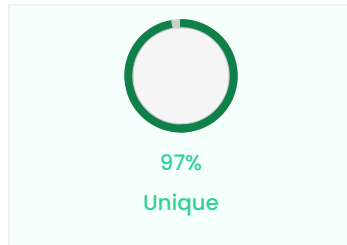
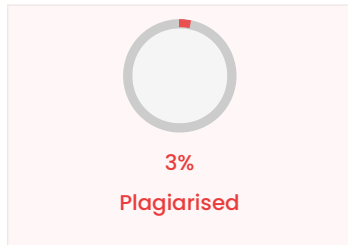


Plagiarism Scan Report



Words Statistics

Words	6838
Characters	54262

Exclude URL: None

Content Checked For Plagiarism

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ» Предметно-циклова комісія інформаційних технологій Д ДОПУЩЕНА Голова ПЦК _____ спеціаліст в/к Сабанов С.О. ВИПУСКНА РОБОТА МОЛОДШОГО С Тема «Розробка прикладної програми для ветеринарної лікарні «Cat-Dog» Виконав ст. гр. _____ І.В. Агапов Керівник викладач _____ Д.О. Костерной Запоріжж: «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ» Предметно-цикл інформаційних технологій ЗАТВЕРДЖУЮ Голова ПЦК спеціаліст в/к Сабанов С.О. _____ " 17 " січня 2022 рок ВИПУСКНОЇ РОБОТИ МОЛОДШОГО СПЕЦІАЛІСТА Студента гр. ІПЗ-118К9 Спеціальності: 121 – Інженерія забезпечення Агапову Ігорю Васильовичу (прізвище, ім'я, по батькові) І. Тема: «Розробка прикладної програми для лікарні «Cat-Dog» затверджена наказом по інституту: № _____ від 04 березня 2022 року 2. Термін зда закінченої роботи: 18 червня 2022 року 3. Перелік питань, що підлягають розробці: 1. Розглянути питання актуальн додатку 2. Провести огляд галузі та аналітику проблеми і її рішень загалом; 3. Провести огляд та аналіз сучас додатків; 4. Провести огляд стеку технологій та для розробки проекту та зробити вибір на підставі вимог; 5. поставленні задачі випускної роботи; 6. Проаналізувати отримані результати; 7. Оформити результати у вигляді пс записки до відповідних ДСТУ норм випускних робіт молодшого спеціаліста. Дата видачі завдання: 17 січня ; Календарний графік підготовки випускної роботи молодшого спеціаліста № етапу Зміст Терміни виконання Готовніс (%), підпис керівника Підпис керівника про повну готовність етапу, дата I Формулювання (корегування) теми випу молодшого спеціаліста та збір практичного матеріалу за темою випускної роботи 17.01.22–26.02.22 2 I атест випускної роботи молодшого спеціаліста 28.03.22–02.04.22 3 II атестація II розділ випускної роботи молодшо 10.05.22–14.05.22 4 III атестація III розділ випускної роботи молодшого спеціаліста, висновки та рекомендації, дода 30.05.22–04.06.22 5 Перевірка випускної роботи програмою «Антиплагіат» 30.05.22–18.06.22 6 Доопрацюван роботи молодшого спеціаліста, підготовка презентації, отримання відгуку керівника і рецензії 06.06.22–11.06.22 ; захист випускної роботи молодшого 14.06.22–18.06.22 8 Подача випускної роботи молодшого спеціаліста на кафедр захисту 9 Захист випускної роботи молодшого спеціаліста 20.06.22–25.06.22 Керівник випуск _____ Д.О. Костерной (підпис) (прізвище та ініціали) Завдання прийняв д _____ І.В. Агапов (підпис студента) (прізвище та ініціали) РЕФЕРАТ Випускна работ сторінок, 36 рисунків, 10 лістингів, 1 додаток. Метою роботи є розробка прикладної програми для ветеринарної ліка дослідження є застосування технологій для створення сучасних прикладних програм. Предметом дослідження прикладної програми для обліку пацієнтів ветеринарної клініки. Здійснено детальний огляд предметної області аналогів. Виявлено, що розробка прикладної програми є доцільною. Проект реалізовано за допомогою С#. Програ є легким у використанні, має привабливий та інтуїтивно зрозумілий інтерфейс, достатній функціонал. Дода заощаджувати час за рахунок мінімального введення даних. Даний програмний продукт стане в нагоді фізич підприємцям та іншим учасникам малого бізнесу. ДОДАТОК, С#, WPF, PostgreSQL ЗМІСТ ПЕРЕЛІК УМОВНИК I

СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 7 ВСТУП 9 РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ 10 1.1 Загальні аспекти 10 1.1.1 Пацієнт 10 1.1.2 Ветеринар 10 1.1.3 Ветеринарна хірургія 11 1.1.4 Ветеринарна клініка 12 1.1.5 Десктопний, Мета програмної системи обліку пацієнтами 14 1.3 Основні аспекти при плануванні впровадження програм ветеринарної лікарні 14 1.4 Огляд останніх досліджень і публікацій 15 1.5 Огляд та аналіз існуючих аналогів 15 1.5.1 1.5.2 БИТ:Айболит 16 1.5.3 Мурмот 17 1.5.4 VetCliniX 18 1.6 Висновки за розділом 19 РОЗДІЛ 2 ІНСТРУМЕНТИ ДОДАТКУ 20 2.1 Засоби розробки 20 2.1.1 Мова програмування C# 20 2.1.2 СУБД PostgreSQL 23 2.1.4 Технологія Візуальне проектування бази даних 31 2.3 UML-діаграма 32 2.4 Висновки за розділом 32 РОЗДІЛ 3 ПРОГРАМНИЙ 3.1 Проектування системи 33 3.2 Програмування системи 36 3.2.1 Основні компоненти багатофайлового процесу Розробка та створення додатку 46 3.3 Перевірка на роботоздатність 52 3.4 Висновки за розділом 55 ПЕРЕЛІК ПК ДОДАТОК А ЛІСТИНГ КОДУ 59 ПЕРЕЛІК УМОВНИК ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Повна назва Пояснення/переклад ПІБ Прізвище, ім'я, по батькові ПК Персональний комп'ютер Однокористувальницька операційна система ОС Операційна система ПЗ Програмне забезпечення Сукупність програм системи інформації та програмних документів БД База даних Сукупність даних, що зберігаються відповідно до схеми даних яких виконується відповідно до правил моделювання даних СУБД Система управління базами даних MS Microsoft Studio Середовище програмування, розроблена компанією Microsoft WinForms Windows Forms Інтерфейс програмування програм, який відповідає за графічний інтерфейс WPF Windows Presentation Foundation Аналог WinForms привабливими можливостями взаємодії з користувачем, що використовує мову XAML XAML eXtensible Application Language Розширювана мова розмітки XML Extensible Markup Language Розширювана мова розмітки програм Application Markup Language Формат файлів Microsoft, який створюється шляхом компіляції файлів XAML C# C# Language Runtime Середовище для байт-коду CIL, в який компілюються програми написані на .NET-середовищі програмування CIL Common Intermediate Language Проміжна мова, розроблена компанією Microsoft для платформ Framework SQL Structured Query Language Мова структурованих запитів MySQL Вільна система керування базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Modeling Language Мова графічного опису для об'єктного моделювання JSON JavaScript Object Notation Текст обміну даними на основі JavaScript GIT Global Information Tracker Система керування версіями з розподіленою UI User Interface Користувальницький інтерфейс WWW World Wide Web Розподілена система, яка надає документи розташованих на різних комп'ютерах HTTP Hyper Text Transfer Protocol Протокол передачі гіпертекстових документів – протокол передачі даних, що використовується в комп'ютерних мережах URL Uniform Resource Identifier Уніфікований локатор ресурсів або адреса ресурсу URI Uniform Resource Identifier Уніфікований ідентифікатор об'єднує абстрактний або фізичний ресурс ВСТУП У наш час медичні організації накопичують величезні обсяги ефективності використання цієї інформації залежить якість медичної допомоги, рівень розвитку країни та кожної області. Необхідність використання великих обсягів інформації під час вирішення діагностичних управлінських, те статистичних та інших завдань, обумовлює створення інформаційних систем у медичних установах. Актуальні ветеринарних клінік звертається досить багато людей зі своїми вихованцями. Щоб знайти в архіві записи пацієнта витратити досить багато часу, саме для цього потрібна ця програма. Вона зробить роботу співробітників більш ефективною і скоротить тимчасові витрати на запис та пошук записів пацієнта в архівах. Метою даної роботи є збір теоретичних про сучасні методи ведення обліку реєстрації пацієнтів на прийом до лікаря відповідних напрямів; огляд та аналіз теоретичної реалізації програми; визначення основних фокусів при розробці; визначення оптимальних сфер використання інструментів засобів, спираючись на аналіз процесу розробки системи обліку пацієнтів. Методи, засоби та технології розробки. Додатку було обрано мову програмування C#, середу програмування Microsoft Visual Studio, базу даних PostgreSQL для зберігання даних. Результатом виконання випускної роботи є реалізація додатку для ветеринарної клініки.

РОЗДІЛ 1 ПРЕДМЕТНОЇ ОБЛАСТІ 1.1 Загальні поняття та аспекти 1.1.1 Пацієнт Пацієнт – фізична особа, яка отримує профілактичну, медичну допомогу, або піддається медико-біологічним випробуванням (дослідженням). Пацієнт поняття і потребує подальшого уточнення. У фармації більш широко використовуються терміни: споживач, відвідувач. Він як багатогранний людський фактор, зі зміною пріоритетів суспільства змінюється і пацієнт. 1.1.2 Ветеринарний фахівець з повною вищою медичною освітою, який в установленому законом порядку постійно займається відвідуванням пацієнтів, через профілактики, діагностики та лікування травм і захворювань. Це можливо завдяки знанням фізіології, анатомії, науки медицини, хвороб та лікування, особистому досвіду та практиці. До списку тварин, об'єктів вивчення ветеринаром входять велика та дрібна рогата худоба, коні, кішки, собаки, свині, свійський птах. Вони вивчаються ветеринаром фізіологію, анатомію і біологічну хімію. Основні знання про хвороби дають вірусологія, мікробіологія, патологія, паразитологія. Крім загальних дисциплін, лікар повинен вивчати приватні відомості про профілактику та лікування тварин, вони отримують з терапії, хірургії, паразитології та епізоотології. 1.1.3 Ветеринарна хірургія Ветеринарна хірургія

вивчає способи та правила виконання хірургічних операцій на тваринах. Використовуючи методи та способи вт вирішує конкретні завдання: * Відновлення зниженої або втраченої продуктивності тварини; * Відновлення абс робочих якостей тварини; * Сприяння якнайшвидшому відтворенню стада. Хірурги виконують такі хірургічні про Ендопротезування суглобів; * Відновлення переломів; * Стабілізація дефектів краніальних хрестоподібних зв'язків; операції; * Лікування міжхребцевих трансплантів; * Малоінвазивні процедури та лікування ран. У ветерин; застосовують неоперативні методи лікування тварин з хірургічною патологією, використовуючи обладнання для р фізіотерапії, що дозволяє прискорити одужання тварин. Стерильність у ветеринарії дуже важлива. Стерильність о відсутність будь-яких форм бактерій, що необхідно для проведення операцій та хірургічних втручань. Стерильним все, для попередження мікробного зараження ран. Усі інструменти стерилізуються у сухожаровій шафі температурах або в автоклаві парую під тиском.

1.1.4 Ветеринарна клініка Ветеринарна клініка – лікувально-пр заклад для надання допомоги хворим тваринам на прийомі в спеціалізованому закладі [1]. Вона має штат сп займаються прийомом пацієнтів за записом. Оплата прийому здійснюється після отримання амбулаторної оформлюється в реєстратурі. Сучасна ветеринарна клініка є спеціалізованою лікувально-профілактичною призначеною здійснювати комплекс профілактичних заходів для виявлення хвороб у тварин та надання їм медичн До функцій ветеринарної клініки входить: * Надання першої медичної допомоги; * Проведення лабораторни Передчасне виявлення хвороб тварини. Клініка проводить велику профілактичну роботу та протиепідемічні за здоров'я тварини, виявляє ранню захворюваність та організовує статистичний облік. Впровадження інформаці підтримки надання медичної допомоги у діяльність медичних установ призведе до забезпечення висококваліфіков сучасної та швидкої медичної допомоги тваринам.

