Департамент образования и науки Костромской области

Областное государственное бюджетное профессиональное образовательное учреждение «Костромской энергетический техникум имени Ф.В. Чижова»

(ОГБПОУ «КЭТ имени Ф.В. Чижова»)

**КУРСОВАЯ РАБОТА**

Разработка информационной системы для

**БАНКОВСКИХ ОПЕРАЦИЙ**

ПМ.06 Сопровождение информационных систем

МДК.06.03 Устройство и функционирование информационной системы

**специальность:** 09.02.07 «Информационные системы и программирование»

Студент **\_\_\_\_\_\_\_\_\_\_\_\_***\_\_\_\_\_\_\_\_\_* \_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_.2023г.

*ФИО Подпись*

Оценка выполнения и защиты курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_

Руководитель Бессараб С.К. **\_\_\_\_\_\_\_\_\_\_\_\_** \_\_.\_\_.2023 г.

*Подпись*

Кострома 2023

**Оглавление**

Оглавление

[ГЛОССАРИЙ 3](#_Toc137127058)

[ВВЕДЕНИЕ 4](#_Toc137127059)

[ГЛАВА 1. ОБОСНОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ 6](#_Toc137127060)

[1.1 Обоснование необходимости внедрения и использования информационной системы 6](#_Toc137127061)

[1.2 Общие требования к информационной системе 7](#_Toc137127062)

[1.3 Краткие сведения об информационной системе 10](#_Toc137127063)

[1.4 Разработка технического задания на разработку информационной системы 11](#_Toc137127064)

[Техническое задание 13](#_Toc137127065)

[1. Введение 13](#_Toc137127066)

[ГЛАВА 2. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ 16](#_Toc137127067)

[2.1 Анализ требований к базе данных 16](#_Toc137127068)

[Анализ требований БД 16](#_Toc137127069)

[2.3 Запросы к информационной системе 18](#_Toc137127070)

[ГЛАВА 3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ 20](#_Toc137127071)

[3.1. Разработка кода информационной системы 20](#_Toc137127072)

[ГЛАВА 4. ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ. 24](#_Toc137127073)

[4.1. План тестирования 24](#_Toc137127074)

[4.2. Написание тест-кейсов 26](#_Toc137127075)

[4.3 Юнит-тестирование 28](#_Toc137127076)

# ГЛОССАРИЙ

*Автоматизация* - процесс автоматической обработки данных и выполнения операций без участия человека.

*Анализ* - процесс изучения и оценки существующих процессов банковских операций с целью их оптимизации.

*Банковские операции* - операции, связанные с денежными средствами, проводимые банками и их клиентами, включая открытие и ведение счетов, переводы, платежи и другие финансовые операции.

*Гипотеза* - предварительное предположение о возможности решения задачи по автоматизации процессов банковских операций.

*Заказчик* - лицо или организация, которая заказывает разработку информационной системы для банковских операций.

*Объект исследования* - банковская деятельность организаций, связанная с проведением банковских операций.

*Программное решение* - комплекс программных средств, предназначенных для автоматизации процессов банковских операций.

*Предмет исследования* - способы и методы автоматизации процессов банковских операций.

*Разработка* - процесс создания программного решения для автоматизации процессов банковских операций.

*Тестирование* - процесс проверки программного решения на соответствие требованиям заказчика и на выявление возможных ошибок и недостатков.

*Экономическое явление* - событие или процесс, связанный с производством, распределением и потреблением материальных и нематериальных благ.

# ВВЕДЕНИЕ

Банковские операции являются неотъемлемой частью современной финансовой системы. Они включают в себя управление счетами, обработку платежей, кредитование, инвестирование и другие финансовые транзакции. Сегодня банки сталкиваются с растущими требованиями клиентов и строгими правовыми нормами, поэтому актуальность автоматизации банковских операций непрерывно возрастает.

**Автоматизация банковских** операций позволяет банкам справиться с растущими требованиями, улучшить эффективность и надежность своих операций, а также обеспечить более высокий уровень безопасности. Целью данного курсового проекта является изучение и обоснование объекта автоматизации банковских операций.

