

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти магістр

на тему: **«Об'єктно-орієнтована модель інформаційної системи на прикладі
автоматизованого робочого місця співробітника підприємства»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи та
технології
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти магістр
групи 126ІСТ_мд_2022
Кошеленко Олег Володимирович
Керівник: Уткін Юрій Вікторович
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2024 року

ВСТУП

Актуальність теми. Протягом останніх десятиліть розвиток інформаційних технологій суттєво вплинув на автоматизацію бізнес-процесів у різних галузях. Інформаційні системи стали невід'ємною частиною діяльності підприємств, забезпечуючи ефективне управління ресурсами, проектами та персоналом. Особливо актуальним є впровадження автоматизованих систем у контексті зростаючої потреби у підвищенні продуктивності та зниженні витрат.

Роботи вчених [2, 3] і фахівців з управлінських інформаційних систем [4, 6], включаючи MIS та DSS, відображають теоретичні основи та практичне застосування досліджень у цій області.

Зв'язок роботи з науковими програмами, темами. Робота відповідає дослідженням в межах науково-дослідної роботи «Розвиток підприємництва: управлінські, економічні, інноваційна та правові аспекти» відповідно до договору №9 від 15.05.2023 р. між ТОВ «ПАФ Гарант» та Полтавським державним аграрним університетом.

Метою кваліфікаційної роботи є розробка об'єктно-орієнтованої моделі автоматизованої інформаційної системи підприємства для управління проектами, працівниками та ресурсами.

Завданнями кваліфікаційної роботи є:

- аналіз сценаріїв проектування архітектури сучасних інформаційних систем;
- застосування методології проектування інформаційної системи підприємства;
- розроблення об'єктно-орієнтованої моделі автоматизованого робочого місця співробітника.

Об'єктом дослідження є автоматизовані інформаційні системи підприємств, що займаються управлінням проектами, працівниками та ресурсами.

Предметом дослідження є об'єктно-орієнтована модель автоматизованої інформаційної системи, її компоненти та алгоритми взаємодії.

Методи дослідження – проведені в роботі дослідження базуються на методах системного аналізу, об'єктно-орієнтованого моделювання, використанні UML для

візуалізації процесів, а також методах економічного аналізу для оцінки ефективності впровадження системи.

Інформаційна база кваліфікаційної роботи складається з наукових статей, міжнародних аналітичних видань і звітів, матеріалів наукових конференцій, інтернет-ресурсів, що містять інформацію про сучасні тенденції розвитку інформаційних систем, їх проєктування та впровадження.

Елементи наукової новизни полягають у розробці та дослідженні об'єктно-орієнтованої моделі інформаційної системи підприємства, яка інтегрує управління проектами, персоналом і ресурсами, забезпечуючи їх ефективну взаємодію.

Практична значущість роботи полягає в можливості повторного застосування та модифікації розробленого програмного коду для автоматизації процесів у підприємствах різних галузей. Отримані результати можуть бути корисними для керівників підприємств, що прагнуть оптимізувати свою діяльність, а також для розробників інформаційних систем, які займаються створенням адаптивних рішень.

Апробація результатів дослідження відбувалася шляхом оприлюднення доповідей на наукових конференціях, семінарах.

Публікації. За результатами проведеного дослідження опубліковано тези: «Моделювання інформаційної системи агропромислового комплексу з використанням об'єктно-орієнтованого та імітаційного методів», Стратегічний менеджмент агропродовольчої сфери в умовах глобалізації економіки: безпека, інновації, лідерство: матеріали II Міжнародної науково-практичної конференції, 27 вересня 2024 р. Полтава; «Розвиток інструментарію для моделювання та розробки інформаційних систем на основі об'єктно-орієнтованого підходу»: Студентські роботи за науковою тематикою кафедри інформаційних систем та технологій: матеріали XXI щорічного міждисциплінарного семінару, 20 листопада 2024 р. Полтава.

Структура та обсяг кваліфікаційної роботи логічно пов'язані з задачами досліджень. Робота містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг текстової частини дипломної роботи складає 65 сторінок формату А4. Вона містить 12 рисунків і 4 таблиці. У роботі використано 45 науково-технічних джерел.

РОЗДІЛ 1

АНАЛІЗ СЦЕНАРІЇВ ПРОЄКТУВАННЯ АРХІТЕКТУРИ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Класифікація інформаційних систем підприємств

Основним елементом інформатизації діяльності будь-якого підприємства є інформаційна система (ІС). Класифікація інформаційних систем підприємств ґрунтується на ієрархічному принципі та відповідає багаторівневій організаційно-штатній структурі. Вони поділяються на такі категорії:

1) Інформаційні системи підприємств базового рівня, основна мета яких – забезпечення комп'ютерної підтримки роботи робітників різних спеціальностей. Ці системи дозволяють підвищити якість профілактичної та діагностичної роботи, особливо у випадках масового обслуговування за умов дефіциту часу кваліфікованих спеціалістів. Залежно від виконуваних завдань виділяють:

- інформаційно-довідкові системи (призначені для пошуку та надання інформації за запитом користувача),

- консультативні системи (для прогнозування різних ситуацій, включаючи прогнозування та розробку рекомендацій щодо покращення етапів виробництва),

- приладово-комп'ютерні системи (для інформаційної підтримки та/або автоматизації виробничих процесів, що проводяться при виробництві та випуску продукції),

- автоматизовані робочі місця спеціалістів (для автоматизації всього технологічного процесу працівника відповідної спеціалізації та забезпечення інформаційної підтримки в прийнятті рішень) [18, 36].

2) Інформаційні системи рівня великих підприємств представлені такими основними групами:

- інформаційні системи консультативних центрів (призначені для підтримки функціонування відповідних підрозділів та надання інформаційної підтримки спеціалістам під час консультування та прийняття рішень у невідкладних

випадках), банки інформації (містять дані про склад працівників, клієнтів, статистичні відомості та інші необхідні дані),

- персоніфіковані реєстри (містять інформацію про прикріплений або спостережуваний контингент клієнтів на основі формалізованої історії закупівель продукції),

- інформаційні системи великих підприємств (об'єднують всі інформаційні потоки в єдину систему та автоматизують різні види діяльності),

- інформаційні системи науково-дослідних інститутів (НДІ) та університетів (забезпечують інформатизацію навчального процесу, науково-дослідної роботи та управлінської діяльності) [13].

3) Інформаційні системи територіального рівня. Сюди входять:

- інформаційні системи об'єднань підприємств територіального рівня,
- інформаційні системи для вирішення виробничих та технологічних завдань, що забезпечують інформаційну підтримку діяльності працівників спеціалізованих служб,

- комп'ютерні телекомунікаційні мережі, що створюють єдиний інформаційний простір на рівні регіону.

1.2 Хронологія розвитку та різновиди інформаційних систем

До 60-х років XX століття функції інформаційних систем були досить простими: діалогова обробка запитів, зберігання записів, ведення бухгалтерського обліку та інші завдання електронної обробки даних (electronic data processing – EDP). Згодом, із появою концепції управлінських інформаційних систем (management information systems – MIS), додалися нові функції, спрямовані на надання персоналу необхідних для прийняття управлінських рішень звітів, складених на основі зібраних даних про процеси (information reporting systems).

У 70-х роках стало очевидно, що жорстко встановлені форми звітів не задовольняли потреби користувачів. Тоді виникла концепція систем підтримки

прийняття рішень (decision support systems – DSS). Ці системи мали забезпечувати спеціалізовану й інтерактивну підтримку процесів ухвалення унікальних рішень у швидко змінюваних умовах (рис. 1.1).



Рисунок 1.1 – Концепція систем підтримки прийняття рішень DSS

У 80-х роках розвиток обчислювальних потужностей мікро-ЕОМ, прикладних програм і телекомунікаційних мереж дав поштовх до виникнення феномена кінцевого користувача (end user computing). Відтоді кінцеві користувачі отримали можливість самостійно використовувати обчислювальні ресурси для вирішення завдань, пов'язаних із їхньою професійною діяльністю, без необхідності звертатися до спеціалізованих інформаційних служб.

З розумінням того, що більшість топ-менеджерів безпосередньо не використовують результати роботи систем звітності або систем підтримки рішень, виникла концепція інформаційних систем для керівників (executive information systems – EIS). Ці системи призначалися для надання керівникам життєво важливої

інформації, зокрема про зовнішнє середовище, у той момент, коли це необхідно, і у форматі, який вони віддають перевагу (рис. 1.2).

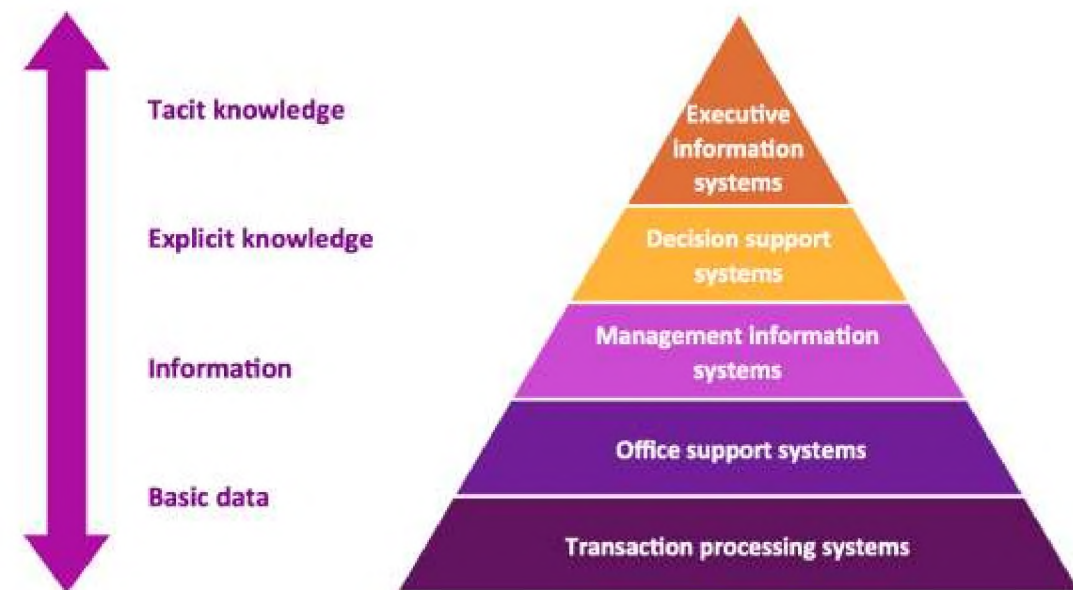


Рисунок 1.2 – Ієрархія інформаційних систем

Великим досягненням стало створення й застосування систем та методів штучного інтелекту (artificial intelligence – AI) в інформаційних системах. Експертні системи (expert systems – ES) та системи на основі знань (knowledge-based systems) визначили нову роль інформаційних систем. Сьогодні вони можуть надавати керівникам якісні рекомендації у спеціалізованих галузях.

У 1980 році з'явилася і продовжувала розвиватися в 90-х концепція стратегічної ролі інформаційних систем, які іноді називають стратегічними інформаційними системами (strategic information systems – SIS). Відповідно до цієї концепції інформаційні системи перестають бути просто інструментом обробки інформації для кінцевих користувачів усередині компанії. Тепер вони стають джерелом нових продуктів і послуг, що базуються на інформації, і мають забезпечити компанії конкурентні переваги на ринку.

Виробничі інформаційні системи включають категорію систем обробки транзакцій (transaction processing systems – TPS). Ці системи здійснюють

реєстрацію даних про процес. Типовими прикладами є інформаційні системи, які реєструють продажі, закупівлі та зміни стану. Результати такої реєстрації використовуються для оновлення баз даних про клієнтів, інвентар та інші організаційні бази даних. Системи обробки транзакцій також генерують інформацію для внутрішнього або зовнішнього використання. Наприклад, вони готують заявки клієнтів, платіжні відомості, товарні чеки, податкові та фінансові звіти. Системи обробки транзакцій обробляють дані двома основними способами. При пакетній обробці дані про операції накопичуються протягом певного часу і періодично обробляються. У реальному часі (або інтерактивно) дані обробляються негайно після завершення операції. Наприклад, система реєстрації продажів (point of sale – POS), яка використовується у роздрібній торгівлі, може використовувати електронні термінали для фіксації та передавання комерційних даних до регіональних комп'ютерних центрів у режимі реального часу або пакетами.

Системи управління процесом приймають найпростіші рішення, необхідні для керування виробничими процесами. Вони належать до категорії інформаційних систем, відомих як системи управління процесом (process control systems – PCS), які автоматично приймають рішення для регулювання фізичних процесів виробництва. Наприклад, нафтопереробні заводи та автоматизовані лінії збірки використовують такі системи. Вони контролюють фізичні процеси, обробляють дані, зібрані датчиками, і здійснюють управління процесом у реальному часі.

Ще однією функцією виробничих інформаційних систем є автоматизація традиційних ручних офісних методів і документообігу. Системи автоматизації офісу (office automation systems – OAS) збирають, обробляють, зберігають і передають інформацію у вигляді електронних документів. Такі автоматизовані системи використовують текстові процесори, системи передачі даних та інші інформаційні технології для підвищення ефективності роботи офісу. Наприклад, можливе використання текстових процесорів для обробки кореспонденції, електронної пошти для обміну повідомленнями, настільних видавничих систем для створення інформаційних бюлетенів компанії, а можливості телеконференцій – для проведення електронних зустрічей (рис. 1.3).