1.1.5 Десктопний додаток Десктопний додаток – програма, яка вст на персональному комп'ютері та працює під керуванням операційної системи. Десктопний додаток встановлюєть через спеціальний інсталятор. Десктопні додатки високопродуктивні та зручні для користувачів, зазвичай інтегровані з різними настільними та офісними програмами. Вважається, що десктопні додатки швидше та їх фуні ширше, але це залежить в основному від параметрів комп'ютера. Головною особливістю десктопних додатків (автономної роботи без підключення до Інтернету, але для їх оновлення без ручного перевстановлення необхідне до інтернету. Переваги десктопних додатків: * Функціональність. Широкі можливості для функціоналу, можливіс практично будь-яку ідею. Інтерфейс буде зручним та інтуїтивно зрозумілим для користувача; * Безпека та Кросплатформність. Можливість розробляти під різні операційні системи та мобільні пристрої, зробивши натие Швидкодійність. Можливості десктопних додатків: * Робота без інтернету. Додаток може працювати автономно, з дані в пам'ять комп'ютера; * Робота з інтернетом; * Швидкий запуск. Десктопний додаток запускає системні фай комп'ютера не потребуючи постійні оновлення даних та завантаження параметрів із мережі; * Інтерфейс. Дозволяє інтерфейс під користувача; * Використання додаткової периферії. Десктопні програми можуть мати доступ до всіх підключені та підключаються до комп'ютера.

1.2 Мета програмної системи обліку пацієнтами За допомогою меди стає простіше налагодити медичні, управлінські, адміністративні та економічні потоки робочого процесу з обліку лікарні. Програмні засоби полегшують роботу ветеринарам, керівникам, бухгалтерам та адміністративному Забезпечити ветеринарну лікарню інформаційною системою як заклад медичного призначення, означає поліп лікування та обслуговування пацієнтів, мінімізування кількості паперової документації та автоматизування багато р зі сторони персоналу та ветеринару.

1.3 Основні аспекти при плануванні впровадження програмного продукту лікарні Основні аспекти: * Оцінити технічну підготовку медичного закладу: чи проведено інтернет, чи є комп'ютери та адміністративного персоналу, яка пропускна здатність інтернету, чи відповідає комп'ютер технічним нормам д медичній автоматизованій системі; * Вибрати конкретний різновид автоматизованої системи: слід проаналізувати : які проблеми має розв'язувати розроблена система. Виходячи з цього вже можна вибрати відповідну інформаційну потреб лікарні; * Знайти постачальників та розробників автоматизованої системи, укласти з ними певний договір провести тренінг для медичного персоналу і навчити його користуватися системою. На усіх етапах роботи програмного продукту система повинна працювати в штатному режимі. По роботі системи можуть бути незначні до і адаптація.

1.4 Огляд останніх досліджень і публікацій Система охорони здоров'я потребує сучасних інформаційни Індустріальні масштаби надання медичної допомоги, проблеми якості лікування, стандартизація медичних по бізнес-процеси потребують впровадження комп'ютерних інформаційних технологій. Інформаційні технології відігра роль в оптимізації діяльності та боротьбі з витратами системи охорони здоров'я. Використання інформаційних те: невід'ємною частиною діяльності лікаря. Ветеринарні клініки є одні з найбільш перспективних та швидко розвиваю але проблема розвитку та формування клініки уповільнює зростання ринку ветеринарних послуг.

1.5 Огляд та ана аналогів Розглянемо існуючі аналоги, щоб виявити переваги та недоліки вже розроблених додатків в цій сфері ді VetDesk Додаток, який повністю автоматизує документообіг невеликої ветеринарної клініки. Програма дозво

