|  |  |  |
| --- | --- | --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** | | |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** | | |
|  | | |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** | | |
|  | | |
| **ОТЧЁТ** | | |
| ПО УЧЕБНОЙ ПРАКТИКЕ  ПРОФЕССИОНАЛЬНОГО МОДУЛЯ | | |
| ПМ.01. «Разработка модулей программного обеспечения для компьютерных систем» | | |
|  | | |
| студента группы 090207-9о-20/2 Асылбек уулу Бакыта,  обучающегося специальности  09.02.07 «Информационные системы и программирование» | | |
|  | | |
|  | | |
| Период практики | | |
| С «08» декабря 2023 г. по «28» декабря 2023 г. | | |
|  | | |
|  | | |
| Руководитель практики: Кочетков Станислав Сафирович | | |
| Выполнил студент: Асылбек уулу Бакыт | | |
|  | | |
|  | Студент |  |
|  | Руководитель практики |  |
|  | Оценка |  |

Москва

2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc160101099)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc160101100)

[ЗАКЛЮЧЕНИЕ 13](#_Toc160101101)

# ВВЕДЕНИЕ

Выполнение заданий УП.01.01. Учебной практики ПМ.01. Сопровождение и обслуживание программного обеспечения компьютерных систем подразумевает освоение ряда умений и компетенций.

Данная учебная практика формирует следующие профессиональные компетенции:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

В ходе работы надо получить практический опыт:

* в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
* использовании инструментальных средств на этапе отладки программного продукта;
* проведении тестирования программного модуля по определенному сценарию;
* разработке мобильных приложений.

Во время учебной практики должны быть приобретены навыки:

* осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
* создавать программу по разработанному алгоритму как отдельный модуль;
* выполнять отладку и тестирование программы на уровне модуля;
* осуществлять разработку кода программного модуля на современных языках программирования;
* уметь выполнять оптимизацию и рефакторинг программного кода;
* оформлять документацию на программные средства.

Помимо этого, были получены общие компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке

ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере.

# ОСНОВНАЯ ЧАСТЬ

Данная учебная практика состоит из 11 работ.

Практическая работа №1 требовала составить технические задания проекта, выполняемые в рамках этой учебной практики, которые следующим образом:

1. Спроектировать базу данных, включающую таблицы для пользователей, семестров, недель, предметов, типов комнат, кабинетов, преподавателей, групп, пар, дней, которая будет использоваться приложением.
2. Разработать приложение на языке C#, взаимодействующее с базой данных, обеспечивающее возможности добавления, редактирования, удаления, сортировки записей и вывода информации о сущностях.
3. Разработать модуль работы с captcha для настольного приложения на C#.
4. Создать пользовательский интерфейс для настольного приложения на C#, обеспечивающий удобное взаимодействие пользователя с приложением.
5. Разработать модуль приложения для вывода данных о сущностях в excel файлы.
6. Протестировать модули настольного приложения для обеспечения их корректной работы.
7. Создать модули мобильного приложения на языке Kotlin: авторизация пользователей и вывод информации из сущностей базы данных, а также их фильтрацию.
8. Спроектировать пользовательский интерфейс для мобильного приложения на языке Kotlin, обеспечивающий удобство использования функционала.
9. Провести тестирование модулей мобильного приложения для проверки их корректной работы.
10. Необходимо составить техническую документацию, описывающую разработанные приложения, их функциональные возможности, структуру, требования к установке и использованию.

Практическая работа №2 состояло в проектировании базы данных для хранения информации, с помощью которой можно составлять расписание, и описать связи между сущностями данной базы данных.

Логическая модель состоит из десяти таблиц:

* Таблица «Предметы» будет содержать в себе:

1. идентификатор предмета;
2. наименование предмета;
3. количество лекционных часов;
4. количество практических часов;
5. итоговое количество часов;
6. часы для консультации;
7. тип зачета;

* Таблица «Преподаватели» будет содержать в себе:

1. идентификатор преподавателя;
2. фамилия преподавателя;
3. имя преподавателя;
4. отчество преподавателя;

* Таблица «Группы» будет содержать в себе:

1. идентификатор группы;
2. код группы;
3. короткое наименование группы;
4. количество студентов.

