?php

```

```

        if($_SESSION['access']=="1")
        {
            echo '<th>Расход</th>';
        }
    ?>

        <th>Заказ</th>
        <th>Менеджер</th>

    </tr>
</thead>
<tbody>
    <?php
    if($_SESSION['access']=="3")
    {
        $query = "SELECT * FROM `scroll_call` WHERE `status`='0' OR
`status`='2' OR `status`='5' AND `manager`='".$_SESSION['id']."' ORDER BY `id`
DESC";
    }
    else
    {
        $query = "SELECT * FROM `scroll_call` WHERE `status`='0' OR
`status`='2' OR `status`='5' ORDER BY `id` DESC";
    }

    $res = mysql_query($query) or die(mysql_error());
    while ($result = mysql_fetch_array($res))
    {
        $number=str_pad($result['id'], 6,
'0', STR_PAD_LEFT);

        $manager=str_pad($result['manager'], 3, '0', STR_PAD_LEFT);
        $date=explode(" ",
$result['date_update']);
        $manager_name=$result['manager'];
        $query_manager_name = "SELECT full_name FROM `users`
WHERE id='".$_SESSION['manager']."'";
        $res_manager_name = mysql_query($query_manager_name) or
die(mysql_error());
        $result_manager_name =
mysql_fetch_array($res_manager_name);
        $manager_name=$result_manager_name['full_name'];

        $name_client=$result['name_client'];

        $order = explode("/",
$result['order_content']);

        $order_count=count($order);
        echo '<tr class="gradeA">';
        echo '<td style="vertical-align:
middle;width: 65px;" ><center><a
href="/info_order.php?id='.$result['id'].'">#'.$number.'</a><br>'.$date[0].<br>'.$date[1].</
center></td>';

        echo '<td style="vertical-align:
middle;width: 150px;"><center>'.$name_client.'</center></td>';
    }
}

```

					БКС 26.08.004 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

```

        echo '<td style="vertical-align:
middle;width: 120px;"><center>'. $result['phone_client'].'</center></td>';
        echo '<td style="vertical-align:
middle;width: 120px;"><center>'. $result['city_client'].'</center></td>';
        if($_SESSION['access']=="1")
        {
            echo '<td style="vertical-align: middle;width:
50px;"><center>'. $result['consumption'].'</center></td>';
        }
        echo '<td style="vertical-align: middle;"><center>';
            for($i=0;$i<$order_count-1;$i++)
            {
                $order_arr =
explode("|", $order[$i]);
                echo '<table
width="100%" cellpadding="2" cellspacing="1" style="border: 1px groove black;" >
                <tr>
                <td ><center>'. $order_arr[0]. '</center></td>
                <td style="width:35px;border: 1px groove
black;"><center>'. $order_arr[1]. '</center></td>
                <td style="width:35px;border: 1px groove
black;"><center>'. $order_arr[2]. '</center></td>
                </tr>
                </table>';
            }
            echo '</center></td>';
            if($_SESSION['access']=="1")
            {
                echo '<td
style="vertical-align: middle;width: 50px;"><center><a
href="/info_user.php?id='. $result['manager'].'" target="_blank"
title="'. $manager_name.'">'. $manager.'</a></center></td>';
            }
            else
            {
                echo '<td
style="vertical-align: middle;width: 50px;"><center>'. $manager.'</center></td>';
            }
            echo '</tr>';
        }
    }
?>
</tbody>
</table>
</div>
</div>
</div>
<?php
include 'include/template/footer.php';
}
else
{

```

					БКС 26.08.004 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

        echo '<script language="JavaScript"> window.location.href =
"/index.php"</script>';
    }

