



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения
(ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: Разработка клиент-серверных приложений
по профилю: Разработка программных продуктов и проектирование
информационных систем
направления профессиональной подготовки: 09.03.04 «Программная
инженерия»

Тема: «Разработка системы автоматизации документооборота групп
студентов учебных курсов»

Студент: Петров Анатолий Валерьевич

Группа: ИКБО-03-18

Работа представлена к защите _____ (дата) _____ / Петров А.В. /
(подпись и ф.и.о. студента)

Руководитель: _____ ст.препод., к.т.н. Куликов А.А.

Работа допущена к защите _____ (дата) _____ / Болбаков Р.Г. /
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: _____

_____/_____/_____
_____/_____/_____

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших
защиту)



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

по дисциплине: Разработка клиент-серверных приложений

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 Программная инженерия

Студент: Петров Анатолий Валерьевич

Группа: ИКБО-03-18

Срок представления к защите: 17.12.2020 г.

Руководитель: Доцент кафедры ИиППО Болбаков Р.Г.

Тема: «Разработка системы автоматизации документооборота групп студентов учебных курсов»

Исходные данные: Документация и средства языка C#, MS Visual Studio.

Перечень вопросов, подлежащих разработке, и обязательного графического материала: 1. Провести анализ предметной области; 2. Провести анализ сравнения с существующими аналогами; 3. Разработать клиентскую часть приложения (принятие и обработка запросов); 4. Разработать алгоритм обработки данных; 5. Создать UML диаграммы структуры приложения.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка

Зав. кафедрой ИППО: _____ /Болбаков Р.Г./, «_____» _____ 2020 г.

Задание на КР выдал: _____ /Болбаков Р.Г./, «_____» _____ 2020 г.

Задание на КР получил: _____ /Петров А.В./, «_____» _____ 2020г.

Петров А.В. «Разработка системы автоматизации документооборота групп студентов учебных курсов» / Курсовая работа по дисциплине «Разработка клиент-серверных приложений» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки магистратуры 09.03.04 «Программная инженерия» (5ый семестр) / Доцент кафедры ИиППО Болбаков Р.Г / кафедра ИППО Института ИТ МИРЭА.

Целью работы является разработка системы автоматизации документооборота групп студентов учебных курсов.

В рамках работы осуществлены анализ предметной области, сравнение с существующими аналогами, разработана клиентская часть приложения (принятие и обработка запросов).

Petrov A.V. « Development of a document management system for groups of students of training courses »/ Coursework in the discipline “Development of client-server applications” of the profile “Development of software products and information systems design” in the direction of professional education for bachelor degree 09.03.04. "Software Engineering" (5th semester) / associate professor of IPPO Bolbakov R.G. / department IPPO of institute IT RTU MIREA.

The aim of the work is to develop a system for automating the workflow of groups of students of training courses.

As part of the work, the analysis of the subject area, comparison with existing analogs, developed the client side of the application (accepting and processing requests).

МИРЭА. Ин-т ИТ. Каф. ИиППО. 2020 г. Петров А.В.

РЕФЕРАТ

В рамках курса «Разработка клиент-серверных приложений» в соответствии с заданием на выполнение курсовой работы необходимо создать программу «Система автоматизации документооборота групп студентов учебных курсов» для работы с документами студентов, обработки, вывода полученного результата (таблицы данных и сообщения о действиях), возможность добавлять новых студентов, удалять, выводить информацию о студентах по группам и курсам, а также изменять данные о студентах, возможность просмотреть всех студентов сразу и получить всю информацию, включая ФИО, группу, курс, в том числе остальные данные о документах каждого студента.

Объём пояснительной записки к курсовой работе составляет 27 страниц. Пояснительная записка состоит из четырёх главных разделов – анализа предметной области, методологии разработки приложения, реализации приложения и руководства по использованию приложения.

В разделе анализа предметной области были разобраны основные аналоги, выведены их достоинства и недостатки, а также выделены дополнительные действительно полезные функции.

В разделе методологии разработки приложения описаны программные средства и решения, которые были использованы для проектирования и разработки приложения и базы данных.

В разделе реализации приложения продемонстрированы задачи, поставленные перед разработчиком, процесс их выполнения, демонстрация интерфейса приложения, пояснение функций, а также зафиксирован сам процесс выполнения основных требований к программе.

В разделе руководства по использованию приложения написано, как пользователю ориентироваться в приложении.

В заключение подводится итог всей курсовой работы, а также пройденному за курс материалу.

ОГЛАВЛЕНИЕ

РЕФЕРАТ	2
СПИСОК СОКРАЩЕНИЙ.....	5
ВВЕДЕНИЕ.....	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1 Описание предметной области	8
1.2 Анализ аналогов разрабатываемого приложения	8
1.2.1 «Dialog»	8
1.2.2 «Kih»	8
1.2.3 «Burokratgroup»	8
1.3 Выводы к разделу 1	8
1.3.1 Границы применимости сервера	9
1.3.2 Границы применимости клиента	9
2 МЕТОДОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ.....	10
2.1 Выбор языка разработки.....	10
2.2 Выбор инструментов для разработки интерфейса приложения	10
2.3 Выбор среды для разработки приложения и БД.....	10
2.4 Выводы к разделу 2.....	11
3 РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ	12
3.1 Стадии и этапы разработки	12
3.2 Структурная схема приложения	12
3.3 UML-диаграмма структуры БД	14
3.4 UML-диаграмма деятельности.....	14
3.5 Разработка интерфейса приложения	17
3.5.1 Общие сведения о разработке интерфейса	17
3.5.2 Главная страница - страница пользователя	17
3.5.3 Страница входа.....	19
3.6 Выводы к разделу 3	20
4 РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРИЛОЖЕНИЯ	21
4.1 Описание главной страницы сайта.....	21
4.2 Описание страницы регистрации	22
4.3 Описание страницы входа в аккаунт.....	Error! Bookmark not defined.
4.4 Выводы к разделу 4.....	22
ЗАКЛЮЧЕНИЕ	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24
ПРИЛОЖЕНИЕ	25

СПИСОК СОКРАЩЕНИЙ

ПК – Персональный компьютер;

ОС – Операционная система;

СУБД – Система управления базами данных;
БД – База данных;
ООП – Объектно-ориентированное программирование;
HTML – HyperText Markup Language (Язык гипертекстовой разметки);
CSS – Cascading Style Sheets (Каскадные таблицы стилей);
IDE – Integrated Development Environment (Интегрированная среда разработки);
SQL – Structured Query Language (Язык структурированных запросов);
XML – eXtensible Markup Language (Расширяемый язык разметки);
HTTP – HyperText Transfer Protocol (Протокол передачи гипертекста);
ПО – Программное обеспечение;
UML – Unified Modeling Language (Унифицированный язык моделирования).

