





## 6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу, створення плану дипломної роботи та побудова плану-графіку виконання робіт.	02.10.2023 – 15.10.2023	
2.	Огляд та аналіз наукової літератури по темі дипломної роботи та написання Розділу 1.	16.10.2023 – 20.10.2023	
3.	Написання Розділу 2 дипломної роботи.	21.10.2023 – 01.11.2023	
4.	Написання Розділу 3 дипломної роботи. Завершення створення пояснювальної записки дипломної роботи.	02.11.2023 – 15.11.2023	
5.	Оформлення та друк пояснювальної записки.	16.11.2023 – 25.11.2023	
6.	Створення презентації, доповіді та підготовка до захисту дипломної роботи.	26.11.2023 – 01.12.2023	
7.	Підготовка матеріалів дипломної роботи для передачі секретарю ДЕК (папка, конверт, диск із файлом диплому, рецензія, відгук).	01.12.2023 – 10.12.2023	

7. Дата видачі завдання: «02» жовтня 2023 р.

Керівник дипломної роботи \_\_\_\_\_ Ігор РАЙЧЕВ

(підпис керівника)

(П.І.Б.)

Завдання прийняв до виконання \_\_\_\_\_ Вадим КОВТУН

(підпис випускника)

(П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Інформаційна система управління персоналом підприємства на основі SAP» складається зі вступу, трьох розділів, висновку, списку бібліографічних посилань та одного додатку і містить 133 сторінки, 1 таблицю та 113 рисунків. Список бібліографічних посилань складається з 10 найменувань.

**Ключові слова:** ІНФОРМАЦІЙНА СИСТЕМА, ERP, SAP, SAP S/4 HANA CLOUD, ФІКСАЦІЯ ВІДПРАЦЬОВАНОГО ЧАСУ, АВАР.

**Об'єктом дослідження** є процес розробки інформаційної системи управління персоналом.

**Предметом дослідження** є методи та засоби які використовуються при дослідженні та розробці інформаційних систем управління персоналом.

**Метою дипломної роботи** є дослідження процесу розробки інформаційної системи управління персоналом, методів та засобів розробки таких систем в контексті SAP S/4 HANA Cloud. Як результат досліджень, на основі отриманих знань необхідно розробити власну інформаційну систему управління персоналом. Додатковою метою є проведення функціонального аналізу програмного продукту.

**Актуальність:** Технологія ERP дала можливість підприємствам значно оптимізувати свої бізнес-процеси. Із зростанням доступності такими системами мають змогу користуватися не тільки великі підприємства, але й середні та малі.

Для досягнення поставленої мети необхідно виконати такі **завдання:**

- виявити місце ERP в сучасному світі;
- детально ознайомитися з документацією розробки в системі SAP S/4 HANA Cloud;
- розглянути стандартні інструменти фіксації часу, для виявлення переваг та недоліків;
- ознайомитися з техніками розробки в системі SAP S/4 HANA Cloud;
- розробити власну інформаційну систему управління персоналом.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	7
ВСТУП.....	8
РОЗДІЛ 1. СИСТЕМИ ПЛАНУВАННЯ РЕСУРСІВ ПІДПРИЄМСТВА.....	<b>Error!</b>
<b>Bookmark not defined.</b>	
1.1. Технологія ERP .....	10
1.2. Аналіз використання ERP у сфері управління персоналом	<b>Error! Bookmark</b>
<b>not defined.</b>	
1.2.1. 1С Підприємство... ..	16
1.2.2. SAP S/4 HANA Cloud .....	17
1.2.3. Microsoft Dynamics .....	18
1.2.4. BAS ERP .....	19
1.2.5. Порівняння розглянутих систем.....	22
1.3. Вимоги до проектованої системи управління персоналом на основі SAP	23
1.4. Висновки до Розділу 1 .....	24
РОЗДІЛ 2. ТЕХНОЛОГІЇ SAP S/4 HANA CLOUD ДЛЯ ОРГАНІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ .....	26
2.1. Архітектура SAP S/4 HANA Cloud.....	26
2.2. Мова програмування ABAP.....	28
2.3. Технології розробки додатків для SAP S/4 HANA Cloud .....	29
2.4. Огляд стандартних засобів управління персоналом у SAP S/4 HANA Cloud .....	31
2.5. Висновки до Розділу 2 .....	34
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ.....	35

3.1. Підготовка робочого простору .....	35
3.2. Розробка моделі даних .....	40
3.3. Розробка поведінки системи .....	82
3.4. Розробка контролю доступу до системи .....	85
3.5. Налаштування інтерфейсу додатку та створення сервісу .....	89
3.6. Налаштування SAP-системи .....	97
3.7. Тестування розробленої системи .....	99
3.8. Висновки до Розділу 3 .....	103
ВИСНОВКИ .....	108
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	110
ДОДАТОК А .....	112

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ**

SAP (System Analysis Program Development) – компанія, що займається розробкою корпоративного програмного забезпечення.

ERP (Enterprise resource planning) – система планування ресурсів підприємства

HR (Human Resources) – відділ кадрів

ПЗ – програмне забезпечення

API (*Application Programming Interface*) – інтерфейс для взаємодії з іншими додатками

## ВСТУП

Сучасне підприємство може мати чимало бізнес-процесів, підтримувати котрі вручну стає занадто складно. Окрім цього, ріст підприємства призводить лише до збільшення кількості та складності цих бізнес-процесів.

Тому такі підприємства використовують системи планування ресурсів підприємства (ERP), що допомагають автоматизувати роботу організації. Перші ERP-системи були монолітними програмними продуктами, кожен з яких після покупки покупець унікально допрацьовував під власні потреби. Такий підхід потребував значних фінансових та часових витрат, адже компанія мала продовжувати працювати під час такої доробки.

Сучасні ERP-системи мають значно більший стартовий функціонал, і потребують значно менших витрат для конфігурації. Окрім цього, змінилась і їх структура – тепер вони розбиті на функціональні модулі: HR, управління фінансами, логістика і т.д.

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».

На українському ринку представлено цілий ряд і інших програмних рішень: 1С Підприємство, BAS ERP, Microsoft Dynamics.

У зв'язку з актуальністю було обрано таку тему дипломної роботи: «Інформаційна система управління персоналом підприємства на основі SAP».

Метою створення такої системи є підвищення ефективності роботи користувачів з системою відслідковування відпрацьованого часу.

Основні завдання, які поставлені в даній дипломній роботі:

- 1) Дослідження технології ERP;
- 2) Огляд ERP, наявних на українському ринку;
- 3) Сформулювати вимоги до інформаційної системи;
- 4) Обрати найбільш придатну ERP для розробки системи;



- 5) Проектування та реалізація інформаційної системи управління персоналом;

Практична значущість розробки інформаційної системи має бути такою, щоб підприємство могло використовувати таку інформаційну систему без витрати значних зусиль на її конфігурацію.

## РОЗДІЛ 1

### АВТОМАТИЗОВАНІ СИСТЕМИ ОБЛІКУ

#### 1.1. Технологія ERP

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. Такий комплекс, при цьому, не є монолітним: організація може використовувати (або не використовувати) ті чи інші модулі на свій розсуд.

Без деяких модулів не може обійтись жодне підприємство: управління фінансами, продажами, персоналом, закупівлями тощо. Такі модулі обслуговують базові бізнес-процеси, без котрого діяльність підприємства неможлива. Та сучасні ERP містять специфічні рішення для різних галузей у сферах виробництва та послуг. Так, 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 ремонти обладнання. Всі ERP-системи, незалежно від того, хто їх розробник, об'єднує спільна архітектура, котру можна описати так (рисунок 1.1. ):

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.

Кафедра КІТ (47)				НАУ 23 30 30 000 ПЗ			
<i>Виконав</i>	Ковтун В.Є.			Автоматизовані системи обліку	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Райчев І. Е.				Д	10	16
<i>Консульт.</i>					10		
<i>Н- контроль.</i>	Райчев І. Е.				УС-212М	122	

Цей функціонал вбудований у платформу, на відміну від модулів його не можна відключити.

- **Управління даними.** База даних, відповідає за 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 підключатися і відключатися, у тому разі якщо потреба в них зникла. Така модульна структура – ключова ознака ERP-систем. Модулі поділяються, у свою чергу, на кілька типів:

1. Модулі внутрішнього користування – 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.

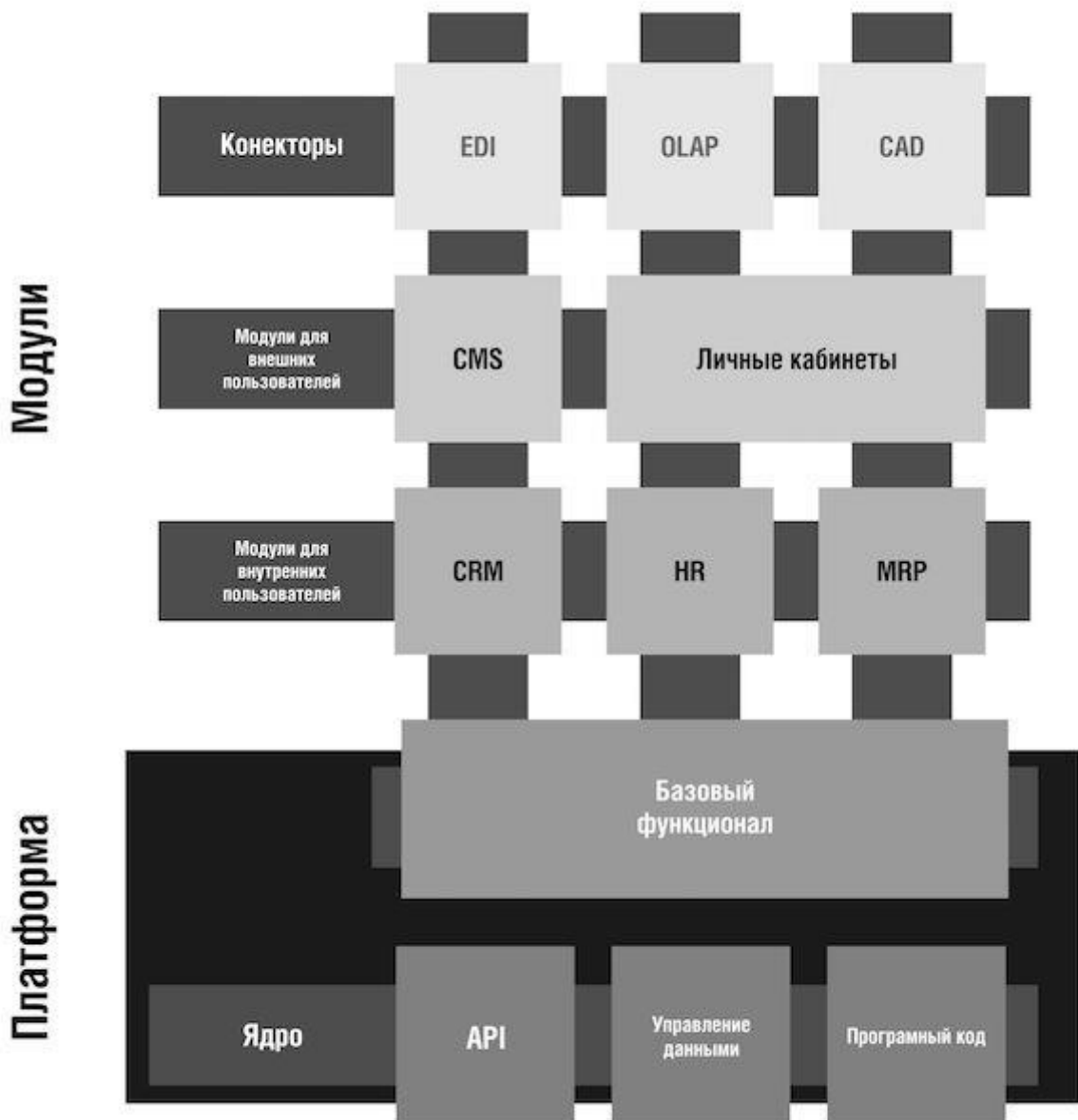


Рисунок 1.1. Типова структура ERP

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 вдосканалілися, 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:

- SAP R3, компанії SAP
- BAS ERP, компанії BAS Soft
- Microsoft Dynamics, компанії Microsoft
- 1С Предприятие v8, компанії 1С

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 1.2.

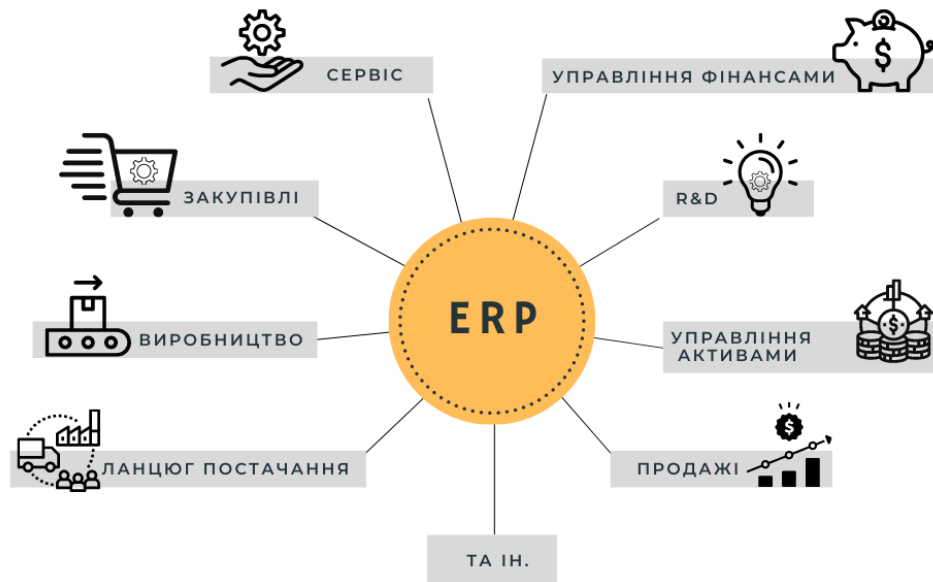


Рисунок 1.2. Сфери застосування ERP-систем

## 1.2. Аналіз використання ERP у сфері управління персоналом

На сьогоднішній день існують такі програмні рішення:

### 1.2.1. 1С: Підприємство 8

Модуль “Зарплата і управління персоналом”. Дане програмне рішення дозволяє:

- 1) Автоматизувати нарахування зарплати працівникам. Розрахунок ведеться на основі робочого календаря та параметрів, що вказує користувач (оклад,

податки, штрафи, премії і т.п.). Інтерфейс програми можна побачити на рисунку 1.3.

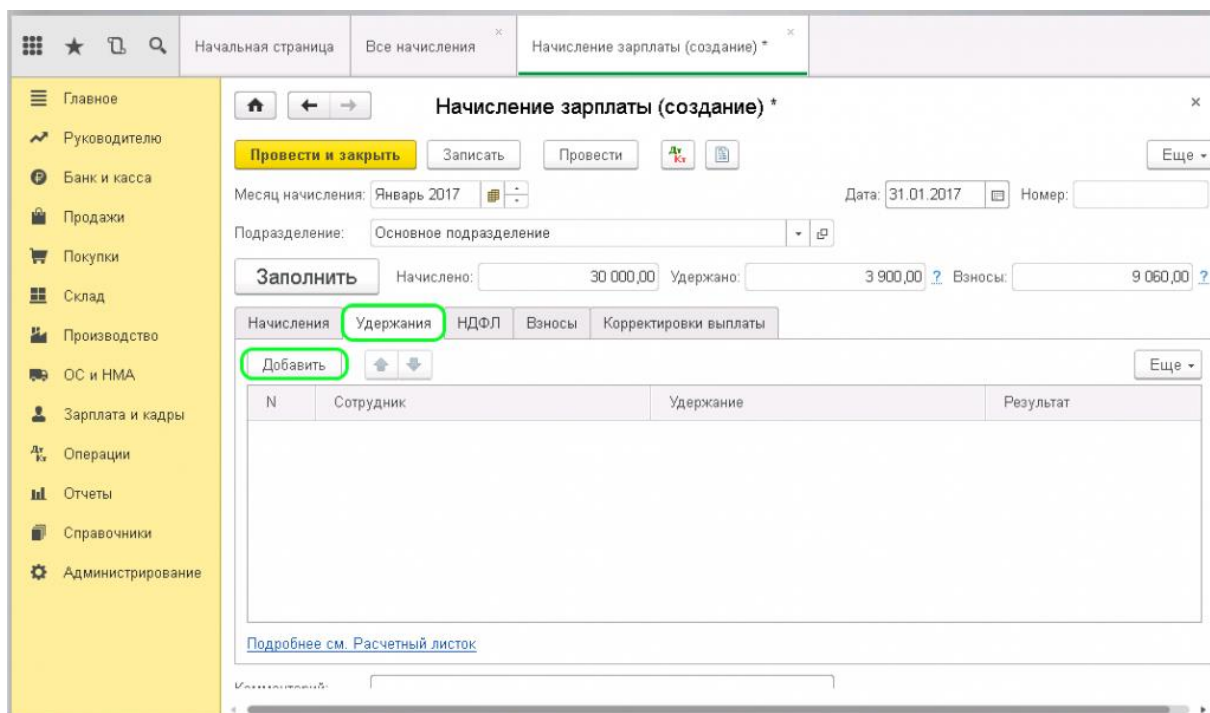


Рисунок 1.3. Интерфейс програми “Зарплата”

- 2) Допомагає впроваджувати кадрову політику (управління атестацією, навчанням, планування відпусток). Також дозволяє керувати пільгами для робітників і публікувати вакансії на сайтах пошуку роботи. Інтерфейс програми представлено на рисунку 1.4.



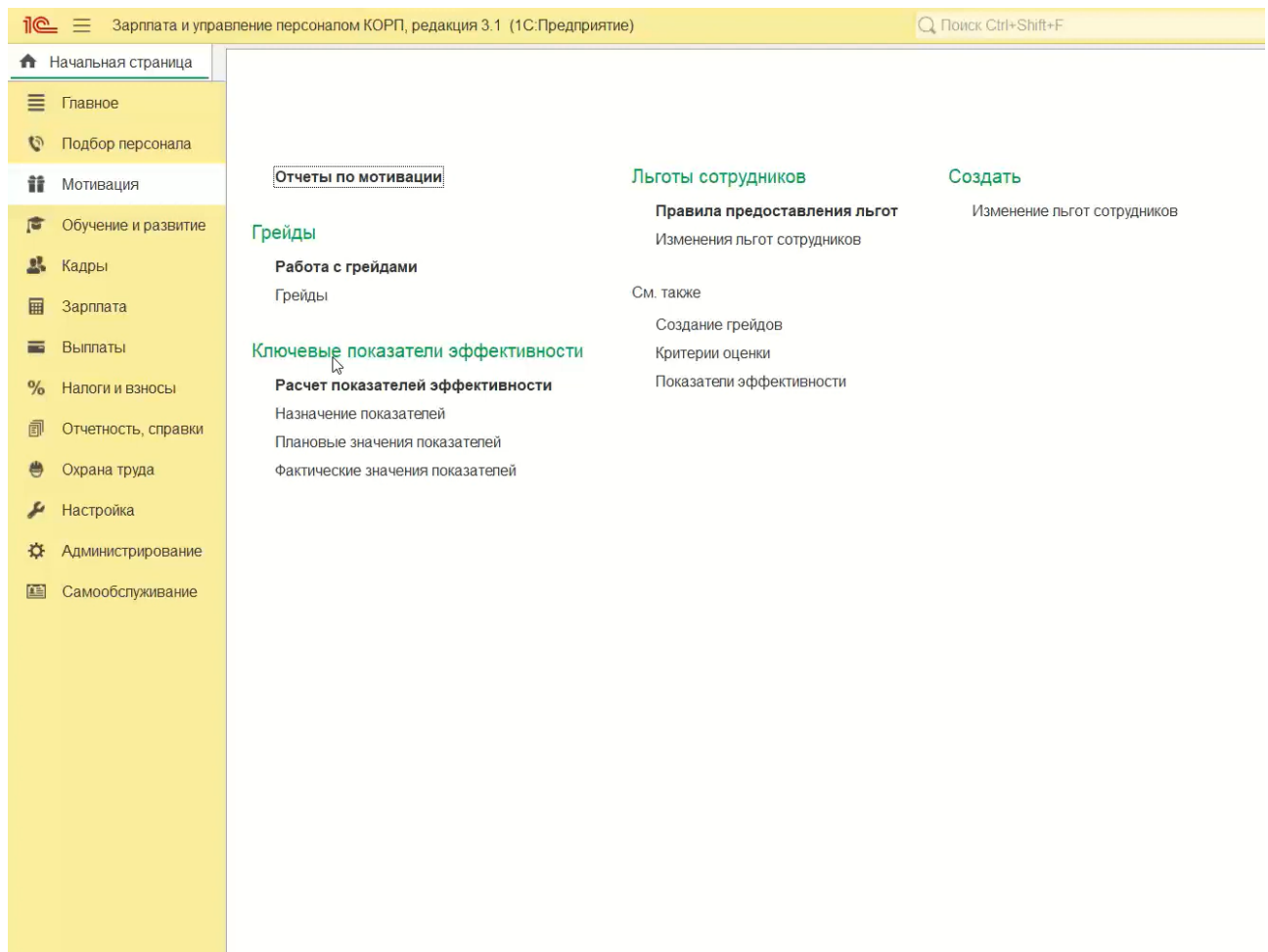


Рис. 1.4. Интерфейс программы “Управление персоналом”

### 1.2.2. SAP S/4 HANA Cloud

Модуль “SAP S/4HANA Cloud for Human Resources” дозволяє:

- Виконувати головні задачі з управління персоналом (керування кадрами підприємства, керування кадрами підрядника).
- Нараховувати заробітню плату.
- Керувати кваліфікацією співробітників.
- Відслідковувати відпрацьований робітниками час.

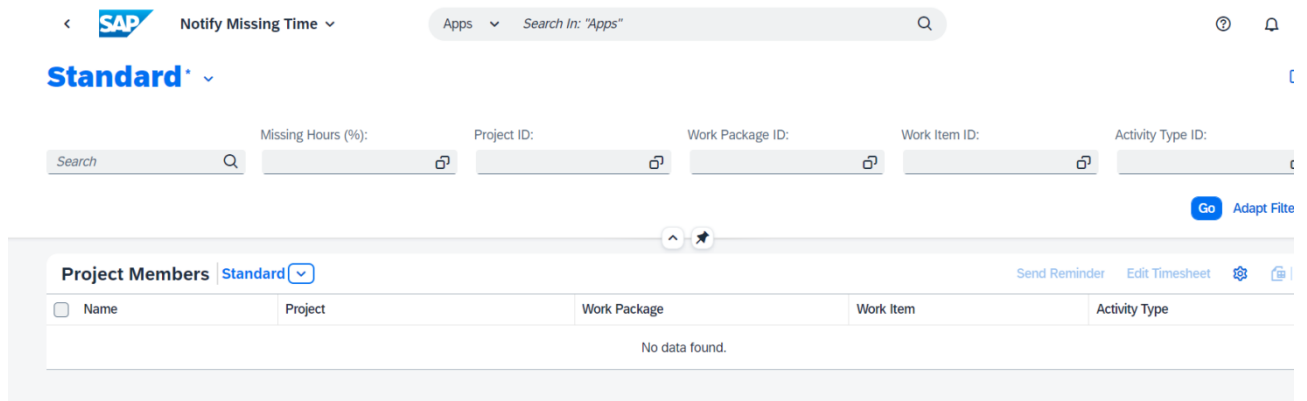


Рис. 1.5. Інтерфейс програми відслідковування часу в SAP S/4 HANA Cloud

### 1.2.3. Microsoft Dynamics

Модуль Dynamics 365 Human Resources дозволяє обслуговувати такі бізнес-процеси:

- Адміністрування організаційних структур.
- Зберігання інформації про робітника.
- Адміністрування пільг.
- Впровадження та моніторинг планів відпусток та відгулів.

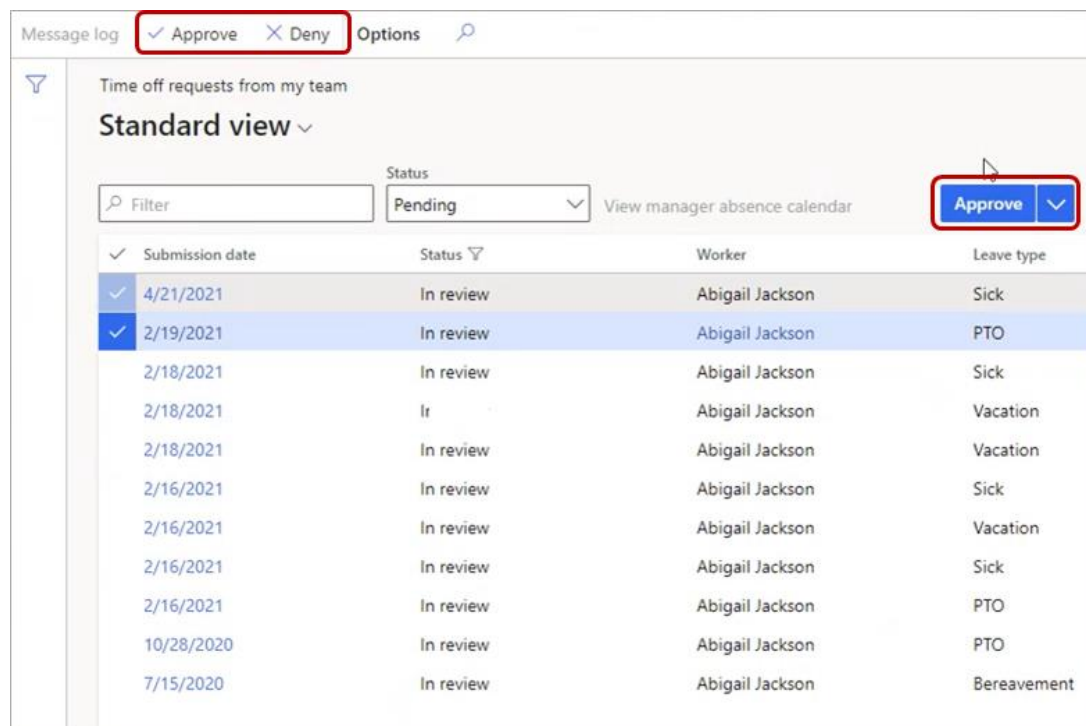


Рисунок 1.6. Інтерфейс управління запитами на відпустки у Microsoft Dynamics 365 Human Resources

- Відслідковування часу, відпрацьованого робітником.

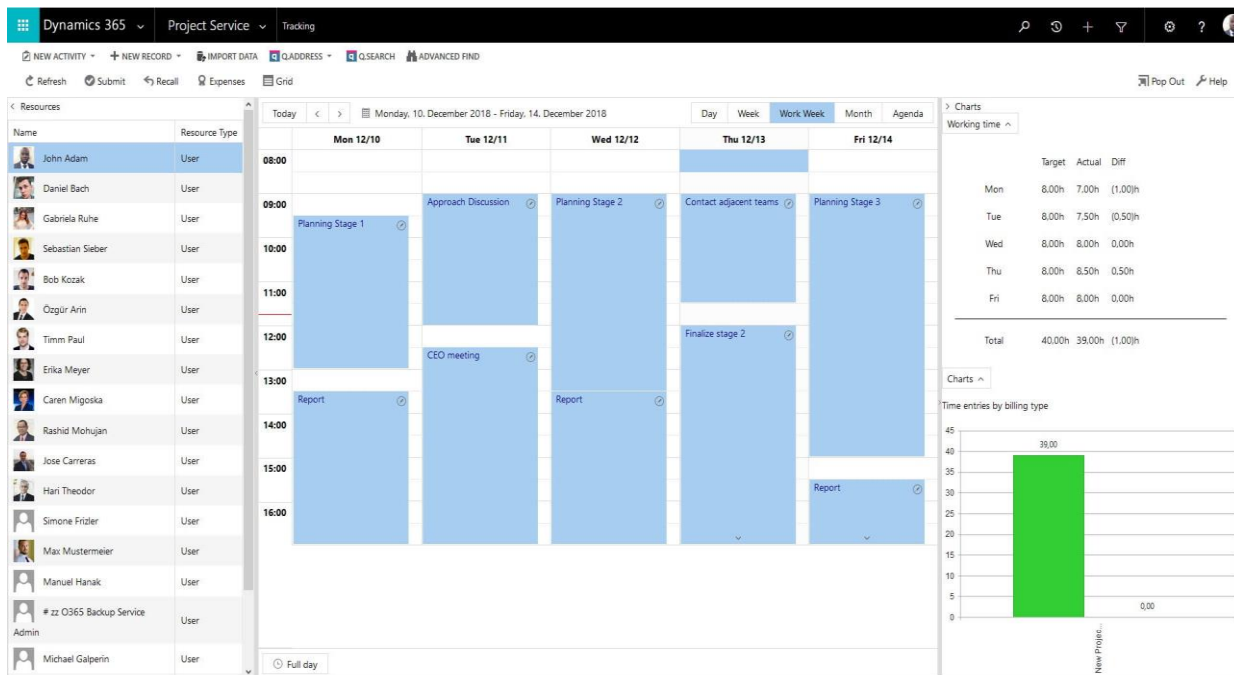


Рисунок 1.7. Інтерфейс відслідковування часу у Microsoft Dynamics 365 Human Resources

- Управління кваліфікацією робітників.
- Моніторинг продуктивності робітників.
- Набор персоналу.

#### 1.2.4. BAS ERP

BAS ERP – програмне рішення для управління підприємством, зроблене на платформі 1С. Містить модуль для управління персоналом – “Управління персоналом та розрахунок заробітної плати”. Здатен виконувати такі функції:

- Робота з персональними даними співробітників.

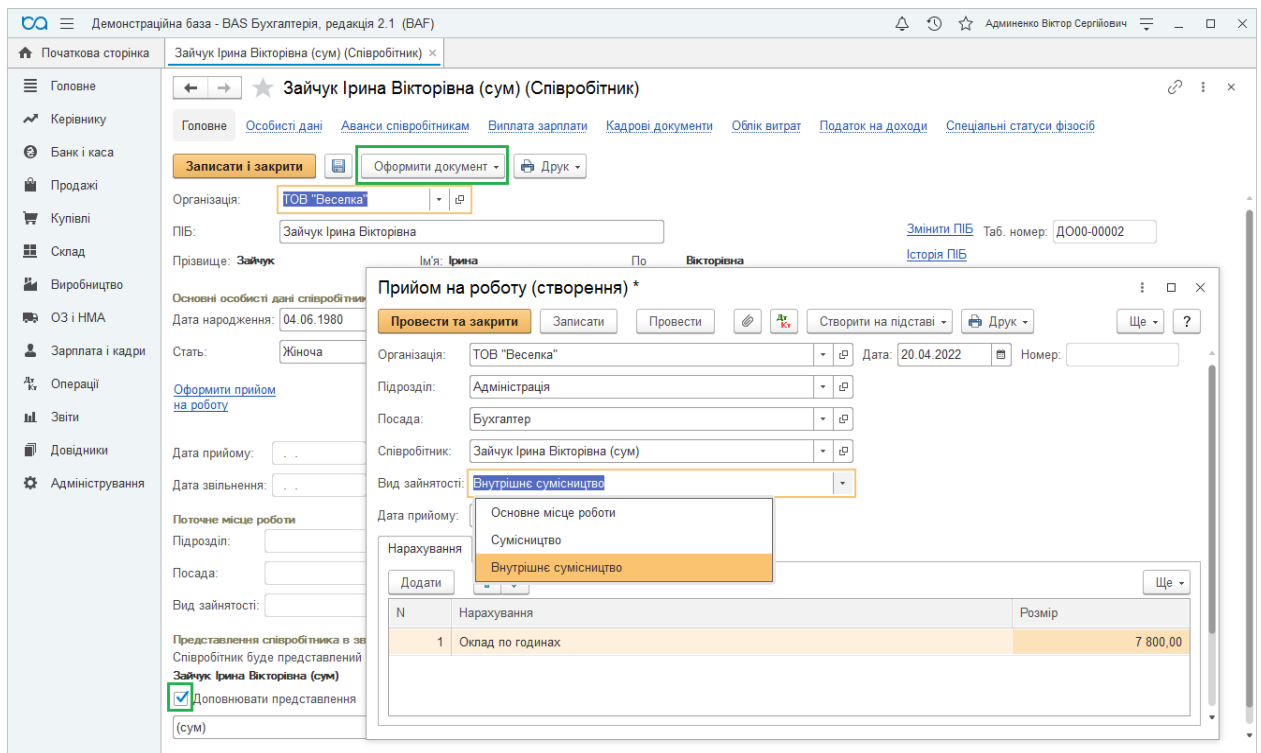


Рисунок 1.8. Інтерфейс роботи з даними клієнта в BAS ERP

- Облік руху кадрів і зайнятості персоналу організацій, включно з отриманням уніфікованих звітних форм і внутрішньої аналітичної звітності
- організація роботи зі штатним розкладом.
- Робота з договорами підряду.
- Облік відпрацьованого часу із застосуванням різних облікових методів.

← → **Начисление за первую половину месяца (создание) \***

Провести и закрыть Записать Провести

Еще ?

Месяц: Март 2020 Организация: Вертикаль ТОВ Дата: 13.03.2020 Номер:

Расчет первой половины месяца до: 15.03.2020

Заполнить Начислено: 16 617,29 ? Удержано: 3 240,38 ? Взносы: 3 335,47 ?

