

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота
за освітнім ступенем «бакалавр»
на тему:
«Система управління проєктами на базі платформи Salesforce»

Здобувача вищої освіти IV курсу
групи ПЗ-41
спеціальності 121 «Інженерія програмного
забезпечення»
Коротчук Андрій Вікторович

Науковий керівник:
к.п.н., доц. Петренко С. В.

Рівне 2025

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЕКТАМИ	7
1.1. Бізнес процес	7
1.2. Основи управління проектами.....	10
1.3. Salesforce як платформа для управління проектами та огляд аналогічних систем	16
РОЗДІЛ 2. ФОРМУВАННЯ ВИМОГ І МОДЕЛЮВАННЯ СИСТЕМИ	22
2.1. Функціональні та нефункціональні вимоги	22
2.2. Профіль користувачів	24
2.3. Модель даних.....	26
2.4. Сценарії користувачів.....	29
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ПРОЄКТАМИ	33
3.1. Реалізація моделі даних.....	33
3.2. Реалізація компонент	42
3.3. Перспективи масштабування системи	48
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53

ВСТУП

У сучасну епоху розвитку інформаційних технологій управління проектами в ІТ-секторі набуває дедалі більшого значення. Організація ефективного моніторингу часу, витраченого на виконання завдань, є одним із основних аспектів успішної діяльності компаній, що спеціалізуються на наданні ІТ-послуг. Це особливо важливо для підприємств, які працюють за контрактами і визначають вартість своїх послуг на основі часу, витраченого на проекти.

В межах даної роботи буде здійснено аналіз предметної області, охарактеризовано структуру системи, визначено вимоги до неї, розглянуто ключові сценарії використання, описано спроектовану модель даних системи, реалізовано модель даних та розроблено інформаційну систему та описано можливі шляхи масштабування системи.

Актуальність дослідження зумовлена швидкими змінами в умовах бізнес-середовища, де ІТ-компанії стикаються з необхідністю адаптації до нових викликів і вимог. У сучасному світі, де конкуренція на ринку послуг стає дедалі жорсткішою, ефективне управління ресурсами та процесами є критично важливим для успішної діяльності підприємств.

Зокрема, автоматизація обліку робочого часу, управління відсутностями та аналітики дозволяє не тільки знизити адміністративні витрати, а й покращити якість надання послуг, що, у свою чергу, підвищує задоволеність клієнтів. Інтеграція різних бізнес-процесів у єдину інформаційну систему допомагає зменшити ризик помилок, підвищити прозорість операцій та сприяє більш ефективному прийняттю управлінських рішень.

Таким чином, розробка інформаційної системи на платформі Salesforce, що враховує специфіку управління ІТ-компаніями, є надзвичайно актуальною та необхідною в умовах сучасного бізнесу.

Мета дослідження полягає у розробці інформаційної системи для управління ІТ-проектами та компанією на базі платформи Salesforce. Ця система

має на меті автоматизацію процесів обліку робочого часу та інших внутрішніх активностей, що, в свою чергу, підвищить ефективність управління персоналом і покращить якість взаємодії з клієнтами.

Основними аспектами цієї мети є:

- **оптимізація обліку часу:** інформаційна система надасть можливість фіксувати час, витрачений на виконання проєктних завдань, та автоматично генерувати звіти, які можуть бути перетворені в рахунки для клієнтів. Це дозволить зменшити помилки у фінансових розрахунках і підвищити прозорість для замовників;

- **автоматизація управління відсутностями:** система спростить процес подання заявок на лікарняні, відпустки та інші види відсутності, не пов'язані з проєктною діяльністю, що забезпечить гнучкість у внутрішніх процесах компанії;

- **покращення аналітики та звітності:** система надасть можливість формувати загальні звіти по компанії та моніторити ефективність роботи кожного співробітника, що дозволить керівництву швидко аналізувати продуктивність та приймати обґрунтовані управлінські рішення;

- **інтегрований підхід до управління:** система об'єднає процеси управління проєктами, фінансового обліку та внутрішніх активностей, що дозволить створити єдину платформу для координації роботи всієї компанії та підвищення її конкурентоспроможності на ринку.

Отже, проєкт має на меті підвищити ефективність і гнучкість роботи ІТ-компанії завдяки комплексній інформаційній системі, яка охоплює як проєктні, так і внутрішні бізнес-процеси.

Завдання дослідження:

1. дослідити особливості управління в ІТ-компаніях, визначити основні процеси та виклики, з якими вони стикаються, зробити оцінку методологій управління проєктами разом із аналізом процесів та викликів; проаналізувати обмеження та можливості платформи Salesforce у контексті розробки інформаційної системи управління проєктами;

2. розробити модель даних та сформулювати вимоги до системи, визначивши основні сутності, їх зв'язки, нефункціональні та функціональні вимоги; розробити ключові сценарії використання основними класами користувачів;

3. реалізувати інформаційну систему відповідно до розроблених вимог та поставленого завдання;

4. проаналізувати перспективи масштабування системи та описати можливі шляхи розвитку системи дослідивши архітектурні, інтеграційні та інфраструктурні рішення.

Об'єктом дослідження є сучасні інформаційні системи для управління проектами, ресурсами, зокрема їх вплив на ефективність бізнес-процесів, автоматизацію обліку робочого часу та управління персоналом.

Предмет дослідження є архітектурні, функціональні і технологічні аспекти реалізації системи управління робочим часом та відсутностями на базі платформи Salesforce, включаючи аналіз її обмежень та можливостей, а також її оптимізацію для потреб сучасних компаній.

Методи дослідження

У процесі дослідження було використано комплекс наукових методів, який дав змогу всебічно охопити та вивчити поставлену проблематику, зокрема: абстрактно-логічне моделювання, системний аналіз, критичний аналіз джерел.

Практичне значення дослідження полягає в наступному:

1. **оптимізація управління ресурсами**, розроблена інформаційна система дозволить ІТ-компанії більш ефективно управляти робочим часом і відсутностями співробітників, що сприятиме кращому використанню ресурсів і підвищенню продуктивності;

2. **покращення процесів прийняття рішень**, надання детальної аналітики про витрачений час та ефективність роботи співробітників дасть можливість керівництву своєчасно приймати обґрунтовані рішення щодо управління проектами та персоналом.

3. зменшення помилок у фінансових розрахунках, автоматизація процесів обліку робочого часу та генерування звітів для виставлення рахунків клієнтам дозволить знизити ризик помилок і неточностей у фінансових розрахунках.

4. покращення задоволеності співробітників, система, що дозволяє зручно подавати заявки на відпустки та лікарняні, а також отримувати зворотний зв'язок про продуктивність, сприятиме підвищенню рівня задоволеності працівників.

5. підвищення рівня обслуговування клієнтів, забезпечення прозорості у виконанні проектів та обліку часу підвищить довіру клієнтів до компанії та її послуг, що позитивно вплине на формування довгострокових відносин.

Таким чином, дослідження та реалізація інформаційної системи матимуть значний вплив на підвищення ефективності роботи компанії, покращення управління проектами та забезпечення високого рівня обслуговування клієнтів.

Апробація і впровадження результатів дослідження. Основні практичні та теоретичні набуті результати дослідження доповідалися та обговорювалися на XVIII Всеукраїнській науково-практичній конференції здобувачів вищої освіти та молодих учених «Наука, освіта, суспільство очима молодих» (м. Рівне, 14 травня 2025 року), звітній науковій конференції викладачів, співробітників і здобувачів вищої освіти Рівненського державного гуманітарного університету за 2025 рік (м. Рівне, 15 травня 2025 року).

Структура роботи. Основна частина науково-дослідницької роботи складається з трьох розділів.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЕКТАМИ

1.1. Бізнес процес

Бізнес-процес — це сукупність взаємопов'язаних дій або завдань, які виконуються організацією для досягнення певної мети. В основі бізнес-процесів лежить логіка створення цінності, яка визначає, як компанія використовує ресурси для задоволення потреб клієнтів та досягнення конкурентних переваг.

У сучасних умовах глобалізації бізнесу та швидкого розвитку технологій управління бізнес-процесами стало важливим елементом ефективної діяльності організацій. Чітко визначені та оптимізовані процеси дозволяють компаніям забезпечити високу якість послуг, скоротити витрати та швидше адаптуватися до змін на ринку.

Основними характеристиками бізнес-процесів є: **цілісність** (кожен бізнес-процес має початок і завершення, а також чітко визначену мету), **клієнтоорієнтованість** (процеси організовані навколо створення цінності для внутрішнього чи зовнішнього клієнта), **повторюваність** (процеси виконуються багаторазово та повинні мати стандартизовану послідовність дій), **вимірюваність** (кожен процес може бути оцінений за допомогою ключових показників ефективності (KPI)).

Процеси поділяють на:

1. **основні** (core processes), вони напряму пов'язані зі створенням продукту або послуги, наприклад, розробка програмного забезпечення чи надання технічної підтримки;

2. **допоміжні** (supporting processes), ці процеси забезпечують підтримку основної діяльності, такі як HR, бухгалтерський облік або IT-обслуговування;

3. **управлінські** (management processes) процеси спрямовані на планування, контроль та оптимізацію основних і допоміжних процесів.

Для ІТ-компаній, зокрема, бізнес-процеси мають специфічні особливості, такі як висока динаміка змін, необхідність інтеграції різноманітних інструментів та платформ, а також значна залежність від кваліфікації працівників. Ефективне управління бізнес-процесами у такому середовищі дозволяє автоматизувати рутинні завдання, підвищити продуктивність команди та забезпечити високу якість кінцевого продукту.

Етапи бізнес процесу продуктів

Розглянемо бізнес-процес продукту на прикладі Agile методології. Бізнес-процес містить 4 етапи:

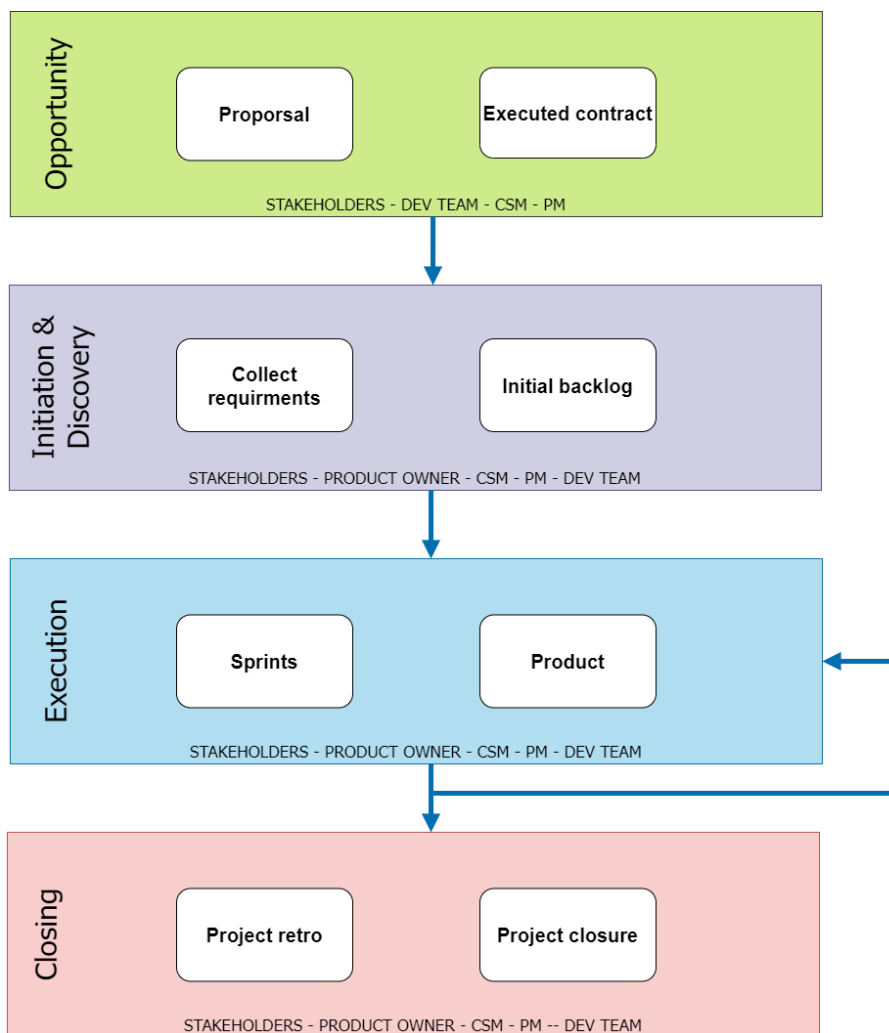


Рисунок. 1.1. Бізнес процес

4. Можливість (Opportunity)

На цьому етапі компанія, або команда отримує певну потенційну бізнес-можливість. Ключовими діями є:

- **пропозиція (Proporsal).** Команда розробляє пропозицію для клієнта або замовника , де викладає план проекту, очікувані результати, бюджет і терміни;

- **укладений контракт (Executed contract).** Після узгодження умов пропозиції підписується контракт на виконання робіт;

Учасниками цього етапу є: зацікавлені сторони (Stakeholders), команда розробників (Dev team), менеджер із успіху клієнта (Customer Success Manager), проектний менеджер (PM)

2. Ініціація та відкриття (Initiation & Discovery)

На цьому етапі розпочинається збір вимог (Collect requirments). Команда спілкується із зацікавленими сторонами для збору повної інформації про вимоги до продукту чи проекту. Далі слідує початковий беклог (Initial backlog). На основі зібраних вимог формується первинний беклог – список завдань і функцій, які мають бути реалізовані.

Учасниками цього етапу є зацікавлені сторони, замовник продукту (Product owner), команда розробників, менеджер із успіху клієнта, проектний менеджер.

3. Виконання (Execution)

Це головний етап цього циклу, коли ведеться активна робота над продуктом.