```

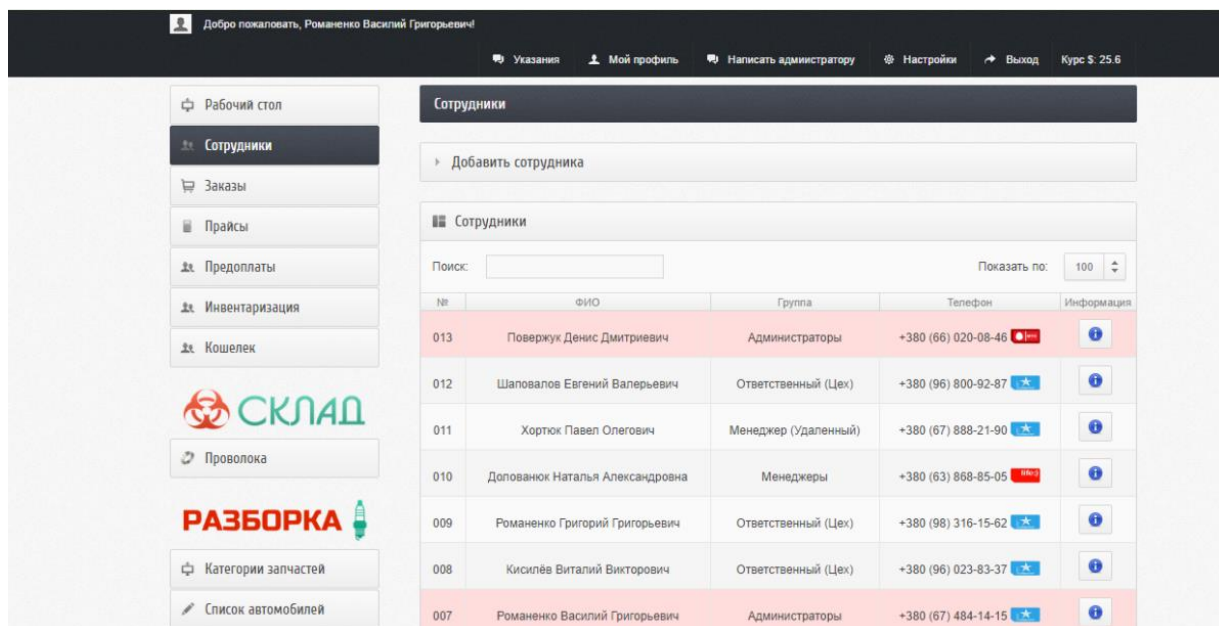


Рисунок 3.3 – Вид вікна «Співробітники»

Нижче приведено код і лістинг програми формування модулю «Співробітники» на мові PHP із функцією додавання нового співробітника. Вікно додавання нового співробітника приведено на рисунку 3.4.

```

</script>
<style>
    .datepicker {
        width: 510px!important;
        cursor: pointer;
    }
</style>
</head>
<body>
<?php
include 'include/template/top_band.php';
include 'include/template/top_menu.php';
?>
<!-- Content wrapper -->
<div class="wrapper">
<?php
include 'include/template/menu.php';
?>

```

```

<!-- Content -->
<div class="content">
  <div class="title"><h5>Сотрудники</h5></div>
  <?php if($_SESSION['access']=="1"){?>
    <div class="widget" style="margin-top: 20px;">
      <div class="head" style="text-align: center;"><h5>Добавить
сотрудника</h5></div>
      <div class="body">
        <form method="POST" id="add_user"
action="javascript:void(null);" onsubmit="add_user()" class="mainForm">
          <!-- Input text fields -->
          <fieldset>
            <div class="widget" style="margin-top: 10px;">
              <div class="head"><h5>
class="iList">Форма добавления</h5></div>
              <div class="rowElem" style="border: 1px solid #ccc; padding: 5px;">
                <div class="formRight"><input type="text"
name="full_name" /></div></div>
              <div class="rowElem"><label>Дата рождения:</label><div class="formRight"><input
name="birthday" class="datepicker" style="width: 0px;" type="text"></div></div>
              <div class="rowElem">
                <label>Группа:</label>
                <div class="formRight">
                  <select style="width: 512px;" data-placeholder="Выберите группу..." class="select"
name="access" tabindex="2">
                    <option value=""></option>
                    <?php
                    $query = "SELECT * FROM `access`";
                    $res = mysql_query($query) or die(mysql_error());
                    while ($result = mysql_fetch_array($res)) {
                      echo '<option value="'. $result['id']. "'>'. $result['name']. '</option> ';
                    }
                    ?>
                  </select>
                </div>
              </div>
            </div>
          </fieldset>
        </form>
      </div>
    </div>
  </div>

