**ДИПЛОМЕН ПРОЕКТ**

**ТЕМА:**

**РАЗРАБОТВАНЕ НА НАСТОЛНО ПРИЛОЖЕНИЕ ЗА ФИРМА ЗА ЕЛЕКТРОНИКА**

**ДИПЛОМАНТ: Константин Любенов Станоев**

**СПЕЦИАЛНОСТ: *код 4810201 „Системно програмиране“***

**ПРОФЕСИЯ:** ***код 481020 „Системен програмист“***

**РЪКОВОДИТЕЛ-КОНСУЛТАНТ: инж. Иван Киров**

**СОФИЯ**

**2024**

ЗАДАНИЕ

**Раздел I – Теоретична част**

1. **Увод:**

В съвременната дигитална епоха ефективното управление на информацията е от съществено значение за успешното функциониране на бизнеса. Компаниите, занимаващи се с търговия и дистрибуция на компютърна техника, ежедневно обработват големи обеми от данни, свързани с продукти, клиенти, поръчки и доставчици. Тази информация трябва да бъде надеждно съхранявана, лесно достъпна и подлежаща на ефективна обработка, за да се гарантира бързо и безпроблемно обслужване. В този контекст необходимостта от специализирани софтуерни решения, които оптимизират тези процеси, става все по-голяма.

Настоящият дипломен проект е посветен на разработването на настолно приложение за фирма, специализирана в продажбата на компютърна техника, включително **настолни компютри, лаптопи и сървъри**. Основната цел на приложението е да автоматизира ключови бизнес процеси, като осигури интуитивен интерфейс за управление на **продуктовия каталог, клиентските профили, поръчките и доставките**. Чрез използването на **релационна база данни** ще се гарантира структурирано и ефективно съхранение на информацията, а интегрираният потребителски интерфейс ще позволи бърз достъп до всички основни функционалности.

Дигитализацията и автоматизацията на бизнес процесите носят значителни предимства за търговските компании, като подобряват вътрешната организация и повишават конкурентоспособността. Въвеждането на софтуерна система, базирана на релационна база данни, ще даде възможност за **оптимизация на работните процеси, минимизиране на грешките и повишаване на ефективността**. Чрез детайлно анализиране на съществуващите решения и проектиране на функционалностите ще бъде създадена платформа, съобразена със спецификите на фирмата, която ще подобри управлението на ресурсите и ще повиши качеството на предлаганите услуги.

В следващите глави ще бъде направен анализ на съществуващите софтуерни решения, ще се разгледат принципите на релационните бази данни и ще се обоснове изборът на използваните технологии. Освен това ще бъдат представени основните функционалности на разработваното приложение и неговите предимства спрямо алтернативните методи за управление на информацията.

1. **Анализ на съществуващи решения:**

В сферата на търговията с компютърна техника съществуват различни софтуерни решения за управление на продажбите, складовите наличности и клиентските взаимоотношения. Те могат да бъдат категоризирани в три основни групи: **комерсиални ERP системи, персонализирани софтуерни решения и облачни платформи**. В този раздел ще бъде направен сравнителен анализ на тези решения, като се разгледат техните предимства и недостатъци.

**2.1 Комерсиални ERP системи**

Големите предприятия често използват интегрирани ERP (Enterprise Resource Planning) системи като **SAP Business One, Microsoft Dynamics и Oracle NetSuite**. Тези системи предоставят цялостно управление на бизнес процесите, включително инвентаризация, фактуриране, анализи и клиентски взаимоотношения. Основните им предимства включват:

* **Автоматизация на процесите** – улеснява управлението на ресурсите и минимизира човешките грешки.
* **Висока степен на интеграция** – позволява обединяване на данните от различни отдели в компанията.
* **Разширени аналитични възможности** – предлагат инструменти за бизнес разузнаване (BI) и детайлни отчети.

Въпреки предимствата си, тези системи имат някои съществени недостатъци, като **висока цена на внедряване, сложност при настройка и необходимост от обучение на персонала**. Това ги прави по-подходящи за големи компании, но затруднява използването им в малки и средни предприятия.

**2.2 Персонализирани софтуерни решения**

Много компании предпочитат да разработят собствени софтуерни решения, съобразени с конкретните им бизнес процеси. Такива системи осигуряват:

* **Гъвкавост и персонализация** – възможност за адаптиране към специфичните нужди на бизнеса.
* **По-добър контрол върху функционалностите** – системата може да бъде проектирана така, че да отговаря напълно на фирмените изисквания.
* **Възможност за интеграция със съществуващи системи** – позволява лесно свързване с други софтуерни решения, които фирмата вече използва.