Проект виконується в коротких циклах(спринтах), під час яких команда виконує конкретні завдання з беклогу, або у відповідності до інших методологій управління.

Продукт (Product). Результатом кожного спринту є інкремент продукту. Після кількох або кількох десятків циклів виконання команда випускає завершений продукт.

В цьому етапі учасниками є зацікавлені сторони, замовник продукту, команда розробників, менеджер із успіху клієнта, проектний менеджер.

4. Завершення (Closing)

Після завершення розробки проєкт переходить на заключний етап закриття.

Ретроспектива проєкту (Project retro). Команда оцінює результати роботи, аналізує помилки та знаходить шляхи для покращення роботи та процесів у майбутньому.

Закриття проєкту (Project closure). Проєкт офіційно завершується, підводяться підсумки, та фіксуються досягнуті цілі

Учасники – є зацікавлені сторони, замовник продукту, команда розробників, менеджер із успіху клієнта, проєктний менеджер.

1.2. Основи управління проєктами

Управління проєктами є дисципліною, яка охоплює планування, організацію, мотивацію та контроль ресурсів для досягнення конкретних цілей проєкту. У бізнес-середовищі управління проєктами сприяє реалізації стратегічних ініціатив, що дозволяє компаніям швидше адаптуватися до змінних умов ринку. Управління проєктами відіграє ключову роль у сучасних організаціях, які часто стикаються з викликами складності, часових обмежень та нестабільності ресурсів.

Методології управління проєктами

Одним з ключових аспектів управління проєктами є вибір методології. Серед найпопулярніших підходів виділяють:

Водоспадний метод (Waterfall): Ця модель розробки була запозичена з системної інженерії, яка активно використовується у виробництві та будівництві – галузях, де внесення змін на пізніх етапах є надзвичайно дорогим або взагалі неможливим. Наприклад, для створення складних інженерних споруд, таких як будівлі, літаки, мости тощо, зміни у проєкті фундаменту після встановлення даху обійдуться надзвичайно дорого, тому необхідно досягти максимальної точності ще на етапі початкового проєктування. Інженери, які прийшли в галузь розробки програмного забезпечення з інших сфер, адаптували знайому їм модель, адже на ранніх етапах розвитку комп'ютерної техніки методології, розроблені спеціально для програмування, практично не існували. Однак подібні підходи і досі застосовуються для програмного забезпечення, коли вимоги є фіксованими, а

якість і надійність особливо важливі – наприклад, у військових або медичних системах. Вперше водоспадна модель була формально описана В. В. Ройсом у 1970 році, після чого набула широкого поширення. Основна особливість водоспадної моделі – це послідовний перехід від однієї стадії до іншої тільки після повного завершення попередньої стадії, без передбаченої можливості повернення до неї. Кожен етап завершується отриманням результатів, які стають вихідними даними для наступного етапу, і створенням повного комплексу документації. Вимоги до програмного забезпечення, визначені на стадії аналізу вимог, документуються як технічне завдання, яке залишається незмінним на весь час розробки. Якість розробки за цією моделлю визначається точністю виконання специфікацій, закладених у технічному завданні [10, 12].

Переваги:

- ніяких, або майже ніяких переробок;
- гарна специфікація здебільшого перетікає в гарну документацію;
- зрозуміла модель;
- програмісти можуть мати низьку кваліфікацію.

Недоліки:

- необхідно досягати досконалості на кожному етапі;
- може бути складно вносити зміни (якщо взагалі можливо);
- надлишкове проєктування;
- поділ розробників на «perfect» та «code monkeys».

Гнучкі методології (Agile): Гнучкі методології розробки програмного забезпечення ґрунтуються на ітеративному підході, при якому вимоги та рішення постійно вдосконалюються завдяки співпраці між багатофункціональними командами, що здатні до самоорганізації. Гнучка розробка спрямована на підвищення ефективності роботи розробників.

Основна мета більшості гнучких методологій полягає в мінімізації ризиків шляхом розбиття процесу розробки на короткі цикли, або ітерації, що зазвичай тривають від одного до двох тижнів. Кожна ітерація є ніби мініатюрним проєктом,

що охоплює всі необхідні етапи для отримання мінімального приросту функціональності: планування, аналіз вимог, проєктування, написання коду, тестування і документування. Хоча окрема ітерація зазвичай не є завершеною версією продукту, ідея полягає в тому, що наприкінці кожної ітерації продукт може бути готовим до випуску. Після завершення ітерації команда переглядає та переоцінює пріоритети розробки [8].

Agile акцентує увагу на прямому особистому спілкуванні. Більшість agile-

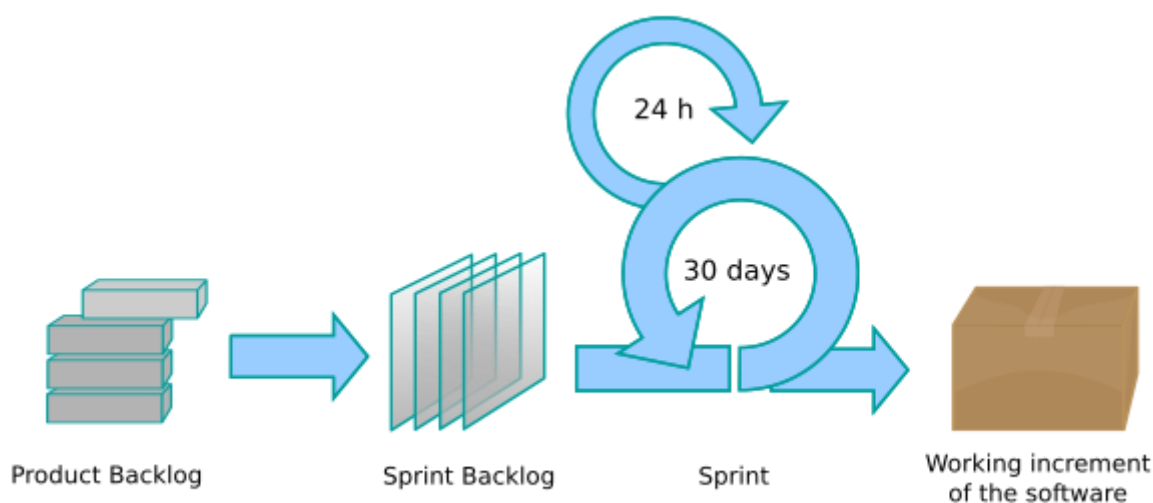


Рисунок. 1.2. Гнучка методологія Agile

команд працюють разом в одному офісі, іноді званому *bullpen*, що забезпечує швидку комунікацію між усіма учасниками процесу, зокрема й «замовниками» (представниками, які формують вимоги до продукту — це можуть бути менеджери продукту, бізнес-аналітики або клієнти). В офісі також можуть бути присутніми тестувальники, дизайнери інтерфейсів, технічні автори і менеджери.

Головним показником ефективності agile є робочий продукт. Завдяки фокусу на особистому спілкуванні обсяг письмової документації зменшується порівняно з іншими методами, що викликає критику щодо відсутності належної дисципліни в таких методах [13, 20].

Переваги:

- гнучкість до змін;
- прозорість та залучення замовника;

- швидке реагування на проблеми;
- релізи та оновлення продукту;
- покращення командної роботи.

Недоліки:

- нестабільні вимоги;
- залежність від залученості клієнта;
- труднощі у прогнозуванні;
- менше документації;
- вимога висококваліфікованої команди.

Scrum — це каркасний процес, що включає набір методів і фіксовані ролі. Основні учасники — ScrumMaster, відповідальний за процеси, який керує і підтримує команду в проєкті, Власник Продукту, який представляє інтереси кінцевих користувачів і зацікавлених сторін, та Команда розробників.

Кожен спринт триває від 15 до 30 днів (тривалість визначається командою), і протягом цього періоду створюється новий функціональний приріст програмного забезпечення.

Перелік функцій, які мають бути виконані в кожному спринті, формується з продуктового беклогу (product backlog) — списку вимог та запитів, розташованих у порядку пріоритетності. На нараді з планування спринту (sprint planning meeting) визначаються завдання (backlog items) для спринту. Під час цієї наради Власник Продукту описує завдання, які він хоче бачити завершеними. Команда розглядає цей перелік і визначає, що реально завершити за період спринту. Після цього всі завдання, відібрані для спринту, фіксуються, що означає заморожування вимог до кінця спринту — змінювати список завдань у цей період заборонено [17, 18].

Переваги:

- гнучкість і адаптивність;
- покращене управління ризиками;
- чітка структура;
- постійна комунікація;

- залучення замовника.

Недоліки:

- вимога до участі замовника;
- складність у масштабуваннях;
- залежність від командної динаміки;
- потенційна поверхнева документація;
- високі вимоги до досвіду команди.

Kanban: це метод управління розробкою програмного забезпечення, який акцентує увагу на доставці продукції вчасно та уникненні перевантаження команди. Цей підхід дозволяє наочно демонструвати процес, від опису задачі до її виконання та доставки результатів користувачу, що дає змогу членам команди самостійно витягувати роботу з черги.

У контексті розробки програмного забезпечення Канбан являє собою систему візуального управління, яка визначає, що виробляти, коли і в яких обсягах. Основою Канбан є виробнича система Тойота та концепція Ощадливого виробництва, що сприяє ефективному використанню ресурсів і зменшенню витрат.

Канбан — це метод управління розробкою програмного забезпечення, який акцентує увагу на вчасній доставці продуктів та уникненні перевантаження команди. Назва «канбан» в японській мові означає «вивіска» або «білборд», і метод був описаний Девідом Андерсеном як спосіб інкрементальних і еволюційних змін в організації [1].

Основні принципи Канбан:

1. почніть з того, що маєте. Необхідно використовувати наявні ролі та процеси, заохочуючи поступові зміни;
2. домагайтеся інкрементних змін. Зміни мають бути малими та еволюційними, оскільки глобальні зміни часто супроводжуються опором;
3. поважайте поточний процес. Визнавайте елементи, що працюють, щоб отримати підтримку з боку команди;

4. лідерство на всіх рівнях. Заохочуйте лідерські дії серед усіх членів організації.

Шість ключових практик Канбан [14]:

1. візуалізуйте. Використовуйте дошки з колонками та картками для відображення процесів роботи.

2. обмежуйте задачі в процесі виконання. Використовуйте систему «витягування» для контролю робочого навантаження.

3. керуйте потоком, моніторьте переходи між станами в процесі для покращення управління.

4. зробіть політики явними. Ясне розуміння процесів дозволяє об'єктивно обговорювати їх вдосконалення.

5. створіть цикли зворотного зв'язку. Наявність другого рівня зворотного зв'язку допомагає вдосконалювати процеси.

6. вдосконалюйте співпрацю. Проводьте експерименти та розвивайте спільне розуміння проблем для досягнення консенсусу.

Метод Канбан сприяє поступовим змінам і використовує науковий підхід для впровадження нововведень [8].

Переваги:

- гнучкість;
- візуалізація процесів;
- управління навантаженнями;
- покращена співпраця;
- легкість у впровадженні.

Недоліки:

- відсутність чіткої структури і ролей;
- ризик застою;
- вимога до самоорганізації;
- складність у масштабуванні;
- обмежена документація.

1.3. Salesforce як платформа для управління проектами та огляд аналогічних систем

Salesforce є однією з найпопулярніших платформ для управління взаємовідносинами з клієнтами (CRM), але вона також має широкий спектр можливостей для управління проектами. Завдяки своїй гнучкості, масштабованості та потужному набору інструментів, Salesforce стає ефективним рішенням для управління проектами в різних галузях. У цьому розділі ми розглянемо можливості Salesforce як платформи для управління проектами, а також інструменти, які дозволяють кастомізувати її під специфічні потреби користувачів.

Основні можливості та кастомізація Salesforce для управління проектами

Salesforce пропонує різноманітні можливості для управління проектами, включаючи [6, 15, 16, 19]:

1. управління завданнями та проектами — Salesforce дозволяє користувачам створювати проекти, завдання, підзадачі та встановлювати терміни виконання. Користувачі можуть легко відстежувати прогрес виконання, призначати завдання на конкретних членів команди та отримувати нагадування про терміни.

2. Інструменти для спільної роботи в Salesforce, такі як Chatter, дозволяють команді обмінюватися повідомленнями, документами та оновленнями в реальному часі. Це сприяє швидшому вирішенню питань і покращує комунікацію між членами команди.

3. Має потужні можливості для створення звітів і аналітики, що дозволяє командам відстежувати продуктивність проектів, аналізувати дані та приймати обґрунтовані рішення на основі отриманої інформації. Користувачі можуть створювати користувацькі звіти для відображення ключових показників ефективності (KPI) проектів.

4. Інтеграція з іншими системами є сильною стороною Salesforce, платформа легко інтегрується з багатьма сторонніми додатками та системами, такими як Slack,

Google Workspace та Microsoft Teams, що дозволяє командам використовувати вже знайомі інструменти для роботи.

5. Автоматизація процесів, за допомогою Flow і Process Builder користувачі можуть автоматизувати рутинні завдання, такі як надсилання сповіщень, створення завдань або зміна статусу проектів. Це знижує навантаження на команду та підвищує ефективність роботи.