```

					БКС 26.08.004 КРБ ПЗ	Арк. 51
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class="rowElem"><label>Логин:</label><div class="formRight"><input type="text"
name="login"/></div></div>

class="rowElem"><label>Карта (ПриватБанк):</label><div class="formRight"><input
type="text" name="card_number" value="5168 7420 1811 3831"/></div></div>

class="rowElem"><label>Пароль:</label><div class="formRight"><input type="text"
name="password" /></div></div>

class="rowElem"><label>Личный E-Mail:</label><div class="formRight"><input
type="text" name="personal_mail" /></div></div>

class="rowElem"><label>Личный телефон:</label><div class="formRight"><input
class="maskIntPhone" name="personal_phone" value="" type="text"></div></div>

class="submitForm"><input type="submit" value="Добавить сотрудника"
class="greenBtn" /></div>

id="result_add_user"></div>

```

```

</div>
</fieldset>
</form>
</div>
</div>
<?php } ?>
<div class="table" style="margin-top: 20px;">
<div class="head">
<h5 class="iFrames">Сотрудники</h5>
</div>
<table cellpadding="0" cellspacing="0" border="0"
class="display" id="example">
<thead>
<tr>
<th>№</th>
<th>ФИО</th>
<th>Группа</th>
<th>Телефон</th>
<?php
if($_SESSION['access']=="1"){ ?> <th>Информация</th><?php } ?>
</tr>
</thead>
<tbody>
<?php
$query = "SELECT * FROM `users`";
$res = mysql_query($query) or die(mysql_error());
while ($result = mysql_fetch_array($res)) {
    $access=$result['access'];

```

					БКС 26.08.004 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

```

`access` WHERE id='".$access."'";
or die(mysql_error());
mysql_fetch_array($res_group);
$result['personal_phone'], $code);
$code[1]=="93"){ $operator='';}
$code[1]=="95" || $code[1]=="99"){ $operator='';}
$code[1]=="97" || $code[1]=="98" || $code[1]=="96"){ $operator='';}

if($access=="1"){ $style="gradeX";}else{ $style="gradeA";}

if($result['activity']=="0"){ $name='<strike>'.$result['full_name'].'</strike>';}
else { $name=$result['full_name'];}
$id=str_pad($result['id'], 3, '0',
STR_PAD_LEFT);

echo '<tr class="'.$style.'">
<td><center>.$id.</center></td>

<td><center>.$name.</center></td>

<td><center>.$group.</center></td>

<td><center>.$result['personal_phone'].'.$operator.</center></td>';
if($_SESSION['access']=="1")
{
echo '<td class="center"><center><a
href="/info_user.php?id='.$result['id'].'" title="" class="btn14 mr5"></a></center></td>';
}
echo '</tr>';
}
?>
</tbody>
</table>
</div>
</div>

<?php
include 'include/template/footer.php';
}
else

```

					БКС 26.08.004 КРБ ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
echo '<script language="JavaScript"> window.location.href = "/index.php"</script>';
}
?>

