|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ЗМІСТ**  СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ 3  ВСТУП 5  1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ОНЛАЙН-БАНКІНГУ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОЇ РОБОТИ 6  1.1. Опис існуючих рішень інтернет банкінгу 6  1.2. Безпека при виконанні дистанційних платіжних операцій 11  1.2.1 Шифрування у банківських системах 12  1.2.2. Рівень захищених сокетів (SSL) 13  1.2.3. Двофакторний метод аутентифікації (2FA) 14  1.3. Обґрунтування теми проекту та обраних засобів розроблення 16  2. АНАЛІЗ ІНСТРУМЕНТІВ РОЗРОБКИ ТА АПАРАТНОГО  ЗАБЕСПЕЧЕННЯ 17  2.1. IntelliJ IDEA, SAP PowerDesigner та MySQL Workbench 17  2.2. Характеристики веб-серверу 20  2.3. Характеристики серверу баз даних 21  2.3.1 Двигун бази даних InnoDB 23  2.3.2 RAID10 24  3. РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 26  3.1.Розробка концепції рішення 26  3.1.1 Етапи розробки системи банківських операцій користувача 24 | | | | | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | | | | | |
|  |  |  |  |  |
| Змін. | Арк. | № докум. | Підпис | Дата |
| Розробив | | Горпинич-Радуженко І.О. |  |  | Веб-орієнтована система банківських операцій користувача.  Пояснювальна записка | Літ. | | | Аркуш | Аркушів |
| Перевірив | | Орлова М.М. |  |  |  |  |  | 1 | 50 |
| Консульт. | |  |  |  | НТУУ «КПІ»  Кафедра СПіСКС  Група КВ-41 | | | | |
| Н. контроль | | Клятченко Я.М. |  |  |
| Зав. каф. | | Тарасенко В.П. |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3.2.Опис програмного забезпечення 17  3.3.Розробка можливих додатків, використовуючи розроблену  систему дистанційних платежів 17 | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 2 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ДОДАТКИ**  Додаток 1. Копії графічного матеріалу  Додаток 2. Лістинг програми | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 3 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ**  Об’єкт – сукупність обладнання, яке підключено до інтерфейсного блоку;  SMS (від англ. Short Message Service) – технологія, яка дозволяє здійснювати прийом і передачу коротких текстових повідомлень; | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 4 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ВСТУП**  На сьогоднішній час, дистанційні фінансові операції є невід’ємною частиною соціального існування кожної людини у світі. Цей факт особливо стосується людей малого та середнього фінансового достатку, що є більшою частиною земного населення. При виконанні банківських операцій вже не потрібна особиста присутність людини у банківській установі, достатньо мати ідентифіковані персональні дані у системі банківського контролю та доступ до мережі Інтернет. Є необхідність у підвищенні рівня банківської освіченості, тому що дистанційні банківські операції виконуються у автоматичному режимі.  Банківські операції користувача включають в себе роботу з банківськими онлайн переказами, депозитними та кредитними рахунками, управління банківськими картками.  Банки та інші фінансові установи працюють над розробкою та впровадженням веб-орієнтованих систем через їх потенціал підвищення ефективності, скорочення витрат і залучення нових клієнтів. Споживачі починають використання цих технологій через підвищену зручність та економію часу. У період з 1995 по 2003 рік, кількість електронних банківських операцій збільшилася восьмикратно. У період з кінця 2002 року до початку 2005 року, використання веб-орієнтованих банківських мереж збільшилася на 47 відсотків[1]. Лише в Україні на початок 2017 року системами дистанційних фінансових операцій користуються більш ніж 10 мільйон громадян.[2]  Проте відсутність фінансової освіченості викликає цифровий розрив та віддалення людей малого та середнього фінансового достатку від використання усіх можливих переваг онлайн банкінгу. Тому є необхідність у розробці веб-орієнтованої банківської моделі операцій користувача, для здобуття необхідних навичок використання цих систем. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 5 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ**  **ТЕМИ ДИПЛОМНОГО ПРОЕКТУ**  **1.1. Аналіз існуючих систем онлайн банкінгу**  Розглянемо існуючі системи онлайн банкінгу в Україні.  На сьогоднішній день в Україні існує багато веб-орієнтованих систем для надання банківських послуг як від самих банків, так і комерційні рішення інших виробників програмного забеспечення. Розглянемо декілька основних.  Система «Privat24»  Лідером на ринку України є система «Privat24», тому що переважна кількість державного безготівкового грошового обігу виконується за допомогою цього додатку.  Програмний реалізація має необхідну, у сучасний час, дворівневу систему автентифікації .  Для початку роботи потрібно лише мати телефон, який був указаний при реєстрації у банку. Після декількох годин користувач автоматично буде зареєстрований у системі онлайн банкінгу.    Рис.1.1 Інтерфейс онлайн-банкінгу «Privat24» | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 6 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Система «Ощад24»  Система державного банку - «Ощадбанку». Процедура аутентифікації дуже схожа на «Приват24», тому що потребує підключення SMS банкінгу для можливості дворівневої аутентифікації.  