Основният недостатък на персонализираните решения е **високата първоначална инвестиция и необходимостта от постоянна поддръжка**, което изисква IT специалисти с експертни познания. Това може да ограничи достъпа на по-малки фирми до подобен тип софтуер.

**2.3 Облачни платформи**

Облачните платформи като **Salesforce, Zoho Inventory и Odoo** предлагат уеб-базирани решения за управление на продажбите и складовите наличности. Те се характеризират със следните предимства:

* **Лесен достъп от всяко устройство** – не изискват инсталация на конкретен хардуер и могат да се използват от всяка точка на света.
* **Автоматични актуализации** – потребителите получават най-новите версии на софтуера без необходимост от ръчна намеса.
* **Гъвкавост и мащабируемост** – компаниите могат да разширяват функциите според нуждите си.

Сред основните недостатъци на облачните решения са **необходимостта от постоянна интернет връзка, рисковете за сигурността на данните и абонаментните такси**, които в дългосрочен план могат да увеличат разходите за софтуера.

**2.4 Избор на подходящо решение**

Въз основа на анализа на съществуващите решения се установява, че **персонализираната разработка на настолно приложение с релационна база данни** е най-подходящият вариант за фирмата, за която се създава настоящият дипломен проект. Това решение съчетава **висока степен на контрол, сигурност и адаптивност** към специфичните нужди на бизнеса, без да налага високи разходи за абонаменти или сложна интеграция с външни ERP системи.

В следващите глави ще бъдат разгледани основите на релационните бази данни, архитектурата на разработваното приложение и използваните технологии.

1. **Основи на релационните бази данни**

Релационните бази данни (RDB) са един от най-разпространените методи за съхранение, управление и извличане на данни в съвременните информационни системи. Те се базират на **релационния модел**, който е разработен от **Едгар Код** през 1970 г. и предоставя ефективен начин за организиране на данните чрез таблици, наречени релации. В този раздел ще бъдат разгледани основните принципи на релационните бази данни, тяхната структура и основните механизми за работа с тях.

**3.1 Основни концепции на релационните бази данни**

Релационният модел се основава на няколко ключови концепции:

* **Таблици (релации)** – основната структура за съхранение на данни, която съдържа редове (записи) и колони (атрибути).
* **Атрибути** – колони в таблицата, които определят характеристиките на съхраняваните данни.
* **Записи (редове)** – отделните единици информация в таблицата.
* **Първичен ключ (Primary Key)** – уникален идентификатор за всеки запис в таблицата.
* **Чужд ключ (Foreign Key)** – атрибут, който създава връзка между две таблици.
* **Нормализация** – процесът на структуриране на базата данни с цел елиминиране на излишните данни и предотвратяване на аномалии при обработка.
* **Схема на базата данни** – логическата структура, която дефинира таблиците, техните атрибути, връзките между тях и наложените ограничения.
* **Ограничения на данните (Constraints)** – правила, които гарантират целостта на базата, като уникалност на ключовете, задължителни стойности и референтна цялост.

**3.2 Предимства на релационните бази данни**

Релационните бази данни предлагат множество предимства, които ги правят предпочитан избор за бизнес приложения:

* **Гъвкавост и структурираност** – позволяват ясна организация и ефективно управление на данните.
* **Консистентност и целостност** – чрез прилагане на релационни зависимости и ограничения се гарантира точността и надеждността на информацията.
* **Мощни инструменти за заявки** – SQL (Structured Query Language) осигурява лесен и ефективен начин за извличане и обработка на данни.
* **Сигурност** – предоставят възможност за контрол на достъпа и защита на информацията чрез механизми като ролево управление и криптиране.
* **Ефективност при обработка на данни** – релационните бази данни предлагат оптимизирани индекси, които ускоряват извличането и обработката на информацията.
* **Мащабируемост** – позволяват разширяване на базата, без да се нарушава нейната функционалност, което е важно за нарастващи компании.

**3.3 Основни операции в релационните бази данни**

Релационните бази данни използват **SQL (Structured Query Language)** като стандартен език за работа с данни. Основните SQL операции включват:

* **SELECT** – извличане на данни от една или повече таблици.
* **INSERT** – добавяне на нови записи в таблицата.
* **UPDATE** – модифициране на съществуващи записи.
* **DELETE** – изтриване на записи от таблицата.
* **JOIN** – свързване на данни от различни таблици въз основа на общи атрибути.
* **GROUP BY** – групиране на данни по определени критерии.
* **HAVING** – филтриране на резултатите след групиране.
* **ORDER BY** – сортиране на резултатите по зададен критерий.

