МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт информационных технологий |
| наименование института (факультета) |
| Кафедра математического и программного обеспечения ЭВМ |
| наименование кафедры |

ОТЧЕТ

по Производственной практике (Научно-исследовательской работе)

Листов 29

Студента Маркелова С. А.,

группы 1ПИб-02-3оп-22

Место прохождения практики

Кафедра математического и программного обеспечения ЭВМ

Студент \_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

|  |  |
| --- | --- |
| Руководитель практики от кафедры | МПО ЭВМ |
| Сальникова О.С., к.филол.н., доцент | |
| Руководитель практики от предприятия, организации, учреждения  (Ф.И.О., должность) | |
| Сальникова О.С., к.филол.н., доцент | |

2024 г.

Оглавление

[Введение 3](#_Toc183556076)

[Основная часть 5](#_Toc183556077)

[1. Описание предметной области 5](#_Toc183556078)

[2. Планирование работ по проекту 8](#_Toc183556079)

[3. Выбор технологии и инструментальных средств реализации проекта 12](#_Toc183556080)

[4. Анализ существующих технических решений 16](#_Toc183556081)

[5. Подбор математического аппарата 21](#_Toc183556082)

[Заключение 23](#_Toc183556083)

[Список литературы 24](#_Toc183556084)

[Приложение 1. Диаграмма Ганта 29](#_Toc183556085)

# Введение

Платформа «Финансовые Информационные Системы» (ФИС-платформа) играет ключевую роль в обеспечении эффективного управления финансовыми ресурсами «Совкомбанка» [27]. В условиях постоянного роста объемов данных и усложнения процессов обработки финансовой информации возникает необходимость в создании новых инструментов, обеспечивающих оперативный доступ к аналитическим данным и их визуализацию [31].

Виджет ФИС – самодостаточный элемент интерфейса, который можно использовать любое количество раз. Он имеет конечный набор свойств и событий (функций/методов). Принимает на вход данные, описывающие то, что мы хотим увидеть на экране. Виджеты, как элементы пользовательского интерфейса, позволяют значительно упростить взаимодействие с системой, повысить удобство пользователей и улучшить восприятие данных [5].

На сегодняшний день отсутствие универсальных решений для создания интерфейсов приводит к тому, что для каждого приложения требуется разрабатывать уникальные интерфейсы «с нуля» [18]. Это увеличивает трудоемкость, временные и финансовые затраты на реализацию приложений ФИС-платформы [45].

Разработанные виджеты могут быть использованы в качестве стандартных компонентов для построения интерфейсов приложений ФИС-платформы. Это позволит значительно сократить время разработки новых приложений, упростить их поддержку и повысить качество взаимодействия пользователей с системой [23].

Исследование и разработка новых виджетов для ФИС-платформы актуальны, так как они способствуют улучшению функциональности и адаптации системы под изменяющиеся потребности бизнеса. (примеры использования еще)

Цель работы: автоматизация и ускорение процесса разработки бизнес-приложений на платформе ФИС путем расширения списка доступных визуальных компонентов – виджетов, позволяющих манипулировать данными из конкретного типа (таблицы баз данных).

Задачи:

1. Изучить предметную область;

2. Узнать, существуют ли готовые решения для обозначенной проблемы;

3. Выяснить, какие алгоритмы используются в подобных моделях в других отраслях;

4. Разработать математическую модель, подходящую для решения поставленной проблемы;

5. Составить план-график работ;

6. Выбрать технологии и инструментальные средства для решения поставленной задачи.

Объект исследования – платформа «Финансовые Информационные Системы».

Предмет исследования – виджеты ФИС-платформы.

# Основная часть

## Описание предметной области

ФИС-платформа – это программная платформа, ориентированная на автоматизацию и оптимизацию процессов в финансовом секторе. Ее возможности позволяют значительно ускорить разработку решений, снизить затраты на создание и поддержку систем, а также адаптировать бизнес-процессы под изменения в экономической среде. Платформа особенно полезна для крупных организаций, которым важно быстро реагировать на новые вызовы рынка [39].

ФИС-платформа основана на использовании low-code подхода [2]. Low-code (в переводе с англ. «мало кода») – это подход к разработке программного обеспечения, в котором большая часть работы выполняется с помощью визуальных инструментов. Разработчики используют визуальные инструменты, такие как конструкторы интерфейсов и процессов, вместо того чтобы вручную писать программный код [13]. Этот метод ускоряет создание программных продуктов и снижает требования к квалификации разработчиков.