```

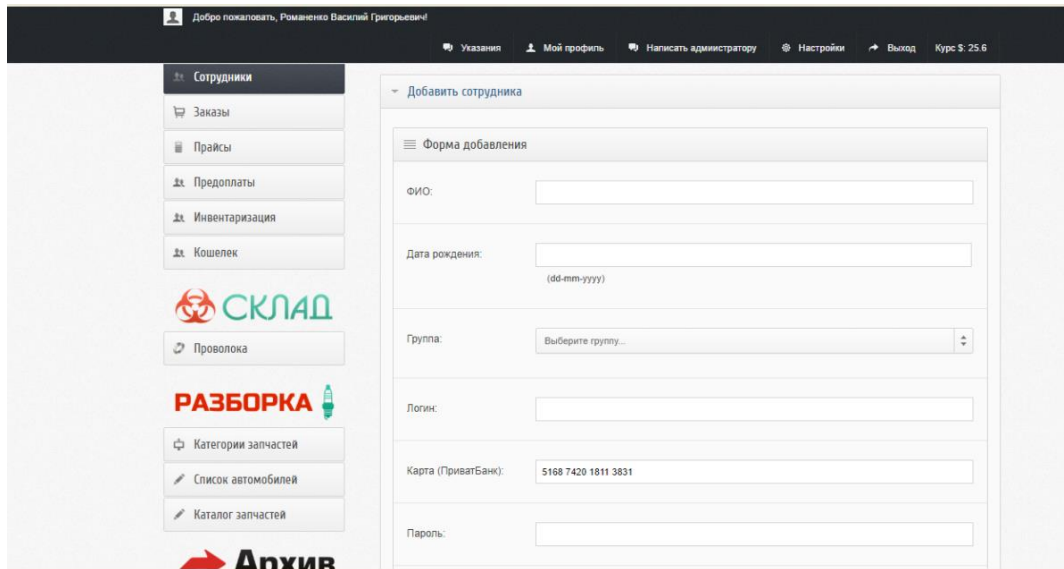


Рисунок 3.3 – Вид вікна «Додавання співробітника»

3.3 Вимоги до технічних засобів

Для забезпечення нормальної працездатності програми необхідно мати комп'ютер з конфігурацією не нижче:

- Процесор - не нижче Core i5;
- Оперативна пам'ять – 8Гб і більше;
- Вільне місце на диску – 200 Мб;
- Монітор SVGA;
- Клавіатура;
- Комп'ютерна миша;
- Підключення до мережі Інтернет.

Дана програма працює під управлінням сімейства операційних систем MS Windows. Також на комп'ютері повинно бути встановлено ПЗ MS Office. На пристроях користувача інформація відображається у вікні браузера під керуванням будь яких операційних систем.

					БКС 26.08.004 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

4 ОХОРОНА ПРАЦІ

4.1 Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці. В Україні діє Закон "Про охорону праці", який є одним з найважливіших актів законодавства про охорону праці. Закон визначає основні принципи державної політики в галузі охорона праці, чільне місце серед яких належить пріоритету життя і здоров'я працівника по відношенню до результатів виробничої діяльності підприємства, принципи повної відповідальності власника за створення безпечних і нешкідливих умов праці, соціального захисту працівників повного відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві чи від професійного захворювання.

Під час роботи на виробництві на людину можуть впливати один, або низка небезпечних та шкідливих виробничих факторів.

Оскільки тема дипломного проекту пов'язана з розробкою програмного забезпечення, то розглянемо саме негативні фактори, що впливають на роботу програміста.

4.2 Коротка характеристика і основні вимоги безпеки до робочого місця програміста

На робочому місці програміста виникають небезпечні та шкідливі фактори: підвищений рівень шуму, несприятливі мікрокліматичні умови, недостатній рівень освітленості, шкідливі речовини, підвищений рівень

					БКС 26.08.004 КРБ ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

електромагнітних випромінювань радіочастот, висока напруга електричної мережі, статична електрика та інші.