**3.4 Нормализация и оптимизация на базите данни**

Процесът на **нормализация** е съществен елемент в проектирането на релационни бази данни. Той включва няколко нормални форми (NF), които намаляват излишъците на данни и подобряват целостта на базата. Основните нормални форми включват:

* **Първа нормална форма (1NF)** – елиминира многозначните атрибути и гарантира, че всички колони съдържат атомарни стойности.
* **Втора нормална форма (2NF)** – премахва частичните зависимости чрез отделяне на данните в различни таблици.
* **Трета нормална форма (3NF)** – елиминира транзитивните зависимости между атрибутите.
* **Четвърта и пета нормална форма (4NF, 5NF)** – допълнително намаляват дублирането на данни и подобряват структурата на базата.

Освен нормализацията, друга важна техника за подобряване на ефективността е **индексирането**, което ускорява заявките чрез създаване на допълнителни структури за бърз достъп до информацията.

**3.5 Релационни бази данни в контекста на проекта**

В рамките на настоящия дипломен проект релационната база данни ще бъде използвана за управление на информацията, свързана с **продукти, клиенти, поръчки и доставки**. Изграждането на добре структуриран модел ще гарантира ефективното функциониране на приложението, като осигури:

* **Бърз достъп до данните** чрез оптимизирани заявки.
* **Гарантирана целостност на информацията** чрез използване на ключове и ограничения.
* **Лесна разширяемост** – възможност за добавяне на нови функционалности при необходимост.
* **Сигурност и защита на данните** чрез ролево управление и ограничения за достъп.
* **Оптимизирани заявки и ефективна обработка на големи обеми данни** чрез индексиране и правилно структуриране на базата.

В следващите раздели ще бъдат разгледани архитектурата на системата, избраните технологии и методите за оптимизация на базата данни.

1. **Технологии за разработка**

Съвременната софтуерна разработка изисква внимателен избор на технологии, които осигуряват висока производителност, сигурност и мащабируемост на системата. Изборът на инструменти и среди за разработка е критично важен за постигането на успешен и ефективен софтуерен продукт. В този раздел ще бъдат разгледани основните технологии, които ще бъдат използвани в рамките на дипломния проект, включително програмни езици, бази данни, рамки за разработка и среди за разгръщане.

**4.1 Програмни езици**

Изборът на програмния език зависи от изискванията на проекта и неговата функционалност. В настоящия проект ще бъдат използвани следните езици:

* **Java** – един от най-популярните езици за уеб и сървърни приложения, осигуряващ добра поддръжка за обектно-ориентирано програмиране и стабилност.
* **JavaScript** – основен език за уеб разработка, използван за създаване на динамични и интерактивни уеб интерфейси.
* **SQL** – езикът за управление на релационни бази данни, използван за съхранение и извличане на информация.

**4.2 Рамки и среди за разработка**

Използването на утвърдени рамки улеснява и ускорява разработката на софтуерни приложения. В проекта ще бъдат използвани следните:

* **Spring Boot** – мощна Java рамка за създаване на уеб приложения, която улеснява конфигурирането и управлението на сървърната част.
* **React.js** – библиотека за създаване на потребителски интерфейси, базирана на компонентен модел и осигуряваща висока производителност.
* **Hibernate** – ORM (Object-Relational Mapping) инструмент за работа с бази данни в Java приложения.

**4.3 Бази данни**

За съхранение и управление на данните ще бъде използвана **релационна база данни**, тъй като тя предоставя структурирано и надеждно съхранение на информация. В проекта ще бъде използвана:

* **MySQL** – мощна и широко използвана релационна база данни, която предлага висока производителност, мащабируемост и сигурност.

**4.4 Среда за разработка и инструменти**

За ефективно разработване и поддръжка на проекта ще бъдат използвани следните среди и инструменти:

* **IntelliJ IDEA** – интегрирана среда за разработка (IDE) за Java, която предлага мощни инструменти за кодиране и дебъгване.
* **Visual Studio Code** – лека и гъвкава среда за разработка на JavaScript и React приложения.
* **Git и GitHub** – за контрол на версиите и съвместна работа по кода.
* **Postman** – инструмент за тестване на API заявки и отговори.
* **Docker** – за контейнеризация и лесно разгръщане на приложенията в различни среди.