Основные характеристики ФИС-платформы:

1. Инструменты для визуальной разработки – платформа предлагает инструменты, которые позволяют проектировать интерфейсы и бизнес-процессы без необходимости программирования. Например:
   * Конструктор форм – визуальный инструмент для создания пользовательских интерфейсов;
   * Дизайнер процессов – инструмент, который позволяет «нарисовать» алгоритм работы системы (например, последовательность обработки заявки);
2. Готовые шаблоны решений – платформа включает набор типовых шаблонов, которые можно адаптировать для конкретных задач. Это ускоряет внедрение системы и снижает затраты на разработку с нуля;
3. Интеграция с другими системами – платформа поддерживает обмен данными с другими информационными системами через стандартные протоколы, такие как REST API (интерфейсы для взаимодействия с другими приложениями) и SOAP (протокол для передачи данных в формате XML);
4. Масштабируемость и высокая производительность – платформа подходит для работы с большими объемами данных и высоким количеством пользователей, что важно для банков и крупных финансовых организаций [30].

ФИС-платформа обладает следующими преимуществами использования:

1. Скорость разработки – благодаря low-code подходу решения создаются в 2-3 раза быстрее, чем при традиционном программировании.
2. Экономия ресурсов – визуальные инструменты снижают потребность в квалифицированных разработчиках, так как бизнес-аналитики и сотрудники без глубоких знаний программирования могут участвовать в создании приложений.
3. Гибкость и удобство настройки – систему легко адаптировать под изменения в бизнес-процессах или требованиях.
4. Снижение количества ошибок – готовые шаблоны и визуальная разработка уменьшают риск ошибок при создании приложений.

«Совкомбанк» активно применяет платформу ФИС для автоматизации финансовых процессов и улучшения операционной эффективности. Основные кейсы использования включают следующие направления:

1. Автоматизация кредитного конвейера – благодаря ФИС-платформе, банк значительно сократил время на принятие решений по кредитам [32]. Например, процесс проверки и одобрения заявок ускорился до нескольких минут, что позволяет банку быть конкурентоспособным за счет быстрого обслуживания клиентов;
2. Оптимизация обработки данных – платформа ФИС помогает интегрировать более 50 внешних систем, что позволяет автоматизировать анализ и сбор информации о клиентах [40]. Это повышает прозрачность процессов и снижает операционные риски;
3. Ускорение цифровых операций – благодаря ФИС-платформе, банк смог перевести до 88% заявок в онлайн-формат, что сократило нагрузку на отделения и увеличило долю онлайн-обслуживания клиентов [35]. Это пример успешного использования low-code подхода для упрощения сложных процессов и настройки бизнес-логики;
4. Система управления операционными рисками – «Совкомбанк» использует решения ФИС-платформы для повышения эффективности управления операционными рисками [10]. Внедрение системы позволило банку систематизировать сбор и анализ данных о рисках, интегрировать их с внутренними процессами и сформировать более гибкую регуляторную отчетность.

## Планирование работ по проекту

В таблице 1 представлен план-график реализации проекта. Другой вариант плана, представленный в виде диаграммы Ганта [38], представлен в приложении 1.

Таблица 1

План реализации проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Этапы проекта | Срок выполнения | Участники проекта  (О – ответственный / У – участники) | Результат |
| 1 | 2 | 3 | 4 | 5 |
| Этап 1. Подготовительный этап | | | | |
| 1.1 | Ознакомление с целью и задачами проекта | 01.04.2024 – 08.04.2024 | О – Коржавин Владислав Сергеевич  У – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Изученная предметная область |
| 1.2 | Изучение аналогов проекта, примеров технических заданий на разработку | 09.04.2024 – 13.05.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Участники проекта ознакомлены с аналогами и примерами технических заданий |
| 1.3 | Подготовка первоначального перечня функций виджета | 14.05.2024 – 20.05.2024 | У – Коржавин Владислав Сергеевич  У – Трофимов Никита Владиславович  О – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Был подготовлен один перечень с функциями |

Продолжение табл. 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1.4 | Обсуждение этапов проекта, бюджета проекта, требуемых внешних ресурсов с заказчиком | 21.05.2024 – 29.05.2024 | У – Коржавин Владислав Сергеевич  У – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  О – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Составлены основные этапы проекта |
| 1.5 | Формирование бюджета проекта: смета расходов | 30.05.2024 – 27.06.2024 | У – Коржавин Владислав Сергеевич  У – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  О – Маркелов Сергей Александрович | Сформирована смета |
| Этап 2. Основной этап | | | | |
| 2.1 | Создание поля ввода для поиска объекта | 28.06.2024 – 30.07.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Разработано поле ввода для поиска объектов |