**Объектом исследования** являются банковские операции, включающие в себя различные финансовые транзакции и процессы. **Предметом исследования** является автоматизация этих операций с помощью информационных систем и технологий.

В рамках данного исследования будут рассмотрены бизнес-требования, пользовательские требования, функциональные и нефункциональные требования, а также анализ существующих технологий и решений, применимых для автоматизации банковских операций.

**Задачи исследования** включают анализ требований, определение ключевых аспектов автоматизации, рассмотрение преимуществ и ограничений различных технологий, а также формулирование рекомендаций по внедрению системы автоматизации банковских операций.

1. Проанализировать требования;
2. Спроектировать архитектуру;
3. Разработать БД;
4. Разработать UI;
5. Протестировать;
6. Внедрить систему;
7. Поддерживать и сопровождать.

**Вывод:**

В заключение можно отметить, что автоматизация банковских операций является важным направлением развития финансовых учреждений. Банки стремятся улучшить свою конкурентоспособность, обеспечивая более эффективные, безопасные и удобные финансовые услуги своим клиентам.

Исследование объекта автоматизации банковских операций позволяет определить ключевые требования и аспекты, которые необходимо учесть при разработке и внедрении системы автоматизации. Анализ бизнес-требований, пользовательских требований, функциональных и нефункциональных требований позволяет сформулировать рекомендации и решения, способствующие эффективной автоматизации банковских операций

# ГЛАВА 1. ОБОСНОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ

## 1.1 Обоснование необходимости внедрения и использования информационной системы

В этом разделе будет проведен анализ текущей ситуации в банковской сфере и обоснована необходимость внедрения автоматизации банковских операций. Ниже представлено более подробное описание этого раздела:

* Описание современной банковской сферы и ее особенностей: В этой части будет рассмотрена современная банковская сфера, ее функции, услуги и процессы. Будут выделены основные характеристики банковских операций, которые требуют автоматизации.
* Анализ текущих процессов и проблем, существующих в банковской автоматизации: В данном разделе будут рассмотрены текущие процессы, связанные с банковскими операциями, и выявлены проблемы, с которыми сталкиваются банки. Это могут быть длительные сроки обработки заявок, высокий уровень ошибок в данных, неэффективное использование ресурсов и другие проблемы, которые могут быть решены с помощью автоматизации.
* Выявление преимуществ и потенциальных выгод от внедрения автоматизации: В этой части будет проведен анализ преимуществ, которые может принести автоматизация банковских операций. Это может включать сокращение времени обработки операций, повышение точности и надежности данных, снижение затрат на обслуживание клиентов, улучшение уровня обслуживания и другие выгоды. Будет также рассмотрено влияние автоматизации на конкурентоспособность банков и их способность адаптироваться к изменяющимся рыночным условиям.

## 1.2 Общие требования к информационной системе

В этом разделе будут определены общие требования к системе автоматизации банковских операций. Вот более подробное описание этого раздела:

**Бизнес-требования**

Бизнес-требования отражают потребности и цели бизнеса, которые должны быть учтены при разработке и внедрении системы автоматизации банковских операций. Включение бизнес-требований поможет достичь следующих целей:

Улучшение операционной эффективности: Система должна обеспечивать оптимизацию банковских процессов и сокращение времени обработки операций. Это позволит снизить издержки, повысить производительность и обеспечить более эффективное использование ресурсов.

Расширение возможностей обслуживания клиентов: Система должна предоставлять широкий спектр банковских услуг и функций, позволяющих клиентам осуществлять различные операции с легкостью и удобством. Это включает возможность онлайн-банкинга, мобильного приложения, обработку платежей и другие функции, удовлетворяющие потребности клиентов.

**Пользовательские требования**

Пользовательские требования определяют потребности и ожидания пользователей, которые будут использовать систему автоматизации банковских операций. Рассмотрение пользовательских требований поможет обеспечить удобство и удовлетворение потребностей пользователей. Важные аспекты пользовательских требований включают

Простота использования: Система должна быть интуитивно понятной и легкой в использовании для различных категорий пользователей, включая клиентов и банковских сотрудников. Интерфейс должен быть понятным, наглядным и иметь удобную навигацию.