Начисления Удержания НДФЛ Взносы ФОТ

Добавить Найти... Отменить поиск Отмена исправлений Пересчитать сотрудника

N	Сотрудник	Подразде...	Начисление	Результат	Отработано (оплачено)	Норма времени	Показатели	Пер
1	Базарова Анна ...	Отдел продаж	Оплата по окладу	3 428,57	9,00 дн. 72,00 чс.	21,00 дн. 168,00 чс.	Оклад 8 000 Доля неполн. времени 1	01.0
2	Говорухин Глеб ...	Производ...	Оплата по окладу	2 500,00	8,00 дн. 96,00 чс.	16,00 дн. 192,00 чс.	Оклад 5 000 Доля неполн. времени 1	01.0
3	Говорухин Глеб ...	Производ...	Доплата за работу в ...	23,98	4,00 дн. 8,00 чс.	16,00 дн. 192,00 чс.	% допл. за вечерн. 10 Час. стоим. 29,97	01.0
4	Говорухин Глеб ...	Производ...	Доплата за работу в ...	191,81	8,00 дн. 32,00 чс.	16,00 дн. 192,00 чс.	% допл. за ночн. 20 Час. стоим. 29,97	01.0
5	Кудряцев Олег ...	Производ...	Оплата по окладу	2 500,00	8,00 дн. 96,00 чс.	16,00 дн. 192,00 чс.	Оклад 5 000 Доля неполн. времени 1	01.0
6	Кудряцев Олег ...	Производ...	Доплата за работу в ...	23,98	4,00 дн. 8,00 чс.	16,00 дн. 192,00 чс.	% допл. за вечерн. 10 Час. стоим. 29,97	01.0
7	Кудряцев Олег ...	Производ...	Доплата за работу в ...	191,81	8,00 дн. 32,00 чс.	16,00 дн. 192,00 чс.	% допл. за ночн. 20 Час. стоим. 29,97	01.0
8	Балашов Эдуард ...	Производ...	Сдельный заработок...	5 400,00	9,00 дн. 72,00 чс.		Час. тариф 75 Сдельно 0	01.0
9	Решетов Анатолий ...	Производ...	Оплата по окладу	2 357,14	9,00 дн. 72,00 чс.	21,00 дн. 168,00 чс.	Оклад 5 500 Доля неполн. времени 1	01.0

Рисунок 1.9. Інтерфейс відслідковування відпрацьованого часу у BAS ERP

- Розрахунок заробітної плати персоналу з використанням різних систем оплати праці: погодинної (в тому числі тарифної), відрядної та їхніх різновидів.
- Використання показників ефективності діяльності організації та безпосередньо конкретного співробітника під час розрахунку винагород за працю.
- Розрахунок утримань із заробітної плати, у тому числі за виконавчими документами.
- Нарахування інших доходів, не пов'язаних з оплатою праці та доходів у натуральній формі.
- Проведення взаєморозрахунків із персоналом у готівковій та безготівковій формі, управління заборгованістю перед співробітниками.
- Аналіз нарахованої заробітної плати з використанням внутрішньої аналітичної звітності.
- Отримання уніфікованих звітних форм.
- Обчислення регламентованих законодавством податків і відрахувань у фонд оплати праці.

- Формування регламентованої звітності із заробітної плати, як зведеної, так і персоніфікованої.

### 1.2.5. Порівняння розглянутих систем

Як можна бачити, розглянуті програмні продукти для управління персоналом мають схожий функціонал, кожен з них є конкурентним продуктом що здатен покрити потреби підприємства.

Для визначення найбільш придатної для розробки системи, з наявної у публічному доступі інформації, було складено порівняльну таблицю 1.1.

Таблиця 1.1

Порівняння описаних систем

	Оцінка системних вимог за 5-бальною шкалою	Ціна	Є своя СУБД	Є у "хмарі"
1С Підприємство	3	від 12300 грн.	+	-
SAP S/4 HANA Cloud	2	9.60\$ за користувача на рік	+	+
Microsoft Dynamics	3	120\$ за користувача на рік	+	-
BAS ERP	3	Від 50000 грн.	+	-

Загалом, можна виділити такі пункти:

- Оскільки розглянуто ERP від великих компаній, всі вони мають власні СУБД, а отже максимальну інтеграцію з СУБД.
- 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.

Таким чином, для подальшої розробки обрано ERP SAP S/4 HANA Cloud, оскільки вона працює “у хмарі”, а отже має найменші системні вимоги. Також, доволі багато матеріалів для розробки під цю систему знаходиться в вільному доступі.

### **1.3. Вимоги до проекрованої системи управління персоналом**

Система управління персоналом є важливою складовою управління підприємством, тож відносно проекрованої системи можна виділити такі вимоги:

- Система має підтримувати можливість відслідковування часу, відпрацьованого працівником.
- “Хмарна” система має працювати надійно та без збоїв.
- Оскільки в SAP S/4 HANA Cloud є стандартні способи фіксації відпрацьованого часу, розроблена система має позитивно відрізнятися від стандартних інструментів. Розробка має дозволяти фіксувати час швидше від стандарту.
- Розроблена система має підтримувати авторизацію користувачів, щоб уникнути фіксування часу неуповноваженими особами.
- Система має однозначно відносити записи часу до задачі, якої цей запис стосується
- Система має підтримувати додаткові функції відносно стандарту: додавання на задачу нових людей керівником проекту та можливість працівнику долучитись до проекту, який підтримує таку можливість.

## 1.4. Висновки до Розділу 1

Сучасне підприємство має чимало бізнес-процесів, обслуговування котрих вручну стає із ростом організації все складнішим. В зв'язку з цим, набули популярності ERP-системи, що дають можливість вирішувати чимало задач: облік товарів на складі, розрахунок зарплати робітників, підтримка документообігу підприємства, проведення транзакцій і чимало інших. Такі системи існують вже кілька десятків років, і весь цей час вони вдосконалювались, аби відповідати запитам користувачів.

Наразі на українському ринку існує чимало таких систем від різних виробників: 1С Підприємство, BAS ERP, SAP ERP, Microsoft Dynamics. Почавши свій шлях як великі програмні продукти, що потребували потужних серверів, поточні версії деяких таких систем почали підтримувати роботу “у хмарі” за допомогою технології хмарних обчислень. Такі версії не потребують від клієнта покупки та обслуговування серверу з встановленою ERP. Натомість, він отримує доступ до потужностей віддаленого сервера, за що платить певну абонентську плату. Такий підхід в довгостроковій перспективі дозволив поширити ERP серед клієнтів, що раніше не наважувались на це через витрати на апаратне забезпечення.

У цьому розділі було розглянуто та порівняно модулі управління персоналом деяких популярних ERP-систем. В результаті було визначено, що модулі надають схожі можливості, але, оскільки вони працюють в контексті свого програмного продукту, було обрано для розробки систему SAP S/4 HANA Cloud через те, що ця система працює “у хмарі” та має достатньо матеріалів щодо розробки під себе у публічному просторі.

Для розроблюваної системи було сформульовано ряд вимог, що можна звести до таких:

- Стабільна робота у “хмарі”.



- Розширений функціонал відносно стандартних інструментів відслідковування відпрацьованого часу.
- Інтерфейс розробки має надавати можливість швидше фіксувати відпрацьований час відносно стандартних інструментів відслідковування відпрацьованого часу.

## РОЗДІЛ 2

### ТЕХНОЛОГІЇ SAP S/4 HANA CLOUD ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ

#### 2.1. Архітектура SAP S/4 HANA Cloud

SAP S/4 HANA Cloud – система планування ресурсів підприємства, продукт компанії SAP. На відміну від звичайної SAP ERP працює в хмарі, та розповсюджується за моделлю абонентської плати, а не одноразового платежу. Користувач платить абонентську плату за базову конфігурацію системи, та додаткову плату за підключення додаткових модулів:

- 1) Управління фінансами (Finance)
- 2) Керування закупками (Sourcing & Procurement)
- 3) Управління поставками (Supply Chain)
- 4) Управління виробництвом (Manufacturing)
- 5) Управління дослідженнями (R&D Engineering)
- 6) Управління активами (Asset Management)
- 7) Управління послугами (Service)
- 8) Управління людськими ресурсами (Human Resources)
- 9) Управління професійними послугами (Industry)

При цьому, підписка надає доступ не виключно до системи SAP S/4 HANA Cloud, але й до ряду допоміжних ресурсів, що формують її оточення.

Система є хмарною версією системи планування ресурсів SAP S/4 HANA і має архітектуру, що представлена на рисунку 2.1.

Кафедра КІТ (47)				НАУ 23 30 30 000 ПЗ			
<i>Виконав</i>	Ковтун В.Є.			Технології SAP S/4 HANA Cloud для організації системи управління персоналом	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Райчев І. Е.				Д	26	9
<i>Консульт.</i>					УС-212М 122		
<i>Н- контроль.</i>	Райчев І. Е.						

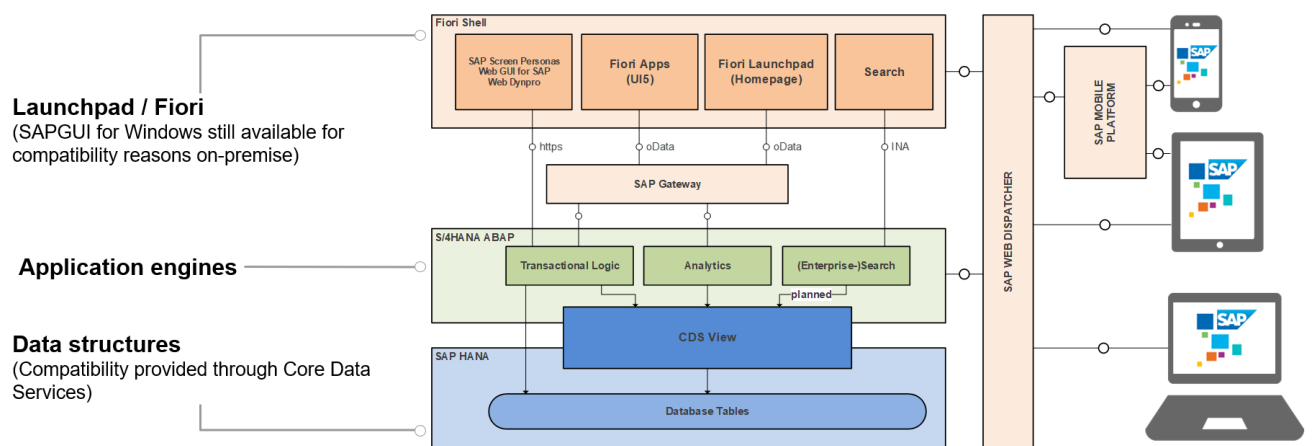


Рис. 2.1. Архітектура SAP S/4 HANA

На рисунку 2.1 можна побачити, що система складається з таких складових:

- 1) SAP Web Dispatcher – це програмне забезпечення, що слугує перемикачем між SAP-системою та мережею Інтернет. Тут приймається рішення про згоду чи відмову на виконання HTTP-запиту. Також він балансує навантаження між серверами для його рівномірного розподілення.
- 2) SAP FIORI – презентаційний рівень, відповідає за взаємодію з користувачем через графічний інтерфейс. Наразі взаємодія може відбуватись як через комп'ютери так і через мобільні пристрої – інтерфейс може формуватися під необхідний розмір.
- 3) SAP Gateway – відповідає за взаємодію користувача із об'єктами в системі. Ставить запити до об'єктів у системі на основі протоколу ODATA, що надає можливість взаємодіяти із системою із сторонніх додатків.
- 4) Application Engine – двигун додатків.
- 5) CDS views – розшифровується як Core Data Services. Представляє собою View – сталий SQL-вираз до бази даних. Також, підтримує ряд додаткових можливостей: додавання анотацій, встановлення зв'язків між різними View, розширення вже існуючих View.

- б) SAP HANA – єдина база даних, що підтримується системою. Це реляційна база даних, що підтримує роботу з OLAP та OLTP-транзакціями. При високій завантаженості системи також може виконувати обчислення.

## **2.2. 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 S/4 HANA Cloud. Надає можливість для об'єктно-орієнтованого програмування, синтаксис заснований на використанні ключових слів англійською. В основному, розробка для SAP S/4 HANA Cloud 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 ).

Під час компіляції програма перетворюється на байт-код, який потім виконується на віртуальній машині ABAP 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.

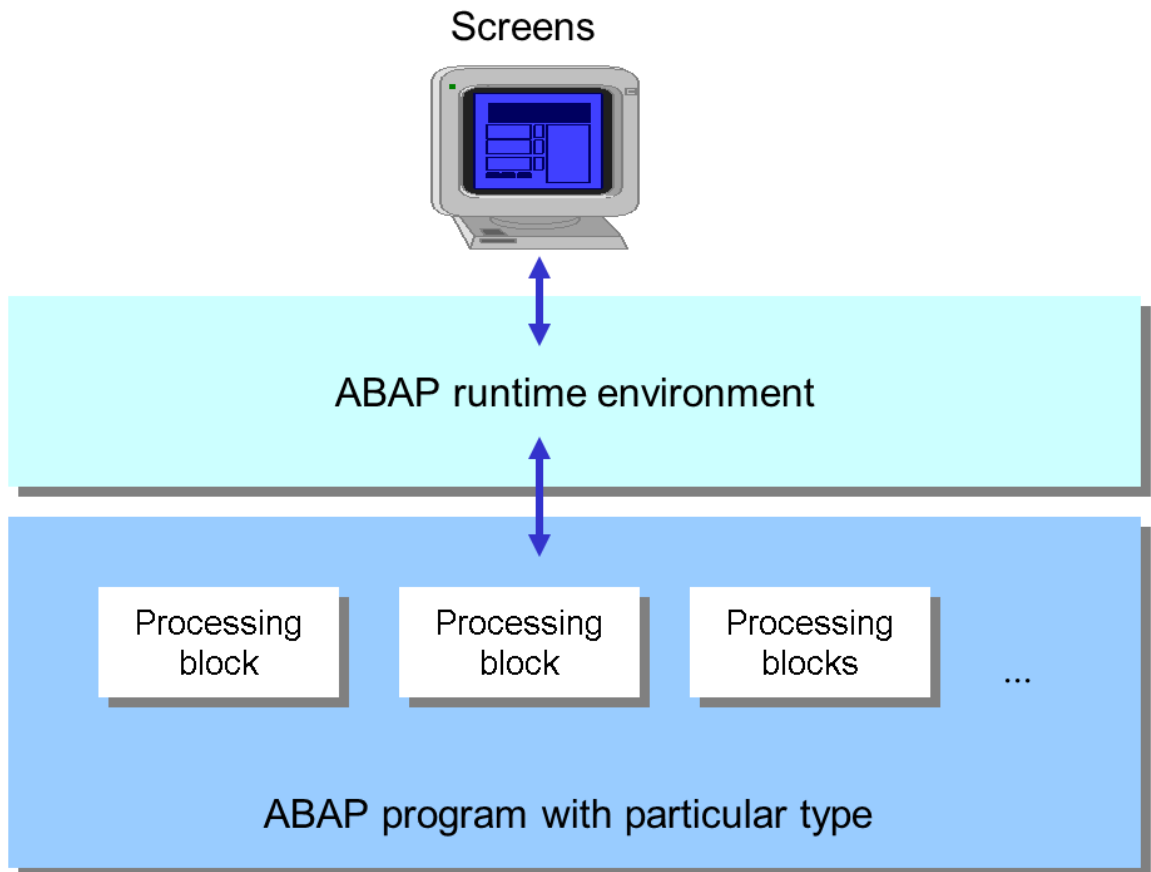


Рис. 2.2. Схема виконання програми на АВАР

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.

### 2.3. Технології розробки додатків для SAP S/4 HANA Cloud

Типовий додаток створюється за технологією RESTfull ABAP Programming model (скорочено RAP), за якою додатки мають будову, яку можна побачити на рисунку 2.3:

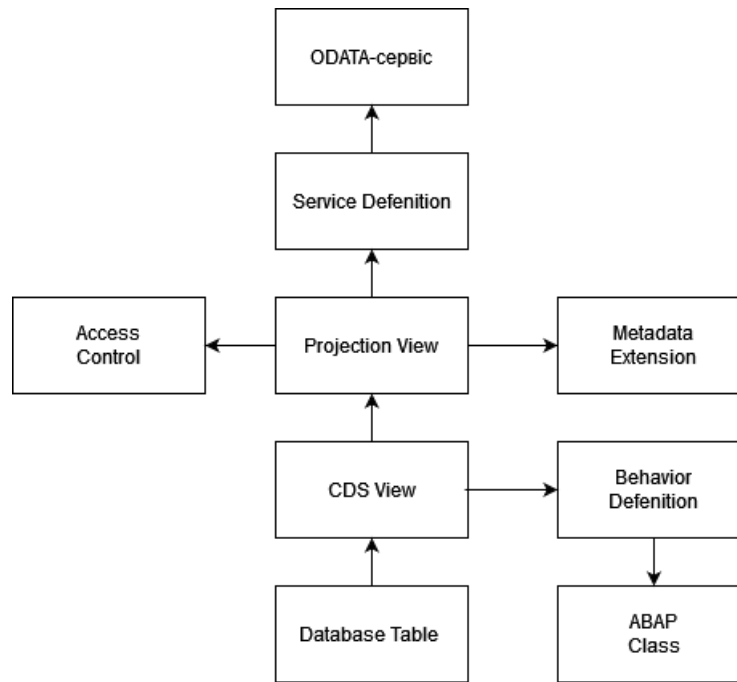


Рис. 3.3. Типова структура RAP-додатку

З рисунку 3.3. видно, що типова структура RAP-додатку складається з таких елементів:

- 1) Database Table – таблиця у базі даних. В ній зберігаються дані додатку.
- 2) CDS View – сталий SQL-вираз із можливістю конфігурації окремих полів (наприклад їх приховуванням). Може мати як джерело даних у вигляді таблиці чи іншого View, так і не мати джерела даних взагалі (і бути в такому разі абстрактним).
- 3) Behavior Definition – об’єкт, що визначає поведінку CDS View. Тут можна прописати можливі дії (actions), що може виконувати View, а також вказати клас, в якому буде імплементовано поведінку.
- 4) ABAP Class – клас, що написаний мовою ABAP. Тут прописують код, що виконуватиметься під час роботи із CDS View.
- 5) Projection View – опціональний елемент, спеціальний тип View, що є надбудовою над CDS View, і використовується для налаштувань контролю за доступом до інформації (наприклад можна позначити її як таку що доступна лише для читання) та для роботи з певними

властивостями полів – анотаціями (завдяки ним, наприклад, можна налаштувати швидкий ввід для поля).

- 6) Metadata Extension – опціональний елемент, об’єкт в якому прописуються правила виводу CDS View на екран (наприклад розмір комірки, що має бути відведений під певне поле, порядок виведення полів, призначення їм псевдонімів і т.п.).
- 7) Access Control – опціональний елемент, модифікує SELECT-вираз із View для контролю доступу до інформації у ній (наприклад щоб дивитись її могли лише конкретні користувачі).
- 8) Service Defenition – об’єкт, що визначає View, на основі яких буде згенеровано сервіс для додатку.
- 9) Service Binding – об’єкт, що створюється на основі Service Defenition, генерує інтерфейс додатку на основі Metadata Extension та URL, за яким до додатку можна буде отримати доступ.

## **2.4. Огляд стандартних засобів управління персоналом у SAP S/4 HANA Cloud**

Розглянемо стандартні засоби управління персоналом у SAP S/4 HANA Cloud в контексті управління задачами.

Задачі (Service Order) складаються з підзадач (Service Order Item), на які, в свою чергу, створюються документи підтвердження виконання завдання (Service Confirmation). Після створення підзадачі, її статус необхідно змінити на Released, і тільки після цього її виконання можна буде підтвердити. При цьому, підзадачі характеризуються наявністю витрат матеріалів – людського ресурсу, певних запасів зі складу тощо. За роботу з задачами відповідає підпрограма Manage Service Orders (рис. 3.4).

The screenshot displays the SAP Manage Service Orders interface. At the top, there are search filters for Service Order, Description, Sold-To Party, and Status. Below the filters, there are tabs for different order statuses: All Service Orders (30), Unassigned Orders (0), Open Orders (0), Without Confirmations (5), With Open Confirmations (4), and Billable Orders (14). The main table lists 10 service orders with the following data:

ID	Description	Life Cycle Status	Sold-To Party	Requested Start	Actions
800000081	Test	🟢-🟡-🔴-🟢	Inlandskunde DE 1 (10100001)	15.11.2023	🔍 ⚙️ 📄 ➤
800000080	test	🟢-🟢-🟢-🟢	Inlandskunde DE 1 (10100001)	09.11.2023	🔍 ⚙️ 📄 ➤
800000070	test	🟢-🟡-🔴-🟢	Inlandskunde DE 1 (10100001)	27.07.2023	🔍 ⚙️ 📄 ➤
800000062	test	🟢-🟢-🟢-🟢	Inlandskunde DE 1 (10100001)	25.07.2023	🔍 ⚙️ 📄 ➤
800000061	test	🟢-🟡-🔴-🟢	Inlandskunde DE 1 (10100001)	20.07.2023	🔍 ⚙️ 📄 ➤
800000060	test	🟢-🟡-🔴-🟢	Inlandskunde DE 1 (10100001)	19.07.2023	🔍 ⚙️ 📄 ➤
800000052	New Service Order	🟢-🟡-🔴-🟢	Inlandskunde DE 1 (10100001)	26.06.2023	🔍 ⚙️ 📄 ➤
800000051	Solution Order for Service	🟢-🟢-🟢-🟢	Inlandskunde DE 1 (10100001)	26.06.2023	🔍 ⚙️ 📄 ➤
800000044	Solution Order for Service	🟢-🟢-🟢-🟢	Inlandskunde DE 1 (10100001)	22.06.2023	🔍 ⚙️ 📄 ➤

Рис. 3.4. Підпрограма Manage Service Orders

В той же час робітник витрачає час на виконання задачі, за який йому треба заплатити. Витрачений час він фіксує у підпрограмі Notify Missing Time (рис. 3.5).

The screenshot displays the SAP Notify Missing Time interface. It features search filters for Time Period (set to Current Month), Missing Hours (%), Project ID, Work Package ID, and Work Item ID. Below the filters, there is a table titled 'Project Members' with the following columns: Name, Project, Work Package, Work Item, and Activity Type. The table is currently empty, and a message at the bottom states: "No data found. Try adjusting the search or filter criteria."

Рис. 3.5. Підпрограма Notify Missing Time

При цьому, фіксуючи час, відбувається робота з такою ієрархією об'єктів: Project -> Work Package -> Project Role (посада на проєкті) -> Timesheet (набір записів про фіксування відпрацьованого часу відносно певної посади). При цьому, кожен учасник проєкту може фіксувати відпрацьований час лише за свою посаду (окрім керівника проєкту). Також, існує можливість зафіксувати витрачений час



на певній посаді людиною поза штатом проекту, за умови що для цієї посади доступна така опція.

Таким чином можна бачити, що існують 2 типи об'єктів, робота з якими взаємопов'язана: наприклад, керівник проекту підтверджує виконання підзадачі, а потім відмічає час, витрачений робітником на її виконання. При цьому, робота з цими об'єктами ведеться у 2 різних підпрограмах: Notify Missing Time та Manage Service Orders. Окрім того що робота з ними повсякденна, додаткові незручності створює те, що Manage Service Orders є підпрограмою підпрограми – Manage Solution Orders, через що її не можна відкрити одразу з головної сторінки системи.

Логічно було б об'єднати функціонал цих двох підпрограм у єдиний додаток, що дозволило б спростило роботу з ними, що й буде зроблено надалі в практичній частині.

## 2.5. Висновки до розділу 2

При виконанні 2 розділу було досліджено елементи системи SAP S/4 HANA Cloud, а саме:

- 1) Розглянуто архітектуру системи.
- 2) Розглянуто основну мову програмування в системі – ABAP.
- 3) Розглянуто технології розробки додатків для SAP S/4 HANA Cloud.
- 4) Розглянуто стандартні засоби управління персоналом в контексті виконання їм завдань та фіксування відпрацьованого часу.

Під час огляду стандартних засобів управління персоналом, був зроблений висновок, що можна спростити роботу користувачів з інструментами фіксування відпрацьованого часу та управлінням виконання завдань, звівши їх в один додаток – оскільки робота з цими інструментами взаємопов'язана та повсякденна.

В той же час створювана система управління персоналом має виконувати всі основні функції, доступні у стандартних засобах, а саме:

- 1) Фіксування відпрацьованого часу
- 2) Додавання на проект людей, що знаходяться поза штатом
- 3) Управління статусом підзавдань
- 4) Підтвердження виконання підзавдань
- 5) Відміна виконання підзавдань
- 6) Призначення відповідального за виконання завдання

Також система має підтримувати налаштування конфіденційності, користувачі мають бачити і працювати лише із своїми завданнями, і лише керівникам проектів та відповідальним за виконання завдань система має давати змогу працювати із відповідними об'єктами в повній мірі (підтверджувати виконання завдання іншою людиною, фіксувати час за своїх робітників і т.д.)

**РОЗДІЛ 3**  
**ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ**  
**УПРАВЛІННЯ ПЕРСОНАЛОМ**

**3.1. Підготовка робочого простору**

Для початку розробки інформаційної системи управління персоналом, потрібно підготувати робочий простір в якому буде проводитись вся розробка. Наразі розробку для хмарних SAP-систем можна вести у ABAP Development Tools, що є плагіном для середовища розробки Eclipse IDE. Використовуватись може як для Windows, так і для Linux, та MacOS.

Для початку перейдемо на сайт розробників та завантажимо середовище розробки, показане на рисунку 3.1.

Кафедра КІТ (47)				НАУ 22 30 30 000 ПЗ			
<i>Виконав</i>	Ковтун В.Є.			Програмна реалізація інформаційної системи управління персоналом	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Райчев І. Е.				Д	35	72
<i>Консульт.</i>					35		
<i>Н- контроль.</i>	Райчев І. Е.				УС-212М 122		



Register for TheiaCon 2023!

Join us November 15-16 for our virtual conference on the Eclipse Theia IDE ecosystem

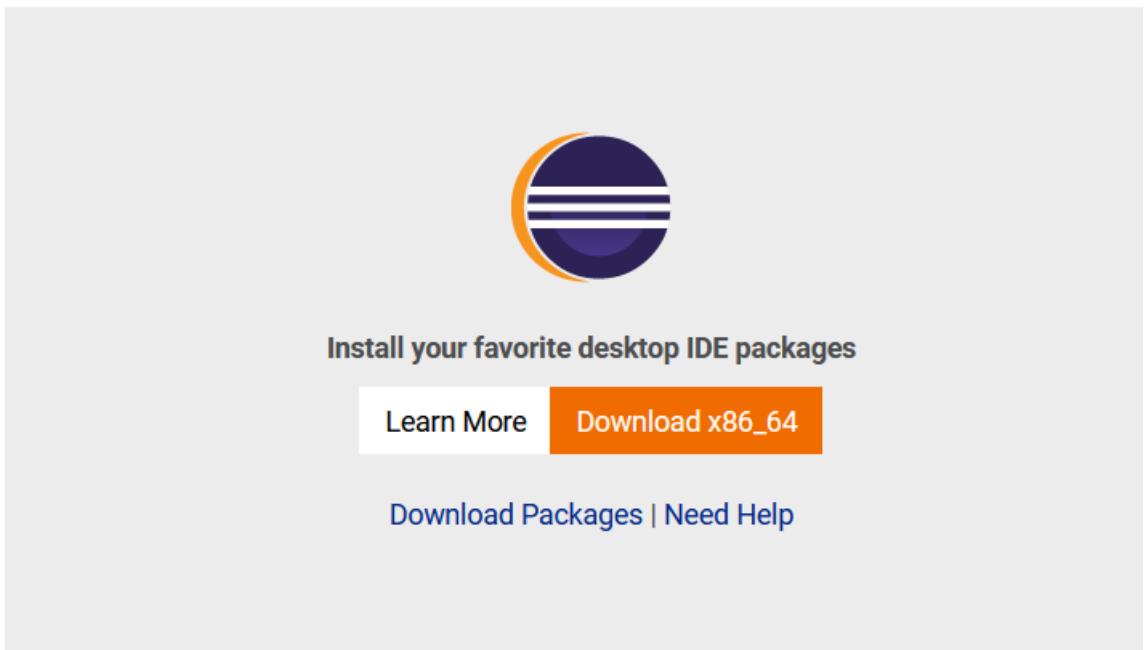


Рис. 3.1. Редактор коду Eclipse IDE

Після завантаження відкриваємо файл та розпочинаємо процес встановлення середовища розробки Eclipse IDE, як це показано на рисунку 3.2:

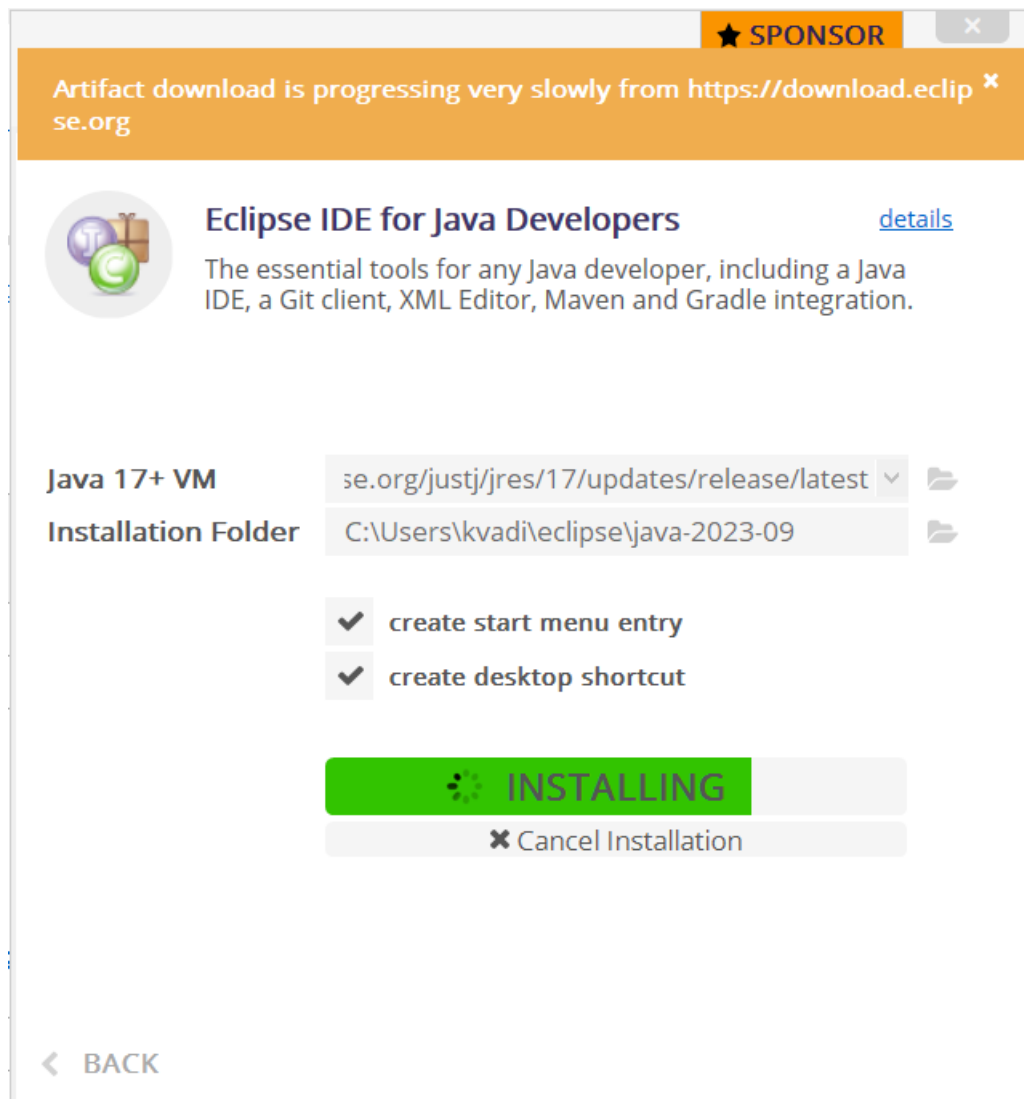


Рисунок 3.2. Встановлення середовища розробки Eclipse IDE

Тепер необхідно встановити плагін ABAP Development Tools. Для цього треба зайти в Eclipse IDE, та піти шляхом Help-> Install new software. В стрічці для URL необхідно вставити посилання на останню версію плагіну, як це показано на рисунку 3.3.

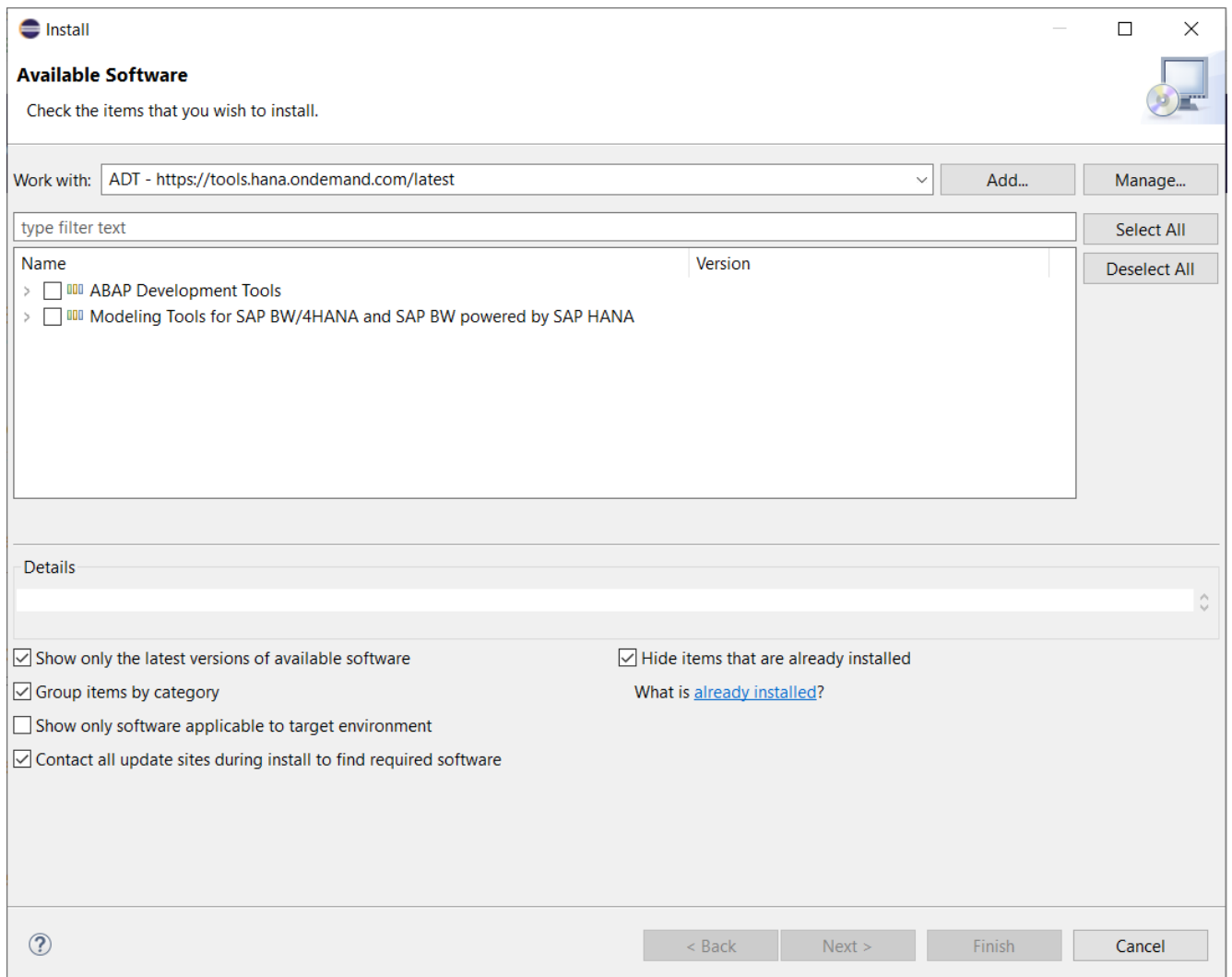
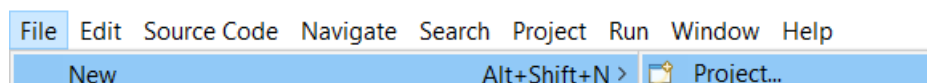


Рисунок 3.3. Встановлення ABAP Development Tools

Після натискання на кнопку Next почнеться встановлення плагіну.