Продолжение табл. 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 2.2 | Создание кнопки расширенного поиска / добавление нового объекта | 31.07.2024 – 27.08.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Разработаны кнопки расширенного поиска / добавление нового объекта |
| 2.3 | Создание истории поиска | 28.08.2024 – 27.09.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Реализовано создание истории поиска |
| 2.4 | Создание всплывающего окна с таблицей объектов | 27.09.2024 –25.10.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Разработано всплывающее окно с таблицей объектов |
| 2.5 | Обсуждение с заказчиком прогресса по созданию виджета | 26.10.2024 –01.11.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Подведены итоги основного этапа разработки виджета |

Продолжение табл. 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Этап 3. Тестирование | | | | |
| 3.1 | Первичный запуск виджета в тестовой среде | 02.11.2024 –09.11.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Проверена работа виджета в тестовой среде |
| 3.2 | Тестирование готового виджета | 10.11.2024 –19.12.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Проверен функционал виджета на наличие ошибок |
| 3.2 | Тестирование готового виджета | 10.11.2024 –19.12.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Проверен функционал виджета на наличие ошибок |
| Этап 4. Финальный этап | | | | |
| 4.1 | Презентация проекта | 20.12.2024 | У – Коржавин Владислав Сергеевич  О – Трофимов Никита Владиславович  У – Беляков Артемий Александрович  У – Подтелков Владислав Владимирович  У – Фатькина Алена Дмитриевна  У – Маркелов Сергей Александрович | Представлен готовый виджет |

## Выбор технологии и инструментальных средств реализации проекта

Проект реализуется на ФИС-платформе. Она предоставляет удобные low-code инструменты для разработки интерфейсов и управления бизнес-процессами. Это позволяет сократить время разработки и упростить внедрение новых решений. Платформа ориентирована на задачи финансового сектора и предоставляет оптимальный набор инструментов для реализации проекта [11].

Для разработки функционала виджетов, как элементов интерфейса, и их интеграции с системами управления требуется выбрать подходящий язык программирования [19]. Язык должен поддерживать разработку клиентской и серверной части, иметь возможность интеграции с API [36] и работы с базами данных, а также соответствовать требованиям производительности и масштабируемости.

Было рассмотрено несколько вариантов языков программирования:

1. JavaScript – универсальный язык программирования, ориентированный на разработку web-интерфейсов [33]. К его преимуществам относится его поддержка всеми браузерами, наличие большого количества библиотек и фреймворков, а также возможность создания динамичных интерфейсов. Однако он не оптимален для высокопроизводительных серверных задач.
2. Python – объектно-ориентированный, интерпретируемый, переносимый язык программирования высокого уровня, популярный в задачах анализа данных, автоматизации и написания серверной логики [12]. Он обладает такими преимуществами, как легкость в освоении, гибкость, поддержка большинства платформ, наличие большой библиотеки готовых решений для API, аналитики и машинного обучения. Недостатком является низкая производительность по сравнению с компилируемыми языками.
3. C++ – язык программирования высокого уровня, ориентированный на высокую производительность и контроль над ресурсами [29]. Его преимуществами являются максимальная производительность и наличие возможностей для работы с системными ресурсами и оптимизации. Недостатком является сложность в поддержке и масштабировании.
4. C# – современный объектно-ориентированный язык, разработанный компанией Microsoft, который используется для создания настольных, мобильных и web-приложений [1]. Подходит для разработки на платформе .NET. Он обладает рядом преимуществ, таких как интеграция с экосистемой Microsoft, высокая производительность для серверной разработки и поддержка инструментов .NET Framework [7]. Недостатком является то, что для использования некоторых технологий (например, Microsoft SQL Server) потребуется покупка платной лицензии.

Для разработки функционала виджетов был выбран JavaScript. Причиной такого выбора является его универсальность и богатый набор библиотек.

Для работы над проектом необходима интегрированная среда разработки (IDE), которая обеспечит удобство написания, отладки и тестирования кода, поддержку выбранных языков программирования и наличие инструментов для работы с системой контроля версий, интеграций и расширений.

Рассматривались следующие варианты IDE [14]:

1. Visual Studio – полноценная IDE от Microsoft, предназначенная для разработки на C#, C++, Python, JavaScript и других языках [42]. Она подходит для больших и сложных проектов. Ее преимуществами являются мощные встроенные инструменты отладки и анализа кода, интеграция с .NET Framework, поддержка сложных проектов, включая многослойные архитектуры и расширенная поддержка работы с базами данных через SQL Server Data Tools. Однако она обладает и рядом недостатков, таких как высокие системные требования и недоступность части функционала без покупки платной лицензии.
2. Visual Studio Code – бесплатная и легковесная среда разработки с поддержкой множества расширений [50]. К ее преимуществам можно отнести удобство работы с JavaScript, Python, HTML и CSS, легкость освоения, поддержку множества плагинов и интеграцию с Git. Однако она не всегда подходит для крупных проектов, требующих сложной структуры, что является ее недостатком.
3. IntelliJ IDEA – IDE для языка программирования Java с возможностью настройки под другие языки программирования через плагины [46]. К ее преимуществам относятся удобство работы с большими проектами и широкий набор встроенных инструментов. Но недостатками являются высокие требования к системе и ориентированность в первую очередь на язык Java, что может ограничить функционал для других языков.
4. Eclipse – IDE с открытым исходным кодом, поддерживающая Java, C++, Python и другие языки [44]. Ее преимуществами являются бесплатность, расширяемость и наличие множества доступных плагинов. К недостаткам относятся устаревший интерфейс и необходимость тратить значительное количество времени на настройку и адаптацию.

Для реализации проекта была выбрана IDE Visual Studio Code за её легкость, удобство работы с JavaScript и доступность множества плагинов.

Для хранения данных проекта требуется реляционная база данных с поддержкой транзакций, сложных запросов и масштабируемости. Для работы с базами данных необходимо выбрать систему управления базами данных (СУБД). К основным критериям при выборе СУБД относятся надежность, производительность, простота в управлении и развертывании, а также возможность работы с большими объемами данных.

Были рассмотрены следующие варианты СУБД [24]:

1. PostgreSQL – высокопроизводительная реляционная СУБД с открытым исходным кодом [25]. Ее преимуществами является надежность, поддержка сложных транзакций, бесплатность и гибкость в использовании. Однако она относительно сложна в настройке, что можно отнести к ее недостаткам.
2. MySQL – популярная СУБД, подходящая для небольших и средних проектов [49]. К ее преимуществам можно отнести простоту освоения и высокую скорость работы. Но ее недостаткам являются ограничения в поддержке сложных запросов.
3. Oracle Database – коммерческая СУБД, подходящая для крупных организаций [16]. Ее преимуществами являются высокая производительность и наличие мощных аналитических инструментов. Однако программа доступна только по покупке лицензии, стоимость которой достаточно высокая.
4. Microsoft Access – инструмент для работы с базами данных, ориентированный на небольшие проекты [6]. Входит в состав Microsoft Office. Он подходит для быстрой разработки небольших приложений и прототипов. Его преимуществами являются простота в освоении и использовании и интеграция с другими продуктами Microsoft. Однако он обладает рядом недостатков. Например, он не подходит для крупных проектов и работы с большими объемами данных, а также обладает ограниченной производительностью в многопользовательском режиме.

В качестве СУБД для проекта была выбрана PostgreSQL, так как она предоставляет баланс между мощностью, бесплатностью и гибкостью, а также более производительна и надежна в условиях больших объемов данных и многопользовательского доступа.

## Анализ существующих технических решений

Для реализации виджетов на платформе ФИС необходимо рассмотреть существующие технические решения, применяемые для создания интерфейсных компонентов в рамках иных платформ.

Сравнение различных решений будем проводить по следующим критериям:

* Назначение – для каких задач создан инструмент;
* Простота освоения – уровень сложности при изучении и начале использования;
* Гибкость – возможность адаптации под сложные и уникальные задачи;
* Производительность – скорость работы и способность обрабатывать большие объемы данных;
* Стоимость – финансовые затраты на покупку лицензии
* Поддержка кастомизации – возможность модификации решения под конкретные задачи.

В таблице 2 представлено сравнение ФИС-платформы и других существующих решений по выделенным критериям.

Таблица 2

Анализ существующих решений

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Решение | Назначение | Простота освоения | Гибкость | Производи- тельность | Стоимость | Поддержка кастомизации |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| ФИС-платформа | Low-code платформа для автомати-зации финансовых процессов | Высокая | Средняя | Высокая | Платная | Средняя |

Продолжение табл. 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| OutSystems | Low-code платформа для быстрого прототипирования | Средняя | Средняя | Высокая | Очень высокая | Средняя |
| Mendix | Low-code платформа с поддержкой визуального проектирования | Высокая | Низкая | Средняя | Высокая | Низкая |
| React | JavaScript-библиотека для создания UI | Низкая | Высокая | Высокая | Бесплатно | Высокая |
| Angular | Полноценный фреймворк для разработки сложных приложений | Низкая | Средняя | Высокая | Бесплатно | Средняя |
| Vue.js | Легковесный фреймворк для разработки интерфейсов | Средняя | Высокая | Средняя | Бесплатно | Высокая |
| Bootstrap | Набор компонентов для адаптивного дизайна | Высокая | Низкая | Средняя | Бесплатно | Низкая |
| Material-UI | Реализация Material Design для React | Средняя | Средняя | Высокая | Бесплатно | Средняя |
| Ant Design | UI-библиотека для корпоративных интерфейсов | Средняя | Средняя | Высокая | Бесплатно | Высокая |