Головним завданням будь-якої галузі промисловості є збільшення продуктивності праці. Разом з тим, людина що працює, проводить на виробництві значну частину свого життя. Тому для її нормальної життєдіяльності в умовах виробництва треба створити умови, які б дали змогу їй плідно працювати та зберігати своє здоров'я. Ці умови створюються забезпеченням для працюючого:

- 1) зручного робочого місця;
- 2) чистого повітря;
- 3) нормованої освітленості;
- 4) захисту від шуму та вібрацій;
- 5) захисту від дії шкідливих речовин та випромінювань;
- 6) робочим одягом та різними засобами індивідуального захисту;
- 7) побутовими приміщеннями та спеціальними службами, що призначені для створення безпечних та нормальних умов праці.

Об'ємно-планувальні рішення будівель та приміщень, де експлуатуються відеодисплейні термінали (ВДТ) мають відповідати вимогам ДСанПІН 3.3.2.007-98. Розміщення робочих місць з ВДТ ЕОМ і ПЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено. Для приміщень, які призначені для роботи з ВДТ, доцільно обрати орієнтацію вікон на північ або північний схід. На вікнах повинні бути жалюзі, що регулюються, або штори, що дають можливість їх повністю закривати. Приміщення повинні мати природне та штучне освітлення. При приміщеннях з ВДТ мають бути обладнані побутові приміщення для відпочинку, психологічного розвантаження тощо. Площа на одне робоче місце для програмістів повинна складати не менше 6 кв.м., а об'єм – не менше 20 куб.м. Стіни повинні бути пофарбовані матовою фарбою. У виробничих приміщеннях, в яких робота на ВДТ і ПЕОМ є допоміжною, температура,

					БКС 26.08.004 КРБ ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

відносна вологість і швидкість руху повітря на робочих місцях повинні відповідати санітарним нормам мікроклімату виробничих приміщень.

Конструкція робочого місця користувача ПК й взаємне розташування всіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи. Конструкція робочих меблів повинна забезпечувати можливість індивідуального регулювання відповідно росту працюючих для підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей на відстані близько 70 см, що укладається в у припустимі рамки від 60 до 90 см. Частота мерехтіння екрана $f_{\text{мер}}=100$ Гц, що відповідає умові $f_{\text{мер}}>70$ Гц.

Робоче місце розташоване перпендикулярно віконним прорізам, це зроблено з тією метою, щоб виключити пряму й відбиту мерехтливність екрана від вікон і приладів штучного освітлення.

Згідно темі кваліфікаційної роботи робоче місце програміста укомплектовано пристроями з електромагнітним випромінюванням.

Мікроклімат

У виробничих приміщеннях, в яких робота на ВДТ і ПЕОМ є основною (диспетчерські, операторські, розрахункові, кабінети і пости управління, зали обчислювальної техніки і ін.) повинні забезпечуватися оптимальні параметри мікроклімату.

Мікроклімат виробничих приміщень нормується в залежності від теплових характеристик виробничих приміщення, категорії робіт по важкості і періоду року. Основні нормативні документи, де наводяться норми мікроклімату – це санітарні норми та стандарти безпеки праці.

Оптимальні мікрокліматичні умови – це такі параметри температури, вологості і швидкості руху повітря, які при тривалому і

					БКС 26.08.004 КРБ ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

систематичному впливі на людину забезпечують нормальний тепловий стан організму без напруги і порушення механізмів терморегуляції.

При нормуванні мікроклімату календарний рік поділяється на два періоди:

- 1) холодний період (середньодобова температура нижча +10 °С);
- 2) теплий період (середньодобова температура становить +10 °С).

Найбільш значним фактором продуктивності й безпеки праці є виробничий мікроклімат, що характеризується температурою й вологістю повітря, швидкістю його руху, а також інтенсивністю радіації, і повинен відповідати ГОСТ 12.1.005-88 і СНиП 2.04.05-86.

