

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Литвиненко О.Є.
“ _____ ” _____ 2022 р.

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

ЗА НАПРЯМОМ ПІДГОТОВКИ 6.050102 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

Тема: «Автоматизована система обліку транспортних послуг на основі SAP»

Виконавець: студент, група СП-426, Ковтун Вадим Євгенійович
(студент, група, прізвище, ім'я, по-батькові)

Керівник: д.т.н., доц. Вавіленкова Анастасія Ігорівна
(науковий ступінь, вчене звання, прізвище, ім'я, по-батькові)

Нормоконтролер:

(підпис)

Тупота Є.В.

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Напрямок (спеціальність) 6.050102 «Комп'ютерна інженерія»

(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.Є. Литвиненко

« _____ » _____ 2022 р.

ЗАВДАННЯ

на виконання дипломної роботи (проєкту)

Ковтуна Вадима Євгенійовича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи (проєкту) «Автоматизована система обліку транспортних послуг на основі SAP»
затверджена наказом ректора від «15»лютого 2022 р. №251/ст.

2. Термін виконання роботи (проєкту): з 16 травня 2022 р. по 19 червня 2022 р.

3. Вихідні дані до роботи (проєкту): системи обліку транспортних послуг, системи планування ресурсів підприємства, структура систем на основі SAP

4. Зміст пояснювальної записки:

- 1) автоматизовані системи обліку;
- 2) технології та алгоритми SAP для організації роботи автоматизованої системи обліку транспортних послуг
- 3) програмна реалізація автоматизованої системи обліку транспортних послуг на основі SAP;

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

- 1) процес роботи Floorplan Manager;
- 2) функціонування системи обліку транспортних послуг;
- 3) трирівнева архітектура SAP;
- 4) демонстрація роботи системи обліку транспортних послуг.

6. Календарний план-графік

| № з/п | Завдання | Термін виконання | Підпис керівника |
|-------|---|---------------------------|------------------|
| 1. | Аналіз джерел за темою дослідження. Розроблення плану дипломного проекту | 16.05.2022- 20.05.2022 | |
| 2. | Затвердження плану дипломного проекту та проведення консультацій з науковим керівником, щодо наповнення пояснювальної записки | 21.05.2022- 23.05.2022 | |
| 3. | Дослідження систем обліку транспортних послуг | 24.05.2022- 26.05.2022 | |
| 4. | Розробка архітектури автоматизованої системи обліку транспортних послуг | 27.05.2022- 29.05.2022 | |
| 5. | Створення автоматизованої системи обліку транспортних послуг | 30.05.2022- 01.06.2022 | |
| 6. | Написання пояснювальної записки | 02.06.2022- 04.06.2022 | |
| 7. | Підготовка демонстраційного матеріалу | 05.06.2022 07.06.2022 | |

7. Дата видачі завдання: « 15 » лютого 2022 р.

Керівник дипломного проекту

_____ (підпис керівника)

д.т.н., доц. Вавіленкова А.І.
(П.І.Б.)

Завдання прийняв до виконання

_____ (підпис випускника)

Ковтун В.Є.
(П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Автоматизована система обліку транспортних послуг на основі SAP»: 68 с., 42 рис., 2 табл., 15 джерел.

УПРАВЛІННЯ, ОБЛІК, СИСТЕМИ ОБЛІКУ, ТРИРІВНЕВА АРХІТЕКТУРА, ТРАНСПОРТ, СИСТЕМИ ПЛАНУВАННЯ РЕСУРСІВ ПІДПРИЄМСТВА

Об'єктом дослідження даного дипломного проекту є процес автоматизації обліку транспортних послуг .

Предметом дослідження є автоматизовані системи обліку транспортних послуг.

Метою даного дипломного проекту розробка автоматизованої системи обліку транспортних послуг на основі SAP.

Методи дослідження – технології систем автоматизованого обліку надання транспортних послуг, порівняльний аналіз, методи об'єктно-орієнтованого програмування, методи проектування за трирівневою архітектурою.

Здійснено огляд існуючих систем для обліку транспортних послуг; здійснено порівняльний аналіз їх функціональних можливостей; проаналізовано структуру ; здійснено огляд принципів трирівневої архітектури; реалізовано автоматизовану систему обліку надання транспортних послуг на основі SAP.

Матеріали дипломного проекту **рекомендується використовувати** при проведенні досліджень систем планування ресурсів підприємства, у навчальному процесі фахівців з системного програмування, а також у сферах, пов'язаних з системами планування ресурсів підприємства.

Прогнозні припущення про розвиток об'єкту та предмету дослідження – застосування в якості системи обліку надання транспортних послуг.

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ | 7 |
| ВСТУП..... | 8 |
| РОЗДІЛ 1 АВТОМАТИЗОВАНІ СИСТЕМИ ОБЛІКУ | 11 |
| 1.1. Застосування систем обліку | 11 |
| 1.2. Аналіз використання ERP-систем у сфері автоматизованого обліку транспортних послуг | 16 |
| 1.2.1. 1С: Предприятие 8..... | 16 |
| 1.2.2. BAS Управління автотранспортом | 18 |
| 1.2.3. Модуль управління транспортом від Jsolutions | 21 |
| 1.2.4. Oracle Transportation Management..... | 23 |
| 1.2.5. Modalkon – Управління автотранспортом | 24 |
| 1.2.6. Висновки до розділу..... | 26 |
| РОЗДІЛ 2 ТЕХНОЛОГІЇ ТА АЛГОРИТМИ SAP ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ТРАНСПОРТНИХ ПОСЛУГ | 27 |
| 2.1. Технологія розробки ERP SAP R/3 | 27 |
| 2.1.1. Загальна архітектура SAP R/3 | 27 |
| 2.1.2. Мова програмування ABAP | 28 |
| 2.1.3. Floorplan Manager | 30 |
| 2.1.4. Business Object Processing Framework | 31 |
| 2.1.5. FPM-BOPF Integration | 33 |
| 2.2. Структура системи автоматизованої систем обліку транспортних послуг на основі SAP | 34 |
| 2.2.1. Архітектура SAP R/3..... | 34 |
| 2.3. Алгоритм роботи автоматизованої системи обліку транспортних послуг . | 38 |
| 2.4. Висновки до розділу..... | 51 |
| РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ТРАНСПОРТНИХ ПОСЛУГ НА ОСНОВІ SAP | 52 |

| | |
|---|----|
| 3.1. Демонстрація роботи автоматизованої системи обліку транспортних послуг на основі SAP | 52 |
| 3.2. Кількісна оцінка роботи автоматизованої системи обліку транспортних послуг на основі SAP | 60 |
| 3.3. Висновки до розділу | 61 |
| ВИСНОВКИ | 62 |
| СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 64 |
| Додаток А | 66 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

ERP – Enterprise resource planning

HR – Human Resources

CRM – Customer Relationship Management

ПЗ – програмне забезпечення

API – Application Programming Interface

ВСТУП

Актуальність. Системи планування ресурсів підприємства (ERP) стали невід’ємною складовою сучасного бізнесу. З ростом підприємства неодмінно збільшуватиметься кількість та складність його бізнес-процесів, обслуговувати котрі вручну буде все складніше.

Раніше ERP-системи були монолітними програмними пакетами, кожен з яких унікально допрацьовувався та не контактував з іншими системами. Такий підхід потребував значних фінансових та часових витрат, адже кожному систему необхідно було щоразу перероблювати під потреби конкретної компанії.

Сучасні ERP-системи широко використовуються у багатьох сферах: HR, транспортні перевезення, робота з клієнтами, бухгалтерія та інших. Такі системи активно взаємодіють, виводячи обсяг доступної інформації на новий рівень.

Однією з найбільших компаній на ринку ERP є німецька компанія SAP, Компанія широко представлена на західних ринках, але присутня і в Україні. Так, наприклад, частково працюють на ERP від компанії SAP такі компанії як АТ «Укргазвидобування» та АТ «Укртранснафта»[1].

Отже, з ростом підприємства невідворотно збільшується і кількість бізнес-процесів, за якими необхідно стежити, які треба налагоджувати та оптимізувати. Для спрощення цієї задачі існують системи автоматизації бізнес-процесів.

Автоматизація є актуальною тенденцією для багатьох галузей. Великий відсоток ринку займає компанія SAP зі своїм продуктом SAP R/3. Тож, оскільки чимало підприємств вже використовують цю систему, її підтримка буде актуальною ще довго.

Окрім SAP, розповсюдженими ERP-системами на українському ринку є:

- SyteLine ERP, компанії Infor
- BAS ERP, компанії BAS
- Oracle Business Suite, компанії Oracle
- Microsoft Dynamics, компанії Microsoft
- 1С Предприятие v8, компанії 1С
- Галактика ERP, компанії Галактика
- Jsolutions ERP, компанії Jsolutions

Об'єктом дослідження даного дипломного проекту є процес автоматизації обліку транспортних послуг .

Предметом дослідження є автоматизовані системи обліку транспортних послуг.

Метою даного дипломного проекту розробка автоматизованої системи обліку транспортних послуг на основі SAP.

Методи дослідження – технології систем автоматизованого обліку надання транспортних послуг, порівняльний аналіз, методи об'єктно-орієнтованого програмування, методи проектування за трирівневою архітектурою.

За допомогою порівняльного аналізу було досліджено існуючі сучасні транспортні модулі ERP-систем, визначено їх переваги та недоліки. З огляду на існуючі технології систем управління ресурсами підприємства був виділений основний функціонал систем для ведення обліку. За допомогою об'єктно-орієнтованих методів програмування та принципів трирівневої архітектури було реалізовано систему для автоматизованого обліку транспортних послуг .

Також, за допомогою відомостей про методи об'єктно-орієнтованого програмування було виявлено принципи побудови класів, методів та інтерфейсів у рамках трирівневої архітектури SAP R/3, визначено необхідні для їх побудови та використання лексеми.

З огляду на функціональні можливості існуючих систем для обліку транспортних послуг, у дипломному проекті створено систему для вирішення таких задач для невеликих організацій. Для цього було запропоновано систему, що має весь основний функціонал для обліку, будучи при цьому простою у використанні.

Новизна дипломного проекту полягає у проектуванні простої та не вибагливої системи для автоматизованого обліку транспортних послуг на основі трирівневої архітектури. Використання трирівневої архітектури дало

зможу реалізувати систему з гнучкою структурою, даючи можливість подальшої модифікації та вдосконалення функціоналу без значних зусиль.

Практичне значення отриманих результатів. У дипломному проєкті створено систему автоматизованого обліку транспортних послуг. Вона надає базовий функціонал систем обліку. Система дає змогу розраховувати дохід підприємства, зарплати робтників, супроводжувати таблиці з інформацією.

Особистий внесок випускника. Всі результати, представлені у дипломній роботі, отримані випускником особисто. Особливої уваги заслуговує створення системи обліку транспортних послуг на основі SAP.

Практичні значення отриманих результатів в дипломному проєкті дають змогу використовувати отримані дані при проведенні досліджень роботи систем планування ресурсів, у навчальному процесі фахівців з системного програмування, а також у сферах, пов'язаних з обліком транспортних послуг.

Прогнозні припущення про розвиток об'єкту та предмету дослідження – застосування в якості системи обліку надання транспортних послуг.

В результаті виконання дипломного проєкту було розроблено порівняльну характеристику ряду крупних ERP-систем, з якої видно, що перевагу мають системи з можливістю виконання у браузері, оскільки такі системи слабше навантажують комп'ютер клієнта та не потребують інсталяції додаткових програм.

В результаті було розроблено систему з мінімальним навантаженням на комп'ютер клієнта, оскільки вона не потребує навіть конкретної версії браузера чи плагінів.

РОЗДІЛ 1

АВТОМАТИЗОВАНІ СИСТЕМИ ОБЛІКУ

| | | | | | | | | | | | |
|--|------------------|--|--|-------------------------------|---------------|--------------|----------------|----------------------------|--|--|--|
| І.І. Застосування систем обліку | | | | КАФЕДРА КСУ | | | | НАУ 22 02 82 000 ПЗ | | | |
| <i>Виконав</i> | Ковтун В. Є. | | | Автоматизовані системи обліку | <i>Літера</i> | <i>Аркуш</i> | <i>Аркушів</i> | | | | |
| <i>Керівник</i> | Вавіленкова А.І. | | | | <i>Д</i> | | 11 | 68 | | | |
| <i>Консульт.</i> | | | | | 123 СП-436 | | | | | | |
| <i>Н- контроль.</i> | Тупота Є. В. | | | | | | | | | | |
| <i>Зав. кафедри</i> | Литвиненко О. Є. | | | | | | | | | | |

ERP-система (*ERP – enterprise resource planning*) — це комплекс програм для управління підприємством, які працюють на єдиній технологічній платформі, опираються на єдину базу даних (що може також бути інтегрованою з баз даних, отриманих із різних джерел чи систем) і синхронізуються між собою в реальному часі. Цей комплекс не є монолітним: підприємство може впроваджувати та використовувати тільки ті програмні модулі, які йому потрібні.

Частина модулів необхідна практично кожному підприємству, оскільки існують базові бізнес-процеси, такі як управління фінансами, продажами, персоналом, закупівлями тощо. Проте є і спеціалізовані рішення для різних галузей виробництва та сфери послуг. Наприклад, банки можуть користуватися особливими модулями для управління платіжними картками, а фабрики — модулями, які контролюють у цеху робочі процеси відповідно до виду виробництва (дискретне чи безперервне) або технічне обслуговування та ремонт обладнання. Всі ERP-системи, незалежно від того, хто їх розробник, об'єднує спільна архітектура, котру можна описати так (рисунок 1.1.1):

- **Платформа.** Базові можливості та середовище для роботи модулів та компонентів. У код платформи зміни може вносити лише розробник. Користувачі та фахівці з впровадження не мають доступу до цього програмного коду. До складу платформи входять:

1. Ядро. Програмне середовище, в якому буде виконуватись робота, для якої можна писати якісь надбудови та компоненти.

2. Базовий функціонал. Перелік довідників та функцій, без яких не може працювати жодна компанія. Це довідник користувачів із правами доступу, довідник клієнтів, довідник товарів/послуг тощо.

Цей функціонал вбудований у платформу, на відміну від модулів, він не може відключатися.

- **Управління даними.** База даних, у тому числі зберігання та методи обробки (інтерпретації) даних. До цієї категорії входять сховища даних на сервері, програмне забезпечення для роботи з базами даних (SQL або будь-яка альтернатива), інструменти для інтерпретації та обробки даних та відправлення їх до програмних модулів.

- **Модулі.** Компоненти, які підключаються до платформи за необхідності. Всі вони працюють з єдиною базою даних та застосовують базовий функціонал (при необхідності). В іншому модулі працюють незалежно один від одного, можуть "безшовно" підключатися і без проблем відключатися, якщо потреба в них зникла. Така модульна структура – важлива риса ERP-систем. Модулі поділяються, у свою чергу, на кілька типів:

1. Модулі внутрішнього користування. Цей рівень – модулі, що підключаються, які використовуються співробітниками компанії. Це управління складом, виробництво, бухгалтерія, CRM та ін. Модулі можна підключати, відключати, налаштовувати силами фахівців із впровадження. У стандартний набір зазвичай входять - MRP, HR, CRM, Управління постачанням та закупівлями.

2. Модулі роботи із зовнішніми користувачами. Цей тип містить у собі модулі, необхідні для взаємодії із зовнішніми користувачами, потенційними та реальними клієнтами компанії, партнерами, користувачами продукції, постачальниками та покупцями. Це може бути інтернет-магазин, особисті кабінети для постачальників та покупців на корпоративному сайті тощо. Деякі ERP-системи містять готові CMS-системи для створення інтернет-магазину або корпоративного сайту з нуля, інші пропонують лише окремі інструменти «надбудови» до сайту та/або клієнтські додатки (для мобільних телефонів та планшетів).

3. Конектори — готові рішення для зв'язку зі сторонніми програмами. Найчастіше використовують API із ядра платформи. Дозволяють інтегрувати телефонію, налаштувати обмін даними з сайтом або будь-якими програмними продуктами та системами. Конектори призначені тільки для обміну даними та

зазвичай використовуються для обміну даними з EDI, CMS, CAD, BI, OLAP та ін. Тобто з тими системами, які не входять в ERP, але використовуються в компанії.

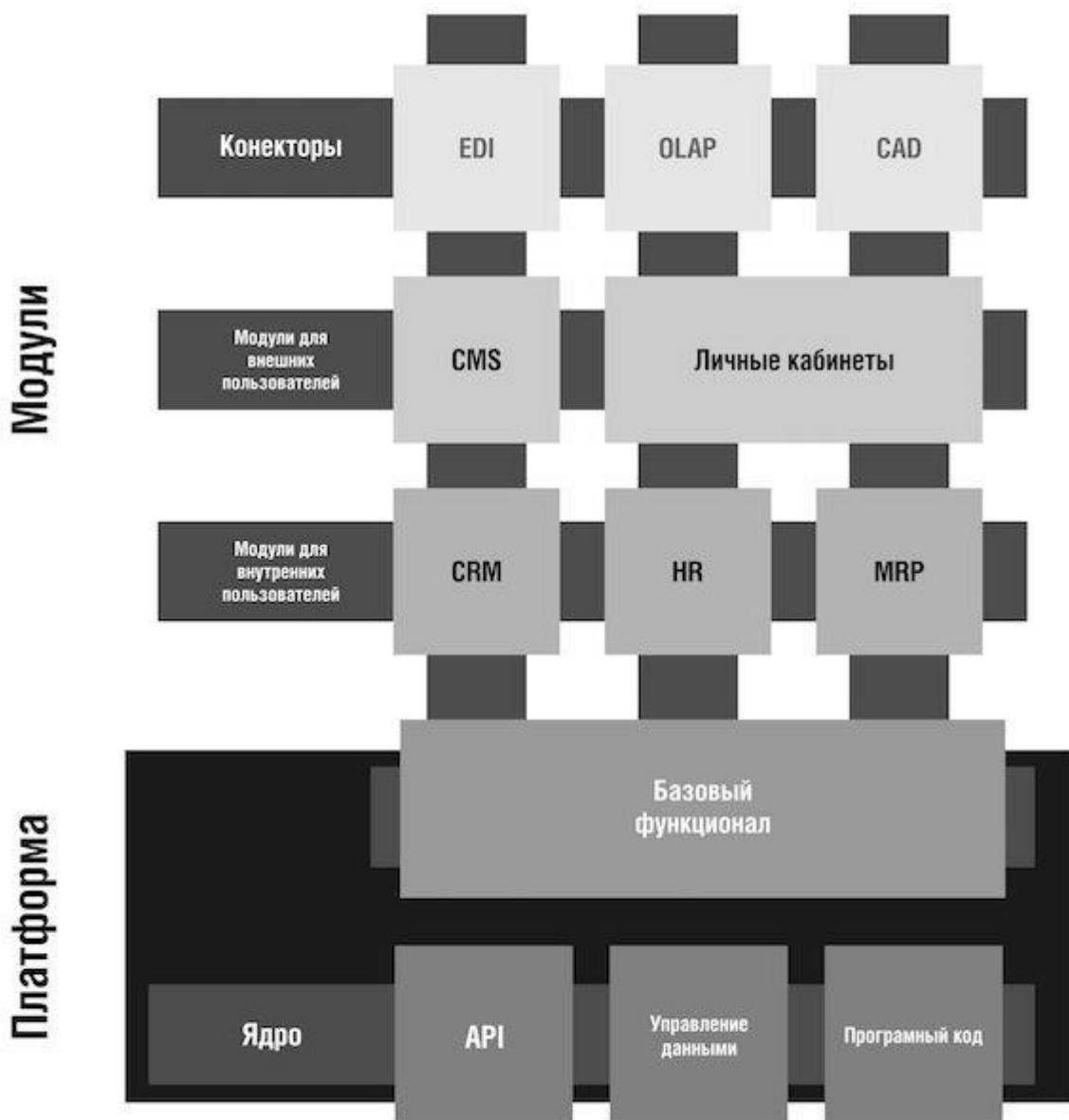


Рисунок 1.1.1. Типова структура ERP

Описана вище структура й у ERP з логічного погляду. У деяких систем немає яскраво вираженої модульності, всі вони вже вбудовані в програму, але використовувати їх можна окремо один від одного при необхідності. Інші називають модулі, що відключаються, підсистемами. А частина ERP-систем виділяють всі модулі в окремі продукти. І пропонують купити ядро, а до нього –

список модулів на вибір. З можливістю в майбутньому купувати та додавати можливості у міру необхідності.

Тож переваги ERP-систем можна сформулювати так:

- Кожен із модулів ERP-системи працює незалежно від інших, він може бути підключений або вимкнений у будь-який момент часу, може просто не використовуватись, при цьому інші модулі зможуть продовжувати роботу. А для підключення того чи іншого модуля немає необхідності вносити зміни до програмного коду ядра, інших модулів.