відмовитися від паперової картотеки, бланків результатів аналізів, призначень та рекомендацій клієнта. Пошук ка або рахунку вихованця займає лічені секунди. Пацієнти залишаються задоволені моментально одержуваними док можна надіслати електронною поштою або надрукувати. Автоматичні нагадування нагадують пацієнту аб ветеринарної клініки про важливі події. Статистика та ведення рахунків допоможе керівнику підприємства завжди фінансових показників. Довідники допоможуть швидко і правильно заповнити потрібні форми. Основні функції т програми [2]: * Картотека пацієнтів; * Облік вакцинацій, товарів та послуг; * Інтегровані нагадування; * База з прийомів; * Запис аналізів та досліджень. Рисунок 1.1 –Додаток «VetDesk» 1.5.2 БИТ:Айболит Додаток для с фінансового, оперативного та складського обліку медичної діяльності та управління взаємовідносинами з ветеринарних клініках. Завдяки електронним медичним карткам можна швидко переглянути результати аналіз пацієнта. Вбудовані шаблони дозволяють ветеринару витратити менше часу на заповнення картки клі «Ветеринарний готель» відображає оформлення та надходження вихованців до стаціонару, контроль призначення т вартості перебування. Розділ «Вакцинація» дозволяє сформувавши нагадування клієнту про майбутні процедури- відображає інформацію про проведені процедури-вакцинації. Основні функції та можливості програми [3]: * перегляду історії хвороб та результатів аналізів у медичній картці пацієнта; * Контроль проведених та майбутн вакцинацій; * Багато шаблонів для оформлення вихованців. Рисунок 1.2 – Програма «БИТ:Айболит» 1.5.3 Мурмот / дозволяє вести облік пацієнта, список ліків та матеріалів, послуг, операцій прийому пацієнта із розрахун обслуговування та можливістю друку. Нагадування про майбутні події. Розрахунок заробітної плати. Основн можливості програми [4]: * Довідник медикаментів, видів порід тварин, послуг у клініці та пов'язані процедури; * Оби медикаментів із нотифікацією про закінчення терміну придатності; * Звіт з витрат медикаментів у прийомах, з і Розрахунок залишків за партіями медикаментів; * Перелік аналізів із зазначенням вартості та врахуванням роздр Рисунок 1.3 – Програма «Мурмот» 1.5.4 VetCliniX Додаток обліку для ветеринарної клініки, надає можливість а ведення обліку та зберігання результатів проведених досліджень у цифровій історії хвороби картотеці пацієнта. Ос та можливості програми [5]: * Цифрова історія хвороби; * Картотека пацієнтів; * Облік та зберігання результаті досліджень. Рисунок 1.4 – Програма «VetCliniX» 1.6 Висновки за розділом Проаналізовано вимоги та тенден пацієнтів та лікарів з використанням помічників. Було прийнято рішення про розробку автоматизованої систем закладу у вигляді прикладної програми під настільні ПК. РОЗДІЛ 2 ІНСТРУМЕНТИ РОЗРОБКИ ДОДАТКУ 2.1 Зас 2.1.1 Мова програмування C# На сьогоднішній день мова C# одна з найпотужніших мов, яка швидко розвивається і в ІТ-галузі. На ній пишуться різні програми від невеликих десктопних до веб-порталів і сервісів. Перша версія мови разом із релізом MS Visual Studio .NET у лютому 2002 року. Поточна версія мови – версія C# 10.0, яка вийшла 2021 року разом із релізом .NET 6. C# – об'єктно-орієнтована мова програмування із Сі-подібним синтаксисом, бли: та Java. C# підтримує успадкування, поліморфізм, статичну типізацію, навантаження операторів. Об'єктно-орієнт дозволяє вирішити з побудови великих, масштабованих, гнучких додатків, що розширюються. Ця мова насамперед для платформи .NET – середовище, що поєднує програмні технології для розробки Windows та Web додатків. мови програмування C# та фреймворку .NET – автоматичне складання сміття. Common Language Runtime ви сміття та очистить пам'ять. Основу середовища .NET становить Common Language Runtime – середовище ви має дві основні частини: * Ядра – набір служб, що управляються завантаженням програм, які зібрані в бібліотеці п Бібліотеки базових класів – забрані в бібліотеціmscorlib.dll. У складі цих бібліотек виділяється Common Typ загальна система типів та підмножина Common Language Specification – загальномовна специфікація, що даних, які підтримуються у всіх мовах .NET. Мова C# спеціально створена для роботи із фреймворком .NET. Фрей потужна платформа для створення додатків. Основні риси платформи .NET: * Підтримка кількох мов. Основа г загальномовне середовище виконання Common Language Runtime. Завдяки їй .NET підтримує кілька мов п поряд з C#: C++, VB.NET, VisualBasic, J#, F#, Jscript та інші. При компіляції код інших мов компілюється загальною мовою Common Intermediate Language. CIL – асемблер платформи .NET. Завдяки цьому можливо зр модулі однієї програми різними мовами програмування; * Кросплатформність. .NET – платформа, що перенос деякими обмеженнями. .NET 6 підтримується на більшості сучасних операційних систем: Window, Linux, використанні різних технологій на платформі .NET є можливість розробляти програми для різних платфк програмування C#: Window, Linux, MacOS, Android, iOS; * Потужна бібліотека класів. .NET представляє єдину бібл Будь-яка програма на мові C# використовує бібліотеку класів .NET; * Різноманітність технологій. Загальномовне виконання (CLR) та базова бібліотека класів – основа стеку технологій. Наприклад, для роботи з базами дани: технологію Entity Framework Core та ADO.NET. Для побудови графічних програм з насиченим інтерфейсом ви технологію WinUI та WPF, для простих графічних – Windows Forms. Для розробки кросплатформових десктопних додатків використовують технологію Xamarin. Для створення веб-додатків та веб-сайтів використовують технолог

Продуктивність. Веб-програми на .NET 6 у ряді категорій випереджають веб-програми побудовані іншими техн компіляція (Just-In-Time). Код C# компілюється в збірки або додатки з розширенням dll або exe мовою CIL. Пр виконання додатка відбувається JIT-компіляція в машинний код, котрий потім виконується. Компілюватися буде л програми до якої йде звернення. Якщо буде звернення до некомпільованої частини, вона буде скомпільована з CIL код. Скомпільована частина програми буде зберігатися до завершення роботи програми. Процес створення прог програмування C# у середовищі .NET зображено у вигляді етапів на рисунку 2.1. Рисунок 2.1 – Етапи створення прс Рисунок 2.2 – Логотип мови програмування C# Відштовхувались від рейтингу на платформі Stack Overflow восьме місце в рейтингу (рисунок 2.3) [6]. Рисунок 2.3 – Популярні мови програмування 2021 року 2.1.2 СУБД PostgreSQL – одна з найпопулярніших об'єктно-реляційних систем управління базами даних. PostgreSQL базує SQL та має безліч додаткових можливостей. Проект PostgreSQL еволюціонував із Ingres – реляційна СУБД, яка під відкритими вихідними текстами. Ingres була створена на початку 1970-х років у Каліфорнійському університ закінчився на початку 1980-х років. Формально розвиток PostgreSQL почався у 1986 році під назвою POSTGRES, і був перейменований на PostgreSQL, що відображало акцент на SQL. PostgreSQL надійна, безпечна та розширк даних, яка має в своєму розпорядженні велику екосистему доступних засобів. Вона сумісна з усіма основними C MacOS, Linux. PostgreSQL підтримує як реляційні, так і нереляційні запити, підтримує текст, зображення, від Переваги PostgreSQL: * Доступ до потужних функцій. PostgreSQL містить багато можливостей для користувача. Мс функції попереджувальне ведення журналу, табличні простору, відновлення на час, елементи деталізованого доступом, вкладені транзакції, багатоваріантне управління паралелізмом та оперативне резервне копіювання; * вимогам та надійність. PostgreSQL відповідає властивостям узгодженості, атомарності, ізолюваності та долгк транзакцій баз даних. Також вона підтримує кілька мов для різних атрибутів, тригерів, зовнішніх ключів, процедур підтримує юнікод, багатобайтове кодування символів та міжнародні кодування; * Ліцензія на програмне заф відкритим кодом. PostgreSQL надається з відкритим кодом, що дає більше гнучкості та можливостей для в інновацій користувачам порівняно з комерційною системою баз даних; * Масштабованість. PostgreSQL може лег великим обсягом даних. Ця масштабованість стосується не тільки обсягу даних, але і кількості одночаснс користувачів у ній; * Повнотекстовий пошук та різноманітні типи індексування. PostgreSQL дає можливість рі індексування, включаючи індексування на основі дерев B+, узагальнене дерево пошуку та узагальнений інверто крім повнотекстового пошуку по рядках і рядків векторних операцій; * Гнучкість. PostgreSQL сумісна з низкою протг програмування: C, C++, Perl, Python, Perl, Java, .NET, Ruby, Tcl та ODBC; * Розвинена екосистема підтримки. Е PostgreSQL забезпечує підтримку спільноти розробників, які постійно удосконалюють систему; * JSON. PostgreSQL нереляційні та реляційні запити, за допомогою виразів шляху JSON та SQL можна отримати доступ до даних JSON; розширення. PostgreSQL дозволяє визначати типи даних та функціональні мови, включаючи типи, що визна настраюються. Рисунок 2.4 – Логотип СУБД PostgreSQL Відштовхувались від рейтингу на платформі Sta PostgreSQL займає друге місце в рейтингу (рисунок 2.5) [6]. Рисунок 2.5 – Популярні СУБД 2021 року pgAdmin – відкритим вихідним кодом для розробки та адміністрування для PostgreSQL. pgAdmin написана на мові програмує та jQuery, вона дозволяє виконувати конфігурування сервера PostgreSQL, обслуговування, завдання моніторингу т SQL-запити. Особливості та можливості pgAdmin: * Відкритий вихідний код; * Безкоштовна; * Можна викор Windows, Linux та macOS; * Розвантаження даних та написання SQL-запитів; * Розробка уявлень, функцій та її бази даних. 2.1.3 Середи програмування Microsoft Visual Studio MS Visual Studio – повнофункціональне середовище розробки (IDE) з підтримкою мов програмування: C, C++, C#, F#, VB.NET, Python, JavaScript, Type Основним розширенням файлу є SLN – Visual Studio Solution File. Visual Studio надає сучасні інструменти др коду, складання, проектування графічних інтерфейсів тестування та налагодження програм, охоплюючи всі ета програмного забезпечення. Шляхом підключення потрібних розширень можна доповнити можливості Visual Stuc Visual Studio: * Підтримує крос-платформну розробку; * Вбудований механізм завантаження та установки NuC проект; * Підтримує безліч мов під час розробки. Visual Studio дозволяє писати код своєю мовою або будь- доступною, використовуючи весь час один інтерфейс (IDE); * Інтуїтивний стиль кодування. Редактор коду підтр фрагментів коду, підсвічування синтаксису, відображення пов'язаних функцій та структури. Технологія IntelliSense істотно прискорити роботу; * Налагоджувач. Використовується для виправлень та пошуку помилок у вихідному код апаратному рівні. Завдяки інструментам діагностики можна оцінити якість коду з погляду використання продуктивності; * Дизайнер форм. Допомагає спроектувати роботу та зовнішній вигляд кожного елемента інтерфей Visual Studio підтримує групову розробку, дозволяючи виконувати спільне налагодження та редагування будь- коду в реальному часі. Як систему управління версіями використовує Git або Team Foundation. Рисунок 2.6 – Ло програмування Microsoft Visual Studio Відштовхувались від рейтингу на платформі Stack Overflow, Microsoft \