Персонализация и адаптивность: Система должна предоставлять возможность настройки и персонализации функциональности в соответствии с потребностями каждого пользователя. Также важно, чтобы система была адаптивной и могла работать на различных устройствах и платформах.

**Функциональные требования**

Функциональные требования определяют конкретные функции и возможности, которые должна предоставлять система автоматизации банковских операций. Важные функциональные требования могут включать:

Управление счетами: Система должна обеспечивать возможность открытия, закрытия и управления банковскими счетами, включая операции по пополнению, снятию средств, переводам и просмотру истории операций.

Обработка платежей: Система должна обеспечивать безопасную и эффективную обработку платежей, включая переводы между счетами, между банками и между странами, а также автоматическую обработку платежных поручений и платежных карточек.

Управление кредитами: Система должна поддерживать управление кредитами и займами, включая расчет процентов, погашение кредитов, установление графиков платежей и просмотр информации о задолженности.

**Нефункциональные требования**

Нефункциональные требования определяют ограничения и критерии качества, которые должны быть учтены при разработке и внедрении системы автоматизации банковских операций. Важные нефункциональные требования могут включать:

Безопасность: Система должна обеспечивать высокий уровень безопасности данных и транзакций, включая защиту от несанкционированного доступа, шифрование данных и обеспечение конфиденциальности клиентов.

Производительность: Система должна быть способной обрабатывать большой объем данных и операций с высокой скоростью и отзывчивостью. Она должна поддерживать высокую производительность, даже при интенсивном использовании и возможных пиковых нагрузках.

Масштабируемость: Система должна быть масштабируемой, чтобы справиться с ростом объема данных и количества пользователей. Она должна быть способна масштабироваться горизонтально (добавление дополнительных серверов) и вертикально (улучшение аппаратного обеспечения).

## 1.3 Краткие сведения об информационной системе

|  |  |
| --- | --- |
| Название | Описание |
| Цель ИС | Оптимизация процессов банковских операций |
| Задачи ИС | Автоматизация операций, обработка заявлений, управление базой данных клиентов и счётов |
| Функции ИС | выдача аккаунтов в банке, автоматическое оповещение операций |
| Преимущества ИС | Ускорение банковских операций, уменьшение ошибок и затрат на персонал, повышение качества обслуживания клиентов |
| Ограничения ИС | Необходимость регулярного обновления базы данных, потенциальная уязвимость для кибератак, возможные ошибки в автоматической обработке операций |
| Технологии | Использование баз данных для хранения информации о клиентах и счётах, программное обеспечение для автоматической обработки заявлений, возможно использование искусственного интеллекта для оптимизации процессов |
| Выгоды для клиентов | Более быстрое и удобное оформление счёта, возможность получения подарков |
| Выгоды для компании | Снижение затрат на персонал и бумажную работу, увеличение конкурентоспособности компании на рынке банков |
| Требования к ИС | Надежность и безопасность, высокая производительность, удобный и интуитивно понятный пользовательский интерфейс, возможность интеграции с другими системами, наличие технической поддержки и обновлений. |
| Технологии | Использование баз данных для хранения информации о клиентах и счётах, программное обеспечение для автоматической обработки операций, возможно использование искусственного интеллекта для оптимизации процессов |

## Таб. 1 Краткие сведения об информационной системе

## Разработка технического задания на разработку информационной системы

Техническое задание (ТЗ) - это документ, который содержит требования и условия, необходимые для разработки, тестирования и внедрения информационной системы.

Разработка ТЗ осуществляется в соответствии с документами, которые устанавливают цели и задачи проекта, в том числе - бизнес-планом, техническим проектом, макетами интерфейса и т.д.

Этапы разработки ТЗ включают в себя: анализ требований к ИС, определение ее функциональных и нефункциональных характеристик, создание структуры базы данных, разработку архитектуры системы, проектирование интерфейса пользователя, определение требований к безопасности и качеству, тестирование и сопровождение системы.