Тепер необхідно під'єднатися до системи, в якій буде вестись розробка. Для цього треба піти шляхом File ->New ->Project ->ABAP Cloud Project (рис.3.4) та у з'явившомуся вікні ввести посилання на систему до якої треба підключитися (рис.3.5).



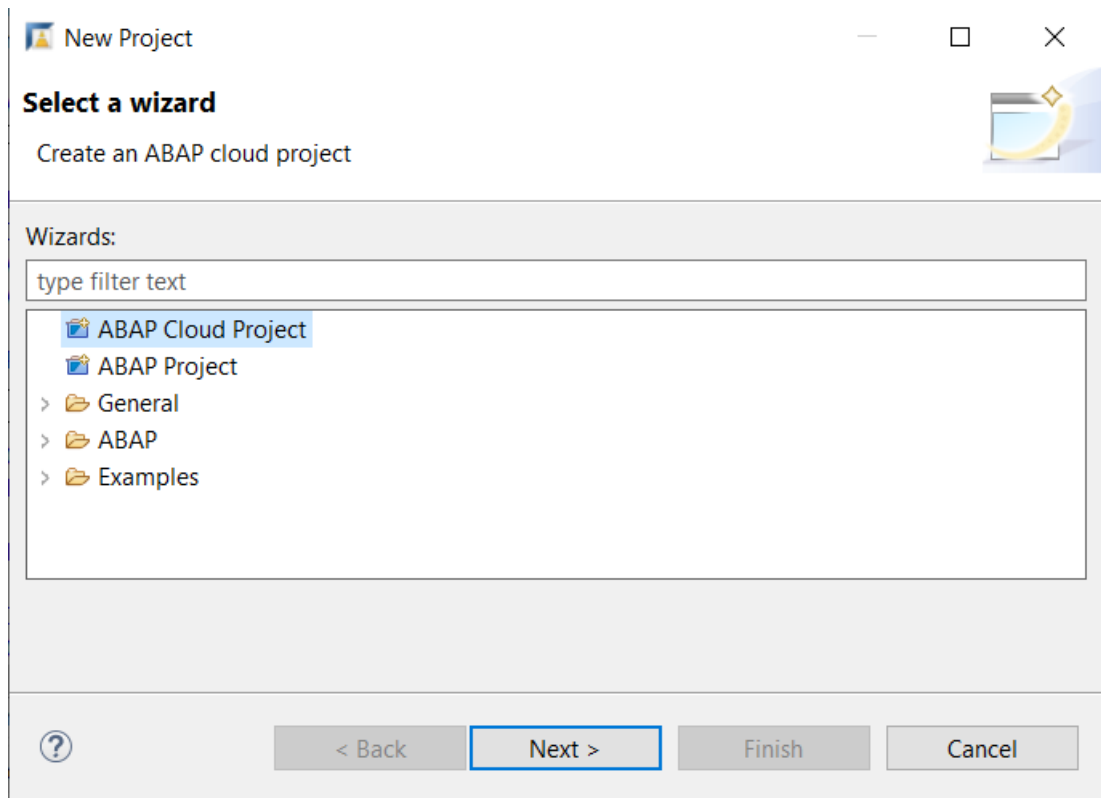


Рис.3.4. Вікно створення нового проекту

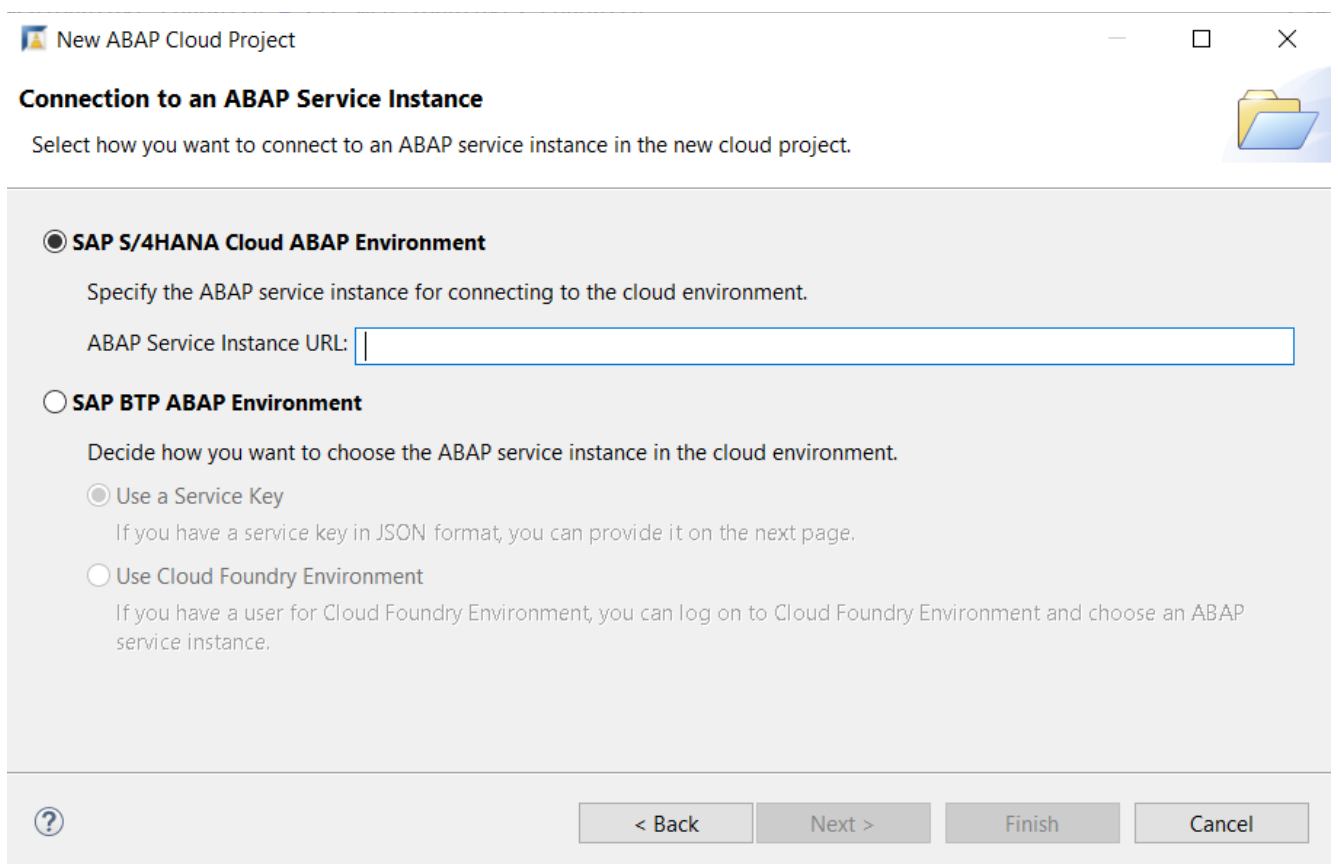


Рис.3.5. Вікно підключення до хмарної SAP-системи

Після вікна авторизації, підключення до системи виконано (рис 3.6). Тепер для входу в систему необхідно буде лише кожного разу авторизуватись при підключенні.

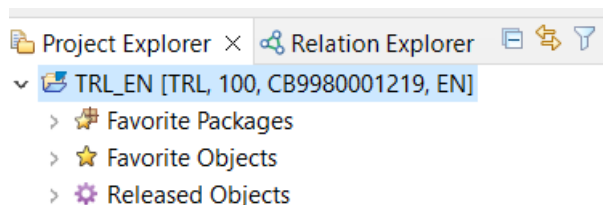


Рисунок 3.6. Систему підключено до середовища розробки

### 3.2. Розробка моделі даних

Для підтримки трирівневої архітектури у SAP використовуються Transport Requests – об’єкти, до яких “прив’язуються” об’єкти у системі для того щоб одразу перемістити всю розробку на інший рівень (наприклад із рівня розробки на рівень тестування).

Об’єкти розроблюються у контексті таких Transport Requests, тому щоб вести розробку необхідно його створити. Для цього необхідно пройти шляхом: Wndow -> Show View -> Other. У з’явившомуся вікні необхідно обрати “Transport Organizer” (рис.3.7).



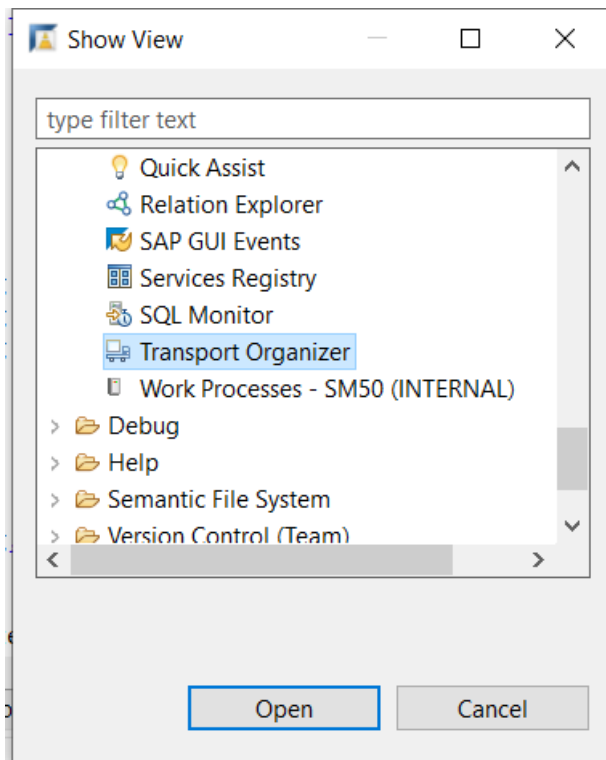


Рис. 3.7. Перехід у транзакцію по роботі з Transport Requests

Для того щоб додати новий об'єкт треба натиснути на New Transport Request (рис. 3.8). У з'явившемся вікні необхідно описати призначення об'єкта (рис. 3.9).

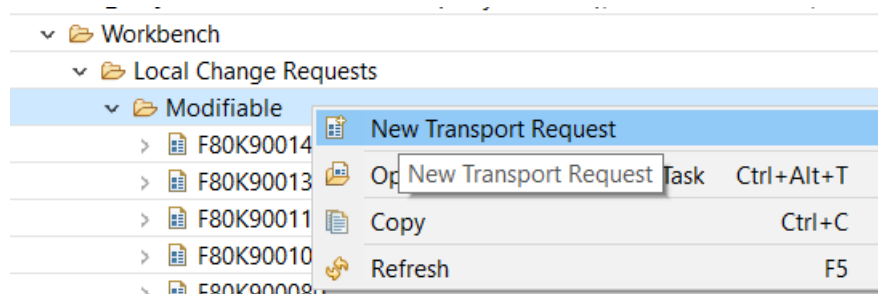


Рис. 3.8. Створення нового Transport Request

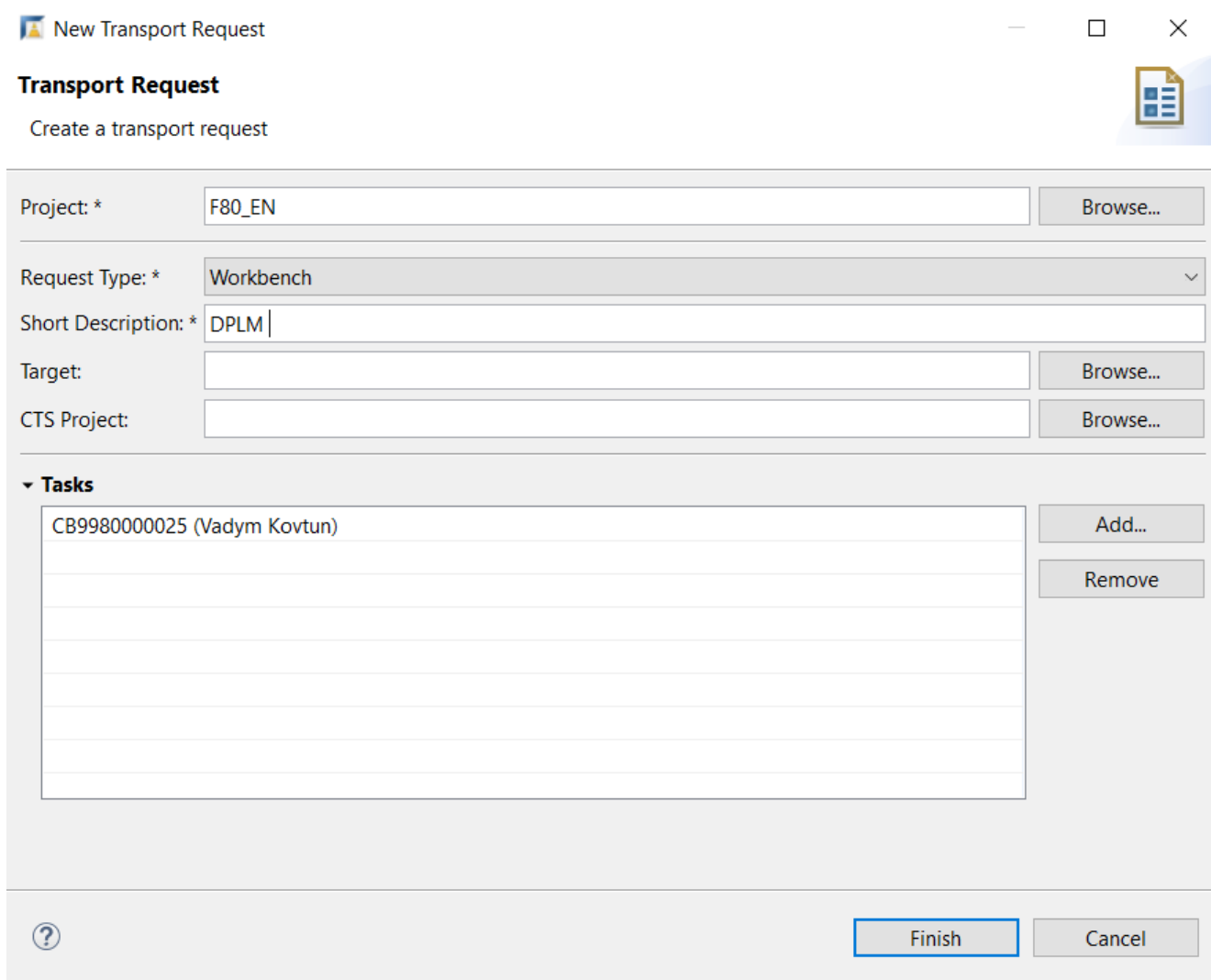
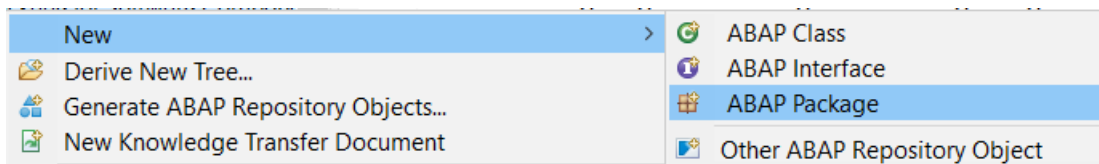


Рис.3.9. Вікно редагування створюваного Transport Request

Після натискання на Finish, об'єкт буде створено.

Об'єкти у SAP-системі групуються по пакетам розробки, і не існують поза ними. Для того щоб створити пакет для розробки необхідно піти шляхом: New -> ABAP Package, після чого ввести назву пакета (рис. 3.10).



New ABAP Package

**ABAP Package**

Create an ABAP package

Project: \* F80\_EN Browse...

Name: \*

Description: \*

Original Language: EN

Add to favorite packages

Superpackage: ZLOCAL Browse...

Package Type: \* Development

? < Back Next > Finish Cancel

Рис. 3.10. Створення пакета розробки

Об'єкти у SAP бувають стандартними та користувацькими. Стандартні розроблює SAP, також вони за замовчуванням постачаються разом із системою. Користувацькі об'єкти розробляються користувачем для задоволення деяких потреб, що не задовольняються стандартним функціоналом. На відміну від стандартних об'єктів, назви користувацьких мають починатися із латинських букв Z або Y.

Після введення назви та опису, з'явиться вікно для вибору Transport Request, до котрого треба "прив'язати" створюваний об'єкт (рис. 3.11).

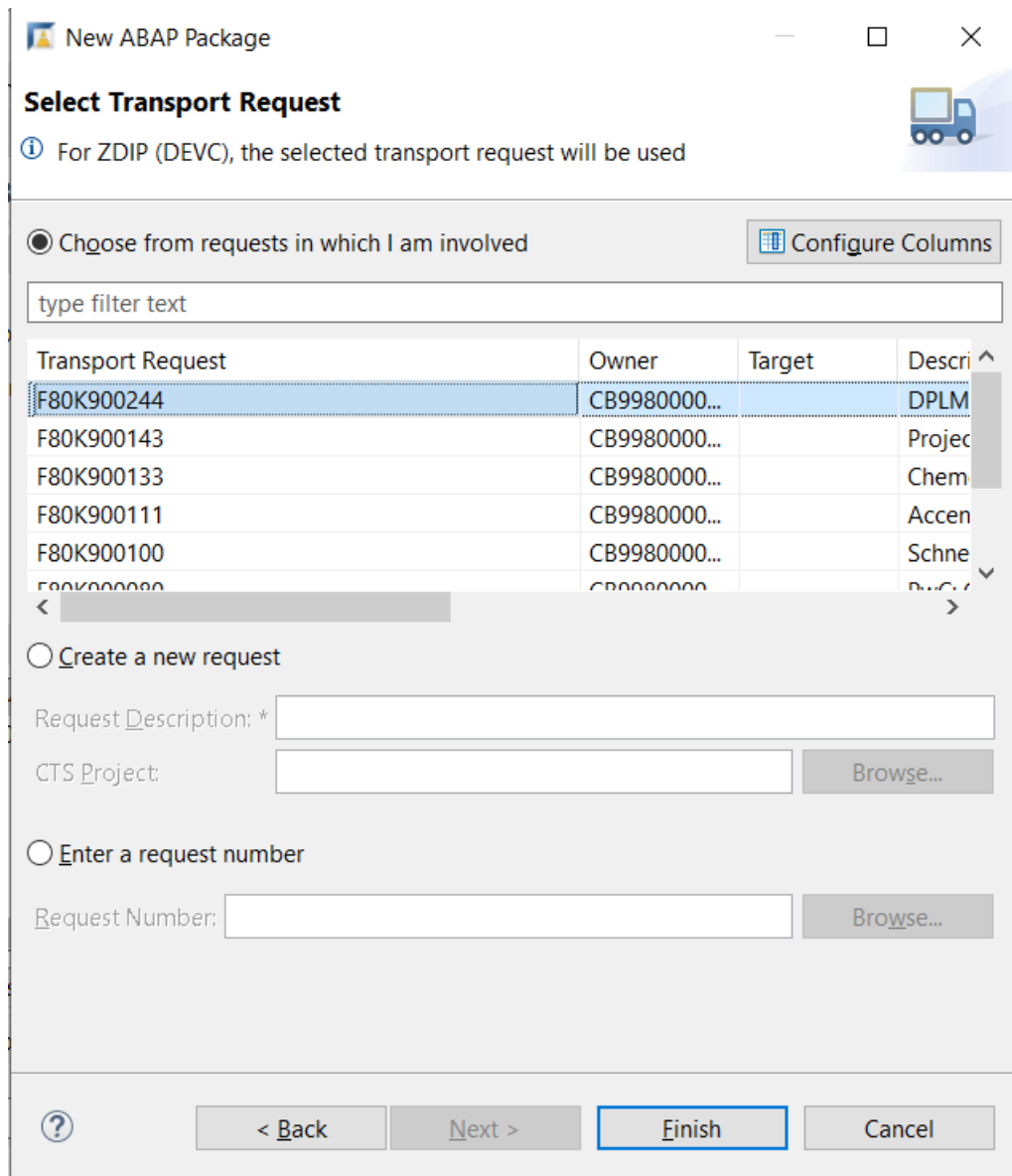


Рис. 3.11. Вікно вибору Transport Request

Тепер необхідно розробити модель даних для створюваної системи. Спочатку необхідно створити таблиці. Оскільки використовуватимуться дані із вже створених стандартних таблиць, власні таблиці необхідні лише для зберігання тимчасових копій об'єктів. Такі копії називаються Draft.

Щоб створити таблицю необхідно натиснути New -> Other ABAP Repository Object та у з'явившомуся вікні (рис. 3.12) обрати Database Table, після чого необхідно ввести назву та опис таблиці (рис. 3.13.) та обрати Transport Request. Після цього, відкриється вікно редагування коду таблиці. Таким чином створено 3 таблиці (рис. 3.14, 3.15, 3.16, 3.17, 3.18).

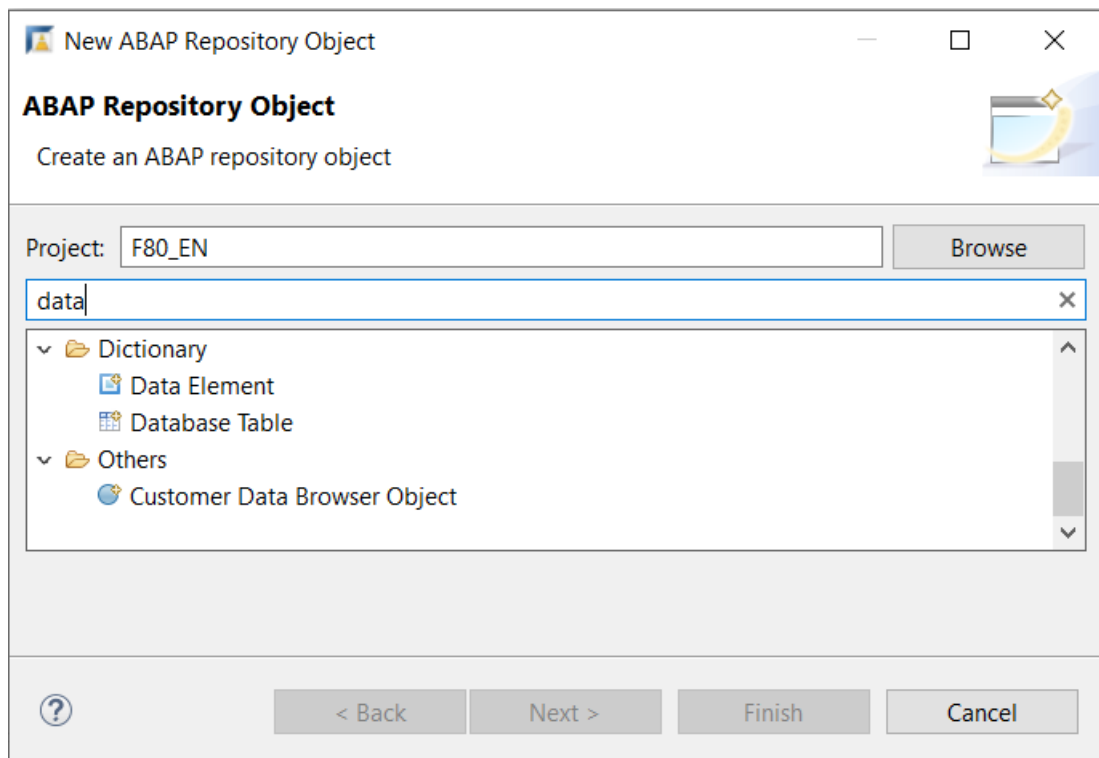


Рис. 3.12. Створення нової Database Table

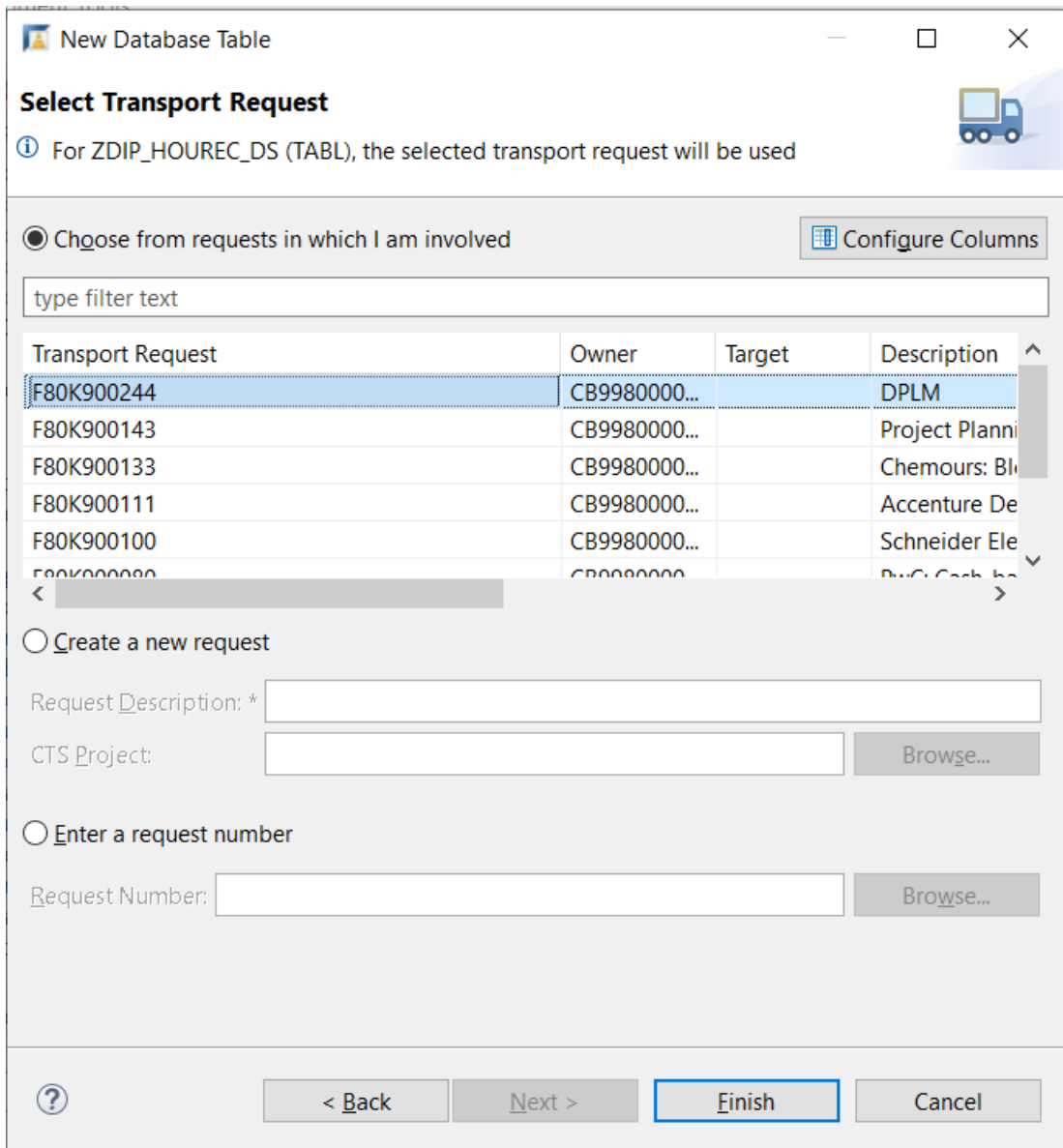


Рис.3.13. Вікно обрання Transport Request

```

1 @EndUserText.label : 'Draft table for ZJTSC_I_WorkTP'
2 @AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #RESTRICTED
6 define table zjtsc_work_dft {
7
8   key objectuuid      : abap.raw(16) not null;
9   type                : abap.char(1);
10  id                  : abap.char(24);
11  description         : abap.char(60);
12  startdate          : abap.dats;
13  enddate            : abap.dats;
14  lastchangedatetime : timestamp;
15  servicedocumentstatus : abap.char(4);
16  wbselementinternalid : abap.numc(8);
17  responsibleemployee : abap.char(10);
18  @Semantics.quantity.unitOfMeasure : 'zjtsc_work_dft.quantityunit'
19  plannedquantity    : abap.quan(13,3);
20  @Semantics.quantity.unitOfMeasure : 'zjtsc_work_dft.quantityunit'
21  actualquantity     : abap.quan(13,3);
22  quantityunit       : abap.unit(3);
23  hideproject        : abap.char(1);
24  processingstatus   : abap.char(2);
25  projectmanager     : abap.raw(16);
26  hideserviceorder   : abap.char(1);
27  projectmanagername : abap.char(80);
28  responsible         : abap.char(80);
29  "%admin"           : include sych_bdl_draft_admin_inc;|
30
31 }

```

Рис. 3.14. Таблиця для зберігання шаблонів проектів

```

1 @EndUserText.label : 'Draft table for ZJTSC_I_RoleTP'
2 @AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #RESTRICTED
6 define table zjtsc_roles_dft {
7
8   key wbelementinternalid : abap.numc(8) not null;
9   key role : lstar not null;
10  key workitem : abap.char(10) not null;
11  key person : abap.numc(8) not null;
12  workpackageuid : abap.raw(16);
13  workpackage : abap.char(24);
14  projectuuid : abap.raw(16);
15  assignmentstatus : abap.char(1);
16  deliveryorganization : abap.char(5);
17  personnumber : abap.char(10);
18  startdate : abap.dats;
19  enddate : abap.dats;
20  unstaffed : abap_boolean;
21  "%admin" : include sych_bdl_draft_admin_inc;
22
23 }

```

Рис. 3.15. Таблица для зберігання шаблонів посад

```

@EndUserText.label : 'Draft table for ZJTSC_I_HoursRecordedTP'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table zjtsc_hourec_d {

  key timesheetrecord : abap.char(12) not null;
  personworkagreement : abap.numc(8);
  personnumber : abap.char(10);
  timesheetdate : abap.dats;
  wbelementinternalid : abap.numc(8);
  role : lstar;
  workitem : abap.char(10);
  @Semantics.quantity.unitOfMeasure : 'zjtsc_hourec_d.hoursunitofmeasure'
  recordedhours : abap.quan(4,2);
  @Semantics.quantity.unitOfMeasure : 'zjtsc_hourec_d.hoursunitofmeasure'
  decimalrecordedhours : abap.quan(4,2);
  hoursunitofmeasure : abap.unit(3);
  controllingarea : kokrs;
  timesheetnote : abap.char(40);
  projectuuid : abap.raw(16);
  "%admin" : include sych_bdl_draft_admin_inc;

}

```

Рис. 3.16. Таблица для зберігання шаблонів відпрацьованих годин



```

1 @EndUserText.label : 'Draft table for ZJTSC_I_ServiceConfirmItemTP'
2 @AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #RESTRICTED
6 define table zjtsc_sconf_dft {
7
8   key serviceconfirmation      : abap.char(10) not null;
9   key serviceconfirmationitem : abap.numc(6) not null;
10  serviceorderuid              : abap.raw(16);
11  serviceorder                 : abap.char(10);
12  serviceorderitem             : abap.numc(6);
13  serviceconfirmationdescription : abap.char(40);
14  servicedocumentdescription  : abap.char(40);
15  serviceconfirmationisfinal   : abap.char(1);
16  isfinal                      : boolean;
17  serviceconfirmationitemuuid  : abap.raw(16);
18  serviceorderitemuuid        : abap.char(32);
19  originallyrequestedproduct    : abap.char(54);
20  product                      : abap.char(40);
21  @Semantics.quantity.unitOfMeasure : 'zjtsc_sconf_dft.servicedocitemquantityunit'
22  srvcdocitmrequestedquantity   : abap.quan(15,3);
23  @Semantics.quantity.unitOfMeasure : 'zjtsc_sconf_dft.servicedocitemquantityunit'
24  srvcdocitmdeliveredquantity  : abap.quan(15,3);
25  @Semantics.quantity.unitOfMeasure : 'zjtsc_sconf_dft.servicedocitemquantityunit'
26  srvcdocitmreleasedquantity    : abap.quan(15,3);
27  @Semantics.quantity.unitOfMeasure : 'zjtsc_sconf_dft.servicedocitemquantityunit'
28  cumulativeconfirmedquantity   : abap.quan(13,3);
29  servicedocitemquantityunit    : abap.unit(3);
30  srvcdocitembasequantityunit  : abap.unit(3);
31  serviceteam                   : abap.char(10);
32  responsibleemployee           : abap.char(10);
33  serviceemployee               : abap.char(10);

```

Рис. 3.17. Таблица для зберігання шаблонів підтвердження завдань

```

@EndUserText.label : 'Draft table for ZJTSC_I_ServiceOrderItemTP'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table zjtsc_sitem_dft {

    key serviceorder          : abap.char(10) not null;
    key serviceorderitem     : abap.numc(6) not null;
    serviceorderdescription  : abap.char(40);
    serviceorderitemdescription : abap.char(40);
    serviceorderitemuuid     : abap.raw(16);
    serviceorderuuid         : abap.raw(16);
    product                  : abap.char(40);
    servicedocitemcategory   : abap.char(4);
    @Semantics.quantity.unitOfMeasure : 'zjtsc_sitem_dft.servicedocitemquantityunit'
    srvcdocitmrequestedquantity : abap.quan(15,3);
    @Semantics.quantity.unitOfMeasure : 'zjtsc_sitem_dft.servicedocitemquantityunit'
    srvcdocitmdeliveredquantity : abap.quan(15,3);
    @Semantics.quantity.unitOfMeasure : 'zjtsc_sitem_dft.servicedocitemquantityunit'
    srvcdocitmreleasedquantity  : abap.quan(15,3);
    @Semantics.quantity.unitOfMeasure : 'zjtsc_sitem_dft.servicedocitemquantityunit'
    cumulativeconfirmedquantity : abap.quan(13,3);
    servicedocitemquantityunit  : abap.unit(3);
    serviceteam                 : abap.char(10);
    responsibleemployee         : abap.char(10);
    serviceemployee             : abap.char(10);
    contactperson               : abap.char(10);
    servicedocumentitemstatus  : abap.char(4);
    servicedocumentitemhaserror : abap.char(1);
    @Semantics.quantity.unitOfMeasure : 'zjtsc_sitem_dft.servicedocitemquantityunit'
    servicedocumentitemquantity : abap.quan(13,3);
    @Semantics.quantity.unitOfMeasure : 'zjtsc_sitem_dft.servicedocitemquantityunit'
}

```

Рис. 3.18. Таблиця для зберігання шаблонів завдань

Оскільки дані планується брати із стандартних таблиць – буде доцільно зберігати їх у вигляді CDS View – сталого SQL-виразу.