ВВЕДЕНИЕ

В качестве темы для своей курсовой работы я выбрал разработку приложения «Система автоматизации документооборота групп студентов учебных курсов».

Для этого большинство сайтов используют систему автоматизации документооборота. Например, для изменения данных о студенте и просмотр его документов.

В процессе разработки курсовой работы планируется приобрести компетенцию ПК-13 – готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание предметной области

Предметной областью для данной курсовой работы является сайт для работы с данными о студентах и просмотра их секретарями.

1.2 Анализ аналогов разрабатываемого приложения

Лучший способ проанализировать предметную область – сопоставить аналоги на рынке. Сравнения проводились с другими сайтами, на которых реализована система автоматизации документооборота.

1.2.1 «Dialog»

Сайт «Dialog» использует систему эффективной автоматизации делопроизводства для обеспечения доступности управления документами [1].

1.2.2 «Kih»

Сайт «Kih» позволяет выводить всю нужную информацию на официальный сайт непосредственно из системы. Есть возможность управления деятельностью на уровне СУБД [2].

1.2.3 «Burokratgroup»

Сайт «Burokratgroup» предоставляет готовое решение для автоматизации документооборота организации, а также помогает оптимизировать процессы канцелярии [3].

1.3 Выводы к разделу 1

Исходя из проведённого анализа, можно сделать вывод, что в основном функционал сайтов примерно одинаков.

С учетом особенностей рассмотренного выше программного обеспечения, можно поставить следующие критически необходимые задачи для реализации собственного приложения:

- 1) создание базы данных для хранения информации
- 2) реализация функции добавление, удаление и управления списком студентов
- 3) создание несколько страниц для удобства просмотра студентов по группам и курсам

Также были определены границы применимости приложения.

1.3.1 Границы применимости сервера

Системные требования для данного приложения: ОС Windows 10 x64, 1 ГБ оперативной памяти.

Установленная СУБД SQL Server с уже созданной базой данных «DocumentsDB» с соответствующей структурой.

Количество хранимых пользователей и других данных в БД зависит от объекта жёсткого диска системы.

1.3.2 Границы применимости клиента

Для использования в качестве клиента подойдёт любой современный браузер на полновесной ОС.

2 МЕТОДОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ

2.1 Выбор языка разработки

Для разработки приложения был выбран язык программирования С#, объектно-ориентированный язык программирования. Именно на С# возможно создать многие функции сайтов и клиент-серверных приложений довольно быстро и несложно, поэтому этот язык отлично подходит для функционала сайта [4].

2.2 Выбор инструментов для разработки интерфейса приложения

Приложение представляет из себя сайт, поэтому обязательным атрибутом разработки является браузер, им выступает Mozilla Firefox для отображения пользовательского интерфейса.

Для оформления сайта будет использоваться базовые возможности HTML5, CSS3.

2.3 Выбор среды для разработки приложения и БД

В качестве среды разработки выступает MS Visual Studio [5], которая предоставляет удобные средства для веб-разработки, дает доступ к использованию множеству технологий, поддерживает использование многих языков программирования, инструменты для баз данных, поддержку SQL и многое другое.

В качестве СУБД для управления данными была использована широко-распространённая SQL Server [6], предоставляющая удобные средства для создания баз данных, обладающая своим инструментом для визуализации данных в виде диаграмм, SQL Server Management Studio, которая позволяет удобно писать запросы.

Для обращения к БД, а именно: добавление, вывод, удаление, проверки существования записи, использовался также язык С# [7], благодаря его гибкости и простоте.

Основной платформой разработки веб-приложений выбрана ASP.NET, входящая в состав .NET Framework, потому что она предоставляет нужный инструментарий для решения поставленной задачи и легкодоступна в Visual Studio [8].

2.4 Выводы к разделу 2

Для разработки клиент-серверного приложения (сайта) на С# были выбраны и задействованы проверенные решения для улучшения хода и скорости разработки, а также качества итогового продукта.

3 РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

3.1 Стадии и этапы разработки

Таблица 1 Содержание этапов работ

№ этапа	Содержание работ	Срок выполнения
1	Исследование концепции	1-2 неделя
2	Выработка требования	3-4 неделя
3	Проектирование	4-6 неделя
4	Реализация компонент: написание логики программы	7-11 неделя
5	Интеграция компонент: написание графического интерфейса	9-12 неделя
6	Тестирование и отладка созданного приложения	13-14 неделя
7	Оформление проектной документации	15-16 неделя
8	Сдача проекта и защита курсовой работы	17 неделя

3.2 Структурная схема приложения

Структурная схема приложения представлена на рисунке 3.1.