**Вывод:**

Глава 1 обосновывает необходимость автоматизации банковских операций через анализ текущей ситуации в банковской сфере и выявление проблем, с которыми сталкиваются банки. Приводятся преимущества и потенциальные выгоды от внедрения автоматизации, а также определяются общие требования к информационной системе для автоматизации банковских операций. Этот раздел является важной основой для понимания необходимости и целей автоматизации банковских процессов.

## Техническое задание

Техническое задание на разработку информационной системы по банковским операциям

### Введение

* 1. **Цель и задачи проекта**
* Целью проекта является разработка информационной системы (ИС) для автоматизации банковских операций. Основные задачи проекта включают:
* Анализ требований и особенностей банковских операций.
* Проектирование и разработка функциональных модулей ИС, учитывающих бизнес-процессы и потребности пользователей.
* Интеграция ИС с существующими системами и базами данных.
* Тестирование, отладка и внедрение ИС в банковскую среду.
* Обеспечение безопасности и защиты данных в рамках ИС.

1. **Основные требования к ИС**

* Полная автоматизация основных банковских операций, включая управление счетами, обработку платежей, кредитование и инвестирование.
* Интеграция с национальными и международными платежными системами для обеспечения быстрой и надежной обработки платежей.
* Удобный и интуитивно понятный пользовательский интерфейс, обеспечивающий легкость в использовании для различных категорий пользователей, включая клиентов банка и сотрудников.
* Высокий уровень безопасности данных, включая шифрование информации, контроль доступа и механизмы защиты от несанкционированного доступа.
* Гибкость и масштабируемость системы, позволяющая адаптироваться к изменяющимся требованиям банка и расширять функциональность в будущем.

1. **Технические требования к ИС:**

* Разработка на современных технологиях и языках программирования, обеспечивающих высокую производительность и надежность.
* Использование баз данных для хранения и обработки информации о клиентах, операциях, счетах и других релевантных данных.
* Совместимость с различными операционными системами и аппаратным обеспечением для обеспечения гибкости и доступности системы.
* Резервное копирование и восстановление данных для обеспечения безопасности и надежности системы.
* Интеграция с другими системами и сервисами, такими как системы электронных платежей, системы учета и отчетности, а также с внешними партнерами и регуляторами.

1. **Ожидаемые результаты**

* Разработка и внедрение функциональной и надежной информационной системы, полностью автоматизирующей банковские операции.
* Улучшение эффективности и точности операций, сокращение времени выполнения транзакций и обработки запросов клиентов.
* Повышение уровня обслуживания клиентов и удовлетворенности их потребностей.
* Обеспечение высокого уровня безопасности и защиты данных клиентов и банка.
* Улучшение взаимодействия между сотрудниками банка и повышение оперативности принятия решений.
* Создание основы для будущего развития и расширения системы с учетом изменяющихся требований и технологических возможностей.

**Вывод:**

Техническое задание на разработку информационной системы по банковским операциям определяет цель и задачи проекта, основные требования к системе и технические требования для ее разработки. Целью проекта является автоматизация банковских операций с помощью информационной системы, которая должна обладать высокой производительностью, безопасностью и гибкостью. Основные требования включают полную автоматизацию банковских операций, интеграцию с платежными системами, удобный пользовательский интерфейс, высокий уровень безопасности данных и гибкость системы. Ожидаемыми результатами являются функциональная и надежная ИС, улучшение операций и обслуживания клиентов, повышение безопасности данных и возможность дальнейшего развития и расширения системы.

# ГЛАВА 2. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 2.1 Анализ требований к базе данных

|  |  |  |
| --- | --- | --- |
| № | Вопрос | Ответ |
| 1 | Какие данные будут храниться в базе данных? | Данные клиентов, данные счетов, данные платежей, данные безопасности |
| 2 | Какой объем данных предполагается хранить в базе данных? | Объем данных зависит от количества пользователей и объема информации о каждом счёте, но может ожидаться большой объем данных. |
| 3 | Как будет организована структура базы данных? | База данных будет организована в виде реляционной модели, состоящей из нескольких таблиц, связанных между собой. |
| 4 | Как будет обеспечена безопасность хранения данных в базе? | Безопасность хранения данных будет обеспечиваться с помощью защиты базы данных паролем и ограничения доступа к информации только уполномоченным лицам. |
| 5 | Какие механизмы резервного копирования будут использоваться? | Будут использоваться механизмы периодического резервного копирования данных и механизмы восстановления данных в случае их потери. |