Щоб створити CDS View необхідно піти шляхом: New ->Other ABAP Repository Object -> Data Defenition (рис. 3.19).

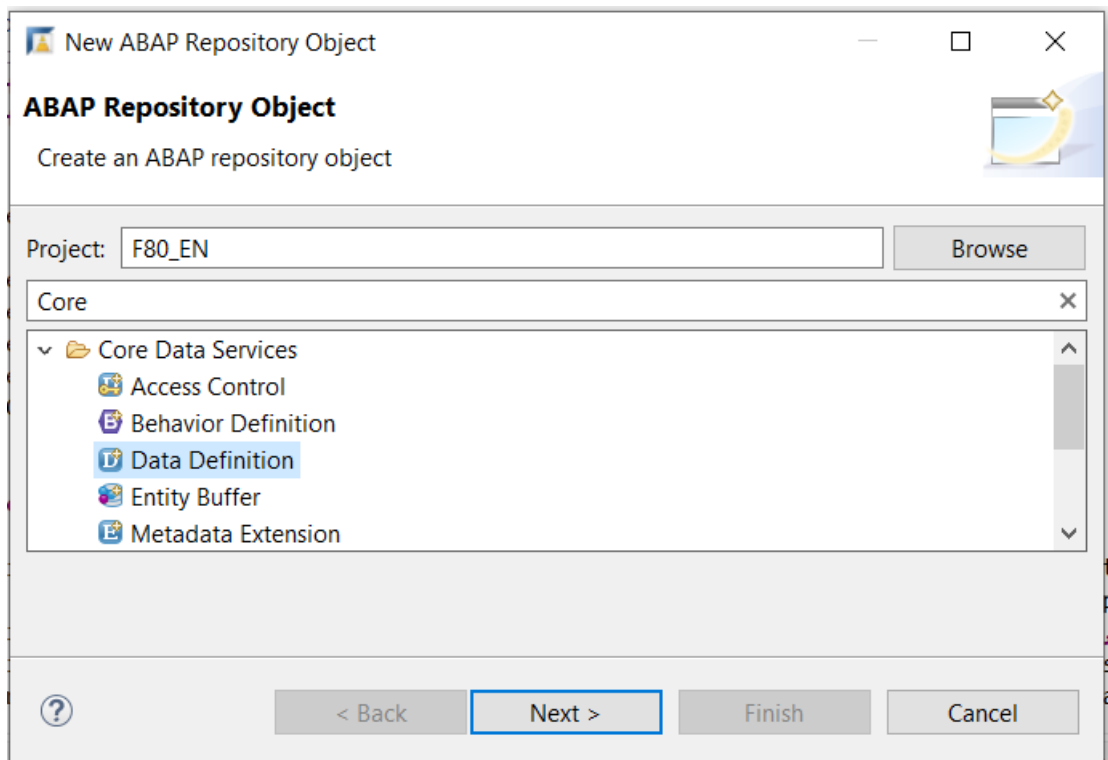


Рис.3.19. Створення CDS view

Для роботи з фіксуванням часу необхідні наступні відомості:

- 1) Відомості про робітників
- 2) Відомості про поточні проекти
- 3) Відомості про необхідні позиції на цих проектах
- 4) Відомості про запланований бюджет часу на проектах
- 5) Відомості про поточну кількість відпрацьованих годин на позиціях проектів
- 6) Відомості про поточні завдання
- 7) Відомості про поточні підтвердження завдань

З огляду на це було створено наступні об'єкти:

- 1) ZJTSC\_I\_HoursAssigned – для зберігання даних про бюджет часу на проектах за посадами (рис. 3.20).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Role Assigned Hours'
7 @Metadata.ignorePropagatedAnnotations: true
8 @ObjectModel.usageType:{
9   serviceQuality: #X,
10  sizeCategory: #S,
11  dataClass: #MIXED
12 }
13 define view entity ZJTSC_I_HoursAssigned
14 as select from I_ProjDmndRsceAssgmtDistr
15 association [1] to ZJTSC_I_Role as _Role on $projection.ResourceAssignmentUUID = _Role.Resour
16 association [1] to ZJTSC_I_Role as _RoleProjectDemand on $projection.ProjectDemandUUID = _RoleProjectDemar
17 {
18   key ProjDmndRsceAssgmtDistrUUID,
19     ProjDmndRsceAssgmtUUID as ResourceAssignmentUUID,
20     ProjectDemandUUID,
21     ProjDmndRsceAssgmtDistrYearVal as PeriodYear,
22     ProjDmndRsceAssgmtDistrPerdVal as PeriodMonth,
23     @Semantics: { quantity : {unitOfMeasure: 'HoursAssignedUnit'} }
24     ProjDmndRsceAssgmtDistrQty as HoursAssigned,
25     ProjDmndRsceAssgmtDistrQtyUnit as HoursAssignedUnit,
26     ProjDmndRsceAssgmtDistrStrtDte,
27     ProjDmndRsceAssgmtDistrEndDte,
28     ProjDmndRsceDistrIsNotSupplied,
29
30     /* Associations */
31     _Assignment,
32     _ProjDmndRsceDistrQtyUnit,
33     _Root,
34     _RoleProjectDemand,
35     _Role
36 }where ProjDmndRsceDistrIsNotSupplied = ' '

```

Рис. 3.20. ZJTSC\_I\_HoursAssigned

- 2) ZJTSC\_I\_HoursRecorded – для зберігання даних про відпрацьовані години за посадами (рис. 3.21).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Role Recorded Hours'
7 @Metadata.ignorePropagatedAnnotations: true
8
9 define view entity ZJTSC_I_HoursRecorded
10 as select from I_TimeSheetRecord
11 association [1] to ZJTSC_I_WorkPackage as _WorkPackage on $projection.WBSElementInternalID
12 association [1] to ZJTSC_I_Role as _Role on $projection.WorkPackageUUID = _R
13 and $projection.Role = _R
14 and $projection.WorkItem = _R
15 association [0..1] to I_TimeSheetRecord as _IsPredecessor on $projection.TimeSheetRecord = _I
16 association [1..1] to I_CalendarDate as _Date on $projection.TimeSheetDate = _
17 association [0..1] to I_PersonWorkAgreement_1 as _PersonWorkAgreement on $projection.Person = _PersonWork
18 {
19 key TimeSheetRecord,
20 PersonWorkAgreement as Person,
21 _PersonWorkAgreement.Person as PersonNumber,
22 TimeSheetDate,
23 _Date.CalendarYear as CalendarYear,
24 _Date.CalendarMonth as CalendarMonth,
25 WBSElementInternalID,
26 _WorkPackage.WorkPackageUUID as WorkPackageUUID,
27 _WorkPackage.ProjectUUID as ProjectUUID,
28 ActivityType as Role,
29 WorkItem,
30 @Semantics.quantity.unitOfMeasure: 'HoursUnitOfMeasure'
31 RecordedHours,
32 HoursUnitOfMeasure,
33 @Semantics.quantity.unitOfMeasure: 'UnitOfMeasure'
34 RecordedQuantity,
35 UnitOfMeasure,
36 ServiceDocumentType.

```

Рис. 3.21. ZJTSC\_I\_HoursRecorded

3) ZJTSC\_I\_PersonAssignment – для зберігання даних про працюючих на проектах людей (рис. 3.22).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Person Assignment'
7 @Metadata.ignorePropagatedAnnotations: true
8 define view entity ZJTSC_I_PersonAssignment
9   as select from ZJTSC_I_Role
10  {
11    key WBSElementInternalID,
12    key Role,
13    key WorkItem,
14    key AssignedPerson as Person
15  }
16 union select distinct from ZJTSC_I_HoursRecorded
17  {
18    key WBSElementInternalID,
19    key Role,
20    key WorkItem,
21    key Person
22  }
23

```

Рис. 3.22. ZJTSC\_I\_PersonAssignment

- 4) ZJTSC\_I\_PersonSH – search help об’єкт, використовується для швидкого вводу під час роботи з системою (рис. 3.23).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Person Search Help'
7 @Metadata.ignorePropagatedAnnotations: true
8 define view entity ZJTSC_I_PersonSH as select from I_PersonWorkAgreement_1
9  {
10   @EndUserText.label : 'Person Work Agreement'
11   key PersonWorkAgreement,
12   @EndUserText.label : 'Person Number'
13   Person,
14   @EndUserText.label : 'PersonFullName'
15   PersonFullName
16  }
17

```

Рис. 3.23. ZJTSC\_I\_PersonSH

- 5) ZJTSC\_I\_Project – відомості про поточні проекти (рис. 3.24).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Project'
7 @Metadata.ignorePropagatedAnnotations: true
8 @ObjectModel.usageType:{
9   serviceQuality: #X,
10  sizeCategory: #S,
11  dataClass: #MIXED
12 }
13 define view entity ZJTSC_I_Project
14 as select from I_EnterpriseProject
15 association [1..*] to ZJTSC_I_WorkPackage as _WorkPackage on $projection.ProjectUUID = _WorkPackage.Proje
16 {
17   key ProjectUUID,
18     RestrictedTimePosting,
19     ProjectCategory,
20     ProjectSummaryTaskUUID,
21     Project,
22     ProjectDescription,
23     ProcessingStatus,
24     PlannedStartDate,
25     PlannedEndDate,
26     PriorityCode,
27     EntProjIsMultiSlsOrdItmsEnbld,
28     ProjectStartDate,
29     ProjectEndDate,
30     WBS_ElementInternalID,
31     ProfitCenter,
32     CompanyCode,
33     ControllingArea,
34     FunctionalArea,
35     Plant,
36     ResponsibleCostCenter

```

Рис. 3.24. ZJTSC\_I\_Project

6) ZJTSC\_I\_ResponsiblePerson – відомості про керівників проектів (рис. 3.25).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Responsible/Assigned Person'
7 @Metadata.ignorePropagatedAnnotations: true
8
9 define view entity ZJTSC_I_ResponsiblePerson
10 as select from I_EntProjectEntitlement
11
12 association [0..*] to I_BusinessUserBasic as _UserID on $projection.businesspartneruuid = _UserID.Busine
13 {
14   key ProjectEntitlementUUID,
15     ProjectUUID,
16     TeamMemberUUID,
17     ProjectRoleUUID,
18     _Role.ProjectRoleCategory,
19     _TeamMember.BusinessPartnerUUID,
20     _UserID
21 }
22 }
23

```

Рис. 3.25. ZJTSC\_I\_ResponsiblePerson

7) ZJTSC\_I\_Role – відомості про посади, що необхідні на проектах (рис. 3.26).

```
4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Project Role'
7 @Metadata.ignorePropagatedAnnotations: true
8 @ObjectModel.usageType:{
9   serviceQuality: #X,
10  sizeCategory: #S,
11  dataClass: #MIXED
12 }
13 define view entity ZJTSC_I_Role
14 as select from I_ProjectDemand
15 association [1] to ZJTSC_I_Project as _Project on $projection.ProjectUUID = _Pro
16 association [1] to ZJTSC_I_WorkPackage as _WorkPackage on $projection.WorkPackageUUID =
17
18 association [0..*] to ZJTSC_I_RoleName as _RoleText on $projection.Role = _RoleText.f
19
20 association [0..*] to ZJTSC_I_HoursAssigned as _HoursAssignedAssignment on $projection.ResourceAssignment
and $projection.AssignedPerson
21
22 {
23 key ProjectDemandUUID,
24 ProjectDemand,
25 _Work.ActivityType as Role,
26 _Work.ProjectElementWorkItem as WorkItem,
27 _Work.ProjDmndAssgmtStatus as AssignmentStatus,
28 _Work.ProjDmndRequestedDeliveryOrg as DeliveryOrganization,
29
30 _Work._ResourceAssignment.ProjDmndRsceAssgmtUUID as ResourceAssignmentUUID,
31 _Work._ResourceAssignment.ProjDmndRsceAssgmt as AssignedPerson,
32 _Work._ResourceAssignment._PersonWorkAgreement.PersonFullName as AssignedPersonFullName,
33
34 ProjectDemandName,
35 ProjectDemandDescription,
36 ProjectDemandCategory.
```

Рис. 3.26. ZJTSC\_I\_Role

8) ZJTSC\_I\_RoleName – назви посад різними мовами (рис. 3.27).

```
4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Role Name'
7 define view entity ZJTSC_I_RoleName
8 as select from I_CostCenterActivityTypeText
9 {
10 key CostCtrActivityType as Role,
11 key Language,
12 CostCtrActivityTypeName as RoleName,
13 _Language
14 }
15
```

Рис. 3.27. ZJTSC\_I\_RoleName



9) ZJTSC\_I\_RoleSH – search help об’єкт, використовується для швидкого вводу під час роботи з системою (рис. 3.28).

```
4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Role SH for Unstaffed Posting'
7 @Metadata.ignorePropagatedAnnotations: true
8 @Search.searchable: true
9 define view entity ZJTSC_I_RoleSH
10   as select from ZJTSC_I_Role
11   {
12     @UI.lineItem: [{ position: 20 } ]
13     @UI.selectionField: [{ position: 20 } ]
14     @ObjectModel.text.element: ['WorkPackageDescription']
15     @UI.textArrangement: #TEXT_LAST
16     @EndUserText.label: 'Work Package'
17   key WorkPackage,
18     @UI.lineItem: [{ position: 30 } ]
19     @UI.selectionField: [{ position: 30 } ]
20     @ObjectModel.text.element: ['RoleName']
21     @UI.textArrangement: #TEXT_ONLY
22     @EndUserText.label: 'Role'
23   key Role,
24     @UI.lineItem: [{ position: 40 } ]
25     @UI.selectionField: [{ position: 40 } ]
26     @EndUserText.label: 'Work Item'
27   key WorkItem,
28     @UI.lineItem: [{ hidden: true } ]
29     max(_RoleText[ Language = $session.system_language ].RoleName) as RoleName,
30     @ObjectModel.virtualElementCalculatedBy: 'ABAP:ZJTSC_CL_VIRTUAL_FIELD'
31     virtual WorkItemName      : abap.char(40),
32     @UI.lineItem: [{ hidden: true } ]
33     max(_WorkPackage.WorkPackageDescription) as WorkPackageDescription,
34     @Search.defaultSearchElement: true
35     @Search.fuzzinessThreshold: 0.8
36     @EndUserText.label : 'Project'
37     @UI.lineItem: [{ position: 10 } ]
```

Рис. 3.28. ZJTSC\_I\_RoleSH

10) ZJTSC\_I\_Task – відомості про проекти за їх задачами (рис. 3.29).

```

4 @AccessControl.authorizationCheck: #NOT_REQUIRED
5 @EndUserText.label: 'Task'
6 @Metadata.ignorePropagatedAnnotations: true
7 define root view entity ZJTSC_I_Task
8   as select from ZJTSC_I_Project
9   {
10
11   key ProjectUUID          as ObjectUUID,
12     'P'                    as Type,
13     Project                as ID,
14     ProjectDescription    as Description,
15     ProjectStartDate      as StartDate,
16     ProjectEndDate        as EndDate,
17     LastChangeDateTime
18
19   }
20 union select from ZJTSC_I_ServiceOrder
21 {
22   key ServiceOrderUUID          as ObjectUUID,
23     'S'                          as Type,
24     ServiceOrder                as ID,
25     ServiceOrderDescription    as Description,
26     tstmp_to_dats(RequestedServiceStartTime, ' ', $session.client, 'NULL') as StartDate,
27     tstmp_to_dats(RequestedServiceEndTime, ' ', $session.client, 'NULL') as EndDate,
28     LastChangeDateTime
29
30 }
31

```

Рис. 3.29. ZJTSC\_I\_Task

11) ZJTSC\_I\_TotalHoursPlanned – відомості про загальний бюджет часу на проектах (рис. 3.30).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Total Planned Hours'
7 define view entity ZJTSC_I_TotalHoursPlanned
8   as select from I_ProjDmndRsceAssgmtDistr
9   {
10    key _Root.ReferencedObjectUUID           as WorkPackageUUID,
11    key _Assignment._Work.ActivityType       as Role,
12    key _Assignment._Work.ProjectElementWorkItem as WorkItem,
13    key _Assignment.ProjDmndRsceAssgmt      as AssignedPerson,
14    key ProjDmndRsceAssgmtDistrYearVal     as CalendarYear,
15    key ProjDmndRsceAssgmtDistrPerdVal     as CalendarMonth,
16    @Semantics.quantity.unitOfMeasure: 'HoursUnitOfMeasure'
17    sum( ProjDmndRsceAssgmtDistrQty )      as PlannedHours,
18    ProjDmndRsceAssgmtDistrQtyUnit        as HoursUnitOfMeasure
19  }
20 }
21 group by
22   _Root.ReferencedObjectUUID,
23   _Assignment._Work.ActivityType,
24   _Assignment._Work.ProjectElementWorkItem,
25   _Assignment.ProjDmndRsceAssgmt,
26   ProjDmndRsceAssgmtDistrYearVal,
27   ProjDmndRsceAssgmtDistrPerdVal,
28   ProjDmndRsceAssgmtDistrQtyUnit

```

Рис. 3.30. ZJTSC\_I\_TotalHoursPlanned

12) ZJTSC\_I\_TotalHoursRecorded – відомості про загальну кількість відпрацьованих годин на проектах (рис. 3.31).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Total Planned and Recorded Hours'
7 define view entity ZJTSC_I_TotalHoursRecorded
8   as select from ZJTSC_I_HoursRecorded
9   {
10    key WBSElementInternalID,
11    key Role,
12    key WorkItem,
13    key Person,
14    key CalendarYear,
15    key CalendarMonth,
16    @Semantics.quantity.unitOfMeasure: 'HoursUnitOfMeasure'
17    sum(RecordedHours) as RecordedHours,
18    HoursUnitOfMeasure
19  }
20 }group by WBSElementInternalID, Role, WorkItem,
21           Person, CalendarYear, CalendarMonth, HoursUnitOfMeasure
22
23
24
25

```

Рис. 3.31. ZJTSC\_I\_TotalHoursRecorded

13) ZJTSC\_I\_UserRoles – відомості про ролі поточного користувача у системі (рис. 3.32).

```
4 @AccessControl.authorizationCheck: #NOT_REQUIRED
5 @EndUserText.label: 'Current User`s roles CDS'
5 define view entity ZJTSC_I_USER_ROLES
7   as select from I_IAMBusinessUserBusinessRole
3   {
3     key UserID,
3     key BusinessRoleUUID,
1     BusinessRole,
2     case
3       when BusinessRole = 'BR_PROJECT_MANAGER_PROF'
4         then 'P'
5     end as TimePostingAllowed,
5     case
7       when BusinessRole = 'BR_BILLING_CLERK' or
3         BusinessRole = 'BR_CREDIT_CONTROLLER' or
3         BusinessRole = 'BR_CUSTOMER_SRVC_MGR' or
3         BusinessRole = 'BR_SERVICE_PLANNER' or
1         BusinessRole = 'BR_SOLN_ORDER_SPCLST'
2       then 'S'
3     end as ServiceOrdersAllowed
4   } where UserID = $session.user
5
```

Рис. 3.32. ZJTSC\_I\_UserRoles

14) ZJTSC\_I\_WorkPackage – відомості про пакет, до якого “прив’язаний” проект (рис. 3.33).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Project Work Package'
7 @Metadata.ignorePropagatedAnnotations: true
8 @ObjectModel.usageType:{
9   serviceQuality: #X,
10  sizeCategory: #S,
11  dataClass: #MIXED
12 }
13 define view entity ZJTSC_I_WorkPackage
14 as select from I_EnterpriseProjectElement
15 association [1] to ZJTSC_I_Project as _Project on $projection.ProjectUUID = _Project.ProjectUUID
16 association [0..*] to ZJTSC_I_Role as _Roles on $projection.WorkPackageUUID = _Roles.WorkPackageUUID
17
18 {
19 key ProjectElementUUID as WorkPackageUUID,
20   ProjectElement      as WorkPackage,
21   ProjectUUID,
22   ProjectElementDescription as WorkPackageDescription,
23   ProcessingStatus,
24   PlannedStartDate,
25   PlannedEndDate,
26   EntProjectElementType,
27   WBSElementInternalID,
28   ProjectInternalID,
29   Plant,
30   /* Associations */
31   _AuthGroup,
32   _AuthRole,
33   _AuthSubst,
34   _AuthUser,
35   _EntProjElmntDlvbrl,
36

```

Рис. 3.33. ZJTSC\_I\_WorkPackage

15) ZJTSC\_I\_WorkTypeVH – відомості про тип проекту (рис. 3.34).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Work Type Value Help'
7 @Metadata.ignorePropagatedAnnotations: true
8
9 @ObjectModel.resultSet.sizeCategory: #XS
10 define view entity ZJTSC_I_WorkTypeVH as select from DDCDS_CUSTOMER_DOMAIN_VALUE_T( p_domain_name: 'ZJTSC_Wk
11@ObjectModel.text.element: ['text']
12 @UI.textArrangement: #TEXT_ONLY
13 key value_low as WorkType,
14 text
15 }where language = $session.system_language
16

```

Рис. 3.34. ZJTSC\_I\_WorkTypeVH

16) ZJTSC\_I\_RoleDetails – додаткові відомості про посади (рис. 3.35).

```

@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Role Details'

define view entity ZJTSC_I_RoleDetails
  as select from I_ProjectDemand
{
  key _WorkPackage.WBSElementInternalID          as WBSElementInternalID,
  key _Work.ActivityType                          as Role,
  key _Work.ProjectElementWorkItem               as WorkItem,
  key _Work._ResourceAssignment.ProjDmndRsceAssgmt as AssignedPerson,
  ProjectUUID,
  ReferencedObjectUUID                          as WorkPackageUUID,
  max(_Work.ProjDmndAssgmtStatus)                as AssignmentStatus,
  max(_Work.ProjDmndRequestedDeliveryOrg)        as DeliveryOrganization,
  min(ProjectDemandStartDate)                   as StartDate,
  max(ProjectDemandEndDate)                     as EndDate
}
where
  ProjectDemandCategory = '03'
group by
  ProjectUUID,
  ReferencedObjectUUID,
  _WorkPackage.WBSElementInternalID,
  _Work.ActivityType,
  _Work.ProjectElementWorkItem,
  _Work._ResourceAssignment.ProjDmndRsceAssgmt

```

Рис. 3.35. ZJTSC\_I\_RoleDetails

17) ZJTSC\_I\_ActualDurationItem – кількість витраченого часу на виконання завдання (рис. 3.36).

```

3 //no maintenance or support is provided for this content.
4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Service Order Item Actual Duration'
7 define view entity ZJTSC_I_ActualDurationItem
8   as select from ZJTSC_I_ServiceConfirmItem
9   {
10    key ServiceOrder,
11    key ServiceOrderItem,
12    @Semantics.quantity.unitOfMeasure: 'ActualServiceDurationUnit'
13    sum(ActualServiceDuration) as ActualServiceDuration,
14    @Semantics.quantity.unitOfMeasure: 'ServiceDocItemQuantityUnit'
15    sum(ServiceDocumentItemQuantity) as ActualQuantity,
16    ActualServiceDurationUnit,
17    ServiceDocItemQuantityUnit
18  }
19 where ServiceDocumentItemIsRejected = ' '
20 group by
21   ServiceOrder,
22   ServiceOrderItem,
23   ActualServiceDurationUnit,
24   ServiceDocItemQuantityUnit
25

```

Рис. 3.36. ZJTSC\_I\_ActualDurationItem

18) ZJTSC\_I\_ActualDurationOrder – кількість витрачених матеріалів при виконанні завдання (рис. 3.37).



```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Service Order Actual Duration'
7 define view entity ZJTSC_I_ActualDurationOrder
8 as select from ZJTSC_I_ServiceOrderItem as PlannedDuration
9 association [0..1] to ZJTSC_I_ActualDurationItem as _ActualDuration on PlannedDuration.ServiceOrder = _ActualDuration
10 and PlannedDuration.ServiceOrderItem = _ActualDuration
11 {
12 key ServiceOrderUUID,
13 ServiceOrder,
14 @Semantics.quantity.unitOfMeasure: 'QuantityUnit'
15 sum(ServiceDocumentItemQuantity) as PlannedQuantity,
16 @Semantics.quantity.unitOfMeasure: 'QuantityUnit'
17 sum(_ActualDuration.ActualQuantity) as ActualQuantity,
18 ServiceDocItemQuantityUnit as QuantityUnit
19 }
20 where ServiceDocumentItemIsRejected = ''
21 group by ServiceOrderUUID, ServiceOrder, ServiceDocItemQuantityUnit
22

```

Рис. 3.37. ZJTSC\_I\_ActualDurationOrder

19) ZJTSC\_I\_ConfirmationStatusVH – search-help для існуючих статусів виконання завдань (рис. 3.38).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Confirmation Status Value Help'
7 @ObjectModel.resultSet.sizeCategory: #XS
8 define view entity ZJTSC_I_ConfirmationStatusVH
9 as select from ZJTSC_I_StatusVH
10 {
11 key ServiceDocumentStatus,
12 StatusName
13 }
14 where
15 ServiceDocumentStatus = 'C'
16 or ServiceDocumentStatus = 'A'
17

```

Рис. 3.38. ZJTSC\_I\_ConfirmationStatusVH

20) ZJTSC\_I\_ServiceConfirmItem – відомості про підтвердження виконання завдань (рис. 3.39).

```
4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Service Confirmation Item'
7 @Metadata.ignorePropagatedAnnotations: true
8 @ObjectModel.usageType:{
9   serviceQuality: #A,
10  sizeCategory: #XXL,
11  dataClass: #TRANSACTIONAL
12 }
13 define view entity ZJTSC_I_ServiceConfirmItem
14   as select from I_ServiceDocumentItem
15   association [1] to ZJTSC_I_ServOrderItemConfirm as _ServOrderItem on $projection.ServiceConfirmation
16   and $projection.ServiceConfirmationItem
17 {
18   key ServiceDocument           as ServiceConfirmation,
19   key ServiceDocumentItem       as ServiceConfirmationItem,
20
21   _ServOrderItem.ServiceOrderUUID as ServiceOrderUUID,
22   _ServOrderItem.ServiceOrderItemUUID as ServiceOrderItemUUID,
23   ServiceDocumentItemUUID       as ServiceConfirmationItemUUID,
24
25   _ServOrderItem.ServiceOrder   as ServiceOrder,
26   _ServOrderItem.ServiceOrderItem as ServiceOrderItem,
27
28
29   ServiceDocumentDescription     as ServiceConfirmationDescription,
30   _ServiceDocument.ServiceConfirmationIsFinal as ServiceConfirmationIsFinal,
31   ServiceObjectType,
32   ServiceDocumentType,
33   ServiceDocumentItemPriority,
34   PostingDate,
35   ServicesRenderedDate,
36   ServiceOrderItemValidityStartDateTime
```

Рис. 3.39. ZJTSC\_I\_ServiceConfirmItem

21) ZJTSC\_I\_ServiceOrder – відомості про поточні завдання (рис. 3.40).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Service Order'
7 @Metadata.ignorePropagatedAnnotations: true
8 @ObjectModel.usageType:{
9   serviceQuality: #X,
10  sizeCategory: #XL,
11  dataClass: #MIXED
12 }
13 define view entity ZJTSC_I_ServiceOrder
14 as select from I_ServiceDocument
15 association [1..*] to ZJTSC_I_ServiceOrderItem as _OrderItem on $projection.ServiceOrder = _OrderItem.Serv
16 association [0..1] to ZJTSC_I_ActualDurationOrder as _ActualDuration on $projection.ServiceOrderUUID = _Ac
17 {
18   key ServiceDocumentUUID          as ServiceOrderUUID,
19     ServiceDocument                  as ServiceOrder,
20
21     ServiceDocumentCharUUID,
22     ServiceDocumentType,
23     ServiceDocumentDescription as ServiceOrderDescription,
24     Language,
25     PostingDate,
26     ServiceDocChangedDateTime as LastChangeDateTime,
27     RefBusinessSolutionOrder,
28     @Semantics.quantity.unitOfMeasure: 'QuantityUnit'
29     _ActualDuration.PlannedQuantity,
30     @Semantics.quantity.unitOfMeasure: 'QuantityUnit'
31     _ActualDuration.ActualQuantity,
32     _ActualDuration.QuantityUnit,
33     ServiceTeam,
34     ResponsibleEmployee,
35     ServiceEmployee,
36     ContactPerson

```

Рис. 3.40. ZJTSC\_I\_ServiceOrder

22) ZJTSC\_I\_ServiceOrderItem – відомості про поточні підзавдання (рис. 3.41).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Service Order Item'
7 @Metadata.ignorePropagatedAnnotations: true
8
9 define view entity ZJTSC_I_ServiceOrderItem
10 as select from I_ServiceDocumentItem
11 association [1..*] to ZJTSC_I_ServiceConfirmItem as _OrderItemConfirmation on $projection.ServiceOrder
12 and $projection.ServiceOrderItem
13 {
14 key ServiceDocument as ServiceOrder,
15 key ServiceDocumentItem as ServiceOrderItem,
16
17 ServiceDocumentDescription as ServiceOrderDescription,
18 ServiceDocumentItemDescription as ServiceOrderItemDescription,
19 ServiceDocumentUUID as ServiceOrderUUID,
20 ServiceDocumentItemUUID as ServiceOrderItemUUID,
21
22 ServiceObjectType,
23 ServiceDocumentType,
24 ServiceDocumentItemPriority,
25 PostingDate,
26 ServicesRenderedDate,
27 SrvcDocItmValdtyStartDateTime,
28 SrvcDocItmValdtyEndDateTime,
29 ServiceDocumentItemUUID,
30 ServiceDocumentItemCharUUID,
31 ServiceDocumentItemObjectType,
32 ParentServiceDocumentItemUUID,
33 OriginallyRequestedProduct,
34 Product,
35 ServiceDocItemCategory,
36 RefBusinessSolutionOrder.

```

Рис. 3.41. ZJTSC\_I\_ServiceOrderItem

23) ZJTSC\_I\_ServOrderItemConfirm – відомості про підтвердження виконання підзавдань (рис. 3.42).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Service Order Item Confirmation'
7 @Metadata.ignorePropagatedAnnotations: true
8 @ObjectModel.usageType:{
9   serviceQuality: #B,
10  sizeCategory: #L,
11  dataClass: #TRANSACTIONAL
12 }
13 define view entity ZJTSC_I_ServOrderItemConfirm
14   as select from I_ServiceDocumentItemSuccssr_2 as ServiceOrderItemConfirmation
15   association [0..*] to I_ServiceDocumentItemEnhcd as _ServiceOrderItem on $projection.ServiceOrderItemUUID
16 {
17
18   key   ServiceDocumentItemCharUUID           as ServiceOrderItemUUID,
19         _ServiceOrderItem.ServiceDocumentUUID as ServiceOrderUUID,
20         _ServiceOrderItem.ServiceDocument     as ServiceOrder,
21         _ServiceOrderItem.ServiceDocumentItem as ServiceOrderItem,
22         ServiceOrderItemConfirmation._ServiceDocItm.ServiceDocument as ServiceConfirmation,
23         ServiceOrderItemConfirmation._ServiceDocItm.ServiceDocumentItem as ServiceConfirmationItem
24 }
25 }
26 where
27   ServiceOrderItemConfirmation._ServiceDocItm.ServiceObjectType = 'BUS2000117'
28 |

```

Рис. 3.42. ZJTSC\_I\_ServOrderItemConfirm

24) ZJTSC\_I\_StatusVH – додаткові відомості про посади (рис. 3.43).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Service Order Status Value Help'
7 @ObjectModel.resultSet.sizeCategory: #XS
8 define view entity ZJTSC_I_StatusVH
9   as select from I_ServiceDocumentStatus
10  {
11  @ObjectModel.text.element: ['StatusName']
12  @UI.textArrangement: #TEXT_ONLY
13  key ServiceDocumentStatus,
14     _ServiceDocumentStatusText[ Language = $session.system_language ].ServiceDocumentStatusName as Status
15  }
16 |

```

Рис. 3.43. ZJTSC\_I\_StatusVH

Дані з деяких створених CDS View можна звести до певного узагальнюючого вигляду, для спрощення роботи з ними. Для цього необхідно створити “над ними” такі CDS View:

1) ZJTSC\_I\_HoursRecordedTP – узагальнені відомості про кількість відпрацьованих годин на посаді (рис. 3.44).