На відміну від програмного рішення «Приватбанку», при переказі грошів між картками власного банку необхідно віконувати лише за номером. При такому підході користувачеві дуже легко помилитися. У системі є базовий функціонал для роботі з депозитними рахунками (можливість роботи лише з декількома видами). Великим недоліком є неможливість використання альтернативних можливостей аутентифікації окрім SMS.  E:\Documents\4 курс\Diploma\o24.jpg  Рис.1.2 Інтерфейс онлайн-банкінгу «Ощад24» | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 7 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Система «UKRSIB online» та «StarAccess»  Один з найголовніших банків України теж запровадив систему дистанційного веб-орієнтованого онлайн банкінгу «UKRSIB online» - для фізичних осіб, «StarAccess» - для юридичних осіб.  C:\Users\John\Downloads\Telegram Desktop\photo_2018-05-20_12-14-23.jpgC:\Users\John\Downloads\Telegram Desktop\photo_2018-05-20_12-14-17.jpgУ 2018 році «Укрсіббанк» зробив ребрендінг свого онлайн банкінгу для фізичних осіб зі «Star24» на «UKRSIB online», і це вдалося їм з великим успіхом. Система побудована на базі новітніх технологій безпеки та багатошарової автентифікації клієнтів. Використовує принцип RESTFul сервісів, тобто з мобільного додатку можна повноцінно користуватись онлайн банкінгом та усіма його функціями. Цікавою особливістю є контроль витрат і надходжень. UKRSIB online в автоматичному режимі проводить аналіз ваших витрат і надходжень, розбиває їх по категоріях та показує в графіках.  Рис.1.3 Інтерфейс мобільного онлайн-банкінгу «UKRSIB online» | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 8 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис.1.4 Інтерфейс онлайн-банкінгу «UKRSIB online»  Система **онлайн-банкінгу** StarAccess надає зручний інструмент дистанційного управління коштами для юридичних осіб, відкриваючи перед ними широкий перелік можливостей, високий рівень надійності і гарантією безпеки.  Система передбачає наявність у користувача спеціального брелока з ЄЦП, який підтверджує особистість при роботі з додатком.  Можливо, це є незручністю підключення до системи - обов'язково потрібно відвідати філію банку і отримати ЄЦП. Але система представляє користувачеві стабільну роботу з достатнім набором функцій управління рахунком та високим рівнем безпеки.  Система StarrAccess використовує додаток розроблений на мові Java для забеспечення максимальної надійності виконаних транзакцій. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 9 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E:\Documents\4 курс\Diploma\s24.jpg  Рис.1.5 Інтерфейс онлайн-банкінгу «StarAccess»  Система онлайн платежів «Portmone»  Система «Portmone» це міжбанківська [електронна платіжна система](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BF%D0%BB%D0%B0%D1%82%D1%96%D0%B6%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), яка виконує послуги доставки і оплати рахунків за різні товари і послуги в дистанційному режимі через [Інтернет](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82), оплата комунальних послуг, переводи з карти на карту (не прив’язуючись до власного банку), поповнення мобільного, отримання грошових переказів за допомогою міжнародних платіжних карток [Visa](https://uk.wikipedia.org/wiki/Visa) і [MasterCard](https://uk.wikipedia.org/wiki/MasterCard). | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 10 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис.1.5 Інтерфейс системи онлайн платежів «Portmone»  1.2. Безпека при виконанні дистанційних платіжних операцій  В онлайн-банкінгу, як і в традиційних банківських системах, безпека є основною проблемою. Розробники спеціалізованого програмного забезпечення намагаються вживати всі необхідні запобіжні заходи, щоб переконатися, що інформація клієнтів передається безпечно та надійно. Найновіші методи в безпеці онлайнової банківської системи використовуються для збільшення моніторингу цілісності та безпеки системи. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 11 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Захист онлайн-банкінгу повинен розглядаєтися на трьох рівнях. Перша проблема полягає в безпеці інформації про клієнтів, оскільки вона надсилається з ПК-клієнта на веб-сервер. Друга частина стосується безпеки навколишнього середовища, в якому розташована мережа банківських серверів та база даних клієнтів. Нарешті, вживаються заходи безпеки, щоб запобігти неавторизованим користувачам входити в розділ онлайн-банкінгу веб-сайту.  1.2.1 Шифрування у банківських системах  Захист конфіденційності повідомлень між браузером та серверами банку забезпечується за допомогою шифрування. Шифрування відбувається так: коли користувач переходе на сторінку входу онлайн-банкінгу, браузер встановлює безпечний сеанс на нашому сервері. Безпечний сеанс встановлюється за протоколом Encryption Secure Sockets Layer (SSL).  Цей протокол вимагає обміну, що називаються публічними та приватними ключами. Ключі - випадкові числа, вибрані для цього сеансу, і вони відомі лише між браузером користувача та сервером банку. Після обміну ключами браузер використовуватиме ці цифри для розбиття (шифрування) повідомлень, надісланих від браузера до сервера. Обидві сторони вимагають ключів, оскільки їм потрібно розшифрувати повідомлення, після їх отримання. Протокол SSL не тільки забезпечує конфіденційність, але також гарантує, що жоден інший веб-користувач не зможе "видати себе" за іншого користувача, а також не змінить жодну надіслану інформацію. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 12 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1.2.2. Рівень захищених сокетів (SSL)  Це стандартна технологія безпеки для встановлення зашифрованого зв'язку між сервером і клієнтом, як правило, веб-сервером (веб-сайтом) та веб-переглядачем, або поштовим сервером і поштовим клієнтом (наприклад, Outlook).  SSL дозволяє безпечно передавати конфіденційну інформацію, таку як номери кредитних карток, номери соціального захисту та облікові дані для входу. Зазвичай, дані, що надсилаються між веб-переглядачами та веб-серверами, надсилаються звичайним текстом, залишаючи вас вразливими до прослуховування та крадіжки ваших даних. Якщо зловмисник може перехопити всі дані, що надсилаються між веб-переглядачем та веб-сервером, він може бачити та використовувати цю інформацію.  Більш конкретно, SSL - це протокол безпеки. Протоколи описують, як слід використовувати алгоритми. У цьому випадку протокол SSL визначає змінні шифрування як для прийнятих, так і для переданих даних.  Всі веб-браузери мають можливість взаємодіяти з захищеними веб-серверами за допомогою протоколу SSL. Однак браузеру та серверу потрібно те, що називається SSL-сертифікатом, щоб мати змогу встановити безпечне з'єднання.  SSL захищає мільйони даних людей в Інтернеті щодня, особливо під час онлайн-транзакцій або передачі конфіденційної інформації. Користувач може визначити, чи знаходиться його браузер у беспечному режимі, побачивши символ захисту у вікні веб-браузера.    Рис.1.6 Приклад відображення захищеного режиму | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 13 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1.2.3. Двофакторний метод аутентифікації (2FA)  З кожним днем зростає кількість способів отримання доступу до приватних даних користувача при використанні стандартних процедур безпеки, що вимагають лише просте ім'я користувача та пароль. Шахраї намагаються отримати як особисті та фінансові дані, а потім використовують цю інформацію здійснення шахрайських дій, загалом фінансового характеру.  Двофакторна автентифікація, також відома як 2FA, є додатковим рівнем безпеки, яка вимагає не тільки пароля та імені користувача, але якусь додаткову унікальну інформацію, яку користувач повинен відразу пред’явити – це може буди інформація яка надходить до вашого телефону у вигляді SMS, але це також може бути USB-ключ (апаратний токен) або інший пристрій, який може генерувати одноразові коди.  Користувач, який використовує сервіс з двофакторною, зазвичай надаєте свій номер мобільного телефону (також може використовуватися мобільний додаток, але це найпоширеніший спосіб використання двофакторної автентифікації.). Більшість форм двофакторної автентифікації вимагають входу за допомогою імені користувача та пароля, а потім введення коду, який надіслано вам через SMS. Цей метод не лише підтверджує, що ви знаєте базові данні для авторизації (ім'я користувача та пароль), але і те, до чого має доступ лише користувач (мобільний телефон), номер якого зареєстрований як пристрій для отримання цих кодів.  Використання імені користувача та пароля разом із інформацією, яку знає лише користувач, ускладнює отримання потенційними зловмисникам доступу та викрадення особистих даних користувача. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 14 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Історично склалося так, що двофакторна автентифікація - це не нова концепція, але її використання стало набагато більш поширеним з початку цифрової ери, в якій ми зараз живемо. У лютому 2011 року, Google оголосила про двофакторну автентифікацію для доступну до своїх сервісів, потім до них приєдналися такі гіганти як MSN і Yahoo, Microsoft.  Багато людей, напевно, не знають, що такий тип процесу безпеки називається двофакторною автентифікацією, і, ймовірно, навіть не замислюється про це, коли використовують апаратні токени, виданих їхнім банком для використання з їхньою карткою та особистим ідентифікаційним номером, коли користувач хоче виконати операцію за допомогою онлайн-банкінгу.  Використання процедури аутентифікації за двома факторами може допомогти зменшити кількість випадків крадіжки особистих даних в Інтернеті, а також фішингу через електронну пошту, оскільки злочинцям потрібно більше, ніж просто дані про ім'я користувача та пароль.  Недоліком цього процесу безпеки є те, що нові апаратні токени (у вигляді ключів або кард-рідерів) мають бути замовлені, а потім видані користувачеві, і це може призвести до уповільнення процесу початку роботи з сервісом, щоб отримати доступ до своїх приватних даних за допомогою цієї процедури автентифікації. Токени також зазвичай мають невеликі габарити, тому легко втрачаються, що викликає більше проблем для оформлення нової заяви для виробництва нового ключа. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 15 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E:\Documents\4 курс\Diploma\photo_2018-05-20_16-39-34.jpg  Рис.1.7 Приклад програмного токену для виконання дворівневої автентифікації у системі онлайн-банкінгу «StarAccess»  1.3. Обґрунтування теми проекту та обраних засобів розроблення  Дана розробка є актуальною, тому що виявлений високий рівень розвитку технологій у сфері дистанційних банківських операцій. Проте відсутність фінансової освіченості викликає цифровий розрив та віддалення людей малого та середнього фінансового достатку від використання усіх можливих переваг онлайн банкінгу. Тому є необхідність у розробці веб-орієнтованої банківської моделі операцій користувача, для здобуття необхідних навичок використання цих систем.  Для розробки дипломного проекту мною була обрана мова високого рівня Java. Технології мови Java включають у себе великий набір API-інтерфейсів, інструментів і реалізацій часто використовуваних алгоритмів безпеки, механізмів і протоколів. API безпеки Java охоплює широке коло областей, включаючи криптографію, інфраструктуру відкритих ключів, захищений зв'язок, автентифікацію та контроль доступу. Технологія безпеки Java надає розробнику всеохоплюючу основу для написання додатків, а також надає користувачеві або адміністратору набір інструментів для безпечного управління програмами. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 16 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2. АНАЛІЗ ІНСТРУМЕНТІВ РОЗРОБКИ ТА АПАРАТНОГО ЗАБЕСПЕЧЕННЯ  2.1. IntelliJ IDEA, SAP PowerDesigner та MySQL Workbench  IntelliJ IDEA  IntelliJ IDEA - це інтегроване середовище розробки (IDE) для мови високого рівня Java. Використовується для розробки комп'ютерного програмного забезпечення. Вона пропонує внутрішній аналіз, мультимедійний синтаксис, автодоповнення, історію змін, інтегроване підключення до бази даних, інтегроване рішення для контролю версій, можливість під’єднання серверу для автоматичного завантаження веб-проекту, підключення додатків для підтримки інших мов програмування. IntelliJ IDEA відрізняється від аналогів простотою використання, гнучкістю та надійним дизайном. При використанні IntelliJ IDEA, ви отримаєте інтегроване середовище розробки, але з великою кількістю доповнень, які роблять розробку набагато простішою і приємнішою.Приведемо декілька з них:  * Вкладка автодоповнення (імена класів, функції, методи, змінні) * Чіткіші повідомлення про помилки і їх виділення кольором * Аналіз потоку даних у реальному часі * Гарна інтеграція з модулями (GIT, Maven, TypeScript) * Розумний рефакторінг коду * Виявлення дублікатів   Особливістю IntelliJ IDEA є підтримка [Spring Integration Framework](http://www.springsource.org/spring-integration) - розширення для веб-програмування, що підтримує моделі корпоративної | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 17 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| інтеграції.  Недоліком цієї системи є платний доступ для використання. Але кожен студент має можливість зробити підписку для навчальних цілей.  SAP PowerDesigner  SAP PowerDesigner – це програмий додаток для спільного інструментального моделювання архітектури програмного забеспечення та баз даних.  За допомогою цього додатку, з’являється можливість швидкого проектування як самої архітектури програмного забезпечення, для наглядного і простого способу протоколювання механізмів взаємодії окремих програмних модулів, так і проектування моделей бази даних, не прив’язуючись до конкретної реалізації обраної системи бази даних, з подальшою можливістю компілювання логічних та концептуальних схем у скрип виконання певної обраної системи управління базою даних.  PowerDesigner підтримує дизайн [моделей архітектури](https://en.wikipedia.org/wiki/Model-driven_architecture) програмного забезпечення. PowerDesigner зберігає моделі за допомогою різноманітних розширень файлів. Внутрішня структура файлу може бути або XML, або у форматі, бінарного файлу. PowerDesigner також може зберігати моделі в сховищі бази даних.  PowerDesigner підтримує побудову таких видів моделей, як:   * Модель вимог (RQM). Допомагає вам проаналізувати будь-які письмові вимоги та пов'язати їх * Модель корпоративної архітектури (EAM). Допомагає аналізувати та документувати бізнес-функції | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 18 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * Концептуальна модель (CDM). Допомагає проаналізувати концептуальну структуру інформаційної системи * Об'єктно-орієнтована модель (OOM). Допомагає проаналізувати інформаційну систему через об’єкти використання * Логічна модель даних (LDM). Допомагає проаналізувати структуру інформаційної системи * Фізична модель даних (PDM). Допомагає проаналізувати таблиці, представлення даних та інші об'єкти в базі даних   PowerDesigner має підтримку:  Створення коду ( Java , C # , Hibernate, JSF, WinForm )   * Моделювання даних (працює з більшістю основних систем RDBMS ) * Моделювання складу даних ( WarehouseArchitect ) * Моделювання об'єкта ( діаграми UML) * Аналіз вимог * XML моделювання - схеми і DTD стандарти   Недоліком є не можливість неспівпадіння типів даних при побудові архитектури бази даних до конкретно обраної мови програмування.  MySQL Workbench  MySQL Workbench - це візуальний інструмент розробки баз даних , який інтегрує розробку , адміністрування , створення баз даних , створення та підтримку SQL запитів в єдине інтегроване середовище розробки для бази даних.  MySQL Workbench надає розробникам повний набір візуальних інструментів для створення, редагування та управління SQL запитами, з'єднаннями з базами даних та об'єктами. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 19 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Візуальний редактор SQL дозволяє розробникам створювати, редагувати та виконувати запити, створювати та редагувати дані, а також переглядати та експортувати результати.  MySQL Workbench надає такі можливості, як:   * Підсвічування синтаксису * контекстно-конфіденційна довідка * автозаповнення * налагодження SQL-інструкцій.   Програма надає набір інструментів для підвищення продуктивності додатків  MySQL. DBA можуть швидко переглянути основні показники ефективності, використовуючи інформаційну панель продуктивності. Звіти про ефективність забезпечують легку ідентифікацію високоякісних SQL-операторів, тощо. Крім того, розробники можуть бачити, де оптимізувати свій запит за допомогою покращеного та простого у використанні плану підказок.  2.2 Характеристики веб-серверу  Плануючи серверні вимоги до обладнання для розгортання веб-серверу, буде необхідно оцінити масштабованість серверів на основі пікових відвідувачів, співвідношення максимальної кількості користувачів до пікового значення одночасних підключень та загальної навантаженості систем бізнес логіки.  Коефіцієнт перегляду – відношення кількості відвідувачів які виконують оновлення, до тих, що переглядають лише вміст сторінки. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 20 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Загальний вміст системи краще всього оцінювати за рахунком загальної пропускної спроможності.  Стійкість системи найкраще оцінюється за умови постійного потоку одночасних користувачів.  Значення, наведені нижче, стосуються мінімального наявного устаткування, необхідного для запуску Apache Tomcat наприклад, розмір мінімальної купи становить 250 МБ. Вам знадобиться додаткове фізичне обладнання, принаймні мережева карта, необхідна будь-яким додаткам, що працюють на сервері.  Мінімальна рекомендація щодо обладнання:   * процесор Intel Core 2 Duo 2.40 Ггц або вище * оперативна пам’ять обсягом 2 ГБ або більше * 64-розрядна операційна система * жорсткий диск не менше 1 ГБ   У особливих випадках навантаження на сервер перш за все залежить від пікових відвідувачів, тому мінімальні системні вимоги важко судити. Ці цифри надаються як керівництво до абсолютного мінімуму, необхідного для запуску сервера, і для вашої конфігурації, швидше за все, буде потрібно обладнання з кращими характеристиками.  Провівши навантажувальне тестування та проаналізувавши необхідні характеристики для запуску веб-серверу, були вирахувані приблизні показники технічних характеристик для можливих випадків одночасного виконання операцій. Зазначено, що обсяг оперативної пам'яті стосується або загальної пам'яті сервера або пам'яті, що виділяється JVM. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 21 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Операції | Процесори | Процесор (ГГц) | ОЗУ (МБ) | | 150 | 1 | 2.6 | 1024 | | 250 | 2 | 2.8 | 1,536 | | 1000 | 4 | 3 | 2,048 | | 4 000 | 2 | 3.8 | 2,048 | | 6 000 | 2 | 3.6 | 4,096 | | 8 000 | 2 | 3.6 | 4,096 | | 10 000 | 4 | 2.6 | 4,096 |   Таблиця.4.1 результати навантажувального тестування  2.3. Характеристики серверу баз даних  Система пред'являє наступні вимоги до апаратного та програмного забезпечення сервера:  Мінімальні вимоги до сервера:   * 2-х ядерний процесор з тактовою частотою 2. Ггц або вище * оперативна пам’ять обсягом 4 ГБ * один завантажувальний диск не менше 1 ГБ * 32-х розрядна операційна система.   Рекомендовані вимоги до сервера:   * 4-х ядерний процесор або більше * оперативна пам’ять обсягом 8 ГБ або більше * мінімум два логічних дискових елемента під’єднаних за технологією RAID10 * один завантажувальний диск не менше 1 ГБ * 64-х розрядна операційна система. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 22 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Існує також можливість інсталяції системи на платформі Windows. При цьому вимоги до попередньо програмного забезпечення залишаються незмінними.  Для роботи з клієнтською частиною системи потрібен комп'ютер, підключений по протоколу TCP / IP до мережі, в якій знаходиться.Система пред'являє наступні вимоги до апаратного та програмного забезпечення сервера.  2.3.1 Двигун InnoDB  InnoDB - це двигун загального зберігання даних, який забезпечує високу надійність та високу продуктивність. У MySQL, InnoDB є двигуном зберігання за замовчуванням. Розглянемо основні особливості:   * InnoDB таблиці організовують дані на диску для оптимізації запитів на основі первинних ключів (PRIMARY KEY). У кожній InnoDB таблиці є індекс первинного ключа, який називається кластерним індексом, який організовує дані для мінімізації введення / виводу та пошуку первинних ключів, використовуючи алгоритм “B-tree”. * Для збереження цілісності даних , InnoDB підтримує зовнішні ключі (FOREIGN KEY). Вставки, оновлення та видалення перевіряються за зовнішніми ключами, щоб попередити загублення даних у пов’язаних таблицях. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 23 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2.3.2. RAID10  RAID (надлишковий масив незалежних дисків) - конфігурація жорстких дисків, яка використовє методи розмітки , дзеркальне відображення або парності для створення великих надійних сховищ даних з декількох загального призначення комп'ютерних накопичувачів на жорстких магнітних дисках (HDD).  