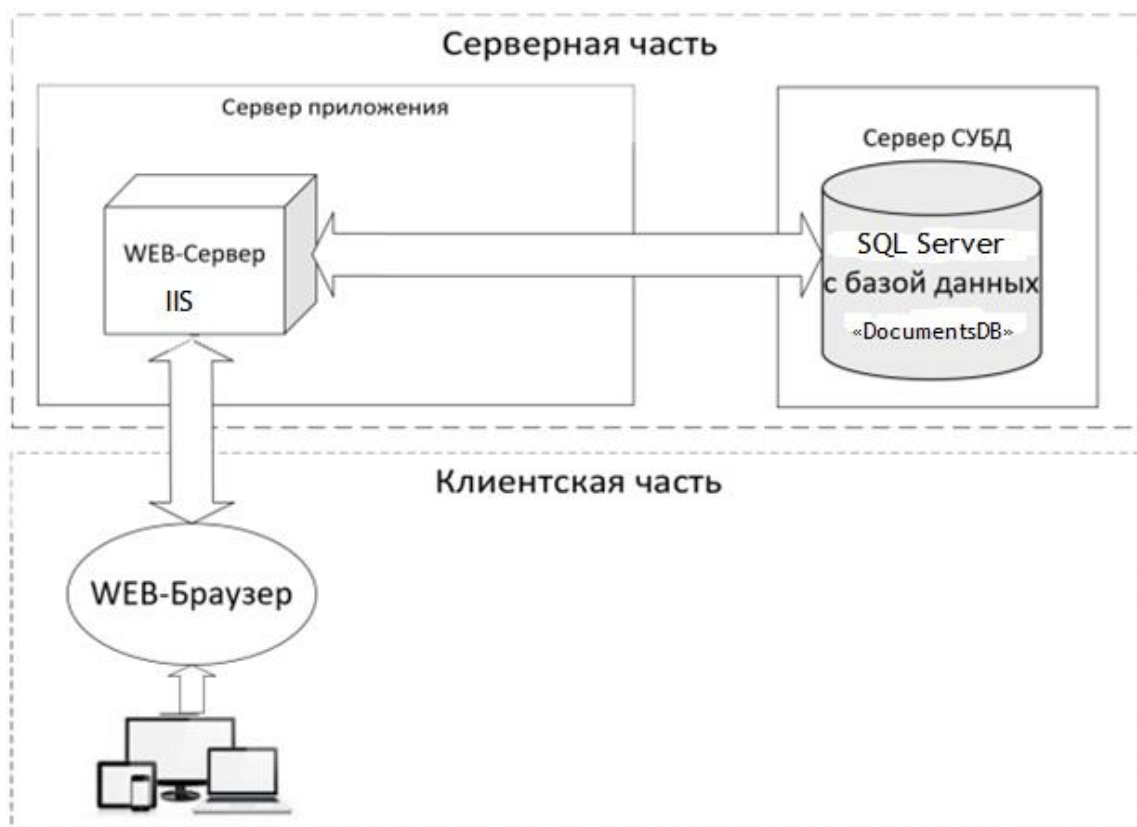


Рисунок 3.1 – Структурная схема приложения

Структура клиент-серверного приложения состоит из серверной и клиентской части.

На стороне серверной части находится два сервера: непосредственно сервер приложения и сервер СУБД SQL Server. Первый отвечает за приём запросов веб-браузера и других устройств по протоколу HTTP, запроса разбирается и в зависимости от запрошенных действий клиенту выдаётся ответ. IIS отвечает за обращение к серверу СУБД. IIS — это компонент Microsoft Windows, популярный и распространенный в силу простоты конфигурации веб-сервер.

Сервер SQL Server отвечает за хранения базы данных, возможность обращения к данным через SQL-синтаксис, а также выполнения триггеров, событий и других операций.

На стороне клиент находится непосредственно физическое устройство пользователя, а также веб-браузер или другое ПО, которое позволяет выполнять основные HTTP-запросы на сервер приложения.

3.3 UML-диаграмма структуры БД

На рисунке 3.2 изображена UML-диаграмма БД, спроектированная с помощью SQL Server Management Studio.

Students			
	Имя столбца	Тип данных	Разрешить значения...
	ID	int	<input type="checkbox"/>
	ФИО	nvarchar(50)	<input type="checkbox"/>
	Группа	nvarchar(50)	<input type="checkbox"/>
	Курс	int	<input type="checkbox"/>
	[Вид возмещения затрат]	nvarchar(50)	<input type="checkbox"/>
	[Форма обучения]	nvarchar(50)	<input type="checkbox"/>
	[Студентический билет продлен]	bit	<input type="checkbox"/>
	[Количество долгов]	int	<input type="checkbox"/>
	[Количество взятых книг из библиотеки]	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 3.2 – UML-диаграмма структуры БД

3.4 UML-диаграмма деятельности

В рамках курсовой работы были реализованы следующие задачи:

- Добавление студента
- Удаление студента
- Изменить данные о студенте
- Просмотр всех студентов (выбор определенной группы, всех групп одного курса или всех студентов со всех курсов)

Реализованные задачи представлены на UML-диаграммах деятельности (рисунок 3.3-3.6).