```
4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #MANDATORY
6 @EndUserText.label: 'Role Recorded Hours'
7
8 define view entity ZJTSC_I_HoursRecordedTP
9   as select from I_TimeSheetRecord
10  association to parent ZJTSC_I_RoleTP as _Role on $projection.WBSElementInternalID
11  and $projection.Role
12  and $projection.WorkItem
13  and $projection.PersonWorkAgreement
14
15  association [1..1] to ZJTSC_I_WorkTP as _Project on $projection.ProjectUUID = _Proj
16
17  association [0..1] to I_BusinessUserBasic as _UserID on $projection.PersonNumber = _Use
18  association [0..1] to I_PersonWorkAgreement_1 as _PersonWorkAgreement on $projection.PersonWorkAgreement
19  association [0..1] to I_TimeSheetRecord as _IsPredecessor on $projection.TimeSheetRecord = _
20 {
21  key TimeSheetRecord,
22  PersonWorkAgreement,
23  _PersonWorkAgreement.Person as PersonNumber,
24  TimeSheetDate,
25  WBSElementInternalID,
26  ActivityType as Role,
27  WorkItem,
28  @Semantics.quantity.unitOfMeasure: 'HoursUnitOfMeasure'
29  RecordedHours,
30  @Semantics.quantity.unitOfMeasure: 'HoursUnitOfMeasure'
31  RecordedHours as DecimalRecordedHours,
32  HoursUnitOfMeasure,
33  ControllingArea,
34  TimeSheetNote,
35  _Role.ProjectUUID as ProjectUUID,
36 }
```

Рис. 3.44. ZJTSC\_I\_HoursRecordedTP

2) ZJTSC\_I\_RoleTP – узагальнені відомості про посади (рис. 3.45).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #MANDATORY
6 @EndUserText.label: 'Project Role'
7
8 define view entity ZJTSC_I_RoleTP
9   as select from ZJTSC_I_PersonAssignment
10  association to parent ZJTSC_I_WorkTP as _Project on $projection.ProjectUUID = _Proje
11  composition [0..*] of ZJTSC_I_HoursRecordedTP as _TimeSheet
12
13  association [0..1] to I_BusinessUserBasic as _UserID on $projection.PersonNumber = _User
14  association [0..1] to I_PersonWorkAgreement_1 as _PersonWorkAgreement on $projection.Person = _PersonWork
15  association [0..1] to ZJTSC_I_RoleDetails as _RoleDetails on $projection.WBSElementInternalID
16  and $projection.Role
17  and $projection.WorkItem
18  and $projection.Person
19  association [0..*] to ZJTSC_I_RoleName as _RoleText on $projection.Role = _RoleText.Rol
20  association [1..1] to ZJTSC_I_WorkPackage as _WorkPackage on $projection.WBSElementInternalID
21
22 {
23  key WBSElementInternalID,
24  key Role,
25  key WorkItem,
26  key Person,
27
28  _WorkPackage.ProjectUUID as ProjectUUID,
29  _WorkPackage.WorkPackageUUID as WorkPackageUUID,
30  _WorkPackage.WorkPackage as WorkPackage,
31
32  _PersonWorkAgreement.Person as PersonNumber,
33  _RoleDetails.AssignmentStatus,
34  _RoleDetails.DeliveryOrganization,
35  _RoleDetails.StartDate,
36  RoleDetails.FndDate.

```

Рис. 3.45. ZJTSC\_I\_RoleTP

3) ZJTSC\_I\_ServiceConfirmItemTP – узагальнені відомості про підтвердження виконання завдання (рис. 3.46).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #MANDATORY
6 @EndUserText.label: 'Service Confirmation Item'
7
8
9 define view entity ZJTSC_I_ServiceConfirmItemTP
10 as select from ZJTSC_I_ServiceConfirmItem
11 association to parent ZJTSC_I_ServiceOrderItemTP as _ServiceOrderItem on $projection.ServiceOrder
12 and $projection.ServiceOrder
13
14 association [1..1] to ZJTSC_I_WorkTP as _ServiceOrder on $projection.ServiceOr
15 association [1] to I_BusinessUserBasic as _UserID on $projection.Responsible
16 association [1] to I_BusinessUserBasic as _ServiceEmployeeUserID on $projection.ServiceEmplc
17 {
18 key ServiceConfirmation,
19 key ServiceConfirmationItem,
20
21 ServiceOrderUUID,
22 ServiceOrder,
23 ServiceOrderItem,
24
25 ServiceConfirmationDescription,
26 ServiceDocumentItemDescription,
27 ServiceConfirmationIsFinal,
28 cast( case
29 when ServiceConfirmationIsFinal = 'Y'
30 then 'X'
31 else ''
32 end as boolean ) as IsFinal,
33
34 ServiceConfirmationItemUUID,
35 ServiceOrderItemUUID,
36

```

Рис. 3.46. ZJTSC\_I\_ServiceConfirmItemTP

4) ZJTSC\_I\_ServiceOrderItemTP – узагальнені відомості про виконані підзадачі (рис. 3.47).



```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #MANDATORY
6 @EndUserText.label: 'Service Order Item'
7
8 define view entity ZJTSC_I_ServiceOrderItemTP
9   as select from ZJTSC_I_ServiceOrderItem
10
11   composition [0..*] of ZJTSC_I_ServiceConfirmItemTP as _ServiceConfirmationItem
12   association          to parent ZJTSC_I_WorkTP          as _ServiceOrder on $projection.ServiceOrderUI
13
14   association [0..1] to I_BusinessUserBasic      as _UserID          on $projection.ResponsibleEmpl
15   association [0..1] to ZJTSC_I_ActualDurationItem as _ActualDuration    on $projection.ServiceOrder
16                                     and $projection.ServiceOrderIte
17 {
18   key ServiceOrder,
19   key ServiceOrderItem,
20     ServiceOrderDescription,
21     ServiceOrderItemDescription,
22     ServiceOrderItemUUID,
23     ServiceOrderUUID,
24
25     Product,
26     ServiceDocItemCategory,
27     SrvcDocItmRequestedQuantity,
28     SrvcDocItmDeliveredQuantity,
29     SrvcDocItmReleasedQuantity,
30     CumulativeConfirmedQuantity,
31     ServiceDocItemQuantityUnit,
32     ServiceTeam,
33     ResponsibleEmployee,
34     ServiceEmployee,
35     ContactPerson,
36     ServiceDocumentItemStatus.

```

Рис. 3.47. ZJTSC\_I\_ServiceOrderItemTP

5) ZJTSC\_I\_WorkTP – узагальнені відомості про поточні проекти (рис. 3.48).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #MANDATORY
6 @EndUserText.label: 'Work (Service Orders and Projects)'
7 @Metadata.ignorePropagatedAnnotations: true
8
9 define root view entity ZJTSC_I_WorkTP
10   as select distinct from ZJTSC_I_Task
11
12   association [0..1] to ZJTSC_I_USER_ROLES      as _UserRoles      on $projection.Type = _UserRoles.Serv
13                                               $projection.Type = _UserRoles.Time
14
15   association [0..1] to ZJTSC_I_ServiceOrder    as _ServiceOrder    on $projection.ObjectUUID = _ServiceO
16                                               and $projection.Type = 'S'
17
18   association [0..1] to ZJTSC_I_Project         as _Project         on $projection.ObjectUUID = _Project.
19                                               and $projection.Type = 'P'
20
21   association [0..1] to ZJTSC_I_WorkTypeVH     as _TypeText       on $projection.Type = _TypeText.WorkT
22   association [0..1] to I_BusinessUserBasic    as _UserID         on $projection.responsibleemployee =
23   association [0..1] to ZJTSC_I_ResponsibilePerson as _ProjectManager on $projection.ObjectUUID
24                                               and (
25                                               _ProjectManager.ProjectRoleCategory
26                                               or _ProjectManager.ProjectRoleCateg
27                                               )
28
29   composition [0..*] of ZJTSC_I_ServiceOrderItemTP as _ServiceOrderItem
30   composition [0..*] of ZJTSC_I_RoleTP as _Role
31 {
32   key ObjectUUID,
33     Type,
34     ID,
35     Description,
36     StartDate,
37     EndDate,|
38     @Semantics.systemDateTime.lastChangedAt: true
39     lastChangeDateTime.

```

Рис. 3.48. ZJTSC\_I\_WorkTP

Окрім цього, необхідно розробити CDS View, що використовуватимуться під час використання функцій додатку, що вимагають появи на екрані вспливаючого вікна. Для цього найдоцільніше буде розробити такі CDS View як абстрактні, оскільки вони не призначені для зберігання даних. Також необхідно розробити CDS View на основі раніше створених таблиць (рис. 3.14 – рис. 3.18), щоб мати змогу контролювати доступ до них.

Таким чином було створено такі об'єкти:

1. ZJTSC\_I\_HouresRecordedDraft – для зберігання шаблонів відпрацьованих годин (рис. 3.49).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Role Recorded Hours Draft'
7 define view entity ZJTSC_I_HoursRecordedDraft
8   as select from zjtsc_hourec_d
9   association [0..1] to I_BusinessUserBasic as _UserID on $projection.PersonNumber = _UserI
10  association [1..1] to ZJTSC_I_WorkDraft as _Project on $projection.ProjectUUID = _Projec
11  association [0..1] to I_PersonWorkAgreement_1 as _PersonWorkAgreement on $projection.PersonWorkAgreement =
12 {
13  key timesheetrecord as TimesheetRecord,
14  personworkagreement as PersonWorkAgreement,
15  _PersonWorkAgreement.Person as PersonNumber,
16  timesheetdate as TimeSheetDate,
17  wbselementinternalid as WBSElementInternalID,
18  projectuuid as ProjectUUID,
19  role as Role,
20  workitem as WorkItem,
21  recordedhours as RecordedHours,
22  decimalrecordedhours as DecimalRecordedHours,
23  hoursunitofmeasure as HoursUnitOfMeasure,
24  controllingarea as ControllingArea,
25  timesheetnote as TimeSheetNote,
26
27  draftentitycreationdatetime as DraftEntityCreationDateTime,
28  draftentitylastchangedatetime as DraftEntityLastChangeDateTime,
29  draftadministrativedatauuid as DraftAdministrativeDataUUID,
30  draftentityoperationcode as DraftEntityOperationCode,

```

Рис. 3.49. ZJTSC\_I\_HoursRecordedDraft

2. ZJTSC\_I\_RoleDraft – для зберігання шаблонів посад (рис. 3.50).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #CHECK
6 @EndUserText.label: 'Role Draft'
7 define view entity ZJTSC_I_RoleDraft
8   as select from zjtsc_roles_dft as Draft
9   association [0..*] to ZJTSC_I_HoursRecordedDraft as _TimeSheet on $projection.WBSElementInternalID = _Tim
10   and $projection.Role = _Tim
11   and $projection.WorkItem = _Tim
12   and $projection.Person = _Tim
13   association [1..1] to ZJTSC_I_WorkDraft as _Project on $projection.ProjectUUID = _Project.Obje
14   association [0..1] to I_BusinessUserBasic as _UserID on $projection.PersonNumber = _UserID.Busi
15   association [0..1] to I_PersonWorkAgreement_1 as _PersonWorkAgreement on $projection.Person = _PersonW
16 {
17  key wbselementinternalid as WBSElementInternalID,
18  key role as Role,
19  key workitem as WorkItem,
20  key person as Person,
21  workpackageuuid as WorkPackageUUID,
22  projectuuid as ProjectUUID,
23  workpackage as WorkPackage,
24  assignmentstatus as AssignmentStatus,
25  deliveryorganization as DeliveryOrganization,
26  _PersonWorkAgreement.Person as PersonNumber,
27  startdate as StartDate,
28  enddate as EndDate,
29  cast( case
30    when assignmentstatus = ' ' then 'X'
31    -1-- ' '

```

Рис. 3.50. ZJTSC\_I\_RoleDraft

3. ZJTSC\_I\_WorkDraft – для зберігання шаблонів полектів (рис. 3.51).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Work Draft'
7 define view entity ZJTSC_I_WorkDraft
8   as select from zjtsc_work_dft as Draft
9   association [0..1] to I_BusinessUserBasic as _UserID           on $projection.ResponsibleEmployee = _UserI
10  association [0..*] to I_BusinessUserBasic as _ProjectManagerID on $projection.ProjectManager = _ProjectMar
11 {
12  key objectuuid           as ObjectUUID,
13     type                  as Type,
14     id                    as ID,
15     description           as Description,
16     startdate             as StartDate,
17     enddate               as EndDate,
18     lastchangedatetime    as LastChangeDateTime,
19     servicedocumentstatus as ServiceDocumentStatus,
20     wbselementinternalid as WBSElementInternalID,
21     responsibleemployee   as ResponsibleEmployee,
22     plannedquantity       as PlannedQuantity,
23     actualquantity        as ActualQuantity,
24     quantityunit          as QuantityUnit,
25     hideproject           as HideProject,
26     processingstatus      as ProcessingStatus,
27     projectmanager        as ProjectManager,
28     hideserviceorder       as HideServiceOrder,
29     projectmanagername    as ProjectManagerName,
30     responsible           as Responsible,

```

Рис. 3.51. ZJTSC\_I\_WorkDraft

4. ZJTSC\_I\_HouresRecordedDraft – для зберігання шаблонів відпрацьованих годин (рис. 3.52).

```

4 @AbapCatalog.viewEnhancementCategory: [#NONE]
5 @AccessControl.authorizationCheck: #NOT_REQUIRED
6 @EndUserText.label: 'Role Recorded Hours Draft'
7 define view entity ZJTSC_I_HouresRecordedDraft
8   as select from zjtsc_hourec_d
9   association [0..1] to I_BusinessUserBasic as _UserID           on $projection.PersonNumber = _UserI
10  association [1..1] to ZJTSC_I_WorkDraft as _Project           on $projection.ProjectUUID = _Projec
11  association [0..1] to I_PersonWorkAgreement_1 as _PersonWorkAgreement on $projection.PersonWorkAgreement =
12 {
13  key timesheetrecord      as TimesheetRecord,
14     personworkagreement   as PersonWorkAgreement,
15     _PersonWorkAgreement.Person as PersonNumber,
16     timesheetdate         as TimeSheetDate,
17     wbselementinternalid as WBSElementInternalID,
18     projectuuid           as ProjectUUID,
19     role                   as Role,
20     workitem              as WorkItem,
21     recordedhours         as RecordedHours,
22     decimalrecordedhours  as DecimalRecordedHours,
23     hoursunitofmeasure    as HoursUnitOfMeasure,
24     controllingarea       as ControllingArea,
25     timesheetnote         as TimeSheetNote,
26
27     draftentitycreationdatetime as DraftEntityCreationDateTime,
28     draftentitylastchangedatetime as DraftEntityLastChangeDateTime,
29     draftadministrativedatauuid as DraftAdministrativeDataUUID,
30     draftentityoperationcode as DraftEntityOperationCode,

```

Рис. 3.52. ZJTSC\_I\_HouresRecordedDraft

5. ZJTSC\_C\_Confirm – впливаюче вікно для підтвердження виконання завдань (рис. 3.53).

```
4 @EndUserText.label: 'Abstract Entity for Confirmation Action'
5 @Metadata.allowExtensions: true
6 define abstract entity ZJTSC_C_Confirm
7 {
8   @Consumption.valueHelpDefinition: [{ entity:
9     { name: 'ZJTSC_I_PersonSH', element: 'Person'}
10
11     }]
12 @EndUserText.label: 'Employee'
13 Employee      : abap.char(10);
14 @EndUserText.label: 'Actual Start Date'
15 ActualStartDate : abap.dats;
16 @EndUserText.label: 'Actual Duration'
17 ActualDuration  : abap.dec( 7, 2 );
18 @EndUserText.label: 'Final Confirmation'
19 IsFinal         : abap_boolean;
20 @EndUserText.label: 'Completed'
21 IsComplited    : abap_boolean;
22
23 }
24
```

Рис. 3.53. ZJTSC\_C\_Confirm

6. ZJTSC\_C\_UnstaffedPosting – впливаюче вікно для фіксації відпрацьованих годин людьми поза штатом проекту (рис. 3.54).

```

4 @EndUserText.label: 'Abstract Entity for Unstaffed Posting'
5
6 define abstract entity ZJTSC_C_UnstaffedPosting
7 {
8   @Consumption.valueHelpDefinition: [{ entity:
9     { name: 'ZJTSC_I_PersonSH', element: 'PersonWorkAgreement'}
10
11     }]
12   @EndUserText.label: 'Person'
13   PersonWorkAgreement : abap.numc(8);
14   @EndUserText.label: 'TimeSheetDate'
15   TimeSheetDate : abap.dats;
16   @EndUserText.label: 'Recorded Hours'
17   RecordedHours : abap.dec(4,2);
18 }
19

```

Рис. 3.54. ZJTSC\_C\_UnstaffedPosting

7. ZJTSC\_C\_UnstaffedProject – впливаюче вікно для додавання нових людей на проект (рис. 3.55).

```

4 @EndUserText.label: 'Abstract Entity for Unstaffed Project'
5 define abstract entity ZJTSC_C_UnstaffedProject
6 {
7   @Consumption.valueHelpDefinition: [
8     {
9       entity: { name: 'ZJTSC_I_RoleSH', element: 'WorkPackage' },
10      additionalBinding: [ { element: 'Role', localElement: 'Role' },
11                          { element: 'WorkItem', localElement: 'WorkIter' }
12      ]
13    }
14  ]
15  @EndUserText.label: 'Work Package'
16  WorkPackage : abap.char(24);
17  @EndUserText.label: 'Role'
18  Role : lstar;
19  @EndUserText.label: 'Work Item'
20  WorkItem : abap.char(10);
21  @EndUserText.label: 'TimeSheetDate'
22  TimeSheetDate : abap.dats;
23  @EndUserText.label: 'Recorded Hours'
24  RecordedHours : abap.dec(4,2);
25 }
26

```

Рис. 3.55. ZJTSC\_C\_UnstaffedProject

Останній крок у розробці моделі даних – створення Projection View “над” узагальнюючими CDS View – для того щоб прописати там анотації для підтримки швидкого вводу:

- 1) ZJTSC\_C\_HouresRecorded – для зберігання шаблонів відпрацьованих годин (рис. 3.56).

```
4 @EndUserText.label: 'Role Recorded Hours Projection View'
5 @AccessControl.authorizationCheck: #MANDATORY
6 @Metadata.allowExtensions: true
7 define view entity ZJTSC_C_HouresRecorded
8   as projection on ZJTSC_I_HoursRecordedTP
9   {
10    key TimeSheetRecord,
11
12    @Consumption.valueHelpDefinition: [{ entity:
13      { name: 'ZJTSC_I_PersonSH', element: 'PersonWorkAgreement'}
14    }]
15    @ObjectModel.text.element: ['PersonFullName']
16    PersonWorkAgreement,
17    PersonNumber,
18    _UserID.PersonFullName as PersonFullName,
19    TimeSheetDate,
20    WBS_ElementInternalID,
21    ProjectUUID,
22    Role,
23    WorkItem,
24    RecordedHours,
25    DecimalRecordedHours,
26    HoursUnitOfMeasure,
27    ControllingArea,
28    TimeSheetNote,
29
30    /* Associations */
```

Рис. 3.56. ZJTSC\_C\_HouresRecorded

- 2) ZJTSC\_C\_Role – для зберігання шаблонів посад (рис. 3.57).

```

4 @EndUserText.label: 'Role Projection View'
5 @AccessControl.authorizationCheck: #MANDATORY
6 @Metadata.allowExtensions: true
7 define view entity ZJTSC_C_Role
8   as projection on ZJTSC_I_RoleTP
9   {
10    key    WBSElementInternalID,
11           @ObjectModel.text.element: ['RoleName']
12    key    Role,
13           @ObjectModel.text.element: ['WorkItemName']
14    key    WorkItem,
15           @ObjectModel.text.element: ['PersonFullName']
16    key    Person,
17
18           ProjectUUID,
19           WorkPackageUUID,
20
21           PersonNumber,
22           _RoleText.RoleName    as RoleName : localized,
23           _UserID.PersonFullName as PersonFullName,
24           AssignmentStatus,
25           DeliveryOrganization,
26           StartDate,
27           EndDate,
28
29           @ObjectModel.virtualElementCalculatedBy: 'ABAP:ZJTSC_CL_VIRTUAL_FIELD'
30    virtual WorkItemName    : abap.char(40),
31

```

Рис. 3.57. ZJTSC\_C\_Role

3) ZJTSC\_C\_ServiceConfirmItem – для зберігання шаблонів полектів (рис. 3.58).

```

4 @EndUserText.label: 'Service Confirm Item Projection View'
5 @AccessControl.authorizationCheck: #MANDATORY
6 @Metadata.allowExtensions: true
7 define view entity ZJTSC_C_ServiceConfirmItem
8   as projection on ZJTSC_I_ServiceConfirmItemTP
9   {
10    key ServiceConfirmation,
11    key ServiceConfirmationItem,
12           ServiceOrderUUID,
13           ServiceOrder,
14           ServiceOrderItem,
15           ServiceConfirmationDescription,
16           ServiceDocumentItemDescription,
17           Product,
18    @Consumption.valueHelpDefinition: [{ entity:
19                                         { name: 'ZJTSC_I_ConfirmationStatusVH', element: 'ServiceDocumentSt
20                                         }}]
21    @ObjectModel.text.element: ['StatusName']
22    ServiceDocumentItemStatus,
23    _SrcvDocItmLifecycleStatus._ServiceDocumentStatusText.ServiceDocumentStatusName as StatusName :locali:
24
25    @ObjectModel.text.element: ['HasErrorName']
26    ServiceDocumentItemHasError,
27    _ServiceDocItemHasError._ServiceDocHasErrorText.ServiceDocumentHasErrorName as HasErrorName :localize:
28
29    IsFinal,
30    ServiceDocumentItemIsRejected,
31

```

Рис. 3.58. ZJTSC\_C\_ServiceConfirmItem



- 4) ZJTSC\_C\_ServiceOrderItem – для зберігання шаблонів відпрацьованих годин (рис. 3.59).

```
4 @EndUserText.label: 'Service Order Item Projection View'
5 @AccessControl.authorizationCheck: #MANDATORY
6 @Metadata.allowExtensions: true
7 define view entity ZJTSC_C_ServiceOrderItem
8 as projection on ZJTSC_I_ServiceOrderItemTP
9 {
10 key ServiceOrder,
11 key ServiceOrderItem,
12 ServiceOrderItemDescription,
13 ServiceOrderItemUUID,
14 ServiceOrderUUID,
15 Product,
16 ServiceDocItemCategory,
17 SrvDocItmReleasedQuantity,
18 ServiceDocItemQuantityUnit,
19 ServiceTeam,
20 @ObjectModel.text.element: ['PersonFullName']
21 ResponsibleEmployee,
22 _UserID.PersonFullName as PersonFullName,
23 ServiceEmployee,
24 ContactPerson,
25 ServiceDocumentItemQuantity,
26 CalculatedActualQuantity,
27 Plant,
28 ActualServiceStartDateTime,
29 ActualServiceEndDateTime,
30 ServiceWorkDuration,
--
```

Рис. 3.59. ZJTSC\_C\_ServiceOrderItem

- 5) ZJTSC\_C\_Work – вспливаюче вікно для підтвердження виконання завдань (рис. 3.60).

```

4 @EndUserText.label: 'Work Projection view'
5 @AccessControl.authorizationCheck: #MANDATORY
6 @Metadata.allowExtensions: true
7
8 define root view entity ZJTSC_C_Work
9   provider contract transactional_query
10  as projection on ZJTSC_I_WorkTP
11  {
12    key      ObjectUUID,
13    @Consumption.valueHelpDefinition: [{ entity: { name: 'ZJTSC_I_WorkTypeVH', element: 'WorkType'} }],
14    @ObjectModel.text.element: ['TypeText']
15    Type,
16    _TypeText.text                               as TypeText,
17    ID,
18    Description,
19    StartDate,
20    EndDate,
21    LastChangeDateTime,
22
23    HideProject,
24    HideServiceOrder,
25
26    @ObjectModel.text.element: ['PersonFullName']
27    ResponsibleEmployee,
28    _ResponsibleEmployeeName.PersonFullName,
29    @Consumption.valueHelpDefinition: [{ entity: { name: 'ZJTSC_I_StatusVH', element: 'ServiceDocument'} }],
30    @ObjectModel.text.element: ['StatusName']

```

Рис. 3.60. ZJTSC\_C\_Work

### 3.3. Розробка поведінки системи

Визначення поведінки системи відбувається у об'єкті типу Behavior Defenition. Він створюється на основі CDS View, і містить інформацію про тип поведінки, доступ до інформації, таблиці шаблонів, реалізуючі класи.

Behavior Defenition створюється аналогічно іншим об'єктам – треба вибрати відповідний тип об'єкту у вікні створення (рис. 3.61).

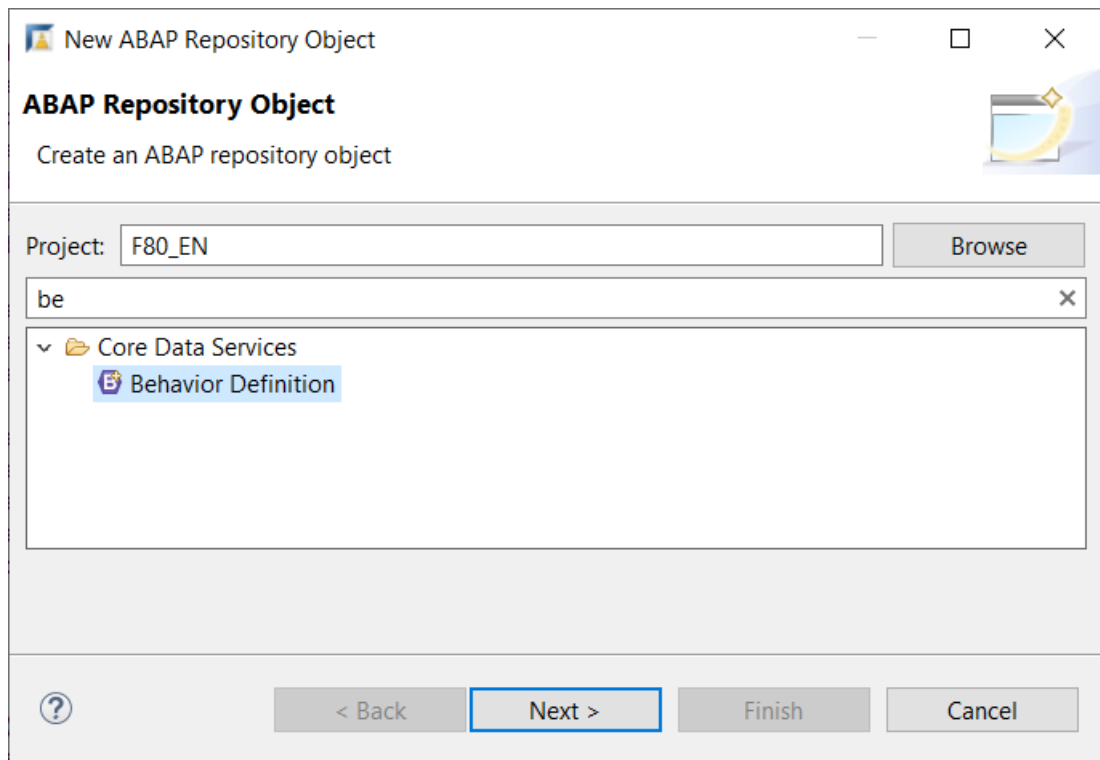


Рис.3.61 Створення Behavior Defenition

Таким же чином створюються і класи. В результаті, було розроблено такі об'єкти:

- 1) ZJTSC\_I\_WorkTP – визначає поведінку системи. Містить опис поведінки та класи реалізації (рис. 3.62).

```

4 managed implementation in class zjtsc_cl_work unique;
5 strict ( 1 );
6 with draft;
7 define behavior for ZJTSC_I_WorkTP alias Work
8 draft table zjtsc_work_dft
9 with unmanaged save
10 lock master total etag LastChangeDateTime
11 authorization master ( instance )
12 etag master LastChangeDateTime
13 {
14 internal create;
15 internal update;
16 internal delete;
17
18 draft action ( features : instance ) Edit;
19 draft action Activate;
20 draft action Discard;
21 draft action Resume;
22 draft determine action Prepare;
23
24 action ( features : instance ) SelfAssign result [1] $self;
25 static action UnstaffedProject parameter ZJTSC_C_UnstaffedProject;
26
27 field ( readonly, numbering : managed ) ObjectUUID;
28 association _ServiceOrderItem { internal create; with draft; }
29 association _Role { create; with draft; }
30 }
31

```

Рис. 3.62. ZJTSC\_I\_WorkTP

- 2) ZJTSC\_CL\_Work – основний клас, реалізує можливість редагування записів про відпрацьований час та записів про підтвердження виконання задач. Також реалізує функції додавання нового персоналу на проект, скасування підтвердження виконання задачі, призначення відповідального за виконання задачі, зміну статусу підзадач (рис. 3.63)

```
4@CLASS lhc_serviceconfirmationitem DEFINITION INHERITING FROM cl_abap_behavior_handler.
5
6 PRIVATE SECTION.
7
8 METHODS get_instance_features FOR INSTANCE FEATURES
9 IMPORTING keys REQUEST requested_features FOR serviceconfirmationitem RESULT result.
10 METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
11 IMPORTING keys REQUEST requested_authorizations FOR serviceconfirmationitem RESULT result.
12
13 METHODS cancelconfirmation FOR MODIFY
14 IMPORTING keys FOR ACTION serviceconfirmationitem~cancelconfirmation RESULT result.
15
16 METHODS finalconfirmation FOR MODIFY
17 IMPORTING keys FOR ACTION serviceconfirmationitem~finalconfirmation RESULT result.
18
19 ENDClass.
20
21@CLASS lhc_serviceconfirmationitem IMPLEMENTATION.
22
23@ METHOD get_instance_features.
24 READ ENTITY IN LOCAL MODE zjtsc_i_serviceconfirmitemtp
25 FIELDS ( servicedocumentitemstatus servicedocumentitemisrejected isfinal ) WITH CORRESPONDING #( keys )
26 RESULT DATA(lt_confirm_item_draft).
27
28 READ ENTITY IN LOCAL MODE zjtsc_i_serviceconfirmitemtp
```

Рис. 3.63. ZJTSC\_CL\_Work

- 3) ZJTSC\_CL\_Request – клас, що відповідає за відправлення HTTP-запитів до відповідних API, для редагування об'єктів у системі (рис.3.64).

```

4 CLASS zjtsc_cl_request DEFINITION
5 PUBLIC
6 FINAL
7 CREATE PUBLIC .
8 PUBLIC SECTION.
9
10 CONSTANTS:
11 BEGIN OF methods,
12     post TYPE string VALUE 'POST',
13     patch TYPE string VALUE 'PATCH',
14     delete TYPE string VALUE 'DELETE',
15 END OF methods.
16
17 TYPES:
18 BEGIN OF ts_status,
19     code TYPE i,
20     reason TYPE string,
21 END OF ts_status,
22
23 BEGIN OF ts_msglongtext_params,
24     msgid TYPE string,
25     msgno TYPE string,
26     message_v1 TYPE string,
27     message_v2 TYPE string,
28     message_v3 TYPE string,

```

Рис. 3.64. ZJTSC\_CL\_Request

- 4) ZJTSC\_CL\_Virtual\_Field – клас, що відповідає за обрахування розраховуваних полів (рис. 3.65).

```

ZJTSC_CL_VIRTUAL_FIELD
4 CLASS zjtsc_cl_virtual_field DEFINITION
5 PUBLIC
6 FINAL
7 CREATE PUBLIC .
8
9 PUBLIC SECTION.
10     INTERFACES if_sadl_exit_calc_element_read .
11
12 PROTECTED SECTION.
13 PRIVATE SECTION.
14     TYPES: BEGIN OF ty_work_item,
15             workitem TYPE string,
16             workpackage TYPE string,
17             workitemname TYPE string,
18         END OF ty_work_item,
19
20     BEGIN OF ty_work_items,
21         BEGIN OF d,
22             results TYPE STANDARD TABLE OF ty_work_item WITH EMPTY KEY,
23         END OF d,
24     END OF ty_work_items,
25
26     tt_work_item TYPE STANDARD TABLE OF ty_work_item WITH EMPTY KEY.
27
28     METHODS: get_work_item_text

```

Рис. 3.65. ZJTSC\_CL\_Virtual\_Field

### 3.4. Розробка контролю доступу до системи

За замовчуванням кожен користувач зможе використовувати систему, що неправильно, адже є ризик що працювати з системою буде неавторизована людина. Для регулювання доступу до інформації в системі (тобто CDS View)

використовуються об'єкти типу Access Control, що, по суті, модифікують вираз вибору інформації (SELECT) за бажаним сценарієм. Такі об'єкти були створені “над” узагальнюючими CDS View:

- 1) ZJTSC\_I\_HourseRecordedTP – регулює доступ до CDS View ZJTSC\_I\_HourseRecordedTP (рис. 3.66).

```
4 @EndUserText.label: 'Role Recorded Hours Access Control'  
5 @MappingRole: true  
6 define role ZJTSC_I_HoursRecordedTP {  
7     grant  
8     select  
9     on  
10        ZJTSC_I_HoursRecordedTP  
11        where  
12            _Project._ProjectManagerID.UserID = aspect user  
13            or _UserID.UserID = aspect user;  
14 }
```

Рис. 3.66. ZJTSC\_I\_HourseRecordedTP Access Control

- 2) ZJTSC\_I\_RoleTP – регулює доступ до CDS View ZJTSC\_I\_RoleTP (рис. 3.67).

```
4 @EndUserText.label: 'Role Access Control'  
5 @MappingRole: true  
6 define role ZJTSC_I_RoleTP {  
7     grant  
8     select  
9     on  
10        ZJTSC_I_RoleTP  
11        where  
12            _Project._ProjectManagerID.UserID = aspect user  
13            or _UserID.UserID = aspect user;  
14 }  
15  
16 }
```

Рис. 3.67. ZJTSC\_I\_RoleTP Access Control

- 3) ZJTSC\_I\_ServiceConfirmItemTP – регулирует доступ до CDS View ZJTSC\_I\_ServiceConfirmItemTP (рис. 3.68).