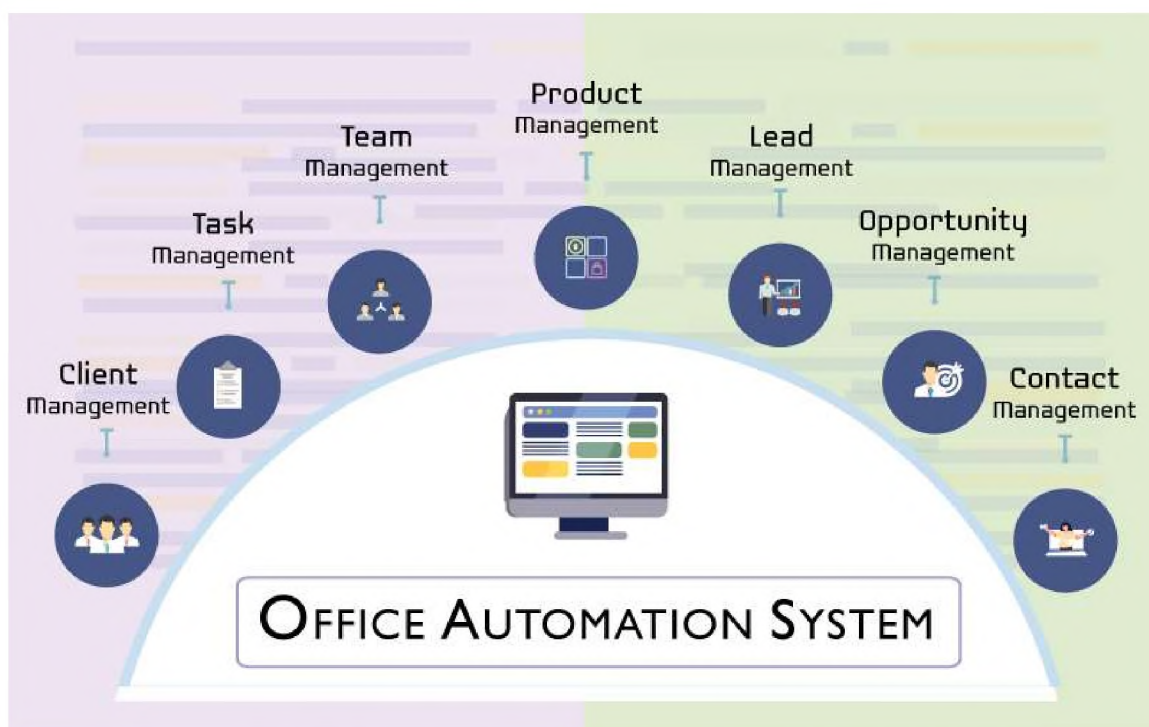


Рисунок 1.3 – Концепти системи автоматизації офісу OAS

Інформаційні системи, призначені для надання керівникам інформації, необхідної для прийняття ефективних рішень, називають управлінськими інформаційними системами (management information systems – MIS). Найважливіші для нас три основні типи управлінських інформаційних систем: системи звітності, системи підтримки прийняття рішень та системи підтримки стратегічних рішень.

Системи звітності (information reporting systems – IRS) є найпоширенішою формою управлінських інформаційних систем. Вони надають керівникам інформацію, необхідну для задоволення їхніх щоденних потреб при прийнятті рішень. Такі системи генерують та оформляють різні типи звітів, зміст яких заздалегідь визначається самими керівниками, щоб у них була лише потрібна інформація. Системи звітності вибирають необхідну інформацію про внутрішні процеси з баз даних, підготовлених виробничими інформаційними системами, а також інформацію ззовні з інших джерел. Звіти, що створюються системами звітності, можуть надаватися керівникам на вимогу, періодично або у зв'язку з подіями.

На додаток до експертних систем, інші типи штучного інтелекту включають нейронні мережі, машинне навчання та обробку природної мови. Нейронні мережі імітують роботу мозку, дозволяючи системам навчатися на основі великих обсягів даних та удосконалювати свої алгоритми з часом. Машинне навчання дозволяє системам знаходити закономірності в даних та робити прогнози або рішення без необхідності чітких інструкцій. Обробка природної мови дозволяє комп'ютерам розуміти та взаємодіяти з людською мовою, що відкриває можливості для створення розмовних систем, таких як віртуальні асистенти.

Загалом, сучасні інформаційні системи стали набагато більш інтегрованими, багатофункціональними і здатними до самостійного аналізу даних та підтримки процесів прийняття рішень. Використання штучного інтелекту та інших новітніх технологій надає організаціям інструменти для більш ефективного управління своїми ресурсами, підвищення конкурентоспроможності та швидшої адаптації до змін на ринку.

1.3 Врахування специфіки діяльності підприємств на прикладі медичних інформаційних систем

Однією з важливих категорій спеціалізованих медичних інформаційних систем є медичні приладово-комп'ютерні системи (МПКС). На сьогодні інформатизація медичної апаратури є одним із ключових напрямків розвитку медицини. Застосування комп'ютерів у поєднанні з вимірювальною та керуючою технікою дозволило створити ефективні засоби для автоматизованого збору інформації про стан пацієнта, її обробки в режимі реального часу та управління його станом. Це призвело до розвитку МПКС, що вивели на новий рівень інструментальні методи дослідження та інтенсивну терапію. МПКС належать до медичних інформаційних систем базового рівня. Основна їх особливість – робота у режимі безпосереднього контакту з об'єктом дослідження та в реальному часі. Ці системи є складними програмно-апаратними комплексами. Для їх роботи, крім

обчислювальної техніки, потрібні спеціальні медичні прилади, обладнання, телетехніка, засоби зв'язку [19, 20].

Типовими представниками МПКС є медичні системи моніторингу стану пацієнтів під час складних операцій, системи комп'ютерного аналізу даних томографії, ультразвукової діагностики, радіографії; системи автоматизованого аналізу даних мікробіологічних та вірусологічних досліджень, аналізу клітин і тканин [10, 21].

У складі МПКС можна виділити три основні компоненти: медичне, апаратне та програмне забезпечення.

1) Медичне забезпечення включає методи реалізації вибраних медичних завдань відповідно до можливостей апаратної та програмної частин системи. Сюди належать набори методик, вимірювані фізіологічні параметри, методи їх вимірювання, а також визначення способів та допустимих меж впливу системи на пацієнта.

2) Апаратне забезпечення охоплює засоби реалізації технічної частини системи, включаючи обладнання для отримання медико-біологічної інформації, засоби лікувального впливу та обчислювальну техніку.

3) Програмне забезпечення включає математичні методи обробки медико-біологічної інформації, алгоритми та програми, що забезпечують функціонування всієї системи [35].

1.4 Аналіз методологій проєктування інформаційних систем

Метою методології створення інформаційних систем є організація процесу побудови ІС та забезпечення управління цим процесом для того, щоб гарантувати виконання вимог як до самої ІС, так і до характеристик процесу її розробки. Основними завданнями, які повинна вирішувати методологія створення корпоративних ІС (разом із відповідним набором інструментальних засобів), є наступні:

- забезпечувати створення корпоративних ІС, які відповідають вимогам щодо автоматизації бізнес-процесів і відповідають цілям та завданням організації;
- гарантувати створення системи з заданою якістю в межах встановлених строків і бюджету;
- підтримувати зручну дисципліну супроводу, модифікації та розширення системи, щоб ІС могла швидко адаптуватися до змінюваних вимог компанії;
- забезпечувати створення корпоративних ІС, що відповідають вимогам відкритості, переносимості та масштабованості;
- забезпечувати використання у розроблюваній ІС наявного інформаційно-технологічного потенціалу організації (ПЗ, баз даних, обчислювальної техніки, телекомунікацій, технологій).

Методологія має забезпечувати зниження складності процесу створення ІС завдяки повному та точному опису цього процесу і застосуванню сучасних методів і технологій на всьому життєвому циклі ІС – від задуму до реалізації.

У 1990-ті роки світ зіткнувся з кардинальними змінами як на ринках товарів і послуг, так і в інформаційних технологіях (ІТ). Сучасні корпоративні ІС стали ключовим чинником успіху компаній на ринку. Для виконання своїх завдань вони повинні вирішувати значно складніші проблеми, ніж раніше. Відповідно до високої динаміки змін на ринку, вимоги як до функціоналу ІС, так і до процесу їх створення значно посилилися. Суттєво скоротилися допустимі терміни розробки окремих додатків і систем в цілому. Також виникла необхідність у внесенні змін до вимог під час розробки для того, щоб система відповідала актуальним потребам організації на момент завершення, а не на момент початку проекту.

Досягнення в галузі ІТ дозволили подолати основні технічні та програмно-інструментальні проблеми створення корпоративних ІС. З'явилися сучасні апаратно-програмні платформи архітектури клієнт-сервер, засоби для проведення розподілених паралельних обчислень і управління обчислювальними процесами в гетерогенних мережах, а також методи і засоби розробки програм і баз даних, які дозволяють створювати відкриті, переносимі, масштабовані додатки та бази даних, а також швидко розробляти програми.

Практика показала, що для успішного створення складних систем, таких як корпоративні ІС, недостатньо мати тільки сучасні платформи і засоби розробки. Старі методології, створені у 1970-80-х роках і орієнтовані на мейнфрейми та однорідне середовище, застаріли і виявилися непридатними в нових умовах. За даними статистики, зібраної компанією Standish Group (США), понад 30% з 8380 проєктів, досліджених у США, виявилися невдалими, загальна вартість яких склала понад 80 мільярдів доларів. При цьому вчасно було завершено лише 16% проєктів, а перевищення витрат склало 189% від запланованого бюджету. Аналіз показав, що більшість невдач були пов'язані з відсутністю або неправильним застосуванням методологій проєктування ІС.

Сильний поштовх розвитку методологій дало впровадження двох принципово нових підходів до створення корпоративних ІС: інформаційного інжинірингу та реінжинірингу бізнес-процесів (BPR). Запропоновані в цих підходах методи дозволили описувати, аналізувати і проєктувати структуру та діяльність корпорацій подібно до технічних систем. Кожен із цих підходів породив свій клас методологій, які мають спільні характеристики. На сьогодні триває активний процес розвитку та вдосконалення методологій створення корпоративних ІС, у якому працюють багато провідних фахівців у всьому світі.

У 1994 році у Великій Британії було створено міжнародний консорціум DSDM (Dynamic Systems Development Method), який об'єднує понад 100 провідних компаній світу і постійно розробляє проєкти стандартів, методи та методології швидкої розробки додатків (ІС).

Сьогодні в нашій країні недостатньо оцінюється роль і значення методологій (на відміну від засобів проєктування прикладного програмного забезпечення). У найближчі роки задача створення корпоративних ІС на основі сучасних методологій постане перед багатьма вітчизняними організаціями. Водночас, на Заході методології продовжують вважати стратегічним продуктом. Багато з них досі є комерційною таємницею і використовуються як ноу-хау.

1.5 Обґрунтування вибору об'єктно-орієнтованого підходу до моделювання інформаційної системи підприємства

Для створення інформаційної моделі підприємства необхідно виконати формальний опис його діяльності. Спочатку проводиться дослідження установи, під час якого необхідно зібрати інформацію про інформаційні процеси, що в ній відбуваються. До цього процесу залучаються системні аналітики та експерти. Аналітики проводять опитування експертів, систематизують зібрані дані, виконують декомпозицію завдань закладу, а також оцінюють їх з точки зору точності та візуальної наочності.

До експертів можуть належати керівництво закладу, управлінці структурних підрозділів, а також персонал підприємства. Існує кілька відомих методів для збору інформації про процеси, які відбуваються у медичному закладі, зокрема:

- вивчення керівних документів;
- спостереження за діяльністю експертів;
- опитування;
- використання накопиченого досвіду;
- опис функціонування підприємства.

Вивчення керівних документів є ефективним способом для отримання базових уявлень про структуру підприємства, а також для формулювання питань до експертів. В процесі цього етапу аналітик формує бібліотеку документів. Спостереження за діями експертів дозволяє зануритися в процес, вивчити наявні проблеми та побачити реальну роботу установи. Проте надмірно тривалі спостереження можуть призвести до звикання до існуючого стану справ і, як наслідок, аналітик може не помітити альтернативні шляхи опису функцій закладу.

Опитування експертів може відбуватися як у форматі діалогу, так і за допомогою анкетування, яке дозволяє швидко зібрати інформацію від великої кількості експертів. Анкетування особливо доцільне, коли необхідно оперативно зібрати дані про роботу спеціалістів та фахівців. Правильне формулювання питань є ключовим для досягнення точних результатів опитування.

При аналізі великої кількості підприємств виявляється, що типові підрозділи, такі як бухгалтерія, реєстратура або цехи, виконують подібні функції і використовують одні й ті самі терміни. Це спрощує можливість застосування аналогічних підходів для опису інших підприємств та організацій із схожими підрозділами.

Ще одним методом збору інформації є створення аналітиком опису роботи підприємства, який пізніше коригується експертами. Проте для успішної реалізації цього методу потрібна активна підтримка експертів.

Симбіоз кількох методів опитування може значно підвищити ефективність дослідження підприємства. Спочатку досліджуються керівні документи, після чого здійснюється спостереження за реальними діями працівників, а потім проводяться опитування експертів для опису критичних ситуацій, які не зазначені у керівних документах, але є важливими у реальному житті закладу. Повторне дослідження документів, регулярне спостереження та експертні коментарі дозволяють сформувати чітку модель роботи підприємства.

Опитування є найменш формалізованим методом збору інформації, а його ефективність значною мірою залежить від правильного вибору експертів. Мета опитування – отримати інформацію, необхідну для подальшого моделювання певних частин інформаційної системи підприємства. Під час аналізу проводиться послідовність опитувань, яка дозволяє уточнити деталі моделі. Існує кілька типів опитувань, які відрізняються за метою, серед них: збирання фактів, визначення проблем та проведення нарад для прийняття рішень.

Для структурування процесу опитування його можна поділити на три основні етапи: підготовка, проведення і завершення. На етапі підготовки визначається експерт, програма зустрічі та вивчаються керівні документи. Під час опитування питання слід формулювати так, щоб отримати максимально повний опис завдань, а результати документувати одразу після зустрічі для уникнення втрати інформації.

Після збору інформації експертів розпочинається процес розробки інформаційної моделі за допомогою обраної мови моделювання, такої як математичне моделювання, структурні нотації або об'єктно-орієнтоване

моделювання. Підприємство можна представити у вигляді ієрархічного дерева об'єктів, де для кожного об'єкта описуються вхідні та вихідні дані, а також його функції.

Створення моделі розпочинається з загального опису об'єкта, який згодом деталізується і рецензується. Цей опис включає ключові функції системи та основні компоненти, а також структурує систему як «чорну скриньку», визначаючи найважливіші входи, управління, виходи та механізми.

Підготовка до моделювання включає визначення мети, вибір точки зору та визначення типу моделі. Збір інформації може включати читання керівних документів, спостереження за операціями, анкетування, опитування експертів та використання набутих знань для формування системи, яку пізніше можна скоригувати за участю експертів.

Декомпозиція системи починається зі складання списку основних типів даних та функцій. Цілі та точки зору на модель визначаються на ранніх етапах моделювання, і це допомагає точно формулювати запитання, які модель повинна вирішити.