**4.5 Разгръщане и хостинг**

След завършване на разработката системата ще бъде разположена в облачна среда, осигуряваща гъвкавост и надеждност. Основните решения за хостинг включват:

* **AWS (Amazon Web Services)** – мащабируема и надеждна облачна инфраструктура.
* **Heroku** – платформа за бързо разгръщане на уеб приложения без сложна конфигурация.
* **Docker Compose и Kubernetes** – инструменти за управление и автоматизация на контейнери.

Използването на тези технологии ще осигури стабилност, гъвкавост и добра производителност на разработваната система. Те ще позволят лесна интеграция, поддръжка и бъдещо разширяване на функционалността.

1. **Архитектура на системата**

Архитектурата на софтуерната система определя начина, по който различните ѝ компоненти взаимодействат помежду си и с външни услуги. Проектирането на добре структурирана архитектура е от съществено значение за осигуряването на мащабируемост, сигурност и лесна поддръжка. В този раздел ще бъде разгледана архитектурната концепция на разработваната система, както и основните компоненти и техните връзки.

**5.1 Архитектурен модел**

Системата ще бъде изградена по модела **клиент-сървър**, който осигурява ясна организация и разпределение на отговорностите между отделните части. Използваният подход е **многослойна архитектура (multi-tier architecture)**, състояща се от:

1. **Представителен слой (Frontend)** – този слой осигурява графичния интерфейс и взаимодействието с потребителя.
2. **Сървърен слой (Backend)** – обработва бизнес логиката, заявките към базата данни и комуникацията между компонентите.
3. **Слой за съхранение на данни (Database)** – управлява и съхранява всички структурирани данни в системата.
4. **Слой за сигурност и удостоверяване (Security Layer)** – осигурява контрол на достъпа, удостоверяване на потребителите и защита от атаки.

**5.2 Описание на компонентите**

**5.2.1 Потребителски интерфейс (Frontend)**

Потребителският интерфейс ще бъде реализиран с помощта на **React.js**, което позволява изграждането на динамични уеб приложения. Основни характеристики:

* Компонентно-базиран подход за лесна поддръжка и преизползваемост.
* Взаимодействие със сървърната част чрез REST API.
* Адаптивен (responsive) дизайн за удобство при използване на различни устройства.
* Използване на **Redux** за централизирано управление на състоянието.
* Оптимизация чрез **lazy loading** за по-добра производителност.

**5.2.2 Сървърна част (Backend)**

Сървърната част ще бъде разработена с **Spring Boot**, осигуряващ лесно управление на заявки, сигурност и ефективност. Основни характеристики:

* Използване на **RESTful API** за комуникация между клиента и сървъра.
* Вградена система за удостоверяване и оторизация.
* Логически обработва заявки, комуникира с базата данни и изпраща отговори към клиентите.
* Използване на **Spring Security** за допълнителна защита на данните.
* Поддръжка на **WebSockets** за осигуряване на реално време при комуникация.

**5.2.3 База данни (Database)**

Системата ще използва **MySQL** като основна релационна база данни, където ще се съхраняват всички необходими данни. Основни характеристики:

* Използване на **ORM Hibernate** за по-лесна работа с данните.
* Оптимизация на заявки чрез индексиране и нормализация на базата.
* Резервни копия и репликация за осигуряване на надеждност.
* Използване на **Stored Procedures и Triggers** за автоматизация на бизнес логиката.

**5.3 Взаимодействие между компонентите**

Общуването между различните слоеве ще се осъществява чрез:

* **HTTP протокол и REST API** – основен метод за обмен на данни между клиента и сървъра.
* **JSON форматиране** – стандартен формат за представяне на данни в заявките и отговорите.
* **JWT (JSON Web Token)** – използван за удостоверяване на потребителите и контрол на достъпа.
* **GraphQL** – алтернативен метод за заявки към сървъра, осигуряващ гъвкавост при извличане на данни.

**5.4 Разгръщане и скалируемост**

За осигуряване на надеждност и ефективност системата ще бъде разположена в облачна среда с помощта на **Docker** и **Kubernetes**. Основни предимства:

* Лесно мащабиране на сървърните услуги при увеличаване на потребителите.
* Изолиране на различните компоненти в контейнери.
* Автоматично управление на натоварването чрез балансировчици на трафик.
* **CI/CD (Continuous Integration/Continuous Deployment)** – автоматизирано разгръщане на нови версии.
* **Load Balancer** – разпределение на натоварването между няколко сървъра за подобряване на производителността.