займає друге місце в рейтингу (рисунок 2.7) [6]. Рисунок 2.7 – Популярні середовища програмування 2021 року 2.1 WPF

Технологія Windows Presentation Foundation – частина екосистеми платформи .NET та підсистема д графічних інтерфейсів. Перша версія WPF 3.0 вийшла разом із .NET Framework 3.0 та ОС Windows Vista у 200 тоді платформа стала частиною екосистеми .NET та розвивається разом із .NET Framework. WPF підтримує Wi (.NET Framework 3.5), Windows 8 (.NET Framework 4.0 та 4.5), Windows 8.1 (.NET Framework 4.5.1) та Wind Framework 4.7). При створенні програм на основі WinForms за малювання графіки та елементів керування відпов та GDI/GDI+, то WPF засновані на DirectX. Використовуючи WPF значна частина роботи з відтворення граф процесор відеокарти, що дозволяє скористатися апаратним прискоренням графіки. Одна з важливих особлив використання мови декларативної розмітки інтерфейсу XAML. XAML заснований на XML, для створення насиченої інтерфейсу необхідно використовувати декларативне оголошення інтерфейсу або код керованих мов C#, F#, поєднувати. eXtensible Application Markup Language – мова розмітки, яка використовується для ініціаліза технологія на платформі .NET. XAML не обов'язкова частина програми, створювати елементи можливо через код мо компіляції програми у Visual Studio, код xaml-файлів компілюється в бінарне представлення коду Binary Applic Language. А код BAML у фінальне складання програми: dll-файл або exe. Переваги XAML: * Можливість ві інтерфейсу від логіки програми. Надає можливість автономної роботи різними фахівцями: інтерфейс – дизайн програмісти; * Компактність та зрозумілість коду. Особливості WPF: * Усталена технологія; * Доступний кс Підтримується у більшості версіях Windows; * Підтримується в .NET 3.0 та пізніших версіях. Переваги WPF: * Е традиційних мов програмування платформи .NET – C#, VB.NET та F# для створення логіки програми; * декларативного визначення графічного інтерфейсу за допомогою мови розмітки XAML, яка представляє альтернати елементів управління та графіки програми; * Незалежність від роздільної здатності екрана. Всі елементи ви незалежних від пристрою одиницях. Програми легко масштабуються під різні екрани; * Нові можливості. При створення тривимірних моделей, використання стилів, шаблонів, тем та інші; * Взаємодія з WinForms. У додатка: використовувати елементи керування WinForms; * Багато можливостей створення: двовимірна та тривимі мультимедіа, анімації, прив'язка даних, шаблони, стилі, теми, можливість створення нових елементів, великий набі елементів управління; * Апаратне прискорення графіки. Всі компоненти програми транслуються в об'є візуалізуються за допомогою процесора відеокарти, що робить графіку більш плавною та підвищує продуктивність; програм під безліч ОС сімейства Windows. Однак WPF має певні обмеження. Для створення програм з вели тривимірних зображень краще використовувати DirectX або спеціальні фреймворки Monogame або Unity. В програмами на WinForms обсяг програм на WPF та використання пам'яті в процесі роботи дещо вище. Створювати WPF можна тільки під Windows. Архітектура WPF (рисунок 2.8). WPF має два рівні: managed API та un Managed API містить код, який виконується під управлінням CLR. Рисунок 2.8 – Архітектура WPF Managed основний функціонал платформи WPF та складається з: * PresentationFramework.dll – містить елементи у основні реалізації компонентів, які можна використовуватися при побудові графічного інтерфейсу; * Presentati містить базові типи для більшості класів з PresentationFramework.dll; * WindowsBase.dll – містить допоміж застосовуються у WPF. Unmanaged API використовується для інтеграції вишого рівня з DirectX: * milcore.dll інтеграцію компонентів WPF із DirectX; * WindowsCodecs.dll – надає низькорівневу підтримку для зображен останньому рівні знаходяться компоненти ОС та DirectX, які відповідають за візуалізацію компонентів програ низькорівневу обробку. Завдяки низькорівневого інтерфейсу Direct3D відбувається трансляція. Для низки обч завдань, які не включають візуалізацію використовується бібліотека user32.dll.