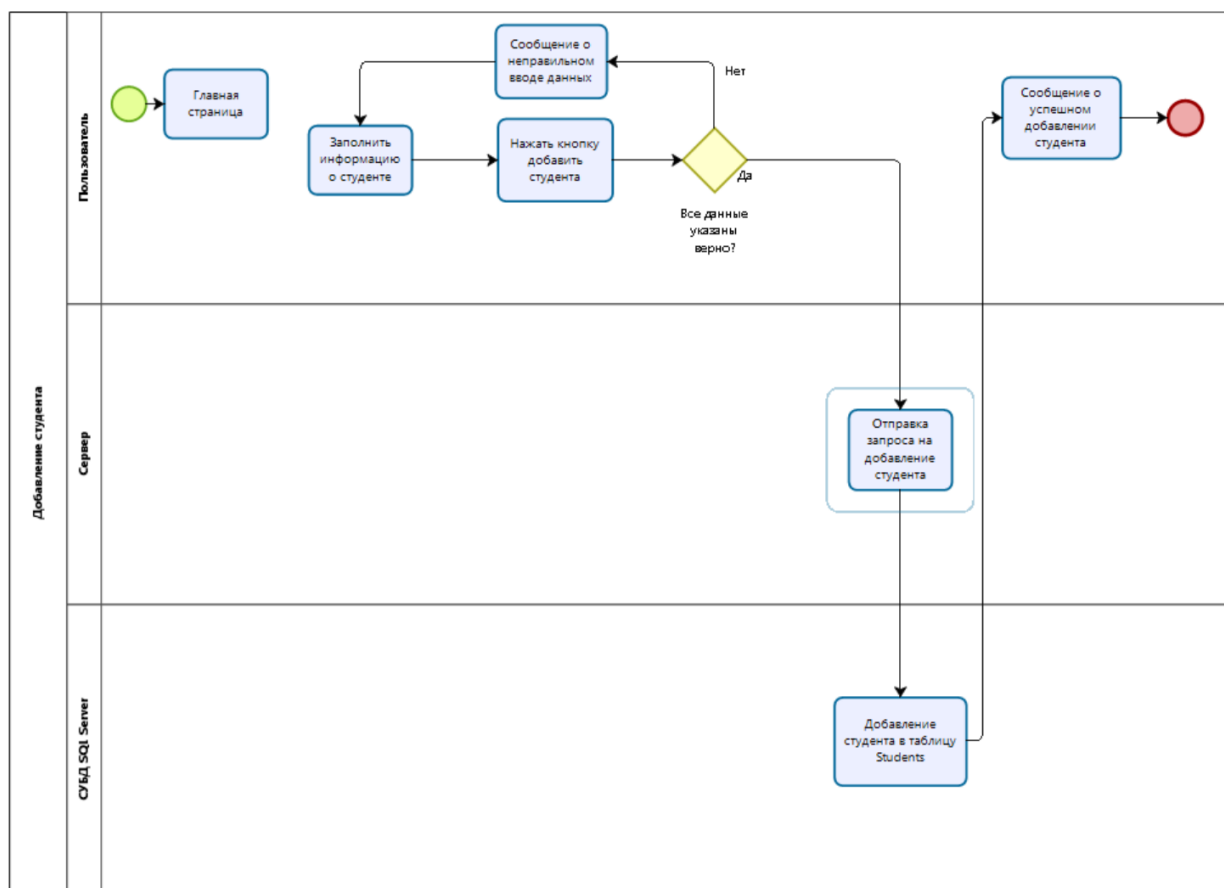


Рисунок 3.3 – Диаграмма задачи добавления студента

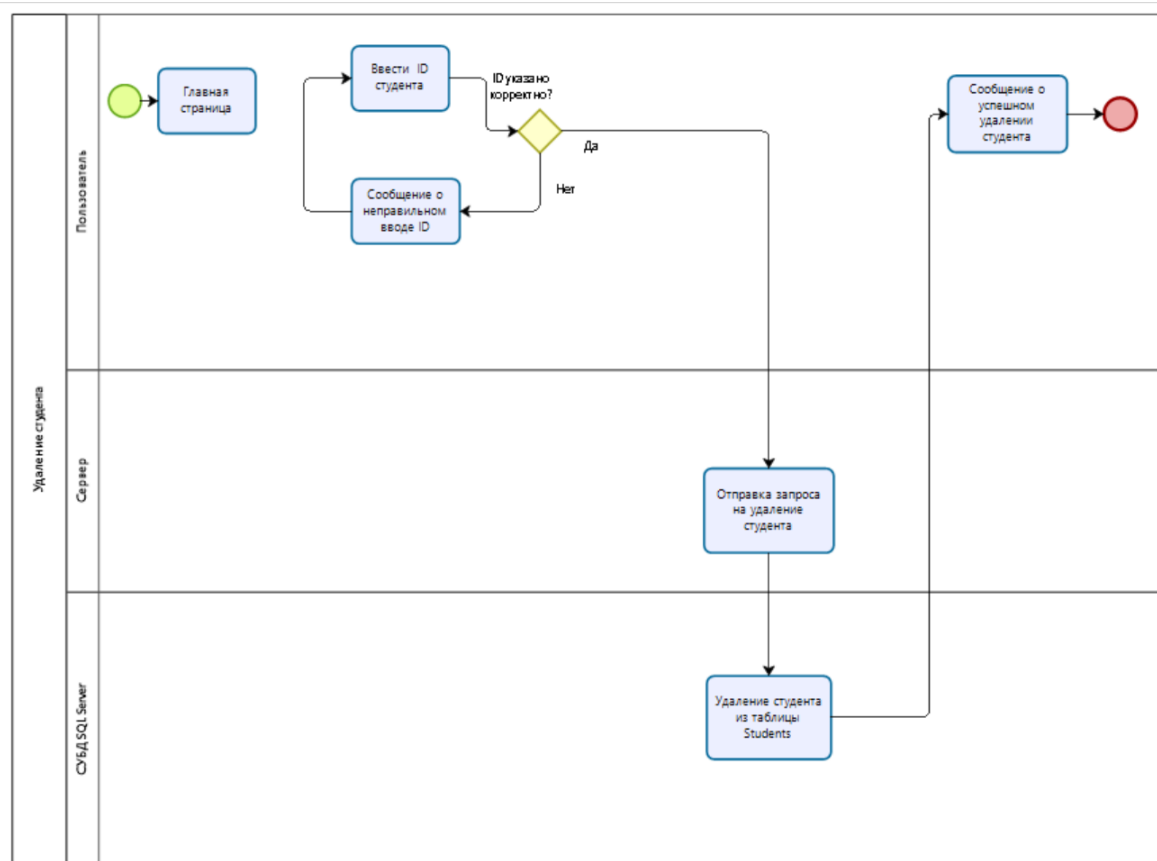


Рисунок 3.4 – Диаграмма задачи удаление студента

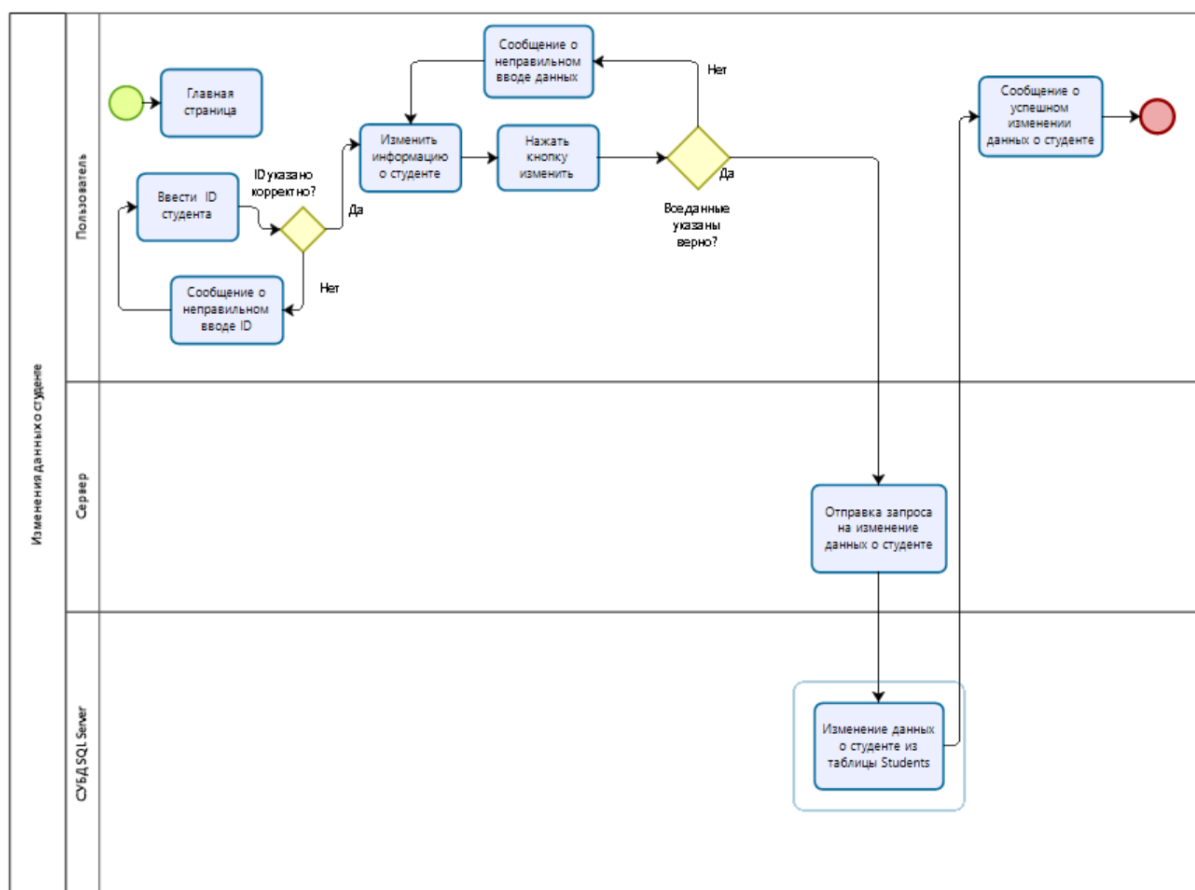


Рисунок 3.5 – Диаграмма задачи изменения данных о студенте

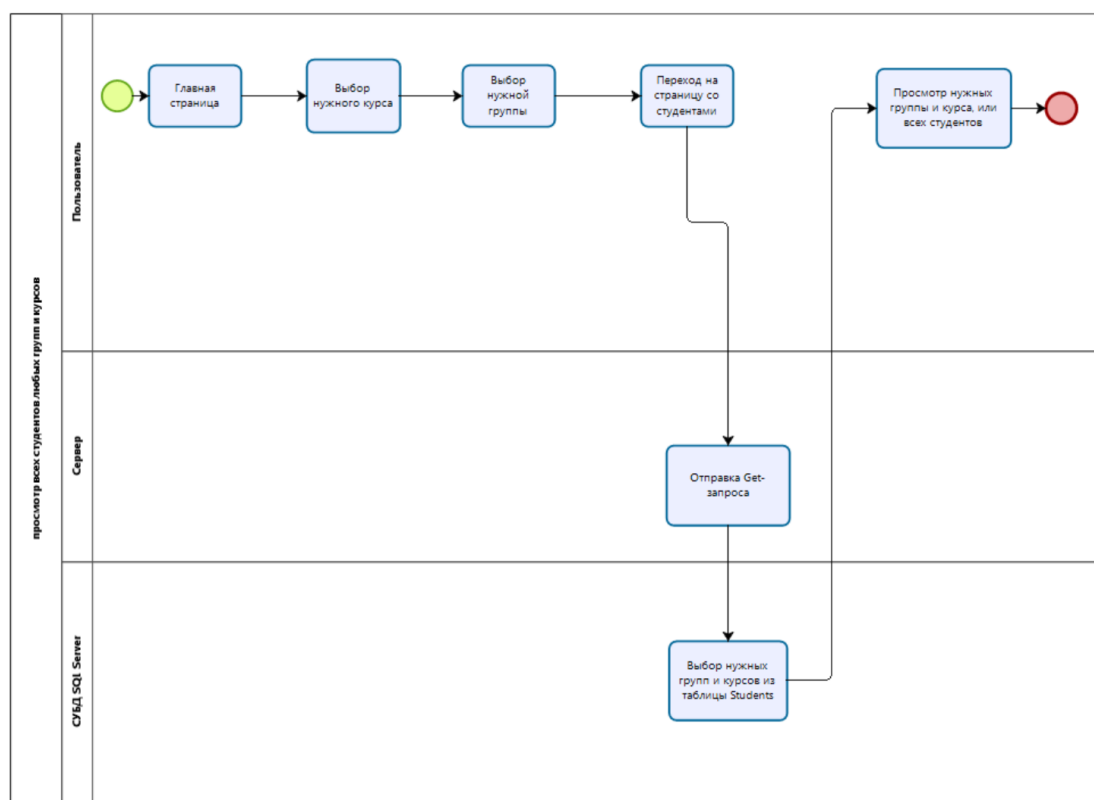


Рисунок 3.6 – Диаграмма задачи просмотра всех студентов любых групп и курсов

3.5 Разработка интерфейса приложения

3.5.1 Общие сведения о разработке интерфейса

Как было сказано во второй главе, для создания приемлемого пользовательского интерфейса был использован HTML, CSS.

3.5.2 Главная страница - страница для работы с данными

На главной странице можно увидеть картинку с эмблемой РТУ МИРЭА, а также инструменты для ввода информации о студенте, в остальном же главная страница и страница со студентами реализованы по одному шаблону. Но все же стоит отметить, что главная страница предназначена именно для изменения данных студента, а вторая — для вывода таблиц студентов.

Главная страница представлена на рисунке 3.7.