Однією з найсильніших сторін Salesforce є його здатність до кастомізації. Користувачі можуть адаптувати платформу під свої потреби, використовуючи різні інструменти та функції. Основні способи кастомізації Salesforce:

- користувачі можуть створювати власні об'єкти для зберігання інформації, специфічної для їхніх проектів. Наприклад, можна створити об'єкти для управління ризиками, бюджетами чи ресурсами. Додатково, можна додавати користувацькі поля для зберігання специфічних даних;

- є можливість створювати складні робочі процеси, які автоматизують управління проектами. Наприклад, можна налаштувати правила для автоматичного переведення проектів у новий статус при виконанні певних умов або надсилання сповіщень командам про важливі зміни;

- за допомогою Salesforce користувачі можуть отримувати доступ до інформації про проекти з мобільних пристроїв. Це дозволяє командам працювати в будь-якому місці та в будь-який час, із будь якої платформи;

- Salesforce надає можливість створення кастомізованих панелей інструментів, які відображають важливу інформацію про проекти в реальному часі. Користувачі можуть налаштувати вигляд панелей, вибираючи відповідні графіки, таблиці та показники;

- Salesforce AppExchange є ринком для сторонніх додатків, які можуть додавати нові функціональності до платформи. Користувачі можуть знайти різноманітні рішення для управління проектами, які можна інтегрувати з Salesforce, що дозволяє швидко підвищити функціональність платформи, також продати свої рішення іншим компаніям.

Виклики та обмеження

Незважаючи на численні можливості, використання Salesforce для управління проектами також має свої виклики [16, 19]:

- для користувачів, які не мають технічних знань, процес налаштування платформи може бути складним і вимагати значних зусиль;
- хоча Salesforce пропонує широкий спектр функцій, вартість ліцензій та кастомізації може бути значною, особливо для малих бізнесів;
- як хмарна платформа, Salesforce вимагає стабільного Інтернет-з'єднання, що є основною проблемою;
- деякі аспекти кастомізації можуть бути обмежені стандартними функціями Salesforce, що може вимагати додаткових зусиль для реалізації специфічних вимог.

Огляд аналогічних систем

У світі управління проектами існує безліч інструментів, які надають командам можливості для ефективної організації роботи, планування та відстеження прогресу. У цьому розділі ми розглянемо такі популярні системи, як JIRA, Asana та Trello аналізуючи їхні можливості, особливості та порівнюючи їх із Salesforce.

JIRA

Опис: розроблена компанією Atlassian, є однією з найпопулярніших платформ для управління проектами, особливо в сфері програмного забезпечення. Вона спеціалізується на управлінні задачами, відстеженні помилок та Agile-методологіях, Scrum і Kanban.

Можливості:

- JIRA дозволяє створювати проекти, завдання, підзадачі та епікеї (epics). Користувачі можуть налаштовувати робочі процеси під свої потреби;
- інтерактивні панелі (dashboards) дають можливість командам візуалізувати дані про виконання проекту та прогрес у реальному часі;
- JIRA легко інтегрується з іншими інструментами, такими як Confluence, Bitbucket і Slack, що підвищує її функціональність;

- потужні звітні функції дозволяють аналізувати продуктивність команди та визначати проблеми в робочому процесі.

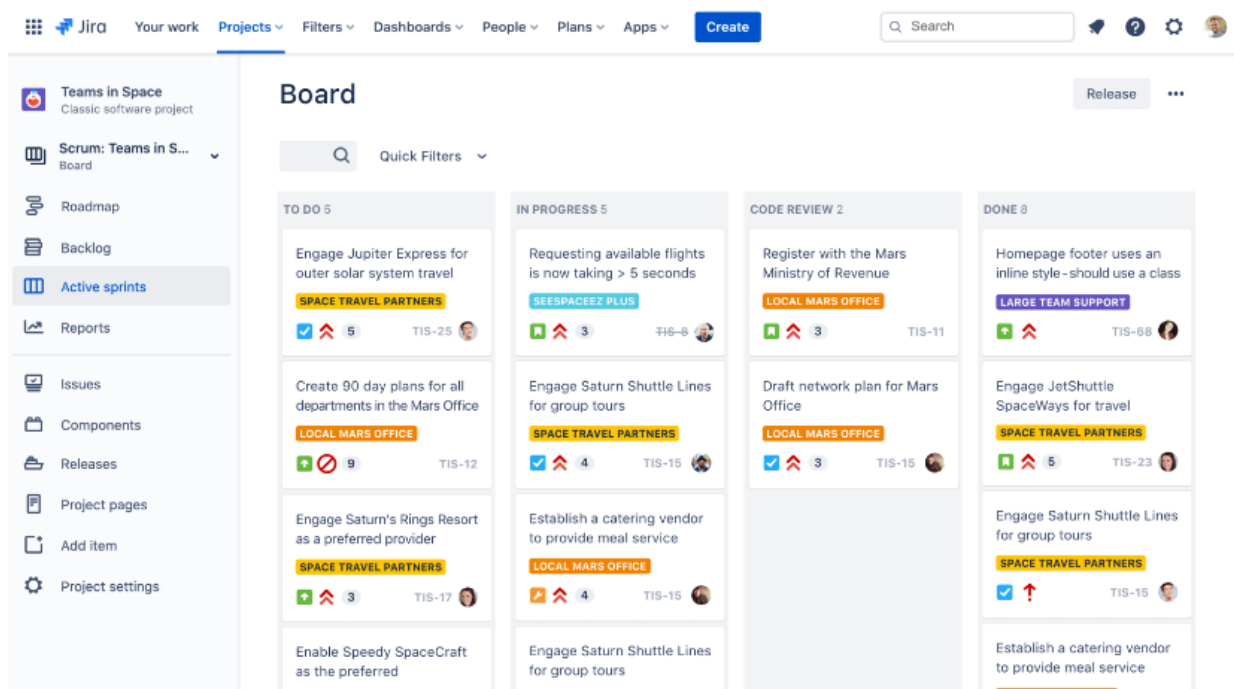


Рисунок. 1.3. Інтерфейс дошки в JIRA

Недоліки:

- новим користувачам може бути важко адаптувати систему до своїх потреб через велику кількість налаштувань та їх складність;
- витрати на використання JIRA можуть бути значними для малих компаній, особливо з великою кількістю користувачів.

Asana

Опис: є ще однією популярною платформою для управління проектами, яка підходить для команд різного розміру. Вона фокусується на організації задач і комунікації в командах.

Можливості:

- Asana має простий і інтуїтивно зрозумілий інтерфейс, що робить її легкою у використанні;
- користувачі можуть переглядати (візуалізовувати) свої завдання в різних форматах, включаючи список, канбан-дошку та графік;

- Asana дозволяє членам команди обмінюватися коментарями, файлами та

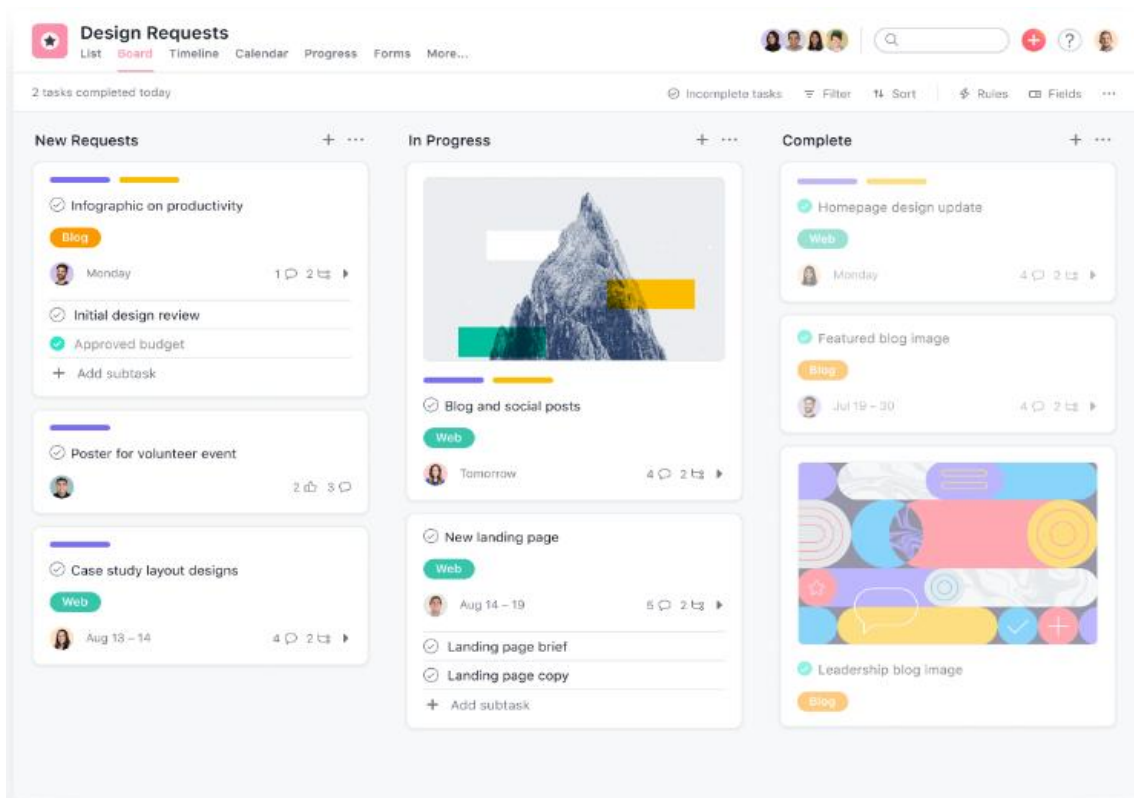


Рисунок. 1.4. Інтерфейс управління завданнями в Asana

оновленнями, що підвищує ефективність співпраці.

Недоліки:

- обмежені функції звітності;
- не підходить для складних проектів. Asana може бути менш ефективною для великих і складних проектів, які вимагають детального планування.

Trello

Опис: це інтуїтивно зрозумілий інструмент для управління проектами на основі методу Kanban. Він дозволяє користувачам організовувати завдання за допомогою візуальних дощок і карток.

Можливості:

- Trello дозволяє створювати дошки для різних проектів і переміщати картки завдань між стовпцями (наприклад, «В роботі», «Завершено»);
- співпраця в реальному часі. Команда може коментувати картки, додавати вкладення та призначати відповідальних за завдання;

- інтеграції. Trello інтегрується з багатьма сторонніми додатками, такими як Slack, Google Drive та Zapier.

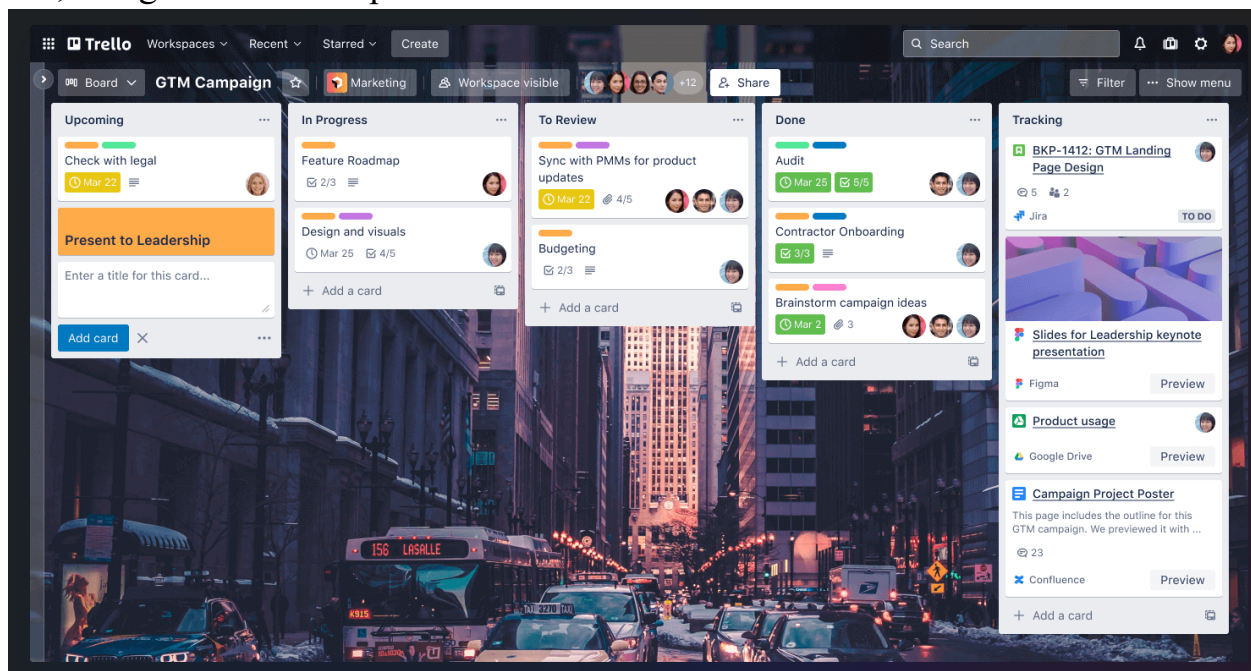


Рисунок. 1.5. Інтерфейс дошки управління завданнями Trello

Недоліки:

- Trello краще підходить для простих проєктів. Вона може бути менш ефективною для управління великими проєктами з багатьма завданнями та залежностями;

- Trello не має потужних аналітичних інструментів для відстеження продуктивності;

РОЗДІЛ 2. ФОРМУВАННЯ ВИМОГ І МОДЕЛЮВАННЯ СИСТЕМИ

2.1. Функціональні та нефункціональні вимоги

Функціональні вимоги

Функціональні вимоги представляють собою систему, або один з її компонентів. Ці вимоги описують необхідний функціонал, який потрібно реалізувати, та функції які продукт повинен містити. Простішими словами за допомогою функціональних вимог ми передбачуємо як система буде працювати [2, 4, 5].

Вимоги сформульовані з урахуванням потреб різних груп користувачів, включаючи керівництво, HR-менеджерів, проєктних менеджерів та співробітників. Основна мета системи полягає в автоматизації процесів обліку робочого часу та управління відсутностями, що дозволить підвищити ефективність роботи компанії в цілому.

Наступна таблиця містить ключові функціональні вимоги до системи, що допоможуть реалізувати заплановані можливості для всіх користувачів.