## Таб. 1 Анализ требований к базе данных

## Анализ требований БД

1. БД сайта по банковским операциям проектируется для решения следующих задач:

* Хранение и управление данными клиентов, счетами, кредитами и платежами.
* Обработка и отслеживание транзакций, переводов и операций.
* Оперативное получение информации о финансовом положении клиентов и состоянии их счетов.
* Поддержка аналитической отчетности для финансового анализа и мониторинга.
* Обеспечение безопасности данных и контроля доступа.

1. Для решения этих задач необходимы следующие информационные объекты:

* Клиенты: информация о клиентах банка, их личные данные, финансовое положение и история операций.
* Счета: данные о счетах клиентов, их типы, балансы, транзакции и история операций.
* Кредиты: информация о выданных кредитах, условиях, графиках погашения и истории платежей.
* Платежи: данные о переводах и платежах между счетами и внешними платежными системами.
* Валюта и курсы обмена: информация о валюте, ее курсах и изменениях для валютных операций.
* Журналы аудита и безопасности: записи о действиях пользователей, журналы доступа и аудита для обеспечения безопасности данных.

1. Текущие запросы, которые должны удовлетворять данные информационной системы:

* Запросы клиентов на просмотр баланса счетов, истории операций и финансового состояния.
* Запросы на выполнение платежей, переводов и других банковских операций.
* Запросы на получение отчетности и аналитической информации для финансового анализа и мониторинга.
* Запросы на управление данными клиентов, счетами и кредитами (создание, изменение, закрытие и т.д.).
* Запросы на обработку и мониторинг транзакций, выявление потенциальных мошеннических операций.

## 2.3 Запросы к информационной системе

1. Запрос на выборку всех клиентов:

SELECT \* FROM Clients;

1. Запрос на выборку всех счетов клиента по его ID:

SELECT \* FROM Accounts WHERE client\_id = 1;

1. Запрос на выборку всех транзакций по определенной дате:

SELECT \* FROM Transactions WHERE transaction\_date = '2023-05-02 15:30:00';

1. Запрос на подсчет общего баланса на счету клиента:

SELECT SUM(balance) AS total\_balance FROM Accounts WHERE client\_id = 1;;

1. Запрос на выборку всех клиентов, у которых баланс на счету превышает определенную сумму:

SELECT \* FROM Clients WHERE client\_id IN (

SELECT client\_id FROM Accounts WHERE balance > 9999

);

1. Запрос на выборку суммы всех депозитов клиента:

SELECT Clients.client\_id, Clients.first\_name, Clients.last\_name, SUM(Accounts.balance) AS total\_deposit

FROM Clients

JOIN Accounts ON Clients.client\_id = Accounts.client\_id

WHERE Accounts.account\_type = 'Текущий счет'

GROUP BY Clients.client\_id, Clients.first\_name, Clients.last\_name;

**Вывод:**

Анализ требований к базе данных для информационной системы по банковским операциям определил основные задачи, информационные объекты и запросы, которые должны быть реализованы в системе. База данных должна хранить данные о клиентах, счетах, кредитах, платежах, валюте и курсах обмена, а также вести журналы аудита и безопасности. Структура базы данных будет реализована в виде реляционной модели с несколькими связанными таблицами.

Для обеспечения безопасности данных, доступ к базе данных будет ограничен и защищен паролем. Также будут применяться механизмы периодического резервного копирования и восстановления данных в случае их потери.

Разработанные запросы к информационной системе позволят выполнять различные операции, такие как просмотр баланса счетов, истории операций, выполнение платежей, получение отчетности и аналитической информации, а также управление данными клиентов, счетами и кредитами. Запросы также будут использоваться для обработки и мониторинга транзакций и выявления потенциальных мошеннических операций.

Таким образом, анализ требований к базе данных позволяет определить основные задачи, структуру и функциональность информационной системы, а также разработать соответствующие запросы для работы с данными.