**5.5 Заключение**

Проектираната архитектура на системата гарантира сигурност, мащабируемост и надеждност. Чрез използването на утвърдени технологии и добри практики, системата ще бъде лесна за разширяване и поддръжка, осигурявайки стабилна работа в дългосрочен план. Освен това, включването на механизми за мониторинг и логване ще осигури детайлно проследяване на грешки и ефективно управление на ресурсите.

1. **Функционални характеристики на приложението**

В този раздел ще бъдат представени основните функционалности на разработваното приложение. Те са проектирани с цел осигуряване на удобство, ефективност и сигурност при използване на системата.

**6.1 Основни функции**

Приложението предлага следните основни функционалности:

* **Регистрация и вход на потребители** – потребителите могат да създадат профил, да влизат в системата чрез потребителско име и парола, както и да възстановяват забравени пароли.
* **Управление на потребители** – възможност за добавяне, редактиране и изтриване на потребители с различни роли и права на достъп.
* **Преглед и управление на продукти** – администраторите могат да добавят, редактират и изтриват продукти, като всеки продукт има описание, цена и наличност.
* **Създаване и управление на поръчки** – потребителите могат да добавят продукти в количка, да правят поръчки и да проследяват статуса им.
* **Филтриране и търсене на продукти** – системата предоставя различни възможности за филтриране по категории, цена, наличност и други критерии.
* **Генериране на отчети и анализи** – администраторите имат достъп до различни справки и статистики за продажби, потребителска активност и други.
* **Система за уведомления** – потребителите ще получават известия за статус на поръчки, нови продукти и други важни събития.

**6.2 Допълнителни функции**

Освен основните функционалности, приложението включва и допълнителни възможности за повишаване на потребителското изживяване:

* **Поддръжка на няколко езика** – възможност за избор на език на интерфейса.
* **Рейтинг и коментари за продукти** – потребителите могат да оценяват и коментират продуктите.
* **Интеграция с платежни системи** – възможност за онлайн разплащане чрез различни методи като банкови карти и електронни портфейли.
* **Роли и права за достъп** – системата ще разграничава права на обикновени потребители, администратори и модератори.
* **Автоматизирани имейл известия** – известия при успешна регистрация, потвърждение на поръчка и промяна на статуса й.

**6.3 Сигурност и защита**

За осигуряване на безопасност на данните и защитен достъп до системата ще бъдат внедрени следните мерки:

* **Шифроване на пароли** – използване на алгоритми като bcrypt за съхранение на паролите.
* **Защита срещу атаки** – предпазване от SQL инжекции, CSRF и XSS атаки.
* **Регулярни архиви на базата данни** – осигуряване на резервни копия за възстановяване при загуба на данни.
* **Логове на потребителска активност** – събиране на данни за влизания, промени в информацията и други действия за проследяване и сигурност.

**6.4 Заключение**

Функционалните характеристики на приложението гарантират гъвкавост, удобство и сигурност при работа. Чрез внедряването на тези функции се осигурява висока ефективност както за крайните потребители, така и за администраторите на системата.

1. **Заключение**

Разработването на софтуерни системи изисква внимателно планиране, анализ и прилагане на най-добрите практики в областта. В настоящата дипломна работа беше представен процесът на проектиране и изграждане на уеб-базирано приложение, като бяха разгледани всички ключови аспекти на системата – от концепцията и архитектурата до функционалностите и сигурността.

В хода на разработката бяха анализирани съществуващите решения и бяха избрани технологии, които осигуряват ефективност, мащабируемост и сигурност. Системата е изградена на базата на многослойна архитектура, като са внедрени методи за защита на данните и осигуряване на безпроблемна работа за крайните потребители.

Основните функционалности на приложението включват управление на потребители, продукти и поръчки, както и възможност за търсене, филтриране и анализ на информация. Освен това са интегрирани допълнителни функции като известия, ролеви модели и защита срещу злонамерени атаки. Всички тези компоненти гарантират, че системата е устойчива, сигурна и лесна за използване.

Разработената система предоставя стабилна основа за бъдещо разширяване и надграждане. Възможностите за подобрения включват интеграция с мобилни приложения, разширяване на аналитичните инструменти и внедряване на изкуствен интелект за по-добро потребителско изживяване.

В заключение, създаденото приложение отговаря на поставените цели и демонстрира ефективно решение в сферата на електронните услуги. То предлага надежден инструмент за управление и оптимизиране на бизнес процесите, като същевременно осигурява високо ниво на сигурност и удобство за крайните потребители.