* Таблица «Типы аудиторий» будет содержать в себе:

1. идентификатор типа аудитории;
2. наименование типа аудитории;
3. описание типа аудитории;

* Таблица «Аудитории» будет содержать в себе:

1. идентификатор аудитории;
2. идентификатор типа аудитории;
3. количество мест в аудитории;
4. количество мест для студентов;
5. номер аудитории;

* Таблица «Семестры» будет содержать в себе:

1. идентификатор семестра;
2. год семестра;
3. номер семестра;

* Таблица «Недели» будет содержать в себе:

1. идентификатор недели;
2. идентификатор семестра;
3. номер недели;

* Таблица «Пары» будет содержать в себе:

1. идентификатор пары;
2. идентификатор аудитории;
3. идентификатор группы;
4. идентификатор предмета;
5. идентификатор преподавателей;

* Таблица «Дни» будет содержать в себе:

1. идентификатор дня;
2. день недели;
3. идентификатор недели;
4. идентификатор первой пары;
5. идентификатор второй пары;
6. идентификатор третьей пары;
7. идентификатор четвертой пары;
8. идентификатор пятой пары;
9. идентификатор шестой пары;
10. идентификатор седьмой пары.

Таблица «Группы» имеет связь один ко многим с таблицей «Пары», так как у одной группы могут быть несколько пар.

Таблица «Преподаватели» имеет связь один ко многим с таблицей «Пары», так как у одного преподавателя могут быть несколько пар.

Таблица «Предметы» имеет связь один ко многим с таблицей «Пары», так как с одним предметом могут быть несколько пар.

Таблица «Типы аудитории» имеет связь один ко многим с таблицей «Аудитории», так как у одного типа аудитории могут быть несколько аудиторий.

Таблица «Аудитории» имеет связь один ко многим с таблицей «Пары», так как в одной аудитории могут быть несколько пар.

Таблица «Семестры» имеет связь один ко многим с таблицей «Недели», так как в одном семестре могут быть несколько недель.

Таблица «Недели» имеет связь один ко многим с таблицей «Дни», так как в одной неделе могут быть несколько дней.

Таблица «Пары» имеет связь многие к одному с таблицей «Дни», так как несколько пар может быть в одном дне.

Практические работы №3-7 требовали создать настольное приложение для составления расписания, модуль с Captcha, модуль для экспортирования данных в excel файл, пользовательский интерфейс приложения, чек-листы для тестирования данного программного продукта.

Прежде чем начать работу с приложением, пользователю необходимо пройти простую капчу (Приложение А, Рисунок 2), которая реализована для проверки, является ли пользователь человеком.

После приложение подключается к веб-сервису и может выполнять такие функции, как чтение, добавление, изменение и удаление информации о:

* семестрах;
* неделях;
* предметах;
* типах аудиторий;
* кабинетах;
* преподавателях;
* группах;
* парах;
* днях.

Важно отметить, что информацию о днях можно только добавлять и удалять, не редактировать. Приложение обеспечивает проверку вводимой информации на корректность, что помогает избежать ошибок при составлении расписания. К тому же благодаря использованию RESTful API приложение обладает гибкостью и может обновлять данные в реальном времени, отображая актуальное расписание.

Пользователям также доступна возможность экспорта данных в Excel файл для удобного сохранения и отображения информации. Необходимо иметь в виду, что процесс экспорта данных может иметь свои особенности и нюансы, связанные с форматированием и структурой Excel файла.

Пользовательский интерфейс разработан с использованием технологии WPF для удобной навигации и управления данными. С помощью этого приложения можно эффективно организовать рабочий процесс по составлению расписания, упростить управление данными и повысить точность информации. А также приложение обеспечит пользователя удобным и интуитивно понятным интерфейсом для работы с расписанием.