Пропонується використати об'єктно-орієнтований підхід (ООП) для побудови інформаційної системи. ООП базується на концепціях об'єктів та класів, що дозволяє створювати гнучкі та добре структуровані системи. Об'єктно-орієнтоване проектування відповідає на два основні запитання: з яких частин складається система та яка відповідальність кожної з них.

Використання об'єктної моделі забезпечує низку переваг:

- можливість повторного використання компонентів;
- створення гнучких систем, що легко змінюються;
- зменшення ризиків розробки;
- краще сприйняття складних систем людьми.

На основі цього аналізу можна зробити висновок, що об'єктно-орієнтований підхід до розробки ІС є доцільним і ефективним завдяки його гнучкості, скороченню витрат часу та ресурсів, а також можливості масштабування для різних галузей виробництва.

Висновки до розділу 1

У першому розділі виконано аналіз сценаріїв проєктування архітектури сучасних інформаційних систем (ІС), їх класифікації, історії розвитку, а також специфіці застосування у різних галузях. Цей розгляд дозволяє краще зрозуміти еволюцію, особливості функціонування і сучасні виклики, пов'язані з проєктуванням інформаційних систем. Розглянуто поділ ІС за рівнями: від базових систем підприємств до територіальних. Такий ієрархічний підхід відображає різноманіття функцій і масштабів використання ІС, що охоплюють різні аспекти діяльності – від автоматизації робочих місць до створення єдиного інформаційного простору регіону. Це підкреслює значущість правильного вибору архітектури ІС відповідно до потреб і рівня підприємства.

Хронологічний огляд розвитку інформаційних систем виявляє еволюцію їх функціоналу: від простих систем обробки даних до комплексних рішень із використанням штучного інтелекту. Особливу увагу приділено концепціям управлінських і стратегічних інформаційних систем, які забезпечують підтримку прийняття рішень у швидко змінних умовах. Застосування інновацій, таких як нейронні мережі та машинне навчання, підкреслює роль ІС як рушія технологічного прогресу і джерела конкурентних переваг.

Розгляд специфіки медичних інформаційних систем (МІС) демонструє, як адаптація загальних принципів проєктування до потреб галузі сприяє покращенню якості послуг. МПКС (медичні приладово-комп'ютерні системи) зосереджуються на роботі в реальному часі і використовуються для діагностики, моніторингу та терапії. Інтеграція медичного, апаратного та програмного забезпечення створює багатофункціональні комплекси, що поліпшують точність і оперативність прийняття рішень у медицині.

Аналіз методологій проєктування ІС розкриває ключові принципи та виклики, пов'язані зі створенням сучасних корпоративних систем. Еволюція підходів, від традиційних методів до динамічних методологій, таких як DSDM, відображає прагнення до підвищення ефективності та адаптивності процесу

розробки. Акцент на інформаційному інжинірингу та реінжинірингу бізнес-процесів засвідчує необхідність глибокого розуміння потреб підприємств для побудови ІС, здатних забезпечити їхню конкурентоспроможність.

Проектування архітектури сучасних ІС є багатогранним процесом, що враховує різні аспекти – від технологічного до організаційного. Ефективність створених рішень значною мірою залежить від вибору методології, адаптації до специфіки галузі та впровадження інноваційних технологій. Ці аспекти створюють основу для подальших досліджень і розробок у сфері інформаційних систем.

На основі проведеного аналізу сформульовано загальне завдання, яке полягає у розробці комплексу об'єктно-орієнтованих моделей інформаційної системи у вигляді UML діаграм для підтримки етапів проектування, розгортання та супроводу. Загальне завдання розділено на три часткові задачі, дві з яких розглянуті у наступних розділах.

РОЗДІЛ 2

МЕТОДОЛОГІЯ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄМСТВА

2.1 Особливості запропонованої методології проєктування інформаційної системи підприємства

Запропонована методологія створення корпоративних інформаційних систем (ІС) складається з двох основних взаємопов'язаних частин: методології аналізу ІС, яка включає опис діяльності організації та формування вимог до ІС на основі бізнес-процесів, і методології проєктування від даних, призначеної для проєктування та швидкої розробки програмного й інформаційного забезпечення ІС. Ця методологія базується на ітераційній спіральній моделі життєвого циклу ІС. Її ключовою особливістю є охоплення всіх етапів життєвого циклу, із фокусом на початкові етапи створення корпоративної ІС. Основне завдання цих етапів – формування вимог до системи, які точно відповідають цілям і задачам організації.

Відповідно до підходу інформаційного інжинірингу, що його Джеймс Мартін визначає як "застосування взаємопов'язаного набору формальних технологій (моделей) для планування, аналізу, проєктування і створення інформаційних систем на рівні корпорацій або їхніх окремих частин...", процес створення ІС відбувається через побудову та розвиток моделей [12]. Реалізація методології базується на використанні узгодженого комплексу інструментальних засобів, що забезпечують високий рівень автоматизації всіх процесів, які виконуються впродовж життєвого циклу ІС.

Отже, основою пропонованої методології є:

- ітераційна спіральна модель життєвого циклу ІС;
- комплекс розвиваючих систем узгоджених моделей;
- методологія аналізу ІС на основі бізнес-процесів;
- методологія проєктування від даних;
- комплекс узгоджених інструментальних засобів.

2.2 Ітераційна модель життєвого циклу інформаційних систем підприємства

Методологія описує процес створення та супроводу інформаційних систем у вигляді життєвого циклу (ЖЦ) ІС, представляючи його як послідовність стадій, кожна з яких поділена на етапи й процеси. Для кожного етапу визначається послідовність робіт, очікувані результати, методи та засоби, необхідні для виконання завдань, а також ролі та відповідальність учасників. Це формальне описання ЖЦ дозволяє планувати й організовувати процес колективної розробки та забезпечувати управління ним.

Життєвий цикл ІС, визначений методологією, включає стадії аналізу, проєктування, розробки, тестування та інтеграції, впровадження, супроводу та розвитку системи. У таблиці 2.1 подані основні етапи кожної стадії ЖЦ, процеси управління та інтегральні процеси:

Таблиця 2.1 – Зміст етапів життєвого циклу ІС

Стадії ЖЦ	Основні етапи
Аналіз	Обстеження, аналіз існуючих моделей ІС, формування вимог до ІС
Проєктування	Концептуальне проєктування, розробка архітектури ІС, формування вимог до додатків
Розробка	Прототипування та тестування додатків, підготовка документації
Тестування та інтеграція	Інтеграція додатків у систему, тестування системи
Впровадження	Навчання користувачів, розгортання системи
Супровід	Експлуатація, усунення помилок, внесення змін

Процеси, що супроводжують життєвий цикл, включають управління конфігурацією, документацію, перевірки та інтеграцію. Сучасні засоби дозволяють швидко проєктувати, прототипувати та тестувати додатки і бази даних на основі моделей, створених на кожному етапі.

2.3 Методологія аналізу ІС на основі бізнес-процесів

Методологія описує процес створення корпоративних ІС як побудову й розвиток систем узгоджених моделей, починаючи з моделей, які описують діяльність організації, і завершуючи готовою інформаційною системою. Відправною точкою цього процесу є моделі бізнес-процесів організації. Це дозволяє об'єктивно сформулювати вимоги до ІС. Важливі моделі включають архітектуру ІС, вимоги до програмного й інформаційного забезпечення, які пізніше трансформуються у проекти баз даних і додатків.

Метою початкових етапів аналізу є формування вимог до ІС, які відображають цілі й задачі організації. Щоб створити ІС, яка відповідатиме потребам організації, необхідно спершу визначити ці цілі та завдання. Основою діяльності організації є її бізнес-процеси, які забезпечують реалізацію виробництва товарів і послуг. Кожен бізнес-процес має чітко визначений початок і кінець, а також зовнішні та внутрішні інтерфейси.

Методологія пропонує будувати опис діяльності організації як процес створення системи моделей, що базуються на бізнес-процесах. У процесі деталізації моделей повинна забезпечуватися узгодженість всіх функцій і цілей організації, що підтримується сучасними CASE-засобами.

В процесі опису діяльності організації формується три основні системи моделей: стратегічна, укрупнена і детальна. Всі ці моделі базуються на бізнес-процесах і описують ключові аспекти організації. Додатково додається система моделей, яка враховує аспекти, що не стосуються безпосередньо бізнес-процесів, але є важливими при створенні інформаційної системи.

Стратегічна система моделей містить загальні концепції і напрямки діяльності організації, її основні цілі та завдання. Вона дозволяє сформулювати довгострокові пріоритети та планування розвитку організації. Ця модель служить основою для створення більш детальних і конкретних описів процесів і вимог до інформаційної системи.

Укрупнена система моделей деталізує стратегічні концепції, зосереджуючись на основних бізнес-процесах та їх взаємозв'язках. Вона дозволяє глибше зрозуміти, як організація функціонує на рівні ключових операцій, і формує основу для подальшого детального моделювання.

Детальна система моделей охоплює всі аспекти діяльності організації, включаючи кожен бізнес-процес, його етапи та підпроцеси. Вона створюється для точної реалізації системи, враховуючи всі вимоги й нюанси організації.

Важливість бізнес-процесів полягає в тому, що вони визначають не тільки те, як організація працює, але й вимоги до її інформаційної системи. Це дозволяє уникнути розриву між бізнес-цілями і технічними вимогами до ІС, що часто стає причиною дорогих і трудомістких коригувань під час впровадження.

Опис діяльності організації на основі бізнес-процесів є більш об'єктивним, ніж ієрархічна функціональна структура, оскільки дозволяє точно оцінити динамічні характеристики процесів і їхній кінцевий результат. Кожен бізнес-процес має чітко визначені початок і кінець, послідовність виконання робіт, зовнішні та внутрішні інтерфейси, а також правила виконання робіт (бізнес-правила).

На кожному етапі створення інформаційної системи збираються дані, будується модель, яка використовується для визначення вимог до ІС. Вимоги до системи можуть бути адаптовані відповідно до змін у бізнес-процесах або умовах діяльності організації.

Опис бізнес-процесів також дозволяє забезпечити гнучкість інформаційної системи. Оскільки бізнес-процеси можуть змінюватися з часом, система повинна легко адаптуватися до цих змін. Завдяки використанню сучасних CASE-засобів і моделей, можна швидко внести необхідні зміни на будь-якому етапі життєвого циклу ІС.

Методологія, описана в цьому розділі, забезпечує узгодженість і гнучкість моделей інформаційної системи на всіх етапах її розробки. Кожна з моделей, побудованих на основі бізнес-процесів, узгоджується з іншими моделями, забезпечуючи єдність і цілісність системи. При цьому зберігається можливість

внесення змін на будь-якому етапі, що дозволяє швидко адаптувати систему до нових умов або вимог замовника. Ця гнучкість є однією з ключових переваг пропонованої методології, оскільки вона дозволяє створювати системи, які не тільки відповідають поточним потребам організації, але й можуть легко адаптуватися до її майбутніх змін. Це особливо важливо для корпоративних ІС, які мають тривалий життєвий цикл і повинні бути здатні розвиватися разом із бізнесом.

Сучасні CASE-засоби відіграють важливу роль у забезпеченні узгодженості моделей і гнучкості системи. Вони надають можливість автоматизованого проектування, тестування й документування моделей, що значно спрощує процес їхньої розробки та підтримки. Це дозволяє забезпечити високу якість ІС і мінімізувати ризики, пов'язані з помилками або недоліками в процесі її створення. Сучасні інструменти також забезпечують можливість швидкого прототипування та тестування окремих компонентів системи, що дозволяє оперативно вносити зміни на основі зворотного зв'язку від замовника або змін у бізнес-процесах.

2.2 Система моделей діяльності підприємства

Процес створення та розвитку систем узгоджених моделей, які описують діяльність організації, починається з формування стратегічної системи моделей організації. Ця система описує основні аспекти діяльності на стратегічному рівні. Основна мета стратегічної системи моделей полягає, по-перше, у визначенні ключових цілей та завдань організації, і по-друге, у формуванні моделей бізнес-процесів, що описують головні види діяльності і реалізують стратегічні цілі. На стратегічному рівні побудова моделей бізнес-процесів не передбачає врахування ієрархічної структури організації, яка описує "вертикальний" розподіл функцій і адміністративних меж між підрозділами. Бізнес-процеси визначають потоки робіт незалежно від ієрархії та меж підрозділів і організацій, що їх виконують, та відображають взаємодію як внутрішніх, так і зовнішніх бізнес-процесів.

Моделі, що входять до стратегічної системи, створюються під час дослідження організації шляхом опитування експертів вищого керівництва, яке відповідає за стратегію, та на основі ключових документів організації. Ці моделі зберігаються в репозиторії проєкту. Усі подальші моделі будуються на основі бізнес-процесів, виявлених під час аналізу діяльності на рівні підрозділів. Моделі створюються за допомогою CASE-засобів та зберігаються в репозиторії. Побудова цих моделей дозволяє паралельно виконувати роботи з дослідження та моделювання.

Подальший розвиток системи узгоджених моделей здійснюється на основі схеми трансформації моделей (табл.2.2), яка описує етапи розвитку за ключовими аспектами діяльності організації (дані, функції, люди, мережа, час, правила).

Таблиця 2.2 – Система узгоджених моделей діяльності підприємства

Групи характеристик моделей	Функції	Дані	Люди	Мережа	Час	Правила (Мотивація)
Згрупована система моделей організації	Список бізнес-процесів і бізнес-функцій організації	Списки документів, списки об'єктів	Структурна модель організації	Перелік структурних підрозділів і зовнішніх організацій	Список робіт у часі	Список цілей і завдань організації, критерії і правила виконання бізнес-процесів
Детальна система моделей організації	Функціональна модель підрозділів: діаграми потоків даних	Інформаційна модель організації (концептуальна модель)	Структурна модель підрозділів, ролі персоналу	Логічна модель мереж підрозділів організації та зовнішніх зв'язків	Тимчасова модель виконання бізнес-процесів і бізнес-функцій	Критерії виконання бізнес-функцій, бізнес-правила
Система моделей вимог до ІСУ	Вимоги до функцій, діаграми потоків даних	Вимоги до даних, структура даних, модель даних	Вимоги до структури інтерфейсу користувача	Вимоги до мережевої архітектури системи	Вимоги до часових характеристик функцій	Вимоги до регламенту роботи іс
Деталізація проєкту	Функції	Дані	Користувачі	Комунікації	Залежності	Правила

Кожний рядок таблиці відповідає певному рівню системи узгоджених моделей і деталізується для кожної групи характеристик. Переходи між рівнями

моделювання здійснюються згідно з методологією, що забезпечує повноту й узгодженість при побудові моделей.