- При використанні ERP системи для розширення можливостей та підключення нового підрозділу немає необхідності дописувати програмний код, створювати нову частину програми з нуля або займатися складною і часом не дуже зручною інтеграцією різних програм. Досить просто вибрати потрібний модуль, підключити та налаштувати його під потреби бізнесу. В системах ERP практично все, що може знадобитися при автоматизації бізнесу, вже реалізовано. Самописні доробки потрібні у поодиноких випадках. Наприклад, можна використовувати CRM-модуль, але відмовитися від модуля керування персоналом. У міру зростання та розвитку компанії будь-який з модулів без особливих складнощів підключається, налаштовується та програмний продукт продовжує роботу вже з новими функціями.

Системи планування ресурсів підприємства стали невід'ємною складовою сучасного малого, середнього та великого бізнесу. З ростом підприємства неодмінно збільшуватиметься кількість та складність його бізнес-процесів: прийматимуться та звільнятимуться люди, змінюватимуться джерела та об'єми постачання, з'являтимуться нові витрати і т.п.

Раніше ERP-системи були монолітними програмними пакетами, кожен з яких унікально допрацьовувався та не контактував з іншими системами. Такий підхід потребував значних фінансових та часових витрат, адже кожному систему необхідно було щоразу перероблювати під потреби конкретної компанії.

Сучасні ERP-системи вдосканалилися, тепер вони можуть зводити дані всіх процесів в єдину систему. Окрім цього, вони тепер активно обмінюються даними,

в результаті чого ці дані стають доступні не тільки в інших системах, а й у різноманітних сторонніх сервісах, як-то: офісні програми, інтернет-магазини, додатки для роботи з клієнтами і т.п.

Не дивлячись на те, що ідеального ПЗ для всіх бізнес-процесів не існує, ERP зводить їх тісніше, відкриваючи нові можливості з оптимізації. Загалом, сучасна ERP надає такі можливості:

- **Забезпечення оптимальної результативності.** Рішення на основі штучного інтелекту дають аналітику, котра допомагає приймати оптимальні рішення та виявити напрямки подальшого збільшення операційної ефективності.

- **Прискорення операційних рішень.** Наглядне поєднання даних про усі процеси дозволяє співробітникам досягати кращих результатів швидше.

- **Забезпечення гнучкості бізнесу.** Чимало ERP-рішень здатні адаптуватися та масштабуватися відповідно до нагальних потреб. Це дозволяє оперативно підготуватися до збоїв у бізнес-процесах або змінах на ринку, а також швидко реагувати на них.

Найрозповсюдженішими ERP-системами на українському ринку є:

- SyteLine ERP, компанії Infor
- SAP R3, компанії SAP
- Oracle Business Suite, компанії Oracle
- Microsoft Dynamics, компанії Microsoft
- 1С Предприятие v8, компанії 1С
- Галактика ERP, компанії Галактика

Сучасні системи ERP оснащені такими новітніми технологіями, як штучний інтелект, інтернет речей, машинне навчання тощо, завдяки яким управління бізнесом стає ще прозорішим та ефективнішим. Сфери застосування ERP можна побачити на рисунку 1.1.2.

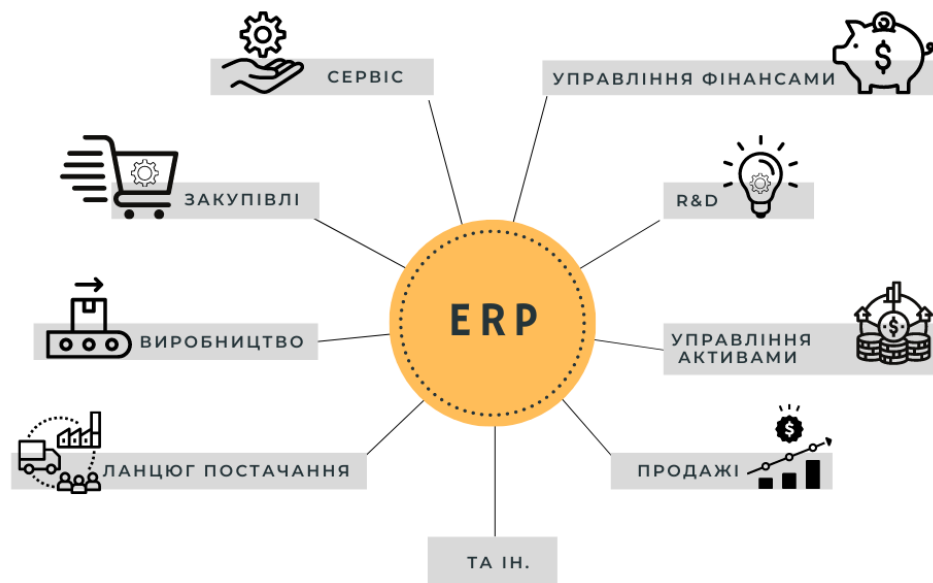


Рисунок 1.1.2. Сфери застосування ERP-систем

1.2. Аналіз використання ERP-систем у сфері автоматизованого обліку транспортних послуг

На сьогоднішній день існують такі програмні рішення:

1.2.1. 1С: Підприємство 8

Транспортна логістика, експедирування та управління автотранспортом КОРП. Дане рішення дозволяє керувати замовленнями, автопарком (як власним так і стороннім) та перевозками. Має наступні можливості[2]:

1) Керування замовленнями на перевозку. У цьому модулі відбувається реєстрація та формування замовлень на перевозку вантажу. Інтерфейс модуля можна побачити на рисунку 1.2.1.1.

Потребность в перевозке M0000000005 от 29.09.2016 13:40:24

Записать и закрыть

Еще ?

Номер: M0000000005 от: 29.09.2016 13:40:24 Детализация: заказ грузовое место товар В работе

Организация: Мегатранс ООО Заказчик: контрагент подразделение Периодическая

Контрагент: ООО "Юниверс" Договор: ООО "Юниверс" в руб. По договору нет долга

Пункты назначения Дополнительно

Добавить Создать заказ

Еще Отображать по тек.пункту:

| FTL | Маршрут | Вид перевозки | Адрес отправле... | Адрес назначения | Проме |
|--------------------------|------------------|---------------------|-------------------|------------------|-------|
| <input type="checkbox"/> | Ялта - Краснодар | Автомобильная гр... | Ялта | Краснодар | |

Даты действия

| Дата отправлен... | Дата приб... |
|-------------------|--------------|
| 02.12.2016 | 02.12.2016 |
| 05.12.2016 | 05.12.2016 |
| 07.12.2016 | 07.12.2016 |

Основное Грузы (1) Товары Плановые доходы Плановые затраты Дополнительно

Параметры отправления

Отправитель: ООО "Юниверс" Получатель: ООО "Юниверс"

Контактное лицо: Кудесников Леонид Петрович Контактное лицо: Кудесников Леонид Петрович

Отправление с: 08:00 по: 09:00 Прибытие с: 16:00 по: 20:00

Стоянка: 0:30 Стоянка: 0:30

Всего: 490,8 км / 8,11 / 1 кг / 0 м3 / мест 0 / на сумму 0 НДС в сумме: 0,00 Всего с НДС: 0,00 р.

Комментарий: Ответственный: ООО "Юниверс"

Рис. 1.2.1.1 Интерфейс модуля керування замовленнями

2) Модуль логіста. Тут відбувається фільтрація замовлень, розподіл замовлень між водіями, відображення поточного розташування машин на карті, комунікація з водіями. Інтерфейс модуля можна побачити на рисунку 1.2.1.2.

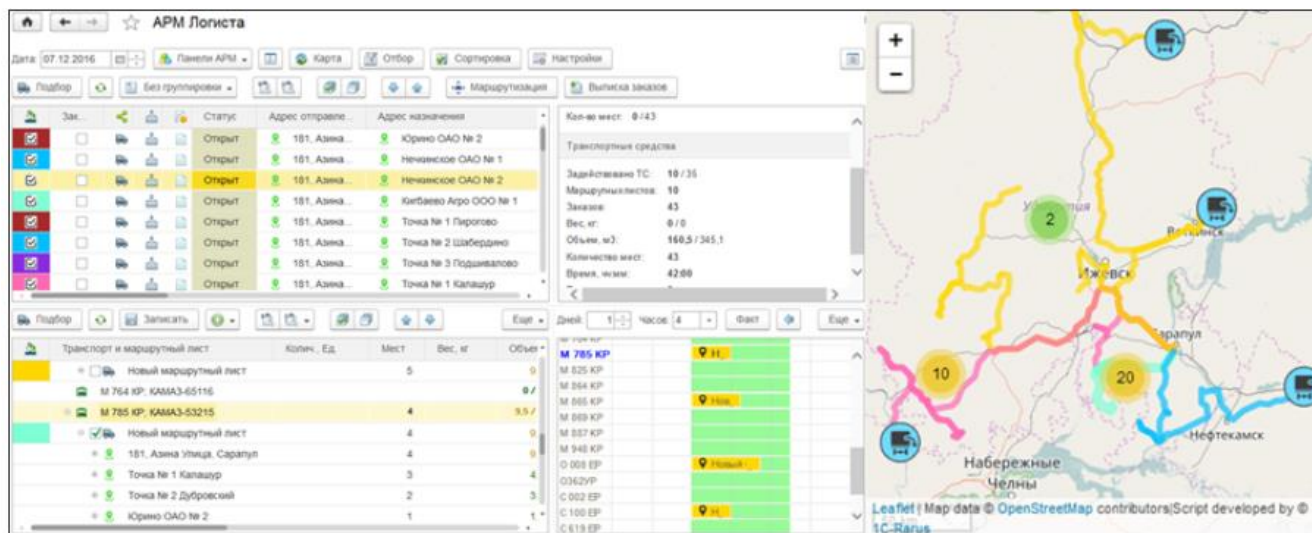


Рис. 1.2.1.2. Інтерфейс модуля керування замовленнями

1.2.2. BAS Управління автотранспортом

Рішення «BAS Управління автотранспортом» призначене для управлінського та оперативного обліку на автотранспортних підприємствах та організаціях, а також транспортних відділах інших підприємств. Працює на основі BAS ERP, інтегровано з «BAS Бухгалтерія»[3]. Програма дозволяє управляти власним автопарком, автоматизувати процес диспетчеризації перевезень та оперативний облік на всіх супутніх ділянках (робота водіїв, паливно-мастильні матеріали, автомобілі та агрегати, ремонти, запчастини та витратні матеріали тощо).

Відмінною особливістю програми є продуманий довідник тарифів, що дозволяє гнучко налаштовувати дію складних тарифів для контрагентів, маршрутів, моделей транспортних засобів та інше. Має такі модулі:

- Управління замовленнями та диспетчеризація. Цей модуль допомагає відпрацьовувати замовлення покупців та формувати завдань на перевезення. Інтерфейс представлено на рисунку 1.2.2.1.

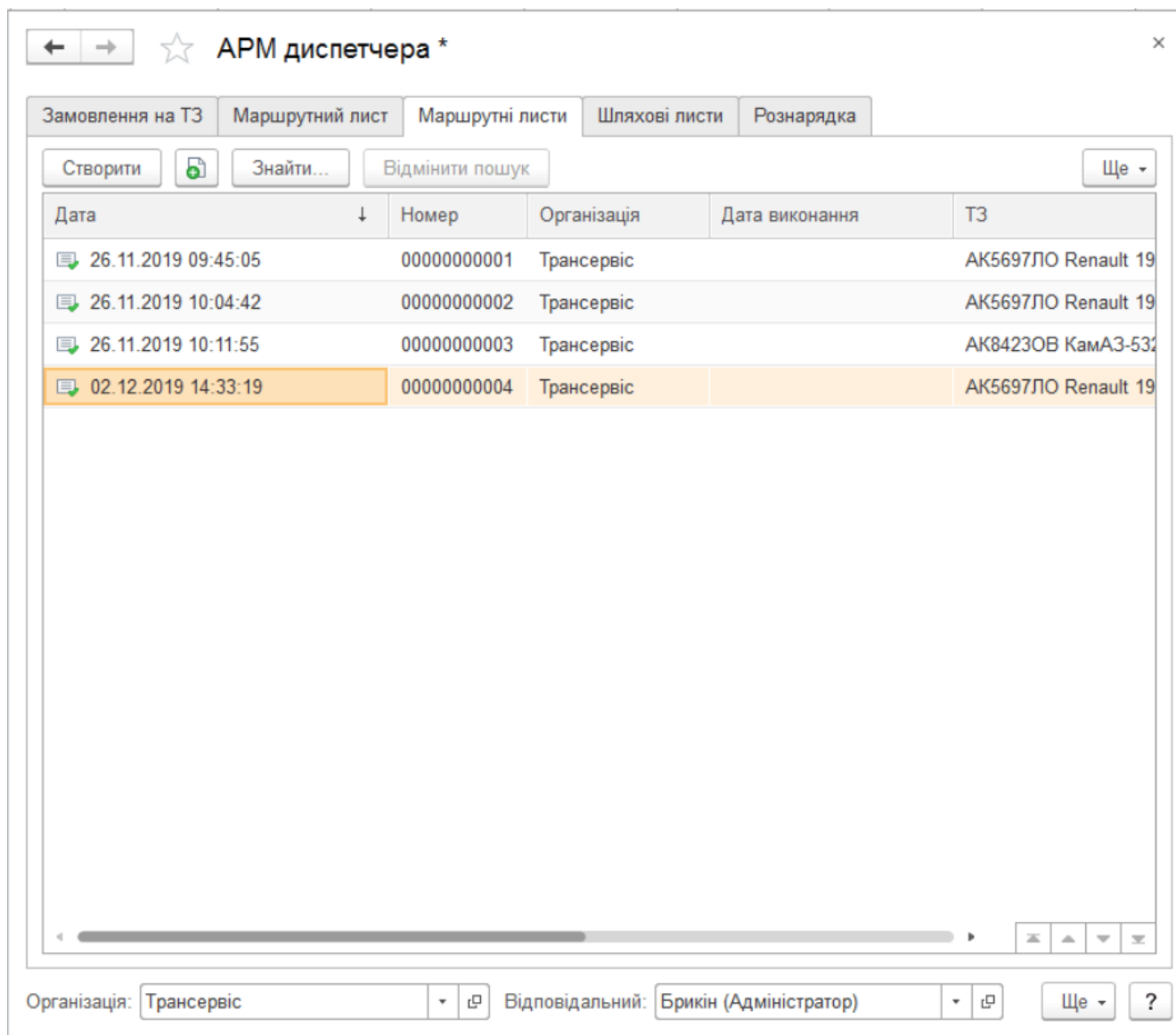


Рис. 1.2.2.1. Інтерфейс модуля диспетчеризації ВАС

– Управління автопарком. Цей модуль дозволяє: вести довідник транспортних засобів, вести облік документів на транспортні засоби, вести облік виробітку транспортних засобів та обладнання. Інтерфейс представлено на рисунку 1.2.2.2.

Renault 19 Europa 1.4 (4L-1,397-75-5M) (Моделі транспортних засобів та устатку... (BAF) [документ] [календар] 31 М М+ М- □ ×

Renault 19 Europa 1.4 (4L-1,397-75-5M) (Моделі транспортних засобів та...

[Головне](#)
[Норми витрати ПММ](#)
[Норми витрати технічних рідин](#)
[Приєднані файли](#)

Найменування: Код:

Входить у групу: ...

Вид моделі ТЗ:

Тип ТЗ: ... Категорія з ЄЕК:

Паливо

Основне паливо: ...

Доп. паливо: ...

Паливний бак: Обсяг бака (л):

Двигун

Тип двигуна: ...

Модель двигуна:

Потужність, л. с.: кВт:

Робочий об'єм: см3

Додаткова інформація

Рис. 1.2.2.2. Інтерфейс модуля управління транспортом

– Облік ремонтів та агрегатів. Цей модуль дає можливість вести облік заявок на ремонт, друкувати накладні для ремонтних листів, вести облік агрегатів. Є можливість скласти рейтинги автомобілів за витратами на ремонт. Інтерфейс можна побачити на рисунку 1.2.2.3.

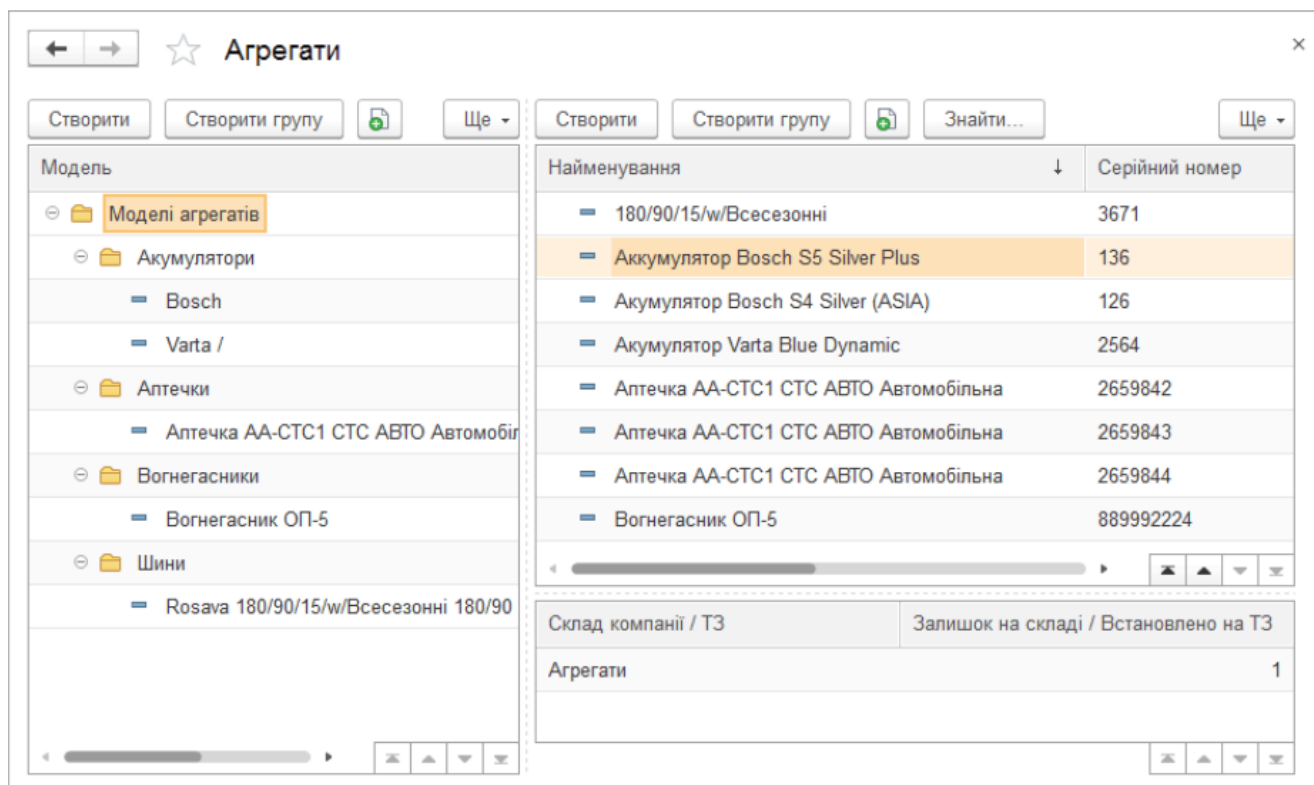


Рис. 1.2.2.3. Інтерфейс модуля обліку ремонтів та агрегатів

1.2.3. Модуль управління транспортом від Jsolutions

Рішення «Система управління автотранспортом» у ERP Jsolutions дозволяє вести облік автотранспорту, здійснювати керування автотранспортним підприємством, компанією і підрозділом з використанням хмарних технологій. Система хмарна, може працювати на будь-якій ОС, здатна взаємодіяти з різними системами управління базами даних, такими як Oracle чи Postgre SQL[4].

Рішення дозволяє: вести облік подорожніх листів, витрат палива, експлуатацію автотранспорту, планувати маршрути транспорту. Приклад інтерфейсу приведено на рисунках 1.2.3.1 та 1.2.3.2.[5]

Автомобілі: Додавання

Основні дані | Технічні характеристики | Додатково | Додаткове обладнання | Норми і тарифи | Документи | Екіпаж | ТО

Гараж

Гаражний номер: 1 Місце зберігання: [Вибір]

Реєстрація

Власник: Прогрес ТОВ Тип: Автомобіль Рік випуску: 2016

Виробник: Volkswagen Модель: Skoda Octavia 1.8 TSI

Державний номер: [Вибір] Дата реєстрації: 10.04.2017

VIN: [Вибір] Номер двигуна: [Вибір]

Номер шасі: [Вибір] Номер рами: [Вибір]

Колір: темно-зелений Об'єм двигуна (куб.см): 1 798 тип: Бензин

Максимальна маса: 1 845 Маса без навантаження: 1 260

Кількість сидячих місць: 5 Кількість стоячих місць: [Вибір]

Зберегти < > Базовий Записати Відмінити ?