RAID10 - гібридний RAID, використовується для отримання додаткової продуктивності, зменшення нагрузки на носії, досягання більшої захищеності даних, в результаті об'єднання властивостей двох стандартних моделей підключення RAID.   * RAID 10, як правило, реалізується RAID-контролерами, та являє собою масив дзеркальних сховищ типу RAID 0, який може являти собою двосторонні та тристоронні дзеркала та вимагає щонайменше чотирьох дисків. Тим не менш, нестандартне визначення Також можливі масиви більше чотирьох дисків. * Його операції DML використовують модель ACID, а транзакції, що передбачаються цією моделлю, дозволяють захистити дані користувачів * Рівень замикання транзакції збільшує можливості багатопоточного використання бази даних | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 24 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| https://upload.wikimedia.org/wikipedia/commons/thumb/e/e6/RAID_10_01.svg/800px-RAID_10_01.svg.png  Рис.4.1 Типова конфігурація RAID10 | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 25 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3. РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ  3.1.Розробка концепції рішення  Вибір мови програмування  Для розробки дипломного проекту мною була обрана мова високого рівня Java. Технології мови Java включають у себе :   * API-інтерфейсів * інструментів і реалізацій часто використовуваних алгоритмів безпеки * механізмів протоколювання.   Java – це об’єктно-орієнтована мова програмування, тому ця парадигма має на увазі використання так званих «об’єктів», для реалізації будь яких зв’язків та операцій, які теж відбуваються безпосередньо на самих об’єктах, змінюючи їх стан.  API безпеки Java охоплює широке коло областей, включаючи криптографію, інфраструктуру відкритих ключів, захищений зв'язок, автентифікацію та контроль доступу. Технологія безпеки Java надає розробнику всеохоплюючу основу для написання додатків, а також надає користувачеві або адміністратору набір інструментів для безпечного управління програмами.  Ще одною перевагою Java є її можливість кросс-платформеного розгортання розробленого програмного забезпечення на будь якому апаратному інтерфейсі, для якого розроблена так звана «віртуальна машина», яка транслює компільований байткод, незалежно від конкретної реалізації центрального процесора (наприклад, у мові C написаний код кожен раз компілюється відносно системи центрального процесора, і тому робота програми на різних пристроях може відрізнятися та приводити до важко виявлених помилок). | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 26 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Такий підхід описує парадигму мови Java: «Написана програма один раз, запускається на будь якій системі однаково (Write ones - run everywhere)».  Віртуальна машина Java ( JVM ) являє собою віртуальну машину , яка дозволяє комп'ютеру працювати Java - програмам, а також програмам , написаних на інших мовах і інтерпретованих на Java байткод . JVM детально визначається специфікацією, яка формально описує, що вимагається від впровадження JVM. Наявність специфікації забезпечує сумісність програм Java з різними реалізаціями, так що авторам програмне потрібно турбуватися про особливі схеми базової апаратної платформи.  https://upload.wikimedia.org/wikipedia/commons/d/dd/JvmSpec7.png  Рис.3.1 Огляд віртуальної машини Java  Обраною версією Java для розробки проекту є восьма версія. Це пов’язано з новими методами реалізації інтерфейсів безпеки, що є дуже важливим аспектом для систем дистанційних банківських систем, та новими методами роботи з функціональними інтерфейсами, таких як «Лямбда-вираження» та «Stream». | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 27 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| За допомогою цих інструментів підвищується можливість абстрагування при розробці програмного забезпечення та більших можливостей при обробці великих масивів даних, використовуючи лише декілька команд.  Наприклад, при розробці сервісу для роботи з користувачами, даний функціонал допомагав перебирати велику кількість користувачів, для знайдення необхідного запису у базі даних, та перевірці правильності введеного паролю.  3.1.1 Етапи розробки системи банківських операцій користувача  Етапи розробки дипломного проекту можна розбити на такі складові:   * розробка концепції об’єктів, які будуть використовуватися * вибір системи управління бази даних, яка задовольняє потребам * налаштування багатопоточного з’єднання з базою даних * розробка реляційної бази даних * розробка архітектури об’єктів DAO, за допомогою яких буде відбуватися спілкування програмного інтерфейсу конкретної програмної реалізації з інтерфейсами обраної бази даних * розробка трансферних об’єктів, які обробляють запити від бази даних, та створюють безпосередньо моделі об’єктів, для подальшої роботи з ними * реалізація сервісних об’єктів, які будуть виконувати конкретні зміни над об’єктами, які передбачені необхідністю виконання бізнес логіки * реалізація командних об’єктів, за допомогою яких буде виконуватись виклик заздалегідь обраного необхідного сервісу * розробка контролера, який буде обробляти запити від клієнта, та ставити на виконання необхідні команди | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 28 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * розроблення контролера, який буде контролювати правильне відображення клієнтської частини * вибір та імплементація серверного програмного забезпечення * розробка моделі відображення (клієнстької частини) яка буде задовольняти потребам бізнес логіки   Розглянемо кожен етап розробки біль детально.  