# ГЛАВА 3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 3.1. Разработка кода информационной системы

Разработка кода информационной системы для банковских операций включает в себя создание программного обеспечения, которое будет реализовывать функциональности, требуемые для работы с данными и обеспечения банковских операций. Вот основные шаги и аспекты, которые следует учесть при разработке кода информационной системы:

Определение языка программирования: Выбор языка программирования будет зависеть от требований проекта, экспертизы разработчиков и рекомендаций. Различные языки программирования, такие как Java, C#, Python или JavaScript, могут использоваться для разработки банковской информационной системы.

Архитектура информационной системы: Определение архитектуры системы, включая разделение на модули или слои, управление базой данных, обработку запросов и взаимодействие с пользовательским интерфейсом.

Разработка базы данных: Создание структуры базы данных, определение таблиц, связей и полей для хранения данных, связанных с банковскими операциями. Использование соответствующих баз данных, таких как MySQL, PostgreSQL или Oracle, и создание схемы БД.

Разработка бизнес-логики: Реализация функций и алгоритмов, связанных с банковскими операциями, включая создание счетов, выполнение переводов, обработку кредитных операций и генерацию отчетов. Это включает обработку данных, проведение расчетов, проверку безопасности и контроль доступа.

Разработка пользовательского интерфейса: Создание интерфейса, через который пользователи смогут взаимодействовать с банковской информационной системой. Это может включать разработку веб-страниц, мобильных приложений или клиентских приложений для доступа к функциональностям системы.

Тестирование и отладка: Проведение тестового окружения и тестирования кода для проверки его работоспособности, выявления ошибок и исправления дефектов. Тестирование может включать модульное тестирование, интеграционное тестирование и системное тестирование.

Документирование: Создание документации, описывающей функциональности, архитектуру и использование информационной системы. Это может включать описание API, инструкции по установке и использованию системы, а также описание бизнес-процессов.

Развертывание и поддержка: Установка и настройка информационной системы на сервере, обеспечение ее работоспособности и поддержка в процессе эксплуатации. Это включает мониторинг производительности, обновление и патчинг системы, а также обеспечение безопасности данных.

**Вывод:**

Разработка кода информационной системы для банковских операций требует выполнения нескольких ключевых шагов и учета различных аспектов. В первую очередь, необходимо определить язык программирования, который будет использоваться для разработки системы, в зависимости от требований проекта, навыков разработчиков и рекомендаций.

Далее, необходимо определить архитектуру информационной системы, включая разделение на модули или слои, управление базой данных, обработку запросов и взаимодействие с пользовательским интерфейсом. Создание структуры базы данных, определение таблиц, связей и полей для хранения данных, также является важным шагом в разработке системы.

Реализация бизнес-логики является ключевым аспектом разработки, так как включает в себя функции и алгоритмы, связанные с банковскими операциями, такими как создание счетов, выполнение переводов, обработка кредитных операций и генерация отчетов. Разработка пользовательского интерфейса также является важной частью процесса разработки, чтобы обеспечить удобное взаимодействие пользователей с системой.

После создания кода информационной системы необходимо провести тестирование и отладку, чтобы проверить работоспособность системы, выявить и исправить ошибки и дефекты. Документирование функциональностей, архитектуры и использования системы также является важной частью разработки, чтобы обеспечить понимание и использование системы другими разработчиками и пользователями.

После завершения разработки кода, необходимо развернуть систему на сервере, настроить ее и обеспечить ее работоспособность и поддержку в процессе эксплуатации. Это включает мониторинг производительности, обновление и патчинг системы, а также обеспечение безопасности данных.

Таким образом, разработка кода информационной системы для банковских операций требует выполнения нескольких этапов и учета различных аспектов, начиная от выбора языка программирования и определения архитектуры до тестирования, документирования и поддержки системы.

# ГЛАВА 4. ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.

## 4.1. План тестирования

*Что такое тестирование?*

Тестирование информационной системы является одним из самых важных этапов ее разработки. Оно позволяет проверить работоспособность и качество системы перед ее внедрением в боевые условия. Тестирование – это процесс исследования системы на предмет выявления ошибок, недочетов и проблем, которые могут возникнуть при ее эксплуатации.