Рис. 1.2.3.1. Інтерфейс додавання нової машини у Jsolutions

Рис. 1.2.3.2. Інтерфейс встановлення норм витрат ПММ у Jsolution

1.2.4. Oracle Transportation Management

Рішення «Transportation Management» у Oracle ERP є складовою пакету Oracle Logistics, дозволяє: обробляти заявки та замовлення на перевезення, оптимізувати та моделювати логістичну мережу, автоматизовано виставляти рахунки, управляти автопарком. Також є можливість використання машинного навчання для ефективного розподілення ресурсів та прогнозування часу доставки[6].

Інтерфейс програми можна побачити на рисунку 1.2.4.1.

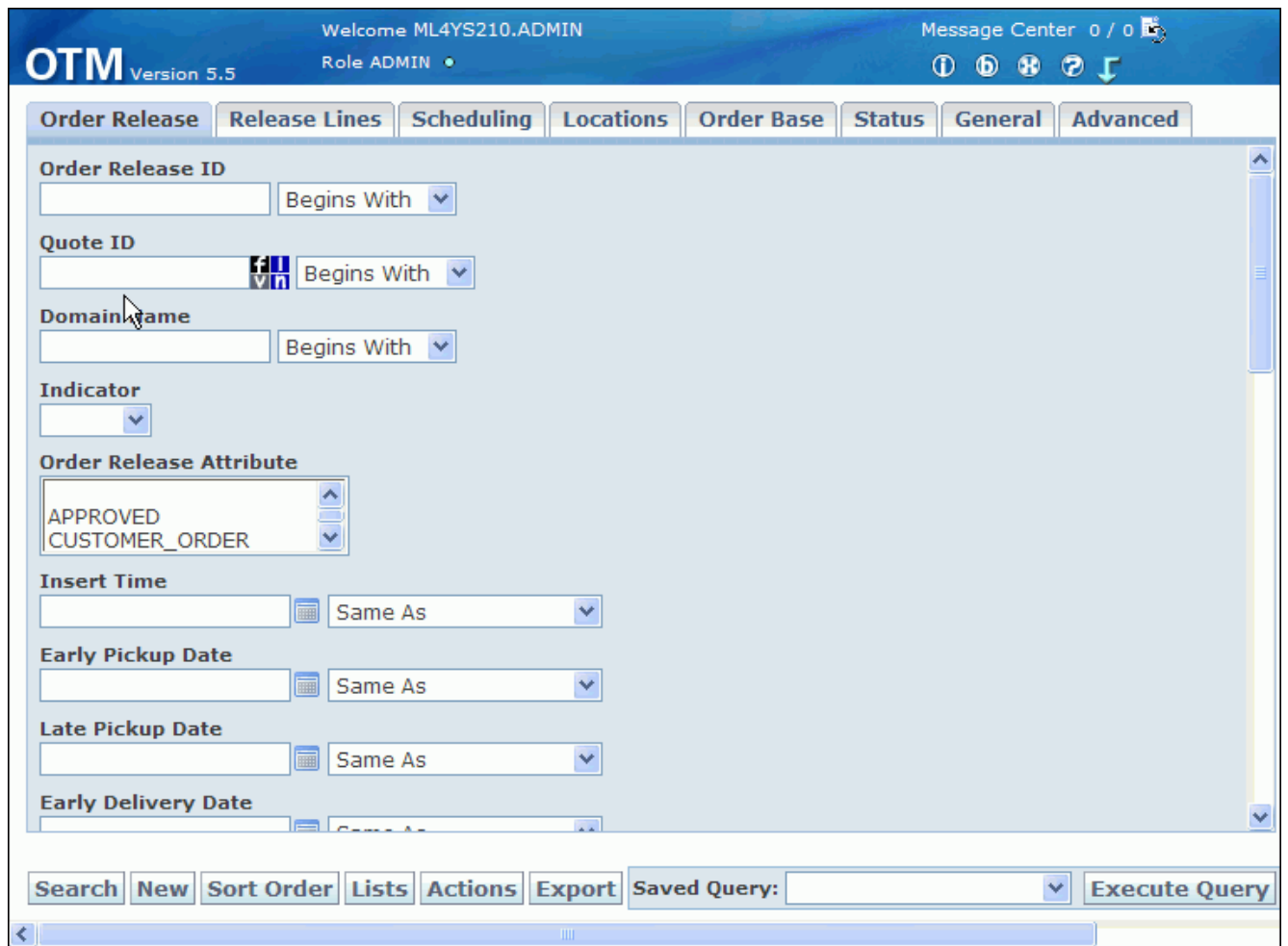


Рис. 1.2.4.1. Інтерфейс Oracle Transportation management

1.2.5. Modalkon – Управління автотранспортом

Рішення «Modalkon – Управління автотранспортом» – це модуль транспортного обліку у ERP Modalkon. Призначений для автоматизації бізнес-процесів управління і оперативного обліку на автотранспортних підприємствах, а також тих, що мають у своєму складі транспортних підрозділів. Функціонал рішення: робота з подорожними листами, зберігання відомостей про транспортні засоби, обробка заявок на перевезення, розрахунок нормативних витрат палива і мастильних матеріалів, списання та ремонт агрегатів[7]. Інтерфейс програми можна побачити на рисунку 1.2.5.1.

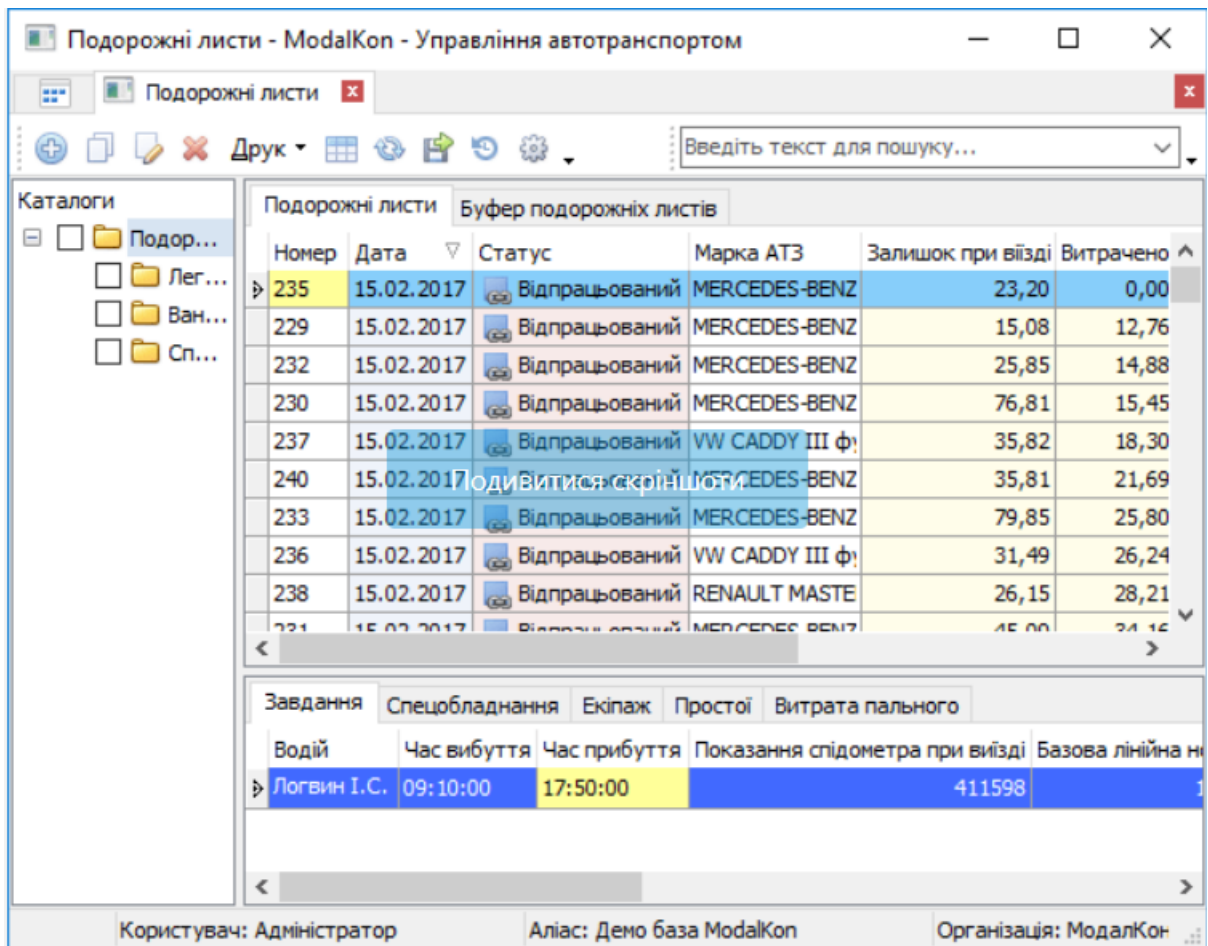


Рис. 1.2.5.1. Інтерфейс ModalKon – Управління автотранспортом

З наявної у публічному доступі інформації було складено порівняльну таблицю 1.2.5.1.

Порівняльна таблиця описаних систем

| | Оцінка системних вимог за 5-бальною шкалою | Ціна | Є своя СУБД | Є Маш.Навч. | Є у "хмарі" |
|----------------|--|-----------------|-------------|-------------|-------------|
| 1С Предприятие | 3 | від 12300 грн. | + | - | - |
| BAS | 3 | від 12300 грн. | - | - | - |
| Jsolutions | 2 | 617 грн./місяць | - | - | + |
| Oracle | 2 | Від 16100\$ | + | + | + |
| Modalcon | 3 | 39280 грн. | - | - | - |

Табл. 1.2.5.1. Порівняльна таблиця описаних систем

Загалом, можна виділити такі пункти:

- Тільки великі системи мають власні СУБД, інтеграція з якими буде найбільша.
- Мінімальні системні вимоги наявні в систем хмарних системах. Єдиною вимогою для клієнтів таких систем є певна версія браузера.
- Для нехмарних систем вимоги для клієнтів більш-менш однакові, для вказаних систем це 4 ГБ оперативної пам'яті та 500 МБ пам'яті на диску.

1.2.6. Висновки до розділу

Сьогодні на українському ринку можна знайти чимало програмних рішень для ведення обліку надання транспортних послуг, причому як іноземних так і вітчизняних. Деякі з них мають “хмарні” опції, як робота в браузері чи зберігання даних на публічних серверах за абонплату (J solutions та Oracle). Інтерфейс рішень приблизно однаковий, як і функціонал. Системні вимоги залежать від версії системи: “хмарна” версія запускатиметься у браузері, а виконуватиметься на сервері, завдяки чому вона буде легшою ніж повноцінний клієнт. В результаті опрацювання першого розділу можна сформулювати постановку задачі дипломного проекту:

- ознайомитися з існуючими автоматизованими системами обліку транспортних послуг;
- вивчити алгоритми реалізації автоматичного обліку;
- дослідити структуру бази даних автоматизованої системи обліку транспортних послуг;
- створити автоматизовану систему обліку транспортних послуг на основі SAP;
- оцінити кількісні показники роботи розробленої системи.

| | | | | | | | |
|---------------------|------------------|--|--|---|---------------|--------------|----------------|
| Кафедра КСУ | | | | РОЗДІЛ НАУ 22 02 82 000 ПЗ | | | |
| <i>Виконав</i> | Ковтун В. Є. | | | Технології та алгоритми SAP для організації роботи автоматизованої системи обліку транспортних послуг | <i>Літера</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| <i>Керівник</i> | Вавіленкова А.І. | | | | Д | 27 | 68 |
| <i>Консульт.</i> | | | | | 123 СП-436 | | |
| <i>Н- контроль.</i> | Тупота Є. В. | | | | | | |
| <i>Зав. кафедри</i> | Литвиненко О. Є. | | | | | | |
| | | | | | | | 26 |

ТЕХНОЛОГІЇ ТА АЛГОРИТМИ SAP ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ТРАНСПОРТНИХ ПОСЛУГ

2.1. Технологія розробки ERP SAP R/3

2.1.1. Архітектура SAP R/3

SAP R/3 (також SAP ERP) – головний продукт компанії SAP, ERP-система, орієнтована на великий та середній бізнес, розробляється та розповсюджується з дев'яностих років. Літера R із позначення R/3 взята зі слова «Realtime», тобто дані оновлюються в режимі реального часу, а цифра 3 означає, що в системі реалізовано архітектуру клієнт-сервер додатків-система управління базами даних (рис.2.1)[8].

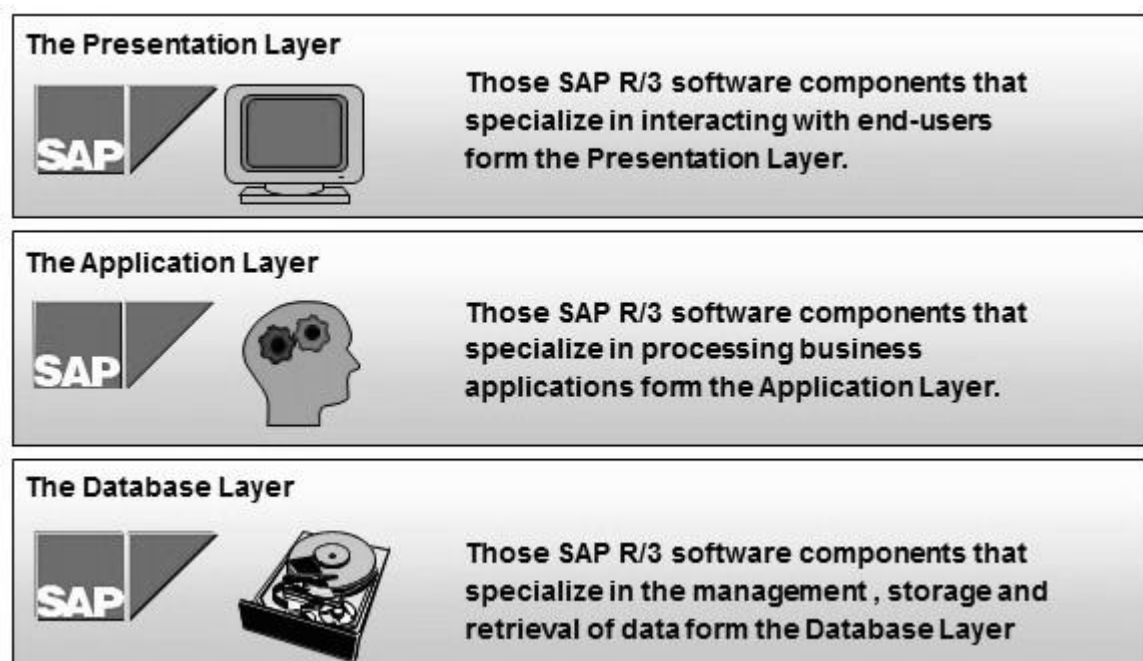


Рис. 2.1. three-tier архітектура SAP

На рис.2.1 можна бачити, що трирівнева архітектура складається з

наступних рівнів[9]:

- Клієнтський рівень – на цьому архітектурному рівні відбувається взаємодія з користувачем за допомогою графічного інтерфейсу користувача (GUI). Сучасні середовища розробки для SAP дозволяють розробляти інтерфейс для персональних комп'ютерів, смартфонів, ноутбуків та інших пристроїв.

- Рівень серверу додатків (також рівень ядра, або базовий рівень) – на цьому архітектурному рівні виконуються програми SAP. Представляє собою спеціалізовані сервери з великою кількістю процесорів та оперативної пам'яті, на котрих розгорнуто систему SAP R/3. Цей рівень виконує роль комунікаторів між клієнтським рівнем та рівнем системи управління базами даних.

- Рівень системи управління базами даних – на цьому архітектурному рівні зберігаються дані (таблиці, програми, структури та інше). Представляє собою системи з великими та швидкими жорсткими дисками. Складається з систем управління базами даних та власне баз даних. SAP R/3 може працювати з більшістю

сучасних систем управління базами даних: Oracle, Microsoft SQL, IBM DB/2, Siebel, Sybase та інші, але також SAP розробила власну систему SAP HANA.

2.1.2. Мова програмування ABAP

ABAP (Advanced Business Application Programming) – пропрієтарна мова програмування, що була розроблена компанією SAP для розробки системи SAP R/2, успадкований в системі SAP R/3. Надає можливість для об'єктно-орієнтованого програмування, синтаксис заснований на використанні ключових слів англійською. В основному, розробка для SAP R/3 проходить саме мовою ABAP. Середовища розробки для цієї мови представлені, як правило, продукцією компанії SAP (наприклад розробка у клієнті SAP GUI), але існують і інтеграції з відкритими середовищами розробки (наприклад можна встановити пакет для розробки на ABAP під назвою ABAP Development Tools для середовища Eclipse).

Програми, написані на АВАР, працюють на рівні серверу додатків під управлінням виконуючої системи, що є частиною ядра SAP. Під час компіляції програма перетворюється на байт-код[10], який потім виконується на віртуальній машині АВАР runtime environment (схему виконання програми можна побачити на рисунку 2.2) . Зберігаються ж програми на рівні системи управління базами даних, а не у вигляді окремих файлів (як у мовах програмування Java чи C++). Через це отримати доступ до програм можна тільки в рамках системи SAP.

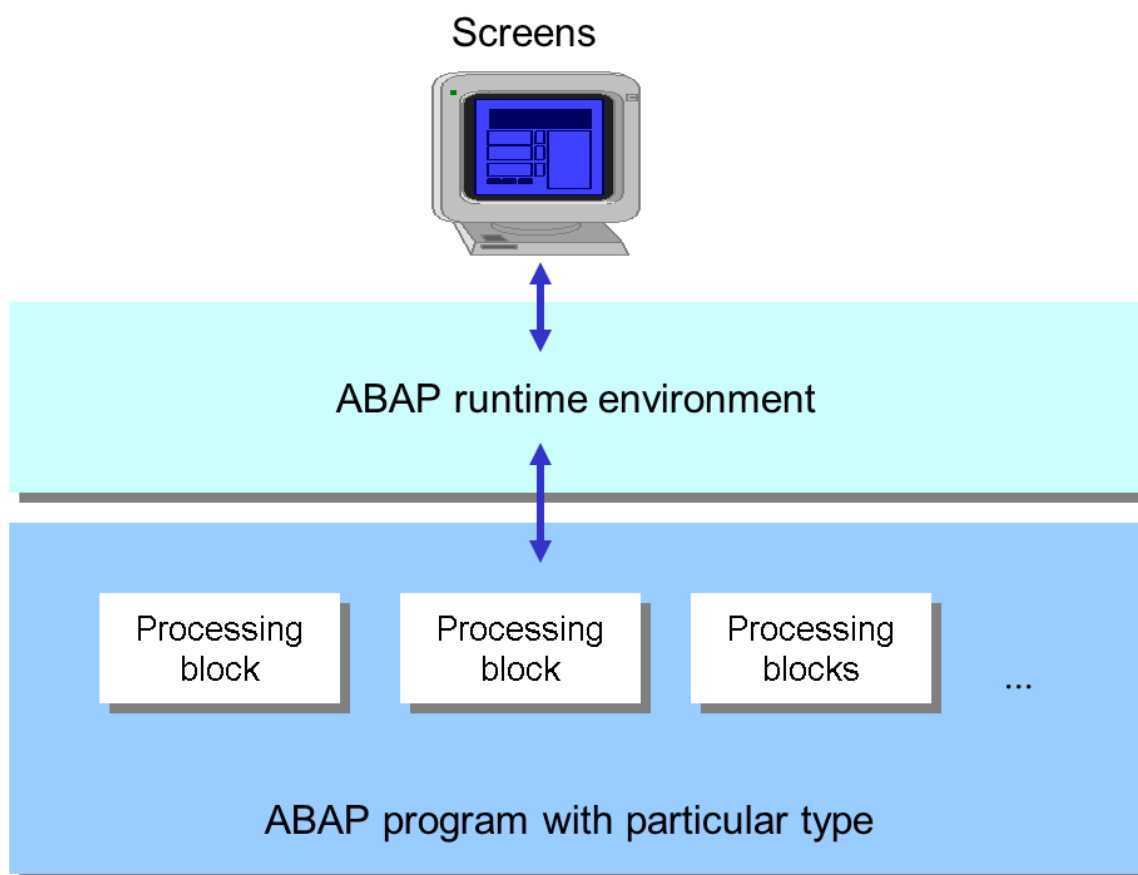


Рис. 2.2. Схема виконання програми на АВАР

АВАР постійно модифікується відділом SAP Standard, що переписує старі програмні модулі у об'єктно-орієнтовному руслі та розробляє нові фреймворки для взаємодії з веб-розробками, кількість котрих значно зросла останнім часом.

2.1.3. Floorplan Manager

Floorplan Manager – це фреймворк для швидкої та простої розробки інтерфейсу користувача, що заснований на технології Web Dynpro ABAP. Представляє собою набір готових типових елементів (списків, вкладок, контейнерів для тексту), що можна розміщати на розбитому на сектори екрані.