Таблиця 2.1

Функціональні вимоги

Вимога	Опис
Авторизація та автентифікація користувачів	Користувачі повинні мати можливість входу в систему за допомогою корпоративних облікових записів.
Введення та редагування робочого часу	Система повинна дозволяти користувачам вводити, редагувати та видаляти записи про робочий час.
Запити на відсутність	Співробітники мають можливість подавати заявки на лікарняні, відпустки та інші види відсутностей.
Автоматичне формування інвойсів	Система автоматично генерує інвойси на основі зареєстрованого робочого часу для клієнтів.
Генерація звітів	Користувачі повинні мати можливість генерувати різноманітні звіти, включаючи звіти про використання робочого часу, статус виконання завдань, ефективність команди та аналітику проєктів.

Нефункціональні вимоги

Ці вимоги охоплюють різні аспекти, які впливають на загальну якість системи, її продуктивність, безпеку та зручність використання. Нефункціональні вимоги важливі для забезпечення стабільної роботи системи та задоволення потреб користувачів, а також для підтримки бізнес-процесів компанії [2, 11].

Наступна таблиця містить основні нефункціональні вимоги до системи, що забезпечать її ефективну і надійну роботу в умовах змінюваного навантаження та потреб користувачів.

Таблиця 2.2.

Нефункціональні вимоги

Вимога	Опис
Продуктивність	Система повинна забезпечувати швидкий відгук при великій кількості запитів, зокрема, інтерфейс повинен завантажуватися за менш ніж 2 секунди.
Масштабованість	Система повинна підтримувати можливість розширення функціоналу та кількості користувачів без значного зниження продуктивності.
Надійність	Система має бути високонадійною з мінімальним часом простою, доступність системи повинна становити не менше 99.9%.
Безпека	Дані користувачів і компанії повинні бути захищені через шифрування, належне управління правами доступу та багаторівневу автентифікацію.
Модульність	Система повинна бути розроблена за модульним підходом, щоб забезпечити легкість внесення змін та додавання нових функцій.
Підтримуваність	Код повинен бути добре документований, щоб забезпечити його легку підтримку, оновлення та розширення іншими розробниками в майбутньому.
Юзабіліті	Інтерфейс повинен бути інтуїтивно зрозумілим і зручним для користувачів різного рівня технічної підготовки.
Сумісність	Система повинна коректно працювати на різних браузерах та пристроях, включаючи мобільні телефони і планшети.

Продовження табл. 2.2

Реакція на помилки	У разі помилок система повинна надавати зрозумілі повідомлення користувачам і не допускати втрати введених даних або зупинки роботи.
Ефективність використання ресурсів	Система повинна оптимально використовувати серверні ресурси, забезпечуючи економію процесорного часу, пам'яті та інших ресурсів.
Гнучкість налаштувань	Система повинна дозволяти гнучко налаштовувати права доступу для різних типів користувачів та модулів, залежно від потреб компанії.
Відновлення після збоїв	У випадку аварійної ситуації система повинна підтримувати механізми резервного копіювання та відновлення даних без втрати важливої інформації.

2.2. Профіль користувачів

Інформаційна система, що моделюється для управління ІТ-компанією, буде використовуватися різними групами користувачів, кожна з яких має свої функціональні потреби та ролі. Основними користувачами системи на яких будується вся взаємодія:

1. співробітники, які використовуватимуть систему для обліку робочого часу, подання заявок на відпустки та лікарняні, а також для фіксації своїх професійних навичок;

2. керівники проєктів, на них буде відповідальність за управління проєктами, контроль виконання завдань, аналіз звітів про витрачений час та оцінювання продуктивності команди;

3. HR-менеджери, вони будуть використовувати систему для управління кадровими даними, обробки заявок на відпустки та лікарняні, а також для моніторингу відсутності співробітників;

4. фінансові спеціалісти. Ці користувачі потребують доступу до даних про витрати часу на проєкти для формування рахунків і звітності, а також для автоматизації фінансових процесів;

5. адміністратори системи — відповідальні за налаштування та підтримку системи, управління доступом користувачів та забезпечення безпеки даних.

Ролі користувачів

1. Проєктний менеджер (Project Manager) відповідає за управління клієнтськими проєктами, планування та моніторинг роботи команди, а також за виконання завдань у встановлені терміни та відповідно до бюджету. У системі його завдання:

- підбір персоналу на проєкти, враховуючи навички співробітників та потреби проєкту;
- облік часу, витраченого командою на проєктні завдання;
- перевірка та затвердження звітів по витраченому часу;
- аналіз продуктивності команди на різних етапах проєкту.

2. Спеціаліст з управління відносинами з клієнтами (Customer Success Manager, CSM) працює безпосередньо з клієнтами для забезпечення їхньої задоволеності послугами та продуктами компанії. Основні завдання CSM у системі:

- моніторинг виконання завдань для клієнтських проєктів;
- доступ до звітів по витраченому часу для надання клієнтам актуальної інформації;
- контроль якості надання послуг та зворотний зв'язок з клієнтами;
- генерація звітів для клієнтів та контроль виставлення рахунків.

3. HR-команда відповідає за управління персоналом, включаючи облік відпусток, лікарняних та інших видів відсутностей. У системі їхні функції охоплюють:

- управління заявками на відсутність (відпустки, лікарняні тощо);
- відстеження ефективності роботи співробітників на основі аналітичних звітів.

4. Розробники та інші працівники проєкту є основними користувачами системи для фіксації та управління своїм робочим часом. Основні завдання для них:

- облік часу, витраченого на проєкти, через інтуїтивно зрозумілий інтерфейс;
- подання заявок на відпустки та лікарняні;
- огляд своїх результатів та ефективності на основі наданих звітів;
- Вибір навичок для участі в нових проєктах.

5. Адміністратор системи відповідає за технічне управління та підтримку платформи Salesforce, зокрема за налаштування прав доступу, забезпечення безперебійної роботи системи та виконання резервного копіювання даних. Основні функції адміністратора включають:

- налаштування прав доступу для різних ролей у системі;
- моніторинг продуктивності системи та забезпечення її стабільної роботи;
- технічна підтримка користувачів і вирішення питань щодо роботи системи.

2.3. Модель даних

Resource — створюється для детального обліку й управління даними про всіх працівників і залучених спеціалістів компанії, а також зовнішніх ресурсів, які можуть брати участь у проєктах компанії. Цей об'єкт необхідний для впорядкування інформації про персонал, їхню кваліфікацію, статус зайнятості, доступність для клієнтських проєктів, і для ефективного управління кадровими ресурсами в цілому.

Region — визначає географічне місце діяльності ресурсу та прив'язує його до певної країни. У системі реалізовано ієрархію регіонів, яка використовується для звітності та аналітики. Найнижчим рівнем у цій ієрархії є країна, що дозволяє точніше прив'язувати ресурси до їхньої локації. Для зручності звітності країни об'єднуються у більші регіони, такі як Східна Європа, Західна Європа, Велика Британія, Північна Америка, Південна Америка та Африка.

Skill — це база даних, яка містить усі навички та сертифікації ресурсів. Це дає можливість одному ресурсу мати кілька навичок, а також одній навичці належати кільком ресурсам.

Resource Skill — використовується для зв'язку об'єктів Resource і Skill у відносинах many-to-many. Обидва поля — Resource та Skill — мають тип зв'язку Master-Detail, що забезпечує автоматичне видалення пов'язаних записів при видаленні основного запису.

Group (Department) призначення — визначати підрозділ або спеціалізацію, до якого належить ресурс. Наприклад, відділи можуть включати такі напрямки, як Salesforce Developers, Salesforce Admins, Фінанси, Vacations (управління відпустками), Лікарняні, Юридичний відділ тощо.

Work Calendar — зберігає загальну інформацію про робочі години, а також офіційними святами. Цей календар можна використовувати як основне джерело даних для планування робочого часу, автоматизації розкладу та розрахунку доступності ресурсів у різних регіонах.

Holiday — зберігає інформацію про офіційні святкові дні в конкретній країні та дає можливість регулювати кількість робочих годин для певної дати.

Project — містить детальну інформацію про проєкт, його статус, тип, плановані дати, замовника та інші важливі атрибути. Завдяки цій структурі можна ефективно управляти проєктом, відстежувати прогрес, ресурсне навантаження, бюджет і часові витрати.

Assignment — зберігає детальну інформацію про участь ресурсу в проєкті, включаючи параметри часу, ролі та умов оплати. Якщо проєкт є оплачуваним, призначення (assignment) має бути підтверджене головою відділу ресурсу, який запрошений на цей проєкт.

Milestone — це об'єкт, що використовується для групування тайм-карт за певними періодами всередині проєкту. Він особливо важливий для великих проєктів, де оплата може відбуватися частинами залежно від завершення певних етапів (щомісячно, щотижня або за кілька місяців). У звітах тайм-карти можна

групувати за полем Milestone, щоб отримати підсумкову кількість відпрацьованих годин і суму, яку має оплатити клієнт за конкретний період.

Timecard — використовується для запису годин, відпрацьованих ресурсами компанії на різних проєктах або завданнях. Він містить деталі, які дають змогу контролювати прогрес роботи та взаємодіяти з іншими об'єктами в Salesforce.

Violation — використовується для зберігання записів про попередження працівників, які не відпрацьовують свою робочу норму. Містить деталі про працівника, зокрема його електронну пошту, на яку прийде попередження.

Account — це стандартний об'єкт, що використовується для зберігання інформації про клієнтів, партнерів, або внутрішні записи компанії. Цей об'єкт містить ключові дані про клієнтів і дозволяє відслідковувати важливі атрибути, пов'язані з бізнес-відносинами.

Схема зв'язків (відношень) виглядає наступним чином:

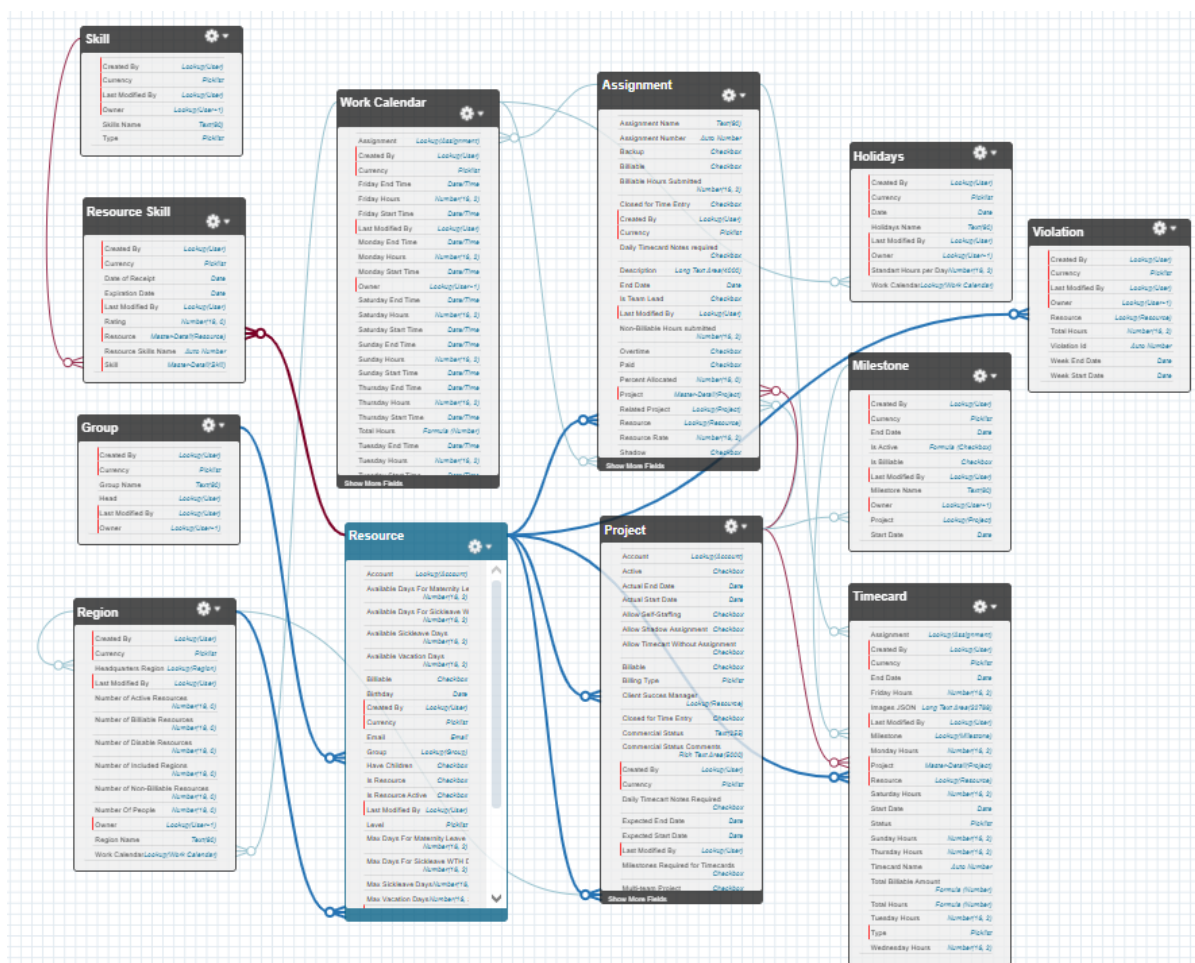


Рисунок. 2.1. Схема відношень

Lookup Relationship — це тип зв'язку між двома об'єктами, який дозволяє створювати зв'язок «один до багатьох». Основна ідея полягає в тому, що один запис може «посилатися» на інший запис, зберігаючи лише посилання, але не впливаючи на життєвий цикл основного запису. На схемі (див. Рисунок. 2.1) позначається синім кольором (світло-блакитним) [16].

Master-Detail Relationship (зв'язок типу «господар-підлеглий») — це тип зв'язку між двома об'єктами, де один об'єкт виступає як головний (master), а інший як підлеглий (detail). Цей зв'язок визначає жорсткішу залежність між об'єктами в порівнянні з Lookup-зв'язком, оскільки підлеглий об'єкт існує тільки в контексті головного. На схемі (див. Рисунок. 2.1.) позначається червоним кольором [16].