Low-code платформы обеспечивают быструю разработку приложений с минимальным объемом ручного программирования. Рассмотрим некоторые популярные решения:

* OutSystems – гибкая платформа для создания пользовательских интерфейсов [28]. Поддерживает интеграцию с различными API. Обладает высокой скоростью прототипирования. Подходит для масштабируемых приложений. Но при этом имеет достаточно высокую стоимость лицензии. Также существует ряд сложностей при создании высококастомизированных интерфейсов;
* Mendix – инструмент для разработки low-code приложений, в том числе и виджетов [48]. Поддерживает визуальное проектирование. Удобен для работы с базами данных. Однако, имеет ограниченный контроль над кодом.

Web-фреймворки позволяют создавать интерфейсы с нуля, предоставляя набор инструментов для управления компонентами, маршрутизацией и состоянием приложения. Среди них можно выделить несколько:

* React – JavaScript-библиотека для создания пользовательских интерфейсов [21]. Обладает высокой гибкостью. Имеет большое количество плагинов и инструментов. Но для полноценной разработки потребуется установка дополнительных библиотек (например, Redux, React Router);
* Angular – полноценный фреймворк от Google для создания интерфейсов [41]. Поддерживает TypeScript. Имеет интегрированные инструменты для работы с REST API. При этом обладает высокой сложностью освоения. Также слишком тяжеловесен для разработки небольших приложений.
* Vue.js – легковесный фреймворк для разработки интерфейсов [34]. Прост в использовании. Легко интегрируется в существующие проекты. Однако, имеет ограничения в поддержке корпоративных проектов.

Готовые UI-библиотеки предоставляют компоненты (кнопки, формы, таблицы) для быстрого создания интерфейсов [3]:

* Bootstrap – набор компонентов для разработки адаптивных интерфейсов [22]. Прост в интеграции. Поддерживает мобильные интерфейсы. При этом обладает устаревшим дизайном и ограниченными возможностями кастомизации.
* Material-UI – реализация принципов Material Design от Google [47]. Интегрирован с React и зависим от него. Имеет более высокую сложность кастомизации.
* Ant Design – UI-библиотека для создания интерфейсов в корпоративной среде [43]. Подходит для разработки сложных финансовых приложений. Включает в себя множество различных компонентов (например, таблицы, графики и т. д.). Также интегрирован с React и требует его глубокого понимания.

ФИС-платформа превосходит аналоги, так как она изначально ориентирована на финансовую отрасль, интегрируется с REST API и XML, поддерживает low-code подход, что упрощает и ускоряет разработку. Она позволяет оптимизировать ресурсы за счет готовых решений для аналитики и визуализации, оставаясь гибкой и масштабируемой.

Задачей проекта является разработка виджетов для платформы ФИС, которые позволят стандартизировать и упростить создание интерфейсов для финансовых приложений. Это решение должно исключить необходимость проектирования интерфейсов «с нуля» для каждого нового приложения, что существенно ускорит процесс разработки и снизит затраты.

К виджетам предъявляются следующие требования:

* Модульность – виджеты должны легко добавляться или удаляться в зависимости от потребностей пользователя;
* Кастомизируемость – в виджетах должна быть реализована возможность изменять внешний вид, параметры и логику;
* Интерактивность – виджеты должны поддерживать связь с пользователем (например, через графики, фильтры и т. д.);
* Оптимизация – виджеты должны нести минимальную нагрузку на систему при максимальной скорости работы;
* Интегрируемость – полная совместимость с другими модулями и внешними сервисами.

К информационному обеспечению виджетов предъявляются следующие требования:

* Входная информация для виджетов должна поступать через стандартизированные форматы, такие как REST API и XML, что обеспечит совместимость с другими системами;
* В виджетах должна быть реализована возможность работы с основными типами финансовых данных: транзакциями, отчетами, графиками, таблицами и аналитикой;
* Виджеты должны поддерживать динамическое обновление данных в реальном времени;
* Должен быть реализован учет сложных взаимосвязей между финансовыми объектами (например, клиентами и их операциями).

К аппаратно-программному обеспечению предъявляются следующие требования:

* Платформа ФИС является основой разработки. Она обеспечивает интеграцию и поддержку low-code решений;
* Виджеты должны поддерживать кроссбраузерность и адаптивный дизайн;
* Для работы виджетов необходимо наличие серверов, обеспечивающих высокую производительность для обработки большого объема данных;
* Клиентские устройства должны соответствовать минимальным системным требованиям: наличие web-браузера с поддержкой современных технологий.