Таким чином, за розробником залишається лише реалізація власне логіки взаємодії з інтерфейсом. Елементи беруть дані та опрацьовуються у допоміжному класі, що називається Feeder class.

Результатом роботи з фреймворком є конфігураційний файл, що використовуватиметься стандартним шаблоном сторінки (він називається floorplan) для відрисовки та обробки інтерфейсу[12]. Таких стандартних шаблонів є декілька, вони відрізняються характером взаємодії користувача з інтерфейсом:

- OVP (Overview Page) – звичайний перегляд сторінки, використовується для огляду та редагування інформації з кількох джерел.

- GAF (Guided Activities Floorplan) – чередує кілька сторінок одна за одною, використовується для покрокової роботи з даними (наприклад для створення об'єктів).

- QAF (Quick Activities Floorplan) – містить набір елементів, що у сукупності виконують роль панелі швидкого доступу. Використовується для розробки таких панелей, наприклад у якості головного меню програми.

- OIF (Object Instances Floorplan) – використовується для редагування записів та об'єктів.

Схему роботи фреймворку можна побачити на рисунку 2.3:

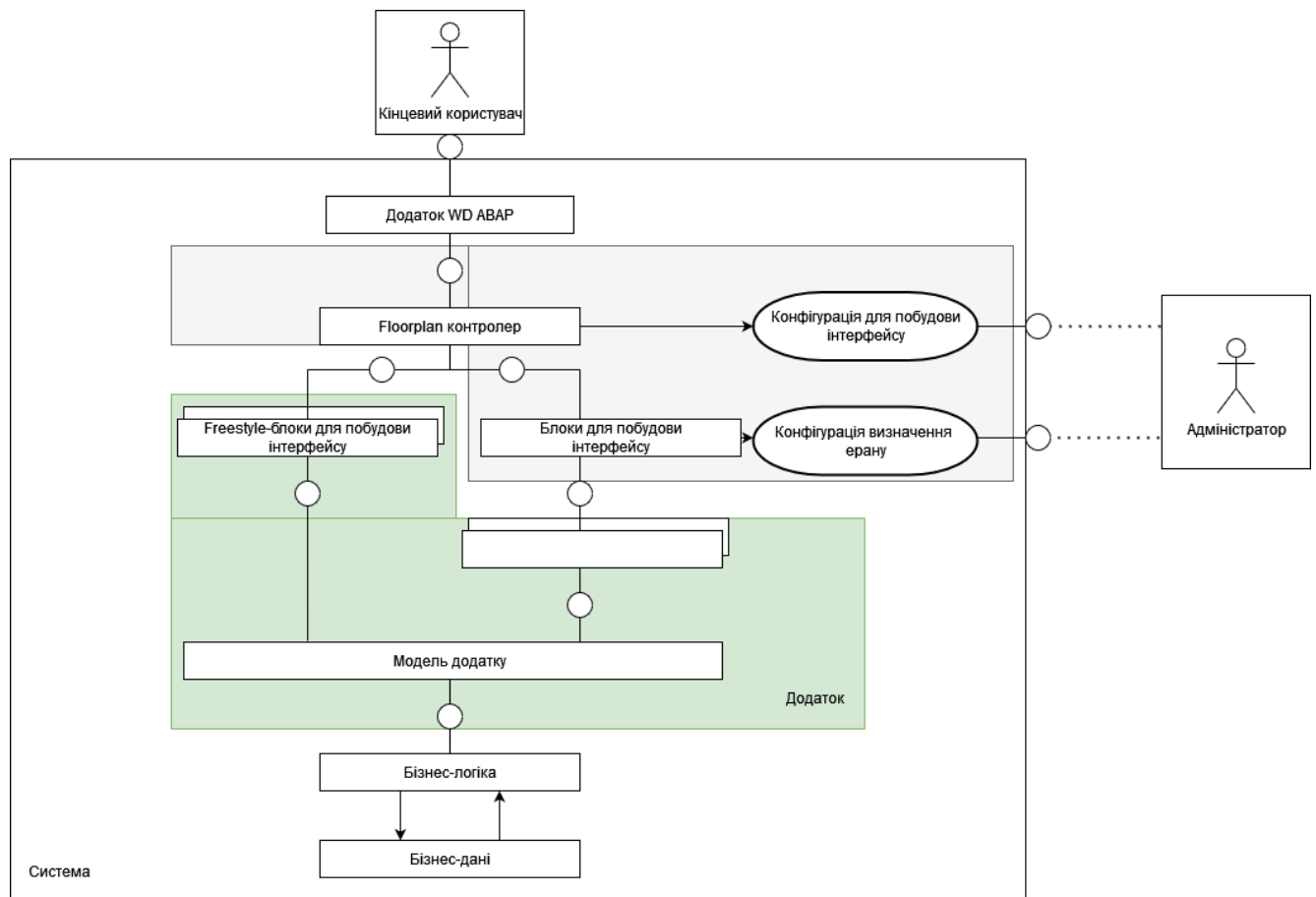


Рис. 2.3. Схема роботи Floorplan Manager

2.1.4. Business Object Processing Framework

BOPF (Business Object Processing Framework) – це фреймворк для спрощеної роботи з даними. Дає можливість створювати особливий тип даних, що називається бізнес-об’єкт. Кожен такий об’єкт складається з мінімум одного запису (node), що представляє певну модель даних. Записи мають ієрархічні відносини, головний запис завжди називається ROOT, він створюється у бізнес-об’єкті за замовчуванням. Модель даних запису генерується фреймворком на основі вказаних розробником структур. В результаті об’єкт зберігатиме інформацію про такі структури:

- Data structure – у цій структурі вказано поля об’єкта, що зберігатимуться в пам’яті, її робить розробник.

– `Transient structure` – у цій структурі вказано поля об'єкта, що не зберігатимуться у пам'яті, а будуть заповнюватись у інший спосіб (наприклад розрахунками), її робить розробник.

– `Combined structure` – поєднує дві попередні структури, це і буде структура даних бізнес-об'єкта, генерується фреймворком.

– `Combined table type` – таблиця, кожним рядком якої є `combined structure`, генерується фреймворком. У ній зберігатимуться дані бізнес-об'єкта.

Таким чином формується модель даних запису. Кожен рядок з `Combined table type` має унікальний ідентифікатор – ключ, що генерується фреймворком. Саме за цими ключами і встановлюються відносини між записами. Окрім моделі даних, кожному запису можна сконфігурувати такі властивості:

– `Associations` – будує логіку взаємовідносин одного запису з іншими. За замовчуванням кожний запис має відношення до головного запису (`TO_ROOT`) та батьківського запису (`TO_PARENT`). За потреби можна створити власне відношення за допомогою візуальних інструментів, прописавши таку логіку у допоміжному класі.

– `Queries` – визначає спосіб видачі даних запису. Таких способів може бути декілька, за замовчуванням записи мають можливість видачі всієї своєї інформації (`SELECT_ALL`). За необхідності новий спосіб можна прописати у допоміжному класі.

– `Actions` – визначає дії, що може обробити запис. Реалізація дії відбувається у допоміжному класі.

– `Validations` – перевірка даних запису (наприклад щоб кількість товару не була від'ємною), їх може бути декілька та вони реалізуються у допоміжних класах. У разі непроходження перевірки під час редагування запису, користувач не зможе зберегти зміни.

Вказані властивості використовуються у роботі, повну структуру бізнес-об'єкта можна побачити на рисунку 2.4.

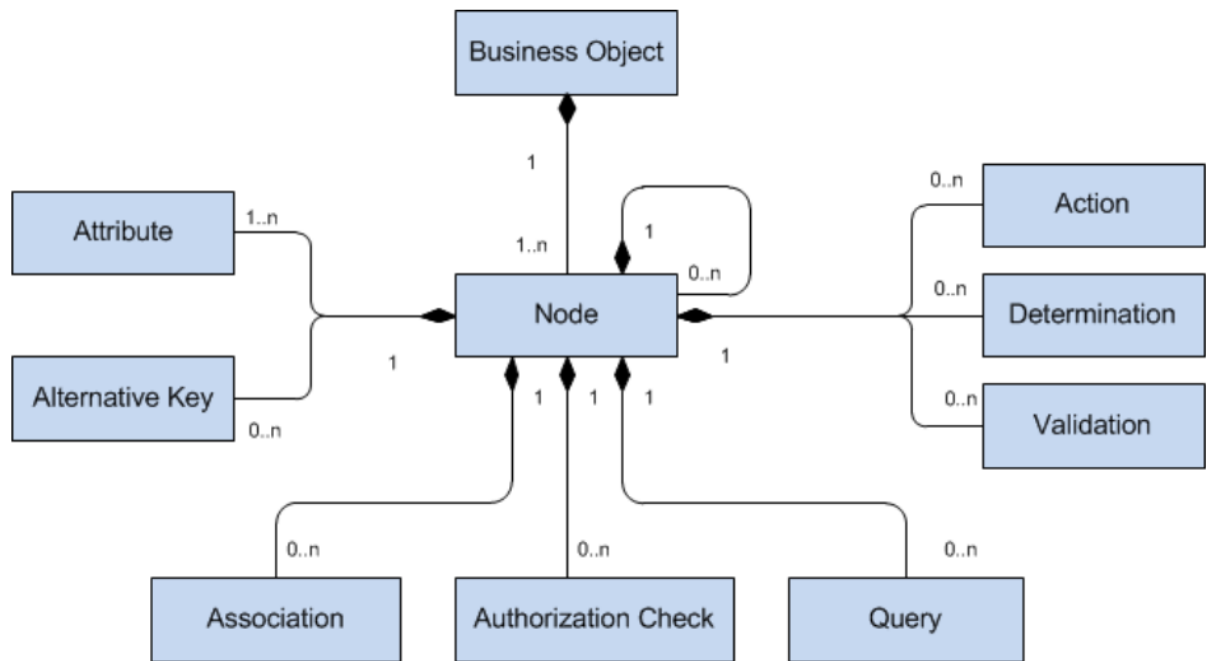


Рис. 2.4. Структура бізнес-об'єкта

2.1.5. FPM-BOPF Integration

FBI (FPM-BOPF Integration) – це засіб поєднання Floorplan Manager та Business Object Processing Framework[13]. Представляє собою набір класів, що можна вказати для елемента у Floorplan Manager в якості Feeder class. Такий клас прийматиме у якості вхідного параметра запис бізнес-об'єкта, і вже має функціонал для CRUD (Create, Read, Update, Delete) дій з об'єктом. Для внесення додаткового функціоналу використовується конфігурації FBI-View, у котрих можна «прив'язати» до допоміжного класу (View exit class). З точки зору логіки виконання, ця можливість дублюватиме властивість запису бізнес-об'єкта actions, але технічно це буде реалізація на різних рівнях, логіка, прописана у View exit class, виконуватиметься тільки для одної програми, в той час як логіка з властивості actions виконуватиметься всюди де буде задіяний бізнес-об'єкт.

2.2. Структура системи автоматизованої систем обліку транспортних послуг на основі SAP

2.2.1. Загальна архітектура R/3

Система R/3 Basis є платформою для всіх інших додатків (фінансовий облік, логістика, управління людськими ресурсами) у системі R/3.

На рисунку 2.2.1.1 зображено узагальнену архітектуру системи R/3[14].

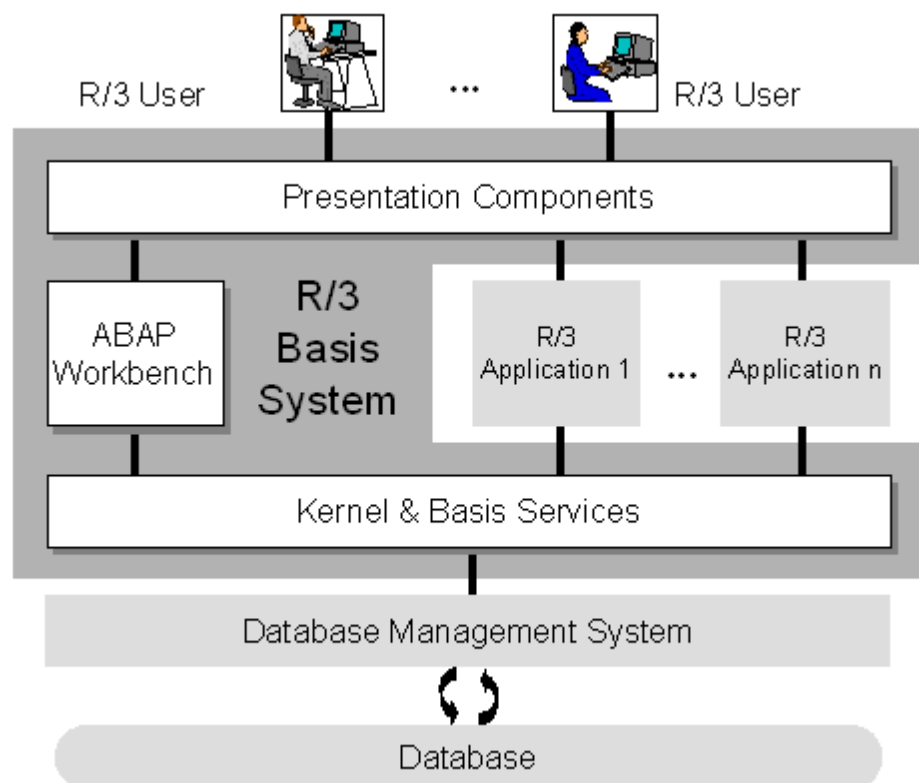


Рисунок 2.2.1.1. Архітектура системи R/3

На рисунку 2.2.1.1 можна бачити, як система R/3 Basis формує центральну платформу в системі R/3. Нижче наведено завдання трьох логічних компонентів системи R/3 Basis:

- **Служби ядра та базові служби** — це середовище виконання для всіх програм R/3, яке залежить від апаратного забезпечення, операційної системи та бази даних. Середовище виконання в основному написано на C і C++. Однак деякі

частини також написані на ABAP. Завдання компонента ядра та базових служб такі:

- **Запуск програм.** Усі програми R/3 працюють на програмних процесорах (віртуальних машинах) у цьому компоненті.

- **Адміністрування користувачів і процесів.** Система R/3 — це багатокористувацьке середовище, і кожен користувач може запускати кілька незалежних програм. Одним словом, цей компонент відповідає за завдання, які зазвичай належать операційній системі. Користувачі входять в систему R/3 і запускають програми в ній. Таким чином, вони не стикаються з фактичною операційною системою хоста. Система R/3 є єдиним користувачем головної операційної системи.

- **Доступ до бази даних.** Кожна система R/3 пов'язана з системою баз даних, що складається з системи керування базами даних (СУБД) і самої бази даних. Програми не взаємодіють безпосередньо з базою даних. Замість цього вони використовують сервіси Basis.

- **Спілкування.** Програми R/3 можуть взаємодіяти з іншими системами R/3 і з системами, які не належать до SAP. Також можливий доступ до програм R/3 із зовнішніх систем за допомогою інтерфейсу BAPI. Служби, необхідні для зв'язку, є частиною компонента ядра та базових служб.

- **Моніторинг та адміністрування системи.** Компонент містить програми, які дозволяють контролювати та керувати системою R/3 під час її роботи, а також змінювати параметри її виконання.

- **Компонент ABAP Workbench** — це повноцінне середовище розробки програм на мові ABAP. Він повністю інтегрований в систему R/3 Basis і, як і інші програми R/3, сам написаний на ABAP.

- **Рівень презентації.** Рівень презентації містить програмні компоненти, які складають SAP GUI (графічний інтерфейс користувача). Цей рівень є інтерфейсом між системою R/3 та її користувачами. Система R/3 використовує SAP GUI для забезпечення інтуїтивно зрозумілого графічного інтерфейсу користувача для введення та відображення даних. Рівень презентації надсилає

введені дані користувача на сервер додатків і отримує від нього дані для відображення. Поки запущено компонент SAP GUI, він залишається пов'язаним із термінальним сеансом користувача в системі R/3. Це програмно-орієнтоване представлення можна розширити, включивши в нього інші рівні, такі як сервер транзакцій Інтернету (ITS).

Компоненти презентації відповідають за взаємодію між системою R/3 і користувачем, а також за інтеграцію елементів робочого столу (таких як обробка текстів і електронні таблиці).

Система R/3 Basis є багаторівневою клієнт-серверною системою. Окремі програмні компоненти розташовуються за рівнями і функціонують залежно від їхнього положення як клієнт для компонентів нижче або сервер для компонентів над ними.

SAP GUI і сервери додатків розроблені таким чином, щоб обсяг даних, що транспортується між двома рівнями, був мінімальним. Це означає, що компоненти рівня презентації можна використовувати навіть на хостах, які мають повільні з'єднання з серверами додатків, які знаходяться далеко. Система має високу масштабованість завдяки тому, що програмні компоненти системи R/3 можуть бути розподілені практично в будь-якій конфігурації між різними хостами.

- **Рівень бази даних.** Рівень бази даних складається з центральної системи бази даних, що містить усі дані в системі R/3. Система баз даних складається з двох компонентів - системи управління базами даних (СУБД) і самої бази даних. Система R/3 підтримує системи баз даних від різних постачальників: ADABAS D, DB2/400 (на AS/400), DB2/Common Server, DB2/MVS, INFORMIX, Microsoft SQL Server, ORACLE і ORACLE Parallel Server . База даних містить не тільки основні дані та дані транзакцій з бізнес-програм, там зберігаються всі дані для всієї системи R/3. Наприклад, база даних містить дані керування та налаштування, які визначають, як працюватиме система R/3. Він також містить програмний код для програм.

Підключення до бази даних. Під час запуску системи R/3, кожен сервер додатків реєструє свої робочі процеси на рівні бази даних і отримує один

виділений канал для кожного. Під час роботи системи кожен робочий процес є користувачем (клієнтом) системи (сервера) бази даних. Перепризначити канал бази даних від одного робочого процесу до іншого не можна. З цієї причини робочий процес може вносити зміни до бази даних лише в межах однієї логічної одиниці роботи бази даних (LUW). LUW — це нерозривна послідовність операцій з базою даних. Це має важливі наслідки для моделі програмування.

Рівень додатків. Рівень програми складається з одного або кількох серверів додатків і сервера повідомлень. Кожен сервер додатків містить набір служб, які використовуються для роботи системи R/3. Теоретично, для запуску системи R/3 достатньо лише одного серверу додатків. На практиці послуги розподіляються між більш ніж одним сервером додатків. Сервер повідомлень відповідає за зв'язок між серверами програм. Він передає запити від одного сервера додатків до іншого. Він також містить інформацію про групи серверів додатків і поточне навантаження в них. Він використовує цю інформацію, щоб вибрати відповідний сервер, коли користувач входить в систему. Додатки ж складаються з програмного коду, визначень екрана, меню, функціональних модулів та різних інших компонентів. Вони зберігаються в спеціальному розділі бази даних, який називається Repository R/3, і відповідно називаються об'єктами Repository.

Сервери додатків. Програми R/3 запускаються на серверах додатків. Вони є важливим компонентом системи R/3. Рівень додатків системи R/3 складається з серверів додатків і сервера повідомлень. Прикладні програми в системі R/3 запускаються на серверах додатків. Сервери додатків спілкуються з компонентами презентації, базою даних, а також один з одним за допомогою сервера повідомлень.

Компоненти серверу додатків:

- **Робочі процеси.** Сервер додатків містить робочі процеси, які є компонентами, які можуть запускати програму. Кожен робочий процес пов'язаний з областю пам'яті, що містить контекст програми, що запускається. Контекст містить поточні дані для прикладної програми.

- **Диспетчер.** Кожен сервер додатків містить диспетчера. Диспетчер є сполучною ланкою між робочими процесами та користувачами, що заходять на сервер додатків. Його завдання — отримувати запити на кроки діалогу від SAP GUI та направляти їх у робочий процес. Таким же чином він спрямовує вихідний екран, отриманий у результаті діалогового кроку, назад до відповідного користувача.

- **Шлюз.** Кожен сервер додатків містить шлюз. Це інтерфейс для протоколів зв'язку R/3 (RFC, CPI/C). Він може спілкуватися з іншими серверами додатків у тій самій системі R/3, з іншими системами R/3, із системами R/2 або з системами, які не належать до SAP. Фіксована кількість робочих процесів і диспетчеризація кроків діалогу веде до оптимального використання пам'яті, оскільки це означає, що певні компоненти та області пам'яті робочого процесу незалежні від програми та їх можна використовувати повторно. Той факт, що окремі робочі процеси працюють незалежно, робить їх придатними для багатопроцесорної архітектури.