Основною метою створення детальної системи моделей є побудова концептуальної моделі даних та функціональної моделі організації. Для цього проводиться деталізація опису діяльності організації від загальних бізнес-процесів до більш конкретних моделей на рівні підрозділів, що дозволяє виділити всі функції, оброблювані документи, основні дані, регламенти роботи персоналу, і створити функціональну модель організації та концептуальну модель даних.

Функції, виявлені під час дослідження, розподіляються по бізнес-процесах кожного підрозділу, доповнюючи та уточнюючи їх. У моделях деталізуються бізнес-процеси, функції, інформаційні потоки, вхідні та вихідні документи, взаємодії всередині організації та із зовнішніми об'єктами, а також бізнес-правила, ролі персоналу та їх регламент. Таке моделювання дозволяє визначити, де, коли і ким виконується кожна функція, які дані та зв'язки для цього потрібні, і звідки ці дані надходять.

Побудована система моделей може використовуватися для трьох основних цілей: по-перше, для формування вимог до інформаційної системи (ІС); по-друге, для аналізу діяльності організації з метою її вдосконалення через реінжиніринг; і по-третє, для створення стратегічного плану розвитку ІС на основі аналізу моделей та існуючих інформаційних систем.

Головною метою системи моделей вимог до ІС є забезпечення коректного переходу від моделей організації до моделей інформаційної системи, що описують її конкретні компоненти (додатки, бази даних, системне ПЗ, технічні та телекомунікаційні засоби). Цей перехід гарантує, що цілі та завдання організації будуть відображені в компонентах ІС. Для цього необхідно створити систему моделей вимог, які будуть пов'язані з проєктованими компонентами.

Система моделей вимог до ІС формується шляхом відображення моделі організації, побудованої на етапі дослідження. Цей процес відбувається через матрицю перетворення, визначену схемою моделювання. У результаті поступового уточнення формуються основні моделі вимог до ІС, включаючи вимоги до

архітектури, даних, інтерфейсу користувача, функціональності та управління системою.

Оскільки дані є основою діяльності будь-якої організації і змінюються значно рідше, ніж функції та структура, то під час проектування корпоративної ІС найбільш раціональним підходом є проектування, що базується на даних. Це забезпечує оптимальну архітектуру системи та спрощує інтеграцію додатків.

Процес проектування та розробки ПЗ базується на методології DATARUN, яка дозволяє швидко створювати переносні розподілені ІС у клієнт-серверній архітектурі. Методологія підтримує моделі, сформовані на основі бізнес-процесів та вимог до ІС, що дозволяє легко інтегрувати всі етапи розробки та забезпечити високу узгодженість модулів системи.

Пропонована методологія підтримується комплексом інструментальних засобів, які забезпечують безперервний цикл автоматизації процесів на всіх етапах життєвого циклу ІС. Узгодженість цих засобів досягається завдяки підтримці стандартів відкритих систем, що дозволяє створювати моделі, визначати вимоги до ІС, розробляти та тестувати додатки. Цей інструментарій також дозволяє вносити зміни на будь-якому етапі життєвого циклу ІС, що значно спрощує її розвиток та адаптацію.

Висновки до розділу 2

У другому розділі роботи розглянуто комплексну методологію створення корпоративної інформаційної системи (ІС), яка забезпечує високий рівень узгодженості між бізнес-процесами організації, вимогами до ІС та її архітектурою. Основною концептуальною основою запропонованого підходу є ітераційна спіральна модель життєвого циклу, що забезпечує гнучкість і адаптивність проектування на всіх етапах. У центрі уваги методології – створення системи узгоджених моделей, які є основою для формування вимог до ІС, розробки архітектури та побудови програмного забезпечення.

Запропонований підхід будується навколо ключових принципів:

1. Моделювання від бізнес-процесів: Початкові етапи аналізу ІС спрямовані на ідентифікацію та опис бізнес-процесів організації. Це дозволяє забезпечити точне відображення цілей і завдань підприємства у вимогах до інформаційної системи. Система моделей будується на трьох рівнях деталізації: стратегічному, укрупненому та детальному.

2. Проектування від даних: Дані розглядаються як ключовий елемент, що забезпечує стабільність і довговічність системи. Цей підхід дозволяє створювати оптимальну архітектуру, легко інтегрувати компоненти системи та забезпечувати їхню переносність.

3. Інтеграція на основі CASE-засобів: Використання сучасних CASE-інструментів сприяє автоматизації всіх етапів життєвого циклу ІС – від аналізу до впровадження. Це дозволяє забезпечити узгодженість між моделями, скоротити час на розробку та мінімізувати ризики помилок.

4. Гнучкість і адаптивність: Ітераційна спіральна модель життєвого циклу забезпечує можливість внесення змін на будь-якому етапі проектування чи впровадження системи. Це дозволяє адаптувати ІС до змін у бізнес-процесах, умовах функціонування чи вимогах замовника.

Методологія охоплює всі аспекти життєвого циклу ІС: аналіз, проектування, розробку, тестування, впровадження та супровід. Вона забезпечує формалізований підхід до побудови системи через інтеграцію різних рівнів моделювання – від стратегічних цілей організації до детальних функціональних і технічних описів.

Суттєвою перевагою підходу є підтримка узгодженості між бізнес-цілями організації та технічними характеристиками ІС. Це досягається через використання моделі бізнес-процесів як основи для побудови вимог до системи. Завдяки цьому мінімізується ризик невідповідності між завданнями підприємства та можливостями розробленої ІС.

РОЗДІЛ 3

РОЗРОБЛЕННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОЇ МОДЕЛІ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ СПІВРОБІТНИКА

3.1 Моделі предметної області в об'єктно-орієнтованому аналізі: принципи, методи та практичні аспекти

Модель предметної області широко використовується як основа для розробки програмних об'єктів [16]. Вона відображає основні класи понять (концептуальні класи) предметної області з точки зору того, хто моделює. Ключовим етапом об'єктно-орієнтованого аналізу є декомпозиція проблемної області на окремі концептуальні класи або об'єкти. Ця декомпозиція стає основою для створення моделей предметної області або об'єктних моделей аналізу [13].

Модель предметної області, представлена за допомогою уніфікованої мови моделювання (UML), зазвичай подається у вигляді діаграм класів, які не визначають жодних операцій. Ця модель може включати такі компоненти:

- об'єкти предметної області або концептуальні класи;
- асоціації між концептуальними класами;
- атрибути концептуальних класів.

Основна ідея полягає в тому, що модель предметної області є візуальним словником абстракцій. Інформацію, подану на діаграмі (у системі позначень UML), можна описати словесно або за допомогою термінів, проте візуальна мова є більш зручною для сприйняття, оскільки людський інтелект добре обробляє візуальну інформацію.

Отже, модель предметної області можна розглядати як візуальний словник важливих абстракцій або термінів предметної області. Це результат візуалізації понять реального світу у термінах конкретної предметної області, а не програмних елементів, таких як класи в Java чи C++. У моделі предметної області не використовуються такі елементи, як програмні артефакти (наприклад, вікна або

бази даних), якщо система, яка розробляється, не є програмним засобом, наприклад, моделлю графічного інтерфейсу користувача [27, 28, 32].

Модель предметної області ілюструє концептуальні класи або словник предметної області. Неформально, концептуальний клас – це подання ідеї або об'єкта, що включає:

- символи (symbol) – слова або образи, що відображають концептуальний клас.
- зміст (intension) – визначення концептуального класу.
- розширення (extension) – набір прикладів, до яких застосовується концептуальний клас.

Наприклад, для поширених економічних систем можна розглянути концептуальний клас, пов'язаний з подією купівлі. Його символ – це Sale. Зміст класу – це подія купівлі, яка відбувається у певний час і дату. Розширенням є всі випадки купівлі.

Під час розробки моделі предметної області, зазвичай, розглядаються як символічний опис, так і зміст концептуального класу, оскільки ці аспекти мають найбільше практичне значення.

Програмні системи можуть бути надзвичайно складними, тому декомпозиція за принципом "розділяй і володарюй" є загальною стратегією для вирішення складних проблем шляхом їх поділу на дрібніші частини. В об'єктно-орієнтованому аналізі декомпозиція фокусується на основних поняттях або сутностях предметної області.

Головна відмінність об'єктно-орієнтованого аналізу від структурного підходу полягає у тому, що він розбиває проблему на поняття або об'єкти, а не на функції. Тому основним завданням на етапі аналізу є ідентифікація ключових понять предметної області та їх подання у вигляді моделі [10, 17, 18, 22].

Наше завдання полягає у створенні моделі предметної області, яка буде відображати найбільш значущі поняття цієї області. Головне завдання – це ідентифікація концептуальних класів, що відповідають сценарію, який

розробляється. Краще на початкових етапах приділяти більше уваги деталізації моделі предметної області, ніж недооцінювати важливі елементи.

Прикладом може бути предметна область торгових систем або систем резервування авіаквитків (табл. 3.1).

Таблиця 3.1 – Список категорій концептуальних класів

Категорія концептуальних класів	Приклади
Фізичні чи матеріальні об'єкти	Register (Ресстр), Airplane (Літак)
Специфікації, елементи проектних рішень або опис об'єктів	ProductSpecification (Специфікація товару)
Місця	Store (Магазин), Airport (Аеропорт)
Транзакції	Sale (Продаж), Payment (Платіж), Reservation (Резервування)
Елементи транзакцій	SalesLineItem (Елемент продажу)
Ролі людей	Cashier (Касир), Pilot (Пілот)
Контейнери інших об'єктів	Store (Магазин), Bin (Бункер), Airplane (Літак)
Вміст контейнерів	Item (Елемент), Passenger (Пасажир)
Інші комп'ютери або електромеханічні системи, зовнішні до цієї системи.	CreditPaymentAuthorizationSystem (система авторизації кредитних платежів), AirTrafficControl (Система управління рухом)
Організації	SalesDepartment (Відділ продажу), ObjectAirline (Авіалінії)
Події	Sale (Продаж), Payment (Платіж), Meeting (Зустріч), Flight (Політ), Crash (Аварія), Landing (Приземлення)
Процеси (часто не видаються у вигляді понять)	SellingAProduct (Продаж продукту), BookingASeat (Бронювання місця)
Правила та політика	RefundPolicy (Правила повернення товару), Cancellation (Політика анулювання замовлення)

Часто деякі концептуальні класи ідентифікуються пізніше, на стадії розгляду атрибутів і асоціацій, або навіть під час проектування. Нові виявлені поняття додаються до моделі по мірі їх виникнення.

Не варто виключати концептуальні класи тільки тому, що на етапі аналізу не видно явної необхідності їх збереження (цей підхід підходить для реляційних баз даних, але не для моделей предметної області), або через те, що клас не має атрибутів [10, 22]. Концептуальні класи без атрибутів є допустимими, так само як і ті, що виконують "поведінкову", а не інформаційну роль у предметній області.

Ще один корисний метод для моделювання предметної області – це шаблони аналізу. Шаблони є готовими моделями предметної області, розробленими

експертами. Під час створення моделі предметної області доцільно скласти список кандидатів на роль концептуальних класів.

3.2 Процес розробки моделі предметної області: ідентифікація асоціацій та ролей

Модель предметної області – це візуалізація важливих понять із словника предметної області. Деякі з цих понять можуть бути представлені концептуальними класами, інші – як концептуальні класи, що не мають відношення до поточної ітерації (наприклад, "бухгалтерська система" або "комісійні"), а треті – як атрибути цих концептуальних класів. Недоліком цього підходу є багатозначність природної мови: для опису одного й того ж концептуального класу чи атрибута можуть використовуватись різні іменники, і водночас деякі іменники можуть мати кілька значень. Однак цей підхід рекомендується використовувати разом із методом вибору понять за списком категорій [8, 44].

У процесі розробки моделі предметної області необхідно ідентифікувати зв'язки (асоціації) між концептуальними класами, що відповідають інформаційним вимогам сценаріїв, що розробляються на поточній ітерації, а також виділити ті з них, які сприяють кращому розумінню моделі предметної області. Розглянемо процес визначення відповідних асоціацій.

Асоціація – це зв'язок між типами (або, точніше, екземплярами типів), який відображає певне значущє і корисне відношення між ними. Асоціації, які заслуговують на увагу, зазвичай містять знання про взаємозв'язок між об'єктами, які повинні зберігатися протягом певного періоду часу. Цей період може вимірюватися в мілісекундах або роках, залежно від конкретного контексту.

До моделі предметної області доцільно включати такі асоціації:

- Асоціації, знання про які слід зберігати протягом деякого періоду (важливі асоціації).
- Асоціації, похідні від стандартних асоціацій, що містяться в списку.

Якщо в моделі предметної області є N різних концептуальних класів, то між ними можна встановити $N * (N-1)$ асоціацій, що може призвести до великої кількості зв'язків. Тому при додаванні асоціацій слід дотримуватися принципу мінімалізму.

Рекомендації щодо призначення асоціацій:

- Основну увагу слід приділяти тим асоціаціям, знання про які потрібно зберігати протягом певного періоду часу (важливим асоціаціям).
- Занадто велика кількість асоціацій може призвести до помилок у моделі предметної галузі, а не до її спрощення. Вивчення асоціацій не повинно забирати занадто багато часу, але має приносити максимальну користь.
- Необхідно уникати відображення надлишкових асоціацій, які не мають самостійного значення.

Кожен кінець асоціації має назву ролі. Роль може мати такі характеристики:

- Ім'я.
- Кратність.
- Напрямок зв'язку.

У цьому розділі буде розглянута кратність, а дві інші характеристики асоціації будуть обговорюватися в наступних розділах.

Кратність визначає, скільки екземплярів класу A може бути асоційовано з одним екземпляром класу B . Значення кратності вказує на кількість екземплярів одного класу, які можуть бути коректно пов'язані з екземпляром іншого класу в певний момент часу, а не на всьому проміжку часу. Імена асоціацій зазвичай базуються на форматі: «Ім'яТипу-ДієслівнаФраза-Ім'яТипу», де дієслівна фраза є значущою в контексті моделі.