Аплікації (див. Рисунок. 2.2.) – це додатковий функціонал для користувачів у системі. Вони дозволяють управляти різними аспектами проєктної діяльності, ресурсами компанії та адміністрування регіонами. Використовуючи дані аплікацій, система Salesforce автоматизує процеси і підтримує актуальність статусів у базі даних, зменшуючи навантаження на адміністративний персонал і підвищуючи ефективність управління.

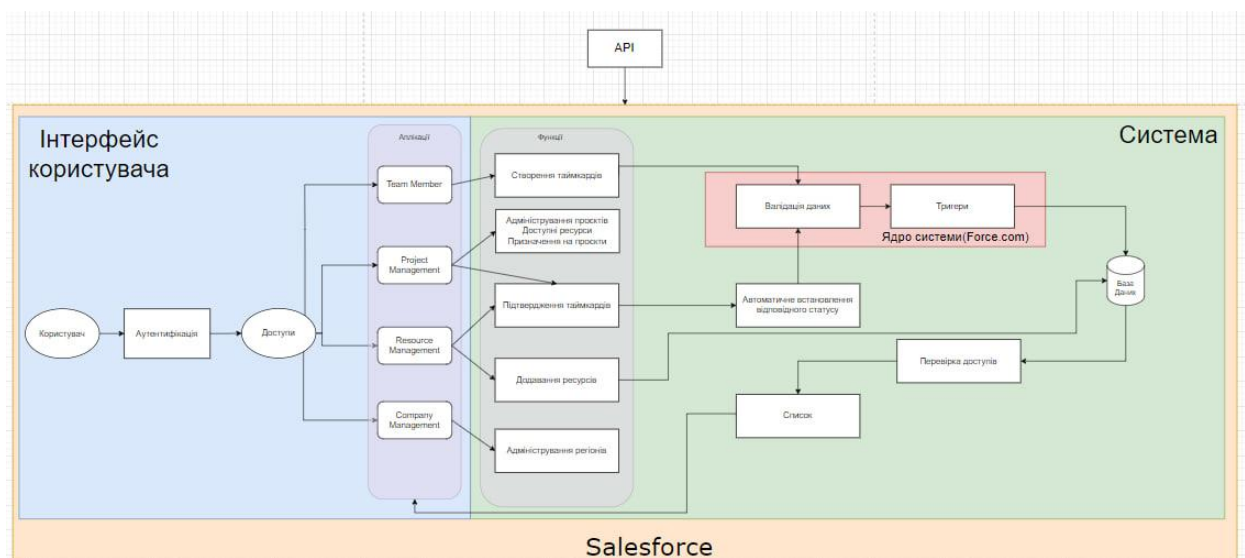


Рисунок. 2.2. Діаграма взаємодій

2.4. Сценарії користувачів

Користувачі, що мають безпосередню взаємодію з системою:

1. Проєктний Менеджер

Проєктний Менеджер відповідає за координацію команди та контроль за виконанням завдань у межах проєктів.

Основні дії:

- перегляд статусу завдань і прогресу проєкту;
- звітування про час, витрачений на виконання завдань;
- генерація інвойсів на основі виконаних робіт;
- огляд аналітики щодо ефективності працівників і проєктів;
- запит на додаткові ресурси або зміни у проєкті;
- відстеження відсутностей у команді та планування ресурсів.

2. CSM (Customer Success Manager)

CSM забезпечує успішність проєктів і взаємодію з клієнтами. Важливою частиною його роботи є моніторинг задоволення клієнтів і забезпечення вчасного завершення проєктів.

Основні дії:

- огляд виконаних робіт і спілкування з клієнтами;
- генерація звітів для клієнтів;
- створення плану робіт, базованого на потребах клієнтів;
- перегляд історії взаємодії з клієнтами та запис комунікацій;
- управління запитами клієнтів на зміни у проєкті.

3. HR Команда

HR відповідає за управління персоналом, облік робочого часу та відсутностей, а також підтримку працівників.

Основні дії:

- управління запитами на відпустки, лікарняні та інші типи відсутностей;
- відстеження робочого часу кожного співробітника;
- огляд навичок співробітників для пошуку відповідних проєктів;
- генерація звітів щодо робочого часу та ефективності команди;

- планування ресурсів на основі прогнозованих відсутностей.

4. Розробники або інші працівники проєкту

Працівники залучені до виконання завдань у межах проєктів. Їхня основна взаємодія з системою стосується обліку робочого часу та відстеження статусу завдань.

Основні дії:

- логування часу, витраченого на завдання;
- перегляд свого розкладу та запланованих завдань;
- запит на відпустки або лікарняні;
- огляд звітів щодо своєї ефективності;
- оновлення свого профілю та навичок.



Рисунок. 2.3. Use cases діаграма

5. Адміністратор системи

Адміністратор відповідає за технічне обслуговування системи, підтримку користувачів та забезпечення її безперебійної роботи.

Основні дії:

- налаштування системи та управління користувацькими ролями;
- адміністрування інтеграцій з іншими системами;
- контроль за безпекою та доступом до даних;
- налаштування кастомних полів та процесів для роботи з проєктами;
- підтримка користувачів та усунення технічних проблем;

Користувачі, що мають опосередкований вплив:

1. Керівництво компанії

Керівництво використовує систему для стратегічного аналізу даних та прийняття рішень. Воно не взаємодіє з системою щодня, але періодично переглядає аналітичні звіти та приймає рішення на їх основі.

Основні дії:

- огляд аналітичних звітів про ефективність проєктів;
- візуалізація ключових показників діяльності компанії;
- оцінка продуктивності працівників і проєктів;
- прийняття рішень щодо розподілу ресурсів та інвестицій у нові проєкти.

2. Клієнти

Клієнти отримують інформацію через звіти та інвойси, які генеруються в системі. Взаємодія безпосередньо з системою обмежена, але вони можуть отримувати доступ до певних даних через інтеграції або звіти, що надсилаються.

Основні дії:

- отримання звітів про виконані завдання;
- огляд інвойсів на основі затверджених робіт;
- обговорення змін у проєкті на основі отриманих звітів.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ПРОЄКТАМИ

3.1. Реалізація моделі даних

Модель даних реалізовано створенням полів в об'єктах, поля в цих об'єктах мають, або відповідають за певну функцію [6, 16, 19]. Опис полів (структура опису: **поле** – **Api_name** – тип поля – опис):

Об'єкт Resource

Account – Account__с – Lookup (Account) – відповідальна організація за ресурс (абстракція компанії). Це поле дасть змогу групувати ресурсів за відповідальними сторонами (компаніями).

Billiable – Billiable__с – Checkbox – визначення оплачуваності ресурсу, значення True встановлюється коли компанія продає ресурса як послугу.

Birthday – Birthday__с – Date – дата народження ресурса.

Currency – CurrencyIsoCode – Picklist – можливість вибору валюти в які ресурс бажає отримувати оплату за роботу.

Email – Email__с – Email – електронна пошта працівника.

Group – Group__с – Lookup (Group) – це відділ компанії.

Have Children – Have_Children__с – Checkbox – наявність дітей у працівника, можливі додаткові вихідні за доглядом хворої дитини.

Is Resource – Is_Resource__с – Checkbox – поле визначає чи може бути задіяний працівник в клієнтських проєктах.

Is Resource Active – Is_Resource_Active__с – Checkbox – стан визначення чи використовується ресурс зараз на проєктах. HR не може бути активним, CSM може, але його години не оплачуються.

Level – Level__с – Picklist – рівень кваліфікації працівника, дає змогу групувати працівників за досвідом, компетентністю що полегшує вибір фахівців відповідного рівня, наприклад: Trainee, Junior, Middle, Senior, Principal.

Primary Expertize – Primary_Expertize__с – Picklist – ключова технологія з якою працює працівник (C#, Java, CI/CD, Docker).

Region – Region__с – Lookup (Region) – регіон в якому живе, або працює ресурс (західна Європа, США, Азія, Африка).

Resource Name – Resource_Name__с – Text (80) – ім'я співробітника або підрядника за яким його можна ідентифікувати в системі.

Roles – Roles__с – Picklist – функціональна роль в команді (Backend, Front-end, DevOps).

Work Start Day – Work_Start_Day__с – Date – дата початку роботи.

Об'єкт Region

Headquarters Region – Headquarter_Region__с – Lookup (Region) – старший регіон (вищий за ієрархією), застосовується для регіонів західна, східна, південна, північна Європа (старшим регіоном буде просто Європа).

Number of Active Resources – Number_of_Active_Resources__с – Number (18, 0) – обрахунок загального числа активних ресурсів в регіоні.

Number of Billable Resources – Number_of_Billable_Resources__с – Number (18, 0) – кількість оплачуваних ресурсів.

Number of Disable Resources – Number_of_Disable_Resources__с – Number (18, 0) – число неактивних ресурсів.

Number of Included Regions – Number_of_Included_Regions__с – Number (18, 0) – загальне число включених регіонів.

Number of Non-Billable Resources – Number_of_Non-Billable_Resources__с – Number (18, 0) – загальне число неоплачуваних ресурсів в регіоні.

Number of People – Number_of_People__с – Number (18, 0) – загальне число людей в регіоні.

Region Name – Name – Text (80) – назва регіону.

Work Calendar – Work_Calendar__с – Lookup (Work Calendar) – робочий календар по якому працюватиме ресурс, він містить прив'язку до офіційних вихідних та свят в регіоні.

Об'єкт Skill

Skill Name – Name – Text (80) – загальна назва навичку (C#, HTML, CSS, Java, тощо.).

Type – Type__с – Picklist – визначає чи навичка підтверджена офіційно сертифікатом (Skill є не підтвердженим, Certification – підтверджено).

Resource Skill

Date of Receipt – Date_of_Receipt__с – Date – початкова дана дійсності сертифікації (дійсне тільки для типу навички сертифікації).

Expiration Date – Expiration_Date__с – Date – кінцева дата дійсності сертифікації (дійсне тільки для типу навички сертифікації).

Rating – Rating__с – Number (18, 0) – абстрактна оцінка рівня володіння від 1 до 5.

Resource – Resource__с – Master-Detail (Resource) – визначає «власника» навички або сертифікації.

Resource Skills Name – Name – Auto Number – унікальних ідентифікатор навички.

Skill – Skill__с – Master-Detail (Skill) – прив'язка до навички.

Об'єкт Group

Group Name – Name – Text (80) – назва групи або відділу компанії(департаменту).

Head – Head__с – Lookup (User) – головна особа по регіону – підтверджує додавання людей до групи (відділ компанії).

Week Day – Week_Day__с – Picklist – день тижня – понеділок, вівторок.

Week Start Day – Week_Start_Day__с – Picklist – початок робочого тижня.

Work Calendar Name – Name – Text (80) – назва календаря.

Workday Start Hour – Workday_Start_Hour__с – Number (16,2) – година початку робочого дня.

Workday End Hour – Workday_End_Hour__с – Number (16,2) – година закінчення роботи.

Об'єкт Holiday

Date – Date__с – Date – дата свята або вихідного.

Holidays Name – Name – Text (80) – назва свята або вихідного.

Standart Hours per Day – Standart_Hours_per_day__с – Number (16, 2) – стандартна кількість годин роботи на день.

Work Calendar – Work_Calendar__с – Lookup (Work Calendar) – прями́язка свят до робочого календаря.

Об’єкт Assignment

Assignment Name – Name – Text (80) – узагальнення назви призначення (посади).

Assignment Number – Assignment_Number__с – Auto Number – унікальний номер призначення або посади.

Backup – Backup__с – Checkbox – статус, який відображає чи є ресурс «запасним», тим кого можна підключити якщо команда не встигає, або не виконує необхідний обсяг робіт.

Billiable – Billiable__с – Checkbox – статус оплачуваності співробітника.

Billiable Hours Submitted – Billiable_Hours_Submitted__с – Number (16, 2) – кількість погоджених оплачуваних робіт.

Closed for Time Entry – Closed_for_Time_Entry__с – Checkbox – заборона на створення або редагування таймкарт.

Daily Timecard Notes required – Daily_Timecard_Notes_required__с – Checkbox – вимагає нотатки на кожен день для таймкарти.

Description – Description__с – Long Text Area (4000) – опис призначення або посади, можливий опис з якою ціллю було призначено ресурс.

End Date – End_Date__с – Date – дата закінчення призначення або роботи на посаді.

Is Team Lead – Is_Team_Lead – Checkbox – визначає чи являється ресурс командним лідером.

Non – Billiable Hours submitted – Non_Billiable_Hours_submitted__с – Number (16, 2) – кількість погоджених неоплачуваних годин.

Overtime – Overtime__c – Checkbox – визначає чи будуть оплачувати години, поза робочим графіком.

Paid – Paid__c – Checkbox – оплачуваність посади (призначення).

Percent Allocated – Percent_Allocated__c – Number (18, 0) – погоджений відсоток який працівник може працювати на інших проєктах, наприклад якщо працівник має робочий день із 8 годин, то при значення 25% працівник зможе працювати на іншому проєкті 2 години.

Project – Project__c – Master-Detail (Project) – на який проєкт призначено.

Related Project – Related_Project__c – Lookup (Project) – споріднений проєкт, у випадку мікросервісної архітектури можуть бути різні або зв'язані сервіси.

Resource – Resource__c – Lookup (Resource) – працівник якого призначають.

Resource Rate – Resource_Rate__c – Number (16, 2) – погоджена оплата за одну годину роботи.

Shadow – Shadow__c – Checkbox – позначає працівників яких залучили у випадках коли терміново потрібно доробити або допомогти з проєктом або продуктом.

Start Date – Start_Date__c – Date – дата початку роботи на проєкті.

Status – Status__c – Picklist – статус погодження на проєкт.