- **Спільна пам'ять.** Усі робочі процеси на сервері додатків використовують загальну область основної пам'яті, що називається спільною пам'яттю, для збереження контекстів або локальної буферизації постійних даних. Ресурси, які використовують усі робочі процеси (наприклад, програми та вміст таблиць), містяться в спільній пам'яті. Управління пам'яттю в системі R/3 гарантує, що робочі процеси завжди звертаються до правильного контексту, тобто даних, що стосуються поточного стану програми, яка виконується. Процес відображення проектує необхідний контекст для кроку діалогу зі спільної пам'яті на адресу відповідного робочого процесу. Це зводить фактичне копіювання до мінімуму. Локальна буферизація даних у спільній пам'яті сервера додатків зменшує кількість необхідних читань бази даних. Це скорочує час доступу до програми.

2.3. Алгоритм роботи автоматизованої системи обліку транспортних послуг

Етапи функціонування розробленої автоматизованої системи обліку транспортних послуг, можна побачити на схемі алгоритму (рис. 2.3.1).

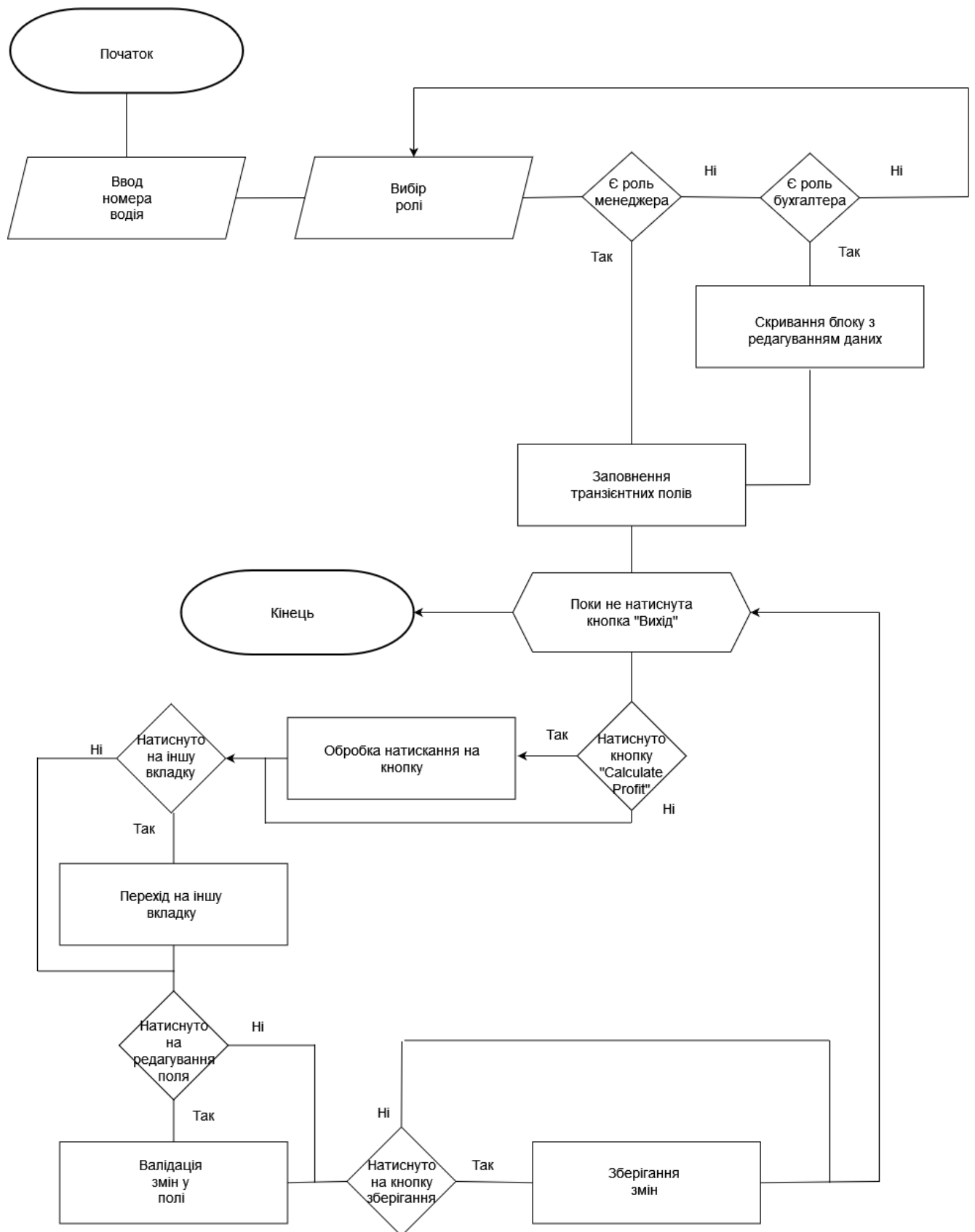


Рисунок 2.3.1. Загальний алгоритм роботи програми

Зі схеми 2.3.1 видно, що алгоритм складається з таких кроків:

1. Ввод ідентифікаційного номера робітника транспортного підприємства.
2. Вибір ролі. Тут на вибір дається дві ролі: бухгалтер (має доступ тільки до дерева об'єктів робітника та блоку з розрахунками) та менеджер (має доступ до всіх блоків).

Вигляд, орієнтований на користувача. На рисунку 2.3.2 представлений орієнтований на користувача вигляд системи R/3:

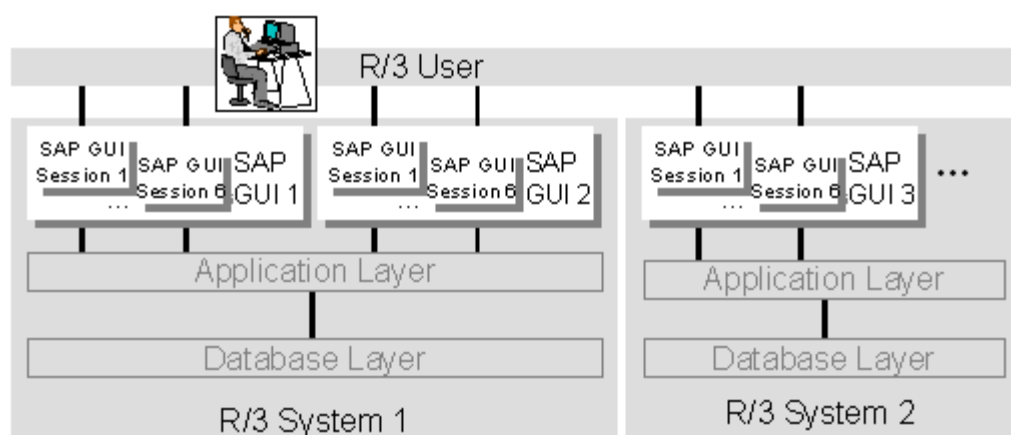


Рисунок 2.3.2. Вигляд системи R/3, орієнтований на користувача

Для користувача видимими компонентами системи R/3 є ті, що відображаються у вигляді вікна на екрані. Вікна генеруються рівнем презентації системи R/3 і є частиною системи R/3 Basis. Перш ніж користувач увійде в систему R/3, він або вона повинні запуснути утиліту під назвою SAP Logon,. У SAP Logon користувач вибирає одну з доступних систем R/3. Потім програма підключається до сервера повідомлень цієї системи і отримує адресу відповідного (найчастіше використовуваного) сервера додатків. Потім запускається SAP GUI, підключений до цього сервера програм. Тоді програма входу в систему SAP більше не потрібна

для цього підключення SAP GUI запускає екран входу. Після того, як користувач успішно увійшов у систему, буде відображено початковий екран системи R/3 на екрані. У SAP GUI вікно R/3 представлено як сеанс. Після входу в систему користувач може відкрити до п'яти додаткових сеансів (вікна R/3) в рамках

одного SAP GUI. Вони поведуться майже як незалежні SAP GUI. Різні сеанси дозволяють запускати різні програми паралельно, незалежно один від одного. Протягом сеансу користувач може запускати програми, які самі викликають інші вікна (наприклад, діалогові та графічні вікна). Ці вікна не є незалежними - вони належать до сеансу, з якого були викликані. Користувач може відкрити інший SAP GUI, використовуючи SAP Logon, щоб увійти в ту ж систему або іншу систему R/3. Окремі SAP GUI та відповідні термінальні сеанси R/3 є повністю незалежними. Це означає, що на настільному комп'ютері можна відкрити кілька систем R/3 одночасно.

Діалогові кроки диспетчеризації. Кількість користувачів, які ввійшли на сервер додатків, часто у багато разів перевищує кількість доступних робочих процесів. Крім того, кожен користувач може запускати кілька програм одночасно. Диспетчер має важливе завдання розподілити всі кроки діалогу між робочими процесами на сервері додатків. На рисунку 2.3.3 наведено приклад такого розподілу.

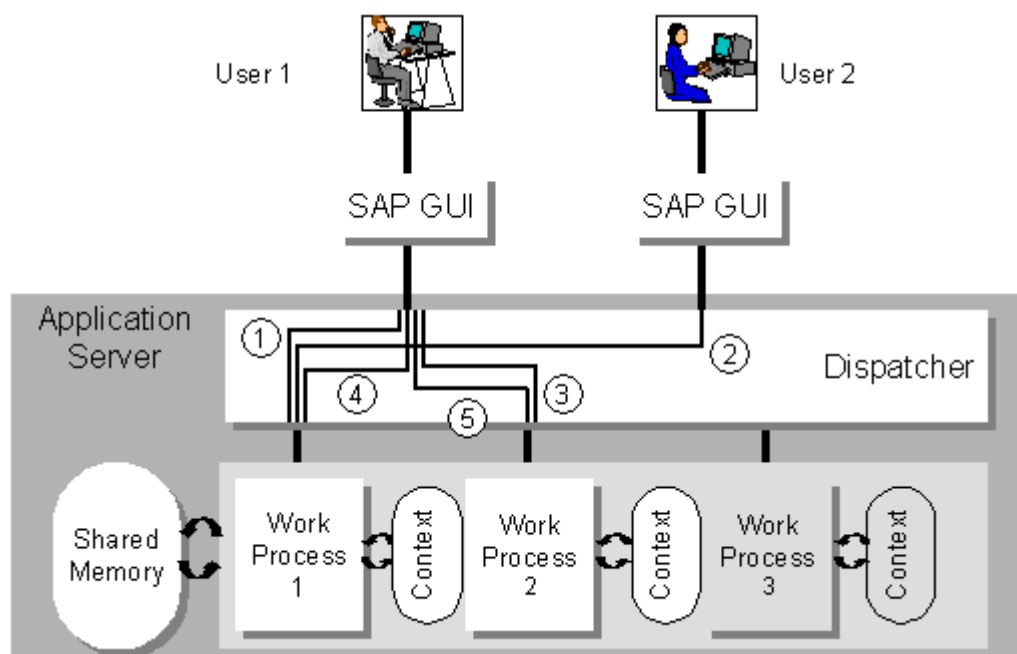


Рисунок 2.3.3. Розподіл ресурсів диспетчером на сервері додатків

На рисунку 2.3.3. представлена наступна послідовність дій:

1) Диспетчер отримує запит на виконання кроку діалогу від користувача 1 і направляє його на робочий процес 1, який виявляється вільним. Робочий процес звертається до контексту прикладної програми (у спільній пам'яті) і виконує крок діалогу. Потім він знову стає вільним.

2) Диспетчер отримує запит на виконання кроку діалогу від користувача 2 і направляє його на робочий процес 1, який тепер знову вільний. Робочий процес виконує крок діалогу, як у пункті 1.

3) Поки робочий процес 1 ще працює, диспетчер отримує подальший запит від користувача 1 і направляє його на робочий процес 2, що вільний.

4) Після того, як робочі процеси 1 і 2 закінчили обробку своїх діалогових кроків, диспетчер отримує ще один запит від користувача 1 і направляє його до робочого процесу 1, який знову вільний.

5) Поки робочий процес 1 ще працює, диспетчер отримує подальший запит від користувача 2 і направляє його на вільний робочий процес 2.

З цього видно, що:

- Крок діалогу з програми призначається окремому робочому процесу для виконання.
- Окремі діалогові кроки програми можуть виконуватися в різних робочих процесах, а контекст програми має розглядатися для кожного нового робочого процесу.
- Робочий процес може виконувати кроки діалогу різних програм від різних користувачів.

Приклад не показує, що диспетчер намагається розподілити запити на робочі процеси таким чином, щоб той самий робочий процес використовувався якомога частіше для послідовних кроків діалогу в програмі. Це корисно, оскільки зберігає контекст програми, до якого потрібно звертатися щоразу, коли виконується крок діалогу.

3. Якщо користувач обрав роль менеджера – відбувається перевірка, чи дійсно його акаунт містить таку роль. Якщо такої ролі акаунт не містить – система не пропускає користувача.

4. Якщо користувач обрав роль бухгалтера – відбувається перевірка, чи дійсно його акаунт містить таку роль. Якщо такої ролі акаунт не містить – система не пропускає користувача.

5. Якщо користувач обрав роль бухгалтера та дійсно має таку роль – система «ховає» блок з редагуванням даних.

6. Відбувається заповнення транзійентних полів, наприклад – зарплат водіїв, що розраховуються відносно проїждженого шляху.

АВАР-процесор. Фактична логіка обробки прикладної програми написана АВАР - власною мовою програмування SAP. Процесор АВАР виконує логіку обробки прикладної програми та взаємодіє з інтерфейсом бази даних. Екранний процесор повідомляє процесору АВАР, який модуль логіки екранного потоку слід обробити наступним. Рисунок 2.3.4 ілюструє взаємодію між екраном і процесорами АВАР під час запуску прикладної програми.

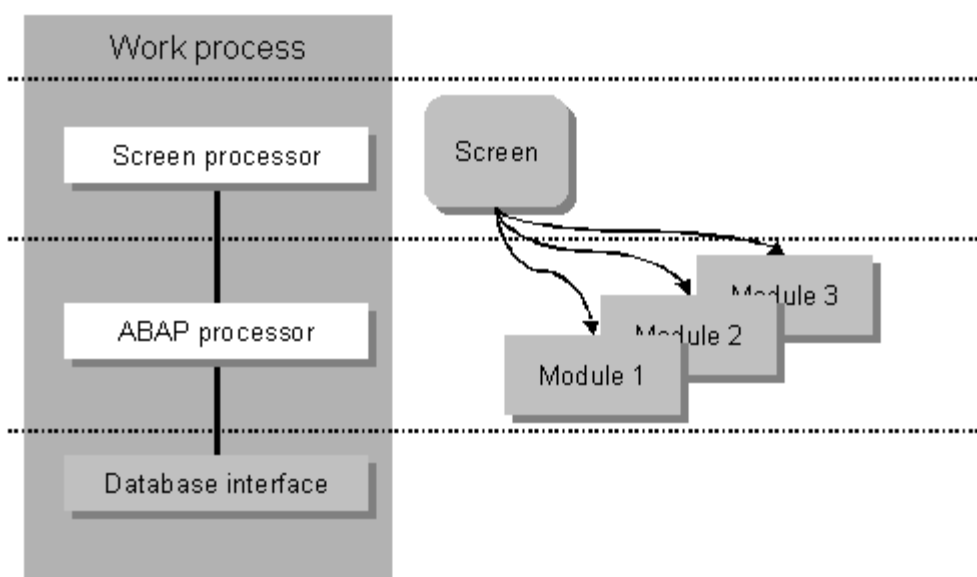


Рисунок 2.3.4. Робота екрану і процесора АВАР у прикладній програмі

Інтерфейс бази даних. Інтерфейс бази даних надає такі можливості:

- Встановлення та припинення зв'язків між робочим процесом і базою даних.
- Доступ до таблиць бази даних
- Доступ до об'єктів репозиторію R/3 (програми АВАР, екрани тощо)
- Доступ до інформації каталогу (словник АВАР)
- Контроль транзакцій (обробка фіксації та відкату)
- Адміністрування буфера таблиці на сервері додатків.

На рисунку 2.3.5 показано окремі компоненти інтерфейсу бази даних:

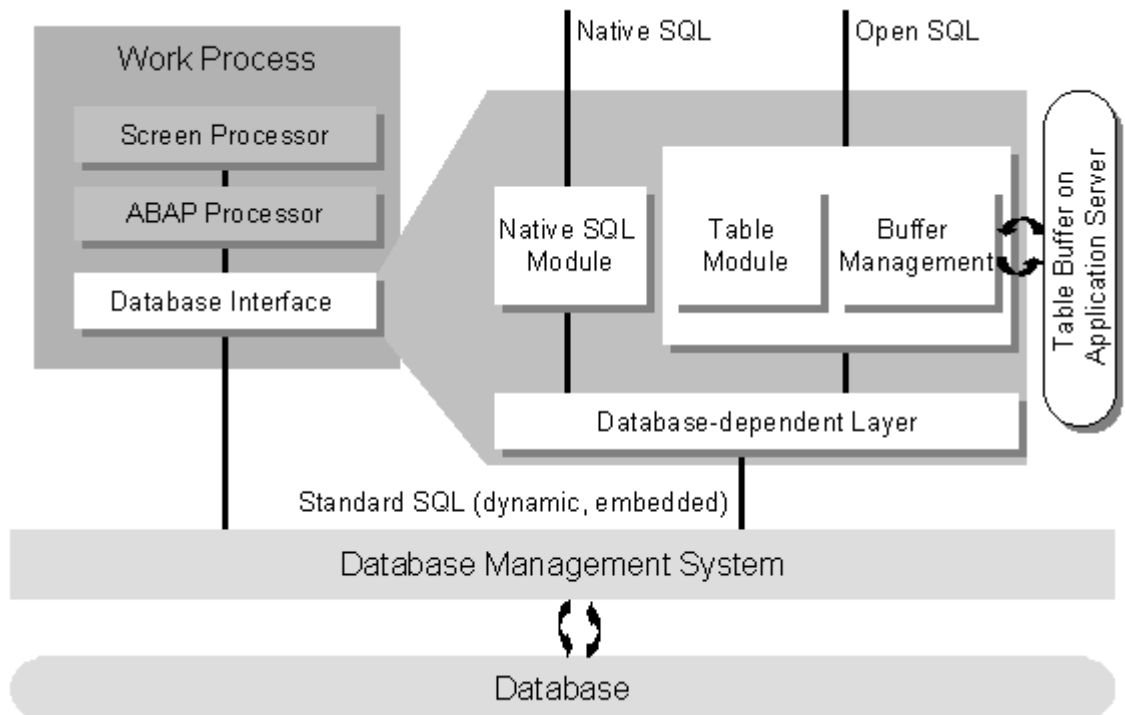


Рисунок 2.3.5. Компоненти інтерфейсу бази даних

Ієрархію node-об'єктів, спроектованих в програмному середовищі *SAP GUI* представлена на рис. 2.3.6. Модель містить в собі запис водія (ROOT), запис закріплених за ним машин (TRUCK), запис закріплених за машиною вантажів (CARRIAGE) та запис з обслуговування машини (SERVICE).

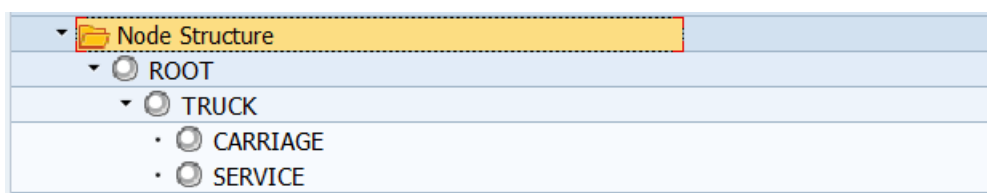


Рис. 2.3.6. Ієрархія node-об'єктів

Після створення Node-структури для BOPF-об'єкта необхідно створити відповідну модель даних. Кінцева структура запису водія представлена на рис. 2.3.8, кінцева структура запису машини представлена на рис. 2.3.7, кінцева структура запису вантажу представлена на рис. 2.3.9, кінцева структура запису обслуговування представлена на рис. 2.3.10.