## Подбор математического аппарата

Разработка виджетов для платформы ФИС не требует использования сложных математических моделей, однако базовые алгоритмы и методы анализа данных играют важную роль в создании эффективного и функционального интерфейса.

Виджеты должны работать с финансовыми данными, предоставляя пользователям удобные способы их анализа и визуализации. Для этого используются следующие алгоритмы обработки и представления данных [20]:

* Алгоритмы фильтрации и сортировки. Например, быстрая сортировка или сортировка слиянием для упорядочивания транзакций, данных клиентов или отчетов по заданным критериям;
* Поиск данных. Для реализации поисковых функций виджетов применяются алгоритмы линейного поиска и бинарного поиска, особенно для больших массивов данных.

Для отображения аналитической информации и графиков требуются математические методы визуализации [15]:

* Метод линейной интерполяции – для построения графиков с плавными кривыми, особенно при работе с временными рядами;
* Метод агрегации данных, например, вычисление сумм, средних значений или медианы для отображения сводной информации по финансам.

Для обработки финансовых показателей виджеты могут использовать методы статистического анализа данных [37]:

* Расчет процентных изменений. Например, для отображения роста или падения показателей;
* Построение распределений данных для анализа клиентских операций или динамики денежных потоков.

Для повышения эффективности работы виджетов применяются методы оптимизации [8]:

* Оптимизация запросов к базам данных. Например, через индексирование и предварительное кэширование часто используемых данных.
* Адаптивное распределение элементов интерфейса – упрощенные эвристические алгоритмы для оптимального размещения виджетов на экране.

Некоторые виджеты, работающие с большими объемами данных, могут использовать базовые численные методы [9]:

* Аппроксимация данных – для построения трендов или прогнозных значений;
* Методы расчета ошибок – для проверки корректности отображаемой информации и устранения аномалий.

В виджетах также могут применяться методы дискретной математики [4]:

* Деревья поиска и хеш-таблицы – для быстрого доступа к данным и их структурирования;
* Графы – если потребуется визуализация связей между объектами (например, транзакции между клиентами или связи между отделами).

Для повышения удобства взаимодействия с виджетами также могут применяться алгоритмы автоматизации пользовательских взаимодействий [26]:

* Эвристические алгоритмы упрощения пользовательского ввода;
* Расчеты предустановленных параметров. Например, автозаполнение полей ввода или расчет предполагаемых значений.

Выбранный математический аппарат направлен на повышение эффективности работы с данными, улучшение визуализации и обеспечение высокой производительности системы. Хотя задачи не требуют сложных вычислений, применение этих методов позволит создавать виджеты, которые соответствуют функциональным требованиям платформы и облегчают взаимодействие пользователей с финансовыми данными.

# Заключение

В рамках научно-исследовательской работы были выполнены задачи, связанные с разработкой виджетов для платформы ФИС: изучена предметная область, разработан план работ по проекту, выбраны технологии и инструменты для реализации проекта, разработаны требования к проекту и подобран математический аппарат. Все это необходимо для успешной реализации проекта.

Разработка виджетов для ФИС-платформы позволила стандартизировать процесс создания интерфейсов для финансовых приложений. Это не только сократило время разработки, но и повысило удобство использования системы.

Внедрение виджетов обеспечило ускорение разработки интерфейсов, улучшение качества визуализации данных и упрощение взаимодействия пользователей с системой.

В результате прохождения практики был получен опыт работы с платформами автоматизации, анализа существующих решений, а также проектирования программных продуктов. Полученные знания и навыки будут полезны в дальнейшей профессиональной деятельности.