```
4 @EndUserText.label: 'Service Confirmation Item Access Control'
5 @MappingRole: true
6 define role ZJTSC_I_ServiceConfirmItemTP {
7     grant
8     select
9     on
10        ZJTSC_I_ServiceConfirmItemTP
11        where
12            ResponsibleEmployee is null
13            or ResponsibleEmployee = ''
14            or _UserID.UserID = aspect user
15            or ServiceEmployee = ''
16            or _ServiceEmployeeUserID.UserID = aspect user
17
18            or _ServiceOrder.ResponsibleEmployee = ''
19            or _ServiceOrder._UserID.UserID = aspect user
20            or _ServiceOrderItem.ResponsibleEmployee = ''
21            or _ServiceOrderItem._UserID.UserID = aspect user
22        ;
23
24 }
```

Рис. 3.68. ZJTSC\_I\_ServiceConfirmItemTP Access Control

- 4) ZJTSC\_I\_ServiceOrderItemTP – регулирует доступ до CDS View ZJTSC\_I\_ServiceOrderItemTP (рис. 3.69).

```
4 @EndUserText.label: 'Service Order Item Access Control'
5 @MappingRole: true
6 define role ZJTSC_I_ServiceOrderItemTP {
7     grant
8     select
9     on
10        ZJTSC_I_ServiceOrderItemTP
11    //     where
12    //         ResponsibleEmployee is null
13    //         or ResponsibleEmployee = ''
14    //         or _UserID.UserID = aspect user
15    //         or _ServiceOrder.ResponsibleEmployee = ''
16    //         or _ServiceOrder._UserID.UserID = aspect user
17    //         or _ServiceConfirmationItem._UserID.UserID = aspect user
18    //         or _ServiceConfirmationItem._ServiceEmployeeUserID.UserID = aspect user
19        ;
20
21 }
```

Рис. 3.69. ZJTSC\_I\_ServiceOrderItemTP Access Control

- 5) ZJTSC\_I\_WorkTP – регулирует доступ до CDS View ZJTSC\_I\_WorkTP (рис. 3.70).

```
4 @EndUserText.label: 'Work Access Control'
5 @MappingRole: true
6 define role ZJTSC_I_WorkTP {
7   grant
8     select
9       on
10        ZJTSC_I_WorkTP
11        where
12          ( Type = 'S'
13            and ( ResponsibleEmployee is null
14                  or ResponsibleEmployee = ''
15                  or _UserID.UserID = aspect user
16                  or _ServiceOrderItem._UserID.UserID = aspect user
17                  or _ServiceOrderItem._ServiceConfirmationItem._UserID.UserID = aspect user
18                  or _ServiceOrderItem._ServiceConfirmationItem._ServiceEmployeeUserID.UserID = aspect user
19                )
20          )
21        or
22          ( Type = 'P'
23            and ( ProjectManager is null
24                  or _ProjectManagerID.UserID = aspect user
25                  or _Role._UserID.UserID = aspect user
26                )
27          )
28 ;
```

Рис. 3.70. ZJTSC\_I\_WorkTP Access Control

- 6) ZJTSC\_I\_HoursRecordedDraft – регулирует доступ до CDS View ZJTSC\_I\_HoursRecordedDraft (рис. 3.71).

```
4 @EndUserText.label: 'Role Recorded Hours Draft'
5 @MappingRole: true
6 define role ZJTSC_I_HoursRecordedDraft {
7   grant
8     select
9       on
10        ZJTSC_I_HoursRecordedDraft
11        where
12          inheriting conditions from entity ZJTSC_I_HoursRecordedTP
13          or PersonWorkAgreement is initial ;
14 }
15 }
```

Рис. 3.71. ZJTSC\_I\_HoursRecordedDraft Access Control

- 7) ZJTSC\_I\_RoleDraft – регулирует доступ до CDS View ZJTSC\_I\_RoleDraft (рис. 3.72).



```
4 @EndUserText.label: 'Role Draft'  
5 @MappingRole: true  
6 define role ZJTSC_I_RoleDraft {  
7     grant  
8         select  
9             on  
10            ZJTSC_I_RoleDraft  
11            where  
12                inheriting conditions from entity ZJTSC_I_RoleTP;  
13  
14 }
```

Рис. 3.72. ZJTSC\_I\_RoleDraft Access Control

### 3.5. Налаштування інтерфейсу додатку та створення сервісу

У SAP S/4 HANA Cloud інтерфейс програми регулюється у спеціальних об'єктах – Metadata Extensions, де прописується розташування полів, їх псевдоніми (за потреби), задається рівень важливості поля. Ці об'єкти створюються з при'язкою до CDS View, з яких вони братимуть дані. Вони створюються аналогічно іншим об'єктам – треба вибрати відповідний тип у підпрограмі для створення (рис. 3.73).

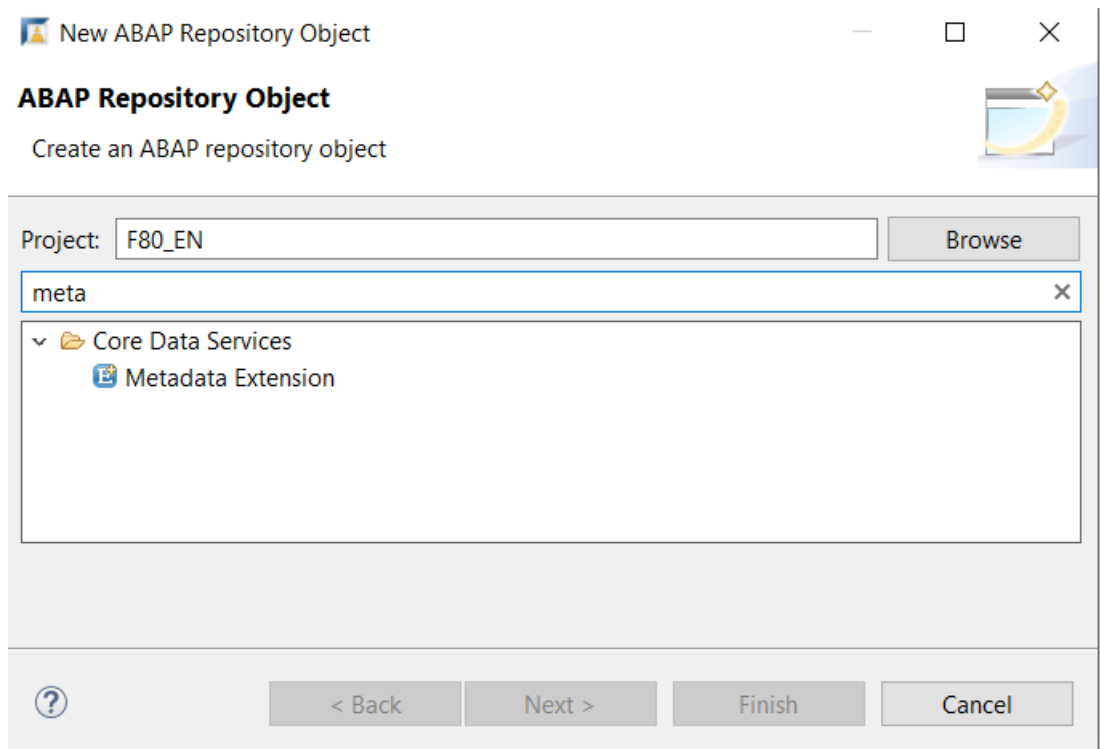


Рис. 3.73. Створення Metadata Extension

Оскільки на екрані робота відбуватиметься з Projection View (рис. 3.56 – 3.60), то логічно прописувати параметри інтерфейсу саме для них. Таким чином, було створено наступні об'єкти:

- 1) ZJTSC\_C\_HoursRecorded – налаштування інтерфейсу для ZJTSC\_C\_HoursRecorded (рис. 3.74).

```

4 @metadata.layer: #LUKE
5 @UI: {
6   headerInfo: {
7     typeName: 'TimeSheet',
8     typeNamePlural: 'TimeSheet',
9     title: { type: #STANDARD, label: 'RoleID', value: 'TimeSheetDate' }
10  },
11  presentationVariant: [{
12    visualizations: [{
13      type: #AS_LINEITEM
14    }]
15  }]
16  }
17  annotate entity ZJTSC_C_HoursRecorded with
18  {
19    @UI.facet: [
20      {
21        targetQualifier: 'Date',
22        type: #DATAPOINT_REFERENCE,
23        purpose: #HEADER
24      },
25      {
26        targetQualifier: 'RecordedHours',
27        type: #DATAPOINT_REFERENCE,
28        purpose: #HEADER
29      },
30      {
31        targetQualifier: 'Note',
32        type: #DATAPOINT_REFERENCE,
33        purpose: #HEADER
34      }
35    ]
36    @UI.dataPoint: { qualifier: 'TimeSheetRecord', title: 'TimeSheet Record' }
37    @EndUserText.label: 'TimeSheet Record'
38    TimeSheetRecord;
39  }
40  @UI: {
41    dataPoint: { qualifier: 'Date', title: 'Date (Work Agreement)' }

```

Рис. 3.74. ZJTSC\_C\_HoursRecorded

- 2) ZJTSC\_C\_Role – налаштування інтерфейсу для ZJTSC\_C\_Role (рис. 3.75).

```

4 @Metadata.layer: #CORE
5 @UI: {
6   headerInfo: {
7     typeName: 'Role',
8     typeNamePlural: 'Roles',
9     title: { type: #STANDARD, label: 'RoleID', value: 'Role' }
10  },
11  presentationVariant: [{
12    visualizations: [{
13      type: #AS_LINEITEM
14    }]
15  }]
16 }
17 annotate entity ZJTSC_C_Role with
18 {
19  @UI.facet: [ {
20    targetQualifier: 'Role',
21    type: #DATAPOINT_REFERENCE,
22    purpose: #HEADER
23  },
24  {
25    targetQualifier: 'WorkItem',
26    type: #DATAPOINT_REFERENCE,
27    purpose: #HEADER
28  },
29  {
30    targetQualifier: 'AssignedPerson',
31    type: #DATAPOINT_REFERENCE,
32    purpose: #HEADER
33  },
34  {
35    targetQualifier: 'StartDate',
36    type: #DATAPOINT_REFERENCE,
37    purpose: #HEADER
38  },
39  {
40    targetQualifier: 'EndDate',
41    type: #DATAPOINT_REFERENCE

```

Рис. 3.75. ZJTSC\_C\_Role

- 3) ZJTSC\_C\_ServiceOrderItem – налаштування інтерфейсу для ZJTSC\_C\_ServiceOrderItem (рис. 3.76).

```

4 @Metadata.layer: #CORE
5 @UI: {
6     headerInfo: {
7         typeName: 'Service Order Item',
8         typeNamePlural: 'Service Order Items',
9         title: { type: #STANDARD, label: 'Service Order Item', value: 'ServiceOrderItem' }
10    }
11 }
12 annotate entity ZJTSC_C_ServiceOrderItem with
13 {
14     @UI.facet: [
15         {
16             targetQualifier: 'Description',
17             type: #DATAPOINT_REFERENCE,
18             purpose: #HEADER
19         },
20         {
21             targetQualifier: 'Product',
22             type: #DATAPOINT_REFERENCE,
23             purpose: #HEADER
24         },
25         {
26             targetQualifier: 'Quantity',
27             type: #DATAPOINT_REFERENCE,
28             purpose: #HEADER
29         },
30         {
31             targetQualifier: 'ActualQuantity',
32             type: #DATAPOINT_REFERENCE,
33             purpose: #HEADER
34         },
35         {
36             targetQualifier: 'ResponsibleEmployee',
37             type: #DATAPOINT_REFERENCE,
38             purpose: #HEADER
39         },
40         {
41             targetQualifier: 'Status'.

```

Рис.3.76. ZJTSC\_C\_ServiceOrderItem

- 4) ZJTSC\_C\_ServiceConfirmItem – налаштування інтерфейсу для ZJTSC\_C\_ServiceConfirmItem (рис. 3.77).

```

4 @Metadata.layer: #CORE
5 @UI: {
6     headerInfo: {
7         typeName: 'Service Confirmation Item',
8         typeNamePlural: 'Service Confirmation Items',
9         title: { type: #STANDARD, label: 'Service Confirmation Item', value: 'ServiceConfirmation' }
10    }
11 }
12 annotate entity ZJTSC_C_ServiceConfirmItem with
13 {
14     @UI.facet: [
15         {
16             targetQualifier: 'ServiceConfirmation',
17             type: #DATAPOINT_REFERENCE,
18             purpose: #HEADER
19         },
20         {
21             targetQualifier: 'ServiceConfirmationItem',
22             type: #DATAPOINT_REFERENCE,
23             purpose: #HEADER
24         },
25         {
26             targetQualifier: 'ServiceConfirmationDesc',
27             type: #DATAPOINT_REFERENCE,
28             purpose: #HEADER
29         },
30         {
31             targetQualifier: 'ItemDesc',
32             type: #DATAPOINT_REFERENCE,
33             purpose: #HEADER
34         },
35         {
36             targetQualifier: 'Quantity',
37             type: #DATAPOINT_REFERENCE,
38             purpose: #HEADER
39         },
40         {
41             targetQualifier: 'ActualDuration'.

```

Рис.3.77. ZJTSC\_C\_ServiceConfirmItem

- 5) ZJTSC\_C\_Work – налаштування інтерфейсу для ZJTSC\_C\_Work (рис. 3.78).

```

4 @Metadata.layer: #CORE
5 @UI: {
6   headerInfo: {
7     typeName: 'Work',
8     typeNamePlural: 'Work',
9     title: { type: #STANDARD, label: '', value: 'ID' },
10    description: { type: #STANDARD, label: 'Type', value: 'TypeText' }
11  }
12 }
13
14
15 @UI.selectionPresentationVariant: [{
16   id: '',
17   text: 'Project',
18   selectionVariantQualifier: 'TypeP',
19   presentationVariantQualifier: 'ProjectPV',
20   qualifier: 'ProjectSPV'
21 }]
22
23 @UI.selectionVariant: [
24   { qualifier: 'TypeP',
25     text: 'Project',
26     filter: 'Type EQ "P"' },
27   { qualifier: 'Types',
28     text: 'Service Order',
29     filter: 'Type EQ "S"' },
30   { qualifier: 'TypeSNA',
31     text: 'Unassigned Service Order',
32     filter: 'Type EQ "S" and ResponsibleEmployee EQ ""' }
33 ]
34
35 @UI.presentationVariant: [
36   {
37     qualifier: 'ProjectPV',
38     visualizations: [{type: #AS_LINEITEM, qualifier: 'ProjectLine'}]
39   }
40 ]
41 annotata entity ZJTSC C Work with

```

Рис. 3.78. ZJTSC\_C\_Work

Тепер можна виводити додаток на екран. Це робиться шляхом створення об'єкту Service Binding, що згенерує URL-адресу для додатку та інтерфейс на основі раніше створених Metadata Extensions (рис. 3.74. – рис. 3.78).

Щоб створити Service Binding необхідно створити його конфігураційний файл – Service Defenition. Вони створюються аналогічно іншим об'єктам, що були розглянуті – обиранням в підсистемі створення об'єкта бажаного типу. Таким чином були створені 2 об'єкта:

- 1) ZJTSC\_TIMESERVICECONFIRM – конфігураційний файл сервісу (рис. 3.79)

```

1 @EndUserText.label: 'Joint Time and Service Confirmation'
2 define service ZJTSC_TimeServiceConfirm {
3   expose ZJTSC_C_Work as Work;
4   expose ZJTSC_C_Role as Role;
5   expose ZJTSC_C_HoursRecorded as TimeSheet;
6   expose ZJTSC_C_ServiceOrderItem as ServiceOrderItem;
7   expose ZJTSC_C_ServiceConfirmItem as ServiceConfirmationItem;
8 }

```

Рис. 3.79. Конфігураційний файл сервісу

2) ZJTSC\_UI\_04 – сам сервіс (рис. 3.80).

**Service Binding: ZJTSC\_UI\_04**

**General Information**  
This section describes general information about this service binding  
Binding Type: OData V4 - UI

**Services**  
Define services associated with the Service Binding  
Default Authorization Values: 92DDF25F9F0CD87FA4F78CD655B68CHT  
Local Service Endpoint:

Service Name	Version	API State	Service Definition
▼ ZJTSC_TIMESERV			
	1.0.0	Not Releas...	ZJTSC_TIMESERVICE

**Service Version Details**  
View information on selected service version  
Service Information  
Service URL: /sap/opu/odata4/sap/zjtsc\_ui\_04/srvd/sap/zjtsc\_timeserviceconfirm/0001/

**Entity Set and Association**

- TimeSheet
  - \_Project
  - \_Role
- Role
  - \_Project
  - \_TimeSheet
- ServiceConfirmationItem
  - \_ServiceOrder
  - \_ServiceOrderItem
- ServiceOrderItem
  - \_ServiceConfirmationItem
  - \_ServiceOrder
- Work
  - \_Role
  - \_ServiceOrderItem

Рис. 3.80. Сервіс ZJTSC\_UI\_04

Для підтримки можливості роботи з шаблонами, сервіс було створено на основі протоколу ODATA V4, оскільки на іншому доступному протоколі – ODATA V2 така можливість буде відсутня.



### 3.6. Налаштування SAP-системи

Для коректної роботи додатку залишилось налаштувати системи, в якій він виконуватиметься. Наразі, можливість відправляти HTTP-запити у створюваному додатку недоступна – в будь-якому випадку відповіддю буде 403 forbidden. Щоб система дозволила звертатися до неї, необхідно створити об'єкт під назвою Communication Arrangement. Для цього треба ввести посилання на свою систему SAP S/4 HANA Cloud у адресній стрічці браузера – та перейти за ним. Відкрита система називається SAP Fiori Launchpad (рис. 3.81).

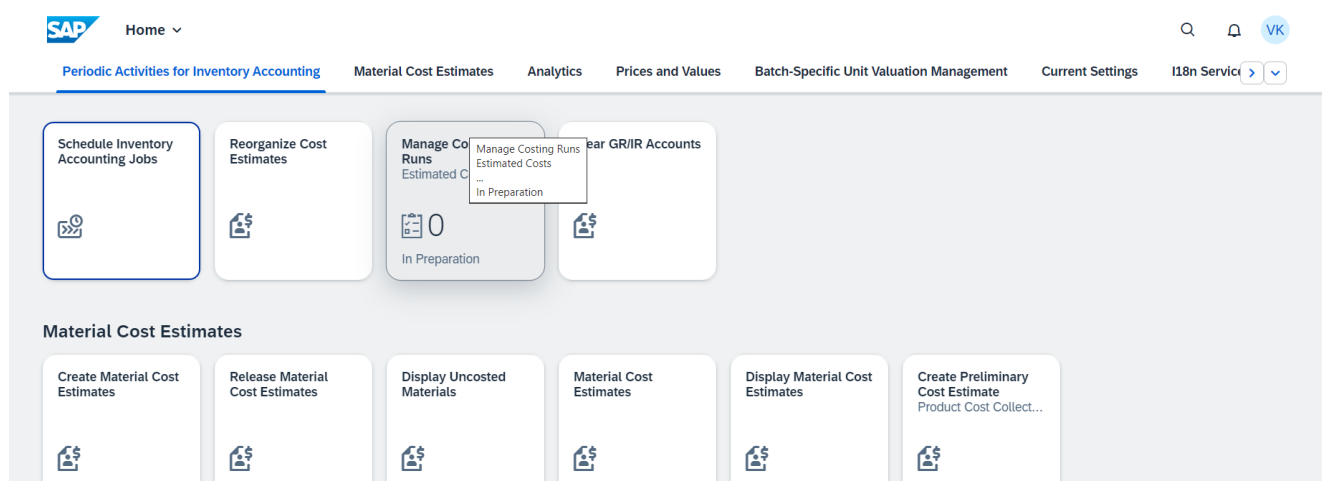


Рис. 3.82. SAP Fiori Launchpad

Скориставшись пошуком треба знайти підпрограму “Communication Arrangements”, та зайти в неї (рис. 3.83)

Standard ▾

Editing Status: Arrangement Name: Scenario ID: Communication System:  Adapt Filters

Arrangement Name	Scenario ID	Scenario	Communication System
<input type="checkbox"/> A	YY1_A	-	OLOVOMP
<input type="checkbox"/> CA_SAP_COM_0193	SAP_COM_0193	Identity Provisioning Integration	CUSTOMER_SAP_COM_0193
<input type="checkbox"/> MS_COM_0466	SAP_COM_0466	Printing - Pull Integration	CPM_COMM_SYS
<input type="checkbox"/> SAP_COM_0001	SAP_COM_0001	Employee Integration	OLOVOMP
<input type="checkbox"/> SAP_COM_0002	SAP_COM_0002	Finance - Posting Integration	OLOVOMP
<input type="checkbox"/> SAP_COM_0008	SAP_COM_0008	Business Partner, Customer and Supplier Integration	Z312680
<input type="checkbox"/> SAP_COM_0009	SAP_COM_0009	Product Integration	OLOVOMP
<input type="checkbox"/> SAP_COM_0053	SAP_COM_0053	Purchase Order Integration	TEST_SYSTEM_FOR_GUIDE
<input type="checkbox"/> SAP_COM_0053	SAP_COM_0053	Purchase Order Integration	OLOVOMP
<input type="checkbox"/> SAP_COM_0054	SAP_COM_0054	Commercial Project Integration	OLOVOMP
<input type="checkbox"/> SAP_COM_0057	SAP_COM_0057	Supplier Invoice Integration	OLOVOMP
<input type="checkbox"/> SAP_COM_0064	SAP_COM_0064	External Scheduler - Application Job Scheduling Integration	OLOVOMP
<input type="checkbox"/> SAP_COM_0077	SAP_COM_0077	Business Manufacturing Integration	OLOVOMP

Рис. 3.84. Communication arrangements

У цій підпрограмі регулюється доступ до різноманітних об'єктів у системі через HTTP-запити. Щоб створити новий треба натиснути на кнопку New та зайти ввести назву створюваного об'єкта, після чого треба ввести систему, до якої ставитиметься запит, та віртуального користувача, від імені якого відбуватиметься авторизація (рис. 3.85).

### Common Data

Arrangement Name:

Communication System: \*

### Outbound Communication

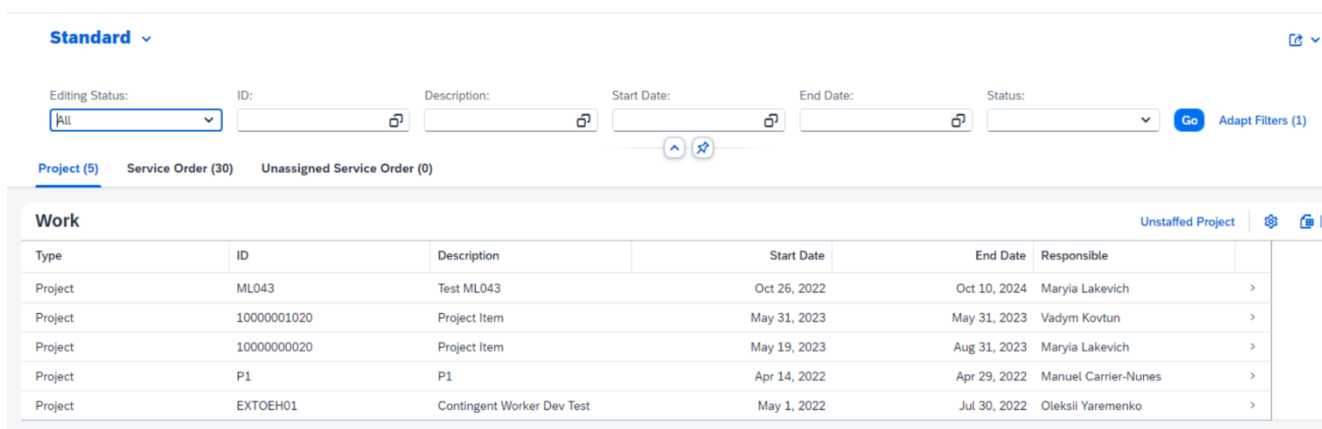
User Name:

Рис. 3.85. Налаштування Communication Arrangement

### 3.7. Тестування розробленої системи

Оскільки CDS View ZJTSC\_C\_Work позначена як корнева (root), буде логічно починати роботу додатка з неї. Для цього треба зайти у створений на рис. 3.80 сервіс, обрати сутність Work та натиснути на кнопку Preview. Відбувається перехід за відповідною URL-адресою, відкривається створена система.

Для того щоб завантижити дані, треба натиснути на кнопку Go (рис. 3.86).



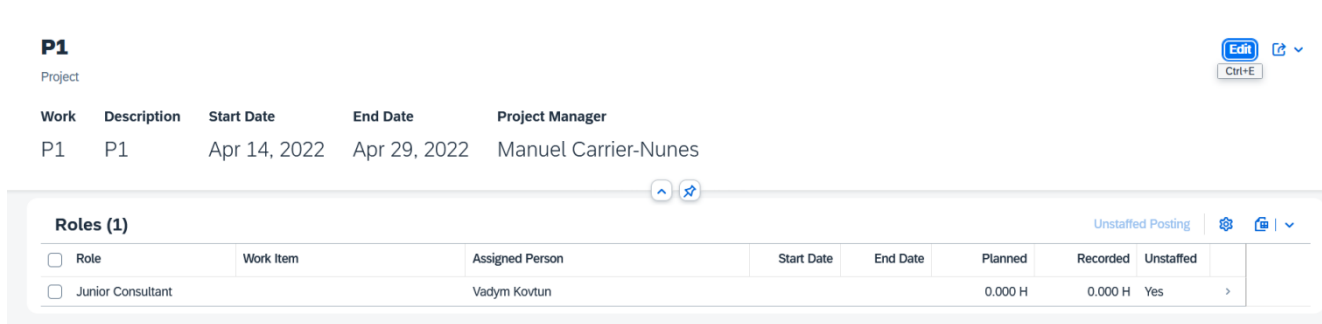
Type	ID	Description	Start Date	End Date	Responsible
Project	ML043	Test ML043	Oct 26, 2022	Oct 10, 2024	Maryia Lakevich
Project	10000001020	Project Item	May 31, 2023	May 31, 2023	Vadym Kovtun
Project	10000000020	Project Item	May 19, 2023	Aug 31, 2023	Maryia Lakevich
Project	P1	P1	Apr 14, 2022	Apr 29, 2022	Manuel Carrier-Nunes
Project	EXTOEH01	Contingent Worker Dev Test	May 1, 2022	Jul 30, 2022	Oleksii Yaremenko

Рис. 3.86. Створена система управління персоналом

Протестуємо основні функції системи:

- 1) Фіксування відпрацьованого часу

Щоб зафіксувати відпрацьований час необхідно знайти зверху вкладки, де вибрати опцію Project (рис. 3.86). Після цього треба зайти в проект, на якому треба зафіксувати відпрацьований час, та обрати посаду (рис. 3.87).



Work	Description	Start Date	End Date	Project Manager
P1	P1	Apr 14, 2022	Apr 29, 2022	Manuel Carrier-Nunes

Role	Work Item	Assigned Person	Start Date	End Date	Planned	Recorded	Unstaffed
<input type="checkbox"/>	Junior Consultant	Vadym Kovtun			0.000 H	0.000 H	Yes

Рис. 3.87. Вибір посади для фіксування часу

При цьому, якщо користувач не є керівником проекту, він бачитиме тільки свою посаду, що відбувається завдяки Access Control. Для фіксування часу треба натиснути на кнопку Edit, та обрати посаду, після чого відбувається перехід на ZJTSC\_C\_HoursRecorded (рис. 3.88).

The screenshot shows the 'Junior Consultant' profile page. At the top, it displays 'P1 / Junior Consultant' and a share icon. Below this is a summary table:

Role	Work Item	Person	Start Date	End Date	Current Month
Junior Consultant		Vadym Kovtun			0.00 Hours Recorded/0.00 Hours Planned (November)

Below the summary is a 'TimeSheet (1)' section with a table:

Date	Recorded Hours	Note
Aug 1, 2023	9.000 H	

Buttons for 'Create', 'Delete', and a settings icon are visible in the top right of the TimeSheet section.

Рис. 3.88. Рівень фіксації відпрацьованого часу

Щоб зафіксувати час за новий день, треба натиснути на Create. Видалимо існуючий запис та зафіксуємо час за 18 та 19 листопада (рис. 3.89).

The screenshot shows the 'Junior Consultant' profile page with the 'TimeSheet (2)' section updated. The summary table remains the same as in the previous screenshot.

Date	Recorded Hours	Note
Nov 18, 2023	5.000 H	
Nov 19, 2023	8.000 H	

Buttons for 'Create', 'Delete', and a settings icon are visible in the top right of the TimeSheet section. A 'Draft updated' notification is visible at the bottom right.

Рис. 3.89. Фіксування відпрацьованого часу

Щоб зміни вступили в силу, треба натиснути на Apply та Save. В результаті, час буде зафіксовано (рис. 3.90).

Work	Description	Start Date	End Date	Project Manager
P1	P1	Apr 14, 2022	Apr 29, 2022	Manuel Carrier-Nunes

Roles (1)		Assigned Person	Start Date	End Date	Planned	Recorded	Unstaffed
<input type="checkbox"/>	Junior Consultant	Vadym Kovtun			0.000 H	13.000 H	Yes

Рис. 3.90. Відпрацьований час зафіксовано


Можна бачити, що за посадою було відпрацьвано  $8 + 5 = 13$  годин, як і було вказано на рис. 3.89.

## 2) Додавання людей поза штатом проекту

Для цього треба натиснути на кнопку Unstaffed Project, та у з'явившемся вікні обрати проект, поза штатом якого знаходиться користувач, та вказати дату і кількість відпрацьованих годин (рис. 3.91).

### Unstaffed Project

Work Package:

DLICOTST1.1.1 (DL\_ICO\_TST1) 


Role:

T001

Work Item:

P002

TimeSheetDate:

20 нояб. 2023 г. 

Recorded Hours:

8,00

**Unstaffed Project** [Отменить](#)

Рис. 3.91. Вікно вводу фіксування відпрацьованих годин поза штатом.

На рис. 3.92 можна побачити, що проект додано до таких, що бачить користувач (оскільки він тепер там зафіксований).


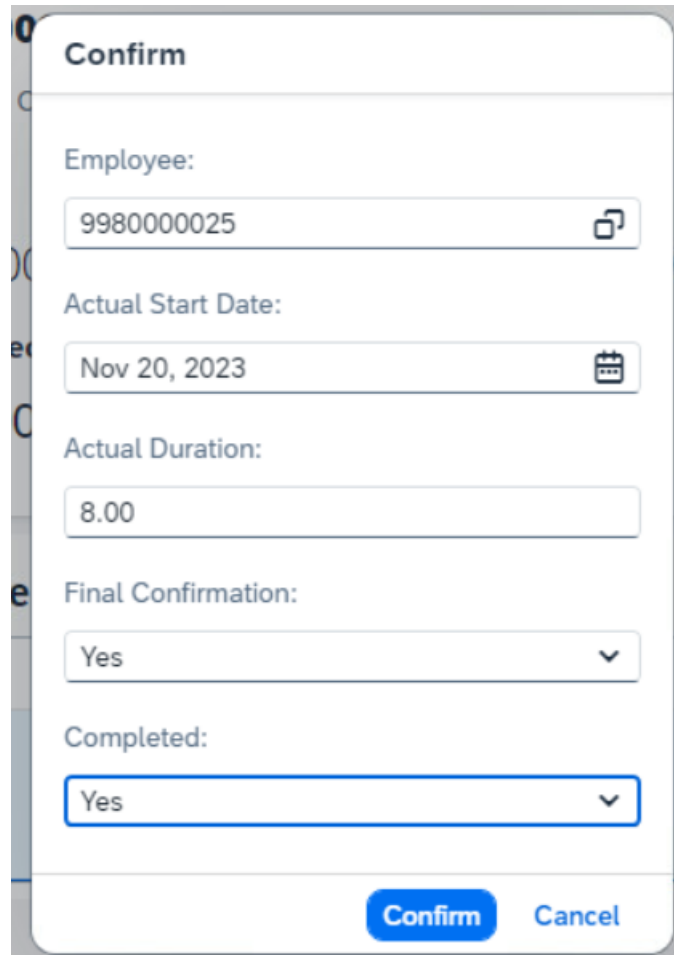
Work							Unstaffed Project	
Type	ID	Description	Start Date	End Date	Responsible			
Project	ML043	Test ML043	Oct 26, 2022	Oct 10, 2024	Maryia Lakevich		>	
Project	10000001020	Project Item	May 31, 2023	May 31, 2023	Vadym Kovtun		>	
Project	10000000020	Project Item	May 19, 2023	Aug 31, 2023	Maryia Lakevich		>	
Project	DLICOTST1	DL_ICO_TST1	Jul 1, 2023	Sep 1, 2024	Leslie Dennis		>	
Project	P1	P1	Apr 14, 2022	Apr 29, 2022	Manuel Carrier-Nunes		>	
Project	EXTOEH01	Contingent Worker Dev Test	May 1, 2022	Jul 30, 2022	Oleksii Yaremenko		>	

Рис. 3.92. На проект додано людину поза штатом

### 3) Підтвердження виконаних підзавдань

Щоб скористатися цією можливістю необхідно перейти на вкладку Service Order, обрати підзавдання, яке треба підтвердити, та натиснути на Confirm. З'явиться вікно підтвердження, як на рис. 3.93.



The image shows a mobile application dialog box titled "Confirm". It contains several input fields and dropdown menus. The fields are: "Employee:" with the value "9980000025" and a copy icon; "Actual Start Date:" with the value "Nov 20, 2023" and a calendar icon; "Actual Duration:" with the value "8.00"; "Final Confirmation:" with a dropdown menu showing "Yes"; and "Completed:" with a dropdown menu showing "Yes". At the bottom of the dialog, there are two buttons: "Confirm" (highlighted in blue) and "Cancel".

Рис. 3.93. Підтвердження виконання підзавдання

У вікні необхідно ввести дані про відповідального працівника, дату виконання, кількість відпрацьованих на підзадачі годин, та вказати чи остаточно виконане завдання.

Якщо зайти у підзавдання, можна побачити підтвердження (рис. 3.94).

800000080 / 10 Edit | [icon] | [icon] | [icon] | [icon]

Description	Product	Planned Quantity	Actual Quantity	Responsible Employee	Status
Service	SRV_01	1.000 HR	8.000 HR	Vadym Kovtun	Completed

[icon] [icon]

Service Confirmation Items (1)										Final Confirmation	Cancel Confirmation	[icon]	[icon]	[icon]	[icon]
<input type="checkbox"/>	Service Confirmati...	Item	Confirmation Desc...	Item Description	Quantity	Duration	Responsible Empl...	Service Employee	Final Confirmation	Cancelled Internally					
<input type="checkbox"/>	9000000150	10	test	Service	8.000 HR	8.000 HR	Vadym Kovtun	Vadym Kovtun	Yes	No					

Рис. 3.94. Створене підтвердження виконання підзавдання

#### 4) Скасування підтвердження виконаних підзавдань

Для скасування підтвердження необхідно обрати необхідне підтвердження та натиснути на Cancel Confirmation. Оберемо те, що на рис. 3.94. Після відпрацювання програмного коду, його статус зміниться на Cancelled Internally (рис. 3.95).

Description	Product	Planned Quantity	Actual Quantity	Responsible Employee	Status
Service	SRV_01	1.000 HR	8.000 HR	Vadym Kovtun	Completed

[icon] [icon]

Service Confirmation Items (1)										Final Confirmation	Cancel Confirmation	[icon]	[icon]	[icon]	[icon]
<input checked="" type="checkbox"/>	Service Confirmati...	Item	Confirmation Desc...	Item Description	Quantity	Duration	Responsible Empl...	Service Employee	Final Confirmation	Cancelled Internally					
<input checked="" type="checkbox"/>	9000000150	10	test	Service	8.000 HR	8.000 HR	Vadym Kovtun	Vadym Kovtun	Yes	Yes					

Рис. 3.95. Скасування підтвердження виконання завдання

#### 5) Призначення відповідального за виконання завдання

Для того щоб побачити завдання, на які ще не призначили відповідальних людей, треба перейти на вкладку Unassigned Service Order. Далі треба обрати необхідний, та натиснути на Self-Assign (рис. 3.96).



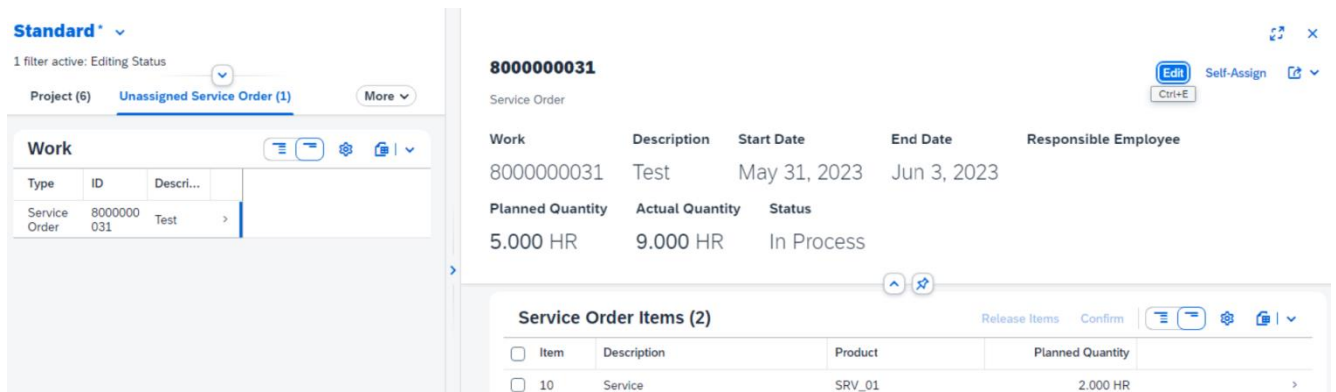


Рис. 3.96. Обрання необхідного завдання без відповідальної особи

Після натискання на кнопку, у полі Responsible Employee буде показано відповідального користувача (рис. 3.97).

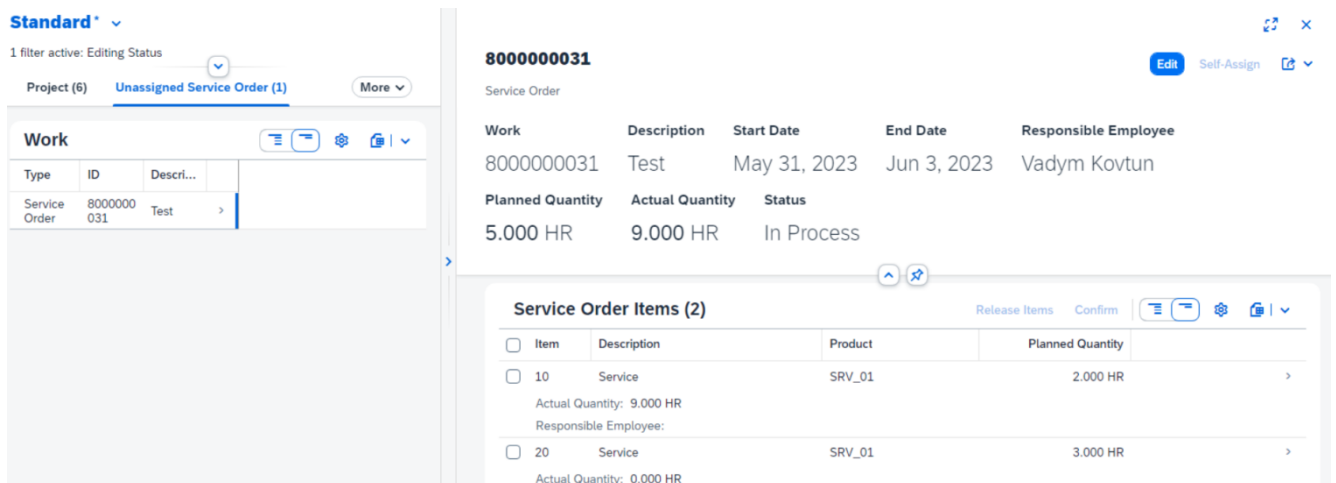


Рис. 3.97. Поточний користувач став відповідальним за виконання завдання

### б) Зміна статусу підзавдань на Released

Доки статус підзавдання не змінений на Released, його виконання не можна буде підтвердити. Щоб змінити статус завдання таким чином, користувачу необхідно обрати завдання, за яке він відповідальний, обрати підзавдання із статусом Open, та натиснути на Release Item (рис. 3.98). Після цього статус буде змінено (рис. 3.99).

800000031

Edit Self-Assign

Service Order

Work	Description	Start Date	End Date	Responsible Employee	Planned Quantity	Actual Quantity	Status
800000031	Test	May 31, 2023	Jun 3, 2023	Vadym Kovtun	8.000 HR	9.000 HR	In Process

Service Order Items (3)

Item	Description	Product	Planned Quantity	Actual Quantity	Responsible Employee
<input type="checkbox"/> 10	Service	SRV_01	2.000 HR	9.000 HR	Vadym Kovtun
Status: <span style="color: red;">▲ Released</span> WorkItemStatusCode: 2					
<input type="checkbox"/> 20	Service	SRV_01	3.000 HR	0.000 HR	Vadym Kovtun
Status: <span style="color: red;">▲ Released</span> WorkItemStatusCode: 2					
<input checked="" type="checkbox"/> 30	Service	SRV_01	3.000 HR	0.000 HR	Vadym Kovtun
Status: Open WorkItemStatusCode:					

Рис. 3.98. Обрано підзавдання із статусом Open

Service Order Items (3)

Item	Description	Product	Planned Quantity	Actual Quantity	Responsible Employee
<input type="checkbox"/> 10	Service	SRV_01	2.000 HR	9.000 HR	Vadym Kovtun
Status: <span style="color: red;">▲ Released</span> WorkItemStatusCode: 2					
<input type="checkbox"/> 20	Service	SRV_01	3.000 HR	0.000 HR	Vadym Kovtun
Status: <span style="color: red;">▲ Released</span> WorkItemStatusCode: 2					
<input checked="" type="checkbox"/> 30	Service	SRV_01	3.000 HR	0.000 HR	Vadym Kovtun
Status: <span style="color: red;">▲ Released</span> WorkItemStatusCode: 2					

Рис. 3.99. Статус підзавдання змінено

### 3.8. Висновки до розділу 3

При виконанні практичної частини було розроблено необхідні об'єкти та потрібним чином налаштовано систему SAP S/4 HANA Cloud для функціонування системи управління персоналом. Робота над системою йшла у такій послідовності:

- 1) Підготовка робочого простору (завантаження Eclipse IDE та ABAP Development Tools).
- 2) Створення пакету та Transport Request для ведення розробки.
- 3) Створення моделі даних (створення необхідних таблиць, CDS View, Projection View).
- 4) Визначення поведінки системи (написання програмного коду для роботи з системою, створення Behaviour Definition для CDS View та класів).
- 5) Розробка контролю доступу до системи (створення відповідних Access Control для CDS View).
- 6) Визначення інтерфейсу додатку та розробка сервісу (створення Metadata Extension для CDS View та публікація Service Binding).
- 7) Налаштування системи для коректної роботи додатку (створення Communication Arrangement).

В результаті було розроблено систему, що дозволяє працювати із фіксацією відпрацьованого часу та підтвердженням підзавдань у рамках одного додатку, в той час як за допомогою стандартних інструментів для цього необхідно користуватися трьома підпрограмами.

Систему було протестовано, вона успішно виконує заплановані функції: фіксація відпрацьованого часу, підтвердження виконання підзавдань, додавання людей поза штатом на проект, зміна статусу підзавдань на Released, призначення відповідального за виконання завдання, скасування підтвердження виконання підзавдання.

## ВИСНОВКИ

В процесі виконання дипломної роботи у Розділі 1 було розглянуто технологію ERP та проаналізовано в контексті управління персоналом різні програмні рішення, представлені на ринку: 1С підприємство, BAS ERP, Microsoft Dynamics, SAP S/4 HANA Cloud. Усі вони мають схожі функції, необхідні будь-якому підприємству, але для виконання дипломної роботи було обрано систему SAP S/4 HANA Cloud, через її можливість працювати у хмарі та простоту розробки.

Під час виконання Розділу 2 було розглянуто архітектуру SAP S/4 HANA Cloud, досліджено її стандартні інструменти з управління персоналом в контексті фіксування відпрацьованого робітниками часу та управління завданнями, розглянуто технологію розробки RAP. Також було розглянуто основну мову програмування у системі – ABAP. В результаті роботи над розділом було виявлено недоліки розглянутих стандартних засобів управління персоналом, звідки було сформовано вимоги для розроблюваної системи управління персоналом. Також було виявлено об'єкти у системі, з якими необхідно буде працювати під час виконання практичної частини.

Під час виконання Розділу 3 було розроблено систему управління персоналом: створено необхідні об'єкти у системі та налаштовано її для роботи із розробленим додатком. Створена система відповідає заявленим вимогам, а саме підтримує такі функції:

- 1) Фіксування відпрацьованого часу
- 2) Додавання на проект людей, що знаходяться поза штатом
- 3) Управління статусом підзавдань
- 4) Підтвердження виконання підзавдань
- 5) Відміна виконання підзавдань
- 6) Призначення відповідального за виконання завдання

Вважаю, що цінність даної розробки цілком підходить для використання у практичній сфері. Розроблена та детально описана інструкція з розробки системи управління персоналом на основі SAP S/4 HANA Cloud може бути використана у якості довідникового матеріалу для розробки додатків для цієї системи планування ресурсів підприємства.

## СПИСОК БІБЛОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційний сайт Нафтогазу: режим доступу: <https://www.naftogaz.com/news/grupa-naftogaz-integrue-it-systemy-na-osnovi-sap> (дата звернення 19.09.2023) .
2. Опис модуля управління персоналом у 1С: режим доступу: <https://sitis.com.ua/software/1c/1s-zarplata-i-upravlenie-personalom-bazovaya-versiya/> (дата звернення 19.09.2023) .
3. Огляд модуля управління персоналом в SAP S/4HANA Cloud: режим доступу: <https://blog.sap-press.com/what-does-hr-look-like-in-sap-s4hana> (дата звернення 19.09.2023) .
4. Офіційний сайт Microsoft Dynamics: режим доступу: <https://dynamics.microsoft.com/ru-ru/human-resources/overview/> (дата звернення 19.09.2023) .
5. Офіційний сайт BAS ERP: режим доступу: <https://www.bas-soft.eu/soft/bas-corp/bas-erp/> (дата звернення 19.09.2023) .
6. Огляд архітектури SAP S/4 HANA Cloud: режим доступу: <https://erpiseasy.com/2021/03/20/what-is-sap-s4hana/> (дата звернення 19.09.2023) .
7. Огляд технології розробки RAP: режим доступу: <https://community.sap.com/topics/abap/rap> (дата звернення 19.09.2023) .
8. Стаття про АВАР: режим доступу: <https://coderlessons.com/tutorials/sap/izuchite-sap-abap/sap-abap-obzor> (дата звернення 19.09.2023) .
9. Виконання програм у рамках трирівневої архітектури SAP: режим доступу: <http://sapbasistutorials.blogspot.com/p/the-r3-basis-system-overview-r3-basis.html> (дата звернення 19.09.2023) .

10. ДСТУ 3008-15 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

11. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.

12. ABAP in the Cloud. Development and Operations with SAP BTP, ABAP environment / Gairic Acharya [та ін.] – Rheinwerk Publishing, 2021 – 316 с.

13. Application Development with SAP Business Technology Platform / Gairic Acharya [та ін.] – Rheinwerk Publishing, 2023 – 574 с.

14. SAP RESTfull Application Programming Model / Lutz Baumbush [та ін.] – Rheinwerk Publishing, 2023 – 508 с.

## ДОДАТОК А

Програмний код, що описує поведінку системи управління персоналом.