Total Billable Amount – Total_Billable_Amount__c – Formula (Number) – загальна сума до оплати яка сформована на основі кількості погоджених оплачуваних годин та погодженої оплати за одну годину.

Work Calendar – Work_Calendar__c – Lookup (Work Calendar) – календар, за яким буде працювати призначена на посаду особа.

Об'єкт Project

Account – Account__c – Lookup (Account) – організація або компанії.

Active – Active__c – Checkbox – стан активності проєкту.

Actual Start Date – Actual_Start_Date__c – Date – фактична дата початку роботи над проєктом.

Actual End Date – Actual_End_Date__c – Date – фактична дата закінчення роботи над проєктом.

Actual Self-Staffing – Actual_Self-Staffing__c – Checkbox – дозвіл на створення призначень (Assignments) на проєкті.

Allow Shadow Assignment – Allow_Shadow_Assignment__c – Checkbox – дозвіл на створення «тіньових» задач, які не будуть оплачена. Це може бути корисно коли команда не втиснулась в графік , і проєкт доопрацьовують після запланованої дати запуску проєкту.

Allow Timecard Without Assignment – Allow_Timecard_Without_Assignment__c – Checkbox – дозвіл на створення таймкарт без призначень, призначено для усіх внутрішніх активностей.

Biliable – Biliable__c – Checkbox – оплачуваність проєкту.

Billing Type – Billing_Type__c – Picklist – тип оплачуваності: фіксований, погодинна оплата, оплата базована не періодах (Milestones).

Client Success Manager – Client_Succes_Manager__c – Lookup (Resource) – менеджер з успіху клієнта, як правило людина із сторони клієнта.

Closed for Time Entry – Closed_for_Time_Entry__c – Checkbox – забороняє створювати таймкарти.

Commercial Status – Commercial_Status__c – Text (255) – коротка інформація про проєкт.

Commercial Status Comments – Comemrcial_Status_Comments__c – Text (5000) – детальний опис проєкту, інформація про те як проходить розробка проєкту.

Daily Timecard Notes Required – Daily_Timecard_Notes_Required – Checkbox – вимагає нотатки в таймкартах.

Expected Start Date – Expected_Start_Date__c – Date – запланована дата початку роботи над проєктом.

Expected End Date – Expected_End_Date__c – Date – запланована дата закінчення роботи над проєктом.

Milestones Required for Timecards – Milestone_Required_for_Timecards – Checkbox – вимагає створення часового періоду для таймкарти.

Multi-team Project – Multi-team_Project__с – Checkbox – статус, який визначає чи цей проєкт розробляється декількома командами.

Planned Billable Hours – Planned_Billable_Hours__с – Number (16, 2) – запланована кількість оплачуваних годин.

Planned Budget – Planned_Budget__с – Number (16, 2) – запланований бюджет на проєкт.

Project Manager – Project_Manager__с – Lookup (Resource) – Проєктний Менеджер.

Project Name – Project_Name__с – Text (80) – назва проєкту.

Project Type – Project_Type__с – Picklist – тип проєкту (внутрішній проєкт, клієнтський проєкт, відпустки).

Region – Region__с – Lookup (Region) – регіон.

Risk Status – Risk_Status__с – Picklist – статус ризику проєкту.

Об'єкт Milestone

End Date – End_Date__с – Date – дата закінчення періоду.

Is Active – Is_Active__с – Checkbox – визначає стан активності періоду.

Is Billable – Is_Billable__с – Checkbox – визначає стан оплачуваності періоду.

Milestone Name – Name – Text (80) – назва періоду.

Project – Project__с – Lookup (Project) – прив'язка періоду до проєкту.

Start Date – Start_Date__с – Date – дата початку періоду.

Об'єкт Timecard

Assignment – Assignment__с – Lookup (Assignment) – призначення таймкарди на працівника.

End Date – End_Date__с – Date – дата закінчення (у випадку з відпусткою, або лікарняним) або період до якого потрібно виконати завдання.

Friday Hours – Friday_Hours__c – Number (16, 2) – кількість відпрацьованих годин у п'ятницю.

PDF JSON – PDF_JSON__c – Long Text Area (32768) – поле зберігає артефакт на лікарняний.

Milestone – Milestone__c – Lookup (Milestone) – умовна одиниця періоду.

Monday Hours – Monday_Hours__c – Number (16, 2) – кількість відпрацьованих годин у понеділок.

Project – Project__c – Master-Detail (Project) – проєкт.

Resource – Resource__c – Lookup (Resource) – ресурс, який буде робити активність.

Saturday Hours – Saturday_Hours__c – Number (16, 2) – кількість відпрацьованих годин у суботу.

Start Date – Start_Date__c – Date - дата початку (у випадку з відпусткою, або лікарняним) або період з якого починається виконання завдання.

Status – Status__c – Picklist – статус таймкарти: New, Submitted, Approved, Rejected.

Thursday Hours – Thursday_Hours__c – Number (16, 2) – кількість відпрацьованих годин у четвер.

Timecard Name – Name – Text (80) – назва таймкарти.

Total Billable Amount – Total_Billable_Amount__c – Formula (Number) – загальна сума до оплати.

Total Hours – Total_Hours__c – Formula (Number) – загальна кількість годин витрачених на роботу або активність.

Tuesday Hours – Tuesday_Hours__c – Number (16, 2) – кількість відпрацьованих годин у вівторок.

Type – Type__c – Picklist – тип таймкарти, наприклад Sickleave, Vacation, Education, Unpadin Leave.

Wednesday Hours – Wednesday_Hours__c – Number (16, 2) – кількість відпрацьованих годин у середу.

Об'єкт Violation

Resource – Resource__c – Lookup (Resource) – якому працівнику прийде попередження.

Total Hours – Total_Hours__c – Number (16, 2) – загальна кількість годин відпрацьованих за тиждень.

Violation Id – Name – Auto Number – унікальний ідентифікатор попередження.

Week End Date – Week_End_Date__c – Date – кінець робочого тижня.

Week Start Date – Week_Start_Date__c – Date – початок робочого тижня.

Об'єкт Account

Name – Name – Text (80) – назва компанії або організації.

Email – Email__c – Email – електронна пошта компанії.

Phone – Phone – Phone – номер телефону компанії.

Active – Active__c – Picklist – статус дійсності компанії (Yes\No).

Customer Priority – Customer_Priority__c – Picklist – пріоритет клієнта\компанії (High, Medium, Low).

Backlisted – Backlisted__c – Checkbox – надійність клієнта, якщо True, краще не мати справу, оскільки клієнт ненадійний.

Type – Type – Picklist – тип клієнта, може бути: Prospect, Customer, Direct, Customer, Channel, Installation Partner, Technology Partner.

Industry – Industry – Picklist – сфера у якій компанія працює, або надає послуги для наочності: Agriculture, Banking, Educations, Electronics.

Rating – Rating – Picklist – рейтинг клієнта (Hot, Warm, Cold).

Shipping Address – Shipping_Address – Address – адреса доставки компанії.

Варто зауважити що Salesforce при створенні полів автоматично додає до API назви поля частину __c. По цій частині зручно орієнтуватись. Також додається така ж частина до API назви кастомних об'єктів. При створенні полів, система також генерує поля за замовчуванням, які не можна видалити, в нашому випадку це

наступні поля: **Created By, Currency, Last Modified By, Owner**. Ці поля існують в усіх об'єктах, в тому числі і стандартних, які представляє система.

3.2. Реалізація компонент

Система складається з 4 аплікацій: **Team Member, Resources Management, Project Management, Company Management**, які в свою чергу складаються з компонент.

Ключовими компонентами є: **Timecard History, Resource Skill, Timecard Buttons, Resource Home, PDF Artifact, Salesforce Approvals**.

Salesforce Approvals Process

Цей процес відтворено в системі як важливий елемент прозорості. Головна мета – це запобігти зловживанням та помилками. Це є свого роду захист від наступних ситуацій: працівник ненавмисно вказав більше годин, ніж фактично працював, працівник навмисно вказав більше часу з метою отримання бонусів у вигляді більшої заробітної плати, тощо. Щоб уникнути такі випадки мухлювання впроваджено офіційну фіксації залогованого робочого часу відповідною людиною, яка за це відповідає.

Компонента впроваджена в іншу компоненту – Timecard History. Як відбувається процес підтвердження: працівник логу свій робочий час у компоненті Timecard History, вибирає всі необхідні записи, або просто натискає на кнопку «Send for Approval» і запит до відповідної людини надходить автоматично, також на електронну пошту приходить відповідне сповіщення. Відповідальна людина може або підтвердити, або відхилити запит. Усі етапи фіксуються автоматично, а також зберігається історія рішень [3, 6].

Team Member

Це головна аплікація для класу користувачів класу «співробітник». Саме тут охоплюються всі ключові сценарії цього класу при взаємодії з системою. У цій аплікації співробітник має можливість вносити, переглядати та редагувати свій

робочий час, логуючи час пропрацьованих на проєнках, навчаннях або інших активностях. Інтерфейс представляє можливість переглядати тижневий розклад часу, включаючи загальне число годин за тиждень. Інтерфейс реалізовано таким чином, тому що може бути дозволено працювати на декількох проєктах одночасно, і працівнику буде зручно це редагувати та вести в одній таблиці. Також дуже зручним є те, що можна взяти відгул на декілька годин, коли працівнику потрібно відлучитись, лише не декілька годин, в такому разі неоплачувані будуть фактичні години відсутності. Варто зазначити, що відпустки, навчання, вихідні, неоплачувані вихідні представлені як об’єкт «проєкт», така реалізація полегшує операції над даними. В додатку було передбачено попередження для працівників які не заповнили та не залогували свій робочий час, оскільки весь час логується, в тому числі і відгули то працівник повинен заповнити свій «табель». Кожного понеділка виконується функція, яка перевіряє залогований час, і якщо він менше то автоматично надсилається попередження на електронну пошту.

Окрім цього, аплікація відображає доступні свята та відображає наявні навички і сертифікації користувача. Для зручності реалізовано кнопки для логування таких типів таймкарт як лікарняні, неоплачувані відпустки, відпустки, участь в тренінгах або навчаннях, тощо. Також присутнє відображення днів доступних для відпустки, вихідних та відгулів по дитині.

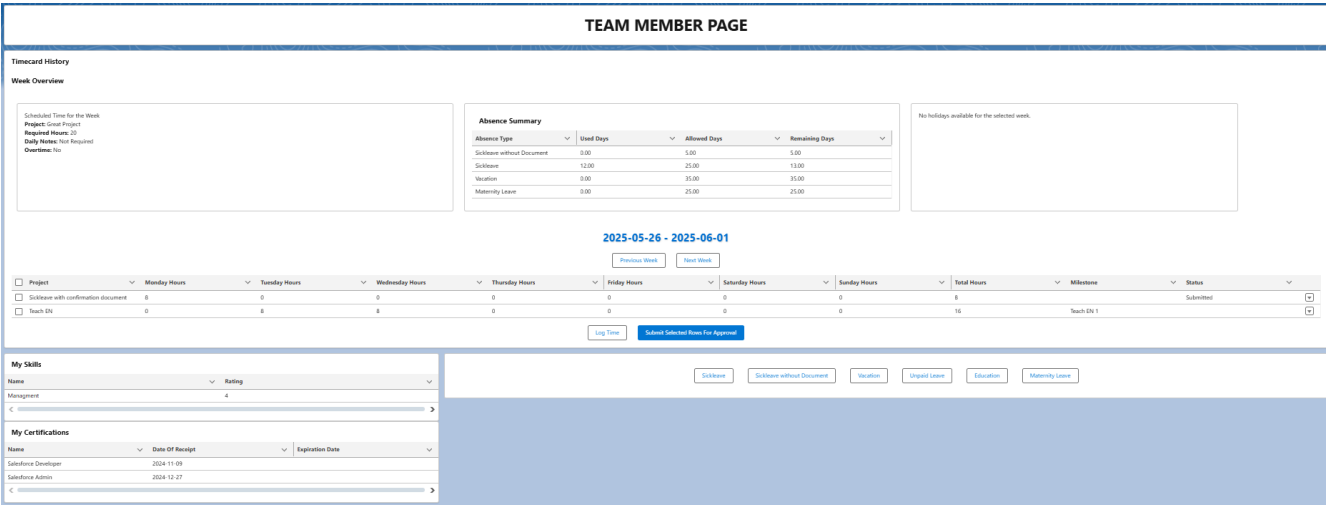


Рисунок. 3.1. Зовнішній вигляд аплікації Team Member

При логуванні часу лікарняного (Sickleave) використовується компонента для прикріплення файлів. Архітектура цієї компоненти реалізовано таким чином, що всі вкладені файли завантажуються на Amazon Web Services Simple Storage Service, що дозволяє безпечно, та надійно зберігати артефакти, які підтверджують наявність причини лікарняного, ця компонента не вимагає відразу прикріплювати артефакт, оскільки деякі медичні установи можуть не відразу видати відповідний документ. Дані необхідні для доступу до сервісу зберігання даних зберігаються надійно, у одному з сервісів Salesforce [9].

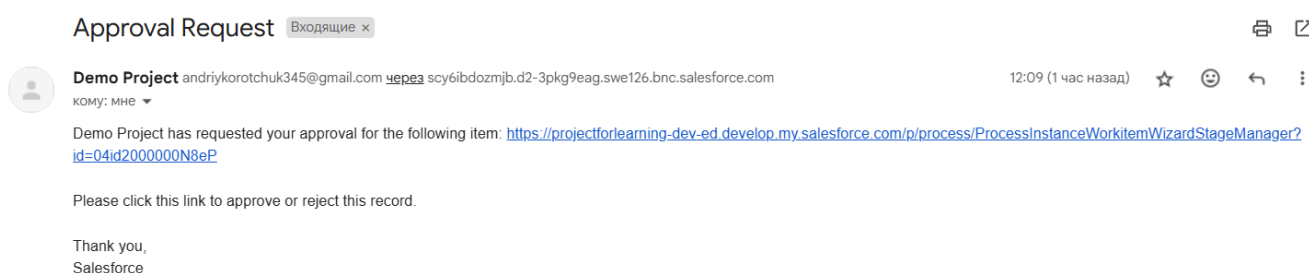


Рисунок. 3.2. Лист із підтверджувальним запитом

Аплікація тісно взаємодіє з підтверджувальним процесом (approval process) при логуванні часу, на електронну пошту приходить відповідний електронний лист і далі з ним оперує відповідна людина.