Імена асоціацій повинні починатися з великої літери, оскільки асоціація зазвичай є класифікатором зв'язків між екземплярами. В UML ім'я класифікатора також починається з великої літери. Для імен асоціацій використовуються два формати:

- Paid-by (з дефісом).
- PaidBy (без дефісу).

При читанні імен асоціацій зазвичай використовують напрямок зліва направо та зверху вниз.

На стадії створення моделі предметної області асоціація вже не є описом потоків даних, екземплярів змінних чи взаємодії об'єктів програмної системи. Вона є лише описом значущого відношення і має суто концептуальний зміст, який походить із реального світу. На практиці багато таких асоціацій зазвичай реалізуються у програмному забезпеченні як методи переміщення та засоби забезпечення видимості, однак їх присутність у концептуальному поданні моделі предметної області не вимагає реалізації [30, 41].

Під час створення моделі предметної області можна визначити асоціації, які будуть необхідні під час проектування. І навпаки, можна виявити асоціації, які потрібно реалізувати, але на стадії моделювання предметної області ще не було визначено. У такому разі модель предметної області слід оновити, щоб відобразити зміни.

3.3 Побудова об'єктної моделі з використанням мови UML

Уніфікована мова моделювання (UML) є стандартом для створення «креслень» програмного забезпечення. Вона дозволяє візуалізувати, специфікувати, конструювати та документувати артефакти програмних систем. UML підходить для моделювання будь-яких систем: від інформаційних систем підприємства до розподілених веб-додатків і вбудованих систем реального часу. Це виразна мова, яка дає змогу розглядати систему з різних точок зору, пов'язаних із її розробкою та подальшим впровадженням. Попри велику кількість можливостей, UML залишається простою для розуміння та використання. Опанувати UML найкраще через вивчення його концептуальної моделі, яка складається з трьох основних елементів: базових будівельних блоків, правил, що визначають, як ці блоки можуть поєднуватися між собою, і загальних механізмів мови [7, 28].

Попри всі переваги, UML є лише мовою і складовою частиною процесу розробки програмного забезпечення. Найкраще її використовувати, коли процес моделювання базується на аналізі прецедентів використання, є ітеративним і покроковим, а сама система має чітко виражену архітектуру.

UML є мовою для візуалізації, специфікації, конструювання та документування артефактів програмних систем. Мова включає словник і правила, що дозволяють комбінувати її елементи, створюючи осмислені конструкції. У мові моделювання словник і правила орієнтовані на концептуальне та фізичне уявлення системи.

Моделювання є важливим етапом розуміння системи. Однак єдиної моделі для цього недостатньо. Для розуміння складних систем потрібно створювати багато взаємозалежних моделей. Відповідно до програмних систем, це означає, що потрібна мова, яка дасть змогу описати архітектуру системи з різних точок зору на всіх етапах її розробки. Визначення, які моделі необхідно створювати і коли, є частиною процесу розробки програмного забезпечення. Добре організований процес підказує, які артефакти потрібно створити, які ресурси для цього необхідні, і як їх використовувати для оцінки виконаної роботи та управління проектом в цілому.

З погляду багатьох програмістів, роздуми про реалізацію проекту майже рівнозначні написанню коду. Однак навіть у таких випадках програміст займається моделюванням, хоч і неформально. Наприклад, він може записати на дошці або серветці малюнок ідеї. Такий підхід має свої недоліки. По-перше, ефективна комунікація щодо концептуальної моделі можлива лише тоді, коли всі учасники розмовляють однією мовою. Зазвичай компанії створюють власні мови для розробки, і новачкові важко розібратися в них. По-друге, неможливо повністю зрозуміти певні аспекти програмної системи без моделі, яка виходить за межі тексту коду. Так, ієрархію класів можна зрозуміти, якщо уважно вивчити код кожного класу, але всю структуру відразу не сприйняти. По-третє, якщо автор коду не зафіксував уявлення про моделі в явному вигляді, ця інформація буде втрачена при зміні місця роботи. Вона може бути частково відновлена лише з реалізації.

Використання UML вирішує третю проблему, оскільки явна модель полегшує комунікацію. Деякі аспекти системи краще моделювати за допомогою тексту, а інші – графічно.

UML – це не просто набір графічних символів, за кожним із яких стоїть чітка семантика. Це дозволяє однозначно інтерпретувати модель, створену одним розробником, іншим або навіть інструментальними програмами, що вирішує першу проблему. UML є мовою специфікації. Це означає, що вона дозволяє будувати точні, однозначні та повні моделі. За допомогою UML можна специфікувати всі важливі рішення щодо аналізу, проектування та реалізації, які необхідно прийняти на етапах розробки і розгортання програмного забезпечення [7, 28]. UML – це також мова конструювання. Хоча вона не є мовою візуального програмування, моделі, створені за допомогою UML, можуть бути безпосередньо переведені на різні мови програмування, а також на таблиці реляційних баз даних або об'єкти об'єктно-орієнтованих баз даних. UML дає змогу генерувати код з моделі в конкретну мову програмування, а також здійснювати зворотне проектування – відновлення моделі з існуючої реалізації. Це дозволяє працювати як у графічному, так і в текстовому форматі, забезпечуючи узгодженість між обома підходами, якщо програмні засоби це підтримують.

Крім того, UML дозволяє виконувати моделі, імітувати поведінку систем і контролювати діючі системи. UML – це також мова документування. Компанії, що випускають програмне забезпечення, створюють не лише виконуваний код, а й інші артефакти, серед яких:

- вимоги до системи;
- архітектура;
- проект;
- вихідний код;
- проектні плани;
- випробування;
- прототипи;
- версії тощо.

В залежності від методології розробки, деякі роботи виконуються більш формально, ніж інші. Ці артефакти необхідні для управління процесом розробки, оцінки результату та як засіб комунікації між членами команди на етапах розробки та після розгортання системи. UML дозволяє вирішити проблему документування архітектури системи та її деталей, надаючи мову для формулювання вимог і визначення тестів, а також інструменти для моделювання етапів планування проекту та управління версіями.

Для розуміння UML необхідно вивчити його концептуальну модель, яка складається з трьох основних складових: основних будівельних блоків мови, правил їх поєднання та деяких загальних для всієї мови механізмів. Ці елементи не є складними, і з часом, з набуттям досвіду, ви зможете використовувати і більш складні можливості мови.

Словник мови UML містить три види будівельних блоків:

- сутності;
- відносини;
- діаграми.

Сутності – це абстракції, які є основними елементами моделі. Відносини з'єднують різні сутності, а діаграми групують сутності, що представляють інтерес [19, 21].

Сутності є основними об'єктно-орієнтованими блоками мови UML, з допомогою яких створюються коректні моделі. Структурні сутності – це іменники в моделях UML, зазвичай статичні частини моделі, що відповідають концептуальним чи фізичним елементам системи.

Асоціація (Association) – це структурне відношення, яке описує сукупність зв'язків між об'єктами. Виглядає як лінія, що може містити стрілку чи мітку, а також додаткові позначення, наприклад, кратність та імена ролей.

Діаграма UML – це графічне представлення елементів системи, що найчастіше зображуються у вигляді графа з вершинами (сутностями) і ребрами (стосунками). Діаграми використовуються для візуалізації системи з різних точок зору.

Однією з головних особливостей UML є його здатність створювати різноманітні моделі, кожна з яких надає певний погляд на систему з певної точки зору. Мова UML дозволяє виділяти різні аспекти програмної системи за допомогою діаграм, що відповідають за опис різних елементів і взаємодій в системі. Існують кілька основних типів діаграм, які поділяються на два великих класи: структурні та поведінкові діаграми.

Структурні діаграми фокусуються на статичних аспектах системи, таких як:

Діаграма класів (Class Diagram): представляє класи та їхні зв'язки, а також атрибути і методи класів. Це основний інструмент для моделювання об'єктно-орієнтованих систем.

Діаграма об'єктів (Object Diagram): показує приклади об'єктів класів і їхні зв'язки на певний момент часу.

Діаграма компонентів (Component Diagram): відображає фізичну структуру системи на рівні компонентів і їхніх взаємодій.

Діаграма розгортання (Deployment Diagram): моделює фізичну архітектуру системи, зокрема її компоненти, що виконуються на конкретних вузлах.

Поведінкові діаграми описують динамічні аспекти системи, такі як:

Діаграма прецедентів (Use Case Diagram): дає змогу описати взаємодію користувача (або іншої зовнішньої системи) з програмною системою. Вона показує, які функції система повинна виконувати.

Діаграма активностей (Activity Diagram): описує потік роботи або алгоритм дій в межах системи.

Діаграма послідовностей (Sequence Diagram): моделює взаємодії між об'єктами або компонентами, що відбуваються у часі.

Діаграма станів (State Diagram): відображає різні стани об'єкта в межах системи та переходи між ними.

Діаграма комунікацій (Communication Diagram): схожа на діаграму послідовностей, але з акцентом на зв'язки між об'єктами.

Моделі в UML надають чітку структуру для розуміння і документування системи. Кожен тип діаграми може бути використаний для конкретної мети: чи це

побудова базової структури, чи моделювання динаміки системи. У UML ключову роль відіграє етап специфікації моделі, на якому задаються всі необхідні вимоги до системи. Специфікація може бути частиною документації, яка визначає деталі, структуру і поведінку системи. В UML специфікація створюється за допомогою діаграм, таких як діаграми класів або діаграми прецедентів. Проте на цьому етапі вона може бути ще досить абстрактною, описуючи систему на високому рівні.

Після етапу специфікації настає етап реалізації системи. Реалізація є конкретизацією специфікацій і створенням моделей, які можуть бути трансформовані в робочий код. В UML реалізація може включати діаграми класів, що відповідають реальній структурі коду, або діаграми компонентів, що відображають структуру програмного продукту на фізичному рівні. Це дозволяє здійснити зворотне проектування та забезпечити узгодженість між моделями та реальним кодом.

У ході реалізації UML може бути використаний для генерації коду, що значно прискорює розробку, особливо коли необхідно трансформувати специфікації в конкретні мови програмування або бази даних. У деяких інструментах для UML (наприклад, Rational Rose або Enterprise Architect) є можливості автоматичної генерації коду з моделей.

Мова UML надає багато переваг у процесі розробки програмного забезпечення. Основні з них:

- можливість точного і детального опису системи та її компонентів;
- уніфікація процесу розробки через використання стандартів для представлення проекту;
- забезпечення зворотного зв'язку між командами розробників, аналітиками та замовниками через чітке візуальне подання моделі;
- моделювання складних взаємодій між компонентами та їх поведінки на різних етапах роботи системи.

У кінцевому підсумку, UML допомагає зменшити кількість помилок, підвищити якість програмного забезпечення та полегшити комунікацію серед

членів команди. Тому в багатьох великих проектах UML є основним інструментом для документації та управління розвитком програмного забезпечення.

3.4 Розроблення об'єктно-орієнтованої моделі інформаційної системи

Розглянемо інформаційну систему підприємства, яка займається управлінням працівниками, проектами та ресурсами. У цій моделі буде представлено 4 класи і відповідні об'єкти. Діаграма класів зображена на рис.3.1.

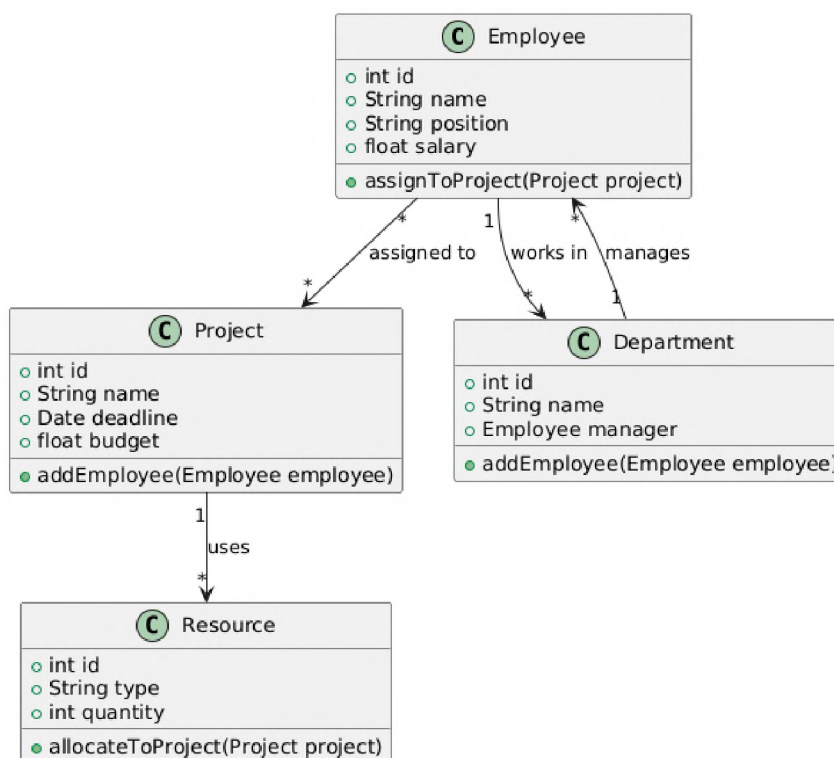


Рисунок 3.1 – Діаграма класів ІС

Класи моделі наступні:

1. Employee (Працівник)

Атрибути:

- id: int – Унікальний ідентифікатор працівника.
- name: str – Ім'я працівника.
- position: str – Посада працівника.

- salary: float – Заробітна плата.

Методи:

- assign_to_project(project: Project) – Призначення на проект.

2. Project (Проект)

Атрибути:

- id: int – Унікальний ідентифікатор проекту.
- name: str – Назва проекту.
- deadline: date – Кінцевий термін проекту.
- budget: float – Бюджет проекту.

Методи:

- add_employee(employee: Employee) – Додавання працівника до проекту.

3. Resource (Ресурс)

Атрибути:

- id: int – Унікальний ідентифікатор ресурсу.
- type: str – Тип ресурсу (наприклад, техніка, матеріали).
- quantity: int – Кількість ресурсу.

Методи:

- allocate_to_project(project: Project) – Виділення ресурсу для проекту.

4. Department (Відділ)

Атрибути:

- id: int – Унікальний ідентифікатор відділу.
- name: str – Назва відділу.
- manager: Employee – Менеджер відділу.