```
CLASS lhc_serviceconfirmationitem DEFINITION INHERITING FROM cl_abap_behavior_handler.

PRIVATE SECTION.

METHODS get_instance_features FOR INSTANCE FEATURES
  IMPORTING keys REQUEST requested_features FOR serviceconfirmationitem RESULT result.
METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
  IMPORTING keys REQUEST requested_authorizations FOR serviceconfirmationitem RESULT
result.

METHODS cancelconfirmation FOR MODIFY
  IMPORTING keys FOR ACTION serviceconfirmationitem~cancelconfirmation RESULT result.

METHODS finalconfirmation FOR MODIFY
  IMPORTING keys FOR ACTION serviceconfirmationitem~finalconfirmation RESULT result.

ENDCLASS.

CLASS lhc_serviceconfirmationitem IMPLEMENTATION.

METHOD get_instance_features.
  READ ENTITY IN LOCAL MODE zjtsc_i_serviceconfirmitemtp
  FIELDS ( servicedocumentitemstatus servicedocumentitemisrejected isfinal ) WITH
CORRESPONDING #( keys )
  RESULT DATA(lt_confirm_item_draft).

  READ ENTITY IN LOCAL MODE zjtsc_i_serviceconfirmitemtp
  FIELDS ( servicedocumentitemstatus servicedocumentitemisrejected isfinal ) WITH
CORRESPONDING #( keys EXCEPT %is_draft )
  RESULT DATA(lt_confirm_item).

  result = VALUE #( FOR key IN keys ( %tky = key-%tky
                                     %features-%update = COND #( WHEN (
lt_confirm_item[ %key = key-%key ]-servicedocumentitemstatus = 'C' )
                                     THEN if_abap_behv=>fc-
o-disabled
                                     ELSE if_abap_behv=>fc-
o-enabled )
                                     %features-%action-cancelconfirmation = COND #(
WHEN ( lt_confirm_item_draft[ %tky = key-%tky ]-servicedocumentitemisrejected = abap_false
)
THEN if_abap_behv=>fc-o-enabled
ELSE if_abap_behv=>fc-o-disabled )
                                     %features-%action-finalconfirmation = COND #( WHEN
( lt_confirm_item_draft[ %tky = key-%tky ]-isfinal = abap_true OR
lt_confirm_item_draft[ %tky = key-%tky ]-servicedocumentitemisrejected = abap_true OR
lt_confirm_item[ %key = key-%key ]-servicedocumentitemstatus = 'C' )
                                     THEN
if_abap_behv=>fc-o-disabled
                                     ELSE
if_abap_behv=>fc-o-enabled )
                                     ) ).

```



```

ENDMETHOD.

METHOD get_instance_authorizations.
ENDMETHOD.

METHOD cancelconfirmation.
  IF keys IS NOT INITIAL.
    SELECT * FROM zjtsc_i_serviceconfirmitemtp
    FOR ALL ENTRIES IN @keys
    WHERE serviceconfirmation = @keys-serviceconfirmation
    INTO TABLE @DATA(lt_confirm_item).

    MODIFY ENTITIES OF zjtsc_i_worktvp IN LOCAL MODE
      ENTITY serviceconfirmationitem
      UPDATE FIELDS ( servicedocumentitemisrejected )
      WITH VALUE #( FOR item IN lt_confirm_item
        ( %key-serviceconfirmation = item-serviceconfirmation
          %key-serviceconfirmationitem = item-serviceconfirmationitem
          %is_draft = keys[ 1 ]-%is_draft
          servicedocumentitemisrejected = abap_true
          %control-servicedocumentitemisrejected = if_abap_behv=>mk-on ) )

      REPORTED reported
      FAILED failed.

  IF failed IS INITIAL.

    SELECT DISTINCT serviceconfirmation
    FROM @lt_confirm_item AS confirm
    INTO TABLE @DATA(lt_changed_confirmations).

    LOOP AT lt_changed_confirmations ASSIGNING FIELD-SYMBOL(<fs_changed>).
      APPEND INITIAL LINE TO reported-%other ASSIGNING FIELD-SYMBOL(<fs_reported>).
      <fs_reported> = new_message_with_text( text = |Service Confirmation {
<fs_changed>-serviceconfirmation } marked as canceled|
                                          severity =
if_abap_behv_message=>severity-success ).
      ENDLIST.
    ENDIF.
  ENDIF.
ENDMETHOD.

METHOD finalconfirmation.
  IF keys IS NOT INITIAL.
    SELECT * FROM zjtsc_i_serviceconfirmitemtp
    FOR ALL ENTRIES IN @keys
    WHERE serviceconfirmation = @keys-serviceconfirmation
    INTO TABLE @DATA(lt_confirm_item).

    MODIFY ENTITIES OF zjtsc_i_worktvp IN LOCAL MODE
      ENTITY serviceconfirmationitem
      UPDATE FIELDS ( serviceconfirmationisfinal isfinal )
      WITH VALUE #( FOR item IN lt_confirm_item
        ( %key-serviceconfirmation = item-serviceconfirmation
          %key-serviceconfirmationitem = item-serviceconfirmationitem
          %is_draft = keys[ 1 ]-%is_draft
          serviceconfirmationisfinal = 'Y'
          isfinal = abap_true
          %control-serviceconfirmationisfinal = if_abap_behv=>mk-on
          %control-isfinal = if_abap_behv=>mk-on ) )

      REPORTED reported
      FAILED failed.
  
```

```

IF failed IS INITIAL.

    SELECT DISTINCT serviceconfirmation
    FROM @lt_confirm_item AS confirm
    INTO TABLE @DATA(lt_changed_confirmations).

    LOOP AT lt_changed_confirmations ASSIGNING FIELD-SYMBOL(<fs_changed>).
        APPEND INITIAL LINE TO reported-%other ASSIGNING FIELD-SYMBOL(<fs_reported>).
        <fs_reported> = new_message_with_text( text = |Service Confirmation {
<fs_changed>-serviceconfirmation } marked as final|
                                                severity =
if_abap_behv_message=>severity-success ).
        ENDLLOOP.
    ENDIF.
ENDIF.
ENDMETHOD.

ENDCLASS.

CLASS lhc_work DEFINITION INHERITING FROM cl_abap_behavior_handler.
PRIVATE SECTION.

METHODS get_instance_features FOR INSTANCE FEATURES
IMPORTING keys REQUEST requested_features FOR work RESULT result.

METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
IMPORTING keys REQUEST requested_authorizations FOR work RESULT result.

METHODS selfassign FOR MODIFY
IMPORTING keys FOR ACTION work~selfassign RESULT result.
METHODS unstaffedproject FOR MODIFY
IMPORTING keys FOR ACTION work~unstaffedproject.

ENDCLASS.

CLASS lhc_work IMPLEMENTATION.

METHOD get_instance_features.

    READ ENTITIES OF zjtsc_i_worktvp IN LOCAL MODE
    ENTITY work
    ALL FIELDS WITH CORRESPONDING #( keys )
    RESULT DATA(lt_order).

    result = VALUE #( FOR order IN lt_order ( %tky = order-%tky
                                                %features-%action-selfassign = COND #( WHEN
( order-type = 'S' AND
order-responsibleemployee IS INITIAL )
                                                THEN
if_abap_behv=>fc-o-enabled
                                                ELSE
if_abap_behv=>fc-o-disabled )
                                                ) ).
    ENDMETHOD.

METHOD get_instance_authorizations.
ENDMETHOD.

```

```

METHOD selfassign.
  READ TABLE keys ASSIGNING FIELD-SYMBOL(<fs_key>) INDEX 1.
  IF sy-subrc EQ 0.

    SELECT SINGLE id FROM zjtsc_i_worktp
      WHERE objectuuid = @<fs_key>-objectuuid
      INTO @DATA(lv_service_order).

    SELECT SINGLE businesspartner, personfullname
      FROM i_businessuserbasic
      WHERE userid = @sy-uname
      INTO @DATA(lv_business_partner).

    DATA(lv_body) = |\{ "ServiceOrder": "{ lv_service_order }",\n| &&
      | "PersonResponsible": "{ lv_business_partner-businesspartner
}","\n| &&
      | "CustMgmtPartnerIsMainPartner": true\n\}||.

    zjtsc_cl_request=>method_post(
      EXPORTING
        iv_path          = zjtsc_if_constants=>c_destination-service_order_api
        iv_next_path     = zjtsc_if_constants=>c_entity-service_order_responsible
        iv_body          = lv_body
      IMPORTING
        ev_response_satus = DATA(lv_resp_satus)
        ev_response_text  = DATA(lv_resp_text)
    ).

    IF lv_resp_satus-code EQ '201'.

      APPEND VALUE #( %tky = <fs_key>-%tky
        %msg = new_message( id = '00' number = '398'
          severity = if_abap_behv_message=>severity-
success
          v1 = |Service Order| v2 = lv_service_order v3
= |assigned to| v4 = lv_business_partner-personfullname )
          ) TO reported-work.

    ELSE.
      zjtsc_cl_request=>parse_response_error(
        EXPORTING
          iv_response = lv_resp_text
        IMPORTING
          et_result   = DATA(lt_errors)
      ).

      failed-work = CORRESPONDING #( keys ).
      LOOP AT failed-work ASSIGNING FIELD-SYMBOL(<fs_failed>).
        LOOP AT lt_errors ASSIGNING FIELD-SYMBOL(<fs_error>).
          IF NOT ( sy-tabix > 1 AND ( <fs_error>-message EQ 'MESSAGE' OR <fs_error>-
message CS 'exception' ) ).
            APPEND CORRESPONDING #( <fs_failed> ) TO reported-work ASSIGNING FIELD-
SYMBOL(<fs_reported>).
            <fs_reported>-%msg = new_message(
              id          = |\{ <fs_error>-longtext-msgid }|
              number     = |\{ <fs_error>-longtext-msgno }|
              severity   = if_abap_behv_message=>severity-error
              v1         = <fs_error>-longtext-message_v1
              v2         = <fs_error>-longtext-message_v2
              v3         = <fs_error>-longtext-message_v3
              v4         = <fs_error>-longtext-message_v4
            ).
          
```

```

        ).
    ENDIF.
  ENDOLOOP.
ENDLOOP.
ENDIF.
ENDIF.

SELECT * FROM zjtsc_i_worktp
  FOR ALL ENTRIES IN @keys
  WHERE objectuuid = @keys-objectuuid
  INTO TABLE @DATA(lt_order).

result = VALUE #( FOR order_key IN keys
  ( %tky = order_key-%tky
    %param = CORRESPONDING #( lt_order[ objectuuid = order_key-
objectuuid ] ) ) ).
  ENDMETHOD.

METHOD unstaffedproject.

  DATA: lv_decimal TYPE p DECIMALS 2,
        lv_integral TYPE i.

  SELECT businesspartner, userid, work_agreement~personworkagreement,
  personworkagreementexternalid, companycode
    FROM i_businessuserbasic          AS user_basic
    INNER JOIN i_personworkagreement_1 AS work_agreement ON user_basic~businesspartner =
work_agreement~person
    INNER JOIN i_personworkagrmttoexternalid AS external_id ON
work_agreement~personworkagreement = external_id~personworkagreement
    WHERE userid = @sy-uname
    INTO TABLE @DATA(lt_person_data).

  READ TABLE lt_person_data ASSIGNING FIELD-SYMBOL(<fs_person_data>) INDEX 1.
  IF sy-subrc EQ 0.

    LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_key>).

      DATA(lv_unix_time) = zjtsc_cl_request=>convert_date_to_unix_time( iv_date =
<fs_key>-%param-timesheetdate ).

      lv_integral = trunc( <fs_key>-%param-recordedhours ).
      IF lv_integral / <fs_key>-%param-recordedhours <> 1 .
        lv_decimal = <fs_key>-%param-recordedhours - lv_integral.
        lv_decimal = lv_decimal * 100 / 60.
        lv_decimal = lv_integral + lv_decimal.
      ELSE.
        lv_decimal = <fs_key>-%param-recordedhours.
      ENDIF.

      DATA(lv_body) = | \{ "TimeSheetDataFields":\{\n| &
| "ActivityType": "{ <fs_key>-%param-role }",\r\n| &
| "WBSElement": "{ <fs_key>-%param-workpackage }",\r\n| &
| "WorkItem": "{ <fs_key>-%param-workitem }",\r\n| &
| "RecordedHours": "{ lv_decimal }",\r\n| &
| "HoursUnitOfMeasure": "H"\r\n| &
| \}, "PersonWorkAgreementExternalID": "{ <fs_person_data>-
personworkagreementexternalid }",\n| &
| "CompanyCode": "{ <fs_person_data>-companycode }",\r\n| &

```

```

}","\r\n| &
| "PersonWorkAgreement": "{ <fs_person_data>-personworkagreement
| "TimeSheetDate": "/Date({ lv_unix_time })/","\r\n| &
| "TimeSheetOperation": "C"\}\r\n|.

zjtsc_cl_request=>method_post(
  EXPORTING
    iv_path          = zjtsc_if_constants=>c_destination-time_sheet_api
    iv_next_path     = zjtsc_if_constants=>c_entity-time_sheet
    iv_body          = lv_body
  IMPORTING
    ev_response_satus = DATA(ls_response_satus)
    ev_response_text  = DATA(lv_response_text)
).

IF ls_response_satus-code EQ '201'.

  DATA(lv_timesheet) = substring_after( val = lv_response_text sub =
|TimeSheetRecord='| ).
  lv_timesheet = substring_before( val = lv_timesheet sub = '|| ).

  APPEND VALUE #( timesheetrecord = lv_timesheet
    %msg = new_message_with_text(
      severity = if_abap_behv_message=>severity-success
      text     = |Time Sheet Record { lv_timesheet ALPHA =
OUT } created.|
    ) ) TO reported-timesheet.

  SELECT SINGLE projectuuid FROM zjtsc_i_workpackage
    WHERE workpackage = @<fs_key>-%param-workpackage
    INTO @DATA(lv_project).

  APPEND INITIAL LINE TO mapped-work ASSIGNING FIELD-SYMBOL(<fs_mapped>).
  <fs_mapped>-%cid = <fs_key>-%cid.
  <fs_mapped>-objectuuid = lv_project.

ELSE.
  zjtsc_cl_request=>parse_response_error(
    EXPORTING
      iv_response = lv_response_text
    IMPORTING
      et_result   = DATA(lt_errors)
  ).

  LOOP AT lt_errors ASSIGNING FIELD-SYMBOL(<fs_error>).
    IF NOT ( sy-tabix > 1 AND <fs_error>-message EQ 'MESSAGE' ).
      APPEND INITIAL LINE TO reported-%other ASSIGNING FIELD-
SYMBOL(<fs_reported>).
      <fs_reported> = new_message(
        id          = |{ <fs_error>-longtext-msgid }|
        number     = |{ <fs_error>-longtext-msgno }|
        severity   = if_abap_behv_message=>severity-error
        v1         = <fs_error>-longtext-message_v1
        v2         = <fs_error>-longtext-message_v2
        v3         = <fs_error>-longtext-message_v3
        v4         = <fs_error>-longtext-message_v4
      ).

    ENDIF.
  ENDLIST.
ENDLOOP.

ENDIF.

```

```

        ENDLOOP.
    ENDIF.

ENDMETHOD.

ENDCLASS.

CLASS lhc_serviceorderitem DEFINITION INHERITING FROM cl_abap_behavior_handler.
    PRIVATE SECTION.

        METHODS get_instance_features FOR INSTANCE FEATURES
            IMPORTING keys REQUEST requested_features FOR serviceorderitem RESULT result.

        METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
            IMPORTING keys REQUEST requested_authorizations FOR serviceorderitem RESULT result.

        METHODS confirm FOR MODIFY
            IMPORTING keys FOR ACTION serviceorderitem~confirm RESULT result.
        METHODS releaseitems FOR MODIFY
            IMPORTING keys FOR ACTION serviceorderitem~releaseitems RESULT result.

ENDCLASS.

CLASS lhc_serviceorderitem IMPLEMENTATION.

    METHOD get_instance_features.
        READ ENTITIES OF zjtsc_i_worktp IN LOCAL MODE
        ENTITY serviceorderitem
        ALL FIELDS WITH CORRESPONDING #( keys )
        RESULT DATA(lt_order_item).

        SELECT SINGLE businesspartner
        FROM i_businessuserbasic
        WHERE userid = @sy-uname
        INTO @DATA(lv_business_partner).
        result = VALUE #( FOR item IN lt_order_item ( %tky = item-%tky
                                                    %features-%action-confirm = COND #( WHEN
( item-servicedocumentitemstatus = 'C' OR
keys[ %tky = item-%tky ]-%is_draft = if_abap_behv=>mk-on
)
                                                    THEN
if_abap_behv=>fc-o-disabled
                                                    ELSE
if_abap_behv=>fc-o-enabled )
                                                    %features-%action-releaseitems = COND #(
WHEN ( item-responsibleemployee = lv_business_partner AND
keys[ %tky = item-%tky ]-%is_draft = if_abap_behv=>mk-off )
                                                    THEN
if_abap_behv=>fc-o-enabled
                                                    ELSE
if_abap_behv=>fc-o-disabled )
                                                    )
                                                    ).

    ENDMETHOD.

```

```

METHOD get_instance_authorizations.
ENDMETHOD.

METHOD confirm.
  DATA: lv_items      TYPE string.

  READ TABLE keys ASSIGNING FIELD-SYMBOL(<fs_service_order_key>) INDEX 1.
  IF sy-subrc EQ 0.

    READ ENTITIES OF i_serviceordertp
      ENTITY serviceorder
      ALL FIELDS WITH VALUE #( ( serviceorder = <fs_service_order_key>-serviceorder ) )
      RESULT DATA(lt_order).

    READ TABLE lt_order ASSIGNING FIELD-SYMBOL(<fs_order_data>) INDEX 1.
    IF sy-subrc EQ 0.

      DATA(lv_unix_time) = zjtsc_cl_request=>convert_date_to_unix_time( iv_date =
<fs_service_order_key>-%param-actualstartdate ).

      READ ENTITIES OF i_serviceordertp
        ENTITY serviceorderitem
        ALL FIELDS WITH CORRESPONDING #( keys )
        RESULT DATA(lt_order_item).

      LOOP AT lt_order_item ASSIGNING FIELD-SYMBOL(<fs_order_item_data>) WHERE
serviceorder = <fs_order_data>-serviceorder.
        lv_items = |{ lv_items }{ COND #( WHEN sy-tabix = 1 THEN '' ELSE ',' ) }| &&
| \{ "Product": "{ <fs_order_item_data>-product }",\n| &&
| "Quantity": "{ <fs_service_order_key>-%param-actualduration
}| "ActualServiceDuration": "{ <fs_service_order_key>-%param-
}| "ServiceConfItemIsCompleted": "{ <fs_service_order_key>-%param-
}| "ExecutingServiceEmployee": "{ <fs_service_order_key>-%param-
}| "ReferenceServiceOrder": "{ <fs_order_item_data>-serviceorder
}| "ReferenceServiceOrderItem": "{ <fs_order_item_data>-
}| "ActualServiceStartDateTime": "/Date({ lv_unix_time })/" \n\}|.
        ENDLOOP.

      DATA(lv_body) = | \{ "ServiceConfirmationType": "SVC1",\n| &&
| "ReferenceServiceOrder": "{ <fs_order_data>-serviceorder
}| "ServiceConfirmationIsFinal": "{ COND #( WHEN
<fs_service_order_key>-%param-isfinal = abap_true
| THEN 'Y'
| ELSE 'N' ) }",\n| &&
| "to_Item": \{\n| &&
| "results": [\n{ lv_items }\n]\}\}|.

      zjtsc_cl_request=>method_post(
        EXPORTING
          iv_path          = zjtsc_if_constants=>c_destination-service_confirmation_api
          iv_next_path     = zjtsc_if_constants=>c_entity-service_confirmation
          iv_body           = lv_body
        IMPORTING

```

```

        ev_response_satus = DATA(lv_resp_satus)
        ev_response_text = DATA(lv_resp_text)
    ).

    IF lv_resp_satus-code EQ '201'.
        DATA(lv_confirmation) = substring_after( val = lv_resp_text sub =
|A_ServiceConfirmation('|' ).
        lv_confirmation = substring_before( val = lv_confirmation sub = '|' ).

        APPEND VALUE #( %key = <fs_service_order_key>-%key
                        %msg = new_message_with_text(
                            severity = if_abap_behv_message=>severity-success
                            text      = |Service Confirmation { lv_confirmation }
created. |
                                ) ) TO reported-serviceconfirmationitem.
*
* IF keys[ 1 ]-%is_draft = if_abap_behv=>mk-on.
* SELECT * FROM zjtsc_i_serviceconfirmationitemtp
* FOR ALL ENTRIES IN @lt_order_item
* WHERE serviceconfirmation = @lv_confirmation AND
*         serviceorder = @lt_order_item-serviceorder AND
*         serviceorderitem = @lt_order_item-serviceorderitem
* INTO TABLE @DATA(lt_new_confirmation_item).
*
* MODIFY ENTITIES OF zjtsc_i_worktvp IN LOCAL MODE
* ENTITY serviceorderitem
* CREATE BY \_serviceconfirmationitem
* AUTO FILL CID
* SET FIELDS WITH VALUE #( FOR item IN keys
*                           ( %tky = item-%tky
*                             %target = VALUE #( FOR confirm IN
lt_new_confirmation_item WHERE ( serviceorder = item-serviceorder AND
*
serviceorderitem = item-serviceorderitem )
*                                     ( "%key = CORRESPONDING #(
confirm )
*                                     %is_draft = if_abap_behv=>mk-
on
*                                     %data = CORRESPONDING #(
confirm ) ) ) ) )
*         FAILED failed
*         REPORTED reported.
*         ENDIF.