# Список литературы

1. Албахари, Дж. C# 7.0. Справочник. Полное описание языка / Дж. Албахари, Б. Албахари. : Пер. с англ. – СПб. : ООО «Альфа-книга», 2018. – 1024 с.
2. Андирякова О. О. Применение low-code технологии для решения бизнес-задач / О. О. Андирякова, А. А. Крюкова, М. И. Иваев – Текст : электронный // Индустриальная экономика. – 2023. – № 2.
3. Антонов С. А. Обзор современных библиотек для разработки интерфейса веб приложения / А. С. Антонов, А. А. Вуколов, К. А. Кононыхина – Текст : электронный // Вестник науки. – 2015. – № 8.
4. Белоусов, И. Н. Дискретная математика и математическая логика : учебник / И. Н. Белоусов, В. И. Белоусова – Екатеринбург: Изд-во Урал. ун-та, 2024. – 178 с.
5. Брусенцова, Т. П. Проектирование интерфейсов пользователя : пособие для студентов специальности 1-47 01 02 «Дизайн электронных и веб-изданий» / Т. П. Брусенцова, Т. В. Кишкурно. – Минск : БГТУ, 2019. – 172 с.
6. Волик, М. В. Разработка базы данных в Access : учеб. пособие – М.: Прометей, 2021. – 88 с.
7. Грачев М. К. Aspect. Net Framework и его применение в задаче протоколирования – Текст : электронный // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. – 2008. – № 4.
8. Гребенникова, И. В. Методы оптимизации: учеб. пособие – Екатеринбург : УрФУ, 2017. – 148 с.
9. Зенков, А. В. Численные методы: учеб. пособие – Екатеринбург : Изд-во Урал. ун-та, 2016. – 124 с.
10. Зубкова С. В. Система управления операционным риском в коммерческом банке с применением кибер-физических систем – Текст : электронный // Финансовые рынки и банки. – 2019. – № 4.
11. Коваленко Ю. М. Финансовый сектор в финансовой системе экономики: теоретический аспект – Текст : электронный // Вестник Киевского национального университета им. Тараса Шевченко. Серия: Экономика. – 2015. – № 6.
12. Кольцов, Д. М. Python. Полное руководство – СПб. : Издательство Наука и Техника, 2022. – 480 с.
13. Купер А. Интерфейс. Основы проектирования взаимодействия – 4-е изд. – СПб. : Питер, 2022. — 720 с.
14. Интегрированные среды разработки клиентской части программ в СУБД / Н. Е. Аскарбекова, К. Б. Есенгелди, А. С. Ешимов [и др.] – Текст : электронный // Academy. – 2020. – № 12.
15. Макаров, Р. И. Методы анализа данных : учеб. пособие / Р. И. Макаров, Е. Р. Хорошева – Владимир : Изд-во ВлГУ, 2021. – 216 с.
16. Малков, О. Б. Oracle SQL. Базовая часть : учеб. пособие – Омск : Изд-во ОмГТУ, 2019. – 132 с.
17. Методика и организация самостоятельной работы студентов: учебно-методическое пособие / Е. В. Ершов, Л. Н. Виноградова, В. В. Селивановских [и др.]. – Череповец: ФГБОУ ВПО ЧГУ, 2015. – 243 с.
18. Методический подход к созданию универсального пользовательского интерфейса / Г. В. Казаков, В. В. Корянов, В. В. Чемирисов, А. В. Уваров – Текст : электронный // Инженерный журнал: Наука и инновации. – 2020. – № 11.
19. Наумов Р. В. Актуальные языки программирования – Текст : электронный // Инженерный журнал: Academy. – 2016. – № 1.
20. Павлов, Л. А. Структуры и алгоритмы обработки данных : учебник / Л. А. Павлов, Н. В. Первова. / 2-е изд., испр. и доп. – СПб. : Лань, 2020. – 256 с.
21. Пацианский, М. React.js для начинающих – М.: ДМК Пресс, 2016. – 90 с.
22. Перепелица Ф. А. Эффективная разработка веб-сайтов. Bootstrap – СПб. : ИТМО, 2020. – 73 с.
23. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов – 2-е изд. – Москва : Издательство Юрайт, 2024. — 204 с.
24. Рогачева Ю. И. Анализ популярных СУБД и необходимость их применения в современных информационных системах / Ю. И. Рогачева, Д. А. Киржаев – Текст : электронный // Экономика и социум. – 2020. – № 7.
25. Рогов, Е. В. PostgreSQL 16 изнутри – М.: ДМК Пресс, 2024. – 664 с.
26. Романов В. Ю. Инструментарий для визуализации программного обеспечения в трехмерном пространстве / В. Ю. Романов, И. В. Шульга – Текст : электронный // International Journal of Open Information Technologies. – 2015. – № 8.
27. Савченко, Н. Л. Управление финансовыми ресурсами предприятия : учеб. пособие – Екатеринбург: Изд-во Урал. ун-та, 2019. – 164 с.