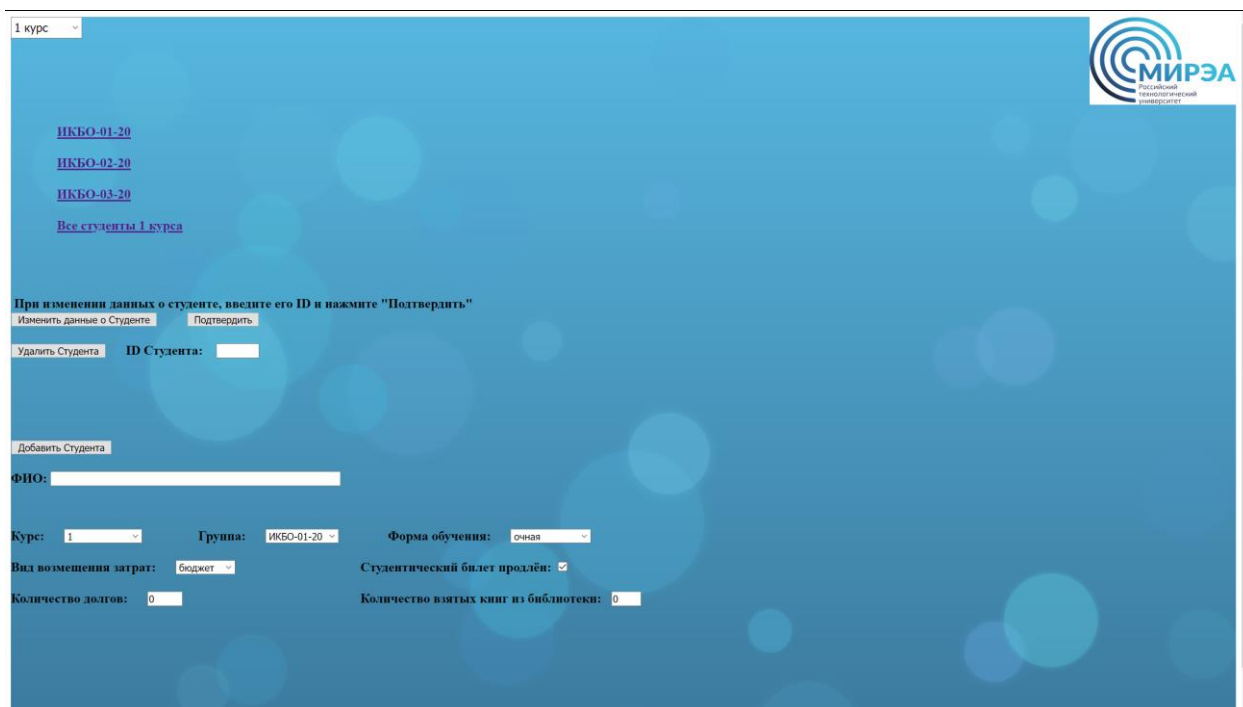


Рисунок 3.7 – главная страница сайта

Элементы управления данными сайта изображены на рисунках 3.8 и 3.9.

При изменении данных о студенте, введите его ID и нажмите "Подтвердить"

Изменить данные о Студенте Подтвердить

Удалить Студента ID Студента:

Рисунок 3.8 – Элементы управления

Добавить Студента

ФИО:

Курс: Группа: Форма обучения:

Вид возмещения затрат: Студентический билет продлён: ☒

Количество долгов: Количество взятых книг из библиотеки:

Рисунок 3.9 – Элементы управления

Также на главной странице можно перейти на вторую страницу с таблицей информации о нужных студентах с помощью ссылок на рисунке 3.10 и выбрать курс (рисунок 3.11), который поменяет ссылки уже на другие исходя из вызова соответствующего курса.

[ИКБО-01-20](#)

[ИКБО-02-20](#)

[ИКБО-03-20](#)

[Все студенты 1 курса](#)