Вентиляція

Вентиляція - це регульований повітрообмін у приміщенні. Вентиляцією називають також пристрої, які її створюють. Для підтримки в приміщеннях нормального складу повітря, що відповідає гігієнічним вимогам складу повітря, видалення з нього шкідливих газів, пару і пилу використовують вентиляцію. Дипломним проектом передбачено встановлення припливно-витяжної системи вентиляції, а також можливе застосування кондиціонерів.

Освітлення виробничих приміщень

Одним з основних питань охорони праці є організація раціонального освітлення виробничих приміщень і робочих місць. Для освітлення приміщення, у якому працює програміст, використовується змішане освітлення, тобто сполучення природного й штучного освітлення. Для загального освітлення приміщення, де перебуває робоче місце програміста, використовуються газорозрядні лампи типу ЛД. Нормами для даних робіт встановлена необхідна освітленість робочого місця $E_{н}=300$ лк (для робіт високої точності, коли найменший розмір об'єкта розрізнення дорівнює 0,3 – 0,5 мм).

Заходи щодо захисту від дії шуму та вібрації

					БКС 26.08.004 КРБ ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

У робочих приміщеннях основними джерелами акустичних шумів є шуми ПЕОМ. ЕОМ є також джерелами шумів електромагнітного походження (коливання елементів електромеханічних пристроїв під впливом змінних магнітних полів). Крім того, у даних приміщеннях, виникає структурний шум, тобто шум, випромінюваний поверхнями коливних конструкцій стін, перекриттів, перегородок будинку у звуковому діапазоні частот. Систематичний шум може викликати стомлення слуху й ослаблення звукового сприйняття, а також значне стомлення всього організму. Однак не всі шуми шкідливі. Так, звичні не різко виражені шуми, що супроводжують трудовий процес, можуть сприятливо впливати на хід роботи; нерізкі шуми, що характеризуються періодичністю звуків, наприклад, музика, у силу своєї ритмічності не тільки не відволікають від роботи, але й викликають позитивні емоції, сприяють підвищенню ефективності праці. Для усунення або ослаблення несприятливих шумових впливів доцільно ізолювати робочі приміщення, розміщаючи їх у частинах будинку, найбільш вилучених від міського шуму - розташованих у глибині будинку, звернених вікнами у двір. Перевіряти герметичність корпусів комп'ютерів та своєчасно міняти вентилятори охолодження.

Електробезпека

Приміщення де використовуються імпульсні джерела живлення відповідно до ОНТП24-86 і ПУЕ-87 відноситься до класу приміщень без підвищеної небезпеки поразки персоналу електричним струмом, оскільки відносна вологість повітря не перевищує 75%, температура не більш 35°C, відсутні хімічно агресивні середовища. Живлення електроприладів усередині приміщення здійснюється від двухфазної мережі з заземленою нейтраллю напругою 220 В і частотою 50 Гц із використанням автоматів токового захисту. У приміщенні повинна бути застосована схема заземлення. Ураження людини електричним струмом може відбутися у випадку:

- 1) дотику до відкритих струмоведучих частин;

					БКС 26.08.004 КРБ ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

2) у результаті дотику до струмопровідних не струмоведучих елементів устаткування, що опинилися під напругою в результаті порушення ізоляції або з інших причин. Заземлення повинно бути зроблено за допомогою гнучкого сплетеного мідного проводу діаметром порядку 1,5 мм². Для зменшення значень напруг дотику і відповідних їм величин струмів, при нормальному й аварійному режимах роботи устаткування необхідно виконати повторне захисне заземлення нульового проводу. Відповідно до ГОСТ-12.2.007.0-75 устаткування (крім ЕОМ - II клас) відноситься до I класу, воно має робочу ізоляцію відповідно до вимог ГОСТ 12.1.009-76. Підключення устаткування виконане відповідно до вимог ПБЕ та ПУЕ. Додаткових заходів по електробезпечності не потрібно.