ELSE.
    zjtsc_cl_request=>parse_response_error(
        EXPORTING
            iv_response = lv_resp_text
        IMPORTING
            et_result    = DATA(lt_errors)
    ).

    failed-serviceorderitem = CORRESPONDING #( keys ).
    LOOP AT failed-serviceorderitem ASSIGNING FIELD-SYMBOL(<fs_failed>).
        LOOP AT lt_errors ASSIGNING FIELD-SYMBOL(<fs_error>).
            IF NOT ( sy-tabix > 1 AND <fs_error>-message EQ 'MESSAGE' ).
                APPEND CORRESPONDING #( <fs_failed> ) TO reported-serviceorderitem
ASSIGNING FIELD-SYMBOL(<fs_reported>).
                <fs_reported>-%msg = new_message(
                    id      = |{ <fs_error>-longtext-msgid }|
                    number  = |{ <fs_error>-longtext-msgno }|
                    severity = if_abap_behv_message=>severity-error

```



```

v1      = <fs_error>-longtext-message_v1
v2      = <fs_error>-longtext-message_v2
v3      = <fs_error>-longtext-message_v3
v4      = <fs_error>-longtext-message_v4
        ).
    ENDIF.
  ENDLOOP.
ENDLOOP.
ENDIF.

ENDIF.
ENDIF.

SELECT * FROM zjtsc_i_serviceorderitemtp
FOR ALL ENTRIES IN @keys
WHERE serviceorder = @keys-serviceorder
      AND serviceorderitem = @keys-serviceorderitem
INTO TABLE @DATA(lt_order_items).

result = VALUE #( FOR item_key IN keys
                  ( serviceorder = item_key-serviceorder
                    serviceorderitem = item_key-serviceorderitem
                    %param = CORRESPONDING #( lt_order_items[ serviceorder = item_key-
serviceorder
                                                                    serviceorderitem =
item_key-serviceorderitem ] ) ) ).
ENDMETHOD.

METHOD releaseitems.

DATA(lv_body) = |\{"ServiceOrderItemIsReleased": "X"\}|.
LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_key>).
  DATA(lv_parameters) = |(ServiceOrder='{ <fs_key>-serviceorder }',ServiceOrderItem='{
<fs_key>-serviceorderitem }')|.
  DATA(lv_next_path) = zjtsc_if_constants=>c_entity-service_order_item &&
lv_parameters.
  zjtsc_cl_request=>method_patch(
    EXPORTING
      iv_path      = zjtsc_if_constants=>c_destination-service_order_api
      iv_next_path = lv_next_path
      iv_body      = lv_body
    IMPORTING
      ev_response_satus = DATA(lv_resp_satus)
      ev_response_text  = DATA(lv_resp_text)
  ).
  IF lv_resp_satus-code NE '204' .
    zjtsc_cl_request=>parse_response_error(
      EXPORTING
        iv_response = lv_resp_text
      IMPORTING
        et_result   = DATA(lt_errors)
    ).
  ENDIF.
ENDLOOP.
ENDMETHOD.

ENDCLASS.

CLASS lhc_role DEFINITION INHERITING FROM cl_abap_behavior_handler.
PRIVATE SECTION.

```

```

METHODS get_instance_features FOR INSTANCE FEATURES
  IMPORTING keys REQUEST requested_features FOR role RESULT result.
METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
  IMPORTING keys REQUEST requested_authorizations FOR role RESULT result.

METHODS unstaffedposting FOR MODIFY
  IMPORTING keys FOR ACTION role~unstaffedposting RESULT result.
METHODS earlynumbering_cba_timesheet FOR NUMBERING
  IMPORTING entities FOR CREATE role\_timesheet.

ENDCLASS.

CLASS lhc_role IMPLEMENTATION.

  METHOD get_instance_features.

    READ ENTITIES OF zjtsc_i_worktsp IN LOCAL MODE
      ENTITY role BY \_project
      ALL FIELDS WITH CORRESPONDING #( keys )
      RESULT DATA(lt_project).

    READ ENTITIES OF zjtsc_i_worktsp IN LOCAL MODE
      ENTITY role
      ALL FIELDS WITH CORRESPONDING #( keys )
      RESULT DATA(lt_role).

    SELECT SINGLE businesspartneruuid FROM i_businessuserbasic
      WHERE userid = @sy-uname
      INTO @DATA(lv_useruuid).

    result = VALUE #( FOR role IN lt_role ( %tky = role-%tky
      %features-%action-unstaffedposting = COND #(
WHEN ( role-assignmentstatus IS INITIAL OR
*
role-%is_draft = if_abap_behv=>mk-off OR
NOT line_exists( lt_project[ objectuuid = role-projectuuid
projectmanager = lv_useruuid ] ) )
THEN if_abap_behv=>fc-o-disabled
ELSE if_abap_behv=>fc-o-enabled ) ) ).
    ENDMETHOD.

  METHOD earlynumbering_cba_timesheet.
    SELECT MAX( timesheetrecord ) FROM zjtsc_i_hoursrecorded
      INTO @DATA(lv_last_record).
    SELECT MAX( timesheetrecord ) FROM zjtsc_hourec_d
      INTO @DATA(lv_last_record_draft).
    DATA(lv_latest_record) = COND #( WHEN |{ lv_last_record ALPHA = OUT }| >
lv_last_record_draft
      THEN |{ lv_last_record ALPHA = OUT }| "lv_last_record
      ELSE lv_last_record_draft ).
    LOOP AT entities ASSIGNING FIELD-SYMBOL(<fs_entity>).
      LOOP AT <fs_entity>-%target ASSIGNING FIELD-SYMBOL(<fs_target>).
        APPEND CORRESPONDING #( <fs_target> ) TO mapped-timesheet ASSIGNING FIELD-
SYMBOL(<fs_timesheet>).
        lv_latest_record += 1.

```

```

    <fs_timesheet>-timesheetrecord = lv_latest_record."|{ lv_latest_record ALPHA = IN
}|.
    ENDLOOP.
ENDLOOP.

ENDMETHOD.

METHOD get_instance_authorizations.
ENDMETHOD.

METHOD unstaffedposting.
    DATA: lv_decimal    TYPE p DECIMALS 2,
           lv_integral   TYPE i,
           lt_batch_data TYPE zjtsc_cl_request=>tt_batch_data.

    SELECT * FROM i_personworkagrmttoexternalid
    FOR ALL ENTRIES IN @keys
    WHERE personworkagreement = @keys-%param-personworkagreement
    INTO TABLE @DATA(lt_personworkagrmt).

*    SELECT * FROM i_personworkagreement_1
*    FOR ALL ENTRIES IN @keys
*    WHERE personworkagreement = @keys-%param-personworkagreement
*    INTO TABLE @DATA(lt_bussiness_user).

*    READ ENTITIES OF zjtsc_i_worktvp IN LOCAL MODE
*    ENTITY role BY \_workpackage
*    ALL FIELDS WITH CORRESPONDING #( keys )
*    RESULT DATA(lt_work_package).
READ ENTITIES OF zjtsc_i_worktvp IN LOCAL MODE
ENTITY role
ALL FIELDS WITH CORRESPONDING #( keys )
RESULT DATA(lt_role).

READ ENTITIES OF zjtsc_i_worktvp IN LOCAL MODE
ENTITY role BY \_project
ALL FIELDS WITH CORRESPONDING #( keys )
RESULT DATA(lt_project).

DATA(lv_project) = lt_project[ 1 ]-objectuuid.
SELECT SINGLE responsiblecostcenter FROM zjtsc_i_project
WHERE projectuuid = @lv_project INTO @DATA(lv_costcenter).

LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_key>).
    lv_integral = trunc( <fs_key>-%param-recordedhours ).
    IF lv_integral / <fs_key>-%param-recordedhours <> 1 .
        lv_decimal = <fs_key>-%param-recordedhours - lv_integral.
        lv_decimal = lv_decimal * 100 / 60.
        lv_decimal = lv_integral + lv_decimal.
    ELSE.
        lv_decimal = <fs_key>-%param-recordedhours.
    ENDIF.
    DATA(lv_unix_time) = zjtsc_cl_request=>convert_date_to_unix_time( iv_date =
<fs_key>-%param-timesheetdate ).
    DATA(ls_person) = VALUE #( lt_personworkagrmt[ personworkagreement = <fs_key>-
%param-personworkagreement ] OPTIONAL ).
    APPEND VALUE #( method = zjtsc_cl_request=>methods-post
                    content_id = <fs_key>-%key
                    path       = zjtsc_if_constants=>c_entity-time_sheet
                    body       = | \{ "TimeSheetDataFields":\{\n| &

```

```

        | "ActivityType":{" <fs_key>-role }",\r\n| &
        | "WBSElement":{" VALUE #( lt_role[ %tky = <fs_key>-
%tky ]-workpackage OPTIONAL ) }",\r\n| &
        | "WorkItem":{" <fs_key>-workitem }",\r\n| &
        | "SenderCostCenter":{" lv_costcenter }",\r\n| &
        | "RecordedHours":{" lv_decimal }",\r\n| &
        | "HoursUnitOfMeasure":"H"\r\n| &
        | \}, "PersonWorkAgreementExternalID":{" ls_person-
personworkagreementexternalid }",\n| &
        | "CompanyCode":{" ls_person-companycode }",\r\n| &
        | "PersonWorkAgreement":{" <fs_key>-%param-
personworkagreement }",\r\n| &
        | "TimeSheetDate":"/Date({ lv_unix_time })/",\r\n| &
        | "TimeSheetIsExecutedInTestRun":{" COND #( WHEN
<fs_key>-%is_draft = if_abap_behv=>mk-on
                                                                THEN |true|
                                                                ELSE
|false| ) }, "TimeSheetOperation":"C"\}\r\n|
        ) TO lt_batch_data.
    ENDLOOP.

    IF lt_batch_data IS NOT INITIAL.
        zjtsc_cl_request=>send_batch(
            EXPORTING
                iv_api          = zjtsc_if_constants=>c_destination-time_sheet_api
                it_batch_data   = lt_batch_data
            IMPORTING
                ev_response_satus = DATA(lv_status)
                ev_response_text  = DATA(lv_response_text)
        ).

        IF lv_response_text CS '201 Created' OR lv_response_text CS '204 No Content'.

            IF lt_project[ 1 ]-%is_draft = if_abap_behv=>mk-on.

                MODIFY ENTITIES OF zjtsc_i_worktpt IN LOCAL MODE
                ENTITY work
                CREATE BY \_role
                SET FIELDS WITH VALUE #( ( %tky = lt_project[ 1 ]-%tky
                    %target = VALUE #( FOR role IN keys
                        ( %cid = |{ role-
wbselementinternalid }{ role-role }{ role-workitem }{ role-%param-personworkagreement }|
                    %is_draft = role-%is_draft
                    wbselementinternalid = VALUE
#( lt_role[ %tky = role-%tky ]-wbselementinternalid OPTIONAL )
                    role = role-role
                    workitem = role-workitem
                    person = role-%param-
personworkagreement
                    workpackage = VALUE #(
lt_role[ %tky = role-%tky ]-workpackage OPTIONAL )
                    workpackageuuid = VALUE #(
lt_role[ %tky = role-%tky ]-workpackageuuid OPTIONAL )
                    deliveryorganization = VALUE
#( lt_personworkagrmt[ personworkagreement = role-%param-personworkagreement ]-companycode
OPTIONAL ) )
                    )
                )
                ENTITY role
                CREATE BY \_timesheet

```

```

        AUTO FILL CID
        SET FIELDS WITH VALUE #( FOR timesheet IN keys
                                ( %cid_ref = |{ timesheet-wbselementinternalid }|
                                timesheet-role }{ timesheet-workitem }{ timesheet-%param-personworkagreement }|
                                %is_draft = timesheet-%is_draft
                                %target = VALUE #( ( %is_draft = timesheet-
%is_draft
                                                                role = timesheet-role
                                                                workitem = timesheet-
workitem
                                                                personworkagreement =
timesheet-%param-personworkagreement
                                                                *
                                                                personnumber = VALUE #(
lt_bussiness_user[ personworkagreement = timesheet-%param-personworkagreement ]-person
OPTIONAL )
                                                                timesheetdate = timesheet-
%param-timesheetdate
                                                                recordedhours = timesheet-
%param-recordedhours )
                                )
                                )
        )
        MAPPED DATA(ls_mapped)
        FAILED DATA(ls_failed)
        REPORTED DATA(ls_reported).
    ENDIF.

ELSE.
    "Error Processing
    zjtsc_cl_request=>parse_response_error(
        EXPORTING
            iv_response = lv_response_text
        IMPORTING
            et_result   = DATA(lt_errors)
    ).

    LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_failed_key>).

        APPEND CORRESPONDING #( <fs_failed_key> ) TO failed-role ASSIGNING FIELD-
SYMBOL(<fs_failed>).
        <fs_failed>-%op-%action-unstaffedposting = if_abap_behv=>mk-on.

        LOOP AT lt_errors ASSIGNING FIELD-SYMBOL(<fs_error>).
            IF NOT ( sy-tabix > 1 AND ( <fs_error>-message EQ 'MESSAGE' OR <fs_error>-
message CS 'exception' ) ).
                APPEND CORRESPONDING #( <fs_failed> ) TO reported-role ASSIGNING FIELD-
SYMBOL(<fs_reported>).
                <fs_reported>-%msg = new_message(
                    id           = |{ <fs_error>-longtext-msgid }|
                    number      = |{ <fs_error>-longtext-msgno }|
                    severity    = if_abap_behv_message=>severity-error
                    v1          = <fs_error>-longtext-message_v1
                    v2          = <fs_error>-longtext-message_v2
                    v3          = <fs_error>-longtext-message_v3
                    v4          = <fs_error>-longtext-message_v4
                ).
            ENDIF.
        ENDLOOP.
    ENDLOOP.
ENDLOOP.

```

```

        ENDIF.
    ENDIF.

*   READ ENTITIES OF zjtsc_i_worktp
*   ENTITY role
*   ALL FIELDS WITH CORRESPONDING #( keys )
*   RESULT DATA(lt_roles).
*
*   result = VALUE #( FOR key IN keys
*                     ( %tky = key-%tky
*                       %param = VALUE #( lt_roles[ %tky = key-%tky ] OPTIONAL ) ) ).

    ENDMETHOD.

ENDCLASS.

CLASS lhc_timesheet DEFINITION INHERITING FROM cl_abap_behavior_handler.
    PRIVATE SECTION.

        METHODS checkuser FOR DETERMINE ON MODIFY
            IMPORTING keys FOR timesheet~checkuser.

    ENDClass.

CLASS lhc_timesheet IMPLEMENTATION.

    METHOD checkuser.

        MODIFY ENTITIES OF zjtsc_i_worktp IN LOCAL MODE
            ENTITY timesheet
            UPDATE FIELDS ( hoursunitofmeasure )
            WITH VALUE #( FOR timesheet IN keys
                ( %tky = timesheet-%tky
                  hoursunitofmeasure = 'H'
                ) )
            REPORTED DATA(ls_reported).

        reported = CORRESPONDING #( DEEP ls_reported ).

    ENDMETHOD.

ENDCLASS.

CLASS lsc_zjtsc_i_worktp DEFINITION INHERITING FROM cl_abap_behavior_saver.
    PROTECTED SECTION.

        METHODS save_modified REDEFINITION.

        METHODS cleanup_finalize REDEFINITION.

    ENDClass.

CLASS lsc_zjtsc_i_worktp IMPLEMENTATION.

    METHOD save_modified.
        DATA: lt_batch_data TYPE zjtsc_cl_request=>tt_batch_data,
              lv_unix_time  TYPE string.
        IF create-timesheet IS NOT INITIAL OR
           update-timesheet IS NOT INITIAL OR
           delete-timesheet IS NOT INITIAL.

```

```

DATA: lt_all_keys  TYPE STANDARD TABLE OF zjtsc_i_hoursrecordedtp,
      lt_temp_keys TYPE STANDARD TABLE OF zjtsc_i_hoursrecordedtp.

lt_all_keys = CORRESPONDING #( create-timesheet ).
lt_temp_keys = CORRESPONDING #( update-timesheet ).
APPEND LINES OF lt_temp_keys TO lt_all_keys.

IF delete-timesheet IS NOT INITIAL.
  SELECT * FROM zjtsc_i_hoursrecordedtp
    FOR ALL ENTRIES IN @delete-timesheet
    WHERE timesheetrecord = @delete-timesheet-timesheetrecord
    INTO TABLE @DATA(lt_existing_timesheet).

  APPEND LINES OF lt_existing_timesheet TO lt_all_keys.
ENDIF.

SELECT * FROM zjtsc_i_roletp
  FOR ALL ENTRIES IN @lt_all_keys
  WHERE wbelementinternalid = @lt_all_keys-wbelementinternalid
  AND   workitem = @lt_all_keys-workitem
  AND   person = @lt_all_keys-personworkagreement
  AND   role = @lt_all_keys-role
  INTO TABLE @DATA(lt_roles).

SELECT * FROM zjtsc_i_project
  FOR ALL ENTRIES IN @lt_all_keys
  WHERE projectuuid = @lt_all_keys-projectuuid
  INTO TABLE @DATA(lt_project).

SELECT personworkagreementexternalid, personworkagreement
  FROM i_personworkagrmttoexternalid
  FOR ALL ENTRIES IN @lt_all_keys
  WHERE personworkagreement = @lt_all_keys-personworkagreement
  INTO TABLE @DATA(lt_agreementexternalid).

IF create-timesheet IS NOT INITIAL.

  DATA: ls_role      TYPE zjtsc_i_roletp,
        ls_project   TYPE zjtsc_i_project,
        ls_agreement TYPE i_personworkagrmttoexternalid,
        lv_decimal   TYPE p DECIMALS 2,
        lv_integral  TYPE i.

  LOOP AT create-timesheet ASSIGNING FIELD-SYMBOL(<fs_create_time_sheet>).
    ls_role = VALUE #( lt_roles[ wbelementinternalid = <fs_create_time_sheet>-
wbelementinternalid
                                workitem = <fs_create_time_sheet>-workitem
                                person = <fs_create_time_sheet>-
personworkagreement
                                role = <fs_create_time_sheet>-role
                                ] OPTIONAL ).
    IF ls_role IS INITIAL.
      ls_role = CORRESPONDING #( VALUE #( create-role[ wbelementinternalid =
<fs_create_time_sheet>-wbelementinternalid
                                                    workitem =
<fs_create_time_sheet>-workitem
                                                    person =
<fs_create_time_sheet>-personworkagreement
                                                    role =
<fs_create_time_sheet>-role ] OPTIONAL ) ).

```

```

ENDIF.
ls_project = VALUE #( lt_project[ projectuuid = <fs_create_time_sheet>-
projectuuid ] OPTIONAL ).
ls_agreement = CORRESPONDING #( VALUE #( lt_agreementexternalid[
personworkagreement = <fs_create_time_sheet>-personworkagreement ] OPTIONAL ) ).
lv_integral = trunc( <fs_create_time_sheet>-recordedhours ).
IF lv_integral / <fs_create_time_sheet>-recordedhours <> 1 .
lv_decimal = <fs_create_time_sheet>-recordedhours - lv_integral.
lv_decimal = lv_decimal * 100 / 60.
lv_decimal = lv_integral + lv_decimal.
ELSE.
lv_decimal = <fs_create_time_sheet>-recordedhours.
ENDIF.
lv_unix_time = zjtsc_cl_request=>convert_date_to_unix_time( iv_date =
<fs_create_time_sheet>-timesheetdate ).
APPEND VALUE #( method = zjtsc_cl_request=>methods-post
content_id = <fs_create_time_sheet>-%key
path = zjtsc_if_constants=>c_entity-time_sheet
body = | \{ "TimeSheetDataFields":\{"ControllingArea": "{
ls_project-controllingarea }",\n| &
| "SenderCostCenter": "{ ls_project-
responsiblecostcenter }", "ActivityType": "{ ls_role-role }", "WBSElement": "{ ls_role-
workpackage }",\r\n| &
| "WorkItem": "{ ls_role-workitem }",\r\n| &
| "TimeSheetNote": "{ <fs_create_time_sheet>-
timesheetnote }",\r\n| &
| "RecordedHours": "{ lv_decimal }",\r\n| &
"RecordedQuantity": "{ <fs_create_time_sheet>-
recordedquantity }",\r\n| &
| "HoursUnitOfMeasure": "{ <fs_create_time_sheet>-
hoursunitofmeasure }",\r\n| &
"RejectionReason": "{ <fs_create_time_sheet>-
rejectionreason }",\r\n| &
| "SendingPubSecFunctionalArea": "{ ls_project-
functionalarea }",\r\n| &
| "ReceiverPubSecFuncnlArea": "{ ls_project-
functionalarea }"\r\n| &
| \}, "PersonWorkAgreementExternalID": "{
ls_agreement-personworkagreementexternalid }",\n| &
| "CompanyCode": "{ ls_role-deliveryorganization
}", "PersonWorkAgreement": "{ ls_agreement-personworkagreement }",\r\n| &
| "TimeSheetDate": "/Date({ lv_unix_time
})/", "TimeSheetIsReleasedOnSave": false,\r\n| &
|
"TimeSheetStatus": "30", "TimeSheetIsExecutedInTestRun": false, "TimeSheetOperation": "C"\}\r\n|
) TO lt_batch_data.
ENDLOOP.
ENDIF.

LOOP AT update-timesheet ASSIGNING FIELD-SYMBOL(<fs_update_time_sheet>).

ls_role = VALUE #( lt_roles[ wbselementinternalid = <fs_update_time_sheet>-
wbselementinternalid
workitem = <fs_update_time_sheet>-workitem
person = <fs_update_time_sheet>-personworkagreement
role = <fs_update_time_sheet>-role ] OPTIONAL ).
ls_project = VALUE #( lt_project[ projectuuid = <fs_update_time_sheet>-projectuuid
] OPTIONAL ).

```



```

ls_agreement = CORRESPONDING #( VALUE #( lt_agreementexternalid[
personworkagreement = <fs_update_time_sheet>-personworkagreement ] OPTIONAL ) ).

lv_unix_time = zjtsc_cl_request=>convert_date_to_unix_time( iv_date =
<fs_update_time_sheet>-timesheetdate ).
APPEND VALUE #( method = zjtsc_cl_request=>methods-post
content_id = <fs_update_time_sheet>-%key
path       = zjtsc_if_constants=>c_entity-time_sheet
body       = |\{"TimeSheetDataFields":\n| &
            |       \{"ControllingArea": "{ ls_project-
controllingarea }",\n| &
            |       "SenderCostCenter": "{ ls_project-
responsiblecostcenter }",\n| &
            |       "ActivityType": "{ ls_role-role }",\n| &
            |       "WBSElement": "{ ls_role-workpackage }",\n| &
            |       "WorkItem": "{ ls_role-workitem }",\n| &
            |       "TimeSheetNote": "{ <fs_update_time_sheet>-
timesheetnote }",\n| &
            |       "RecordedHours": "{ <fs_update_time_sheet>-
recordedhours }",\n| &
            |       "RecordedQuantity": "{
<fs_update_time_sheet>-recordedquantity }",\n| &
            |       "HoursUnitOfMeasure": "{
<fs_update_time_sheet>-hoursunitofmeasure }",\n| &
            |       "RejectionReason": "{ <fs_update_time_sheet>-
rejectionreason }",\n| &
            |       "SendingPubSecFunctionalArea": "{ ls_project-
functionalarea }",\n| &
            |       "ReceiverPubSecFuncnlArea": "{ ls_project-
functionalarea }"\n\},| &
            | "PersonWorkAgreementExternalID": "{ ls_agreement-
personworkagreementexternalid }",\n| &
            | "CompanyCode": "{ ls_role-deliveryorganization
}",\n| &
            | "PersonWorkAgreement": "{ <fs_update_time_sheet>-
personworkagreement }",\n| &
            | "TimeSheetDate": "/Date({ lv_unix_time })/",\n| &
            | "TimeSheetRecord": "{ <fs_update_time_sheet>-
timesheetrecord }",\n| &
            | "TimeSheetIsReleasedOnSave": false,\n| &
            | "TimeSheetStatus": "30",\n| &
            | "TimeSheetIsExecutedInTestRun": false,\n| &
            | "TimeSheetOperation": "U"\}\n|
) TO lt_batch_data.

ENDLOOP.

LOOP AT delete-timesheet ASSIGNING FIELD-SYMBOL(<fs_delete_time_sheet>).
  READ TABLE lt_existing_timesheet ASSIGNING FIELD-SYMBOL(<fs_existing_timesheet>)
  WITH KEY timesheetrecord = <fs_delete_time_sheet>-timesheetrecord.
  IF sy-subrc EQ 0.

    ls_role = VALUE #( lt_roles[ wbselementinternalid = <fs_existing_timesheet>-
wbselementinternalid
                                workitem = <fs_existing_timesheet>-workitem
                                person   = <fs_existing_timesheet>-
personworkagreement
                                role     = <fs_existing_timesheet>-role ] OPTIONAL
).

    ls_agreement = CORRESPONDING #( VALUE #( lt_agreementexternalid[
personworkagreement = <fs_existing_timesheet>-personworkagreement ] OPTIONAL ) ).

```

```

        APPEND VALUE #( method = zjtsc_cl_request=>methods-post
                        content_id = <fs_delete_time_sheet>-timesheetrecord
                        path       = zjtsc_if_constants=>c_entity-time_sheet
                        body       = |\{"TimeSheetRecord": "{ <fs_delete_time_sheet>-
timesheetrecord }",\n| &
                                |"PersonWorkAgreementExternalID": "{ ls_agreement-
personworkagreementexternalid }",\n| &
                                |"PersonWorkAgreement": "{ <fs_existing_timesheet>-
personworkagreement }",\n| &
                                |"CompanyCode": "{ ls_role-deliveryorganization
}" ,"TimeSheetOperation": "D"\}|
                        ) TO lt_batch_data.
    ENDIF.
ENDLOOP.

IF lt_batch_data IS NOT INITIAL.
    zjtsc_cl_request=>send_batch(
        EXPORTING
            iv_api          = zjtsc_if_constants=>c_destination-time_sheet_api
            it_batch_data   = lt_batch_data
        IMPORTING
            ev_response_satus = DATA(lv_status)
            ev_response_text  = DATA(lv_response_text)
    ).

    IF lv_response_text CS '201 Created' OR lv_response_text CS '204 No Content'.
        "Will be saved
    ELSE.
        "Error Processing
        zjtsc_cl_request=>parse_response_error(
            EXPORTING
                iv_response = lv_response_text
            IMPORTING
                et_result   = DATA(lt_errors)
        ).

        reported-timesheet = CORRESPONDING #( lt_all_keys ).
        LOOP AT reported-timesheet ASSIGNING FIELD-SYMBOL(<fs_reported>).
            LOOP AT lt_errors ASSIGNING FIELD-SYMBOL(<fs_error>).
                IF NOT ( sy-tabix > 1 AND ( <fs_error>-message EQ 'MESSAGE' OR <fs_error>-
message CS 'exception' ) ).
                    <fs_reported>-%msg = new_message(
                        id          = |\{ <fs_error>-longtext-msgid }|
                        number      = |\{ <fs_error>-longtext-msgno }|
                        severity    = if_abap_behv_message=>severity-error
                        v1          = <fs_error>-longtext-message_v1
                        v2          = <fs_error>-longtext-message_v2
                        v3          = <fs_error>-longtext-message_v3
                        v4          = <fs_error>-longtext-message_v4
                    ).
                ENDIF.
            ENDLOOP.
        ENDLOOP.
    ENDIF.
ENDIF.

ENDIF.

IF update-serviceconfirmationitem IS NOT INITIAL.

```

```

IF 1 = 1. "new logic
  CLEAR: lt_batch_data.
  LOOP AT update-serviceconfirmationitem ASSIGNING FIELD-SYMBOL(<fs_confirmation>)
WHERE %control-servicedocumentitemdescription = if_abap_behv=>mk-on

OR %control-servicedocumentitemquantity = if_abap_behv=>mk-on

OR %control-actualserviceduration = if_abap_behv=>mk-on

OR %control-servicedocumentitemstatus = if_abap_behv=>mk-on

OR %control-serviceemployee = if_abap_behv=>mk-on

OR %control-actualservicestartdatetime = if_abap_behv=>mk-on.
  IF <fs_confirmation>-%control-actualservicestartdatetime = if_abap_behv=>mk-on.
    DATA(lv_unixtime) = zjtsc_cl_request=>convert_date_to_unix_time( iv_timestamp
= <fs_confirmation>-actualservicestartdatetime ).
    ENDIF.

    APPEND VALUE #( method = zjtsc_cl_request=>methods-patch
                    content_id = <fs_confirmation>-%key
                    path       = zjtsc_if_constants=>c_entity-
service_confirmation_item &&
                    |(ServiceConfirmation='{ <fs_confirmation>-
serviceconfirmation }',ServiceConfirmationItem='{ <fs_confirmation>-
serviceconfirmationitem }')|
                    body       = |\{ "d":\{ \n| &&
| "ServiceConfItemDescription": "{
<fs_confirmation>-servicedocumentitemdescription }",\n| &&
| "Quantity": "{ <fs_confirmation>-
servicedocumentitemquantity }",\n| &&
| "ActualServiceDuration": "{ <fs_confirmation>-
actualserviceduration }",\n| &&
| "ServiceConfItemIsCompleted": "{ COND #( WHEN
<fs_confirmation>-servicedocumentitemstatus = 'C'
THEN
abap_true
ELSE
abap_false ) }",\n| &&
| "ExecutingServiceEmployee": "{
<fs_confirmation>-serviceemployee }"\n| &&
| { COND #( WHEN <fs_confirmation>-%control-
actualservicestartdatetime = if_abap_behv=>mk-on
THEN |,"ActualServiceStartDateTime":
"/Date({ lv_unixtime })/" \n|
ELSE || ) }| && |\}\}|
                    ) TO lt_batch_data.
  ENDLOOP.

  DATA(lt_updated_confirmation) = update-serviceconfirmationitem.
*   DELETE lt_updated_confirmation WHERE NOT ( %control-isfinal = if_abap_behv=>mk-on
*   OR %control-servicedocumentitemisrejected
= if_abap_behv=>mk-on ).

  IF lt_updated_confirmation IS NOT INITIAL.
    SORT lt_updated_confirmation BY serviceconfirmation.
    DELETE ADJACENT DUPLICATES FROM lt_updated_confirmation COMPARING
serviceconfirmation.

    LOOP AT lt_updated_confirmation ASSIGNING FIELD-SYMBOL(<fs_confirm_header>).

```

```

        APPEND VALUE #( method = zjtsc_cl_request=>methods-patch
                        content_id = <fs_confirm_header>-%key-serviceconfirmation
                        path       = zjtsc_if_constants=>c_entity-service_confirmation
&&
                        body       = |({' <fs_confirm_header>-serviceconfirmation }')|
&&
                        |"ServiceConfirmationIsCanceled": { COND #( WHEN
<fs_confirm_header>-servicedocumentitemisrejected = abap_true
                                                                THEN
|true|
                                                                ELSE
|false| ) },\n| &&
                        |"ServiceConfirmationIsFinal": "{ COND #( WHEN
<fs_confirm_header>-isfinal = abap_true
                                                                THEN 'Y'
                                                                ELSE 'N'
) }"| &&
                        |}\}\}
                        ) TO lt_batch_data.
    ENDLOOP.
ENDIF.

IF lt_batch_data IS NOT INITIAL.
    zjtsc_cl_request=>send_batch(
        EXPORTING
            iv_api           = zjtsc_if_constants=>c_destination-
service_confirmation_api
            it_batch_data   = lt_batch_data
        IMPORTING
            ev_response_satus = DATA(lv_status_confirm)
            ev_response_text  = DATA(lv_response_text_confirm)
    ).

    IF lv_response_text_confirm CS '201 Created' OR lv_response_text_confirm CS '204
No Content'.
        "Will be saved
    ELSE.
        "Error Processing
        zjtsc_cl_request=>parse_response_error(
            EXPORTING
                iv_response = lv_response_text_confirm
            IMPORTING
                et_result   = DATA(lt_errors_confirm)
        ).
        READ ENTITIES OF zjtsc_i_worktp IN LOCAL MODE
        ENTITY serviceconfirmationitem
        ALL FIELDS WITH CORRESPONDING #( update-serviceconfirmationitem )
        RESULT DATA(lt_confirm).

        reported-serviceconfirmationitem = CORRESPONDING #( lt_confirm ).
        LOOP AT reported-serviceconfirmationitem ASSIGNING FIELD-
SYMBOL(<fs_reported_confirm>).
            LOOP AT lt_errors_confirm ASSIGNING FIELD-SYMBOL(<fs_error_confirm>).
                IF NOT ( sy-tabix > 1 AND ( <fs_error>-message EQ 'MESSAGE' OR <fs_error>-
message CS 'exception' ) ).
                    <fs_reported_confirm>-%tky-%is_draft = if_abap_behv=>mk-on.
                    <fs_reported_confirm>-%msg = new_message(
                                                                id           = |{ <fs_error_confirm>-
longtext-msgid }|
                                                                number        = |{ <fs_error_confirm>-
longtext-msgno }|

```

```

if_abap_behv_message=>severity-error
message_v1
message_v2
message_v3
message_v4 ).
        ENDIF.
        ENDLOOP.
        ENDLOOP.
        ENDIF.
        ENDIF.

        ENDIF.
        ENDIF.

        ENDMETHOD.

        METHOD cleanup_finalize.

        ENDMETHOD.

ENDCLASS.

```

```

severity =
v1      = <fs_error_confirm>-longtext-
v2      = <fs_error_confirm>-longtext-
v3      = <fs_error_confirm>-longtext-
v4      = <fs_error_confirm>-longtext-

```