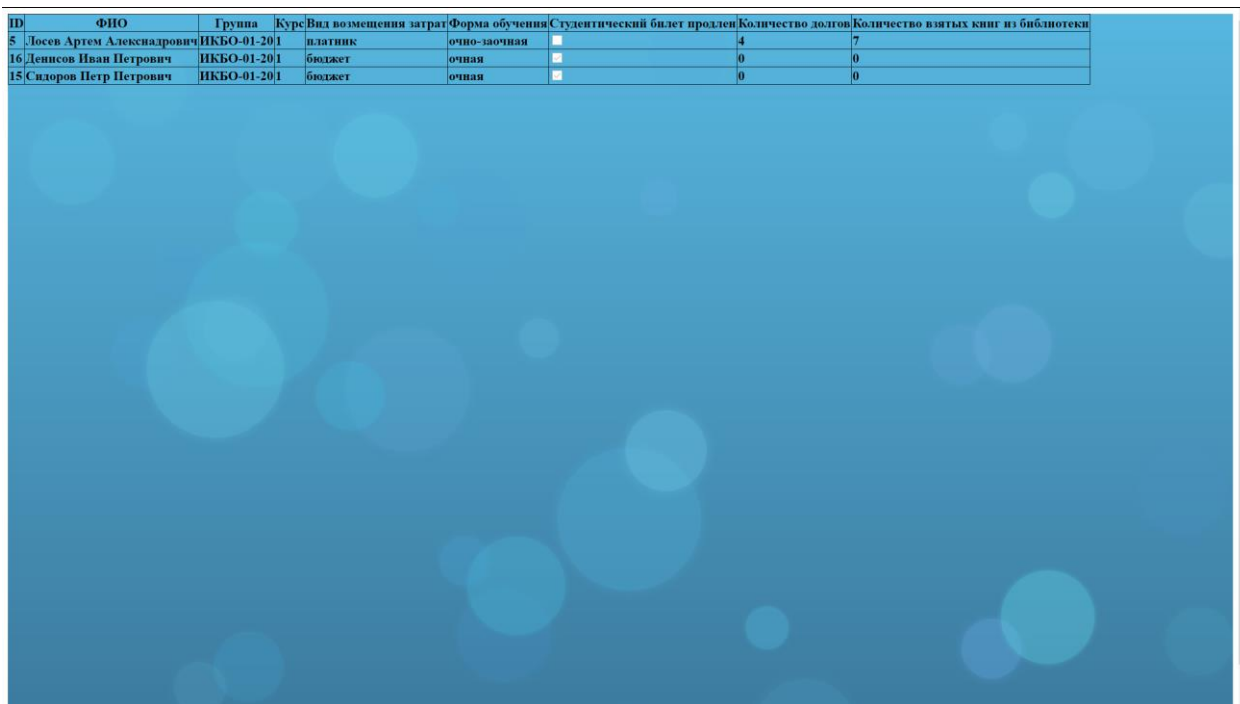
Рисунок 3.10 – Ссылки на таблицы

▾

Рисунок 3.11 – Элемент для выбора курса

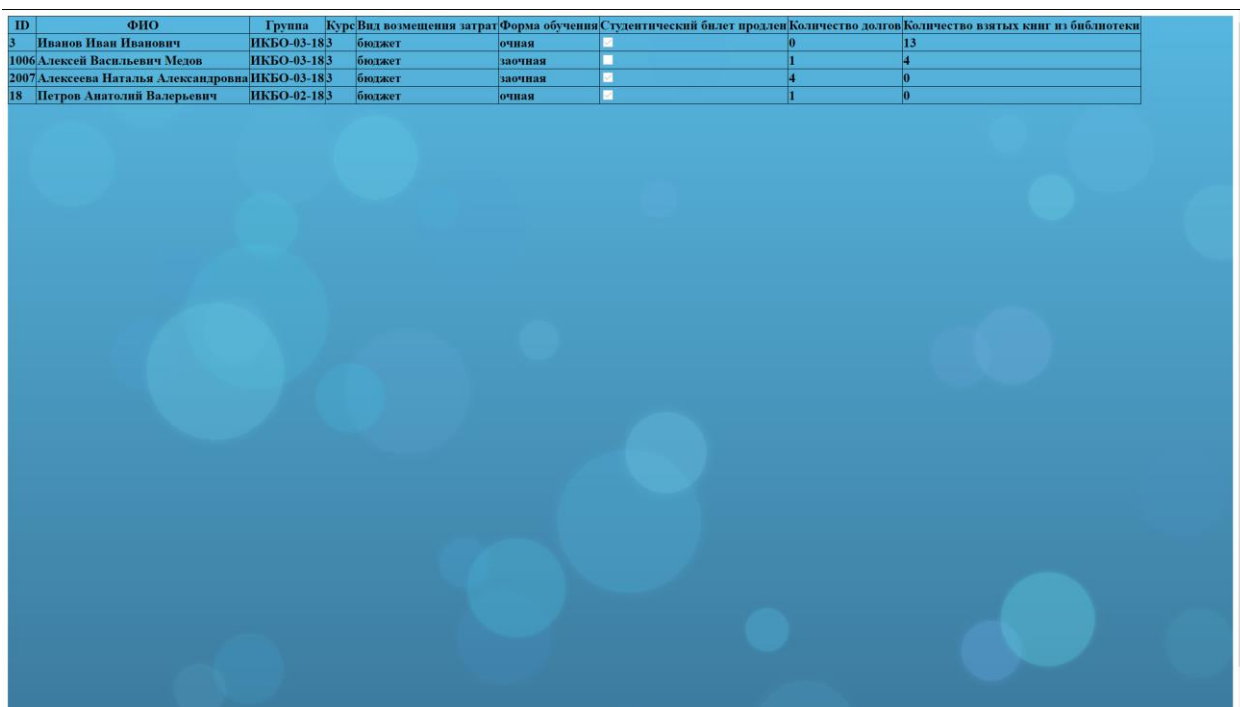
3.5.3 Страница с таблицей

На ней отображается таблица со студентами и их данными: курс, группа, количество долгов и т.д. Страницы с таблицей изображены на рисунках 3.13 и 3.14 в зависимости от выбранной группы.



ID	ФИО	Группа	Курс	Вид возмещения затрат	Форма обучения	Студентический билет продлен	Количество долгов	Количество взятых книг из библиотеки
5	Лосев Артем Александрович	ИКБО-01-20	1	платник	очно-заочная	<input checked="" type="checkbox"/>	4	7
16	Денисов Иван Петрович	ИКБО-01-20	1	бюджет	очная	<input checked="" type="checkbox"/>	0	0
15	Сидоров Петр Петрович	ИКБО-01-20	1	бюджет	очная	<input checked="" type="checkbox"/>	0	0

Рисунок 3.13 – Страница с таблицей студентов группы ИКБО-01-20



ID	ФИО	Группа	Курс	Вид возмещения затрат	Форма обучения	Студентический билет продлен	Количество долгов	Количество взятых книг из библиотеки
3	Иванов Иван Иванович	ИКБО-03-18	3	бюджет	очная	<input checked="" type="checkbox"/>	0	13
1006	Алексей Васильевич Мелов	ИКБО-03-18	3	бюджет	заочная	<input checked="" type="checkbox"/>	1	4
2007	Алексеева Наталья Александровна	ИКБО-03-18	3	бюджет	заочная	<input checked="" type="checkbox"/>	4	0
18	Петров Анатолий Валерьевич	ИКБО-02-18	3	бюджет	очная	<input checked="" type="checkbox"/>	1	0

Рисунок 3.14 – Страница с таблицей студентов 3 курса

3.6 Выводы к разделу 3

В третьем разделе:

- были продемонстрированы UML-диаграммы задач и диаграмма структуры приложения;
- были разработаны и продемонстрированы функции приложения и его интерфейс;
- была спроектирована логика приложения.

4 РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРИЛОЖЕНИЯ

4.1 Описание главной страницы сайта

Когда пользователь оказывается на главной странице, перед ним предстаёт интерфейс сайта, поля для ввода данных, кнопки управления функционалом (добавить студента, изменить данные о студенте, а затем подтвердить, удалить студента), ссылки на группы, несколько элементов для выбора вариантов при заполнении данных о студенте, дабы сохранить логику (студент 1 курса не может учиться в группе 4 курса и т.д.), а также элемент, позволяющий выбрать курс и группу и перейти по ссылке на страницу с данными о студенте.

При нажатии на кнопку «добавить», начинается проверка всех заполненных полей на корректность введенных данных, чтобы исключить добавление неверной информации. Если данные верные, студент добавляется в группу.

При нажатии на кнопку «изменить», все поля заполняются исходными данными студента, чье ID указали в поле перед нажатием. Перед нажатием на кнопку «подтвердить», начинается проверка всех заполненных полей на корректность измененных данных, чтобы исключить добавление неверной информации. Если данные верные, информация о студенте изменяется.

Перед нажатием на кнопку «удалить», пользователь должен ввести ID студента, чтобы функция сработала. При правильном ID студент удалится из группы. Пример сообщения об ошибке представлен на рисунке 4.1.