4.3 Пожежна безпека

Пожежа - неконтрольоване горіння поза спеціальним вогнищем, яке призводить до матеріальної шкоди. Пожежна безпека – стан об'єкта, при якому з регламентованою ймовірністю виключається можливість виникнення та розвиток пожежі і впливу на людей її небезпечних факторів, а також забезпечується захист матеріальних цінностей. Причинами пожеж та вибухів на підприємстві є порушення правил і норм пожежної безпеки, невиконання Закону “Про пожежну безпеку”. За стан пожежної безпеки на підприємстві відповідають її керівники, начальники цехів, майстри та інші керівники. Можливими причинами виникнення пожежі в приміщенні є:

- 1) коротке замикання проводки;
- 2) користування побутовими електрорадіоприладами .
- 3) не дотримання умов протипожежної безпеки.

Для гасіння пожеж на робочому місці використовують вуглекислотні та порошкові вогнегасники.

Вуглекислотні вогнегасники випускаються як ручні (ВВК-5).

Порошкові вогнегасники ВП-2, ВП-5, ВП-10 та інш.

					БКС 26.08.004 КРБ ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Наявність первинних засобів пожежегасіння і вогнегасників, їхня кількість і зміст відповідає вимогам ГОСТ 12.4.009-75 і ISO3941-77. У приміщенні виконуються усі вимоги по пожежній безпеці відповідно до вимог НАПБ А.0.001-95 “Правила пожежної безпеки в Україні”. У приміщенні також мається план евакуації на випадок виникнення пожежі. Час евакуації відповідає вимозі СНиП 2.01.02-85, а максимальне видалення робочих місць від евакуаційних виходів відповідає СНиП 2.09.02-85.

Висновки. Виконання вищеперерахованих заходів забезпечує комфортну та небезпечну експлуатацію комп’ютерів та компонентів комп’ютерних мереж.

					БКС 26.08.004 КРБ ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфікаційній роботі досліджено актуальні питання розробки програмного забезпечення і створення ERP на мові PHP за архітектурою «клієнт-сервер».

В результаті проведеної роботи розроблено динамічний сайт, з підключеною до нього базою даних MySQL, і можливістю управління даними, авторизацією і реєстрацією користувачів.

При цьому мною були вирішені наступні завдання:

- вибір стратегії розробки програмного продукту;
- використання мови PHP для розробки динамічного інтуїтивно зрозумілого сайту зі зручним для користувача інтерфейсом;
- створення, структурування і робота з базами даних.

В результаті проведених робіт була створена система управління документообігу з дистанційним управлінням.

Використання даної системи дозволить керівникові знизити тимчасові витрати на обробку і узагальнення відомостей про бази даних складу, тим самим підвищить ефективність його діяльності.

Результати проведеної роботи свідчать про придатність даного програмного комплексу до впровадження і широкого застосування на різних складах підприємств України.

У розділі охорона праці дані рекомендації по охороні праці програмістів іа операторів персональних комп'ютерів.