Методи:

- add_employee(employee: Employee) – Додавання працівника до відділу.

Діаграма об'єктів моделі зображена на рис. 3.2. Вона демонструє ключові об'єкти системи, їх атрибути та взаємозв'язки, які можна зобразити у вигляді UML-діаграми класів та об'єктів.

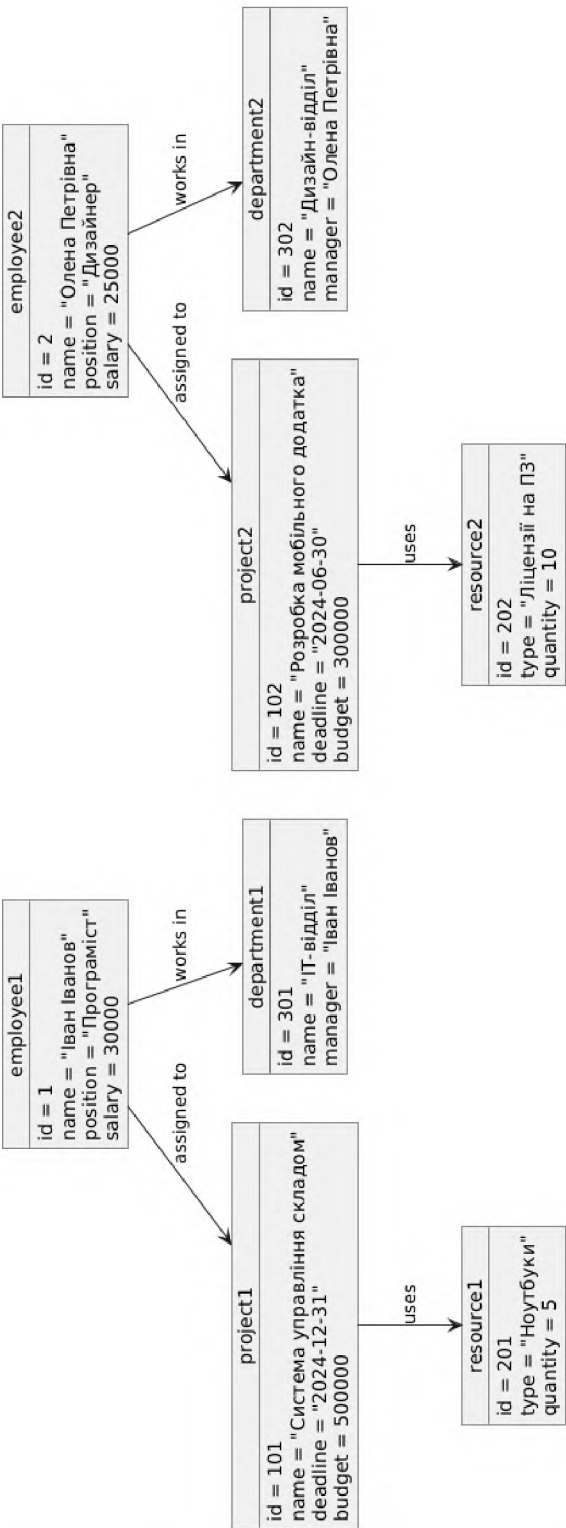


Рисунок 3.2 – Діаграма об’єктів ІС

Об’єкти моделі ІС підприємства наступні:

1. Employee Objects:

- employee1: Employee – Петро Іванов, програміст, зарплата 30,000 грн.
- employee2: Employee – Олена Петрівна, дизайнер, зарплата 25,000 грн.

2. Project Objects:

– project1: Project – Система управління складом, дедлайн – 2024-12-31, бюджет – 500,000 грн.

– project2: Project – Розробка мобільного додатка, дедлайн – 2024-06-30, бюджет – 300,000 грн.

3. Resource Objects:

– resource1: Resource – Ноутбуки, кількість – 5.

– resource2: Resource – Ліцензії на ПЗ, кількість – 10.

4. Department Objects:

– department1: Department – IT-відділ, менеджер – Іван Іванов.

– department2: Department – Дизайн-відділ, менеджер – Олена Петрівна.

Взаємозв'язки:

1. employee1 працює у department1 та призначений до project1.

2. employee2 працює у department2 та призначений до project2.

3. resource1 виділено для project1.

4. resource2 виділено для project2.

На рисунку 3.3 показана діаграма варіантів використання об'єктної моделі ІС підприємства. Діаграма ілюструє взаємодію акторів із системою через основні варіанти використання. Ці сценарії забезпечують основні функції системи, відповідаючи її об'єктній моделі. Розгорнутий опис варіантів використання наступний:

1. Створення нового проекту.

ID: UC_CreateProject.

Актор: Керівник відділу (Manager).

Опис: Керівник створює новий проект у системі, задаючи його назву, дедлайн і бюджет.

Цей проект буде використовуватися для планування робіт і призначення ресурсів.

Основний сценарій:

1. Керівник входить у систему.

2. Вибирає опцію "Створити новий проект".
 3. Вводить дані про проект: назва, дедлайн, бюджет.
 4. Система перевіряє коректність даних (дата не може бути в минулому, бюджет має бути більше нуля).
 5. Система додає проект до бази даних і підтверджує створення.
- Альтернативні сценарії: 3а. Якщо дані введені некоректно, система виводить повідомлення про помилку, і керівник повторює введення.



Рисунок 3.3 – Діаграма варіантів використання ІС

2. Призначення працівників на проект.
- ID: UC_AssignEmployees.
- Актор: Керівник відділу (Manager).
- Опис: Керівник вибирає проект і додає до нього працівників, які будуть виконувати завдання.

Основний сценарій:

1. Керівник входить у систему.
2. Вибирає проект зі списку.
3. Вибирає працівників зі списку доступних.
4. Система додає працівників до проекту і зберігає зміни.

Альтернативні сценарії: 3а. Якщо працівник вже призначений на інший проект у цей час, система повідомляє про конфлікт і пропонує вибрати іншого працівника.

3. Виділення ресурсів для проекту.

ID: UC_AllocateResources.

Актор: Адміністратор системи (Admin).

Опис: Адміністратор закріплює ресурси (обладнання, матеріали) за проектом.

Основний сценарій:

1. Адміністратор входить у систему.
2. Вибирає проект зі списку.
3. Вибирає ресурси зі списку доступних.
4. Вказує кількість кожного ресурсу.
5. Система зберігає дані про виділення ресурсів.

Альтернативні сценарії: 4а. Якщо кількість ресурсу недостатня, система повідомляє про помилку і пропонує змінити параметри.

4. Перегляд списку проектів.

ID: UC_ViewProjects.

Актори: Керівник відділу (Manager), Працівник (Employee).

Опис: Керівник або працівник переглядає інформацію про проекти, до яких має доступ.

Основний сценарій:

1. Актор входить у систему.
2. Вибирає опцію "Перегляд проектів".
3. Система виводить список проектів і основну інформацію (назва, дедлайн, статус).

4. Актор вибирає проект для перегляду детальної інформації.

5. Перегляд інформації про працівників.

ID: UC_ViewEmployees.

Актори: Керівник відділу (Manager), Працівник (Employee).

Опис: Керівник або працівник переглядає дані про працівників у відділі.

Основний сценарій:

1. Актор входить у систему.
2. Вибирає опцію "Перегляд працівників".
3. Система виводить список працівників і їхні основні дані (ім'я, посада, проект).
4. Актор вибирає працівника для перегляду додаткової інформації (зарплата, контакти).
6. Оновлення даних працівників.

ID: UC_UpdateEmployee.

Актор: Адміністратор системи (Admin).

Опис: Адміністратор оновлює дані про працівників (зміна посади, зарплати тощо).

Основний сценарій:

1. Адміністратор входить у систему.
2. Вибирає працівника зі списку.
3. Оновлює дані (посада, зарплата, відділ).
4. Система перевіряє введені дані і зберігає зміни.

Альтернативні сценарії: За. Якщо дані некоректні, система повідомляє про помилку.

Лінійна діаграма активностей, яка демонструє основні кроки процесу "Створення нового проекту" показана на рис. 3.4. Опис діаграми активностей "Створення нового проекту" наступний:

1. Вхід у систему: Керівник відділу входить у систему.
2. Вибір опції "Створити новий проект": Керівник обирає опцію для створення нового проекту.
3. Введення даних про проект: Керівник вводить необхідні дані про проект, такі як назва, дедлайн, бюджет.
4. Перевірка даних: Перевіряється коректність введених даних (наприклад, дата не повинна бути в минулому, бюджет повинен бути додатним).

5. Рішення: Якщо дані коректні, проект створюється і система підтверджує це.

6. Альтернативний шлях: Якщо дані некоректні, система повідомляє про помилку, і керівник виправляє введені дані.

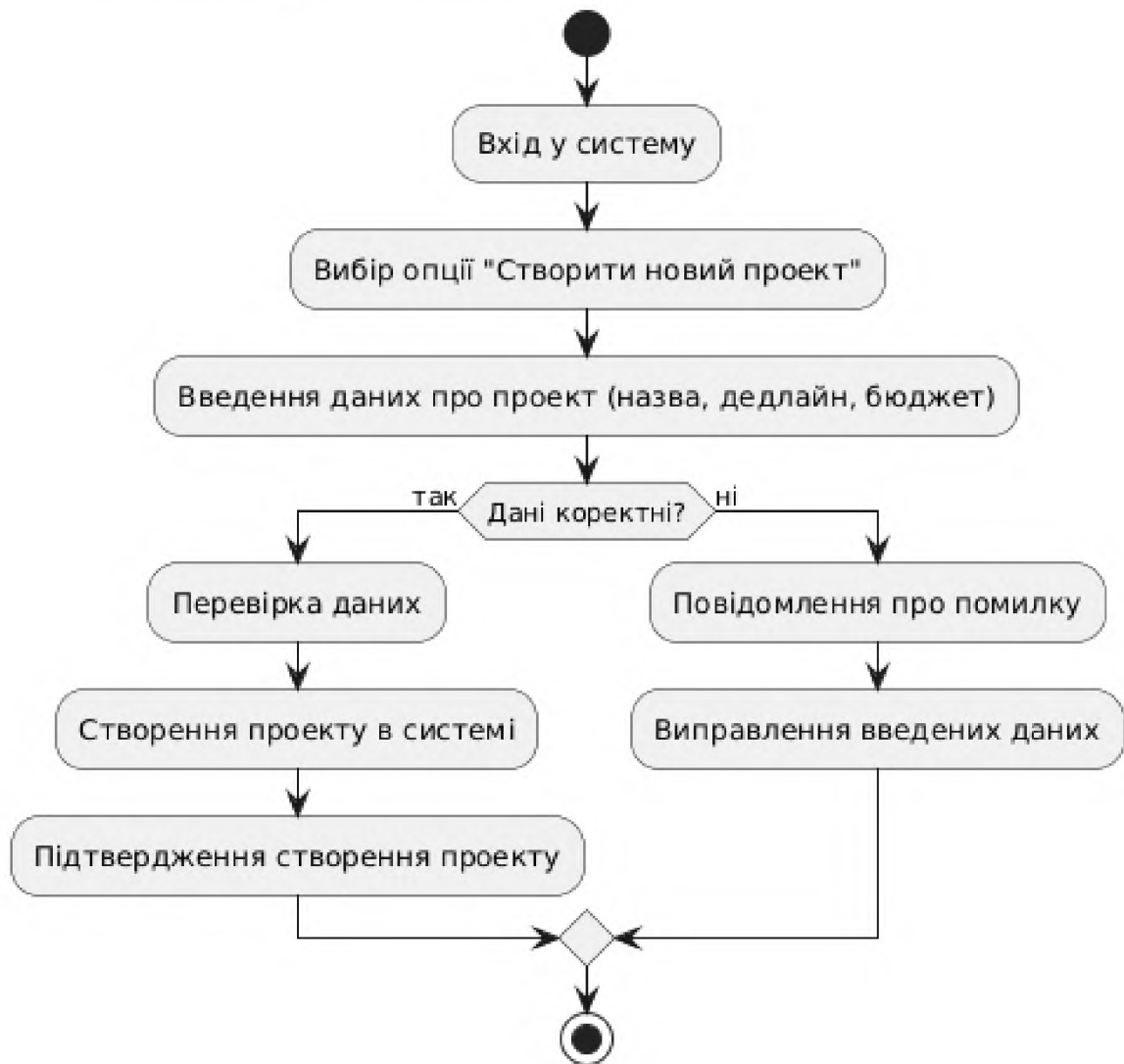


Рисунок 3.4 – Діаграма активностей процесу "Створення нового проекту"

Діаграма активностей на рис. 3.5 показує основний процес "Призначення працівників на проект" із перевіркою на конфлікти та підтвердженням дій.

Опис діаграми активностей на рис. 3.5:

1. Вхід у систему: Керівник відділу входить у систему.
2. Вибір проекту для призначення працівників: Керівник обирає проект, на який потрібно призначити працівників.

3. Перегляд списку працівників: Керівник переглядає доступних працівників.
4. Вибір працівника для призначення: Керівник вибирає працівника для призначення на проект.