Розробка концепції об’єктів програмних сутностей.  Об'єктно-орієнтоване програмування — одна з [парадигм](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [програмування](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка розглядає програму як множину [«об'єктів»](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), що взаємодіють між собою. Тому для нашої обраної концепції потрібно спроектувати (розробити) об’єкти, які будуть зодовольняти нашим потребам та в повній мірі будуть описувати необхідну інформацію про сутності програми та їх взаємодію один на одного.  Для цього було розроблено об’єкти описання реальних сутностей, якими ми їх уявляємо у реальному світі, з неохідними властивостями для конкретної задачі:   * об’єкт користувача, який містить у собі поля: ім’я, прізвище, електронної пошти, номера телефону, паролю та ролі * об’єкти рахунків, які містять у собі поля: номер рахунку, тип, ідентифікаційний номер власника, статусу та балансу. Для кодної окремої реалізації рахунків (депозитний, кредитний, дебетовий) були додані необхідні поля які властиві лише цьому виду рахунків. Це зроблено для запобігання надлишковості інформації, та дял економії пам’яті у системі. * об’єкти карток, які містять у собі поля: номер картки, ідентифікаційний номер рахунку, до якого прив'язана картка, пін-код, cvv, дата закінчення терміну дії картки, тип картки. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 29 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * додаткові об'єкти типу (роль, статус, банківська операція, тип, кредитний запит)   Для кожного створеного об’єкта були створені методи взаємодії з їх полями та роботи з ними. Одною з особливостей розробки об’єктів програмної сутності, є використання шаблону проектування «Будівельник».  Часто об'єкти можуть бути складними і їх створення вимагає виконання цілого набору операцій по їх складанню з простіших об'єктів. При цьому може знадобитися використовувати різні імплементації цих простих об'єктів або алгоритм складання може бути різним. Для відділення створення об'єкта від його внутрішніх деталей використовується шаблон Будівельник (Builder).  У шаблоні Будівельник ховається алгоритм збірки і вибір реалізації частин об'єкта всередині класу конкретного будівельника, що реалізує базовий інтерфейс. Залежно від того, який саме алгоритм збірки потрібно, вибирається конкретна імплементація.  Також, при розробці даних об’єктів було дотримано принципу успадкування — механізму утворення нових класів на основі використання вже існуючих. При цьому властивості та функціональність батьківського класу переходять до класу нащадка (дочірнього).  Вибір системи управління бази даних  Незважаючи на те, що всі системи управління базами даних виконують одну і ту ж основну задачу (тобто дають можливість створювати, редагувати і | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 30 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| отримувати доступ до інформації, що зберігається в базах даних), сам процес виконання цього завдання варіюється в широких межах. Крім того, функції і можливості кожної СУБД можуть істотно відрізнятися. Різні СУБД документовані по-різному: більш-менш ретельно. По-різному надається і технічна підтримка.  При порівнянні різних популярних баз даних, слід враховувати, чи зручна для розробки і масштабування дана конкретна СУБД, а також переконатися, що вона буде добре інтегруватися з іншими інтерфейсами, які вже використовуються. Крім того, під час вибору слід взяти до уваги вартість системи і підтримки, що надається розробником.  MySQL - найпопулярніша у світі база даних з відкритим кодом. Завдяки своїй перевіреній продуктивності, надійності та простоті використання, MySQL стала провідним вибором баз даних для веб-застосунків, що використовуються високопрофесійними веб-ресурсами.  MySQL - це система управління базами даних. У реляційних базах даних дані зберігаються не всі разом, а в окремих таблицях, завдяки чому можна досягти підвищення швидкості і гнучкості. Таблиці мають зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мову структурованих запитів плюс найбільш поширений стандартний мова, яка використовується для доступу до баз даних.  MySQL є системою клієнт-серверщо забезпечує підтримку різних обчислювальних машин баз даних, а також кілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API). | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 31 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Система безпеки заснована на привілеях та паролях з можливістю верифікації з віддаленого комп'ютера, за рахунок чого забезпечується гнучкість і безпека. Паролі при передачі по мережі при з'єднанні з сервером шифруються. Клієнти можуть з'єднуватися з MySQL, використовуючи сокети TCP / IP, сокети Unix або іменовані канали.  