Структура: ZVKO_S_DRIVERS_C Активно
 Краткое описание: Driver node - #GENERATED#

Свойства | Компоненты | Справка/проверка знач. ввода | Поля валюты/количества

Встроенный тип 1 / 13

| Компонент | Вид типизации | Тип компонента | Тип данных | Длина | ДесР... | Краткое описание | Группа |
|------------|---------------|-----------------|------------|-------|---------|------------------|--------------------|
| .INCLUDE | Тип | /BOBF/S_FRW_KEY | RAW | 0 | 0 | | |
| KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| PARENT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| ROOT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| .INCLUDE | Тип | ZVKO_S_DRIVERS | RAW | 0 | 0 | | NODE_DATA |
| ID | Тип | ZVKO_DRIVER_ID | CHAR | 20 | 0 | | |
| NAME | Тип | ZVKO_EMPLOYEE_N | CHAR | 20 | 0 | | |
| SURNAME | Тип | ZVKO_EMPLOYEE_S | CHAR | 25 | 0 | | |
| MILEAGE | Тип | ZVKO_TRUCKS_MIL | INT2 | 5 | 0 | | |
| OCCUPATION | Тип | ZVKO_DRIVER_OCC | CHAR | 40 | 0 | | |
| .INCLUDE | Тип | ZVKO_TS_DRIVERS | RAW | 0 | 0 | | TRANSIENT_NODE_... |
| SALARY | Тип | ZVKO_CARRIAGE_C | CURR | 10 | 2 | | |
| CURRENCY | Тип | ZVKO_CARRIAGE_C | CURY | 5 | 0 | | |

Рис. 2.3.7. Кінцева структура запису водія

Структура: ZVKO_S_TRUCKS_C Активно
 Краткое описание: TRUCK NODE - #GENERATED#

Свойства | Компоненты | Справка/проверка знач. ввода | Поля валюты/количества

Встроенный тип 1 / 10

| Компонент | Вид типизации | Тип компонента | Тип данных | Длина | ДесР... | Краткое описание | Группа |
|-------------|---------------|-----------------|------------|-------|---------|------------------|-----------|
| .INCLUDE | Тип | /BOBF/S_FRW_KEY | RAW | 0 | 0 | | |
| KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| PARENT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| ROOT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| .INCLUDE | Тип | ZVKO_S_TRUCKS | RAW | 0 | 0 | | NODE_DATA |
| ID | Тип | ZVKO_CARS_ID | CHAR | 20 | 0 | | |
| CONSUMPTION | Тип | ZVKO_TRUCKS_CON | INT1 | 3 | 0 | | |
| LICPLATE | Тип | ZVKO_CARS_LICPL | CHAR | 10 | 0 | | |
| DRIVER_ID | Тип | ZVKO_DRIVER_ID | CHAR | 20 | 0 | | |
| MILEAGE | Тип | ZVKO_TRUCKS_MIL | INT2 | 5 | 0 | | |

Рис. 2.3.8. Кінцева структура запису машини

| Компонент | Вид типизации | Тип компонента | Тип данных | Длина | ДесПа... | Краткое описание | Группа |
|------------|---------------|-----------------|------------|-------|----------|------------------|-----------------|
| .INCLUDE | Тип | /BOBF/S_FRW_KEY | 000 | 0 | 0 | | |
| KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| PARENT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| ROOT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| .INCLUDE | Тип | ZVKO_S_TRUCKS_C | 000 | 0 | 0 | | NODE_DATA |
| CAR_ID | Тип | ZVKO_CARS_ID | CHAR | 20 | 0 | | |
| PRICE | Тип | ZVKO_CARRIAGE_P | CURR | 10 | 2 | | |
| COST | Тип | ZVKO_CARRIAGE_C | CURR | 10 | 2 | | |
| MILEAGE | Тип | ZVKO_TRUCKS_MIL | INT2 | 5 | 0 | | |
| CURRENCY | Тип | ZVKO_CARRIAGE_C | CUKY | 5 | 0 | | |
| .INCLUDE | Тип | ZVKO_TS_TRUCKS_ | 000 | 0 | 0 | | TRANSIENT_NODE_ |
| INCOME | Тип | ZVKO_CARRIAGE_P | CURR | 10 | 2 | | |

Рис. 2.3.9. Кінцева структура запису вантажу

| Компонент | Вид типизации | Тип компонента | Тип данных | Длина | ДесПа... | Краткое описание | Группа |
|------------|---------------|-----------------|------------|-------|----------|------------------|-----------|
| .INCLUDE | Тип | /BOBF/S_FRW_KEY | 000 | 0 | 0 | | |
| KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| PARENT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| ROOT_KEY | Тип | /BOBF/CONF_KEY | RAW | 16 | 0 | | |
| .INCLUDE | Тип | ZVKO_S_TRUCKS_S | 000 | 0 | 0 | | NODE_DATA |
| SERV_ID | Тип | ZVKO_CARS_ID_SE | CHAR | 20 | 0 | | |
| SERV_DATE | Тип | DATS | DATS | 8 | 0 | Поле типа DATS | |
| STATUS | Тип | ZVKO_TRUCKS_STA | CHAR | 20 | 0 | | |
| TRUCK_ID | Тип | ZVKO_CARS_ID | CHAR | 20 | 0 | | |
| DESCR | Тип | ZVKO_CARS_DESC | CHAR | 50 | 0 | | |
| PRICE | Тип | ZVKO_CARRIAGE_C | CURR | 10 | 2 | | |
| CURRENCY | Тип | ZVKO_CARRIAGE_C | CUKY | 5 | 0 | | |

Рис. 2.3.10. Кінцева структура запису обслуговування

Таким чином, база даних складається з таких таблиць:

- Водії
- Вантажівки
- Роботи з обслуговування
- Вантаж

Таблиці з вантажівками є підлеглими для таблиць водіїв, а таблиці обслуговування та вантажу підлеглі для таблиць вантажівок.

Діалоговий робочий процес. Діалогові робочі процеси працюють із запитамі від активного користувача на виконання кроків діалогу.

Робочий процес оновлення. Робочі процеси оновлення виконують запити на оновлення бази даних. Запити на оновлення є частиною SAP LUW, які об'єднують операції з базою даних, що виникають у результаті діалогу в LUW, для обробки у фоновому режимі.

Фоновий робочий процес. Програми обробки фонових робочих процесів, які можна виконувати без взаємодії з користувачем (фонові завдання).

Робочий процес блокування. Робочий процес блокування адмініструє таблицю блокування в області спільної пам'яті. Таблиця блокування містить логічні блокування бази даних для системи R/3 і є важливою частиною концепції SAP LUW.

Робочий процес передачі. Цей процес передає послідовні набори даних на друк або в архівування. Кожен сервер додатків може містити лише один робочий процес передачі.

7. Програма входить в цикл обробки екрану, що продовжуватиметься поки користувач не натисне на кнопку виходу.

8. Перевіряється чи натиснута кнопка підрахунку доходу. Якщо не натиснута – її обробка не викликається.

9. Якщо було натиснуто кнопку підрахунку доходу – виконується блок з її обробкою.

10. Перевіряється чи натиснуто на іншу вкладку. Якщо не натиснуто – перехід на іншу вкладку не відбувається.

11. Якщо було натиснуто на іншу вкладку – на неї відбувається перехід.

12. Перевіряється чи натиснута кнопка редагування. Якщо не натиснута – блок обробки кнопки не викликається.

13. Якщо було натиснуто на кнопку редагування – система переходить у стан редагування, під час якого виконуватиметься валідація введених даних.

14. Перевіряється чи натиснута кнопка зберігання. Якщо не натиснута – блок обробки кнопки не викликається.

15. Якщо було натиснуто на кнопку зберігання – система збереже нові дані, що пройшли валідацію.

Диспетчеризація та модель програмування. Розділення прикладного та презентаційного рівня спричинило необхідність розділення прикладних програм

на кроки діалогу. Це, а також той факт, що кроки діалогу спрямовуються на окремі робочі процеси, мали важливі наслідки для моделі програмування. Як згадувалося вище, робочий процес може вносити зміни в базу даних лише в межах однієї логічної одиниці роботи бази даних (LUW). Вміст бази даних має бути послідовним на початку і в кінці. Початок і кінець LUW бази даних визначаються командою фіксації для системи бази даних. Під час LUW, тобто між двома фіксаціями бази даних, сама система бази даних забезпечує узгодженість. Іншими словами, він бере на себе такі завдання, як блокування записів бази даних під час їх редагування або відновлення старих даних (відкат), якщо крок закінчується помилкою. Типова прикладна програма SAP поширюється на кілька екранів і відповідні кроки діалогу. Користувач запитує зміни бази даних на окремих екранах, які повинні призвести до узгодженості бази даних після того, як усі екрани будуть оброблені. Однак окремі кроки діалогу виконуються в різних робочих процесах, а один робочий процес може обробляти кроки діалогу з інших програм. Зрозуміло, що двом або більше незалежним програмам, кроки діалогу яких обробляються в одному робочому процесі, не можна дозволити працювати з однією LUW. Отже, робочий процес повинен відкривати окремі таблиці LUW для кожного кроку діалогу. Робочий процес надсилає команду фіксації до бази даних в кінці кожного кроку діалогу, під час якого він вносить зміни в базу даних. Ці команди фіксації називаються неявними фіксаціями бази даних, оскільки вони явно не записані в прикладну програму. Ці неявні фіксації бази даних означають, що LUW може залишатися відкритим максимум один крок діалогу. Це призводить до значного зниження навантаження на базу даних, а також дозволяє великій кількості користувачів використовувати ту саму систему. Схему цього процесу можна побачити на рисунку 2.3.11.

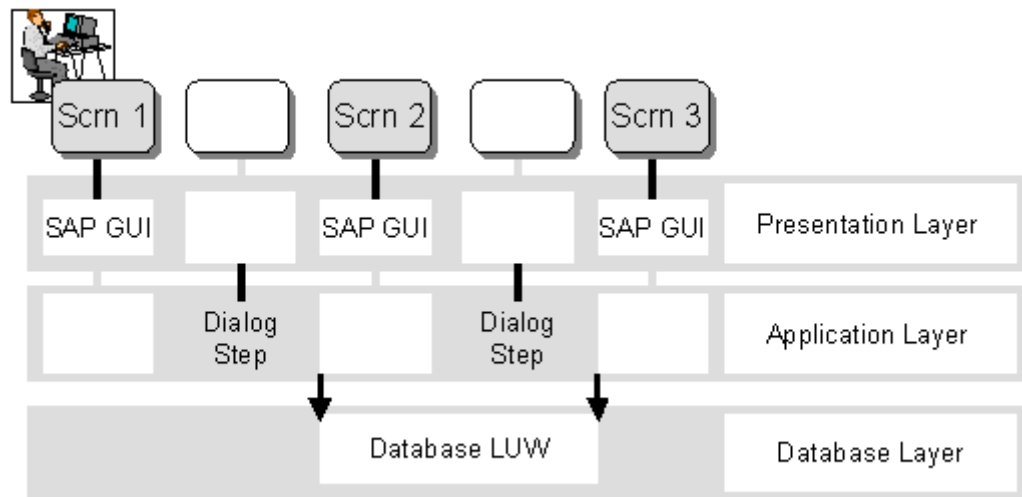


Рисунок 2.3.11. Схема внесення змін до бази даних

Однак тепер постає питання, як цей метод (1 крок діалогу = 1 база даних LUW) можна узгодити з фіксацією та відкатом, залежними від логічного потоку прикладної програми. Запити на оновлення бази даних, утворюють логічні блоки в програмі, які поширюються більше ніж на один крок діалогу. Зміни в базі даних, пов'язані з цими логічними одиницями, повинні виконуватися разом, а також мати можливість скасовувати їх разом. Модель програмування SAP містить низку методів групування, які дозволяють групувати оновлення бази даних разом у логічні одиниці. Розділ прикладної програми R/3, який об'єднує набір логічно пов'язаних операцій з базою даних, називається SAP LUW. На відміну від LUW бази даних, SAP LUW включає всі кроки діалогу в логічному блоці, включаючи оновлення бази даних.

Робочі процеси. Робочі процеси виконують окремі кроки діалогу в додатках R/3. Вони є компонентами сервера додатків. Структуру робочого процесу представлено на рисунку 2.3.12.

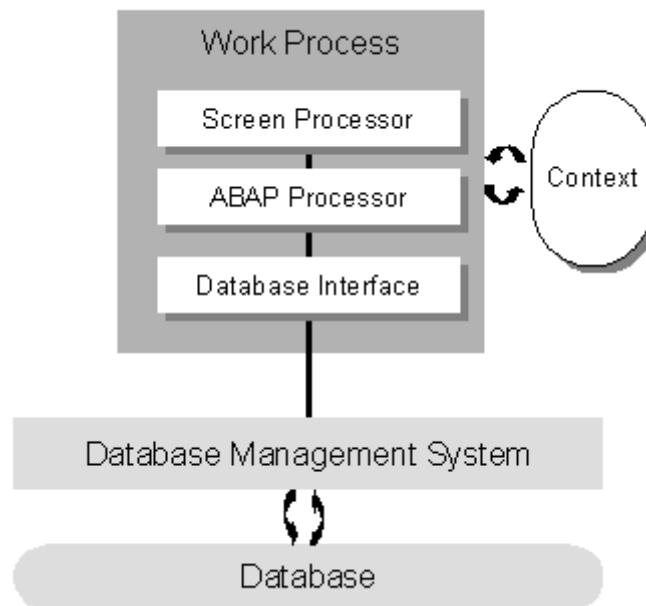


Рисунок 2.3.12. Структура робочого процесу

Кожен робочий процес містить два програмних процесора та інтерфейс бази даних. Екранний процесор У прикладному програмуванні R/3 існує різниця між взаємодією з користувачем та логікою обробки. З точки зору програмування, взаємодія з користувачем керується екранами. Крім полів для вводу, екран також складається з логіки потоку. Логіка екранного потоку контролює велику частину взаємодії з користувачем. Система R/3 Basis містить спеціальну мову для програмування логіки екранного потоку. Логіку екранного потоку виконує екранний процесор. Через диспетчер він бере на себе відповідальність за зв'язок між робочим процесом і SAP GUI, викликає модулі в логіці потоку і забезпечує передачу вмісту поля з екрана в логіку потоку.

2.4. Висновки до розділу

SAP R/3 – це корпоративне ПЗ з чималою часткою ринку. За даними агентства Forbes, станом на 2018 рік компанія SAP займала 26% ринку тільки у сфері управління поставками[15]. Для цього ПЗ вже написано багато технічних

рішень, котрі треба доповнювати та обслуговувати, тому розробку для цієї системи буде актуальною ще довго. ПЗ постійно еволюціонує, з'являються нові технології та фреймворки, деякі використовуватимуться в цій роботі, а саме: Floorplan Manager (для розробки інтерфейсу), Business Object Processing Framework (для розробки моделі даних), та FPM-BOPF Integration (для поєднання двох попередніх пунктів). В результаті виконання дипломної роботи було розроблено програму для атоматизації обліку підприємства з транспортних послуг на основі SAP. Програма складається з екрану ініціалізації, де користувач обирає бажану роль та робітника. Якщо акаунт користувача має таку роль, то відбудеться перехід на головну сторінку, де можна буде переглянути дерево об'єктів водія, розрахувати дохід підприємства та редагувати таблиці об'єктів водія (тільки для користувачів з роллю Manager).

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ТРАНСПОРТНИХ ПОСЛУГ НА ОСНОВІ SAP

3.1. Демонстрація роботи автоматизованої системи обліку транспортних послуг на основі SAP

Як вже зазначалося, робота в програмі йде з 4 об'єктами: водій, вантажівка, сервіс та вантаж. Нижче представлено приклад виконання CRUD-операцій та підрахунку доходу:

Запуск програми відбувається шляхом переходу за посиланням у браузері. Після авторизації на сервері, відкриється вікно ініціалізації, яке показано на рисунку 3.1.1.

| | | | | | | | |
|---------------------|------------------|--|--|--|---------------|--------------|----------------|
| Кафедра КСУ | | | | НАУ 22 02 82 000 ПЗ | | | |
| <i>Виконав</i> | Ковтун В. Є. | | | Програмна реалізація автоматизованої системи обліку транспортних послуг на основі SAP | <i>Літера</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| <i>Керівник</i> | Вавіленкова А.І. | | | | Д | 52 | 68 |
| <i>Консульт.</i> | | | | | 123 СП-436 | | |
| <i>Н- контроль.</i> | Тупота Є. В. | | | | | | |
| <i>Зав. кафедри</i> | Литвиненко О. Є. | | | | | | |

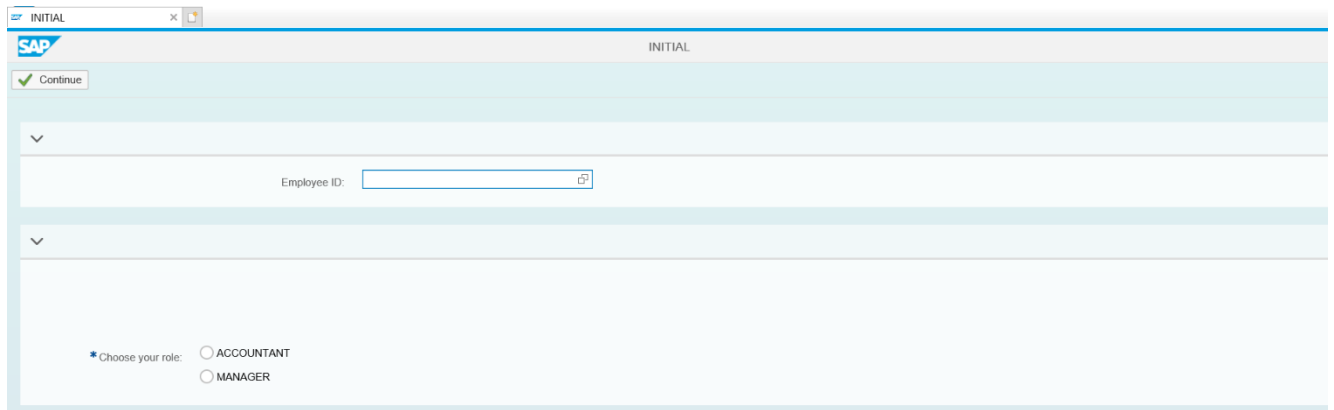


Рисунок 3.1.1. Вікно ініціалізації

Першим ділом обирається ідентифікаційний номер співробітника, по якому буде показано інформацію. Номер можна ввести одразу, або скористатися допомогою вводу, що покаже наявні в системі дані в новому вікні (кнопка зправа у полі вводу Employee ID). Відкрита допомога вводу зображена на рисунку 3.1.2.

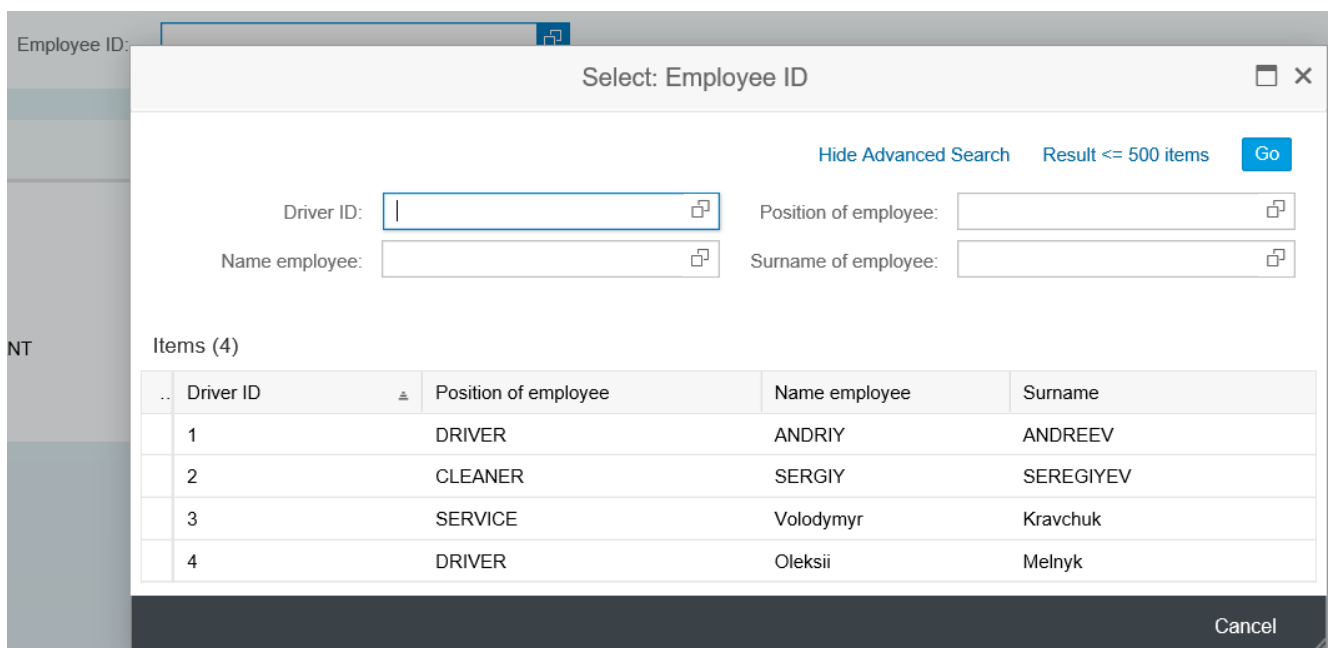


Рисунок 3.1.2. Вікно ініціалізації

Після обрання потрібного співробітника обирається роль входу, за допомогою двох кнопок на екрані, що зображені на рисунку 3.1.3.