28. Сидорова, Д. Компания OutSystems: платформа для кодовой разработки и интеграции / Д. Сидорова. – Текст : электронный // rb.ru : [сайт]. – 2021. – 10 дек. – URL: https://rb.ru/story/outsystems/ (дата обращения: 26.11.2024).
29. Страуструп, Б. Язык программирования C++. Краткий курс, 2-е изд. : Пер. с англ. – СПб. : ООО «Диалектика», 2019. – 320 с.
30. Стренгхольт П. Масштабируемые данные. Лучшие шаблоны высоконагруженных архитектур – СПб. : Питер, 2022. — 368 с.
31. Уилке, К. Основы визуализации данных : пособие по эффективной и убедительной подаче информации : Пер. с англ. – М.: Эскмо, 2024. – 352 с.
32. Урсаке К. Автоматизация кредитного конвейера на основе BPM – Текст : электронный – 2021.
33. Флэнаган, Д. JavaScript. Подробное руководство, 5-е изд. : Пер. с англ. – СПб. : Символ-Плюс, 2008. – 992 с.
34. Хэнчетт Э. Vue.js в действии / Э. Хэнчетт, Б. Листуон – СПб. : Питер, 2015. – 304 с.
35. Цифровая трансформация: ожидания и реальность: докл. к XXIII Ясинской (Апрельской) междунар. науч. конф. по проблемам развития экономики и общества, Москва, 2022 г. [Текст] / Г. И. Абдрахманова, С. А. Васильковский, К. О. Вишневский [и др.] – М.: Изд. дом Высшей школы экономики, 2022. – 221 с.
36. Шор А. М. Сравнительный анализ подходов в разработке api веб-приложений – Текст : электронный // StudNet. – 2020. – № 1.
37. Шорохова, И. С. Статистические методы анализа : учеб. пособие / И. С. Шорохова, Н. В. Кисляк, О. С. Мариев – Екатеринбург : Изд-во Урал. ун-та, 2015. – 300 с.
38. Диаграмма Ганта: что такое и как построить — Skillbox [электр.ресурс] https://skillbox.ru/media/management/razbiraem-diagrammu-ganta-instrument-kotoryy-dolzhen-znat-kazhdyy-menedzher/. Дата обращения: 25.11.2024.
39. Единая экосистема FIS — Fisgroup [электр.ресурс]. https://fisgroup.ru/blog/ Дата обращения: 25.11.2024.
40. Оптимизация работы с базами данных — ITentika [электр.ресурс] https://itentika.ru/news/kak-optimizirovat-rabotu-s-bazami-dannykh. Дата обращения: 26.11.2024.
41. Учебник Angular для новичков — Code.mu [электр.ресурс] https://code.mu/ru/javascript/framework/angular/book/prime/. Дата обращения: 26.11.2024.
42. Что такое Visual Studio? — Microsoft Learn [электр.ресурс] https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022. Дата обращения: 26.11.2024.
43. Ant Design of React — Ant Design [электр.ресурс] https://ant.design/docs/react/introduce/. Дата обращения: 26.11.2024.
44. Eclipse Platform User Guide — Eclipse [электр.ресурс] https://help.eclipse.org/2024-09/index.jsp. Дата обращения: 26.11.2024.
45. FIS Platform: low-code платформа для автоматизации финансового сектора от компании FIS — IaaSSaaSPaaS [электр.ресурс] https://iaassaaspaas.ru/rating/low-code-sistemy/fis-platform-low-code-platforma-dlya-avtomatizatsii-finansovogo-sektora-ot-kompanii-fis. Дата обращения: 25.11.2024.
46. IntelliJ IDEA Documentation — JetBrains [электр.ресурс] https://www.jetbrains.com/help/idea/getting-started.html. Дата обращения: 25.11.2024.
47. Material-UI: Overview — Material-UI [электр.ресурс]. Дата обращения: 26.11.2024.
48. Mendix: quality and assurance — Mendix [электр.ресурс] https://www.mendix.com/platform/quality-assurance/. Дата обращения: 26.11.2024.
49. MySQL 8 для больших данных / Ш. Чаллавала, Дж. Лакхатария, Ч. Мехта, К. Патель : Пер. с англ. – М.: ДМК Пресс, 2018. – 226 с.
50. Tutorial: Get started with Visual Studio Code — Visual Studio Code [электр.ресурс] https://code.visualstudio.com/docs/getstarted/getting-started. Дата обращения: 26.11.2024.

# Приложение 1

Таблица П1.1

Диаграмма Ганта. План работ по проекту

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дата  Этап | Апр | Май | Июн | Июл | Авг | Сен | Окт | Нояб | Дек |
| 1.1 |  |  |  |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |  |  |  |
| 1.3 |  |  |  |  |  |  |  |  |  |
| 1.4 |  |  |  |  |  |  |  |  |  |
| 1.5 |  |  |  |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |  |  |  |
| 2.2 |  |  |  |  |  |  |  |  |  |
| 2.3 |  |  |  |  |  |  |  |  |  |
| 2.4 |  |  |  |  |  |  |  |  |  |
| 2.5 |  |  |  |  |  |  |  |  |  |
| 3.1 |  |  |  |  |  |  |  |  |  |
| 3.2 |  |  |  |  |  |  |  |  |  |
| 4.1 |  |  |  |  |  |  |  |  |  |