В тестирование настольного приложения входило:

* общая функциональность:

1. загрузка приложения без ошибок;
2. наличие всех основных компонентов интерфейса;
3. навигация между различными разделами приложения;
4. возможность добавления, изменения и удаления информации об объектах базы данных;

* проверка информации на корректность:

1. валидация вводимых данных на стороне клиента;
2. обработка ошибок ввода данных;

* взаимодействие с RESTful API:

1. проверка корректности отправки запросов к API;
2. обработка ответов API;

* экспорт данных в Excel:

1. возможность экспорта данных в Excel файл;
2. проверка корректности данных в экспортированном Excel файле;

* тестирование удаления данных:

1. проверка корректности удаления записей из различных сущностей, у которых нет связей в других таблицах;
2. проверка удаления записи, у которых есть связи в других таблицах;
3. убедиться, что запись осталась нетронутой;

* тестирование графического интерфейса:

1. проверка отображения данных;
2. проверка работы элементов управления;
3. адаптивность интерфейса под различные разрешения экранов;

* тестирование производительности:

1. проверка скорости отклика приложения при работе с большим объемом данных;
2. тестирование при работе с медленным интернет-соединением;

Практические работы №8-10 имели цель разработать мобильное приложение, пользовательский интерфейс приложения и его тестирование.

Для доступа к данным есть система авторизации(Приложение А, Рисунок 3), которая позволяет войти в приложение с использованием учетных данных, полученных через веб-сервис. Приложение предоставляет пользователю возможность просматривать информацию о расписании, получаемой через API, с использованием удобного и интуитивно понятного интерфейса.

Пользовательский интерфейс приложения разработан с фокусом на простоту и удобство использования. Все элементы интерфейса должны быть интуитивно понятными для пользователя. В итоге разработанное приложение предоставляет пользователю удобный, понятный и эффективный способ просмотра информации о расписании с возможностью фильтрации данных, авторизации и использования API для получения актуальных данных.

В тестирование мобильного приложения входило:

* авторизация:

1. вход с корректными данными происходит успешно;
2. проверка поведения при неправильно введенных данных;
3. проверка возможности выхода из учетной записи;

* загрузка данных из API:

1. успешная загрузка данных при корректной работе API;
2. обработка ошибок в случае недоступности API или ошибок соединения;

* отображение расписания:

1. верный вывод всех данных в удобочитаемом формате;
2. проверка внешнего вида данных (цвета, шрифты, размещение);

* фильтрация по поисковой строке:

1. фильтрация данных происходит корректно в зависимости от введенной строки;
2. проверка реакции интерфейса при отсутствии результатов поиска;

* навигация:

1. проверка работы выдвижной панели;
2. проверка перехода в личный кабинет;
3. проверка переходов между экранами;

Практическая работа №11 требовала составить короткую документацию проекта.

Веб-сервис разработан на ASP.NET Core Web API и обеспечивает взаимодействие с базой данных MS SQL Server для выполнения CRUD операций с использованием HTTP запросов. API предназначен для хранения данных, необходимых для составления расписания.

Особенности API (Приложение А, Рисунок 1) - При попытке удаления элемента API проверяет, связан ли данный элемент с другими таблицами. Если элемент уже используется в другой таблице, API возвращает ошибку и не позволяет удалить запись, чтобы избежать нарушения целостности данных.

Таким образом, этот RESTful API (Приложение А, Рисунок 1) обеспечивает безопасное взаимодействие с данными, предотвращая нецелевые действия при попытке удаления элементов, которые имеют зависимости в других таблицах.

Настольное приложение подключается к внешнему RESTful API (Приложение А, Рисунок 1) для получения необходимых данных, а после чего имеет возможности создавать отчеты и расписания в удобном для них формате. Оно имеет простой и понятной пользовательский интерфейс, разработанный с использованием технологии WPF.