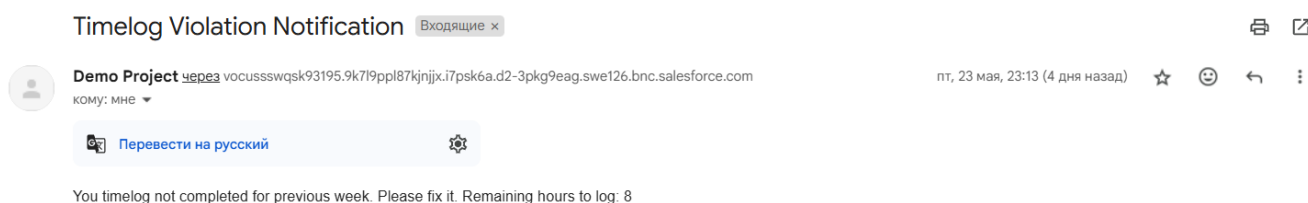


Рисунок. 3.3. Лист із попередженням про незалогований час

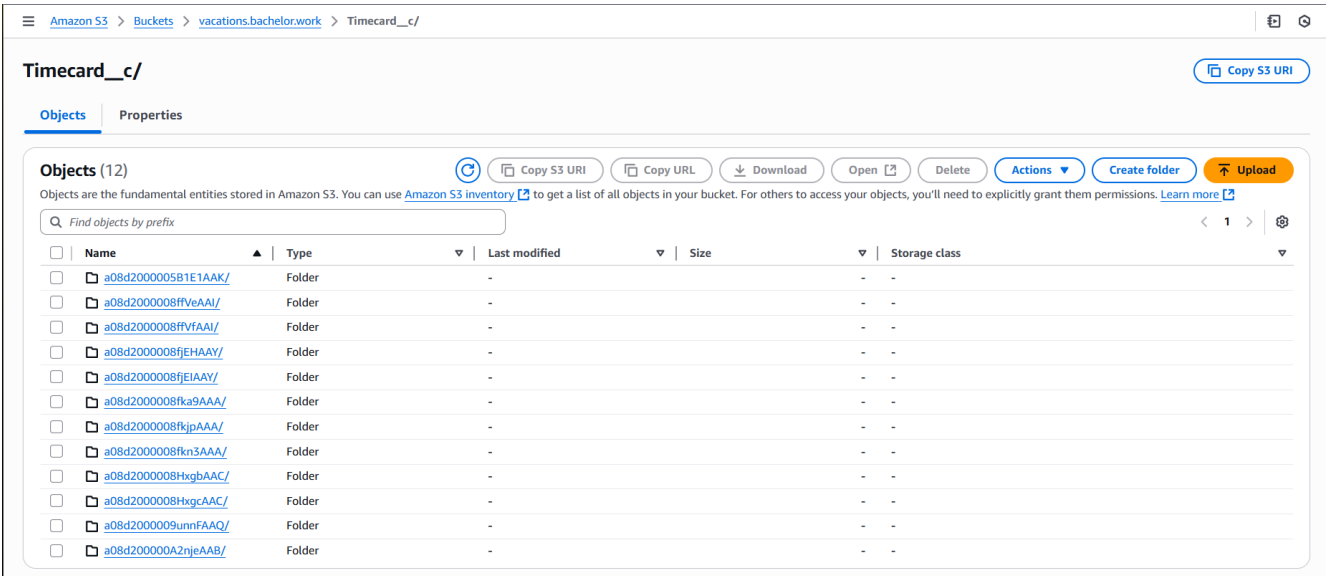


Рисунок. 3.4. AWS S3 Bucket для зберігання артефактів

Resources Management

Ця аплікація реалізує функціонал для підбору та додавання ресурсів на проєкт. Цією аплікацією буде користуватись клас користувачів проєктний менеджер та HR – менеджер. В межах своєї ролі цей клас користувачів зможе переглядати доступні ресурси, фільтрувати їх за допомогою **List Views** (фільтрувати можна по навичкам, рівнем досвіду та іншими критеріями) та додавати необхідних людей на проєкти. Разом з цим, проєктний менеджер зможе приймати певну участь у контролі: змінювати статуси призначень, надсилати

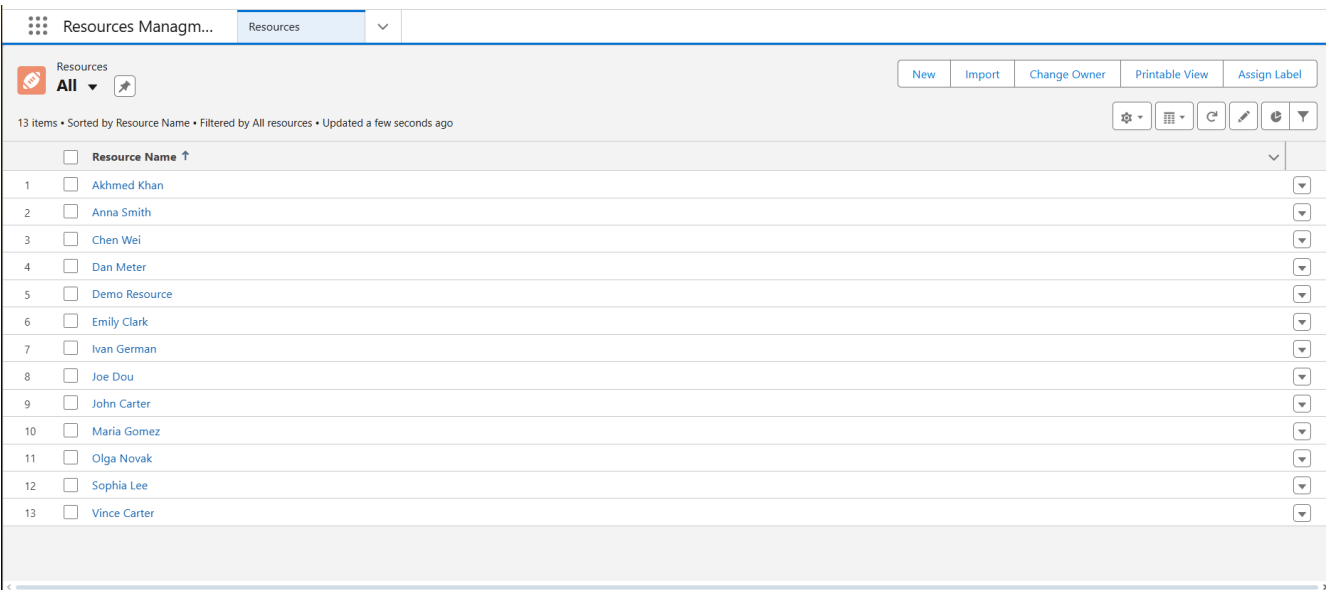


Рисунок. 3.5. Інтерфейс аплікації Resources Management

запити на затвердження, залишати коментарі і генерувати прості звіти про

ефективність ресурса. HR – менеджер в свою чергу матиме можливість лише генерувати звіти по ресурсах та можливо разом з проєктним менеджером підтверджувати запити на відпустки, лікарняні, тощо.

Усі ресурси будуть відображатись у списку.

List Views – це система фільтрації, яку можна комбінувати з різними варіантами фільтрів або застосовувати велику кількість різних фільтрів, List Views присутні у всіх аплікаціях і це дає можливість комфортно фільтрувати та шукати певні об'єкти.

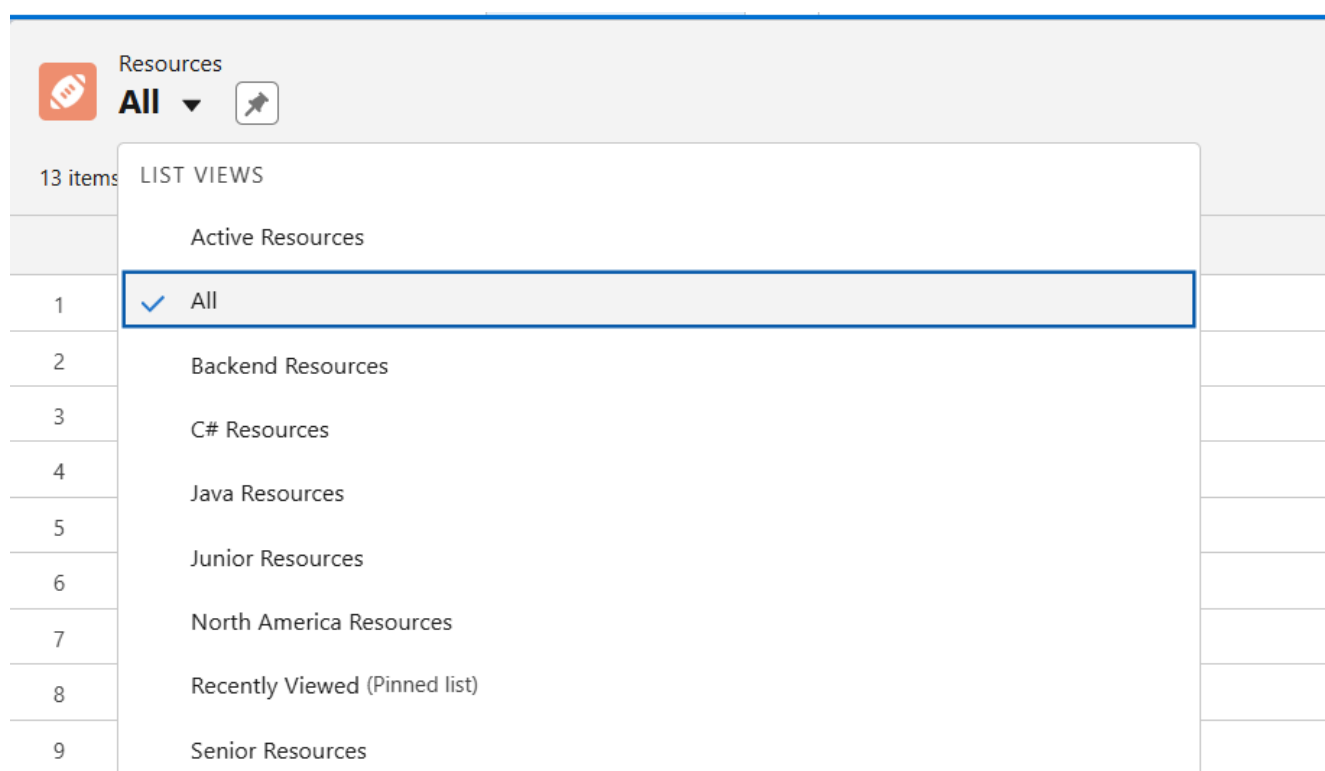


Рисунок. 3.6. Приклад List Views у Resource Management

Цей функціонал можна дуже легко та розширювати для персоналу.

HR – менеджер генеруватиме для керівництва компанії інвойси по ресурсах у наступному вигляді:

Project Management

Аплікація реалізує функціонал з створенням проєктів, додаванням та підбором ресурсів. В рамках цієї аплікації проєктний менеджер отримує наступні інструменти: створення проєктів, заповнення інформації щодо них, призначення ресурсів, створення робочих календарів, періодів (можуть бути подані як спринти),

також проєктний менеджер може перейти на певне призначення (Assignment) та зробити короткий звіт про його роботу на основі його таймкарт.