2.2 Візуальне проектування

Необхідним етапом розробки системи є проектування бази даних. Яка забезпечить зберігання даних та надасть проводити операції над ними. Основні етапи проектування бази даних: * Концептуальне проектування – побудова моделі предметної області. Конкретний зміст та вигляд концептуальної моделі визначається обранням для цього апаратом. Концептуальна модель включає в себе: опис понять предметної області або інформаційних об'єкт проектування – створення схеми бази даних на основі, наприклад, реляційної моделі бази даних. Перетворення ко моделі в логічну модель відбувається за формальними правилами; * Фізичне проектування – створення схеми ба СУБД. Специфіка СУБД може включати в себе обмеження на підтримувані типи даних та іменування об'єктів. На етапі проектування було створено зв'язок між трьома таблицями. На етапі фізичного проектування було підключено б середовища програмування MS Visual Studio та налаштовано зв'язок між базою даних та додатком.

2.3 UML-діаг Modeling Language

– уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого пр невід'ємна частина уніфікованого процесу розробки програмного забезпечення. UML-модель – абстрактна модель використовує графічні позначення. UML необхідний для візуалізації, визначення, документування та проектування систем. Діаграму прецедентів додатку для ветеринарної клініки зображено на рисунку 2.9. Рисунок 2.9 – Діаграме

додатку 2.4 Висновки за розділом Проаналізувавши засоби розробки, було прийнято рішення реалізувати за допомогою мови програмування C# в середовищі програмування MS Visual Studio, для збереження даних буде використано СУБД PostgreSQL.

РОЗДІЛ 3 ПРОГРАМНИЙ МОДУЛЬ 3.1 Проектування системи

На рисунку 3.1.1 зображено прецедентів. Діаграма містить одного актора – користувач. Рисунок 3.1.1 – Діаграма прецедентів Користувач вже зареєстрований та має повний функціонал додатку, що складається з можливості редагування даних клієнтів, редагування картки клієнта, редагування записів відвідувань. Для розробки додатку буде створено п'ять таблиць: авторизація, співробітники, пацієнти та журнал. На рисунку 3.1.2 зображена ER діаграма бази даних. Рисунок 3.1.2 – ER діаграма бази даних користувача із системою зображено на рисунку 3.1.3. Рисунок 3.1.3 – Діаграма активності Після входу у додаток запропоновано пройти авторизацію. Успішна авторизація направляє користувача на головне вікно додатку для вибору дій. Користувач має можливість вибрати один з чотирьох розділів: співробітники, клієнти, пацієнти, журнал. Вибравши розділ користувач може переглянути/додати/змінити/видалити інформації розділу. Діаграма класів проекту зображено на рисунку 3.1.4. Рисунок 3.1.4 – Діаграма класів При відкритті додатку викликається клас «Authorization». Він відповідає за авторизацію в додатку. Після успішної авторизації викликається клас «MainWindow», який відповідає за показ та управління інтерфейсом користувача. Клас «Employees» відповідає за показ списку всіх співробітників ветеринарної лікарні. Клас «Clients» відповідає за показ списку всіх клієнтів. Клас «Cards» відповідає за показ списку всіх пацієнтів ветеринарної лікарні. Клас «Visits» відповідає за показ записів журналу. Дані класи включають клас «InteractionDB», в якому реалізовані запити до бази даних. Також розширює клас помилок «SystemException», щоб відловлювати конкретні помилки додатку та виводити їх у вигляді повідомлення розшифрованою вказуючи користувачеві в чому саме помилка.

3.2 Програмування системи 3.2.1 Основні компоненти багатобайтового проекту

Під час створення додатку було використано багатобайтову систему проекту, компоненти побачити на рисунку 3.2.1. Рисунок 3.2.1 – Компоненти проекту Розглянемо більш детально ці компоненти кожного модуля: * Helpers (Cat-Dog) – призначений для класів, які відстежують помилки у роботі додатку; * Resources – зберігає картинки, іконки та стилі, які були використані у проекті; * View – призначений для всіх вікон проекту; * Helpers (Cat-Dog) – призначений для класів, які допомагають в реалізації моделей у каталозі «DatabaseModels»; * Database – призначений для класів, які використовуються інтерфейсом для відображення даних; * DatabaseTableClasses – призначений для класів, які необхідні для отримання даних з бази даних; * Generated – призначений для використання бібліотеки «PetaPoco».

3.2.2 Розробка та створення бази даних

База даних буде складатися з п'яти таблиць: авторизації, співробітники, клієнти, пацієнти та журнал. Таблиця авторизації повинна мати такі поля: логін та пароль. На лістингу 3.2.1 описано SQL-запит для створення таблиці «Auth» (id integer NOT NULL, login character varying, pass character varying, PRIMARY KEY (id)) Рисунок 3.2.2 Таблиця співробітників повинна мати такі поля: ім'я, прізвище, по батькові, номер телефону. На лістингу 3.2.2 описано SQL-запит для створення таблиці співробітників. Структуру таблиці показано на рисунку 3.2.2 – SQL-запит для створення таблиці «Employees» CREATE TABLE employees (id integer NOT NULL DEFAULT nextval('employees_id_seq'::regclass), name character varying, lname character varying, pname character varying, phone character varying, part character varying, PRIMARY KEY (id)); Рисунок 3.2.3 – Структура таблиці «Clients» Таблиця клієнтів повинна мати такі поля: ім'я, прізвище, по батькові, номер телефону та адреса. На лістингу 3.2.3 описано SQL-запит для створення таблиці клієнтів. Структуру таблиці показано на рисунку 3.2.4. Лістинг 3.2.4 – SQL-запит для створення таблиці «Clients» CREATE TABLE clients (id integer NOT NULL DEFAULT nextval('clients_id_seq'::regclass), name character varying, lname character varying, pname character varying, address character varying, phone character varying, PRIMARY KEY (id)) Рисунок 3.2.4 – Структура таблиці «Cards» Таблиця пацієнтів повинна мати такі поля: номер картки, кличка, стать та вік. На лістингу 3.2.4 описано SQL-запит для створення таблиці пацієнтів. Структуру таблиці показано на рисунку 3.2.4 – SQL-запит для створення таблиці «Cards» CREATE TABLE cards (id integer NOT NULL DEFAULT nextval('cards_id_seq'::regclass), idcard character varying, name character varying, type character varying, gender character varying, age character varying, PRIMARY KEY (id)) Рисунок 3.2.5 – Структура таблиці «Visits» Таблиця журналу повинна мати такі поля: дата, номер картки, лікар та діагноз. На лістингу 3.2.5 описано SQL-запит для створення таблиці журналу. Структуру таблиці показано на рисунку 3.2.6. Лістинг 3.2.5 – SQL-запит для створення таблиці «Visits» CREATE TABLE visits (id integer NOT NULL DEFAULT nextval('visits_id_seq'::regclass), date character varying, doctor character varying, diagnos character varying, PRIMARY KEY (id)) Рисунок 3.2.6 Таблиця журналу