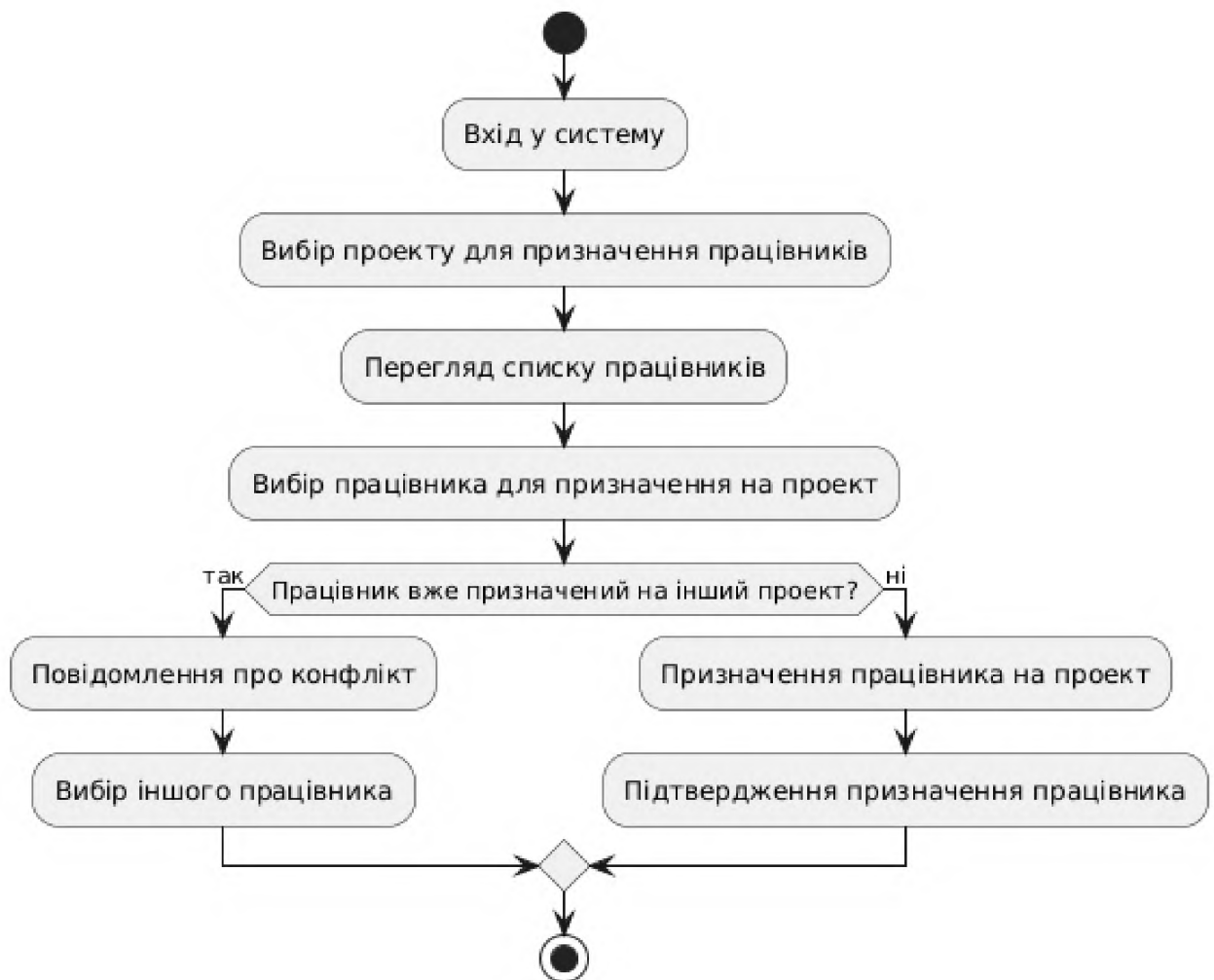


Рисунок 3.5 – Діаграма активностей процесу "Призначення працівників на проект"

5. Перевірка на наявність призначень: Якщо працівник уже призначений на інший проект у той самий час, система повідомляє про конфлікт.
6. Альтернативний шлях: Якщо працівник не призначений на інший проект, керівник підтверджує призначення.
7. Призначення працівника на проект: Якщо все правильно, система записує зміни, і працівник додається до проекту.

Діаграма послідовності на рис. 3.6 показує послідовність взаємодій між адміністратором, системою, проектом і ресурсами під час процесу виділення ресурсів.

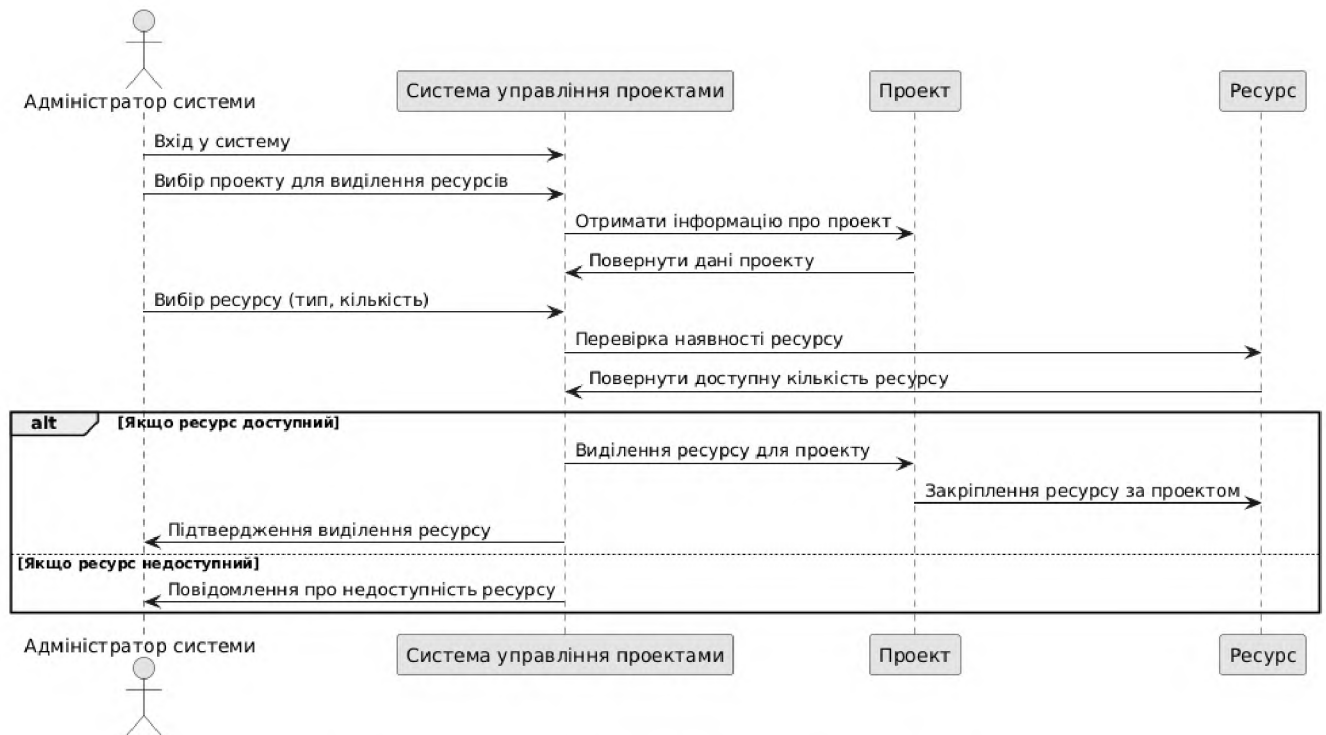


Рисунок 3.6 – Діаграма послідовності ІС

Опис діаграми послідовності на рис. 3.6:

1. Адміністратор системи входить у систему.
2. Вибір проекту для виділення ресурсів: Адміністратор вибирає проект, для якого необхідно виділити ресурси.
3. Отримання інформації про проект: Система отримує інформацію про вибраний проект.
4. Вибір ресурсу: Адміністратор вибирає ресурс (тип і кількість) для проекту.
5. Перевірка наявності ресурсу: Система перевіряє наявність вибраного ресурсу.
6. Якщо ресурс доступний: Система виділяє ресурс проекту, закріплюючи його за проектом.

7. Якщо ресурс недоступний: Система повідомляє адміністратору про відсутність ресурсу.

8. Підтвердження виділення ресурсу: Якщо ресурс доступний, система підтверджує виділення ресурсу проекту.

Діаграма станів (рис. 3.7) показує життєвий цикл проекту від початкової стадії до його завершення.



Рисунок 3.7 – Діаграма станів ІС

Опис діаграми станів на рис. 3.7:

1. Стан "Чернетка" (Draft). Це початковий стан проекту. Проект знаходиться на етапі підготовки, і ще не розпочато його виконання. Вхід у цей стан може бути після створення проекту у системі.

2. Перехід з "Чернетка" в "У процесі" (In Progress). Проект переходить в цей стан, коли починається його виконання. У цьому стані проект активно виконується: ресурси виділені, працівники працюють, а задачі вирішуються. Перехід до цього стану ініціюється керівником або автоматично після початку роботи над проектом.

3. Перехід з "У процесі" в "Завершено" (Completed). Коли всі основні завдання проекту виконані і проект досяг своєї мети, він переходить в стан "Завершено". Це може означати, що проект успішно завершений, але ще не закритий в системі.

4. Перехід з "Завершено" в "Закрито" (Closed). Коли проект офіційно закривається після завершення всіх етапів і завдань, він переходить у стан "Закрито". У цьому стані проект більше не активний і всі дії з ним завершені (наприклад, всі фінансові звіти та оцінки виконані).

5. Початковий і кінцевий стани. Початковий стан позначений як [*], що вказує на те, що проект розпочинається з "Чернетка". Кінцевий стан позначений як Закрито, що є остаточним станом проекту.

Опис змін станів:

- Проект починається в стані "Чернетка" після створення.
- Після початку робіт проект переходить в "У процесі", де триває виконання завдань.
- Коли всі завдання виконано, проект переміщається в "Завершено", після чого, коли проект офіційно закривають, він переходить в "Закрито".

Діаграма розгортання на рис. 3.8 показує архітектуру системи, де основні компоненти (сервери додатків та бази даних) взаємодіють з клієнтськими пристроями для різних користувачів. Опис діаграми розгортання наступний:

1. Сервер додатків містить компонент "Додаток для управління проектами", на якому розгортаються всі бізнес-логіка та функціональність системи.

Він відповідає за обробку запитів від клієнтських додатків.



Рисунок 3.8 – Діаграма розгортання ІС

2. Сервер бази даних містить "Базу даних проектів". Ця база зберігає всю інформацію про проекти, працівників, ресурси та інші важливі дані.

3. Клієнтські пристрої:

– Менеджер: Клієнтський додаток для менеджера, який використовує систему для управління проектами.

– Працівник: Клієнтський додаток для працівника, який взаємодіє з проектами в системі, отримує завдання тощо.

– Адміністратор: Клієнтський додаток для адміністратора, який має доступ до налаштувань, керує користувачами, проектами та ресурсами.

4. Взаємодія між компонентами наступна: клієнтські пристрої (для менеджера, працівника та адміністратора) надсилають запити на сервер додатків для взаємодії з системою; сервер додатків звертається до бази даних для отримання чи запису інформації.

Діаграма компонентів ІС на рис. 3.9 показує логічну структуру основних компонентів системи та їх взаємозв'язки. Опис діаграми компонентів наступний:

1. Пакет "Управління проектами":

– Модуль управління проектами: Основний модуль для управління проектами. Він дозволяє створювати, редагувати та моніторити проекти.

– Модуль планування ресурсів: Модуль, який займається плануванням і виділенням необхідних ресурсів для проекту.

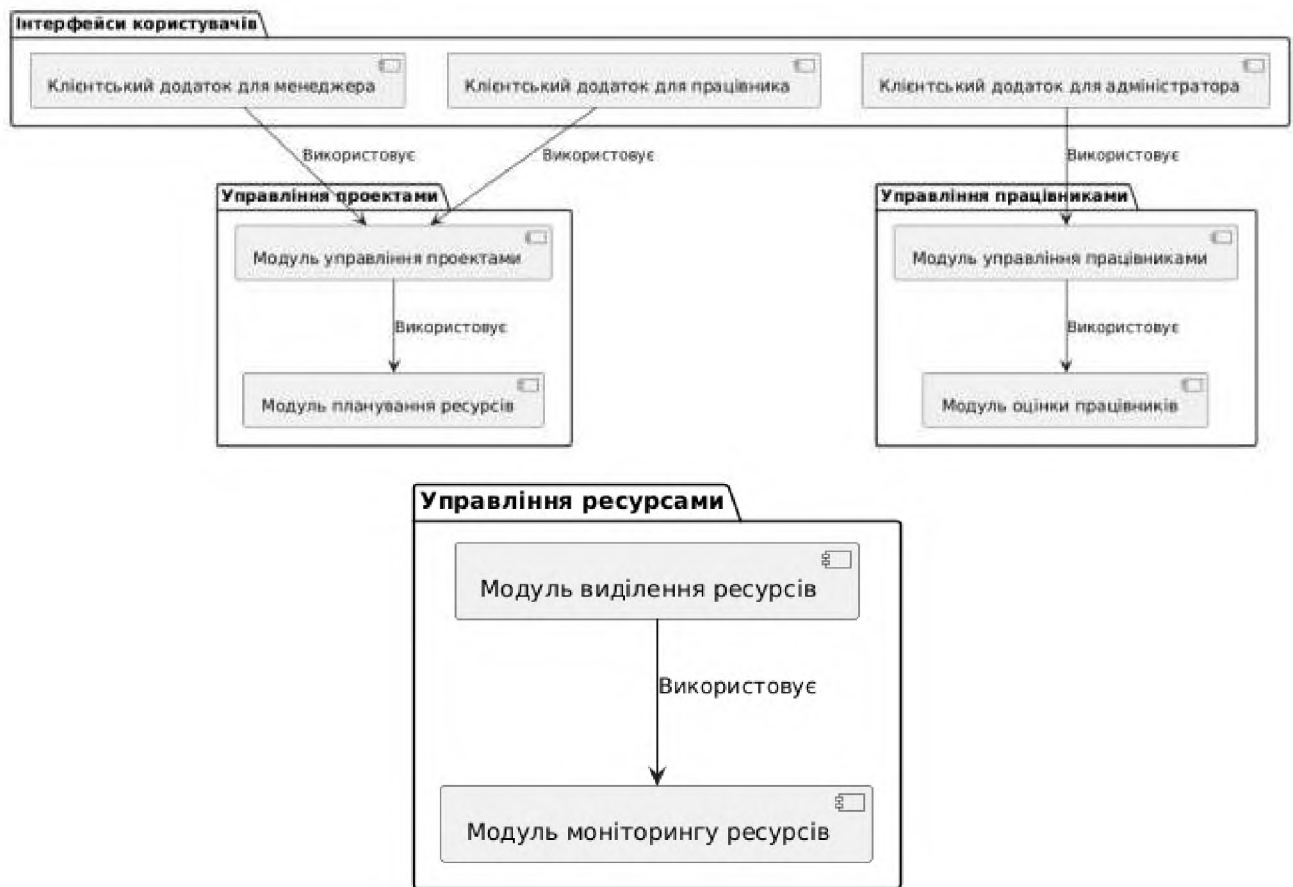


Рисунок 3.9 – Діаграма компонентів ІС

2. Пакет "Управління ресурсами":

– Модуль виділення ресурсів: Модуль, який дозволяє виділяти ресурси для проекту, такі як матеріали, обладнання, час.