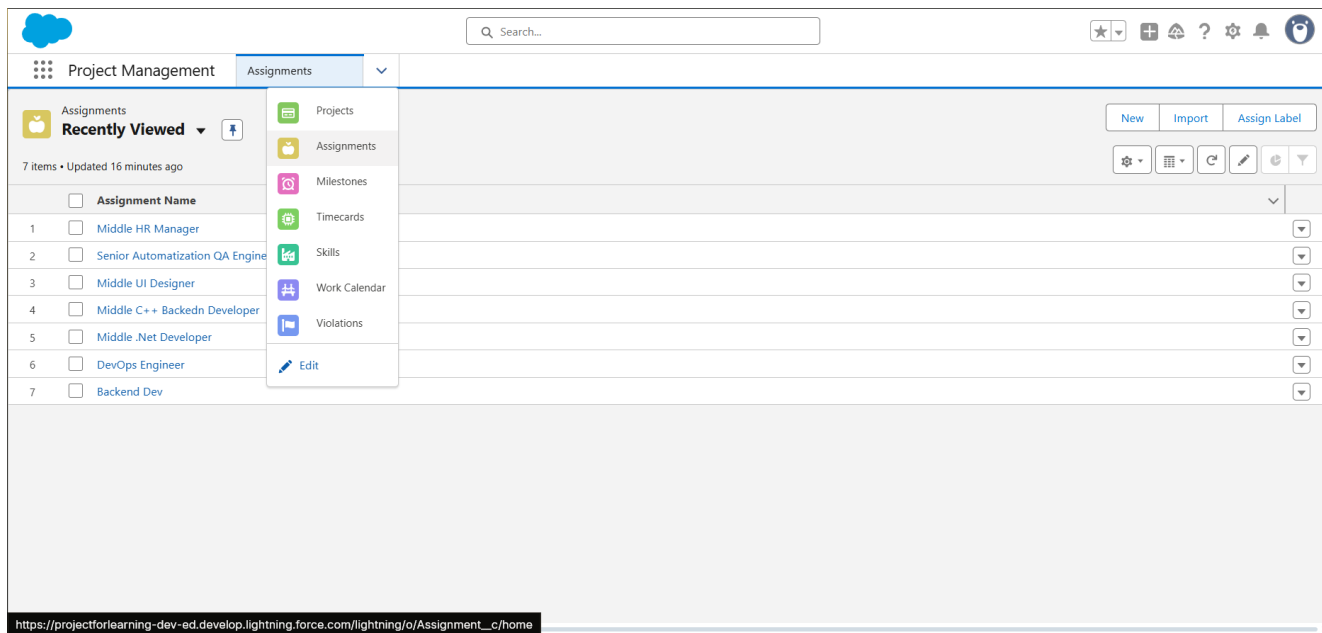


Рисунок. 3.7. Інтерфейс аплікації Project Management

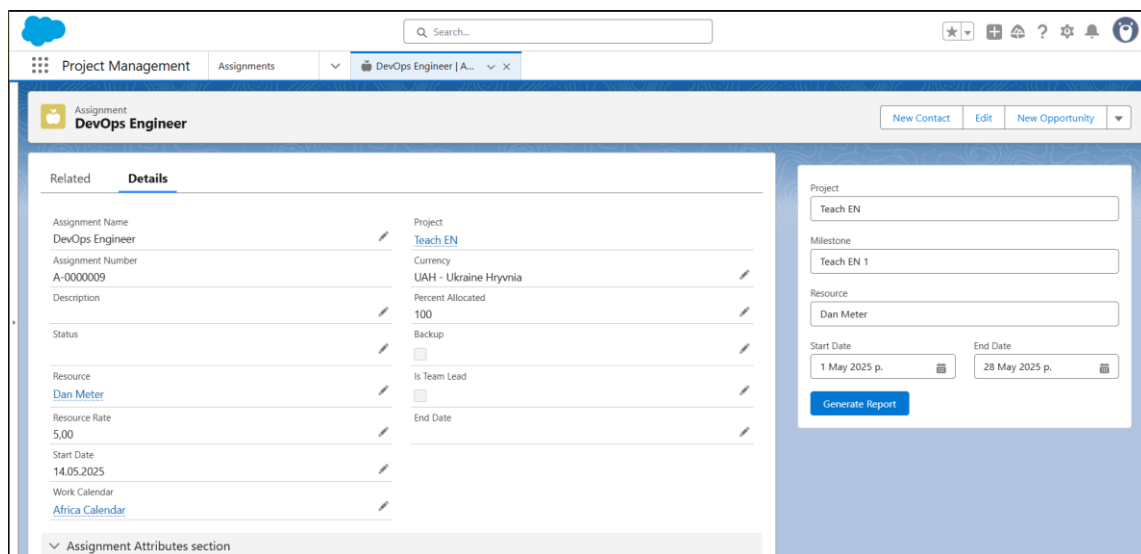


Рисунок. 3.8. Генерація репортів в Assignment

Company Management

Являється аплікацією для керівництва компанії, або людини з її сторони. В цій аплікації є доступ до всіх об'єктів.

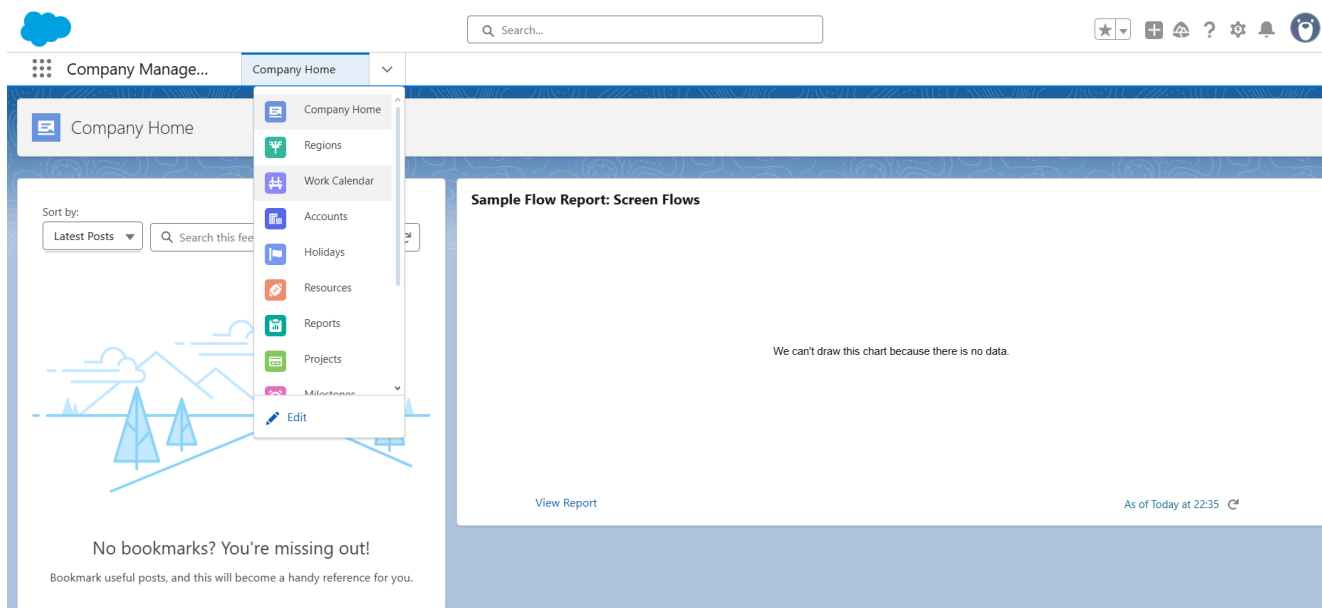


Рисунок. 3.9. Інтерфейс аплікації Company Management

Оскільки репорти реалізовано в мінімальній версії, то домашню сторінку додатку зроблено в перспективі переходу на більш розгорнуту і складну систему вбудованих репортів платформи Salesforce.

Цей клас користувачів можуть вільно переглядати всю інформації про проєкти та співробітників компанії, включаючи їхні заробітні плати. Також може генерувати інвойси відповідно відпрацьованого часу.

Вся фінансова інформація буде недоступна класам проєктний менеджер, співробітник та HR - менеджер завдяки гнучким налаштуванням доступу, який змогу налаштувати доступність того, чи іншого функціоналу, відображення різних полів, тощо. Така розгалужена система ролей і дозволів забезпечує безпечний і контрольований доступ різних класів до різних компонент та полів відповідно до їхніх обов'язків.

3.2. Перспективи масштабування системи

Масштабування системи є важливим етапом її подальшого розвитку, оптимізації, стійкості до навантаження і адаптації до змін у бізнес – процесах. Одним із головних напрямків масштабування системи являється функціональне розширення системи. Зокрема можна імплементувати нові модулі, наприклад такі

як розширений фінансовий облік, канбан – дошка та управління завданнями, розширення додаванням комунікативних модулів, можна використовувати стандартний модуль платформи Salesforce – Chatter, що може спростити комунікацію. Перехід на розширену систему репортів та звітів збільшить можливості аналізу проєктів, їхньої вигоди, продуктивності працівників. Для побудови аналітичних звітів доцільно інтегруватися з платформами які спеціалізуються на бізнес – аналітиці, такі як Salesforce Einstein Analytics, Tableau, Power BI.

Не менш важливим аспектом є горизонтальне масштабування користувачів. Система повинна підтримувати використання кількох командами або відділами. Можна переглянути перспективу додавання нових класів користувачів які будуть мати різні ролі. Розширення користувацької бази може потребувати введення підтримки багатомовної моделі для співробітників, що дає змогу адаптувати інтерфейс до міжнародних команд.

Технічне масштабування має не менш важливе значення у забезпеченні надійності та продуктивності. Платформа надає дуже потужні інструменти для інтеграції через REST API, і це дозволяє підключати зовнішні сервіси або створювати додаткові застосунки. Один випадок технічного масштабування було впроваджено – це зберігання артефактів на AWS S3. У складних випадках можна частини логіки повиносити у зовнішні мікросервіси що будуть незалежно обмінюватись даними з системою через API.

Із збільшенням обсягів даних важливо впровадити механізми обробки та зберігання. Можливими рішеннями є архівування історичних записів, оптимізація запитів, та автоматичне розбиття.

Можливі зовнішні інтеграції з системами управління проєктів, наприклад, можна інтегрувати відслідковування часу в JIRA та зробити автоматизацію, яка буде цей час переносити у таблицю, яку повинен заповнити співробітник, також можна інтегрувати канбан – дошку з JIRA.

Хорошим варіантом є інтеграція з HRM – системами які дозволять працювати з реальними даними людей і охоплювати набагато більший обсяг людей для пошуку відповідної людини для проєкту відповідно до спеціалізації, досвіду, зайнятості, сертифікацій та іншого.

Таким чином, ці всі варіанти підвищують гнучкість, масштабованість, та саме головне – корисність системи управління проєктами. Це відкриває можливість для розбудови цілої та повноцінної цифрової екосистеми, де дані з різних джерел поєднуюватимуться для досягнення більшого рівня точності, ефективності в управлінні проєктами і ресурсами та автоматизації.

ВИСНОВКИ

В межах проведеного дослідження було проведено повний цикл дослідження та розробки інформаційної системи управління проєктами. Проведено комплексне дослідження предметної області, було визначено основні виклики та потреби з якими стикаються компанії у сфері ІТ – послуг, в особливості у контексті управління ресурсами та забезпеченні прозорості бізнес – процесів у компанії.

Відповідно до поставлених завдань, було досягнуто наступних результатів:

- досліджено особливості процесів управління в ІТ – компаніях, визначено основні бізнес – процеси та виклики з якими компанії стикаються, в особливості у контексті обліку робочого часу, управління відсутностями та іншими внутрішніми активностями; проаналізовано сучасні методології, які використовуються в управління проєктами, також винесено можливості, переваги, недоліки та обмеження платформи Salesforce для створення інформаційних систем, що допомогло обґрунтувати вибір цієї платформи для імплементації;

- розроблено модель даних відповідно до завдання, яка включає ключові компоненти, такі як: об'єкти, сутності, їхні атрибути, поля та зв'язки, які є необхідними для реалізації вимог користувачів; зібрано та формалізовано функціональні та нефункціональні вимоги до інформаційної системи; розроблено сценарії використання системи різними групами користувачів (співробітник, HR – менеджер, проєктний менеджер, керівник компанії або менеджер з успіху клієнтів та адміністратора системи);

- реалізовано інформаційну систему, яка в цілому відповідає сформульованим вимогам та включає такий функціонал як управління відсутностями, облік робочого часу, генерація звітів та інтеграція бізнес-процесів у одну систему;

- проаналізовано можливі шляхи масштабування системи управління проєктами шляхом додавання нового функціоналу або інтеграції з іншими

сервісами/додатками дослідивши різні архітектурні, інтеграційні та інфраструктурні рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anderson D. Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, 2010. 290 с. (дата звернення 15.10.2024).
2. Chung L. Non-Functional Requirements in Software Engineering. Springer, 2000. 472 с. (дата звернення 29.10.2024).
3. Classic Approval Processes. URL: https://help.salesforce.com/s/articleView?id=platform.what_are_approvals.htm&type=5 (дата звернення 01.04.2025).
4. Gause D., Weinberg G. Exploring Requirements: Quality Before Design. Dorset House, 1989. 304 с. (дата звернення 24.10.2024).
5. IEEE Guide to Software Requirements Specifications. IEEE Std 830-1998. IEEE, 1998. 40 с (дата звернення 27.10.2024).
6. McCollum P. Practical Salesforce Architecture: Understanding and Deploying the Salesforce Ecosystem for the Enterprise. O'Reilly, 2023. 183 с.
7. Simsion G., Witt G. Data Modeling Essentials. 4th ed. Morgan Kaufmann, 2018. 560 с. (дата звернення 23.10.2024).
8. Stellman A., Greene J. Learning Agile: Understanding Scrum, XP, Lean, and Kanban. O'Reilly, 2015. 420 с. (дата звернення 16.10.2024).
9. Vohra D. AWS Development Essentials. Birmingham : Packt Publishing, 2015. 170 с. (дата звернення 23.10.2024).
10. Waterfall – Методологія Каскад в розробці. URL: <https://skalar.ua/ua/methodology/waterfall> (дата звернення 25.09.2024).
11. Wiegers K., Beatty J. Software Requirements. 3rd ed. Microsoft Press, 2013. 608 с. (дата звернення 25.10.2024).
12. Водоспадна модель. URL: https://uk.wikipedia.org/wiki/Водоспадна_модель (дата звернення 24.09.2024).

13. Гнучка розробка програмного забезпечення. URL: https://uk.wikipedia.org/wiki/Гнучка_розробка_програмного_забезпечення (дата звернення 01.10.2024).

14. Канбан. URL: <https://uk.wikipedia.org/wiki/Канбан> (дата звернення: 20.10.2024).

15. Навчальні матеріали. URL: <https://trailhead.salesforce.com/> та <https://www.salesforce.com/services/professional-services/resources/> (дата звернення 01.10.2024).

16. Офіційна документація платформи Salesforce. URL: <https://help.salesforce.com/s> (дата звернення 27.10.2024).

17. Скрам – це ефективне управління проєктами. URL: <https://brainrain.com.ua/uk/scrum-upravlinnya-proektom/> (дата звернення: 17.10.2024).

18. Скрам. URL: <https://uk.wikipedia.org/wiki/Скрам> (дата звернення 11.10.2024).

19. Форум спільноти в Україні. URL: <https://salesforce.in.ua/> (дата звернення 04.11.2024).

20. Що таке Agile і як його застосувати в бізнесі. URL: <https://brainrain.com.ua/uk/hto-takoe-agile-ua/> (05.10.2024).