Студента с ID = 0 нет или ID не указано

Рисунок 4.1 – Сообщение о не корректном ID

Элемент DropDownList с номером курса позволит отобразить ссылки на группы, которые отобразят таблицу с соответствующими данными.

4.2 Описание страницы с таблицей

На странице с самими данными, пользователь может наблюдать всех студентов из выбранной группы или курса и всей ранее указанной информацией. При необходимости, можно вернуться с помощью стрелки браузера назад на главную страницу.

4.3 Выводы к разделу 4

Было написано руководство по использованию и эксплуатации приложения исходя из созданных и принятых решений в отношении пользовательского интерфейса на этапе разработки серверной и клиентской частей.

ЗАКЛЮЧЕНИЕ

В процессе выполнения данной курсовой работы была создано клиент-серверное приложение - сайт для автоматизации документооборота студентов. Спроектированное приложение дает пользователям возможность добавлять, удалять и изменять данные о студентах, а также отображать нужные курсы и группы по отдельным ссылкам. Программа обладает достаточным функционалом, чтобы её можно было использовать, как плацдарм для будущих разработок в улучшении функционала и общей системы.

Создание такого приложения позволило закрепить знания и навыки написания клиент-серверных приложений, а также научиться разрабатывать, проектировать и тестировать программы, актуальные в современном мире.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Dialog [электронный ресурс]. Режим доступа: <http://dialog.krtech.ru/> – Заглавие с экрана. (Дата обращений: 25.01.2021).
2. kih [электронный ресурс]. Режим доступа: <http://www.kih.ru/> – Заглавие с экрана. (Дата обращений: 24.01.2021).
3. Burokratgroup [электронный ресурс]. Режим доступа: <http://burokratgroup.ru/> – Заглавие с экрана. (Дата обращений: 23.01.2021).
4. Албахари Б. С 7.0 Справочник. / Патрик Нимейер, Дэниел Леук.— 4-е изд.— М.: Эксмо, 2014— 1214 с.
5. Мэтью М. Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0: – М.: Вильямс, 2013. – 1018 с.
6. Крис Фиайли. SQL. Руководство по изучению языка— ДМК Пресс, 2003.— 456 с.
7. Рихтер Д. CLR via C#: – СПб.: Питер, 2017. – 896 с.
8. Вагнер Б. Наиболее эффективное программирование на C#: – М.: Вильямс, 2017. – 240 с.

ПРИЛОЖЕНИЕ

Приложение 1: Техническое задание на разработку ПО

Введение

Составленное техническое задание по дисциплине “Разработки клиент-серверных приложений” является документом к курсовой работе, который отражает все этапы разработки программного продукта, а также процесс проектирования и выявление требований, предъявляемых конечному продукту.

Основание для разработки

Основанием для разработки выступает тот факт, что сейчас вузам необходимо реализовать современные системы управления сайтом для автоматизации системы документооборота студентов.

Назначение разработки

Создание приложение-сайт под названием «Documents», где можно будет добавлять, удалять, изменять студентов, и иметь удобный сервис для того, чтобы посмотреть на имеющиеся документы и прочие данные у студентов разных групп и курсов.

Требования, предъявляемые к программе

Взаимодействие продукта

Программа должна реализовывать переход к другим страницам сайта, при нажатии соответствующих ссылок на странице, где находится пользователь сайта в данный момент. Также должны высказывать соответствующие сообщения об ошибке или важной информации для клиента.

Функциональные требования

На главной странице должно быть реализовано добавление, изменение и удаление записей о студентах, возможность перейти на страницу с таблицей нужной группы.

Страница с таблицей представляет из себя набор записей в БД, которые отображаются в таблице в соответствии с выбранной группой и курсом.

Все эти функции должны корректно работать и взаимодействовать друг с другом.

Требования к внешним интерфейсам

Программный продукт должен корректно отображать данные, которые подаются пользователем на ввод, избегая проблем с чтением и восприятием информации (за исключением тех случаев, когда приложение запускается на мелких устройствах или размер окна браузера слишком мал).

Требования к производительности

ПО на стороне сервера может использовать большую часть ресурсов системы, т.к. для серверным частям приложения обычно устройства выделяются целиком (должны оставаться ресурсы для работы сервера СУБД и другого дополнительного ПО).

Программный продукт на стороне клиента должен использовать ресурсы системы оптимизированно, избегая чрезмерного потребления оперативной памяти в браузере клиента.

Нефункциональные требования

Программа должна корректно обрабатывать различные исключения и ошибки программы для избавления ситуация с потерей или повреждением данных, как внутренних, так и внешних, а также нарушением стабильности работы устройства пользователя.

Требования к программной документации

1) Пояснительная записка в соответствии со стандартами РТУ МИРЭА (СМКО МИРЭА 7.5.1/04.И.05-16);

2) Проектная документация, составленная в соответствии с ГОСТ.

В процессе создания приложения вся проделанная работа документируется, должны быть сохранены все детали разработки, а также трудности, с которыми пришлось столкнуться. Всё перечисленное должно быть отражено в пояснительной записке, которая прилагается проекту.

Технико-экономические показатели

Клиентская часть приложения «Documents» доступна всем пользователям интернета. Серверная часть использована на базовом уровне на ПС, входящая в инструментарий Visual Studio.

Стадии и этапы разработки

В «жизненный цикл» приложения включены следующие этапы:

- 1) Исследование концепции (02.09.20-16.09.20);
- 2) Выработка требования (16.09.20-30.09.20);
- 3) Проектирование (30.09.20-21.10.20);
- 4) Реализация компонент: написание логики программы (21.10.20-11.11.20);
- 5) Интеграция компонент: написание графического интерфейса (28.03.20-18.11.20);
- 6) Тестирование и отладка созданного приложения (18.11.20-09.12.20);
- 7) Оформление проектной документации (09.12.20-15.12.20);
- 8) Сдача проекта и защита курсовой работы (17.12.20-29.12.20).

Дальнейшее сопровождение программы после сдачи и защиты курсовой работы не предусмотрено.

Порядок контроля и приёмки

Контроль, приём и проверка программного продукта осуществляется исходя из различных требований к программе в соответствии с техническим заданием.

Требования, обозначенные в техническом задании, позволят оценить корректность работы всех заявленных функций.