					БКС 26.08.000.00 КРБ ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про інформацію: Закон України// Відомості Верховної Ради України. - 2001.- № 11.- С. 25-27.
2. Про електронні документи та електронний документообіг: Закон України// Вісник Держ. комітету архівів України. - 2003. - Вип. 2 (14). - С. 15-22. .
3. Гавриш С. Система автоматизації діловодства та електронного документообігу/ С.Гавриш// Довідник секретаря та офіс-менеджера. – 2007. – № 2. – С. 25-31.
4. Мешков Д.Ю. Електронний документ: обіг, зберігання, використання, законодавчі аспекти/ Д.Ю. Мешков// Архіви України. - 2001. - № 6. - С. 73-87.
5. Основи інформаційних систем: Навчальний посібник / За ред. проф. В.Ф. Ситника. – К.: КНЕУ, 2007. -252 с.
6. Чорновол Н.І. Методичні вказівки до розділу з охорони праці. О. ОТК ОНАХТ, 2022.
7. Якоб Нильсен Веб-дизайн. — СПб: Символ-Плюс, 2003. — 512 с. — ISBN 5-93286-004-9
8. Якоб Нильсен, Хоа Лоранжер Web-дизайн: удобство использования Web сайтов = Prioritizing Web Usability. — М.: «Вильямс», 2007. — С. 368. — ISBN 0-321-35031-6
9. Роббинс Д. Web-дизайн. Справочник.. — "КУДИЦ-ПРЕСС", 2008. — С. 816. — ISBN : 978-5-91136-039-9
10. Гончаров А. Ю. Web-дизайн: HTML, JavaScript и CSS. Карманный справочник. — "КУДИЦ-ПРЕСС", 2007. — С. 320. — ISBN : 978-5-91136-024-5
11. Кристиан Уэнц PHP. Карманный справочник = PHP Phrasebook. — М.: «Вильямс», 2007. — С. 384. — ISBN 0-672-32817-8

					БКС 26.08.000.00 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

12. Эд Леки-Томпсон, Алек Коув, Стивен Новицки, Хьяо Айде-Гудман РНР 5 для профессионалов = Professional PHP 5. — М.: «Диалектика», 2006. — С. 608. — ISBN 0-7645-7282-2
13. Основи охорони праці. [Электронный ресурс] – Режим доступа: <http://library.if.ua/book/9/984.html> (Дата звернення 14.06.22)

					БКС 26.08.000.00 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

ДОДАТОК Презентація 10 слайдів

«Дослідження та впровадження
програмної системи планування ресурсів
на підприємстві»

виконав гр. 2БКК-26

Марченко О.В.

Керівник Краснієнко Н.В.

Мета комплексної роботи:

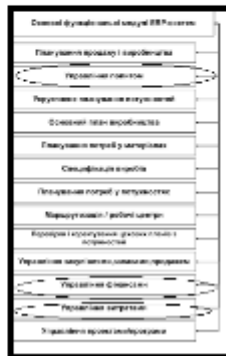
- -вибір стратегії розробки програмного продукту
- - використання мови PHP для розробки динамічного інтуїтивно зрозумілого сайту зі зручним інтерфейсом користувача;
- - створення, структурування та робота з базами даних складу.

					БКС 26.08.000.00 КРБ ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Зв'язок різних ERP-систем



Основні функціональні модулі ERP-системи. Для системи складського обліку обрані модулі 2,11,12



Змн.	Арк.	№ докум.	Підпис	Дата

БКС 26.08.000.00 КРБ ПЗ

Арк.

66

Схема структурна системи



Скрін-шот вікон системи



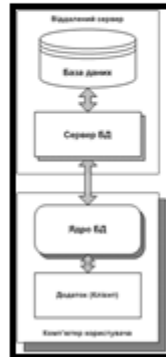
Вид вікна входу користувача в систему за логіном та паролем



Вид вікна Робочого стола користувача системи

					БКС 26.08.000.00 КРБ ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

Архітектура клієнт-сервер, яка передбачає зв'язок між користувачем та базою даних через інтернет



Модель системи в Інтернеті, яка дозволяє використовувати доступ до бази даних з різних пристроїв.



Скріншоти вікон системи

Вид вікна Співробітники



Вид вікна Додавання нового співробітника



ВИСНОВКИ

У комплексній роботі представлено систему управління ресурсами у відповідності з дистанційним управлінням. Результатом є розробка програмних додатків: Робочий стіл користувача, система «Співробітники» та «Додавання нового співробітника».

Використання складського обліку дозволяє своєчасно ознайомлювати персонал із усіма показниками системи компанії.

Розроблена система підлягає вдосконаленню за допомогою розробки додаткових модулів контенту.

Змн.	Арк.	№ докум.	Підпис	Дата

БКС 26.08.000.00 КРБ ПЗ

Арк.

69