– Модуль моніторингу ресурсів: Модуль для моніторингу та управління використанням ресурсів, забезпечує ефективність та оптимізацію.

3. Пакет "Інтерфейси користувачів":

– Клієнтський додаток для менеджера: Інтерфейс для менеджера проектів, який взаємодіє з усіма модулями для управління проектами та ресурсами.

– Клієнтський додаток для працівника: Інтерфейс для працівників, який дозволяє їм переглядати завдання, ресурси та прогрес проектів.

– Клієнтський додаток для адміністратора: Інтерфейс для адміністратора для управління користувачами, проектами та загальними налаштуваннями системи.

Взаємодії між компонентами:

– Модуль управління проектами використовує Модуль планування ресурсів для того, щоб забезпечити всі необхідні ресурси для проекту.

– Модуль управління працівниками використовує Модуль оцінки працівників для того, щоб мати інформацію про ефективність роботи кожного працівника.

– Модуль виділення ресурсів залежить від Модуля моніторингу ресурсів, щоб гарантовано забезпечити необхідні ресурси.

– Клієнтські додатки для менеджера, працівника і адміністратора використовують відповідні модулі для виконання своїх завдань:

– Менеджер працює з модулем управління проектами.

– Працівник має доступ до модуля управління проектами, щоб отримувати завдання.

– Адміністратор використовує модуль управління працівниками для налаштування та контролю.

Таким чином, проаналізовано структуру та функціональність інформаційної системи підприємства, що займається управлінням працівниками, проектами та ресурсами. Було розроблено чотири ключові класи: Employee, Project, Resource та Department, кожен із яких описує специфічні аспекти діяльності системи. На основі цих класів створено об'єкти, які демонструють конкретні приклади використання, зокрема працівників, проекти та виділені ресурси.

Діаграми UML (класів, об'єктів, активностей, станів, послідовностей, компонентів і розгортання) забезпечують візуалізацію основних елементів системи, їх взаємозв'язків та динамічних процесів. Особливу увагу приділено автоматизації процесів, таких як призначення працівників на проекти, виділення ресурсів і перегляд проектів, що покращує ефективність управління.

Запропонована модель забезпечує логічну узгодженість між компонентами, інтеграцію даних і гнучкість системи для подальшого розвитку, що підтверджує її

ефективність і відповідність сучасним вимогам підприємств. Розроблені діаграми UML дозволяють спростити оцінку та представлення процесів проектування, розгортання та супроводу ІС та її компонентів.

3.5 Обґрунтування методики та оцінка економічної ефективності

Аналіз основних показників порівняльної ефективності створення та функціонування інформаційних систем включає визначення таких показників:

1. річна економія поточних витрат, отримана від функціонування системи;
2. додаткові капітальні вкладення (КВ), необхідні для створення системи;
3. термін окупності додаткових КВ;
4. розрахунковий коефіцієнт ефективності додаткових КВ;
5. річний економічний ефект;
6. річна економія трудових витрат на обробку даних у системі.

Оцінимо ці показники:

1. Річна економія поточних витрат складається з прямої економії та інших складових. Пряма економія виникає завдяки автоматизації обробки інформації:

$$\Delta C_{\text{т}} = \Delta C_{\text{п}} + \Delta C_{\text{н}}, \quad (3.1)$$

де $\Delta C_{\text{п}}$ – пряма економія,

$\Delta C_{\text{н}}$ – непряма економія (не розраховується в межах цієї роботи).

Пряма економія обчислюється за формулою:

$$\Delta C_{\text{п}} = \Delta C_{\text{б}} - \Delta C_{\text{пор}}, \quad (3.2)$$

де $\Delta C_{\text{б}}$ – базовий період (до застосування системи),

$\Delta C_{\text{пор}}$ – порівнюваний період (система в автоматизованому режимі).

Розрахунок показника за порівнюваний період:

$$\Delta C_{\text{пор}} = C_1 + C_2 + C_3 + C_4 + C_5 + C_{\text{спр}}, \quad (3.3)$$

де C_1 – витрати на оплату праці персоналу;

C_2 – нарахування на фонд оплати праці (22%);

C_3 – витрати на сировину і матеріали (картриджі, папір, комплектуючі);

C4 – амортизація обладнання (лінійна, термін служби – 5 років);

C5 – інші витрати (відрядження, інформаційні витрати, плата за кредит, податки);

Спр – передвиробничі витрати (розробка ПЗ, навчання персоналу).

Оновлені дані представлені в таблиці 3.2.

Таблиця 3.2 – Результати розрахунку базових економічних показників

Затрати	Базовий період (ΔC_b), грн.	Порівнюваний період ($\Delta C_{пор}$), грн.
C1	480000	350000
C2	105600	77000
C3	1500	3000
C4	0	1500
C5	500	500
Спр	–	1500
Всього	587600	433500

$$\Delta C_{п} = 587600 - 433500 = 154100 \text{ грн.}$$

$$\Delta C_{т} = 154100 \text{ грн.}$$

2. Додаткові капітальні вкладення

– Оренда приміщення (на рік) = $5000 \text{ грн} \cdot 60 \text{ м}^2 = 300000 \text{ грн.}$

– Придбання комп'ютера (Ryzen 5, 16 ГБ ОЗУ, SSD 500 ГБ) = 18000 грн.

– Придбання монітора (23" LG FullHD) = 4500 грн.

– Придбання принтера (HP LaserJet Pro) = 5000 грн.

Сума капітальних вкладень за рік:

$$300000 + \frac{18000 + 4500 + 5000}{5} = 306100 \text{ грн.}$$

Термін окупності капітальних вкладень

$$T = \frac{306100}{154100} = 1,99 \text{ років.}$$

4. Розрахунковий коефіцієнт ефективності (E_r)

$$E_r = \frac{1}{1,99} = 0,503$$

Оскільки $E_r > E_n(0,33)$, проект вважається ефективним.

5. Річний економічний ефект

$$E = \Delta \text{Ст} - K_D \cdot E_n, \quad (3.4)$$

де $E_n=0,15$.

$$E = 154100 - 306100 \cdot 0,15 = 107185 \text{ грн.}$$

6. Річна економія трудових витрат

$$\Delta T = 480000 - 350000 = 130000 \text{ грн.}$$

Виконані розрахунки показали, що впровадження системи принесе підприємству вигоду у розмірі 107185 грн на рік. Термін окупності капітальних вкладень складає 1,99 років, що підтверджує економічну доцільність проекту.

Висновки до розділу 3

У розділі розроблена об'єктно-орієнтована модель автоматизованого робочого місця співробітника. Розглянуто принципи, методи та аспекти застосування об'єктно-орієнтованого аналізу для моделювання предметної області, визначення асоціацій, а також побудови об'єктних моделей із використанням мови UML.

Основою розроблення моделі стала концепція предметної області, яка представлена у вигляді ключових концептуальних класів, асоціацій між ними та атрибутів. Модель була описана як візуальний словник абстракцій, що є зручною формою для сприйняття та аналізу. Значну увагу приділено методу декомпозиції, що дозволяє структурувати предметну область і виділити ключові компоненти.

Об'єктна модель була розроблена з використанням UML, що є стандартом для візуалізації та документування програмних систем. У роботі розглянуто основні елементи UML, включаючи діаграми класів, об'єктів, варіантів використання, активностей, станів, послідовностей, розгортання і компонентів. Кожна з цих діаграм забезпечує різні аспекти представлення та аналізу системи, що дозволяє отримати повну картину її функціонування та архітектури.

Розглянуто діаграму класів, яка включає основні класи: працівники, проекти, ресурси, відділи. Описані їх атрибути, методи та взаємозв'язки, що демонструють логічну структуру системи. Діаграма об'єктів ілюструє конкретні екземпляри класів, наприклад, працівників, які призначені на проекти, чи ресурси, виділені для реалізації завдань.

Діаграми активностей і послідовностей описують динамічні аспекти процесів, зокрема призначення працівників на проекти, виділення ресурсів і створення нових проектів. Вони наочно ілюструють потік дій і взаємодій між компонентами системи, дозволяючи зрозуміти алгоритми її роботи.

Діаграма станів моделює життєвий цикл проекту, включаючи стани «Чернетка», «У процесі», «Завершено» та «Закрито». Це дозволяє чітко окреслити, як проект змінюється на різних етапах його виконання.

Розглянута діаграма розгортання описує фізичну архітектуру системи, включаючи сервер додатків, базу даних і клієнтські пристрої для менеджера, працівників і адміністратора. Діаграма компонентів деталізує логічну структуру основних функціональних частин системи, їх взаємодію та відповідність бізнес-процесам.

Виконано аналіз основних показників функціонування системи, таких як економія витрат і трудових ресурсів, термін окупності капітальних вкладень і річний економічний ефект. Результати показали, що впровадження системи є економічно доцільним, із терміном окупності менше двох років.

Отже, розроблена об'єктно-орієнтована модель забезпечує комплексний підхід до створення автоматизованого робочого місця співробітника, з урахуванням як функціональних, так і економічних аспектів. Це підкреслює ефективність запропонованої методики й надає чітке уявлення про процеси проектування, реалізації та впровадження системи.

ВИСНОВКИ

У роботі була поставлена і вирішена актуальна задача розробки методології та об'єктно-орієнтованих моделей інформаційної системи (ІС) підприємства для автоматизації робочих місць співробітників. Комплексний підхід до аналізу, проектування та впровадження системи дозволив створити ефективну модель, яка забезпечує адаптацію до змін у бізнес-процесах, гнучкість і масштабованість.

1. Виконано аналіз сучасного стану інформаційних систем, зокрема їх класифікації, історії розвитку, типів та напрямків використання. Виявлено, що сучасні ІС є невід'ємною складовою підприємств різних рівнів: від базових систем для автоматизації окремих робочих місць до комплексних корпоративних рішень, що охоплюють всі аспекти діяльності організації. Особливу увагу приділено розвитку ІС у виробничих та медичних галузях, де автоматизація процесів і прийняття рішень є критичним чинником ефективності.

2. Досліджено ітераційну модель життєвого циклу інформаційних систем, яка забезпечує можливість внесення змін на будь-якому етапі розробки та впровадження системи. Модель дозволяє розподіляти проєкт на кілька фаз, кожна з яких завершується отриманням прототипу, що відповідає певному набору вимог. Це сприяє покращенню узгодженості між замовником і розробником, скороченню термінів впровадження та зменшенню витрат. Ітераційний підхід довів свою ефективність у складних проєктах, що вимагають гнучкості та адаптивності.

3. Розглянуто методологію створення системи узгоджених моделей, яка включає стратегічний, укрупнений та детальний рівні опису діяльності організації. Моделювання бізнес-процесів стало відправною точкою для визначення вимог до ІС, що дозволило забезпечити тісний зв'язок між цілями підприємства та функціональними можливостями системи. У роботі виділено ключові аспекти створення моделей, такі як врахування динамічних характеристик процесів, їх кінцевий результат та взаємодії між елементами системи.

4. Обґрунтовано доцільність застосування об'єктно-орієнтованого підходу до моделювання. Виявлено, що використання UML-діаграм для візуалізації структури

та функціонування системи дозволяє значно спростити процес розробки, зробити його прозорим для всіх учасників проєкту та забезпечити високу узгодженість між різними компонентами ІС. Застосування об'єктно-орієнтованого підходу забезпечує повторне використання компонентів, гнучкість і можливість масштабування.

5. Розроблено об'єктно-орієнтовану модель автоматизованого робочого місця співробітника. Модель включає чотири ключові класи: Employee, Project, Resource та Department. Було розроблено відповідні UML-діаграми, які відображають структуру класів, їх атрибути, методи та взаємозв'язки. Крім того, побудовано діаграми об'єктів, активностей, станів, послідовностей, варіантів використання, компонентів і розгортання, які описують усі аспекти функціонування системи.

6. Розглянуто конкретні варіанти використання системи, такі як створення проєктів, призначення працівників на проєкти, виділення ресурсів та оновлення даних. Це дозволило деталізувати ключові процеси і визначити їх алгоритми, що сприяє точній реалізації функціоналу системи. Кожен варіант використання супроводжується описом основних і альтернативних сценаріїв, що забезпечує повноту охоплення можливих ситуацій під час експлуатації системи.

7. Проаналізовано економічну ефективність впровадження інформаційної системи. Результати показали, що автоматизація процесів дозволяє суттєво скоротити витрати, оптимізувати використання ресурсів і зменшити трудові витрати. Розрахунки економічних показників, таких як термін окупності капітальних вкладень та річний економічний ефект, підтвердили доцільність впровадження розробленої системи.

На основі проведених досліджень зроблено такі висновки:

1. Сучасні інформаційні системи мають критичне значення для автоматизації бізнес-процесів, підвищення ефективності підприємств та їх адаптації до динамічних змін ринку.

2. Використання ітераційної моделі життєвого циклу дозволяє гнучко підходити до розробки системи, забезпечуючи її відповідність вимогам замовника та можливість внесення змін на будь-якому етапі проєкту.

3. Методологія побудови узгоджених моделей забезпечує систематичний підхід до визначення вимог, що дозволяє уникнути помилок і неточностей у процесі розробки.

4. Об'єктно-орієнтований підхід є ефективним інструментом для проектування ІС, забезпечуючи зрозумілість структури системи, повторне використання компонентів та їх легку інтеграцію.

5. Автоматизація процесів управління працівниками, проектами та ресурсами сприяє зменшенню витрат часу і ресурсів, а також підвищує точність і прозорість управлінських рішень.

На основі отриманих висновків запропоновано такі рекомендації:

1. Використовувати ітераційну модель життєвого циклу для розробки корпоративних інформаційних систем, особливо в умовах постійно змінних вимог.

2. Застосовувати об'єктно-орієнтоване моделювання для створення структурованих, гнучких і масштабованих систем.

3. Використовувати сучасні CASE-засоби для автоматизації всіх етапів життєвого циклу ІС, включаючи моделювання, тестування та документування.

4. Інтегрувати систему із зовнішніми інформаційними джерелами, що сприятиме розширенню її функціональності та підвищенню конкурентоспроможності підприємства.

Таким чином, поставлені задачі розв'язано у повному обсязі. Напрямок подальших досліджень є інтеграція розроблених моделей у практичні інформаційні системи та оцінка їх ефективності на реальних підприємствах.