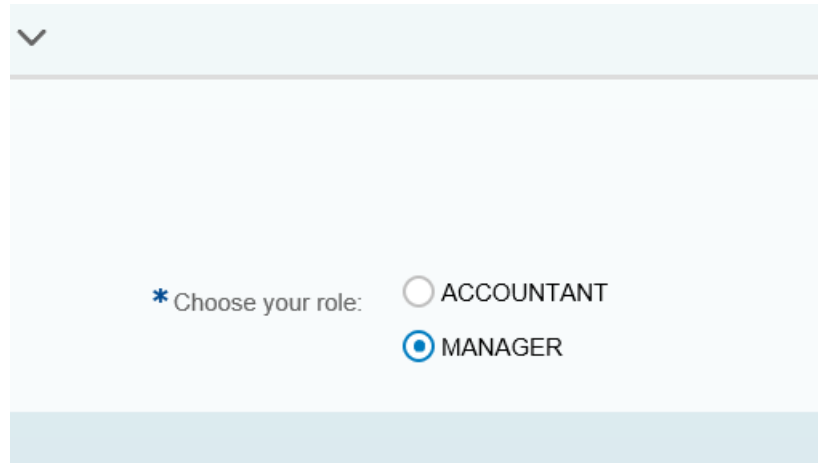


Рисунок 3.1.3. Вікно ініціалізації

Оскільки обраний шлях тестування вимагає права для CRUD-операцій, обрано роль MANAGER (тільки під такою роллю можна редагувати таблиці в програмі). Після цього відбудеться перевірка, чи справді акаунту користувача присвоєна така роль (вона видається адміністратором системи). В правому верхньому кутку виводиться результат перевірки, оскільки акаунт має відповідну роль – результат позитивний. Повідомлення можна побачити на рисунку 3.1.4.

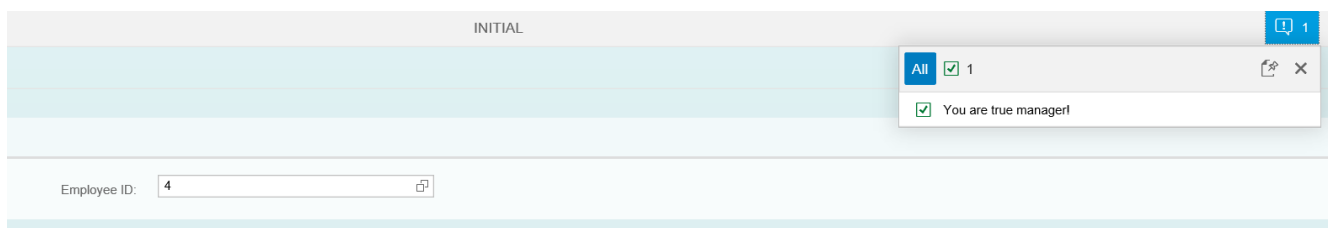


Рисунок 3.1.4. Результат перевірки ролі

Тепер можна переходити на наступний екран. Для цього треба натиснути на кнопку «Continue», після чого відбудеться перехід на головну сторінку. Результат переходу показано на рисунку 3.1.5.

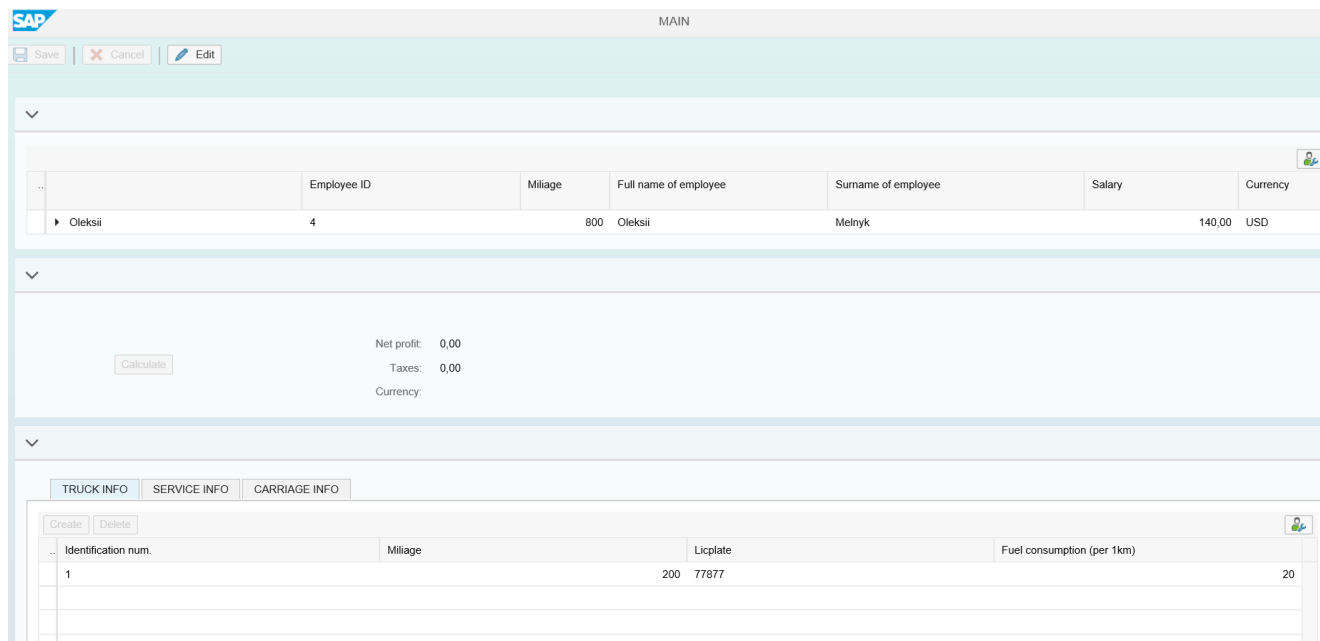


Рисунок 3.1.5. Головний екран

Інтерфейс головного екрану складається з трьох блоків: дерево об'єктів водія, з інформацією про нього, підрахунок доходів підприємства та інформаційний блок про об'єкти водія. Дерево можна розгорнути, і виконувати навігацію вкладками інформаційного блоку шляхом подвійного кліку по відповідному об'єкту. Розгорнуте дерево, та перемкнута вищезгаданим чином вкладка інформаційного блоку показані на рисунку 3.1.6.

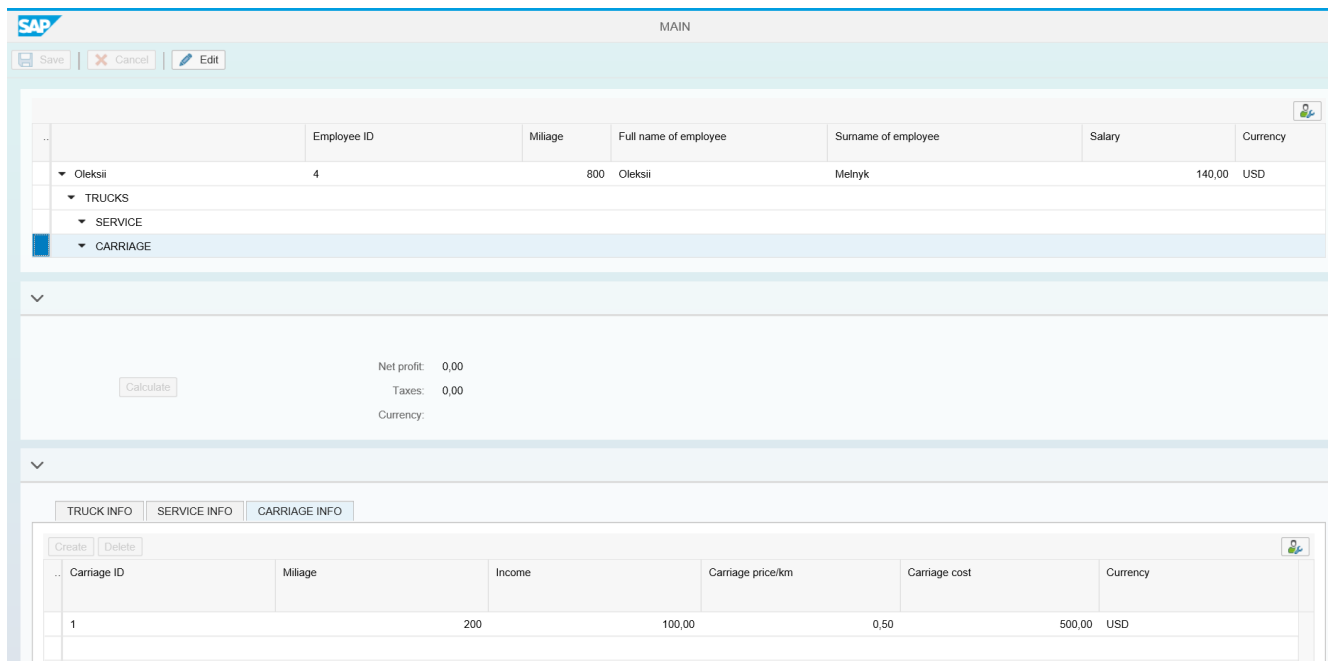


Рисунок 3.1.6. Розгорнуте дерево об'єктів

Далі йде підрахунку доходів. Для цього необхідно натиснути на кнопку «Calculate» у блоці посередині екрану (рис.3.1.6), але на цьому етапі вона заблокована, оскільки система знаходиться в стані читання. Щоб перевести її у стан редагування і розблокувати кнопку, необхідно натиснути на кнопку «Edit», що знаходиться на верхній панелі інструментів (рис.3.1.6). Розблокований екран можна побачити на рисунку 3.1.7.

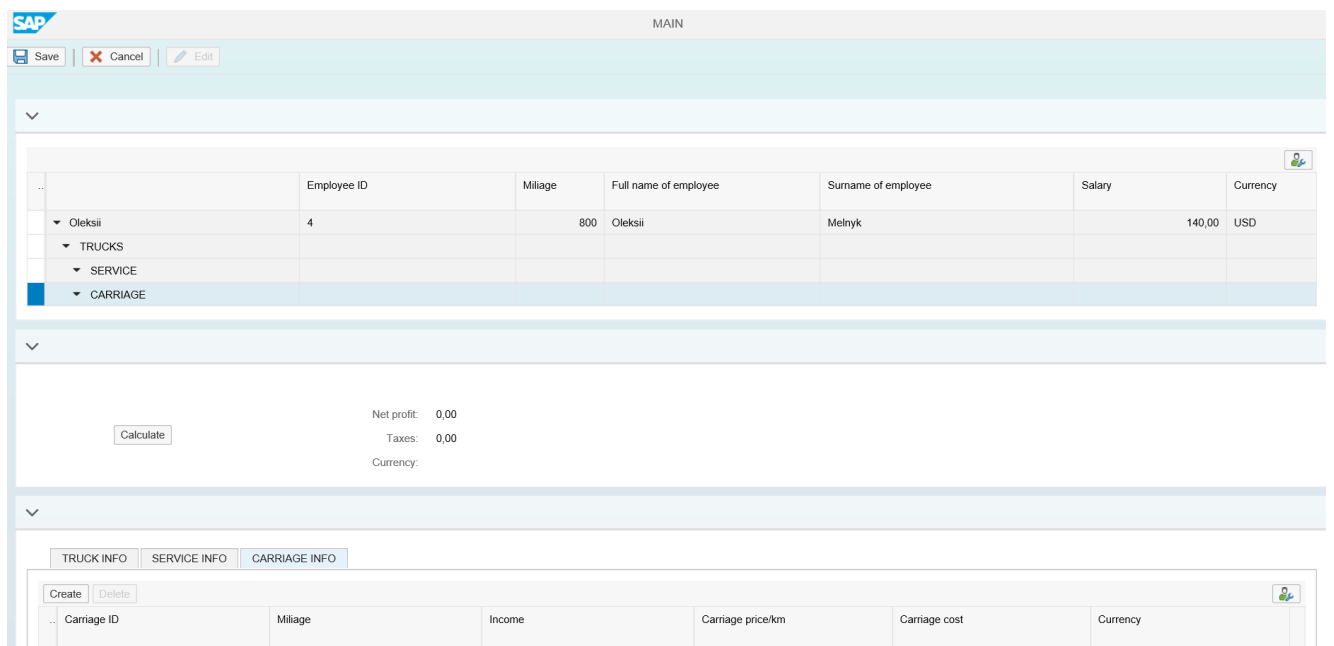


Рисунок 3.1.7. Інтерфейс в режимі редагування

Натиснувши на кнопку «Calculate», відбудеться заповнення транз'єєнтних полів блоку: чистий дохід, розмір заплаченого податку та валюта. Результат обрахунку показано на рисунку 3.1.8.

The screenshot displays a software interface with a light blue header and a white main area. At the top left, there is a dropdown menu with a downward arrow. Below it, a 'Calculate' button is visible. To the right of the button, the following values are displayed: Net profit: 756,00, Taxes: 324,00, and Currency: USD. Below this, another dropdown menu is present. The main content area features three tabs: 'TRUCK INFO', 'SERVICE INFO', and 'CARRIAGE INFO', with 'CARRIAGE INFO' being the active tab. Underneath the tabs, there is a table with columns for Carriage ID, Mileage, Income, Carriage price/km, Carriage cost, and Currency. The table contains one row of data: Carriage ID 1, Mileage 200, Income 100,00, Carriage price/km 0,50, Carriage cost 500,00, and Currency USD. The table has a 'Create' button and a 'Delete' button at the top left, and a user profile icon at the top right. The table is followed by several empty rows and a '...' indicator at the bottom.

Рисунок 3.1.8. Підраховані доходи підприємства

Далі йде інформаційний блок. У ньому є вкладки з інформацією про підлеглі об'єкти водія: вантажівки, роботи з обслуговування машин, відомості про вантажі. Тут розкривається CRUD-функціонал програми. Для додавання нового запису використовується кнопка «Create» на панелі інструментів інформаційного блоку. Після натискання додається порожній запис, це можна побачити на рисунку 3.1.9.

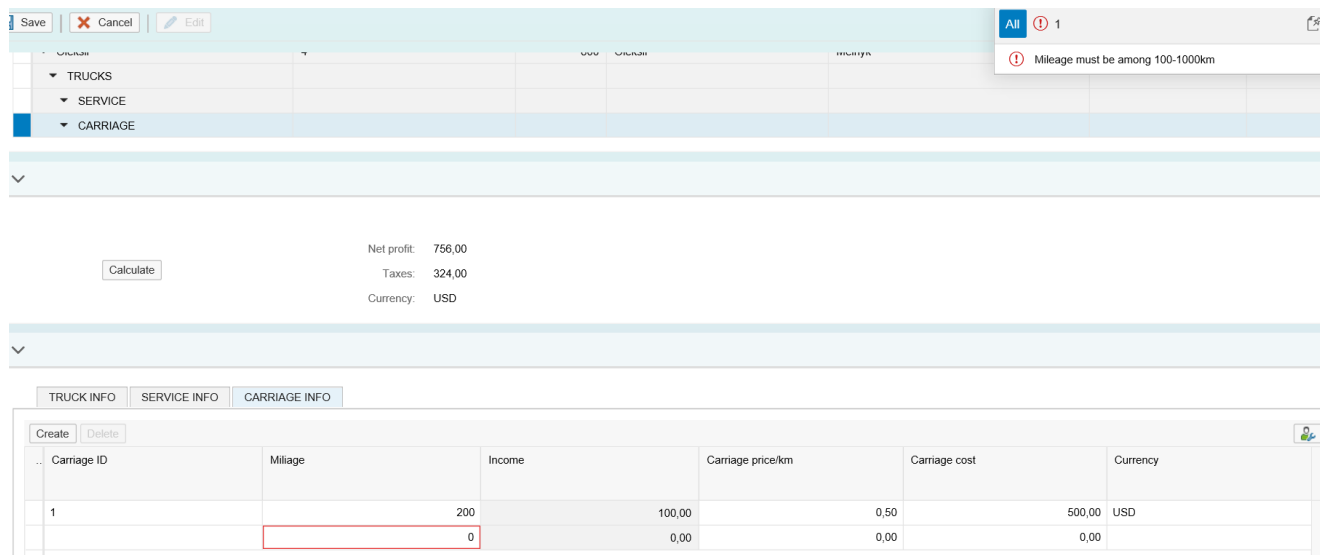


Рисунок 3.1.9. Додавання нового запису

З рисунку 3.1.9. також можна побачити, що до полів почали застосовуватися валідації. Заповнивши поля валідними значеннями, новий запис треба зберегти, для чого використовується кнопка «Save» на верхній панелі інструментів. Результат збереження можна побачити на рисунку 3.1.10.

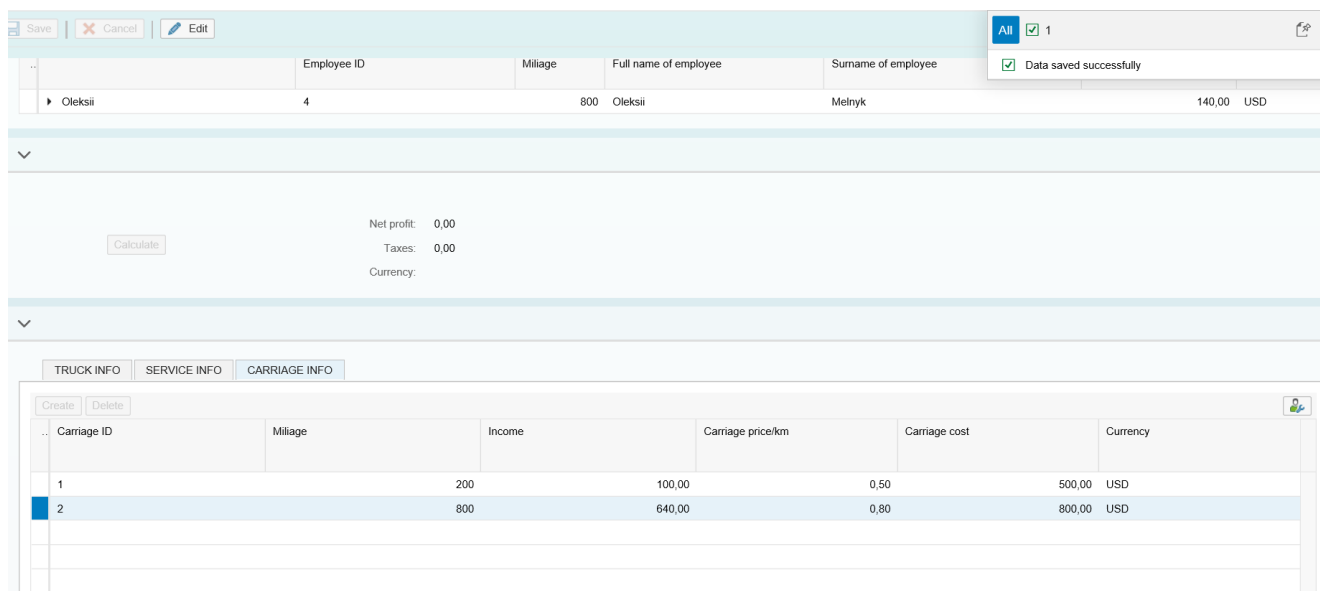


Рисунок 3.1.10. Збереження змін

Для видалення записів використовується кнопка «Delete». Щоб скористатися нею, треба обрати запис для видалення та перевести інтерфейс в режим редагування, як показано на рисунку 3.1.11.

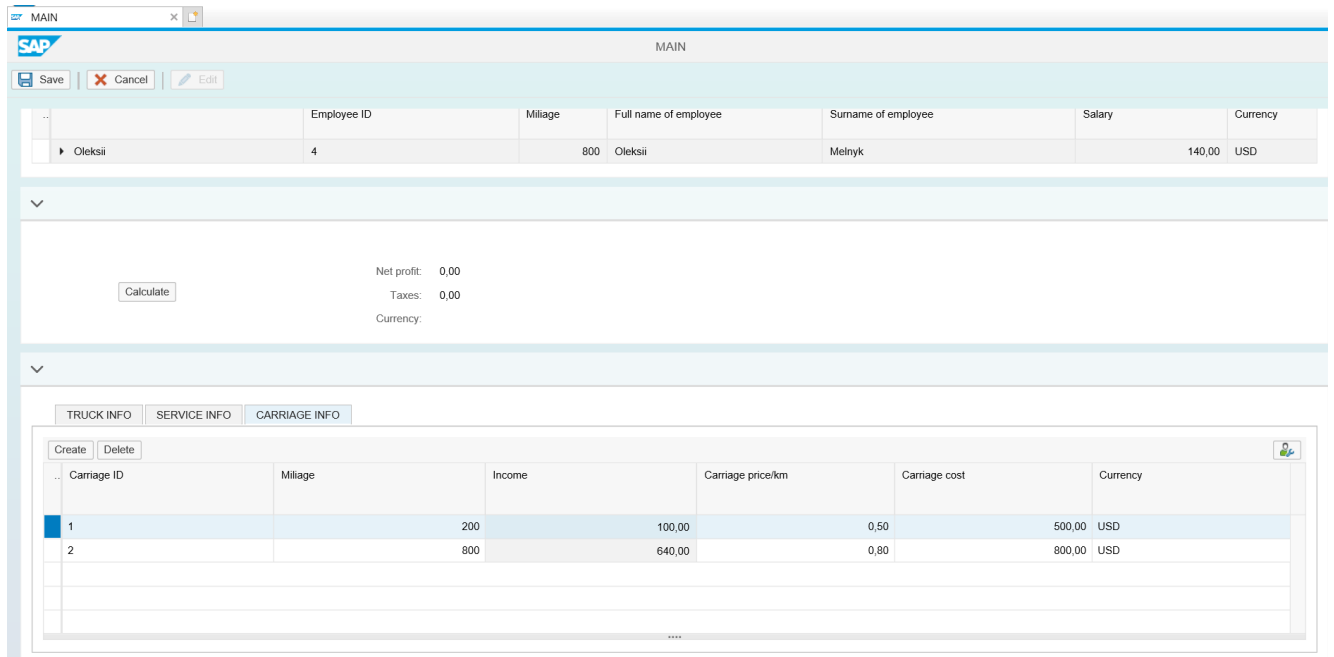


Рисунок 3.1.11. Розблокування кнопки видалення

Результат видалення запису можна побачити на рисунку 3.1.12.