Для підключення бази даних до додатку використовується бібліотека «PetaPoco» та «Npgsql». «Data Provider for PostgreSQL» необхідний для можливості працювати з базою даних PostgreSQL. «PetaPoco» програмування, котра пов'язує концепції об'єктно-орієнтованих мов програмування із базою даних, створюючи об'єктну базу даних». «PetaPoco» необхідний для виконання всіх операцій над таблицями бази даних. Для підключення необхідно додати строку підключення у файл проекту «App.config» (лістинг 3.2.6). Лістинг 3.2.6 – Код для підключення бази даних

Клас «InteractionDB» є з'єднувачем між програмою та базою даних, у якому знаходяться всі запити до бази даних (Лістинг 3.2.7 – Код класу «InteractionDB» public class InteractionDB { public bool CheckAuthorization(string password = "") { return new PetaPoco.Database("DatabaseConnect").Fetch(\$"select * from auth where login = '{login}' and pass = '{password}'").Count > 0 ? true : false; } public List GetListEmployees() { PetaPoco.Database("DatabaseConnect").Fetch("select * from employees"); } public void AddEmployees(string lName, string pName, string phone, string part) { try PetaPoco.Database("DatabaseConnect").Execute(\$"insert into employees(name, lname, pname, phone) values('{name}', '{lName}', '{pName}', '{phone}', '{part}')"); } catch (System.Exception) { } } public void UpdateEmployees(int id, string name, string lName, string pName, string phone, string part) { PetaPoco.Database("DatabaseConnect").Execute(\$"update employees set name = '{name}', lname = '{lName}', pname = '{pName}', phone = '{phone}', part = '{part}' where id = {id}"); } catch (System.Exception) { } } public void DeleteEmployees(int id) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"delete from employees where id = '{id}'"); } catch (System.Exception) { } } public List GetListClients() { return PetaPoco.Database("DatabaseConnect").Fetch("select * from clients"); } public void AddClients(string rName, string lName, string pName, string address, string phone) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"insert into clients(name, lname, pname, address, phone) values('{name}', '{lName}', '{pName}', '{address}', '{phone}')"); } catch (System.Exception) { } } public void UpdateClients(int id, string name, string lName, string pName, string address, string phone) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"update clients set name = '{name}', lname = '{lName}', pname = '{pName}', address = '{address}', phone = '{phone}' where id = {id}"); } catch (System.Exception) { } } public void DeleteClients(int id) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"delete from clients where id = '{id}'"); } catch (System.Exception) { } } public List GetListVisits() { return PetaPoco.Database("DatabaseConnect").Fetch("select * from visits"); } public void AddVisits(string date, string idCard, string doctor, string diagnos) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"insert into visits(date, idcard, doctor, diagnos) values('{date}', '{idCard}', '{doctor}', '{diagnos}')"); } catch (System.Exception) { } } public void UpdateVisits(int id, string date, string idCard, string doctor, string diagnos) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"update visits set date = '{date}', idcard = '{idCard}', doctor = '{doctor}', diagnos = '{diagnos}' where id = {id}"); } catch (System.Exception) { } } public void DeleteVisits(int id) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"delete from visits where id = '{id}'"); } catch (System.Exception) { } } public List GetListCards() { return new PetaPoco.Database("DatabaseConnect").Fetch("select * from cards"); } public void AddCards(string idCard, string name, string type, string breed, string gender, string age) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"insert into cards(idcard, name, type, breed, gender, age) values('{idCard}', '{name}', '{type}', '{breed}', '{gender}', '{age}')"); } catch (System.Exception) { } } public void UpdateCards(int id, string idCard, string name, string type, string breed, string gender, string age) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"update cards set idcard = '{idCard}', name = '{name}', type = '{type}', breed = '{breed}', gender = '{gender}', age = '{age}' where id = {id}"); } catch (System.Exception) { } } public void DeleteCards(int id) { try { new PetaPoco.Database("DatabaseConnect").Execute(\$"delete from cards where id = {id}"); } catch (System.Exception) { } } }

Для отримання кожного списку даних з бази даних необхідно створити відповідні методи. Для таблиці авторизації показано на лістингу 3.2.8. Лістинг 3.2.8 – Код класу таблиці авторизації public class Authorization { public int id { get; set; } public string login { get; set; } public string pass { get; set; } }

3.2.3 Розробка та створення програми. Після запуску програми спочатку повинно відкритися вікно авторизації (рисунок 3.2.7). Для успішної авторизації необхідно ввести логін, пароль та натиснути кнопку «Увійти». Рисунок 3.2.7 – Вікно авторизації. Якщо користувач введе неправильні дані для верифікації, буде викликано вікно помилки, приклад зображено на рисунку 3.2.8. Рисунок 3.2.8 – Приклад вікна помилки.