*Из каких этапов состоит тестирование?*

Тестирование состоит из следующих этапов:

- Планирование тестирования;

- Разработка тест-кейсов и тест-планов;

- Выполнение тестов;

- Анализ результатов тестирования;

- Отчетность о тестировании.

*Какие документы используются при тестировании?*

При тестировании используются следующие документы:

- Тест-планы;

- Тест-кейсы;

- Отчеты о тестировании.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| № | Описание теста | Ожидаемый результат | Статус (Пройден/Не пройден) |
| 1 | Регистрация нового пользователя | Пользователь успешно зарегистрирован |  |
| 2 | Вход пользователя в систему | Пользователь успешно авторизован |  |
| 3 | Оформление счёта | Счёт успешно оформлена |  |
| 4 | Пополнение на счёт | Счёт успешно пополнен |  |

## Таб. 2 Описание теста

*План тестирования*

План тестирования должен содержать следующие данные:

- Цель тестирования;

- Описание тестируемой системы;

- Тест-кейсы;

- Критерии завершения тестирования;

- Сроки проведения тестирования;

- Ответственные за проведение тестирования.

Пример плана тестирования для информационной системы по банковским операциям представлен в таблице ниже:

Таким образом, план тестирования является важным этапом разработки информационной системы. Он помогает выявить ошибки и недочеты системы, что позволяет улучшить ее качество и надежность. План тестирования должен содержать описание

## 4.2. Написание тест-кейсов

Тест-кейс - это документ, который описывает шаги, которые необходимо выполнить, чтобы протестировать определенную функциональность.

Цель тест-кейса - убедиться, что функциональность работает правильно.

Этапы создания тест-кейсов:

1. Определение цели тестирования
2. Выбор функциональности, которую необходимо протестировать
3. Разработка шагов тестирования
4. Выполнение тестов
5. Фиксация результатов тестирования
6. Анализ результатов тестирования

Примеры тест-кейсов для сайта банковских операций (в виде таблицы):

|  |  |  |
| --- | --- | --- |
| Название тест-кейса | Цель тестирования | Шаги выполнения |
| Регистрация нового пользователя | Проверка, что пользователь может зарегистрироваться на сайте | 1. Открыть страницу регистрации.  2. Ввести данные для регистрации (имя, фамилия, адрес электронной почты, пароль).  3. Нажать кнопку "Зарегистрироваться".  4. Проверить, что пользователь успешно зарегистрирован. |
| Открытие дополнительного счёта | Проверка правильности счёта | 1. Зайти на страницу дополнительные продукты  2. Выбрать дополнительный счёт  3. Нажать кнопку "Добавить".  4. Проверить, счёт добавился. |
| Пополнение счёта | Проверка, что пользователь пополнить свой баланс счёта | 1. Открыть страницу мои финансы.  2. Ввести данные об счёте (номер, на сколько пополнить).  3. Нажать кнопку "Пополнить".  4. Проверить, что счёт успешно пополнен. |
| Переводы между счётами | Проверка, что пользователь может переводить между счётов | 1. Открыть страницу переводы.  2. Выбрать с какого счёта и на какой перевести.  3. Ввести сколько перевести.  4. Нажать кнопку "Перевести".  5. Проверить, что перевод успешно перевелся. |

## Таб. 3 Тест к сайту

Таблица тест-кейсов помогает системе и тестировщикам более эффективно проводить функциональное тестирование системы. Она обеспечивает описание целей и процесса каждого тест-кейса, что помогает тестировщикам проводить более точное и эффективное тестирование.

Таблица тест-кейсов позволяет оценивать результаты тестирования, а также вносить изменения в них, если таковые будут нужны. Благодаря таблице тест-кейсов, тестировщики могут более эффективно тестировать функциональность системы, что упрощает процесс ее разработки и повышает качество и надежность всей системы.

## 4.3 Юнит-тестирование

Юнит-тестирование - это процесс тестирования отдельных модулей программного обеспечения с целью подтверждения их корректности и правильности работы.