Cтруктура MySQL трирівнева: бази даних - таблиці - записи. Логічно - таблиця являє собою сукупність записів. А записи - це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці - в межах бази даних, поля - в межах таблиці. Один сервер MySQL може підтримувати відразу кілька баз даних, доступ до яких може розмежовуватися логіном і паролем. Знаючи ці логін і пароль, можна працювати з конкретною базою даних. Наприклад, можна створити або видалити в ній таблицю, додати записи і т.д. Зазвичай ім'я-ідентифікатор і пароль призначаються хостинг провайдерами, які і забезпечують підтримку самої бази даних.  Налаштування багатопоточного з’єднання з базою даних  При розробці програмного забеспечення завжди виникає питання поєднання розробленого інтерфейсу та обраної реалізації системи управління базою даних. У мові Java розроблений API для рішення цієї проблеми, так званий JDBC.  Конектор бази даних Java (JDBC) - це інтерфейс прикладного програмування (API) для мови програмування Java , який визначає, як клієнт може отримати доступ до бази даних . Це технологія доступу до даних на базі Java, яка використовується для підключення бази даних до реалізованого інтерфейсу на мові Java.  Для його налагодження достатньо лиже задати параметри необхідної бази даних (адреса, логін, пароль, ім’я необхідної бази даних), | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 32 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| та виконати інтеграцію драйвера конектора, який обирається вібповідно до обраної системи управління базою даних.  MySQL забезпечує з'єднання для клієнтських додатків, розроблених на мові програмування Java, за допомогою MySQL Connector / J, драйвера, який реалізує API Java Database Connectivity (JDBC) .  Для вирішення проблеми багатопоточного підключення до бази даних, було обрано методологію «сховища з’єднань».  Сховище з'єднань (Connection Pool) є кешем з з'єднань з базою даних , підтримуваних таким чином , що з'єднання можуть бути повторно використані. Сховищя з'єднання використовуються для підвищення продуктивності виконання команд у базі даних. Відкриття та підтримка підключення до бази даних для кожного користувача є дуже дорогим та витрачає ресурси. Після створення з'єднання, воно розміщується в пулі та знову використовується, щоб не встановлювати нове з'єднання. Якщо всі з'єднання зайняті, створюється нове з'єднання і додається до сховища. Використання сховища також зменшує кількість часу, яке програмний інтерфейс повинен чекати, щоб встановити з'єднання з базою даних.  Розробка реляційної бази даних  При розробці реляційної бази даних, була необхідність повністю проаналізувати сферу її використання, та інтерпретувати всі аспекти для подальшої її використання.  Розробка архітектури зав’язків один з найважливіших кроків у розробці. Якщо на цьому етапі не охопити усі подальші аспекти інформативності необхідних програмних сутностей, на наступних етапах розробки програмного забезпечення можуть виникати великі проблеми при реалізації необхідної логіки, а іноді навіть стає неможливим її інтегрування та подальше масштабування програми. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 33 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Насамперед, архітектура бази даних повинна максимально повністю повторювати інформативність розроблених заздалегідь сутностей (користувача, рахунків і т.д.), для можливості сериалізації об’єктів, тобто повного збереження стану об’єктів, з якими працює програмне забезпечення, та можливості у подальшому «відтворити його стан».  Були розроблені наступні таблиці:   * користувачів * кредитних рахунків * депозитних рахунків * дебетових рахунків * запитів на відкриття кредитних рахунків * банківських карт * платежів * допоміжні (роль, статус, тип)   Цікавою особливістю є реалізація таблиць для рахунків. При їх розробці було використано принцип успадкування класів, але перенесеного на реляційну модель.  Для цього потрібно створення таблиці для класу рахунків, а атрибути об'єкта (тобто спеціальні поля, які притаманні кожному окремому виду рахунків) зберігаються у спеціальних таблицях, відповідно до кожного спеціалізованого типу рахунків. Такий спосіб реалізації призводить до скорочення надлишковості інформації та можливості роботи з колекціями рахунків як з одним типом. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 34 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E:\Documents\4 курс\Diploma\My\db.jpg  Рис.3.2 принцип успадкування на реляційній моделі бази даних  Розробка архітектури об’єктів доступу до бази даний на основі шаблону проектування DAO  DAO – Data Access Object (об’єкт доступу до даних) абстрагує і інкапсулює доступ до джерела даних. DAO управляє з'єднанням з джерелом даних для отримання та запису даних.  DAO реалізує необхідний для роботи з джерелом даних механізм доступу. DAO повністю приховує деталі реалізації джерела даних від клієнтів. Оскільки при змінах реалізації джерела даних представлений DAO інтерфейс не змінюється, цей шаблон дає можливість DAO приймати різні схеми сховищ без впливу на бізнес-компоненти. По суті, DAO виконує функцію адаптера між компонентом і джерелом даних. | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 35 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 36 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 37 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 38 |
| Змін. | Арк. | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
|  |  |  |  |  | ДП.468300.004 ПЗ | Арк. |
|  |  |  |  |  | 39 |
| Змін. | Арк. | № докум. | Підпис | Дата |