Лістинг 3.2.9 – Код класу «Errors» class Errors : SystemException { public Errors() : base() { } public Errors(string message) : base(message) { } private void ShowErrorMessage(string error_msg) { System.Windows.MessageBox.Show(error_msg); } public void ErrorMessage() { switch (Message) { case "login_empty": ShowErrorMessage("Поле \"Login\" не повинно бути порожнім"); break; case "pass_empty": ShowErrorMessage("Поле \"Password\" не может бути порожнім"); break; case "err_auth": ShowErrorMessage("Даного користувача не існує"); break; default: break; } } }

Лістинг 3.2.10 – Код блоку try { if (textBoxLogin.Text == "") throw new Helpers.Errors("login_empty"); else if (textBoxPassword.Text == "") throw new Helpers.Errors("pass_empty"); else if (InteractionDB.CheckAuthorization(textBoxLogin.Text, textBoxPassword.Text)) { Main Window mainWindow = new Main Window(); mainWindow.Show(); this.Close(); } else

Режим доступу: www. URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/?view=netdesktop>
Інформаційний портал Metanit С#. [Електронний ресурс]. – Режим доступу: www. URL: <https://metanit.com>
Інформаційний портал Habr С#. [Електронний ресурс]. – Режим доступу: www. URL: <https://habr.com/ru/hub>
Посібник з WPF. [Електронний ресурс]. – Режим доступу: www. URL: <https://metanit.com/sharp/wpf/> 13. Пат
[Електронний ресурс]. – Режим доступу: www. URL: <https://metanit.com/sharp/wpf/22.1.php> 14. Пат
[Електронний ресурс]. – Режим доступу: www. URL: <https://habr.com/ru/post/338518/> 15. PetaPoco. [Електронн
Режим доступу: www. URL: <https://github.com/CollaboratingPlatypus/PetaPoco/wiki> 16. PetaPoco. [Електронн
Режим доступу: www. URL: <https://russianblogs.com/article/35231377149/> 17. PostgreSQL. [Електронний ресу
доступу: www. URL: <https://ru.wikipedia.org/wiki/PostgreSQL> 18. PostgreSQL. [Електронний ресурс]. – Режим д
URL: <https://www.postgresql.org/> 19. PostgreSQL. [Електронний ресурс]. – Режим доступу:
<https://postgrespro.ru/docs/postgresql/9.6/index> 20. pgAdmin 4. [Електронний ресурс]. – Режим доступу
<https://www.pgadmin.org/docs/pgadmin4/development/index.html> 21. XAML. [Електронний ресурс]. – Ре
www. URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-6.0> ДОДАТОК
КОДУ

Source

системне пз системне пз	0.29%
https://kobi.com.ua/21979/#:~:text=%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%CC%81%D0%BC%D0%BD%D0%B5%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%CC%81%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20%E2%80%94%D1%81%D1%83%D0%BA%D1%83%D0%BF%D0%BD%D1%96%D1%81%D1%82%D1%8C%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8,%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%D1%96%D0%B4%D0%BD%D0%B8%D1%85%20%D0%B4%D0%BB%D1%8F%20%D0%B5%D0%BA%D1%81%D0%BF%D0%BB%D1%83%D0%B0%D1%82%D0%B0%D1%86%D1%96%D1%97%20%D1%86%D0%B8%D1%85%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC.	
Комп'ютерні програми для ветеринарів Програма обліку ... Комп'ютерні програми для ветеринарів Програма обліку ...	0.29%
https://bumotors.ru/uk/kompyuternye-programmy-dlya-veterinarov-programma-dlya-ucheta.html	
Feb 12, 2021 — На даний момент на ньому пишуться найрізноманітніші про... Feb 12, 2021 — На даний момент на ньому пишуться найрізноманітніші програми: від невеликих десктопних програм до великих веб-порталів і веб-сервісів, ...	0.29%
http://www.dut.edu.ua/uploads/n_9058_51926054.pdf	

by ЄЮ Нога · 2020 — Загальнономовне середовище виконання CLR і базова ...

0.29%

by ЄЮ Нога · 2020 — Загальнономовне середовище виконання CLR і базова бібліотека класів є основою для цілого стека технологій, які розробники можуть задіяти при.

https://essuir.sumdu.edu.ua/bitstream/123456789/79132/1/Noga_bac_rob.pdf

by СС Зубенко · 2017 — Технологія WPF (Windows Presentation Foundatio...

0.29%

by СС Зубенко · 2017 — Технологія WPF (Windows Presentation Foundation) є частина екосистеми платформи .NET і являє собою підсистему для побудови графічних інтерфейсів.

http://eprints.library.odeku.edu.ua/1527/1/Zubenko_Rozrob_avromat_V_2017.pdf

by ОМ Косяк · 2020 — Одна з найважливіших особливостей являється вик...

0.29%

by ОМ Косяк · 2020 — Одна з найважливіших особливостей являється використання мови декларативної розмітки інтерфейсу XAML, яка в свою чергу засновується на XML.

https://ela.kpi.ua/bitstream/123456789/40913/1/Kosiak_magistr.pdf

Домашня бібліотека - Курсова - 9с. - СтудБаза

0.29%

<https://studbase.com/file/1152/9>

Домашня бібліотека - Курсова - 9с. - СтудБаза

by ОА Іванова · 2020 — Ще нижче власне знаходяться компоненти операц...

0.29%

by ОА Іванова · 2020 — Ще нижче власне знаходяться компоненти операційної системи і DirectX, які здійснюють візуалізацію компонентів програми, або виконують іншу низькорівневу.

https://er.nau.edu.ua/bitstream/NAU/41929/1/%D0%A4%D0%9A%D0%9A%D0%9F%D0%86_2020_122_%D0%86%D0%B2%D0%B0%D0%BD%D0%BE%D0%B2%D0%B0%D0%9E%D0%90.pdf

Мета роботи - Elib LNTU

0.29%

Мета роботи - Elib LNTU

https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%95%D0%9D%D0%9F_%D0%A1%D0%B0%D0%B2%D0%B0%D1%80%D0%B8%D0%BD_%D0%9B%D0%B5%D0%BF%D0%BA%D0%B8%D0%B9/practic/pr9.html

2.2 Перетворення концептуальної моделі в логічну модель ...

0.29%

2.2 Перетворення концептуальної моделі в логічну модель ...

<https://prog.bobrodobro.ru/91896>

nástroj pro konverzi statických webových stránek do kenticos cms

0.29%

nástroj pro konverzi statických webových stránek do kenticos cms

https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=117286



[Home](#)

[Blog](#)

[Testimonials](#)

[About Us](#)

[Privacy](#)

Copyright © 2022 Plagiarism Detector. All right reserved