Цель юнит-тестирования - убедиться, что отдельные модули программного обеспечения работают верно и не влияют на работу других модулей.

Этапы создания юнит-тестов:

1. Выбор модуля для тестирования
2. Разработка тестовых случаев
3. Написание тестов
4. Выполнение тестов
5. Фиксация результатов тестирования
6. Анализ результатов тестирования

Для информационной системы по банковским операциям можно составить юнит-тесты для следующих основных модулей (в виде таблицы):

|  |  |  |
| --- | --- | --- |
| Название модуля | Цель тестирования | Тестовые случаи |
| Модуль авторизации пользователя | Проверка правильности работы процесса авторизации | 1. Проверка, что пользователь может войти в систему с правильными учетными данными.  2. Проверка, что при попытке входа с неправильными учетными данными, система не позволяет пользователю авторизоваться. |
| Модуль расчета процента | Проверка правильности расчета процента | 1. Проверка, что система правильно рассчитывает процент. |
| Модуль оформления заявки на открытие счёта | Проверка правильности работы процесса оформления заявки на дополнительный счёт | 1. Проверка, что пользователь может успешно оформить заявку на дополнительный счёт.  2. Проверка, что система не позволяет пользователю оформить заявку без заполнения всех обязательных полей. |
| Модуль переводов | Проверка правильности работы процесса перевода | 1. Проверка, что пользователь может успешно перевести онлайн.  2. Проверка, что система не позволяет пользователю перевести без правильного ввода данных карты. |
| Модуль управления профилем пользователя | Проверка правильности работы процесса управления профилем пользователя | 1. Проверка, что пользователь может успешно изменить свои данные профиля (например, имя, фамилия, пароль).  2. Проверка, что система не позволяет пользователю изменить данные профиля без правильного ввода старого пароля |

## Таб. 4 Юнит тест

Таблица юнит-тестов помогает системе и разработчикам программного обеспечения провести тестирование отдельных модулей программы и обеспечить их корректность и надежность.

Тестирование модулей позволяет обнаружить ошибки и недостатки во время разработки, что помогает избежать проблем в будущем. Таблица юнит-тестов позволяет составить план тестирования каждого модуля системы, что упрощает процесс тестирования и оценку результатов. Благодаря таблице юнит-тестов, разработчики могут удостовериться в том, что тестируемые модули работают должным образом, что повышает качество и надежность всей системы.

**Вывод:**

Тестирование информационной системы является важным этапом ее разработки, позволяющим выявить ошибки, недочеты и проблемы до внедрения системы в боевые условия. Оно состоит из нескольких этапов, включая планирование тестирования, разработку тест-кейсов, выполнение тестов, анализ результатов и отчетность о тестировании.

При тестировании используются документы, такие как тест-планы, тест-кейсы и отчеты о тестировании. Тест-план содержит данные о целях тестирования, описание тестируемой системы, тест-кейсы, критерии завершения тестирования, сроки проведения и ответственных за проведение тестирования.

Тест-кейс представляет собой документ, описывающий шаги, необходимые для протестирования определенной функциональности. Он помогает убедиться, что функциональность работает правильно. Создание тест-кейсов включает определение цели тестирования, выбор функциональности для тестирования, разработку шагов тестирования, выполнение тестов, фиксацию результатов и анализ результатов тестирования.

Юнит-тестирование представляет собой процесс тестирования отдельных модулей программного обеспечения с целью подтверждения их корректности и правильности работы. Оно позволяет проверить работу модулей независимо друг от друга. Создание юнит-тестов включает выбор модуля для тестирования и разработку тестовых случаев.

Таблицы тест-кейсов и юнит-тестов помогают системе и тестировщикам более эффективно проводить функциональное тестирование и тестирование модулей соответственно. Они обеспечивают описание целей, процесса и ожидаемых результатов каждого теста, что способствует более точному и эффективному тестированию.

Тестирование и написание тест-кейсов и юнит-тестов помогают обнаружить ошибки и недостатки в информационной системе на ранних этапах разработки, что позволяет улучшить ее качество, надежность и безопасность.