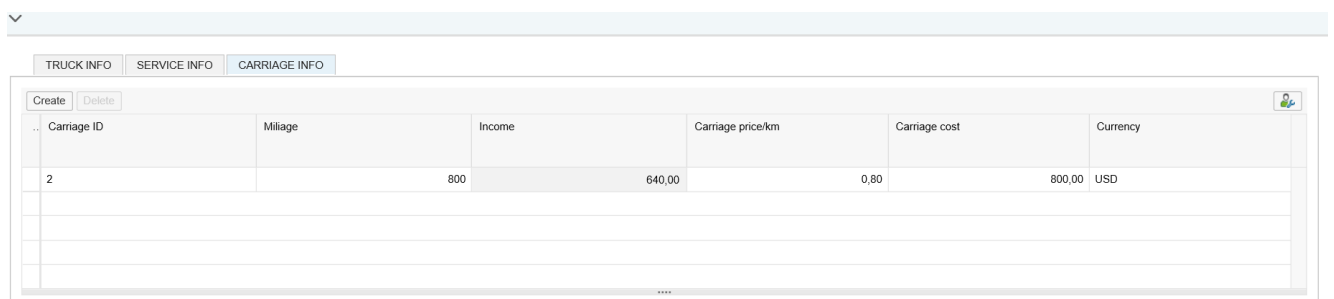


Рисунок 3.1.12. Видалення запису

Для збереження змін використовується кнопка «Save». Якщо ж, наприклад, зміни були внесені помилково, можна натиснути на кнопку «Cancel» на панелі інструментів, після чого зміни «відкатяться». Результат можна побачити на рисунку 3.1.13.

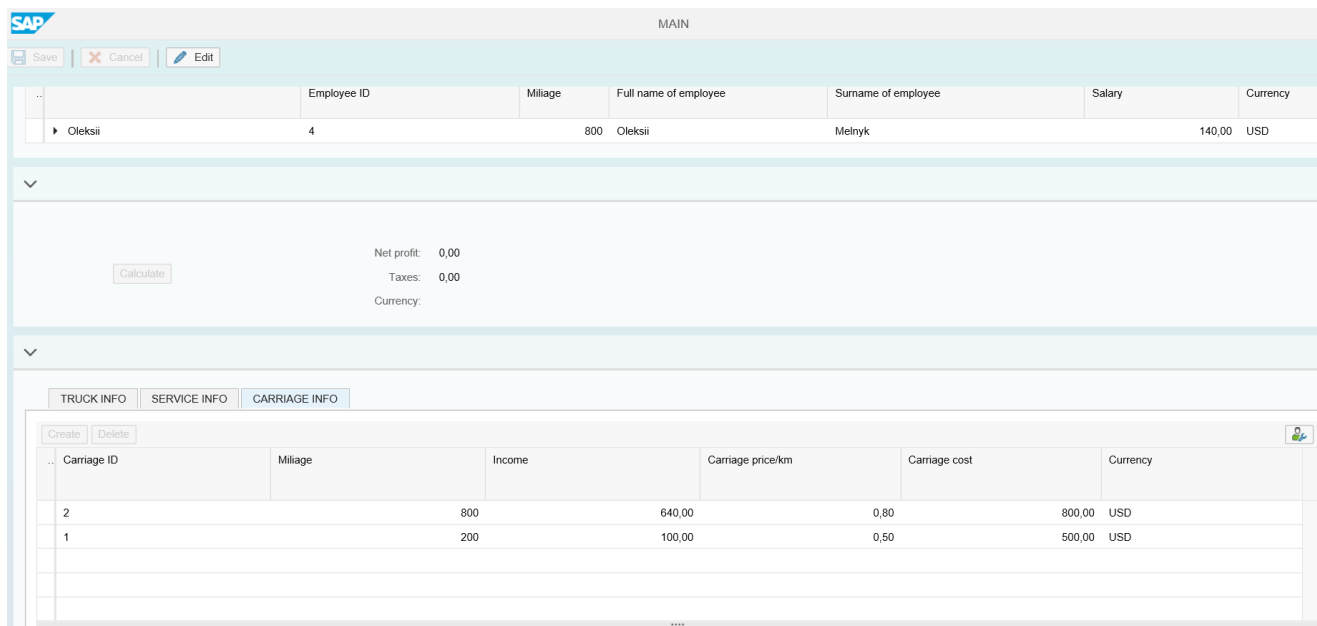


Рисунок 3.1.13. «Відкат» змін

Оновлення відбувається так само як і інші операції, але без участі кнопок з панелі інструментів інформаційного блоку. Треба перевести інтерфейс в режим редагування, зробити зміни та зберегти. Результат таких дій можна побачити на рисунку 3.1.14. В рамках тестування, змінено шлях, на який треба відвезти вантаж, з 800 км до 600 км.

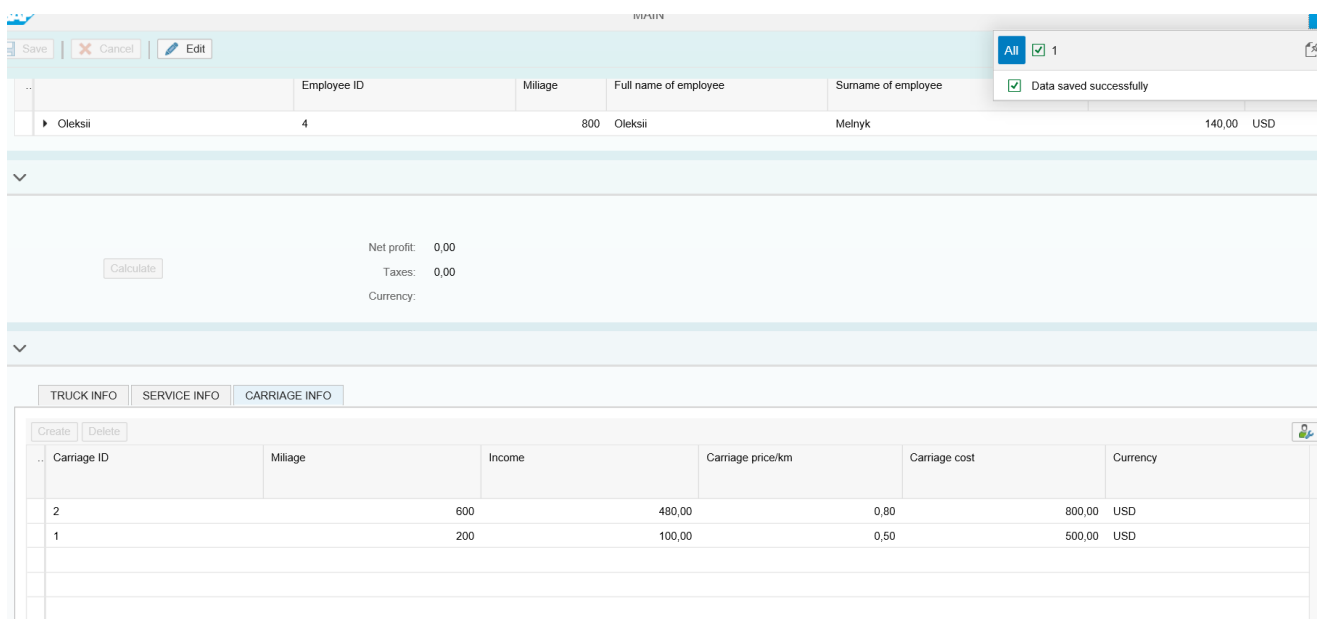


Рисунок 3.1.14. Результат оновлення полів

Вихід з програми виконується шляхом закриття вкладки браузера.

3.2. Кількісна оцінка роботи автоматизованої системи обліку транспортних послуг на основі SAP

Для оцінки доповнимо таблицю 1.2.5.1 розробкою:

| | Оцінка системних вимог | Ціна | Є своя СУБД | Є Маш.Навч. | Є у "хмарі" |
|-------------------|------------------------|---------------------|-------------|-------------|-------------|
| | за 5-бальною шкалою | | | | |
| 1С Предприятие | 3 | від 12300 грн. | + | - | - |
| BAS | 3 | від 12300 грн. | - | - | - |
| Jsolutions | 2 | 617 грн./ місяць | - | - | + |
| Oracle | 2 | Від 16100\$ | + | + | + |
| Розробка | 1 | 100 \$/місяць | + | - | - |

Табл. 3.2.1. Порівняльна таблиця розробленої системи з описаними

Дипломна розробка додана останнім рядком до таблиці. Системні вимоги для клієнта оцінені низько, оскільки розробка навіть не потребує від клієнта конкретної версії браузера – достатньо стандартного для Windows браузера Internet Explorer. За ціну взято абонентську плату за користування SAP-системою – 100 доларів на місяць за користувача. SAP-система, на базі якої створено розробку, використовує СУБД Hana від SAP, тобто «свою». Машинне навчання ніяк в розробці не використовується, «хмарною» назвати її теж не можна, оскільки вона виконується на приватному сервері, а не публічному.

3.3. Висновки до розділу

В результаті виконання дипломного проекту було розроблено систему автоматизованого обліку транспортного підприємства. Інтерфейс розробки складається з двох екранів: ініціалізації та головного екрану. На першому екрані користувач вводить ідентифікаційний номер робітника, по якому він хоче переглянути інформацію. Після цього користувач задає роль під якою він хоче зайти. Після перевірки відповідності обраній ролі, користувач потрапляє на головний екран, що складається з трьох блоків: дерево об'єктів водія з інформацією про нього, блок розрахунку доходу підприємства та блок з інформацією про об'єкти водія (машини, сервіс та вантаж), що надає можливість виконувати CRUD-операції відносно відповідних таблиць. Після внесення змін їх можна зберегти або відмінити, за допомогою відповідних кнопок на панелі інструментів. Оскільки в рамках клієнта програма виконується у браузері, для її закриття достатньо просто закрити вкладку, а для перезапуску достатньо перезавантажити веб-сторінку. Також були визначені кількісні характеристики розробки у рамках SAP R/3. Головною перевагою системи встановлено низькі системні вимоги, оскільки система не потребує конкретної версії браузера та плагінів, а підходить і для стандартного для системи Windows браузера Internet Explorer.

ВИСНОВКИ

В роботі велику увагу приділено сучасним ERP-системам, які не просто корисні в різних сферах підприємства, але дають абсолютно необхідний інструментарій для подальшого розвитку.

Особливу увагу також приділено питанню системних характеристик, зменшуючи які можна мінімізувати вимоги для комп'ютера клієнта.

Спираючись на результати порівняння різних ERP-систем можна прийти до висновку, що найменші системні вимоги мають системи, для роботи з якими не потрібно інстальовати додаткові програми, роботу з якими можна вести за допомогою браузера. Така система використовує комп'ютер клієнта тільки у для вводу та виводу інформації, в той час як вся робота з даними виконуватимуться на сервері.

Було зроблено такі висновки:

- Найменше навантаження на комп'ютер клієнта потребують системи, працювати з якими можна у браузері
- Системи, працювати з котрими можна у браузері, зазвичай єдиною вимогою ставлять певну його версію, виробника чи наявність певних плагінів.
- Отже, шляхом до мінімізації навантаження є розробка такої системи, що не буде прив'язана до певної версії браузера, а потребуватиме від клієнта лише доступу до мережі Інтернет та можливості вводу даних.

В результаті виконання дипломного проєкту було розроблено автоматизовану систему для обліку надання транспортних послуг на основі системи SAP R/3. Розроблена система дає змогу взаємодіяти з логікою, що зберігається на сервері у файлах класів, завдяки якій відбувається взаємодія з користувачем, конкретно розрахунок доходів підприємства та можливість виконувати CRUD-операції на таблицями з BOPF-моделей підлеглих об'єктів водія: вантажівки, сервіс та вантаж. Також створена система не потребує

конкретної версії браузера, а може запускатись навіть за допомогою стандартного для систем на базі ОС Windows браузера Internet Explorer.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Офіційний сайт Нафтогазу // <https://www.naftogaz.com/news/grupa-naftogaz-integrue-it-systemy-na-osnovi-sap> (дата звернення 28.05.2022) .
- 2) Офіційний сайт 1С // <https://solutions.1c.ru/catalog/tmsexp/features> (дата звернення 28.05.2022) .
- 3) Офіційний сайт Integra Soft // <https://soft.integra.ua/uk/product/bas-upravlinnya-avtotransportom-versiya-standart> (дата звернення 28.05.2022) .
- 4) Офіційний сайт Jsolutions // <https://jsolutions.ua/ua/preimuschestva-oblachnoy-sistemy-jsolutions> (дата звернення 28.05.2022) .
- 5) Офіційний сайт Національного Університету Біоресурсів і Природокористування України // <https://nubip.edu.ua/node/94359> (дата звернення 28.05.2022) .
- 6) Офіційний сайт Oracle // <https://www.oracle.com/cis/scm/logistics/transportation-management> (дата звернення 28.05.2022) .
- 7) Офіційний сайт ModalKon // <https://modalkon.com.ua/products/vehicles-management/> (дата звернення 28.05.2022) .
- 8) Стаття про трирівневу архітектуру SAP // https://www.tutorialspoint.com/sap/sap_architecture.htm (дата звернення 28.05.2022) .
- 9) Стаття про трирівневу архітектуру SAP // <https://www.guru99.com/learning-sap-architecture.html> (дата звернення 28.05.2022) .
- 10) Стаття про байт-код у SAP-довіднику // https://help.sap.com/doc/abapdocu_751_index_htm/7.51/enus/abenbytecode_glosry.htm (дата звернення 28.05.2022) .
- 11) Стаття про АВАР // <https://coderlessons.com/tutorials/sap/izuchite-sap-abap/sap-abap-obzor> (дата звернення 28.05.2022) .
- 12) Стаття про Floorplan Manager у SAP-довіднику // https://help.sap.com/doc/saphelp_nw73ehp1/7.31.19/enus/fc/182711c34a4684a7c0214b42554514/content.htm?no_cache=true (дата звернення 28.05.2022) .

13) Стаття про FBI у SAP-довіднику // https://help.sap.com/docs/SAP_BS_FOUNDATION/ebd178b5b21246de85ed21b34dbde2d1/031c7fb2a6fe409da7fd5e3a405854a7.html?version=7.02.17&locale=en-US (дата звернення 28.05.2022) .

14) Стаття про виконання програм у рамках трирівневої архітектури SAP // <http://sapbasistutorials.blogspot.com/p/the-r3-basis-system-overview-r3-basis.html> (дата звернення 28.05.2022) .

15) Стаття про частку SAP на ринку ланцюгів постачання // <https://www.forbes.com/sites/louiscolombus/2018/07/28/sap-leading-the-fast-growing-scm-market-with-26-share/?sh=ac3773b70dfa> (дата звернення 28.05.2022) .

Додаток А

Лістинг коду для обробки кнопки розрахунку доходу підприємства

```
class ZCL_VKO_TRUCKS_CALC_EXIT definition
public
  create public .

public section.

  interfaces /BOFU/IF_FBI_VIEW_EXIT_INTF .
  interfaces /BOFU/IF_FBI_VIEW_EXITINTF_RUN .
  interfaces /BOFU/IF_FBI_VIEW_EXITINTF_DEF .
protected section.
private section.
ENDCLASS.

CLASS ZCL_VKO_TRUCKS_CALC_EXIT IMPLEMENTATION.

METHOD /bofu/if_fbi_view_exitintf_run~adapt_data.

CASE io_event->mv_event_id.
  WHEN 'CALCULATE'.

  DATA: lo_srvmgr TYPE REF TO /bobf/if_tra_service_manager,
    lv_profit TYPE i,
    lv_gas TYPE i,
    lv_rent TYPE i VALUE 3000,
    lv_income TYPE i,
    lv_salary TYPE i,
    lv_service TYPE i,
    lv_tax TYPE i,
    lv_coff(5) TYPE p DECIMALS 2 VALUE '0.05',
    lv_gas_price(5) TYPE p DECIMALS 2 VALUE '0.6',
    lt_trq_root TYPE zvko_t_drivers_c,
    lt_carriage TYPE zvko_t_trucks_carriage_c,
    lt_service TYPE zvko_t_trucks_service_c,
    lt_truck TYPE zvko_t_trucks_c.
    lo_srvmgr = /bobf/cl_tra_serv_mgr_factory=>get_service_manager(
zif_zvkocargo_bo=>sc_bo_key ).
    lo_srvmgr->query(
EXPORTING
```

```
iv_query_key      = zif_zvkocargo_bo=>sc_query-root-select_all
iv_fill_data      = abap_true
IMPORTING
et_data           = lt_trq_root
et_key            = DATA(lt_so_root_key) ).
```

```
lo_srvmgr->retrieve_by_association(
EXPORTING
  iv_node_key      = zif_zvkocargo_bo=>sc_node-root
  it_key           = lt_so_root_key
  iv_association   = zif_zvkocargo_bo=>sc_association-root-truck
IMPORTING
  et_data          = lt_truck
  et_key_link      = DATA(lt_truck_key_link)
  et_target_key    = DATA(lt_truck_target_key)
).
```

```
lo_srvmgr->retrieve_by_association(
EXPORTING
  iv_node_key      = zif_zvkocargo_bo=>sc_node-root
  it_key           = lt_so_root_key
  iv_association   = zif_zvkocargo_bo=>sc_association-root-to_carriage
  iv_fill_data     = abap_true
IMPORTING
  et_data          = lt_carriage
  et_key_link      = DATA(lt_carriage_key_link)
  et_target_key    = DATA(lt_carriage_target_key)
).
```

```
lo_srvmgr->retrieve_by_association(
EXPORTING
  iv_node_key      = zif_zvkocargo_bo=>sc_node-root
  it_key           = lt_so_root_key
  iv_association   = zif_zvkocargo_bo=>sc_association-root-to_service
  iv_fill_data     = abap_true
IMPORTING
  et_data          = lt_service
  et_key_link      = DATA(lt_service_key_link)
  et_target_key    = DATA(lt_service_target_key)
).
```

```
LOOP AT lt_trq_root ASSIGNING FIELD-SYMBOL(<fs_line>).
  IF <fs_line>-occupation EQ 'DRIVER'.
```

*<fs_line>-salary = 100 + lv_coff * <fs_line>-mileage.*

ELSE.

<fs_line>-salary = 500.

ENDIF.

<fs_line>-currency = 'USD'.

lv_salary = lv_salary + <fs_line>-salary.

ENDLOOP.

LOOP AT lt_truck ASSIGNING FIELD-SYMBOL(<fs_truck_line>).

*lv_gas = lv_gas + <fs_truck_line>-consumption / 100 * <fs_truck_line>-
mileage * lv_gas_price.*

ENDLOOP.

LOOP AT lt_service ASSIGNING FIELD-SYMBOL(<fs_service_line>).

lv_service = lv_service + <fs_service_line>-price.

ENDLOOP.

LOOP AT lt_carriage ASSIGNING FIELD-SYMBOL(<fs_carriage_line>).

*lv_income = lv_income + <fs_carriage_line>-price * <fs_carriage_line>-
mileage.*

ENDLOOP.

lv_profit = lv_income - lv_service - lv_gas - lv_salary.

IF lv_profit > 0.

*lv_tax = lv_profit * 30 / 100.*

ELSE.

lv_tax = 0.

ENDIF.

lv_profit = lv_profit - lv_tax.

*ASSIGN COMPONENT 'CURRENCY' OF STRUCTURE cs_data TO FIELD-
SYMBOL(<fs_curr>).*

<fs_curr> = 'USD'.

ASSIGN COMPONENT 'TAXES' OF STRUCTURE cs_data TO <fs_curr>.

<fs_curr> = lv_tax.

ASSIGN COMPONENT 'NET_PROFIT' OF STRUCTURE cs_data TO <fs_curr>.

<fs_curr> = lv_profit.

cv_data_changed = 'X'.

ENDCASE.

ENDMETHOD.