В мобильном приложении для доступа к данным есть система авторизации, которая позволяет войти в приложение с использованием учетных данных, полученных через веб-сервис. Если данные отсутствуют, то будет следует пройти процесс регистрации, который находится в стадии разработки. Приложение предоставляет пользователю возможность просматривать информацию о расписании, получаемой через API, с использованием удобного и интуитивно понятного интерфейса.

Пользовательский интерфейс приложения разработан с фокусом на простоту и удобство использования. Все элементы интерфейса должны быть интуитивно понятными для пользователя. В итоге разработанное приложение предоставляет пользователю удобный и эффективный способ просмотра информации о расписании с возможностью авторизации и использования API для получения актуальных данных.

# ЗАКЛЮЧЕНИЕ

В результате прохождения учебной практики был получен практический опыт в разработке кода программного продукта на основе готовой спецификации на уровне модуля, использовании инструментальных средств на этапе отладки программного продукта, проведении тестирования программного модуля по определенному сценарию, разработке мобильных приложений, а также были приобретены навыки осуществлять разработку кода программного модуля на языках низкого и высокого уровней, создавать программу по разработанному алгоритму как отдельный модуль, выполнять отладку и тестирование программы на уровне модуля, осуществлять разработку кода программного модуля на современных языках программирования, уметь выполнять оптимизацию и рефакторинг программного кода, оформлять документацию на программные средства.

Помимо этого, был получен опыт в работе с коллективом, способы выполнения профессиональных задач, оценены их эффективность и качество. Приобретен навык осуществление поиска и использования информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития, использование информационно-коммуникационных технологий в профессиональной деятельности.

Учебная практика позволила закрепить и углубить знания в сфере разработки модулей программного обеспечения для компьютерных систем.

# ПРИЛОЖЕНИЕ А

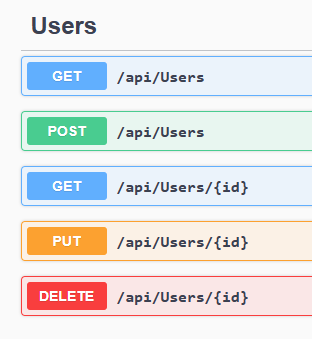


Рисунок 1 - API

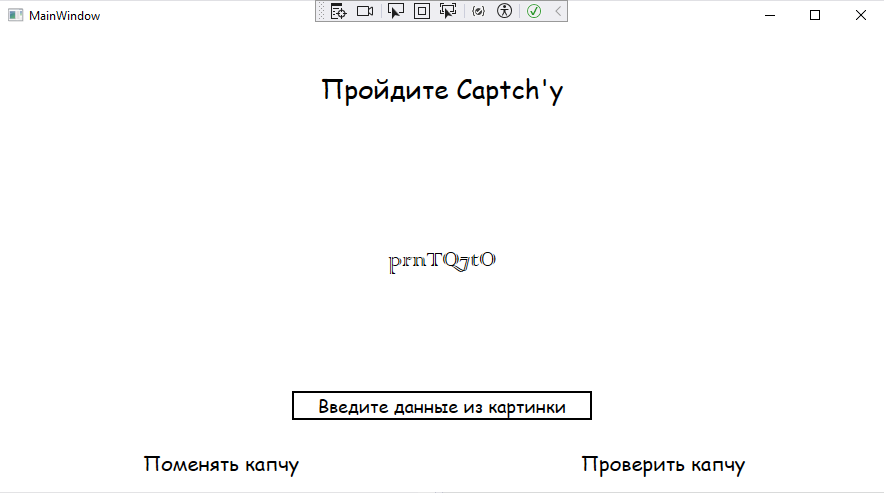


Рисунок 2 - Captha

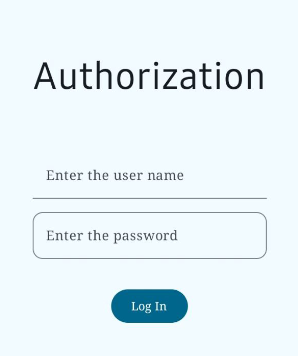


Рисунок 3 - Экран авторизации